

Акронис Инфозащита

Acronis Инфраструктура 3.5

Administrator's Command Line Guide

16 июля 2020 г.

Заявление об авторских правах

Авторские права ©ООО «Акронис-Инфозащита» 2020. Все права защищены.

Наименование Linux является зарегистрированным товарным знаком Линуса Торвальдса.

VMware и VMware Ready являются торговыми знаками и (или) зарегистрированными торговыми знаками компании VMware, Inc. в США и (или) других странах.

Windows и MS-DOS — зарегистрированные товарные знаки корпорации Майкрософт.

Все остальные упоминаемые товарные знаки могут быть зарегистрированными товарными знаками тех или иных фирм.

Распространение существенно измененных версий данного руководства запрещено без явного разрешения владельца авторских прав.

Распространение настоящих или переработанных материалов, входящих в данное руководство, в виде печатного издания (книги) запрещено без письменного разрешения их владельца.

ДОКУМЕНТАЦИЯ ПОСТАВЛЯЕТСЯ «КАК ЕСТЬ». НЕ СУЩЕСТВУЕТ НИКАКИХ ЯВНО ВЫРАЖЕННЫХ ИЛИ ПОДРАЗУМЕВАЕМЫХ ОБЯЗАТЕЛЬСТВ, ПОДТВЕРЖДЕНИЙ ИЛИ ГАРАНТИЙ, В ТОМ ЧИСЛЕ И СВЯЗАННЫХ С ТОВАРНОСТЬЮ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ ИЛИ ПРИГОДНОСТЬЮ ЕГО ДЛЯ ИСПОЛЬЗОВАНИЯ В ОПРЕДЕЛЕННЫХ ЦЕЛЯХ, НАСКОЛЬКО ТАКАЯ ОГРАНИЧЕННОСТЬ ОТВЕТСТВЕННОСТИ ДОПУСКАЕТСЯ ЗАКОНОМ.

С ПО или Услугой может быть предоставлен исходный код сторонних производителей. Лицензии этих сторонних производителей подробно описаны в файле license.txt, находящемся в корневом каталоге установки. Обновляемый список кода сторонних производителей и условия лицензии, применимые к программному обеспечению и/или службе, см. по адресу <http://kb.acronis.com/content/7696>.

Оглавление

1. Введение	1
1.1 Указание учетных данных	2
1.2 Управление заданиями	2
2. Управление кластером хранилища данных	4
2.1 Управление токенами	4
2.1.1 <code>vinfra node token show</code>	4
2.1.2 <code>vinfra node token create</code>	4
2.1.3 <code>vinfra node token validate</code>	5
2.2 Управление типами трафика и сетями	5
2.2.1 <code>vinfra cluster traffic-type create</code>	5
2.2.2 <code>vinfra cluster traffic-type list</code>	6
2.2.3 <code>vinfra cluster traffic-type show</code>	7
2.2.4 <code>vinfra cluster traffic-type set</code>	7
2.2.5 <code>vinfra cluster traffic-type delete</code>	8
2.2.6 <code>vinfra cluster network create</code>	8
2.2.7 <code>vinfra cluster network list</code>	9
2.2.8 <code>vinfra cluster network show</code>	9
2.2.9 <code>vinfra cluster network set</code>	10
2.2.10 <code>vinfra cluster network set-bulk</code>	11
2.2.11 <code>vinfra cluster network delete</code>	12
2.3 Управление серверами хранения	13
2.3.1 <code>vinfra node join</code>	13
2.3.2 <code>vinfra node list</code>	14
2.3.3 <code>vinfra node show</code>	15
2.3.4 <code>vinfra node maintenance precheck</code>	16
2.3.5 <code>vinfra node maintenance start</code>	16

2.3.6	vinfra node maintenance status	18
2.3.7	vinfra node maintenance stop	19
2.3.8	vinfra node release	19
2.3.9	vinfra node forget	20
2.4	Управление сетевыми интерфейсами сервера	21
2.4.1	vinfra node iface list	21
2.4.2	vinfra node iface show	22
2.4.3	vinfra node iface up	23
2.4.4	vinfra node iface down	24
2.4.5	vinfra node iface set	25
2.4.6	vinfra node iface create-bond	28
2.4.7	vinfra node iface create-vlan	31
2.4.8	vinfra node iface delete	34
2.5	Управление дисками серверов	35
2.5.1	vinfra node disk list	35
2.5.2	vinfra node disk show	35
2.5.3	vinfra node disk assign	36
2.5.4	vinfra node disk release	38
2.5.5	vinfra node disk blink on	39
2.5.6	vinfra node disk blink off	40
2.5.7	vinfra node iscsi target add	40
2.5.8	vinfra node iscsi target delete	41
2.6	Создание и удаление кластера хранилища данных	42
2.6.1	vinfra cluster create	42
2.6.2	vinfra cluster delete	44
2.7	Вывод обзора и подробных данных о кластере хранилища	44
2.7.1	vinfra cluster overview	44
2.7.2	vinfra cluster show	46
3.	Управление вычислительным кластером	47
3.1	Создание и удаление вычислительного кластера	47
3.1.1	vinfra service compute create	47
3.1.2	vinfra service compute delete	49
3.2	Вывод обзора и подробных данных о вычислительном кластере	50
3.2.1	vinfra service compute show	50
3.2.2	vinfra service compute stat	52

3.3	Изменение параметров вычислительного кластера	52
3.4	Управление вычислительными серверами	53
3.4.1	vinfra service compute node add	53
3.4.2	vinfra service compute node list	54
3.4.3	vinfra service compute node show	55
3.4.4	vinfra service compute node fence	55
3.4.5	vinfra service compute node unfence	56
3.4.6	vinfra service compute node release	56
3.5	Управление виртуальными машинами	57
3.5.1	vinfra service compute server create	57
3.5.2	vinfra service compute server list	60
3.5.3	vinfra service compute server show	60
3.5.4	vinfra service compute server stat	61
3.5.5	vinfra service compute server set	62
3.5.6	vinfra service compute server iface attach	63
3.5.7	vinfra service compute server iface list	64
3.5.8	vinfra service compute server iface detach	65
3.5.9	vinfra service compute server volume attach	65
3.5.10	vinfra service compute server volume list	66
3.5.11	vinfra service compute server volume show	66
3.5.12	vinfra service compute server volume detach	67
3.5.13	vinfra service compute server log	67
3.5.14	vinfra service compute server migrate	68
3.5.15	vinfra service compute server resize	68
3.5.16	vinfra service compute server start	69
3.5.17	vinfra service compute server pause	69
3.5.18	vinfra service compute server unpause	69
3.5.19	vinfra service compute server suspend	70
3.5.20	vinfra service compute server resume	70
3.5.21	vinfra service compute server reboot	71
3.5.22	vinfra service compute server reset-state	71
3.5.23	vinfra service compute server stop	72
3.5.24	vinfra service compute server shelve	72
3.5.25	vinfra service compute server unshelve	72
3.5.26	vinfra service compute server evacuate	73

3.5.27	vinfra service compute server delete	73
3.6	Управление образами	74
3.6.1	vinfra service compute image create	74
3.6.2	vinfra service compute image list	75
3.6.3	vinfra service compute image show	75
3.6.4	vinfra service compute image set	76
3.6.5	vinfra service compute image save	77
3.6.6	vinfra service compute image delete	78
3.7	Управление размещениями	78
3.7.1	vinfra service compute placement create	78
3.7.2	vinfra service compute placement assign	79
3.7.3	vinfra service compute placement delete assign	80
3.7.4	vinfra service compute placement list	81
3.7.5	vinfra service compute placement show	81
3.7.6	vinfra service compute placement update	82
3.7.7	vinfra service compute placement delete	82
3.8	Управление типами виртуальных машин	83
3.8.1	vinfra service compute flavor create	83
3.8.2	vinfra service compute flavor list	83
3.8.3	vinfra service compute flavor show	84
3.8.4	vinfra service compute flavor delete	84
3.9	Управление SSH-ключами вычислительного кластера	85
3.9.1	vinfra service compute key create	85
3.9.2	vinfra service compute key list	86
3.9.3	vinfra service compute key show	86
3.9.4	vinfra service compute key delete	87
3.10	Управление виртуальными сетями	87
3.10.1	vinfra service compute network create	87
3.10.2	vinfra service compute network list	89
3.10.3	vinfra service compute network show	89
3.10.4	vinfra service compute network set	90
3.10.5	vinfra service compute network delete	91
3.11	Управление виртуальными маршрутизаторами	91
3.11.1	vinfra service compute router create	91
3.11.2	vinfra service compute router list	93

3.11.3	vinfra service compute router show	93
3.11.4	vinfra service compute router set	94
3.11.5	vinfra service compute router iface add	95
3.11.6	vinfra service compute router iface list	96
3.11.7	vinfra service compute router iface remove	96
3.11.8	vinfra service compute router delete	97
3.12	Управление плавающими IP-адресами	97
3.12.1	vinfra service compute floatingip create	97
3.12.2	vinfra service compute floatingip list	98
3.12.3	vinfra service compute floatingip show	99
3.12.4	vinfra service compute floatingip set	100
3.12.5	vinfra service compute floatingip delete	101
3.13	Управление балансировщиками нагрузки	101
3.13.1	vinfra service compute load-balancer create	101
3.13.2	vinfra service compute load-balancer list	103
3.13.3	vinfra service compute load-balancer show	103
3.13.4	vinfra service compute load-balancer stats	104
3.13.5	vinfra service compute load-balancer set	105
3.13.6	vinfra service compute load-balancer pool create	106
3.13.7	vinfra service compute load-balancer pool list	109
3.13.8	vinfra service compute load-balancer pool show	109
3.13.9	vinfra service compute load-balancer pool set	111
3.13.10	vinfra service compute load-balancer pool delete	113
3.13.11	vinfra service compute load-balancer delete	114
3.14	Управление томами	114
3.14.1	vinfra service compute volume create	114
3.14.2	vinfra service compute volume list	116
3.14.3	vinfra service compute volume show	116
3.14.4	vinfra service compute volume set	117
3.14.5	vinfra service compute volume extend	118
3.14.6	vinfra service compute volume delete	119
3.15	Управление моментальными снимками томов	119
3.15.1	vinfra service compute volume snapshot create	119
3.15.2	vinfra service compute volume snapshot list	120
3.15.3	vinfra service compute volume snapshot show	120

3.15.4	vinfra service compute volume snapshot set	121
3.15.5	vinfra service compute volume snapshot upload-to-image	122
3.15.6	vinfra service compute volume snapshot revert	123
3.15.7	vinfra service compute volume snapshot reset-state	123
3.15.8	vinfra service compute volume snapshot delete	124
3.16	Управление политиками хранилища	124
3.16.1	vinfra cluster storage-policy create	124
3.16.2	vinfra cluster storage-policy list	125
3.16.3	vinfra cluster storage-policy show	126
3.16.4	vinfra cluster storage-policy set	126
3.16.5	vinfra cluster storage-policy delete	127
3.17	Управление кластерами Kubernetes	128
3.17.1	vinfra service compute k8saas create	128
3.17.2	vinfra service compute k8saas list	130
3.17.3	vinfra service compute k8saas config	131
3.17.4	vinfra service compute k8saas show	131
3.17.5	vinfra service compute k8saas set	132
3.17.6	vinfra service compute k8saas delete	133
3.18	Управление квотами вычислительных ресурсов	134
3.18.1	vinfra service compute quotas show	134
3.18.2	vinfra service compute quotas update	134
3.19	Управление обновлениями	135
3.19.1	vinfra software-updates check-for-updates	135
3.19.2	vinfra software-updates eligibility-check	136
3.19.3	vinfra software-updates download	137
3.19.4	vinfra software-updates start	138
3.19.5	vinfra software-updates pause	139
3.19.6	vinfra software-updates resume	139
3.19.7	vinfra software-updates cancel	140
3.19.8	vinfra software-updates status	141
4.	Управление резервным кластером	143
4.1	Создание, отображение и удаление резервного кластера	143
4.1.1	vinfra service backup cluster create	143
4.1.2	vinfra service backup cluster show	149
4.1.3	vinfra service backup cluster release	150

4.2	Управление резервными серверами	151
4.2.1	vinfra service backup node add	151
4.2.2	vinfra service backup node list	152
4.2.3	vinfra service backup node release	152
4.3	Обновление сертификатов резервного кластера	153
4.4	Изменение параметров хранилища	154
4.4.1	vinfra service backup storage-params show	154
4.4.2	vinfra service backup storage-params change	155
4.5	Изменение параметров тома	158
4.5.1	vinfra service backup volume-params show	158
4.5.2	vinfra service backup volume-params change	159
4.6	Управление георепликацией резервного кластера	160
4.6.1	vinfra service backup geo-replication show	161
4.6.2	vinfra service backup geo-replication master setup	161
4.6.3	vinfra service backup geo-replication master download-configs	162
4.6.4	vinfra service backup geo-replication slave setup	162
4.6.5	vinfra service backup geo-replication master establish	163
4.6.6	vinfra service backup geo-replication slave update-certificates	164
4.6.7	vinfra service backup geo-replication master disable	164
4.6.8	vinfra service backup geo-replication slave promote-to-master	165
4.6.9	vinfra service backup geo-replication slave cancel	166
4.6.10	vinfra service backup geo-replication master cancel	166
5.	Управление общими параметрами	168
5.1	Управление лицензиями	168
5.1.1	vinfra cluster license load	168
5.1.2	vinfra cluster license show	169
5.2	Управление доменами	169
5.2.1	vinfra domain create	169
5.2.2	vinfra domain list	170
5.2.3	vinfra domain show	170
5.2.4	vinfra domain set	171
5.2.5	vinfra domain delete	172
5.3	Управление пользователями домена	172
5.3.1	vinfra domain user list-available-roles	172
5.3.2	vinfra domain user create	173

5.3.3	vinfra domain user list175
5.3.4	vinfra domain user show175
5.3.5	vinfra domain user set176
5.3.6	vinfra domain user delete177
5.4	Управление проектами домена178
5.4.1	vinfra domain project create178
5.4.2	vinfra domain project list179
5.4.3	vinfra domain project show179
5.4.4	vinfra domain project set180
5.4.5	vinfra domain project user list181
5.4.6	vinfra domain project user remove181
5.4.7	vinfra domain project delete182
5.5	Управление SSH-ключами182
5.5.1	vinfra cluster sshkey add182
5.5.2	vinfra cluster sshkey list184
5.5.3	vinfra cluster sshkey delete184
5.6	Управление внешними серверами DNS185
5.6.1	vinfra cluster settings dns show185
5.6.2	vinfra cluster settings dns set186
5.7	Настройка высокой доступности сервера управления186
5.7.1	vinfra cluster ha create186
5.7.2	vinfra cluster ha update188
5.7.3	vinfra cluster ha show190
5.7.4	vinfra cluster ha delete190
5.8	Управление резервными копиями кластера191
5.8.1	vinfra cluster backup create191
5.8.2	vinfra cluster backup show192
5.9	Управление шифрованием уровней хранилища192
5.9.1	vinfra cluster settings encryption show192
5.9.2	vinfra cluster settings encryption set193
5.10	Управление автоматической конфигурацией дисков хранилища193
5.10.1	vinfra cluster settings automatic-disk-replacement show193
5.10.2	vinfra cluster settings automatic-disk-replacement set194
5.11	Управление оповещениями195
5.11.1	vinfra cluster alert list195

5.11.2	vinfra cluster alert show	196
5.11.3	vinfra cluster alert delete	196
5.12	Управление журналом аудита	197
5.12.1	vinfra cluster auditlog list	197
5.12.2	vinfra cluster auditlog show	198
5.13	Отправка отчетов о неполадках	199
6.	Мониторинг кластера хранилища	201
6.1	Мониторинг общих параметров кластера хранилища данных	201
6.2	Мониторинг серверов метаданных	204
6.3	Мониторинг серверов фрагментов данных	206
6.3.1	Объяснение данных об использовании дискового пространства	208
6.3.1.1	Объяснение подлежащего выделению дискового пространства	209
6.3.1.2	Просмотр пространства, занятого фрагментами данных	211
6.3.2	Определение состояния фрагментов	211
6.4	Мониторинг клиентов	213
6.5	Мониторинг физических дисков	215
6.6	Мониторинг журналов событий	217
6.7	Мониторинг параметров репликации	223
7.	Доступ к кластерам хранилища через iSCSI	224
7.1	Обзор последовательности операций с iSCSI	226
7.1.1	Управление устаревшими целевыми устройствами iSCSI	227
7.2	Настройка инструмента командной строки	227
7.3	Управление группами целевых устройств	228
7.3.1	Создание групп целевых устройств	228
7.3.2	Запуск и остановка групп целевых устройств	230
7.3.3	Вывод списка групп целевых устройств	230
7.3.4	Печать сведений о группах целевых устройств	231
7.3.5	Управление постоянным резервированием групп целевых устройств	231
7.3.6	Удаление групп целевых устройств	232
7.4	Управление томами iSCSI	233
7.4.1	Создание томов iSCSI	233
7.4.2	Вывод и печать сведений о томах iSCSI	233
7.4.3	Присоединение томов iSCSI к группам целевых устройств	233
7.4.4	Установка активного/оптимизированного пути для томов iSCSI	234

7.4.5	Просмотр информации ALUA для тома iSCSI	234
7.4.6	Просмотр и настройка параметров тома iSCSI	235
7.4.7	Увеличение размера тома iSCSI	235
7.4.8	Установка ограничений для тома iSCSI	235
7.4.9	Отсоединение томов iSCSI от групп целевых устройств	236
7.4.10	Удаление томов iSCSI	236
7.5	Управление серверами в группах целевых устройств	236
7.5.1	Добавление серверов в группы целевых устройств	236
7.5.2	Вывод списка серверов в группах целевых устройств	237
7.5.3	Включение и отключение серверов в группах целевых устройств	238
7.5.4	Удаление серверов из групп целевых устройств	238
7.6	Управление целевыми устройствами и порталами	239
7.6.1	Создание целевых устройств	239
7.6.2	Добавление и удаление порталов целевых устройств	239
7.6.3	Удаление целевых устройств	240
7.7	Управление учетными записями CHAP	240
7.7.1	Создание и вывод списка учетных записей CHAP	241
7.7.2	Изменение сведений учетной записи CHAP	241
7.7.3	Назначение учетных записей CHAP группам целевых устройств	241
7.7.4	Удаление учетных записей CHAP	241
7.8	Управление представлениями LUN	242
7.8.1	Создание представлений LUN	242
7.8.2	Вывод списка представлений LUN	243
7.8.3	Изменение сведений о представлениях LUN	243
7.8.4	Удаление представлений LUN	243
8.	Расширенные задачи	244
8.1	Обновление ядра с помощью ReadyKernel	244
8.1.1	Автоматическая установка исправлений ReadyKernel	245
8.1.2	Управление исправлениями ReadyKernel вручную	245
8.1.2.1	Скачивание, установка и загрузка исправлений ReadyKernel	245
8.1.2.2	Загрузка и выгрузка исправлений ReadyKernel	246
8.1.2.3	Установка и удаление исправлений ReadyKernel для определенных ядер	246
8.1.2.4	Понижение версии исправлений ReadyKernel	246
8.1.2.5	Отключение загрузки исправлений ReadyKernel при загрузке машины	247
8.1.2.6	Управление журналами ReadyKernel	247

8.2	Управление дополнениями гостевой ОС247
8.2.1	Установка дополнений гостевой ОС247
8.2.1.1	Установка дополнений гостевой ОС в новые VM248
8.2.1.2	Установка дополнений гостевой ОС в существующие VM249
8.2.2	Удаление дополнений гостевой ОС251
8.2.2.1	Удаление дополнений гостевой ОС из VM Linux251
8.2.2.2	Удаление дополнений гостевой ОС из VM Windows252
8.3	Выполнение команд в виртуальных машинах без сетевого подключения253
8.3.1	Выполнение команд в виртуальных машинах Linux253
8.3.2	Выполнение команд в виртуальных машинах Windows254
8.4	Настройка модели ЦП виртуальных машин254
8.5	Создание шаблонов Linux255
8.6	Подключение к интерфейсу командной строки OpenStack257
8.7	Установка доменного имени для API вычислений258
8.8	Защита трафика API OpenStack с помощью SSL259
8.9	Использование учета для вычислительных ресурсов260
8.9.1	Включение сервиса учета262
8.9.2	Использование программы командной строки Gnocchi263
8.9.3	Просмотр использования ресурсов на уровне проекта265
8.10	Настройка политики памяти для сервисов хранилища265
8.10.1	vinfra memory-policy vstorage-services per-cluster change267
8.10.2	vinfra memory-policy vstorage-services per-cluster show268
8.10.3	vinfra memory-policy vstorage-services per-cluster reset268
8.10.4	vinfra memory-policy vstorage-services per-node change269
8.10.5	vinfra memory-policy vstorage-services per-node show270
8.10.6	vinfra memory-policy vstorage-services per-node reset271
8.11	Миграция виртуальных машин из VMware vCenter271
8.11.1	Развертывание VM устройства272
8.11.2	Настройка проверки подлинности в VM устройства273
8.11.3	Онлайн-миграция виртуальных машин в продукт Acronis Инфраструктура274
8.11.4	Офлайн-миграция виртуальных машин в продукт Acronis Инфраструктура275
8.12	Изменение типа VM по умолчанию для балансировщика нагрузки276
8.13	Configuring Retention Policy for Prometheus Metrics278
8.14	Viewing Cluster Logs278

ГЛАВА 1

Введение

В этом руководстве описываются параметры и синтаксис программы командной строки `vinfra`, которую можно использовать для управления продуктом Acronis Инфраструктура из консоли, а также для автоматизации подобных задач управления.

Примечание: Хотя в следующих главах представлены сведения о конкретных операциях, которые можно выполнять с помощью `vinfra`, вы также можете запустить команду `vinfra help`, чтобы получить список всех доступных команд с описаниями. Для получения справки по определенной команде выполните либо `vinfra help <command>`, либо `vinfra <command> --help`.

Кроме того, в этом руководстве показано, как использовать командную строку для выполнения операций, которые в настоящее время не поддерживаются программой `vinfra`.

Обратите внимание, что следующие операции не следует выполнять из командной строки:

- установка пользовательских путей для сервисов продукта Acronis Инфраструктура, в частности:
 - создание только кластеров S3 в `/mnt/vstorage/vols/s3`,
 - создание только целевых устройств iSCSI в `/mnt/vstorage/vols/iscsi`,
- подключение кластеров или изменение параметров подключения кластеров,
- настройка брандмауэра с помощью `firewall-cmd`,
- переименование сетевых подключений,
- управление сервисами MDS/CS,
- управление разделами, томами LVM или программными RAID-массивами,

- изменение файлов в каталогах `/mnt/vstorage/vols` и `/mnt/vstorage/webcp/backup`,
- установка шифрования или репликации корня кластера.

1.1 Указание учетных данных

Для программы командной строки `vinfra` требуется следующая информация:

- IP-адрес или имя хоста сервера управления (по умолчанию установлено значение `backend-api.svc.vstoragedomain`).
- Имя пользователя (по умолчанию `admin`).
- Пароль (созданный при установке продукта Acronis Инфраструктура).

Эта информация может предоставляться посредством параметров командной строки `--vinfra-portal`, `--vinfra-username` и `--vinfra-password` с каждой командой. Как вариант, для этого можно настроить переменные среды `VINFRA_PORTAL`, `VINFRA_USERNAME` и `VINFRA_PASSWORD` (например, в вашем каталоге `~/.bash_profile`). В этом случае можно будет работать с программой командной строки без вышеупомянутых параметров.

Поскольку `vinfra`, как правило, запускается с сервера управления от имени администратора, из переменных обычно требуется задать только пароль. Например:

```
# export VINFRA_PASSWORD=12345
```

Если вы установили `vinfra` на удаленной машине и/или запускаете программу как другой пользователь, на этой машине потребуется задать переменные `VINFRA_PORTAL` и/или `VINFRA_USERNAME` в дополнение к `VINFRA_PASSWORD`.

1.2 Управление заданиями

Программа командной строки `vinfra` выполняет некоторые команды сразу, а для других команд (выполнение которых может занять некоторое время) создает системные задания, которые ставятся в очередь. Примеры действий, выполняемых посредством заданий: создание кластера хранилища или вычислительного кластера и добавление в него серверов.

Для отслеживания заданий, выполняемых программой `vinfra`, используйте команды `vinfra task list` и `vinfra task show`. Например:

```
# vinfra task list
+-----+-----+
| task_id          | state  | name
+-----+-----+
| 8fc27e7a-ba73-471d-9134-e351e1137cf4 | success | backend.tasks.cluster.CreateNewCluster
| e61377db-9df4-4282-99aa-6a4ae73a7f96 | success | backend.tasks.disks.ApplyDiskRoleTask
| a005b748-cb85-40f8-a09d-291a8599bb9c | success | backend.tasks.node.AddNodeInClusterTask
+-----+-----+
# vinfra task show 8fc27e7a-ba73-471d-9134-e351e1137cf4
+-----+-----+
| Field  | Value
+-----+-----+
| args   | - stor1
|        | - 7ffa9540-5a20-41d1-b203-e3f349d62565
|        | - null
|        | - null
| kwargs | {}
| name   | backend.tasks.cluster.CreateNewCluster
| result | cluster_id: 1
| state  | success
| task_id | 8fc27e7a-ba73-471d-9134-e351e1137cf4
+-----+-----+
```


ГЛАВА 2

Управление кластером хранилища данных

2.1 Управление токенами

2.1.1 `vinfra node token show`

Отображение токена внутреннего хранилища:

```
usage: vinfra node token show
```

Пример:

```
# vinfra node token show
+-----+-----+
| Field | Value |
+-----+-----+
| host  | 10.37.130.101 |
| token | dc56d4d2 |
| ttl   | 86398 |
+-----+-----+
```

Эта команда отображает подробные данные текущего токена.

2.1.2 `vinfra node token create`

Создание токена внутреннего хранилища:

```
usage: vinfra node token create [--ttl <ttl>]
```

--ttl <время_существования>

Время существования (TTL) токена в секундах

Пример:

```
# vinfra node token create --ttl 86400
+-----+-----+
| Field | Value |
+-----+-----+
| host  | 10.37.130.101 |
| token | dc56d4d2 |
| ttl   | 86398 |
+-----+-----+
```

Эта команда создает новый токен со временем существования (TTL) 86 400 секунд.

2.1.3 vinfra node token validate

Проверка токена внутреннего хранилища:

```
usage: vinfra node token validate <token>
```

<токен>

Значение токена

Пример:

```
# vinfra node token validate dc56d4d2
+-----+-----+
| Field | Value |
+-----+-----+
| status | valid |
+-----+-----+
```

Эта команда проверяет токен dc56d4d2.

2.2 Управление типами трафика и сетями

2.2.1 vinfra cluster traffic-type create

Создание нового типа трафика:

```
usage: vinfra cluster traffic-type create --port <port> <traffic-type-name>
```

```
--port <порт>
```

Порт для типа трафика

```
<имя_типа_трафика>
```

Имя типа трафика

Пример:

```
# vinfra cluster traffic-type create "MyTrafficType" --port 6900
+-----+-----+
| Field   | Value           |
+-----+-----+
| exclusive | False           |
| name      | MyTrafficType   |
| port      | 6900            |
| type      | custom          |
+-----+-----+
```

Эта команда создает пользовательский тип трафика MyTrafficType на порту 6900.

2.2.2 vinfra cluster traffic-type list

Список доступных типов трафика:

```
usage: vinfra cluster traffic-type list
```

Пример:

```
# vinfra cluster traffic-type list
+-----+-----+-----+-----+
| name           | type           | exclusive | port |
+-----+-----+-----+-----+
| Storage        | predefined     | True      |      |
| Internal management | predefined     | True      |      |
| OSTOR private  | predefined     | True      |      |
| S3 public      | predefined     | False     |      |
| iSCSI          | predefined     | False     |      |
| NFS            | predefined     | False     |      |
| ABGW private   | predefined     | True      |      |
| ABGW public    | predefined     | False     |      |
| Admin panel    | predefined     | False     |      |
| SSH            | predefined     | False     |      |
| VM public      | predefined     | False     |      |
| VM private     | predefined     | True      |      |
| Compute API    | predefined     | True      |      |
```

```
| MyTrafficType      | custom      | False      | 6900 |
+-----+-----+-----+-----+
```

Эта команда выдает список всех доступных в Acronis Инфраструктура типов трафика.

2.2.3 vinfra cluster traffic-type show

Отображение подробных данных для типа трафика:

```
usage: vinfra cluster traffic-type show <traffic-type>
```

<тип_трафика>

Имя типа трафика

Пример:

```
# vinfra cluster traffic-type show Storage
+-----+-----+
| Field      | Value      |
+-----+-----+
| exclusive  | True       |
| name       | Storage    |
| port       |            |
| type       | predefined  |
+-----+-----+
```

Эта команда отображает подробные данные о типе трафика Storage (Хранилище).

2.2.4 vinfra cluster traffic-type set

Изменение параметров типа трафика:

```
usage: vinfra cluster traffic-type set [--name <name>] [--port <port>] <traffic-type>
```

--name <имя>

Новое имя для типа трафика

--port <порт>

Новый порт для типа трафика

<тип_трафика>

Имя типа трафика

Пример:

```
# vinfra cluster traffic-type set "MyTrafficType" \
--name "MyOtherTrafficType" --port 6901
+-----+-----+
| Field   | Value           |
+-----+-----+
| exclusive | False           |
| name     | MyOtherTrafficType |
| port    | 6901            |
| type    | custom          |
+-----+-----+
```

Эта команда переименует тип трафика MyTrafficType в MyOtherTrafficType и меняет его порт на 6901.

2.2.5 vinfra cluster traffic-type delete

Удаление типа трафика:

```
usage: vinfra cluster traffic-type delete <traffic-type>
```

<тип_трафика>

Имя типа трафика

Пример:

```
# vinfra cluster traffic-type delete "MyOtherTrafficType"
Operation successful
```

Эта команда удаляет пользовательский тип трафика MyOtherTrafficType.

2.2.6 vinfra cluster network create

Создание новой сети:

```
usage: vinfra cluster network create [--traffic-types <traffic-types>] <network-name>
```

--traffic-types <типы_трафика>

Разделенный запятыми список идентификаторов или имен типов трафика

<имя_сети>

Имя сети

Пример:

```
# vinfra cluster network create MyNet --traffic-types ssh
+-----+-----+
| Field | Value |
+-----+-----+
| id    | 03d5eeb3-1833-4626-885d-dd066635f5de |
| name  | MyNet |
| roles | - SSH |
| type  | Custom |
+-----+-----+
```

Эта команда создает пользовательскую сеть MyNet и назначает ей тип трафика SSH.

2.2.7 vinfra cluster network list

Отображение списка доступных сетей:

```
usage: vinfra cluster network list
```

Пример:

```
# vinfra cluster network list
+-----+-----+-----+
| id | name | roles |
+-----+-----+-----+
| 358bdc39-cd8b-4565-8ebf-e7c12dcd1cf7 | Public | - ABGW public  
| | | - iSCSI  
| | | - NFS  
| | | - S3 public  
| | | - SSH  
| 6095a997-e5f1-493d-a750-41ddf277153b | Private | - Admin Panel  
| | | - ABGW private  
| | | - Internal Management  
| | | - OSTOR private  
| | | - SSH  
| | | - Storage |
+-----+-----+-----+
```

Эта команда выдает список всех сетей в Acronis Инфраструктура.

2.2.8 vinfra cluster network show

Отображение подробных данных о сети:

```
usage: vinfra cluster network show <network>
```

<сеть>

Идентификатор или имя сети

Пример:

```
# vinfra cluster network show MyNet
+-----+-----+
| Field | Value |
+-----+-----+
| id    | 03d5eeb3-1833-4626-885d-dd066635f5de |
| name  | MyNet |
| roles | - SSH |
| type  | Custom |
+-----+-----+
```

Эта команда отображает подробные данные о пользовательской сети MyNet.

2.2.9 vinfra cluster network set

Изменение параметров сети:

```
usage: vinfra cluster network set [--name <network-name>]
                                   [--traffic-types <traffic-types> |
                                   --add-traffic-types <traffic-types> |
                                   --del-traffic-types <traffic-types>]
                                   <network>
```

--name <имя_сети>

Имя сети

--traffic-types <типы_трафика>

Разделенный запятыми список имен типов трафика (которые заменяют собой текущие типы трафика в сети)

--add-traffic-types <типы_трафика>

Разделенный запятыми список имен типов трафика (указанные типы трафика добавляются к сети)

--del-traffic-types <типы_трафика>

Разделенный запятыми список имен типов трафика (указанные типы трафика удаляются из сети)

<сеть>

Идентификатор или имя сети

Пример:

```
# vinfra cluster network set MyNet --name MyOtherNet --add-traffic-types iscsi,nfs
+-----+-----+
| Field  | Value                                |
+-----+-----+
| task_id | b29f6f66-37d7-47de-b02e-9f4087ad932b |
+-----+-----+
```

Эта команда создает задачу для переименования сети MyNet в сеть MyOtherNet и назначения ей типов трафика iSCSI и NFS.

Результат выполнения задачи:

```
# vinfra task show b29f6f66-37d7-47de-b02e-9f4087ad932b
+-----+-----+
| Field  | Value                                |
+-----+-----+
| args    | - 03d5eeb3-1833-4626-885d-dd066635f5de |
| kwargs  | name: MyOtherNet                      |
|         | roles:                                |
|         | - ssh                                 |
|         | - iscsi                               |
|         | - nfs                                 |
| name    | backend.presentation.network.roles.tasks.RolesSetChangeTask |
| result  | id: 03d5eeb3-1833-4626-885d-dd066635f5de |
|         | name: MyOtherNet                      |
|         | roles:                                |
|         | - iSCSI                               |
|         | - NFS                                 |
|         | - SSH                                 |
|         | type: Custom                          |
| state   | success                                |
| task_id | b29f6f66-37d7-47de-b02e-9f4087ad932b |
+-----+-----+
```

2.2.10 vinfra cluster network set-bulk

Изменение типов трафика многих сетей одновременно:

```
usage: vinfra cluster network set-bulk --network <network>:<traffic-types>
```

```
--network <сеть>:<типы_трафика>
```

Конфигурация сети в формате:

- <сеть>: идентификатор или имя сети.
- <типы_трафика>: разделенный запятыми список имен типов трафика (этот параметр может использоваться многократно).

Пример:

```
# vinfra cluster network set-bulk --network MyNet1:snmp --network MyNet2:ssh,snmp
+-----+-----+
| Field   | Value                                     |
+-----+-----+
| task_id | c774f55d-c45b-42cd-ac9e-16fc196e9283 |
+-----+-----+
```

Эта команда создает задачу для изменения типа трафика, заданного в сети MyNet1, на SNMP, а заданного в сети MyNet2 — на SSH и SNMP.

Результат выполнения задачи:

```
# vinfra task show c774f55d-c45b-42cd-ac9e-16fc196e9283
+-----+-----+
| Field   | Value                                     |
+-----+-----+
| details |                                           |
| name    | backend.presentation.network.roles.tasks.RolesSetBulkChangeTask |
| result  | - id: adf49487-9deb-4180-bb0c-08a906257981 |
|         |   name: MyNet1 |
|         |   roles: |
|         |     - SNMP |
|         |   type: Custom |
|         | - id: 3f6ff4a3-31bc-440b-a36f-d755c80d5932 |
|         |   name: MyNet2 |
|         |   roles: |
|         |     - SNMP |
|         |     - SSH |
|         |   type: Custom |
| state   | success |
| task_id | c774f55d-c45b-42cd-ac9e-16fc196e9283 |
+-----+-----+
```

2.2.11 vinfra cluster network delete

Удаление сети:

```
usage: vinfra cluster network delete <network>
```

<сеть>

Идентификатор или имя сети

Пример:

```
# vinfra cluster network delete MyOtherNet
Operation successful
```



```
--disk sdc:cs
+-----+-----+
| Field   | Value                                     |
+-----+-----+
| task_id | a2713068-9544-4ea1-8ec8-69a068cf86f2 |
+-----+-----+
```

Эта команда создает задачу для добавления сервера с идентификатором f59dabdb-bd1c-4944-8af2-26b8fe9ff8d4 в кластер хранения и назначает роли дискам: mds-system диску sda, cs дискам sdb и sdc.

Результат выполнения задачи:

```
# vinfra task show a2713068-9544-4ea1-8ec8-69a068cf86f2
+-----+-----+
| Field   | Value                                     |
+-----+-----+
| args    | - f59dabdb-bd1c-4944-8af2-26b8fe9ff8d4 |
|         | - 1                                       |
| kwargs  | disks:                                    |
|         | - id: 85F32403-94A9-465A-9E6C-C1A2B41294FC |
|         |   role: mds-system                       |
|         |   service_params: {}                    |
|         | - id: FE0B5876-E054-489B-B0FD-72429BEFD46A |
|         |   role: cs                               |
|         |   service_params: {}                    |
|         | - id: D3BEF4BB-AA3B-4DB6-9376-BC7CDA636700 |
|         |   role: cs                               |
|         |   service_params: {}                    |
| name    | backend.tasks.node.AddNodeInClusterTask |
| result  | {}                                        |
| state   | success                                  |
| task_id | a2713068-9544-4ea1-8ec8-69a068cf86f2 |
+-----+-----+
```

2.3.2 vinfra node list

Перечисление серверов хранения:

```
usage: vinfra node list
```

Пример:

```
# vinfra node list
+-----+-----+-----+-----+-----+-----+
| id          | host          | is_primary | is_online | is_assigned | is_in_h |
+-----+-----+-----+-----+-----+-----+
| 09bb6b84-70a5-41ae-b342- | node001.<...> | True      | True     | True       | False  |
```

```

| 23e5fc7cc126 | | | | | |
| 187edb11-38c5-487b-bd7f- | node002.<...> | False | True | True | False
| 57b0fa4b733c | | | | | |
| e6255aed-d6e7-41b2-ba90- | node003.<...> | False | True | True | False
| 86164c1cd9a6 | | | | | |
+-----+-----+-----+-----+-----+

```

Эта команда выдает список всех серверов, зарегистрированных в Acronis Инфраструктура (как неназначенных, так и используемых в кластере хранения).

2.3.3 vinfra node show

Отображение данных сервера хранения:

```
usage: vinfra node show <node>
```

<сервер>

Идентификатор или имя хоста сервера

Пример:

```

# vinfra node show 4f96acf5-3bc8-4094-bcb6-4d1953be7b55
+-----+-----+
| Field      | Value |
+-----+-----+
| cpu_cores  | 2     |
| host       | stor-1.example.com.vstoragedomain. |
| id         | 4f96acf5-3bc8-4094-bcb6-4d1953be7b55 |
| ipaddr     | stor-1.example.com.vstoragedomain. |
| is_assigned | False |
| is_in_ha   | False |
| is_installing | False |
| is_online  | True  |
| is_primary | True  |
| is_virt    | True  |
| mem_total  | 8201310208 |
| roles      | management: |
|            |   is_primary: true |
| tasks     | |
+-----+-----+

```

Эта команда показывает подробные данные сервера с идентификатором 4f96acf5-3bc8-4094-bcb6-4d1953be7b55.

2.3.4 vinfra node maintenance precheck

Запуск предварительной проверки обслуживания сервера:

```
usage: vinfra node maintenance precheck <node>
```

<сервер>

Идентификатор или имя хоста сервера

Пример:

```
# vinfra node maintenance precheck 9dcc9632-911c-4cc5-9a89-5a6fa5db2314
+-----+-----+
| Field   | Value                                     |
+-----+-----+
| task_id | 7c7f0afa-10f4-41b7-9b2e-973f3d392178 |
+-----+-----+
```

Эта команда создает задачу для запуска предварительной проверки обслуживания для сервера с идентификатором f59dabdb-bd1c-4944-8af2-26b8fe9ff8d4.

Результат выполнения задачи:

```
# vinfra task show 7c7f0afa-10f4-41b7-9b2e-973f3d392178
+-----+-----+
| Field   | Value                                     |
+-----+-----+
| details | |
| name    | backend.business.models.maintenance.tasks.MaintenancePrecheckTask |
| result  | |
| state   | success                                   |
| task_id | 7c7f0afa-10f4-41b7-9b2e-973f3d392178 |
+-----+-----+
```

2.3.5 vinfra node maintenance start

Запуск обслуживания сервера:

```
usage: vinfra node maintenance start [--iscsi-mode <mode>] [--compute-mode <mode>]
                                     [--s3-mode <mode>] [--storage-mode <mode>]
                                     [--alua-mode <mode>] [--nfs-mode <mode>] <node>
```

--iscsi-mode <режим>

Игнорировать эвакуацию iSCSI во время обслуживания (ignore).

--compute-mode <режим>

Игнорировать эвакуацию вычислительных серверов во время обслуживания (ignore).

--s3-mode <режим>

Игнорировать эвакуацию S3 во время обслуживания (ignore).

--storage-mode <режим>

Игнорировать эвакуацию серверов хранения во время обслуживания (ignore).

--alua-mode <режим>

Игнорировать группы целевых устройств блочного хранения данных во время обслуживания (ignore).

--nfs-mode <режим>

Игнорировать эвакуацию серверов NFS во время обслуживания (ignore).

<сервер>

Идентификатор или имя хоста сервера

Пример:

```
# vinfra node maintenance start 9dcc9632-911c-4cc5-9a89-5a6fa5db2314 \
--iscsi-mode ignore --compute-mode ignore
+-----+-----+
| Field  | Value |
+-----+-----+
| task_id | 3d4c23b6-9f62-412c-ad7c-ec9537a36fa7 |
+-----+-----+
```

Эта команда создает задачу для запуска обслуживания для сервера с идентификатором 9dcc9632-911c-4cc5-9a89-5a6fa5db2314 без эвакуации его сервисов iSCSI и вычислительных сервисов.

Результат выполнения задачи:

```
# vinfra task show 3d4c23b6-9f62-412c-ad7c-ec9537a36fa7
+-----+-----+
| Field  | Value |
+-----+-----+
| details |       |
| name    | backend.business.models.maintenance.tasks.MaintenanceStartTask |
| result  |       |
| state   | success |
| task_id | 3d4c23b6-9f62-412c-ad7c-ec9537a36fa7 |
+-----+-----+
```

2.3.6 vinfra node maintenance status

Отображение данных об обслуживании сервера:

```
usage: vinfra node maintenance status <node>
```

<сервер>

Идентификатор или имя хоста сервера

Пример:

```
# vinfra node maintenance status 9dcc9632-911c-4cc5-9a89-5a6fa5db2314
+-----+-----+
| Field   | Value                                     |
+-----+-----+
| node_id | 9dcc9632-911c-4cc5-9a89-5a6fa5db2314   |
| params  | compute_mode: ignore                    |
|         | iscsi_mode: ignore                      |
|         | nfs_mode: evacuate                      |
|         | s3_mode: evacuate                       |
|         | storage_mode: suspend                   |
| precheck |                                           |
| resources | compute:                                 |
|         |   failed: []                             |
|         |   handled: []                           |
|         |   initial:                               |
|         |     - id: fa69e3f1-461c-4f4a-b442-3ffb356130c1 |
|         |       status: ACTIVE                     |
|         |     - id: f336f631-cc82-43e6-a582-8dcfcd24a722 |
|         |       status: ACTIVE                     |
|         |     - id: 48864c3f-d9bf-4c32-abdc-901395c3b5f9 |
|         |       status: ACTIVE                     |
|         |     - id: 5fd82e2a-3fef-4171-bfa4-67daa99ae64f |
|         |       status: ACTIVE                     |
|         |   untouched: []                         |
|         | iscsi:                                   |
|         |   failed: []                             |
|         |   handled: []                           |
|         |   initial: []                           |
|         | nfs: null                                |
| state   | suspended_complete                       |
| task    | flow: complete                           |
|         | id: fe8e5ff5-48c1-4a23-a533-28233aaaae4db |
|         | state: success                           |
|         | updated_at: '2019-11-20T13:52:15.849492' |
+-----+-----+
```

Эта команда показывает подробные данные об обслуживании сервера с идентификатором 9dcc9632-911c-4cc5-9a89-5a6fa5db2314.

2.3.7 vinfra node maintenance stop

Возврат сервера к работе:

```
usage: vinfra node maintenance stop <node>
```

<сервер>

Идентификатор или имя хоста сервера

Пример:

```
# vinfra node maintenance stop 9dcc9632-911c-4cc5-9a89-5a6fa5db2314
+-----+-----+
| Field  | Value                                |
+-----+-----+
| task_id | 34e0b546-aa2c-466c-93fe-7dff28c543c6 |
+-----+-----+
```

Эта команда создает задачу для останова обслуживания для сервера с идентификатором 9dcc9632-911c-4cc5-9a89-5a6fa5db2314.

Результат выполнения задачи:

```
# vinfra task show 34e0b546-aa2c-466c-93fe-7dff28c543c6
+-----+-----+
| Field  | Value                                |
+-----+-----+
| details |                                       |
| name    | backend.business.models.maintenance.tasks.MaintenanceStopTask |
| result  |                                       |
| state   | success                              |
| task_id | 34e0b546-aa2c-466c-93fe-7dff28c543c6 |
+-----+-----+
```

2.3.8 vinfra node release

Освобождение сервера из кластера хранилища. Запускает миграцию данных с сервера, а также репликацию и перебалансировку кластера, чтобы обеспечить соответствие настроенному уровню избыточности.

```
usage: vinfra node release [--force] <node>
```

--force

Освобождение сервера без миграции данных

<сервер>

Идентификатор или имя хоста сервера

Пример:

```
# vinfra node release f59dabdb-bd1c-4944-8af2-26b8fe9ff8d4
+-----+-----+
| Field   | Value                                     |
+-----+-----+
| task_id | c2a653a2-8991-4b3a-8bdf-5c0872aa75b3 |
+-----+-----+
```

Эта команда создает задачу для освобождения сервера с идентификатором f59dabdb-bd1c-4944-8af2-26b8fe9ff8d4 из кластера хранилища с миграцией данных, чтобы обеспечить соответствие заданному уровню избыточности.

Результат выполнения задачи:

```
# vinfra task show c2a653a2-8991-4b3a-8bdf-5c0872aa75b3
+-----+-----+
| Field   | Value                                     |
+-----+-----+
| args    | - f59dabdb-bd1c-4944-8af2-26b8fe9ff8d4 |
|         | - false                                   |
| kwargs  | {}                                         |
| name    | backend.tasks.node.ReleaseNodeTask      |
| state   | success                                   |
| task_id | c2a653a2-8991-4b3a-8bdf-5c0872aa75b3 |
+-----+-----+
```

2.3.9 vinfra node forget

Исключение сервера из кластера хранилища:

```
usage: vinfra node forget <node>
```

<сервер>

Идентификатор или имя хоста сервера

Пример:

```
# vinfra node forget fd1e46de-6e17-4571-bf6b-1ac34ec1c225
+-----+-----+
| Field   | Value                                     |
+-----+-----+
| task_id | 0eac3b74-e8f5-4974-9efe-a9070187d83c |
+-----+-----+
```

Эта команда создает задачу для отмены регистрации сервера с идентификатором fd1e46de-6e17-4571-bf6b-1ac34ec1c225 в Acronis Инфраструктура.

Результат выполнения задачи:

```
# vinfra task show 0eac3b74-e8f5-4974-9efe-a9070187d83c
+-----+-----+-----+-----+
| Field | Value |
+-----+-----+-----+-----+
| args  | - fd1e46de-6e17-4571-bf6b-1ac34ec1c225 |
| kwargs | {} |
| name   | backend.tasks.node.DeleteNodeTask |
| state  | success |
| task_id | 0eac3b74-e8f5-4974-9efe-a9070187d83c |
+-----+-----+-----+-----+
```

2.4 Управление сетевыми интерфейсами сервера

2.4.1 vinfra node iface list

Вывод списка сетевых интерфейсов сервера:

```
usage: vinfra node iface list [-a | --node <node>]
```

-a, --all

Вывод списка всех сетевых интерфейсов на всех серверах

--node <сервер>

Идентификатор или имя хоста сервера, список сетевых интерфейсов на котором необходимо отобразить (по умолчанию: node001.vstoragedomain)

Пример:

Эта команда отображает сетевые интерфейсы сервера с идентификатором 4f96acf5-3bc8-4094-bcb6-4d1953be7b55.

```
# vinfra node iface list --node 4f96acf5-3bc8-4094-bcb6-4d1953be7b55
+-----+-----+-----+-----+-----+
| name | node_id | ipv4 | state | network |
+-----+-----+-----+-----+-----+
| eth0 | 4f96acf5-3bc8-4094-bcb6-4d1953be7b55 | - 10.94.29.218/16 | up | Public |
```

```
| eth1 | 4f96acf5-3bc8-4094-bcb6-4d1953be7b55 | - 10.37.130.101/24 | up | Private |
+-----+-----+-----+-----+-----+-----+-----+-----+-----+
```

2.4.2 vinfra node iface show

Отображение подробных данных о сетевом интерфейсе:

```
usage: vinfra node iface show [--node <node>] <iface>
```

--node <сервер>

Идентификатор или имя хоста сервера (по умолчанию: node001.vstoragedomain)

<интерфейс>

Имя сетевого интерфейса

Пример:

```
# vinfra node iface show eth0 --node 4f96acf5-3bc8-4094-bcb6-4d1953be7b55
+-----+-----+-----+-----+-----+-----+-----+-----+-----+
| Field | Value |
+-----+-----+-----+-----+-----+-----+-----+-----+-----+
| contained_in | |
| dhcp4 | 10.94.29.218 |
| dhcp4_enabled | True |
| dhcp6 | fe80::21c:42ff:fe2a:4fdf |
| dhcp6_enabled | True |
| dns4 | - 127.0.0.1 |
| dns6 | [] |
| duplex | |
| gw4 | 10.94.0.1 |
| gw6 | |
| ignore_auto_dns_v4 | False |
| ignore_auto_dns_v6 | False |
| ignore_auto_routes_v4 | False |
| ignore_auto_routes_v6 | False |
| ipv4 | - 10.94.29.218/16 |
| ipv6 | - fe80::21c:42ff:fe2a:4fdf/64 |
| mac_addr | 00:1c:42:2a:4f:df |
| mtu | 1500 |
| multicast | True |
| name | eth0 |
| node_id | 4f96acf5-3bc8-4094-bcb6-4d1953be7b55 |
| plugged | True |
| roles_set | 237e58dd-6c10-49c1-be7f-7ddf7de2efd1 |
| rx_bytes | 1844502614 |
| rx_dropped | 0 |
| rx_errors | 0 |
| rx_overruns | 0 |
```

```

| rx_packets      | 11543284      |
| speeds          | current: null |
|                 | max: null     |
| state          | up            |
| tx_bytes       | 28477979     |
| tx_dropped     | 0             |
| tx_errors      | 0             |
| tx_overruns    | 0             |
| tx_packets     | 107649       |
| type           | iface        |
+-----+-----+

```

Эта команда отображает подробные данные о сетевом интерфейсе eth0, расположенном на сервере с идентификатором 4f96acf5-3bc8-4094-bcb6-4d1953be7b55.

2.4.3 vinfra node iface up

Включение сетевого интерфейса:

```
usage: vinfra node iface up [--node <node>] <iface>
```

--node <сервер>

Идентификатор или имя хоста сервера (по умолчанию: node001.vstoragedomain)

<интерфейс>

Имя сетевого интерфейса

Пример:

```

# vinfra node iface up eth2 --node 4f96acf5-3bc8-4094-bcb6-4d1953be7b55
+-----+-----+
| Field          | Value        |
+-----+-----+
| contained_in   |              |
| dhcp4          | 10.37.130.138 |
| dhcp4_enabled  | True         |
| dhcp6          | fe80::21c:42ff:fe8:5b90 |
| dhcp6_enabled  | True         |
| dns4           | - 127.0.0.1  |
| dns6           | []           |
| duplex         |              |
| gw4            | 10.94.0.1    |
| gw6            |              |
| ignore_auto_dns_v4 | False       |
| ignore_auto_dns_v6 | False       |
| ignore_auto_routes_v4 | False     |
| ignore_auto_routes_v6 | False     |

```

```

| ipv4           | - 10.37.130.138/24 |
| ipv6           | - fe80::21c:42ff:fe8:5b90/64 |
| mac_addr      | 00:1c:42:f8:5b:90 |
| mtu           | 1500 |
| multicast     | True |
| name          | eth2 |
| node_id       | 4f96acf5-3bc8-4094-bcb6-4d1953be7b55 |
| plugged       | True |
| roles_set     | |
| rx_bytes      | 97632 |
| rx_dropped    | 0 |
| rx_errors     | 0 |
| rx_overruns   | 0 |
| rx_packets    | 1258 |
| speeds        | current: null |
|               | max: null |
| state         | up |
| tx_bytes      | 1116 |
| tx_dropped    | 0 |
| tx_errors     | 0 |
| tx_overruns   | 0 |
| tx_packets    | 8 |
| type          | iface |
+-----+-----+

```

Эта команда включает (активирует) сетевой интерфейс eth2, расположенный на сервере с идентификатором 4f96acf5-3bc8-4094-bcb6-4d1953be7b55.

2.4.4 vinfra node iface down

Отключение сетевого интерфейса:

```
usage: vinfra node iface down [--node <node>] <iface>
```

--node <сервер>

Идентификатор или имя хоста сервера

<интерфейс>

Имя сетевого интерфейса

Пример:

```

# vinfra node iface down eth2 --node 4f96acf5-3bc8-4094-bcb6-4d1953be7b55
+-----+-----+
| Field          | Value |
+-----+-----+
| contained_in   |       |

```

```

| dhcp4                |                |
| dhcp4_enabled        | True           |
| dhcp6                |                |
| dhcp6_enabled        | True           |
| dns4                  | - 127.0.0.1   |
| dns6                  | []             |
| duplex                |                |
| gw4                   | 10.94.0.1     |
| gw6                   |                |
| ignore_auto_dns_v4   | False         |
| ignore_auto_dns_v6   | False         |
| ignore_auto_routes_v4 | False        |
| ignore_auto_routes_v6 | False        |
| ipv4                  | []            |
| ipv6                  | []            |
| mac_addr              | 00:1c:42:f8:5b:90
| mtu                   | 1500          |
| multicast             | True          |
| name                  | eth2          |
| node_id               | 4f96acf5-3bc8-4094-bcb6-4d1953be7b55
| plugged               | False         |
| roles_set             |                |
| rx_bytes              | 97984         |
| rx_dropped            | 0             |
| rx_errors             | 0             |
| rx_overruns           | 0             |
| rx_packets            | 1264          |
| speeds                | current: null
|                       | max: null     |
| state                 | down          |
| tx_bytes              | 1116          |
| tx_dropped            | 0             |
| tx_errors             | 0             |
| tx_overruns           | 0             |
| tx_packets            | 8             |
| type                  | iface         |
+-----+-----+

```

Эта команда отключает (деактивирует) сетевой интерфейс eth2, расположенный на сервере с идентификатором 4f96acf5-3bc8-4094-bcb6-4d1953be7b55.

2.4.5 vinfra node iface set

Изменение параметров сетевого интерфейса (при этом пропущенные параметры перезаписываются значениями по умолчанию для данного интерфейса):

```

usage: vinfra node iface set [--ipv4 <ipv4>] [--ipv6 <ipv6>] [--gw4 <gw4>] [--gw6 <gw6>]
                             [--mtu <mtu>] [--dhcp4 | --no-dhcp4] [--dhcp6 | --no-dhcp6]

```

```
[--auto-routes-v4 | --ignore-auto-routes-v4]  
[--auto-routes-v6 | --ignore-auto-routes-v6]  
[--network <network> | --no-network]  
[--connected-mode | --datagram-mode] [--node <node>] <iface>
```

--ipv4 <ipv4>

Разделенный запятыми список адресов IPv4

--ipv6 <ipv6>

Разделенный запятыми список адресов IPv6

--gw4 <gw4>

Адрес шлюза IPv4

--gw6 <gw6>

Адрес шлюза IPv6

--mtu <mtu>

Значение MTU (максимального размера передаваемого пакета) для интерфейса

--dhcp4

Включение DHCPv4

--no-dhcp4

Отключение DHCPv4

--dhcp6

Включение DHCPv6

--no-dhcp6

Отключение DHCPv6

--auto-routes-v4

Включить автоматические маршруты IPv4

--ignore-auto-routes-v4

Игнорировать автоматические маршруты IPv4

--auto-routes-v6

Включить автоматические маршруты IPv6

--ignore-auto-routes-v6

Игнорировать автоматические маршруты IPv6

`--network <сеть>`
Идентификатор или имя сети

`--no-network`
Удаление сети из интерфейса

`--connected-mode`
Включение подключенного режима (только для интерфейсов InfiniBand)

`--datagram-mode`
Включение режима дейтаграмм (только для интерфейсов InfiniBand)

`--node <сервер>`
Идентификатор или имя хоста сервера (по умолчанию: `node001.vstoragedomain`)

`<интерфейс>`
Имя сетевого интерфейса

Пример:

```
# vinfra node iface set eth2 --network Private \
--node 4f96acf5-3bc8-4094-bcb6-4d1953be7b55
+-----+-----+
| Field  | Value |
+-----+-----+
| task_id | 8a378098-6760-4fe9-ac20-1f18a8ed9d2e |
+-----+-----+
```

Эта команда создает задачу для назначения сетевому интерфейсу `eth2`, расположенному на сервере с идентификатором `4f96acf5-3bc8-4094-bcb6-4d1953be7b55`, в сети `Private` (Частная).

Результат выполнения задачи:

```
# vinfra task show 8a378098-6760-4fe9-ac20-1f18a8ed9d2e
+-----+-----+
| Field  | Value |
+-----+-----+
| args   | - 4f96acf5-3bc8-4094-bcb6-4d1953be7b55 |
|        | - eth2 |
| kwargs | roles_set: 6095a997-e5f1-493d-a750-41ddf277153b |
| name   | backend.presentation.network.tasks.NetworkInterfaceChangeTask |
| result | contained_in: null |
|        | dhcp4: null |
|        | dhcp4_enabled: false |
|        | dhcp6: null |
|        | dhcp6_enabled: false |
|        | duplex: null |
|        | gw4: null |
+-----+-----+
```



```

|         | gw6: null
|         | ignore_auto_routes_v4: true
|         | ignore_auto_routes_v6: true
|         | ipv4:
|         | - 10.37.130.103/24
|         | ipv6:
|         | - fe80::21c:42ff:fe75:7c4d/64
|         | mac_addr: 00:1c:42:75:7c:4d
|         | mtu: 1500
|         | multicast: true
|         | name: eth2
|         | node_id: 4f96acf5-3bc8-4094-bcb6-4d1953be7b55
|         | plugged: true
|         | roles_set: 6095a997-e5f1-493d-a750-41ddf277153b
|         | rx_bytes: 38156
|         | rx_dropped: 0
|         | rx_errors: 0
|         | rx_overruns: 0
|         | rx_packets: 225
|         | speeds:
|         |   current: null
|         |   max: null
|         | state: up
|         | tx_bytes: 13087
|         | tx_dropped: 0
|         | tx_errors: 0
|         | tx_overruns: 0
|         | tx_packets: 145
|         | type: iface
| state   | success
| task_id | 8a378098-6760-4fe9-ac20-1f18a8ed9d2e
+-----+

```

2.4.6 vinfra node iface create-bond

Создание объединения сетевых интерфейсов:

```

usage: vinfra node iface create-bond [--ipv4 <ipv4>] [--ipv6 <ipv6>] [--gw4 <gw4>]
      [--gw6 <gw6>] [--mtu <mtu>] [--dhcp4 | --no-dhcp4]
      [--dhcp6 | --no-dhcp6] [--network <network>]
      [--auto-routes-v4 | --ignore-auto-routes-v4]
      [--auto-routes-v6 | --ignore-auto-routes-v6]
      [--bonding-opts <bonding_opts>] [--node <node>]
      --bond-type <bond-type> --ifaces <ifaces>

```

--ipv4 <ipv4>

Разделенный запятыми список адресов IPv4

--ipv6 <ipv6>
Разделенный запятыми список адресов IPv6

--gw4 <gw4>
Адрес шлюза IPv4

--gw6 <gw6>
Адрес шлюза IPv6

--mtu <mtu>
Значение MTU (максимального размера передаваемого пакета) для интерфейса

--dhcp4
Включение DHCPv4

--no-dhcp4
Отключение DHCPv4

--dhcp6
Включение DHCPv6

--no-dhcp6
Отключение DHCPv6

--auto-routes-v4
Включить автоматические маршруты IPv4

--ignore-auto-routes-v4
Игнорировать автоматические маршруты IPv4

--auto-routes-v6
Включить автоматические маршруты IPv6

--ignore-auto-routes-v6
Игнорировать автоматические маршруты IPv6

--network <сеть>
Идентификатор или имя сети

--bonding-opts <параметры_объединения>
Дополнительные параметры объединения

--bond-type <тип объединения>

Тип объединения (balance-rr, active-backup, balance-xor, broadcast, 802.3ad, balance-tlb, balance-alb)

--node <сервер>

Идентификатор или имя хоста сервера (по умолчанию: node001.vstoragedomain)

--ifaces <интерфейсы>

Разделенный запятыми список имен сетевых интерфейсов, например:

интерфейс1, интерфейс2...интерфейс<N>

Пример:

```
# vinfra node iface create-bond --ifaces eth2,eth3 --bond-type balance-xor \
--dhcp4 --node fd1e46de-6e17-4571-bf6b-1ac34ec1c225
+-----+-----+
| Field  | Value                                |
+-----+-----+
| task_id | becf96ad-9e39-4bec-b82c-4e1219a196de |
+-----+-----+
```

Эта команда создает задачу для объединения сетевых интерфейсов eth2 и eth3 в интерфейс bond0 типа balance-xor на сервере с идентификатором fd1e46de-6e17-4571-bf6b-1ac34ec1c225.

Результат выполнения задачи:

```
# vinfra task show becf96ad-9e39-4bec-b82c-4e1219a196de
+-----+-----+
| Field  | Value                                |
+-----+-----+
| args   | - fd1e46de-6e17-4571-bf6b-1ac34ec1c225 |
| kwargs | bond_type: balance-xor                |
|        | ifaces:                                |
|        | - eth2                                 |
|        | - eth3                                 |
|        | registration_token: 3102ed1a          |
| name   | backend.presentation.network.tasks.NetworkInterfaceCreateBondingTask |
| result | bond_type: balance-xor                |
|        | dhcp4: 10.37.130.117                  |
|        | dhcp4_enabled: true                   |
|        | dhcp6: fe80::21c:42ff:fe81:27d0       |
|        | dhcp6_enabled: true                   |
|        | duplex: null                           |
|        | gw4: 10.94.0.1                         |
|        | gw6: null                              |
|        | ignore_auto_routes_v4: false          |
|        | ignore_auto_routes_v6: false          |
|        | ipv4:                                  |
|        | - 10.37.130.117/24                    |
+-----+-----+
```

```

|         | ipv6:
|         | - fe80::21c:42ff:fe81:27d0/64
|         | mac_addr: 00:1c:42:81:27:d0
|         | mtu: 1500
|         | multicast: true
|         | name: bond0
|         | node_id: fd1e46de-6e17-4571-bf6b-1ac34ec1c225
|         | plugged: true
|         | roles_set: ''
|         | rx_bytes: 3048
|         | rx_dropped: 0
|         | rx_errors: 0
|         | rx_overruns: 0
|         | rx_packets: 22
|         | speeds:
|         |   current: null
|         |   max: null
|         | state: up
|         | tx_bytes: 1782
|         | tx_dropped: 0
|         | tx_errors: 0
|         | tx_overruns: 0
|         | tx_packets: 13
|         | type: bonding
| state   | success
| task_id | becf96ad-9e39-4bec-b82c-4e1219a196de
+-----+

```

2.4.7 vinfra node iface create-vlan

Создание сети VLAN:

```

usage: vinfra node iface create-vlan [--ipv4 <ipv4>] [--ipv6 <ipv6>] [--gw4 <gw4>]
                                     [--gw6 <gw6>] [--mtu <mtu>] [--dhcp4 | --no-dhcp4]
                                     [--dhcp6 | --no-dhcp6] [--network <network>]
                                     [--auto-routes-v4 | --ignore-auto-routes-v4]
                                     [--auto-routes-v6 | --ignore-auto-routes-v6]
                                     [--node <node>] --iface <iface> --tag <tag>

```

--ipv4 <ipv4>

Разделенный запятыми список адресов IPv4

--ipv6 <ipv6>

Разделенный запятыми список адресов IPv6

--gw4 <gw4>

Адрес шлюза IPv4

--gw6 <gw6>
Адрес шлюза IPv6

--mtu <mtu>
Значение MTU (максимального размера передаваемого пакета) для интерфейса

--dhcp4
Включение DHCPv4

--no-dhcp4
Отключение DHCPv4

--dhcp6
Включение DHCPv6

--no-dhcp6
Отключение DHCPv6

--auto-routes-v4
Включить автоматические маршруты IPv4

--ignore-auto-routes-v4
Игнорировать автоматические маршруты IPv4

--auto-routes-v6
Включить автоматические маршруты IPv6

--ignore-auto-routes-v6
Игнорировать автоматические маршруты IPv6

--network <сеть>
Идентификатор или имя сети

--node <сервер>
Идентификатор или имя хоста сервера (по умолчанию: node001.vstoragedomain)

--iface <интерфейсы>
Имя интерфейса

--tag <тег>
Номер тега VLAN

Пример:

```
# vinfra node iface create-vlan --iface eth2 --tag 100 --dhcp4 \
--node fd1e46de-6e17-4571-bf6b-1ac34ec1c225
+-----+-----+
| Field   | Value                                     |
+-----+-----+
| task_id | 0b978acd-367b-47ad-8572-4f4e6ffb8877 |
+-----+-----+
```

Эта команда создает задачу для создания сети VLAN с тегом 100 на сетевом интерфейсе eth2 сервера с идентификатором fd1e46de-6e17-4571-bf6b-1ac34ec1c225.

Результат выполнения задачи:

```
# vinfra task show 0b978acd-367b-47ad-8572-4f4e6ffb8877
+-----+-----+
| Field   | Value                                     |
+-----+-----+
| args    | - fd1e46de-6e17-4571-bf6b-1ac34ec1c225 |
| kwargs  | iface: eth2                               |
|         | tag: 100                                   |
| name    | backend.presentation.network.tasks.NetworkInterfaceCreateVlanTask |
| result  | built_on: eth2                             |
|         | dhcp4: null                               |
|         | dhcp4_enabled: false                      |
|         | dhcp6: null                               |
|         | dhcp6_enabled: false                      |
|         | duplex: null                              |
|         | gw4: null                                 |
|         | gw6: null                                 |
|         | ignore_auto_routes_v4: true               |
|         | ignore_auto_routes_v6: true               |
|         | ipv4: []                                  |
|         | ipv6:                                     |
|         | - fe80::21c:42ff:fe81:27d0/64             |
|         | mac_addr: 00:1c:42:81:27:d0               |
|         | mtu: 1500                                 |
|         | multicast: true                           |
|         | name: eth2.100                             |
|         | node_id: fd1e46de-6e17-4571-bf6b-1ac34ec1c225 |
|         | plugged: true                              |
|         | roles_set: ''                              |
|         | rx_bytes: 0                                |
|         | rx_dropped: 0                              |
|         | rx_errors: 0                               |
|         | rx_overruns: 0                             |
|         | rx_packets: 0                              |
|         | speeds:                                     |
|         |   current: null                            |
|         |   max: null                                |
|         | state: up                                  |
|         | tag: 100                                   |
+-----+-----+
```

```

|         | tx_bytes: 738 |
|         | tx_dropped: 0 |
|         | tx_errors: 0 |
|         | tx_overruns: 0 |
|         | tx_packets: 7 |
|         | type: vlan |
| state   | success |
| task_id | 0b978acd-367b-47ad-8572-4f4e6ffb8877 |
+-----+

```

2.4.8 vinfra node iface delete

Удаление сетевого интерфейса:

```
usage: vinfra node iface delete [--node <node>] <iface>
```

--node <сервер>

Идентификатор или имя хоста сервера (по умолчанию: node001.vstoragedomain)

<интерфейс>

Имя сетевого интерфейса

Пример:

```

# vinfra node iface delete --node fd1e46de-6e17-4571-bf6b-1ac34ec1c225 eth2.100
+-----+
| Field  | Value |
+-----+
| task_id | 16503616-6c1c-48f9-999a-9d87b617d9ee |
+-----+

```

Эта команда создает задачу для удаления интерфейса VLAN eth1.100 с сервера с идентификатором fd1e46de-6e17-4571-bf6b-1ac34ec1c225.

Результат выполнения задачи:

```

# vinfra task show 16503616-6c1c-48f9-999a-9d87b617d9ee
+-----+
| Field  | Value |
+-----+
| details | |
| name    | backend.presentation.network.tasks.NetworkInterfaceRemoveTask |
| result  | |
| state   | success |
| task_id | 16503616-6c1c-48f9-999a-9d87b617d9ee |
+-----+

```

2.5 Управление дисками серверов

2.5.1 vinfra node disk list

Вывод списка дисков серверов:

```
usage: vinfra node disk list [-a | --node <node>]
```

`-a, --all`

Вывод списка дисков на всех серверах

`--node <сервер>`

Идентификатор или имя хоста сервера, список дисков на котором необходимо отобразить (по умолчанию: `node001.vstoragedomain`)

Пример:

```
# vinfra node disk list --node 94d58604-6f30-4339-8578-adb7903b7277 \
-c id -c node_id -c device -c used -c size -c role
+-----+-----+-----+-----+-----+-----+
| id          | node_id      | device | used   | size    | role      |
+-----+-----+-----+-----+-----+-----+
| E0B7CE6F-<...> | 94d58604-<...> | sda    | 5.5GiB | 239.1GiB | mds-system |
| EAC7DF5D-<...> | 94d58604-<...> | sdb    | 2.1GiB | 1007.8GiB | cs         |
| 49D792CA-<...> | 94d58604-<...> | sdc    | 2.1GiB | 1007.8GiB | cs         |
+-----+-----+-----+-----+-----+-----+

```

Эта команда выводит список дисков на сервере с идентификатором `94d58604-6f30-4339-8578-adb7903b7277`. (Выходные данные усекаются, чтобы они поместились на странице)

2.5.2 vinfra node disk show

Отображение подробных данных о диске:

```
usage: vinfra node disk show [--node <node>] <disk>
```

`--node <сервер>`

Идентификатор или имя хоста сервера

`<диск>`

Идентификатор или имя устройства диска (по умолчанию: `node001.vstoragedomain`)

Пример:

```
# vinfra node disk show EAC7DF5D-9E60-4444-85F7-5CA5738399CC \
--node 94d58604-6f30-4339-8578-adb7903b7277
+-----+
| Field          | Value                                     |
+-----+-----+
| being_released | False                                   |
| device         | sdb                                     |
| disk_status    | ok                                     |
| encryption     |                                         |
| id             | EAC7DF5D-9E60-4444-85F7-5CA5738399CC |
| is_blink_available | False                                 |
| is_blinking    | False                                 |
| latency        |                                         |
| lun_id         |                                         |
| model          | Vz_HARDDISK2                           |
| mountpoint     | /vstorage/33aac2d5                     |
| node_id        | 94d58604-6f30-4339-8578-adb7903b7277 |
| role           | cs                                     |
| rpm            |                                         |
| serial_number  | 45589b5823ce4c188b55                   |
| service_id     | 1026                                   |
| service_params | journal_type: inner_cache              |
|                | tier: 0                                 |
| service_status | ok                                     |
| slot           |                                         |
| smart_status   | not_supported                           |
| space          | full_size: 109951162776                 |
|                | size: 1082101518336                    |
|                | used: 2246164480                        |
| tasks          |                                         |
| temperature    | 0.0                                     |
| transport      |                                         |
| type           | hdd                                     |
+-----+-----+
```

Эта команда отображает подробные данные о диске с идентификатором EAC7DF5D-9E60-4444-85F7-5CA5738399CC, присоединенном к серверу с идентификатором 94d58604-6f30-4339-8578-adb7903b7277.

2.5.3 vinfra node disk assign

Добавление многих дисков в состав кластера хранения:

```
usage: vinfra node disk assign --disk <disk>:<role>[:<key=value,...>]
      [--node <node>]
```

```
--disk <диск>:<роль>[:<ключ=значение, ...>]
```

Конфигурация диска в формате:

- <диск>: идентификатор или имя дискового устройства
- <роль>: роль диска (cs, mds, journal, mds-journal, mds-system, cs-system, system)
- разделенные запятыми пары ключ=значение со следующими ключами (каждый из которых необязателен):
 - tier: уровень диска (0, 1, 2 или 3)
 - journal-tier: уровень диска журнала (кэша) (0, 1, 2 или 3)
 - journal-type: тип диска журнала (кэша) (no_cache — без кэша, inner_cache — внутренний кэш или external_cache — внешний кэш)
 - journal-disk: идентификатор или имя устройства диска журнала (кэша)
 - journal-size: размер диска журнала (кэша), в байтах
 - bind-address: IP-адрес привязки для сервиса метаданных

Например: sda:cs:tier=0, journal-type=inner_cache. Этот параметр можно указывать несколько раз.

```
--node <сервер>
```

Идентификатор или имя хоста сервера (по умолчанию: node001.vstoragedomain)

Пример:

```
# vinfra node disk assign --disk sdc:cs --node f59dabdb-bd1c-4944-8af2-26b8fe9ff8d4
+-----+-----+
| Field  | Value                                |
+-----+-----+
| task_id | 080337ba-0508-44a0-9363-eddc9df9f0d |
+-----+-----+
```

Эта команда создает задачу для назначения роли cs диску sdc на сервере с идентификатором f59dabdb-bd1c-4944-8af2-26b8fe9ff8d4

Результат выполнения задачи:

```
# vinfra task show 080337ba-0508-44a0-9363-eddc9df9f0d
+-----+-----+-----+
| Field  | Value                                |
+-----+-----+-----+
| args   | []                                    |
+-----+-----+-----+
```

```

| kwargs | cluster_id: 1
|         | disks:
|         | - id: D3BEF4BB-AA3B-4DB6-9376-BC7CDA636700
|         |   role: cs
|         |   service_params: {}
|         | logger:
|         |   __classname: backend.logger.tracer.TracingLogger
|         |   __dict:
|         |     prefix: POST /api/v2/1/nodes/f59dabdb-bd1c-4944-8af2-26b8fe9ff8d4/disks/
|         |     token: '3215629651314950'
|         |   node_id: f59dabdb-bd1c-4944-8af2-26b8fe9ff8d4
| name    | backend.tasks.disks.BulkAssignDiskTask
| result  | {}
| state   | success
| task_id | 080337ba-0508-44a0-9363-eddcd9df9f0d
+-----+

```

2.5.4 vinfra node disk release

Освобождение диска из кластера хранилища. Запускает миграцию данных с сервера, а также репликацию и перебалансировку кластера, чтобы обеспечить соответствие настроенному уровню избыточности:

```
usage: vinfra node disk release [--force] [--node <node>] <disk>
```

`--force`

Освобождение без миграции данных

`--node <сервер>`

Идентификатор или имя хоста сервера (по умолчанию: `node001.vstoragedomain`)

`<диск>`

Идентификатор или имя устройства диска

Пример:

```

# vinfra node disk release sdc --node f59dabdb-bd1c-4944-8af2-26b8fe9ff8d4
+-----+
| Field  | Value |
+-----+
| task_id | 587a936d-3953-481c-a2cd-b1223b890bec |
+-----+

```

Эта команда создает задачу для освобождения от роли `cs` диска `sdc` на сервере с идентификатором `f59dabdb-bd1c-4944-8af2-26b8fe9ff8d4`.

Результат выполнения задачи:

```
# vinfra task show 587a936d-3953-481c-a2cd-b1223b890bec
+-----+
| Field | Value
+-----+
| args  | []
| kwargs | cluster_id: 1
|        | disk_id: 43EF3400-EA95-43DE-B624-3D7ED0F9DDDD
|        | force: false
|        | logger:
|        |   __classname: backend.logger.tracer.TracingLogger
|        |   __dict:
|        |     prefix: POST /api/v2/1/nodes/f59dabdb-
|        |     bd1c-4944-8af2-26b8fe9ff8d4/disks/43EF3400-EA95-43DE-B624-3D7ED0F9DDDD/release/
|        |     token: '3217122839314940'
|        |     node_id: f59dabdb-bd1c-4944-8af2-26b8fe9ff8d4
| name   | backend.tasks.disks.ReleaseDiskTask
| state  | success
| task_id | 587a936d-3953-481c-a2cd-b1223b890bec
+-----+
```

2.5.5 vinfra node disk blink on

Включить мигание в указанном отсеке накопителей, чтобы можно было проще найти диск для обслуживания:

```
usage: vinfra node disk blink on [--node <node>] <disk>
```

--node <сервер>

Идентификатор или имя хоста сервера (по умолчанию: node001.vstoragedomain)

<диск>

Идентификатор или имя устройства диска

Пример:

```
# vinfra node disk blink on sda --node f59dabdb-bd1c-4944-8af2-26b8fe9ff8d4
```

Эта команда включает мигание индикатора на диске sda сервера с идентификатором f59dabdb-bd1c-4944-8af2-26b8fe9ff8d4.

2.5.6 vinfra node disk blink off

Выключить мигание в указанном отсеке накопителей:

```
usage: vinfra node disk blink off [--node <node>] <disk>
```

--node <сервер>

Идентификатор или имя хоста сервера (по умолчанию: node001.vstoragedomain)

<диск>

Идентификатор или имя устройства диска

Пример:

```
# vinfra node disk blink off sda --node f59dabdb-bd1c-4944-8af2-26b8fe9ff8d4
```

Эта команда выключает мигание индикатора на диске sda сервера с идентификатором f59dabdb-bd1c-4944-8af2-26b8fe9ff8d4.

2.5.7 vinfra node iscsi target add

Добавление целевого устройства iSCSI в качестве диска на сервер:

```
usage: vinfra node iscsi target add [--auth-username <auth-username>]
                                     [--auth-password <auth-password>]
                                     --portal <portal> --node <node> <target-name>
```

--auth-username <имя_пользователя_аутентификации>

Имя пользователя

--auth-password <пароль_аутентификации>

Пароль пользователя

--portal <портал>

IP-адрес портала в формате IP-адрес:порт (этот параметр можно указывать несколько раз)

--node <сервер>

Идентификатор или имя хоста сервера

<имя_целевого_устройства>

Имя целевого устройства

Пример:

```
# vinfra node iscsi target add iqn.2014-06.com.vstorage:target1 \
--portal 172.16.24.244:3260 --node f1931be7-0a01-4977-bfef-51a392adcd94
+-----+-----+
| Field   | Value                                     |
+-----+-----+
| task_id | c42bfbe5-7292-41c2-91cb-446795535ab9 |
+-----+-----+
```

Эта команда создает задачу для подсоединения целевого устройства iSCSI `iqn.2014-06.com.vstorage:target1` с IP-адресом `172.16.24.244` и портом `3260` к серверу с идентификатором `f1931be7-0a01-4977-bfef-51a392adcd94`.

Результат выполнения задачи:

```
# vinfra task show c42bfbe5-7292-41c2-91cb-446795535ab9
+-----+-----+
| Field   | Value                                     |
+-----+-----+
| args    | - f1931be7-0a01-4977-bfef-51a392adcd94 |
| kwargs | portals:                                  |
|         | - address: 172.16.24.244                 |
|         |   port: 3260                             |
|         | target_name: iqn.2014-06.com.vstorage:target1 |
| name    | backend.presentation.nodes.iscsi_initiators.tasks.ConnectTask |
| result  | connected: true                           |
|         | portals:                                  |
|         | - address: 172.16.24.244                 |
|         |   port: 3260                             |
|         | state: connected                          |
|         | target_name: iqn.2014-06.com.vstorage:target1 |
| state   | success                                    |
| task_id | c42bfbe5-7292-41c2-91cb-446795535ab9 |
+-----+-----+
```

2.5.8 vinfra node iscsi target delete

Удаление целевого устройства iSCSI с сервера:

```
usage: vinfra node iscsi target delete --node <node> <target-name>
```

`--node <сервер>`

Идентификатор или имя хоста сервера

`<имя_целевого_устройства>`

Имя целевого устройства

Пример:

```
# vinfra node iscsi target delete iqn.2014-06.com.vstorage:target1 \
--node f1931be7-0a01-4977-bfef-51a392adcd94
+-----+
| Field  | Value |
+-----+
| task_id | c8dc74ee-86d6-4b89-8b6f-153ff1e78cb7 |
+-----+
```

Эта команда создает задачу для отсоединения целевого устройства iSCSI `iqn.2014-06.com.vstorage:target1` от сервера с идентификатором `f1931be7-0a01-4977-bfef-51a392adcd94`.

Результат выполнения задачи:

```
# vinfra task show c8dc74ee-86d6-4b89-8b6f-153ff1e78cb7
+-----+
| Field  | Value |
+-----+
| args   | - f1931be7-0a01-4977-bfef-51a392adcd94 |
| kwargs | target_name: iqn.2014-06.com.vstorage:target1 |
| name   | backend.presentation.nodes.iscsi_initiators.tasks.DisconnectTask |
| state  | success |
| task_id | c8dc74ee-86d6-4b89-8b6f-153ff1e78cb7 |
+-----+
```

2.6 Создание и удаление кластера хранилища данных

2.6.1 vinfra cluster create

Создание кластера хранилища данных:

```
usage: vinfra cluster create [--disk <disk>:<role>[:<key=value,...>]]
                             [--tier-encryption {0,1,2,3}] --node <node> <cluster-name>
```

`--disk <диск>:<роль> [:<ключ=значение, ...>]`

Конфигурация диска в формате:

- `<диск>`: идентификатор или имя дискового устройства
- `<роль>`: роль диска (`cs`, `mds`, `journal`, `mds-journal`, `mds-system`, `cs-system`, `system`)
- разделенные запятыми пары `ключ=значение` со следующими ключами (каждый из которых необязателен):

- tier: уровень диска (0, 1, 2 или 3)
- journal-tier: уровень диска журнала (кэша) (0, 1, 2 или 3)
- journal-type: тип диска журнала (кэша) (no_cache — без кэша, inner_cache — внутренний кэш или external_cache — внешний кэш)
- journal-disk: идентификатор или имя устройства диска журнала (кэша)
- journal-size: размер диска журнала (кэша), в байтах
- bind-address: IP-адрес привязки для сервиса метаданных

Например: sda:cs:tier=0, journal-type=inner_cache. Этот параметр можно указывать несколько раз.

--tier-encryption {0,1,2,3}

Включение шифрования для определенных уровней кластера хранилища. По умолчанию шифрование отключено. Этот параметр можно указывать несколько раз.

--node <сервер>

Идентификатор или имя хоста сервера

<имя_кластера>

Имя кластера хранилища данных

Пример:

```
# vinfra cluster create stor1 --node 94d58604-6f30-4339-8578-adb7903b7277
+-----+-----+
| Field  | Value          |
+-----+-----+
| task_id | d9ca8e1d-8ac8-4459-898b-2d803efd7bc6 |
+-----+-----+
```

Эта команда создает задачу для создания кластера хранилища данных stor1 на сервере с идентификатором 94d58604-6f30-4339-8578-adb7903b7277. Поскольку роли дисков не указаны явно, они назначаются автоматически: mds-system для системного диска и cs для всех остальных дисков.

Результат выполнения задачи:

```
# vinfra task show d9ca8e1d-8ac8-4459-898b-2d803efd7bc6
+-----+-----+
| Field  | Value          |
+-----+-----+
| args   | - stor1        |
|        | - 94d58604-6f30-4339-8578-adb7903b7277 |
+-----+-----+
```



```

|         | - null          |
|         | - null          |
| kwargs  | {}              |
| name    | backend.tasks.cluster.CreateNewCluster |
| result  | cluster_id: 1  |
| state   | success         |
| task_id | d9ca8e1d-8ac8-4459-898b-2d803efd7bc6 |
+-----+-----+

```

2.6.2 vinfra cluster delete

Удаление кластера хранилища данных.

```
usage: vinfra cluster delete
```

Пример:

```

# vinfra cluster delete
Operation waiting (timeout=600s) [Elapsed Time: 0:01:09] ... |
Operation successful

```

Эта команда освобождает все серверы из кластера хранилища данных.

2.7 Вывод обзора и подробных данных о кластере хранилища

2.7.1 vinfra cluster overview

Вывод обзора по кластеру хранилища:

```
usage: vinfra cluster overview
```

Пример:

```

# vinfra cluster overview
+-----+-----+
| Field          | Value          |
+-----+-----+
| chunks         | blocked: 0     |
|                 | degraded: 0    |
|                 | deleting: 0    |
|                 | healthy: 2     |

```

```

|                                     | offline: 0 |
|                                     | overcommitted: 0 |
|                                     | pending: 0 |
|                                     | replicating: 0 |
|                                     | standby: 0 |
|                                     | total: 2 |
|                                     | unique: 2 |
|                                     | urgent: 0 |
|                                     | void: 0 |
| fs_stat                             | chunk_maps: 2 |
|                                     | chunk_nodes: 2 |
|                                     | file_maps: 2 |
|                                     | files: 9 |
|                                     | inodes: 9 |
|                                     | used_size: 11335680 |
| id                                   | 1 |
| license                              | capacity: 1099511627776 |
|                                     | expiration_ts: null |
|                                     | keynumber: null |
|                                     | status: 0 |
|                                     | used_size: 11335680 |
| logic_space                          | free: 1099500292096 |
|                                     | total: 1099511627776 |
|                                     | used: 11335680 |
| name                                  | stor1 |
| repl                                  | eta: null |
|                                     | reads: 0 |
|                                     | writes: 0 |
| resistance                            | to_lose: 0 |
|                                     | total: 1 |
| space_per_service                    | abgw: null |
|                                     | compute: null |
|                                     | iscsi: null |
|                                     | nfs: null |
|                                     | s3: null |
| status                                | healthy |
| tiers                                 | - id: 0 |
|                                     | phys_space: |
|                                     |   free: 2164191700992 |
|                                     |   total: 2164203036672 |
|                                     |   used: 11335680 |
+-----+-----+

```

Эта команда выводит обзорные сведения о кластере.

2.7.2 vinfra cluster show

Вывод подробных данных о кластере:

```
usage: vinfra cluster show
```

Пример:

```
# vinfra cluster show
+-----+-----+
| Field | Value |
+-----+-----+
| id    | 1     |
| name  | stor1 |
| nodes | - host: stor-4.example.com.vstoragedomain
|       | id: 4b83a87d-9adf-472c-91f0-782c47b2d5f1
|       | is_installing: false
|       | is_releasing: false
|       | - host: stor-3.example.com.vstoragedomain
|       | id: 7d7d37b8-4c06-4f1a-b3a6-4b54257d70ce
|       | is_installing: false
|       | is_releasing: false
|       | - host: stor-5.example.com.vstoragedomain
|       | id: fd1e46de-6e17-4571-bf6b-1ac34ec1c225
|       | is_installing: false
|       | is_releasing: false
|       | - host: stor-1.example.com.vstoragedomain
|       | id: 94d58604-6f30-4339-8578-adb7903b7277
|       | is_installing: false
|       | is_releasing: false
|       | - host: stor-2.example.com.vstoragedomain
|       | id: f59dabdb-bd1c-4944-8af2-26b8fe9ff8d4
|       | is_installing: false
|       | is_releasing: false
+-----+-----+
```

Эта команда выводит подробные данные о кластере.

ГЛАВА 3

Управление вычислительным кластером

3.1 Создание и удаление вычислительного кластера

3.1.1 `vinfra service compute create`

Создание вычислительного кластера:

```
usage: vinfra service compute create [--public-network <network>]
                                     [--subnet cidr=CIDR[,key=value,...]]
                                     [--cpu-model <cpu-model>] [--force]
                                     [--enable-k8saas] [--enable-lbaas]
                                     [--enable-metering] --nodes <nodes>
```

`--public-network <network>`

Физическая сеть для подключения к ней публичной виртуальной сети. Должна включать тип трафика «VM внешн.».

`--subnet cidr=CIDR[,key=value,...]`

Подсеть для управления IP-адресами в публичной виртуальной сети (требуется параметр `--public-network`):

- `cidr`: маска подсети в нотации CIDR;
- разделенные запятыми пары `key=value` с ключами (необязательно):
 - `gateway`: IP-адрес шлюза.

- `dhcp`: включение/отключение виртуального DHCP-сервера.
- `allocation-pool`: пул IP-адресов из CIDR в формате `ip1-ip2`, где `ip1` и `ip2` — начальный и конечный IP-адреса. Укажите ключ несколько раз, чтобы создать несколько пулов IP-адресов.
- `dns-server`: IP-адрес сервера DNS; укажите несколько раз, чтобы задать несколько DNS-серверов.

Пример: `--subnet cidr=192.168.5.0/24,dhcp=enable`.

`--cpu-model <cpu-model>`

Модель ЦП для виртуальных машин. Просмотрите список доступных моделей ЦП с помощью команды `compute cluster show`.

`--force`

Пропустить проверку минимальных требований к оборудованию.

`--enable-k8saas`

Включение сервиса «Kubernetes как услуга».

`--enable-lbaas`

Включение сервиса «балансировка нагрузки как услуга».

`--enable-metering`

Включение сервисов учета.

`--nodes <nodes>`

Список имен хостов или идентификаторов серверов через запятую.

Пример:

```
# vinfra service compute create --nodes 7ffa9540-5a20-41d1-b203-e3f349d62565,\
02ff64ae-5800-4090-b958-18b1fe8f5060,6e8afc28-7f71-4848-bdbe-7c5de64c5013,\
37c70bfb-c289-4794-8be4-b7a40c2b6d95,827a1f4e-56e5-404f-9113-88748c18f0c2 \
--public-network Public --subnet cidr=10.94.0.0/16,dhcp=enable,\
gateway=10.94.0.1,allocation-pool=10.94.129.64-10.94.129.79,\
dns-server=10.30.0.27,dns-server=10.30.0.28
+-----+-----+
| Field   | Value                                     |
+-----+-----+
| task_id | be517afa-fae0-457e-819c-f4d6399f3ae2 |
+-----+-----+
```

Эта команда создает задание для создания вычислительного кластера из пяти серверов с указанными идентификаторами. Она также указывает внешнюю сеть для VM, шлюз, пул IP-адресов для назначения виртуальным машинам и DNS-серверы, которые следует использовать.

Результат выполнения задания:

```
# vinfra task show be517afa-fae0-457e-819c-f4d6399f3ae2
+-----+-----+
| Field   | Value                                     |
+-----+-----+
| details |                                           |
| name    | backend.presentation.compute.tasks.DeployComputeClusterTask |
| progress | 100                                       |
| result  |                                           |
| state   | success                                   |
| task_id | be517afa-fae0-457e-819c-f4d6399f3ae2   |
+-----+-----+
```

3.1.2 vinfra service compute delete

Удаление всех серверов из вычислительного кластера:

```
usage: vinfra service compute delete
```

Пример:

```
# vinfra service compute delete
+-----+-----+
| Field   | Value                                     |
+-----+-----+
| task_id | 063e8a15-fcfe-4629-865f-b5e5fa44b38f |
+-----+-----+
```

Эта команда создает задание для освобождения серверов из вычислительного кластера.

Результат выполнения задания:

```
# vinfra task show 063e8a15-fcfe-4629-865f-b5e5fa44b38f
+-----+-----+
| Field   | Value                                     |
+-----+-----+
| details |                                           |
| name    | backend.presentation.compute.tasks.DestroyComputeClusterTask |
| result  |                                           |
| state   | success                                   |
| task_id | 063e8a15-fcfe-4629-865f-b5e5fa44b38f   |
+-----+-----+
```

3.2 Вывод обзора и подробных данных о вычислительном кластере

3.2.1 `vinfra service compute show`

Вывод подробных данных о вычислительном кластере:

```
usage: vinfra service compute show
```

Пример:

```
# vinfra service compute show
+-----+-----+
| Field          | Value                                |
+-----+-----+
| capabilities   | cpu_models:                          |
|                | - Nehalem                            |
|                | - Nehalem-IBRS                       |
|                | - SandyBridge                        |
|                | - SandyBridge-IBRS                  |
|                | - IvyBridge                          |
|                | - IvyBridge-IBRS                    |
|                | - Haswell                            |
|                | - Haswell-IBRS                      |
|                | - Haswell-noTSX                     |
|                | - Haswell-noTSX-IBRS                |
|                | - Broadwell                          |
|                | - Broadwell-IBRS                    |
|                | - Broadwell-noTSX                   |
|                | - Broadwell-noTSX-IBRS              |
|                | - Skylake-Client                     |
|                | - Skylake-Client-IBRS                |
|                | - Skylake-Server                     |
|                | - Skylake-Server-IBRS                |
|                | - HostPassthrough                    |
|                | os_distributions:                    |
|                | - id: linux                           |
|                |   os_type: linux                     |
|                |   title: Generic Linux                |
|                | - id: centos7                         |
|                |   os_type: linux                     |
|                |   title: CentOS 7                     |
|                | - id: centos6                         |
|                |   os_type: linux                     |
|                |   title: CentOS 6                     |
|                | - id: rhel7                           |
|                |   os_type: linux
```

```

|         | title: Red Hat Enterprise Linux 7 |
|         | - id: rhel8                      |
|         |   os_type: linux                 |
|         | title: Red Hat Enterprise Linux 8 |
|         | - id: ubuntu18.04                |
|         |   os_type: linux                 |
|         | title: Ubuntu 18.04              |
|         | - id: ubuntu16.04                |
|         |   os_type: linux                 |
|         | title: Ubuntu 16.04              |
|         | - id: debian9                    |
|         |   os_type: linux                 |
|         | title: Debian 9                  |
|         | - id: windows                     |
|         |   os_type: windows               |
|         | title: Generic Windows           |
|         | - id: win2k19                     |
|         |   os_type: windows               |
|         | title: Windows Server 2019       |
|         | - id: win2k16                     |
|         |   os_type: windows               |
|         | title: Windows Server 2016       |
|         | - id: win2k12r2                   |
|         |   os_type: windows               |
|         | title: Windows Server 2012 R2    |
|         | - id: win2k12                     |
|         |   os_type: windows               |
|         | title: Windows Server 2012       |
|         | - id: win2k8r2                    |
|         |   os_type: windows               |
|         | title: Windows Server 2008 R2    |
|         | - id: win2k8                      |
|         |   os_type: windows               |
|         | title: Windows Server 2008       |
|         | - id: win10                       |
|         |   os_type: windows               |
|         | title: Windows 10                 |
|         | - id: win8.1                      |
|         |   os_type: windows               |
|         | title: Windows 8.1                |
|         | - id: win7                        |
|         |   os_type: windows               |
|         | title: Windows 7                  |
| options | cpu_model: null                   |
| status  | active                            |
+-----+-----+

```

Эта команда отображает состояние и функциональные возможности вычислительного кластера.

3.2.2 vinfra service compute stat

Вывод статистических показателей для вычислительного кластера:

```
usage: vinfra service compute stat
```

Пример:

```
# vinfra service compute stat
+-----+-----+
| Field   | Value                                     |
+-----+-----+
| compute | block_capacity: 0                         |
|         | block_usage: 0                           |
|         | cpu_usage: 0.0                           |
|         | mem_total: 0                              |
|         | mem_usage: 0                              |
|         | vcpus: 0                                  |
| datetime | 2018-09-11T15:50:18.758258               |
| physical | block_capacity: 1099511627776            |
|         | block_free: 1099498911464                |
|         | cpu_cores: 10                             |
|         | mem_total: 41006247936                    |
| reserved | cpus: 5                                   |
|         | memory: 17721982976                       |
| servers  | count: 0                                  |
|         | error: 0                                   |
|         | in_progress: 0                            |
|         | running: 0                                 |
|         | stopped: 0                                 |
|         | top:                                       |
|         |   disk: []                                |
|         |   memory: []                              |
|         |   vcpus: []                               |
+-----+-----+
```

Эта команда выводит обзорные сведения о вычислительном кластере.

3.3 Изменение параметров вычислительного кластера

Изменение параметров вычислительного кластера:

```
usage: vinfra service compute set [--cpu-model <cpu-model>] [--enable-k8saas]
                                   [--enable-lbaas] [--enable-metering]
```

```
--cpu-model <cpu-model>
```

Задайте модель ЦП по умолчанию для виртуальных машин. Просмотрите список доступных моделей ЦП с помощью команды `compute cluster show`.

```
--enable-k8saas
```

Включение сервиса «Kubernetes как услуга».

```
--enable-lbaas
```

Включение сервиса «балансировка нагрузки как услуга».

```
--enable-metering
```

Включение сервисов учета.

Пример:

```
# vinfra service compute set --cpu-model Haswell
```

Эта команда устанавливает `Haswell` в качестве модели ЦП по умолчанию для виртуальных машин.

3.4 Управление вычислительными серверами

3.4.1 `vinfra service compute node add`

Добавление сервера в вычислительный кластер:

```
usage: vinfra service compute node add [--compute] [--controller] [--force] <node>
```

```
--compute
```

Роль вычислительного сервера

```
--controller
```

Роль вычислительного сервера контроллера

```
--force
```

Пропустить проверку минимальных требований к оборудованию

```
<node>
```

Идентификатор сервера или имя хоста

Пример:

```
# vinfra service compute node add 827a1f4e-56e5-404f-9113-88748c18f0c2 --compute
+-----+-----+
| Field   | Value                                     |
+-----+-----+
| task_id | 4c58e63c-31b6-406a-8070-9197445ec794 |
+-----+-----+
```

Эта команда создает задание для добавления сервера с идентификатором 827a1f4e-56e5-404f-9113-88748c18f0c2 в вычислительный кластер с ролью compute.

Результат выполнения задания:

```
# vinfra task show 4c58e63c-31b6-406a-8070-9197445ec794
+-----+-----+
| Field   | Value                                     |
+-----+-----+
| details |                                           |
| name    | backend.presentation.compute.tasks.AddComputeNodesTask |
| result  |                                           |
| state   | success                                   |
| task_id | 4c58e63c-31b6-406a-8070-9197445ec794 |
+-----+-----+
```

3.4.2 vinfra service compute node list

Вывод списка вычислительных серверов:

```
usage: vinfra service compute node list
```

Пример:

```
# vinfra service compute node list
+-----+-----+-----+-----+
| id                | hypervisor_hostname          | state | vms |
+-----+-----+-----+-----+
| 7ffa9540-5a20-41d1-b203-e3f349d62565 | stor-1.example.com.vstagedomain | up    | 1   |
| 6e8afc28-7f71-4848-bdbe-7c5de64c5013 | stor-3.example.com.vstagedomain | up    | 1   |
| 02ff64ae-5800-4090-b958-18b1fe8f5060 | stor-2.example.com.vstagedomain | up    | 1   |
| 827a1f4e-56e5-404f-9113-88748c18f0c2 | stor-5.example.com.vstagedomain | up    | 0   |
| 37c70bfb-c289-4794-8be4-b7a40c2b6d95 | stor-4.example.com.vstagedomain | up    | 1   |
+-----+-----+-----+-----+
```

Эта команда выводит список серверов в вычислительном кластере.

3.4.3 vinfra service compute node show

Отображение сведений о вычислительном сервере:

```
usage: vinfra service compute node show <node>
```

<node>

Идентификатор сервера или имя хоста

Пример:

```
# vinfra service compute node show 7ffa9540-5a20-41d1-b203-e3f349d62565
+-----+-----+
| Field          | Value                               |
+-----+-----+
| host           | stor-1.example.com.vstagedomain    |
| host_ip       | 10.37.130.101                       |
| hypervisor    | id: 86f1ca2c-71c7-47a0-9c7f-bb9dd705e67e |
|               | state: up                            |
|               | status: enabled                      |
|               | vms: 0                               |
| id            | 7ffa9540-5a20-41d1-b203-e3f349d62565 |
| orig_hostname | stor-1.example.com                  |
| roles         | - controller                         |
|               | - compute                            |
| state        | healthy                             |
+-----+-----+
```

Эта команда отображает сведения о вычислительном сервере с идентификатором 7ffa9540-5a20-41d1-b203-e3f349d62565.

3.4.4 vinfra service compute node fence

Ограждение вычислительного сервера:

```
usage: vinfra service compute node fence <node>
```

<node>

Идентификатор сервера или имя хоста

Пример:

```
# vinfra service compute node fence e6255aed-d6e7-41b2-ba90-86164c1cd9a6
Operation successful
```

Эта команда ограждает сервер с идентификатором e6255aed-d6e7-41b2-ba90-86164c1cd9a6.

3.4.5 vinfra service compute node unfence

Отмена ограждения вычислительного сервера:

```
usage: vinfra service compute node unfence <node>
```

<node>

Идентификатор сервера или имя хоста

Пример:

```
# vinfra service compute node unfence e6255aed-d6e7-41b2-ba90-86164c1cd9a6
Operation successful
```

Эта команда снимает ограждение сервера с идентификатором e6255aed-d6e7-41b2-ba90-86164c1cd9a6.

3.4.6 vinfra service compute node release

Освобождение сервера из вычислительного кластера:

```
usage: vinfra service compute node release [--compute] [--controller] <node>
```

--compute

Роль вычислительного сервера

--controller

Роль вычислительного сервера контроллера

<node>

Идентификатор сервера или имя хоста

Пример:

```
# vinfra service compute node release 827a1f4e-56e5-404f-9113-88748c18f0c2
+-----+-----+
| Field  | Value                                |
+-----+-----+
| task_id | 3b39738c-80a6-40a6-a50d-c3c8118ed212 |
+-----+-----+
```

Эта команда создает задание для освобождения сервера с идентификатором 827a1f4e-56e5-404f-9113-88748c18f0c2 из вычислительного кластера.

Результат выполнения задания:

```
# vinfra task show 3b39738c-80a6-40a6-a50d-c3c8118ed212
+-----+-----+-----+
| Field | Value |
+-----+-----+-----+
| details |
| name | backend.presentation.compute.tasks.DeleteComputeNodesTask |
| result |
| state | success |
| task_id | 3b39738c-80a6-40a6-a50d-c3c8118ed212 |
+-----+-----+-----+
```

3.5 Управление виртуальными машинами

3.5.1 vinfra service compute server create

Создание нового сервера вычислений:

```
usage: vinfra service compute server create [--description <description>]
                                           [--metadata <metadata>]
                                           [--user-data <user-data>]
                                           [--key-name <key-name>]
                                           [--config-drive] [--count <count>]
                                           [--ha-enabled {true,false}]
                                           [--placements <placements>]
                                           --network <id=id[,key=value,...]>
                                           --volume <source=source[,key=value,...]>
                                           --flavor <flavor> <server-name>
```

--description <description>

Описание сервера

--metadata <метаданные>

Метаданные сервера

--user-data <пользовательские_данные>

Файл пользовательских данных

--key-name <key-name>

Пара ключей для внедрения

--config-drive

Использовать временный (эфемерный) диск

`--count <число>`

Если указано число, и оно больше 1, то аргумент имя рассматривается как шаблон присвоения имен.

`--ha-enabled {true, false}`

Включение или отключение высокой доступности для сервера вычислений.

`--placements <размещения>`

Имена или идентификаторы размещений, в которые следует добавить вычислительный сервер.

`--network <id=идентификатор[, ключ=значение, ...]>`

Создает сервер вычислений в указанной сети. Укажите этот параметр несколько раз, чтобы создать несколько сетей.

- `id`: присоединить сетевой интерфейс к указанной (по идентификатору или имени) сети
- разделенные запятыми пары `key=value` с ключами (необязательно):
 - `mac`: MAC-адрес для сетевого интерфейса
 - `fixed-ip`: фиксированный IP-адрес для сетевого интерфейса
 - `spoofing-protection`: включение или отключение защиты от спуфинга пакетов на сетевом интерфейсе (`on` для включения или `off` для выключения)

`--volume <source=источник[, ключ=значение, ...]>`

Создает сервер вычислений с указанным томом. Укажите этот параметр несколько раз, чтобы создать несколько томов.

- `source`: тип источника (`volume` для тома, `image` для образа, `snapshot` для снимка или `blank` — пустой)
- разделенные запятыми пары `key=value` с ключами (необязательно):
 - `id`: идентификатор или имя ресурса для указанного типа источника (требуется для источников типа `volume` — том, `image` — образ и `snapshot` — снимок)
 - `size`: размер блочного устройства в гигабайтах (требуется для источников типа `image` — образ и `blank` — пустой)
 - `boot-index`: загрузочный индекс блочного устройства (требуется при наличии нескольких томов с типом источника `volume`)
 - `bus`: тип контроллера блочного устройства (`scsi`)

- `type`: тип блочного устройства (`disk` или `cdrom`)
- `rm`: удалить блочное устройство по прекращении работы сервера вычислений (`yes` — да или `no` — нет)
- `storage-policy`: политика хранилища блочного устройства

`--flavor <flavor>`

Идентификатор или имя типа VM

`<имя_сервера>`

Новое имя для сервера вычислений

Пример:

```
# vinfra service compute server create myvm \
--network id=private,fixed-ip=192.168.128.100 \
--volume source=image,id=cirros,size=1 --flavor tiny
+-----+
| Field      | Value                                     |
+-----+
| config_drive |                                           |
| created      | 2019-05-29T11:24:04Z                     |
| description  |                                           |
| flavor       | disk: 0                                   |
|              | ephemeral: 0                             |
|              | extra_specs: {}                          |
|              | original_name: tiny                      |
|              | ram: 512                                  |
|              | swap: 0                                   |
|              | vcpus: 1                                  |
| ha_enabled   | True                                       |
| host         |                                           |
| id           | 8cd29296-8bee-4efb-828d-0e522d816c6e     |
| key_name     |                                           |
| metadata     | {}                                         |
| name         | myvm                                       |
| networks     | []                                         |
| power_state  | NOSTATE                                   |
| project_id   | b4267de6fd0c442da99542cd20f5932c       |
| status       | BUILD                                     |
| task_state   | scheduling                                |
| updated      | 2019-05-29T11:24:21Z                     |
| user_data    |                                           |
| volumes      | []                                         |
+-----+
```

Эта команда создает виртуальную машину `myvm`, основанную на образе `Cirros` по умолчанию и типе `tiny`, подключает ее к сети `private` с фиксированным IP-адресом `192.168.128.100` и включает для нее высокую доступность.

3.5.2 vinfra service compute server list

Вывод списка серверов вычислений:

```
usage: vinfra service compute server list
```

Пример:

```
# vinfra service compute server list
+-----+-----+-----+-----+
| id                | name | status | host                |
+-----+-----+-----+-----+
| 8cd29296-8bee-4efb-828d-0e522d816c6e | myvm | ACTIVE | node001.vstoragedomain |
+-----+-----+-----+-----+
```

Эта команда выводит список всех виртуальных машин в вычислительном кластере.

3.5.3 vinfra service compute server show

Вывод подробных сведений о сервере вычислений:

```
usage: vinfra service compute server show <server>
```

<сервер>

Идентификатор или имя сервера вычислений

Пример:

```
# vinfra service compute server show myvm
+-----+-----+
| Field          | Value                |
+-----+-----+
| config_drive   |                       |
| created        | 2019-05-29T11:24:04Z |
| description    |                       |
| flavor         | disk: 0              |
|                | ephemeral: 0         |
|                | extra_specs: {}      |
|                | original_name: tiny  |
|                | ram: 512              |
|                | swap: 0               |
|                | vcpus: 1              |
| ha_enabled     | True                 |
| host           | node001.vstoragedomain |
| id            | 8cd29296-8bee-4efb-828d-0e522d816c6e |
| key_name      |                       |
| metadata      | {}                   |
+-----+-----+
```

```

| name          | myvm          |
| networks     | - id: 79b3da71-c6a2-49e8-97f8-9431a065bed7 |
|               | ipam_enabled: true |
|               | ips:          |
|               | - 192.168.128.100 |
|               | mac_addr: fa:16:3e:d8:42:f6 |
|               | name: private  |
|               | spoofing_protection: true |
| orig_hostname | node001      |
| power_state  | RUNNING      |
| project_id   | b4267de6fd0c442da99542cd20f5932c |
| status       | ACTIVE       |
| task_state   |              |
| updated      | 2019-05-29T11:24:21Z |
| user_data    |              |
| volumes      | - delete_on_termination: false |
|               | id: edd3df0a-95f5-4892-9053-2793a3976f94 |
+-----+-----+

```

Эта команда отображает подробные данные о виртуальной машине myvm.

3.5.4 vinfra service compute server stat

Вывод статистических показателей для сервера вычислений:

```
usage: vinfra service compute server stat <server>
```

<сервер>

Идентификатор или имя сервера вычислений

Пример:

```

# vinfra service compute server stat myvm
+-----+-----+
| Field    | Value |
+-----+-----+
| datetime | 2019-05-29T11:39:46.429000+00:00 |
| metrics  | block_capacity: 1073741824 |
|           | block_usage: 268435456 |
|           | cpu_usage: 0 |
|           | mem_usage: 149876736 |
+-----+-----+

```

Эта команда отображает статистические показатели для виртуальной машины myvm.

3.5.5 vinfra service compute server set

Изменение параметров сервера вычислений:

```
usage: vinfra service compute server set [--name <name>] [--description <description>]
                                           [--ha-enabled <ha_enabled>]
                                           [--no-placements | --placement placement]
                                           <server>
```

--name <name>

Новое имя для сервера вычислений

--description <description>

Новое описание сервера вычислений

--ha-enabled {true, false}

Включение или отключение высокой доступности для сервера вычислений.

--no-placements

Очистка размещений с сервера вычислений.

--placement <размещение>

Имя или идентификатор размещения, к которому необходимо добавить сервер вычислений.

Укажите этот параметр несколько раз, чтобы добавить сервер вычислений в несколько размещений.

<сервер>

Идентификатор или имя сервера вычислений

Пример:

```
# vinfra service compute server set myvm --description "My new VM" --ha-enabled false
+-----+-----+
| Field      | Value                                |
+-----+-----+
| config_drive |                                       |
| created      | 2019-05-29T11:24:04Z                 |
| description  | My new VM                             |
| flavor      | disk: 0                               |
|              | ephemeral: 0                         |
|              | extra_specs: {}                      |
|              | original_name: tiny                  |
|              | ram: 512                              |
|              | swap: 0                               |
|              | vcpus: 1                              |
| ha_enabled  | False                                 |
+-----+-----+
```

```

| host           | node001.vstoragedomain |
| id            | 8cd29296-8bee-4efb-828d-0e522d816c6e |
| key_name     | |
| metadata     | {} |
| name         | myvm |
| networks     | - id: 79b3da71-c6a2-49e8-97f8-9431a065bed7 |
|              |   ipam_enabled: true |
|              |   ips: |
|              |   - 192.168.128.100 |
|              |   mac_addr: fa:16:3e:d8:42:f6 |
|              |   name: private |
|              |   spoofing_protection: true |
| orig_hostname | node001 |
| power_state  | RUNNING |
| project_id   | b4267de6fd0c442da99542cd20f5932c |
| status       | ACTIVE |
| task_state   | |
| updated      | 2019-05-29T11:24:21Z |
| user_data    | |
| volumes     | - delete_on_termination: false |
|              |   id: edd3df0a-95f5-4892-9053-2793a3976f94 |
+-----+-----+

```

Эта команда добавляет описание к виртуальной машине myvm и отключает для нее высокую доступность.

3.5.6 vinfra service compute server iface attach

Присоединение сети к серверу вычислений:

```

usage: vinfra service compute server iface attach [--mac <mac>] [--ip <ip-address>]
                                                [--spoofing-protection {on,off}]
                                                --server <server> --network <network>

```

--mac <mac-адрес>

MAC-адрес

--ip <ip-адрес>

IP-адрес

--spoofing-protection {on|off}

Включает защиту от спуфинга пакетов на сетевом интерфейсе

--server <сервер>

Идентификатор или имя сервера вычислений

```
--network <network>
```

Идентификатор или имя сети

Пример:

```
# vinfra service compute server iface attach --network myprivnet --server myvm
+-----+-----+-----+-----+
| Field          | Value                                     |
+-----+-----+-----+-----+
| fixed_ip       | 192.168.129.8                             |
| id             | 690ed3f2-2301-40e2-879a-126db2ecb57b     |
| mac_address    | fa:16:3e:54:59:08                         |
| network_id     | 0710372e-2bdf-4dfe-b413-eb763da37e68     |
| spoofing<...> | False                                     |
+-----+-----+-----+-----+
```

Эта команда присоединяет частную сеть myprivnet к виртуальной машине myvm.

3.5.7 vinfra service compute server iface list

Вывод списка сетей сервера вычислений:

```
usage: vinfra service compute server iface list --server <server>
```

```
--server <сервер>
```

Идентификатор или имя сервера вычислений

Пример:

```
# vinfra service compute server iface list --server myvm
+-----+-----+-----+-----+
| id          | network_id | mac_address | fixed_ip |
+-----+-----+-----+-----+
| 690ed3f2-<...> | 0710372e-<...> | fa:16:3e:54:59:08 | 192.168.129.8 |
| a5b13bf3-<...> | 1bf2c9da-<...> | fa:16:3e:b9:33:bb | 192.168.128.100 |
+-----+-----+-----+-----+
```

Эта команда выводит список виртуальных сетей, к которым подсоединена виртуальная машина myvm. Она также отображает IP-адрес виртуальной машины в каждой из этих сетей.

3.5.8 vinfra service compute server iface detach

Отсоединение сетевого интерфейса от сервера вычислений:

```
usage: vinfra service compute server iface detach --server <server> <interface>
```

--server <сервер>

Идентификатор или имя сервера вычислений

<интерфейс>

Идентификатор сетевого интерфейса

Пример:

```
# vinfra service compute server iface detach 471e37fd-13ae-4b8f-b70c-90ac02cc4386 \
--server 6c80b07f-da46-4a8a-89a4-eeeb8faceb27
Operation successful
```

Эта команда отсоединяет сетевой интерфейс с идентификатором 471e37fd-13ae-4b8f-b70c-90ac02cc4386 от виртуальной машины с идентификатором 6c80b07f-da46-4a8a-89a4-eeeb8faceb27.

3.5.9 vinfra service compute server volume attach

Присоединение тома к серверу вычислений:

```
usage: vinfra service compute server volume attach --server <server> <volume>
```

--server <сервер>

Идентификатор или имя сервера вычислений

<том>

Идентификатор или имя тома

Пример:

```
# vinfra service compute server volume attach e4cb5363-1fb2-41f5-b24b-18f98a388cba \
--server 871fef54-519b-4111-b18d-d2039e2410a8
+-----+-----+
| Field | Value |
+-----+-----+
| device | /dev/vdb |
| id     | e4cb5363-1fb2-41f5-b24b-18f98a388cba |
+-----+-----+
```

Эта команда присоединяет доступный том с идентификатором e4cb5363-1fb2-41f5-b24b-18f98a388cba к виртуальной машине с идентификатором 871fef54-519b-4111-b18d-d2039e2410a8.

3.5.10 vinfra service compute server volume list

Вывод списка томов сервера вычислений:

```
usage: vinfra service compute server volume list --server <server>
```

--server <сервер>

Идентификатор или имя сервера вычислений

Пример:

```
# vinfra service compute server volume list --server myvm
+-----+-----+
| id                | device  |
+-----+-----+
| e4cb5363-1fb2-41f5-b24b-18f98a388cba | /dev/vdb |
| b325cc6e-8de1-4b6c-9807-5a497e3da7e3 | /dev/vda |
+-----+-----+
```

Эта команда выводит список томов, присоединенных к виртуальной машине myvm.

3.5.11 vinfra service compute server volume show

Отображение подробных данных о томе сервера вычислений:

```
usage: vinfra service compute server volume show --server <server> <volume>
```

--server <сервер>

Идентификатор или имя сервера вычислений

<том>

Идентификатор или имя тома

Пример:

```
# vinfra service compute server volume show --server myvm \
e4cb5363-1fb2-41f5-b24b-18f98a388cba
+-----+-----+
| Field | Value |
+-----+-----+
| device | /dev/vdb |
+-----+-----+
```

```
| id      | e4cb5363-1fb2-41f5-b24b-18f98a388cba |
+-----+-----+
```

Эта команда отображает подробные данные о томе с идентификатором e4cb5363-1fb2-41f5-b24b-18f98a388cba, присоединенном к виртуальной машине myvm.

3.5.12 vinfra service compute server volume detach

Отсоединение тома от сервера вычислений:

```
usage: vinfra service compute server volume detach --server <server> <volume>
```

--server <сервер>

Идентификатор или имя сервера вычислений

<том>

Идентификатор или имя тома

Пример:

```
# vinfra service compute server volume detach e4cb5363-1fb2-41f5-b24b-18f98a388cba \
--server 871fef54-519b-4111-b18d-d2039e2410a8
Operation successful
```

Эта команда отсоединяет том с идентификатором e4cb5363-1fb2-41f5-b24b-18f98a388cba от виртуальной машины с идентификатором 871fef54-519b-4111-b18d-d2039e2410a8.

3.5.13 vinfra service compute server log

Отображение журнала сервера вычислений:

```
usage: vinfra service compute server log <server>
```

<сервер>

Идентификатор или имя сервера вычислений

Пример:

```
# vinfra service compute server log myvm > myvm.log
```

Эта команда сохраняет журнал виртуальной машины myvm в файл myvm.log.

3.5.14 vinfra service compute server migrate

Миграция сервера вычислений на другой хост:

```
usage: vinfra service compute server migrate [--cold] [--node <node>] <server>
```

`--cold`

Выполнение холодной миграции. Если параметр не установлен, тип миграции определяется автоматически.

`--node <сервер>`

Идентификатор или имя хоста для сервера назначения

`<сервер>`

Идентификатор или имя сервера вычислений

Пример:

```
# vinfra service compute server migrate 6c80b07f-da46-4a8a-89a4-eeeb8faceb27 \  
--node e6255aed-d6e7-41b2-ba90-86164c1cd9a6  
Operation successful
```

Эта команда запускает миграцию виртуальной машины с идентификатором 6c80b07f-da46-4a8a-89a4-eeeb8faceb27 на сервер вычислений с идентификатором e6255aed-d6e7-41b2-ba90-86164c1cd9a6.

3.5.15 vinfra service compute server resize

Изменение размеров сервера вычислений:

```
usage: vinfra service compute server resize --flavor <flavor> <server>
```

`--flavor <flavor>`

Применение типа с идентификатором или именем

`<сервер>`

Идентификатор или имя сервера вычислений

Пример:

```
# vinfra service compute server resize myvm --flavor small  
Operation successful
```

Эта команда меняет тип виртуальной машины myvm на small.

3.5.16 `vinfra service compute server start`

Запуск сервера вычислений:

```
usage: vinfra service compute server start <server>
```

<сервер>

Идентификатор или имя сервера вычислений

Пример:

```
# vinfra service compute server start myvm  
Operation successful
```

Эта команда запускает виртуальную машину `myvm`.

3.5.17 `vinfra service compute server pause`

Приостановка работы сервера вычислений:

```
usage: vinfra service compute server pause <server>
```

<сервер>

Идентификатор или имя сервера вычислений

Пример:

```
# vinfra service compute server pause myvm
```

Эта команда приостанавливает работающую виртуальную машину `myvm`.

3.5.18 `vinfra service compute server unpause`

Возобновление работы сервера вычислений после приостановки:

```
usage: vinfra service compute server unpause <server>
```

<сервер>

Идентификатор или имя сервера вычислений

Пример:

```
# vinfra service compute server unpause myvm
```

Эта команда возобновляет работу ранее приостановленной виртуальной машины myvm.

3.5.19 vinfra service compute server suspend

«Замораживание» сервера вычислений:

```
usage: vinfra service compute server suspend <server>
```

<сервер>

Идентификатор или имя сервера вычислений

Пример:

```
# vinfra service compute server suspend myvm  
Operation successful
```

Эта команда «замораживает» работающую виртуальную машину myvm.

3.5.20 vinfra service compute server resume

Возобновление работы «замороженного» сервера вычислений:

```
usage: vinfra service compute server resume <server>
```

<сервер>

Идентификатор или имя сервера вычислений

Пример:

```
# vinfra service compute server resume myvm  
Operation successful
```

Эта команда возобновляет работу ранее «замороженной» виртуальной машины myvm.

3.5.21 vinfra service compute server reboot

Перезагрузка сервера вычислений:

```
usage: vinfra service compute server reboot [--hard] <server>
```

`--hard`

Выполнение «жесткой» перезагрузки

`<сервер>`

Идентификатор или имя сервера вычислений

Пример:

```
# vinfra service compute server reboot myvm
Operation successful
```

Эта команда перезагружает виртуальную машину `myvm`.

3.5.22 vinfra service compute server reset-state

Сброс состояния сервера вычислений:

```
usage: vinfra service compute server reset-state [--state-error] <server>
```

`--state-error`

Сброс сервера в состояние «ERROR» (ошибка)

`<сервер>`

Идентификатор или имя сервера вычислений

Пример:

```
# vinfra service compute server reset-state myvm
Operation successful
```

Эта команда сбрасывает переходное состояние виртуальной машины `myvm` до предыдущего состояния.

3.5.23 vinfra service compute server stop

Выключение сервера вычислений:

```
usage: vinfra service compute server stop [--hard] <server>
```

--hard

Полное выключение сервера вычислений

<сервер>

Идентификатор или имя сервера вычислений

Пример:

```
# vinfra service compute server stop myvm
Operation successful
```

Эта команда останавливает работу виртуальной машины myvm.

3.5.24 vinfra service compute server shelve

Освобождение ресурсов сервера вычислений:

```
usage: vinfra service compute server shelve <server>
```

<сервер>

Идентификатор или имя сервера вычислений.

Пример:

```
# vinfra service compute server shelve myvm
```

Эта команда отменяет привязку виртуальной машины myvm к серверу, на котором она была размещена, и высвобождает ее зарезервированные ресурсы, такие как ЦП и ОЗУ.

3.5.25 vinfra service compute server unshelve

Назначение ресурсов сервера вычислений:

```
usage: vinfra service compute server unshelve <server>
```

<сервер>

Идентификатор или имя сервера вычислений.

Пример:

```
# vinfra service compute server unshelve myvm
```

Эта команда порождает виртуальную машину `myvm` на сервере с достаточным количеством ресурсов для ее размещения.

3.5.26 `vinfra service compute server evacuate`

Эвакуация остановленного сервера вычислений с хоста, на котором произошел отказ:

```
usage: vinfra service compute server evacuate <server>
```

<сервер>

Идентификатор или имя сервера вычислений

Пример:

```
# vinfra service compute server evacuate myvm  
Operation successful
```

Эта команда эвакуирует остановленную виртуальную машину `myvm` с ее сервера на другой, работоспособный сервер вычислений.

3.5.27 `vinfra service compute server delete`

Удаление сервера вычислений:

```
usage: vinfra service compute server delete <server>
```

<сервер>

Идентификатор или имя сервера вычислений

Пример:

```
# vinfra service compute server delete myvm  
Operation successful
```

Эта команда удаляет виртуальную машину `myvm`.

3.6 Управление образами

3.6.1 vinfra service compute image create

Создание нового вычислительного образа:

```
usage: vinfra service compute image create [--min-disk <size-gb>] [--min-ram <size-mb>]
      [--os-distro <os-distro>] [--protected]
      [--disk-format <disk_format>]
      [--container-format <format>]
      --file <file> <image-name>
```

`--min-disk <size-gb>`

Минимальный размер диска, необходимый для загрузки с образа, в гигабайтах

`--min-ram <size-mb>`

Минимальный размер ОЗУ, необходимый для загрузки с образа, в мегабайтах

`--os-distro <os-distro>`

Дистрибутив ОС. Чтобы вывести список доступных дистрибутивов, выполните команду `service compute cluster show`.

`--protected`

Защита образа от удаления

`--disk-format <disk_format>`

Формат диска: `aki`, `ami`, `ari`, `detect`, `iso`, `ploop`, `qcow2`, `raw`, `vdi`, `vhd`, `vhdx`, `vmdk` (по умолчанию `detect`)

`--container-format <format>`

Формат контейнера: `aki`, `ami`, `ari`, `bare`, `docker`, `ovf`, `ova` (по умолчанию `bare`)

`--file <file>`

Создание образа из локального файла

`<image-name>`

Имя образа

Пример:

```
# vinfra service compute image create mycirrosimg \
--file /distr/cirros-0.4.0-x86_64-disk.img
Uploading image to server [elapsed time: 0:00:04]... |
```

```
+-----+-----+
| Field   | Value                                     |
+-----+-----+
| task_id | 03874663-d03f-4891-a10b-64837e7faf43 |
+-----+-----+
```

Эта команда создает задание для создания образа Cirros из локального файла и его загрузки в продукт Acronis Инфраструктура.

Результат выполнения задания:

```
# vinfra task show 03874663-d03f-4891-a10b-64837e7faf43
+-----+-----+
| Field   | Value                                     |
+-----+-----+
| details | |
| name    | backend.presentation.compute.images.tasks.ImportComputeImageTask |
| result  | id: 179f45ef-c5d6-4270-b0c0-085b542544c5 |
| state   | success                                   |
| task_id | 03874663-d03f-4891-a10b-64837e7faf43 |
+-----+-----+
```

3.6.2 vinfra service compute image list

Вывод списка вычислительных образов:

```
usage: vinfra service compute image list
```

Пример:

Эта команда выводит список образов, доступных для вычислительного кластера.

```
# vinfra service compute image list
+-----+-----+-----+-----+-----+
| id                | name          | size  | status | disk_format |
+-----+-----+-----+-----+-----+
| 179f45ef-c5d6-4270-b0c0-085b542544c5 | mycirrosimg  | 12716032 | active | qcow2       |
| 4741274f-5cca-4205-8f66-a2e89fb346cc | cirros       | 12716032 | active | qcow2       |
+-----+-----+-----+-----+-----+
```

3.6.3 vinfra service compute image show

Отображение сведений о вычислительном образе:

```
usage: vinfra service compute image show <image>
```


<image>

Идентификатор или имя образа

Пример:

```
# vinfra service compute image show 4741274f-5cca-4205-8f66-a2e89fb346cc
+-----+-----+
| Field          | Value                                     |
+-----+-----+
| checksum       | 443b7623e27ecf03dc9e01ee93f67afe       |
| container_format | bare                                     |
| created_at     | 2018-09-11T13:29:10Z                   |
| disk_format    | qcow2                                    |
| file           | /api/v2/compute/images/4741274f-5cca-4205-8f66-a2e89fb346cc/file/ |
| id             | 4741274f-5cca-4205-8f66-a2e89fb346cc   |
| min_disk      | 1                                        |
| min_ram       | 0                                        |
| name           | cirros                                   |
| os_distro      | linux                                    |
| os_type        | linux                                    |
| project_id     | 72a5db3a033c403a86756021e601ef34      |
| protected      | False                                    |
| size           | 12716032                                 |
| status         | active                                    |
| tags           | []                                       |
| updated_at     | 2018-09-11T13:29:13Z                   |
| virtual_size   |                                           |
| visibility     | public                                    |
+-----+-----+
```

Эта команда отображает сведения о стандартном образе Cirros.

3.6.4 vinfra service compute image set

Изменение параметров вычислительного образа:

```
usage: vinfra service compute image set [--min-disk <size-gb>] [--min-ram <size-mb>]
                                         [--os-distro <os-distro>] [--protected]
                                         [--name <name>] <image>
```

`--min-disk <size-gb>`

Минимальный размер диска, необходимый для загрузки с образа, в гигабайтах

`--min-ram <size-mb>`

Минимальный размер ОЗУ, необходимый для загрузки с образа, в мегабайтах

`--os-distro <os-distro>`

Дистрибутив ОС. Чтобы вывести список доступных дистрибутивов, выполните команду `service compute cluster show`.

`--protected`

Защита образа от удаления

`--name <name>`

Имя образа

`<image>`

Идентификатор или имя образа

Пример:

```
# vinfra service compute image set 4741274f-5cca-4205-8f66-a2e89fb346cc --protected --min-r
+-----+-----+-----+-----+-----+-----+-----+-----+-----+
| Field          | Value                                                                 |
+-----+-----+-----+-----+-----+-----+-----+-----+-----+
| checksum       | 443b7623e27ecf03dc9e01ee93f67afe                                     |
| container_format | bare                                                                    |
| created_at     | 2018-09-11T13:29:10Z                                                  |
| disk_format    | qcow2                                                                    |
| file           | /api/v2/compute/images/4741274f-5cca-4205-8f66-a2e89fb346cc/file/    |
| id             | 4741274f-5cca-4205-8f66-a2e89fb346cc                                  |
| min_disk       | 1                                                                        |
| min_ram        | 1                                                                        |
| name           | cirros                                                                    |
| os_distro      | linux                                                                    |
| os_type        | linux                                                                    |
| project_id     | 72a5db3a033c403a86756021e601ef34                                       |
| protected      | True                                                                      |
| size           | 12716032                                                                  |
| status         | active                                                                    |
| tags           | []                                                                        |
| updated_at     | 2018-09-12T09:26:29Z                                                  |
| virtual_size   |                                                                            |
| visibility     | public                                                                    |
+-----+-----+-----+-----+-----+-----+-----+-----+-----+
```

Эта команда защищает стандартный образ Cirros и задает для него минимальный размер ОЗУ в 1 Гб.

3.6.5 vinfra service compute image save

Скачивание вычислительного образа:

```
usage: vinfra service compute image save [--file <filename>] <image>
```

```
--file <filename>
```

Файл, в который следует сохранить образ (по умолчанию stdout)

```
<image>
```

Идентификатор или имя образа

Пример:

```
# vinfra service compute image save 4741274f-5cca-4205-8f66-a2e89fb346cc --file cirros.qcow2
Operation successful
```

Эта команда скачивает стандартный образ Cirros на локальный диск как `cirros.qcow2`.

3.6.6 vinfra service compute image delete

Удаление вычислительного образа:

```
usage: vinfra service compute image delete <image>
```

```
<image>
```

Идентификатор или имя образа

Пример:

```
# vinfra service compute image delete 179f45ef-c5d6-4270-b0c0-085b542544c5
Operation successful
```

Эта команда удаляет образ с идентификатором `179f45ef-c5d6-4270-b0c0-085b542544c5`.

3.7 Управление размещениями

3.7.1 vinfra service compute placement create

Создание нового вычислительного размещения:

```
usage: vinfra service compute placement create [--description <description>]
        [--nodes <nodes>]
        [--images <images>]
        <placement-name>
```

`--description <description>`

Описание размещения

`--nodes <nodes>`

Список идентификаторов вычислительных серверов через запятую для назначения в размещение

`--images <images>`

Список идентификаторов образов через запятую для назначения в размещение

`<placement-name>`

Имя размещения

Пример:

```
# vinfra service compute placement create placement1
+-----+-----+
| Field      | Value                                |
+-----+-----+
| description |                                       |
| id          | e4230b75-a858-404c-be3b-4b3f2dedb057 |
| images      | 0                                     |
| name        | placement1                            |
| nodes       | 0                                     |
| servers     | 0                                     |
+-----+-----+
```

Эта команда создает размещение с именем `placement1`.

3.7.2 `vinfra service compute placement assign`

Назначение серверов или образов в размещение:

```
usage: vinfra service compute placement assign (--images <images> | --nodes <nodes>)
        <placement>
```

`--images <images>`

Список идентификаторов образов через запятую для назначения в размещение

`--nodes <nodes>`

Список идентификаторов вычислительных серверов через запятую для назначения в размещение

`<placement>`

Идентификатор или имя размещения

Если на сервере есть виртуальные машины, они не будут автоматически наследовать изменения в конфигурации размещения сервера. То есть, если добавить сервер с виртуальными машинами в размещение, VM не будут назначены этому размещению. Если удалить такой сервер из размещения, то его VM сохранят это размещение. Необходимо будет соответствующим образом изменить конфигурацию размещения VM с помощью команды `vinfra service compute server set`. Убедитесь, что сервер и его VM имеют одинаковую конфигурацию размещения.

Пример:

```
# vinfra service compute placement assign \  
--images b23e23e8-7338-4a09-a827-3c9c509cf35c placement1  
Operation successful.
```

Эта команда назначает образ с идентификатором `b23e23e8-7338-4a09-a827-3c9c509cf35c` в размещение `placement1`.

3.7.3 `vinfra service compute placement delete assign`

Удаление образов и серверов из вычислительного размещения:

```
usage: vinfra service compute placement delete assign  
      (--image-id <images> | --node-id <nodes>)  
      <placement>
```

`--image-id <images>`

Идентификатор образа для удаления из вычислительного размещения

`--node-id <nodes>`

Идентификатор вычислительного сервера для удаления из размещения

`<placement>`

Идентификатор или имя размещения

Если на сервере есть виртуальные машины, они не будут автоматически наследовать изменения в конфигурации размещения сервера. То есть, если добавить сервер с виртуальными машинами в размещение, VM не будут назначены этому размещению. Если удалить такой сервер из размещения, то его VM сохранят это размещение. Необходимо будет соответствующим образом изменить конфигурацию размещения VM с помощью команды `vinfra service compute server set`. Убедитесь, что сервер и его VM имеют одинаковую конфигурацию размещения.

Пример:

```
# vinfra service compute placement delete assign \
--image b23e23e8-7338-4a09-a827-3c9c509cf35c placement1
Operation successful.
```

Эта команда удаляет образ с идентификатором b23e23e8-7338-4a09-a827-3c9c509cf35c из размещения placement1.

3.7.4 vinfra service compute placement list

Вывод списка вычислительных размещений:

```
usage: vinfra service compute placement list
```

Пример:

```
# vinfra service compute placement list
+-----+-----+-----+-----+-----+
| id          | name          | description | nodes | images | servers |
+-----+-----+-----+-----+-----+
| 2d152d33-... | placement1    |             | 0     | 1     | 0     |
| 11857e11-... | placement2    |             | 1     | 0     | 0     |
+-----+-----+-----+-----+-----+
```

Эта команда выводит список доступных вычислительных размещений.

3.7.5 vinfra service compute placement show

Отображение сведений о вычислительном размещении:

```
usage: vinfra service compute placement show <placement>
```

<placement>

Идентификатор или имя размещения

Пример:

```
# vinfra service compute placement show placement1
+-----+-----+
| Field      | Value                                     |
+-----+-----+
| description |                                           |
| id         | e4230b75-a858-404c-be3b-4b3f2dedb057 |
| images     | 0                                         |
| name      | placement1                               |
| nodes     | 0                                         |
+-----+-----+
```

```
| servers | 0 |
+-----+-----+
```

Эта команда отображает сведения о размещении placement1.

3.7.6 vinfra service compute placement update

Обновление вычислительного размещения:

```
usage: vinfra service compute placement update [--name <placement-name>]
        [--description <placement-description>]
        <placement>
```

--name <placement-name>

Новое имя для размещения

--description <placement-description>

Новое описание для размещения

<placement>

Идентификатор или имя размещения

Пример:

```
# vinfra service compute placement update --name placement1-UPD placement1
Operation successful
```

Эта команда изменяет имя размещения с placement1 на placement1-UPD.

3.7.7 vinfra service compute placement delete

Удаление вычислительного размещения:

```
usage: vinfra service compute placement delete <placement>
```

<placement>

Идентификатор или имя размещения

Пример:

```
# vinfra service compute placement delete placement1
Operation successful
```

Эта команда удаляет размещение placement1.

3.8 Управление типами виртуальных машин

3.8.1 `vinfra service compute flavor create`

Создание нового вычислительного типа VM:

```
usage: vinfra service compute flavor create [--swap <size-mb>] --vcpus <vcpus>
      --ram <size-mb> <flavor-name>
```

`--swap <size-mb>`

Размер пространства подкачки в мегабайтах

`--vcpus <vcpus>`

Количество виртуальных ЦП

`--ram <size-mb>`

Размер памяти в мегабайтах

`<flavor-name>`

Имя типа VM

Пример:

```
# vinfra service compute flavor create myflavor --vcpus 1 --ram 3072
+-----+-----+
| Field | Value |
+-----+-----+
| id    | 561a48ea-0c1c-4152-8b7d-e4b4af276c2d |
| name  | myflavor |
| ram   | 3072 |
| swap  | 0 |
| vcpus | 1 |
+-----+-----+
```

Эта команда создает тип VM `myflavor` с 1 виртуальным ЦП и 3 ГБ ОЗУ.

3.8.2 `vinfra service compute flavor list`

Вывод списка вычислительных типов VM:

```
usage: vinfra service compute flavor list
```


Пример:

```
# vinfra service compute flavor list
+-----+-----+-----+-----+
| id          | name      | ram   | swap  | vcpus |
+-----+-----+-----+-----+
| 100         | tiny      | 512   | 0     | 1     |
| 101         | small     | 2048  | 0     | 1     |
| 102         | medium    | 4096  | 0     | 2     |
| 103         | large     | 8192  | 0     | 4     |
| 104         | xlarge    | 16384 | 0     | 8     |
| 561a48ea-0c1c-4152-8b7d-e4b4af276c2d | myflavor  | 3072  | 0     | 1     |
+-----+-----+-----+-----+
```

Эта команда выводит список всех типов VM.

3.8.3 vinfra service compute flavor show

Отображение сведений о вычислительном типе VM:

```
usage: vinfra service compute flavor show <flavor>
```

<flavor>

Идентификатор или имя типа VM

Пример:

```
# vinfra service compute flavor show myflavor
+-----+-----+
| Field | Value |
+-----+-----+
| id     | 561a48ea-0c1c-4152-8b7d-e4b4af276c2d |
| name   | myflavor |
| ram    | 3072 |
| swap   | 0 |
| vcpus  | 1 |
+-----+-----+
```

Эта команда отображает сведения о типе VM myflavor.

3.8.4 vinfra service compute flavor delete

Удаление типа VM:

```
usage: vinfra service compute flavor delete <flavor>
```

<flavor>

Идентификатор или имя типа VM

Пример:

```
# vinfra service compute flavor delete myflavor
Operation successful
```

Эта команда удаляет тип VM myflavor.

3.9 Управление SSH-ключами вычислительного кластера

3.9.1 vinfra service compute key create

Создание нового SSH-ключа вычислительного кластера:

```
usage: vinfra service compute key create --public-key <public-key>
      [--description <description>] <ssh-key>
```

---public-key <public-key>

Имя файла для загружаемого открытого ключа

--description <description>

Описание SSH-ключа

<ssh-key>

Имя SSH-ключа

Пример:

```
# vinfra service compute key create publickey --public-key /root/.ssh/id_rsa.pub \
--description 'public key'
+-----+-----+
| Field      | Value                               |
+-----+-----+
| created_at | 2019-04-25T13:41:14.241736+00:00    |
| description | public key                           |
| name       | publickey                            |
+-----+-----+
```

Эта команда создает открытый SSH-ключ publickey.

3.9.2 vinfra service compute key list

Вывод списка SSH-ключей вычислительного кластера

```
usage: vinfra service compute key list
```

Пример:

```
# vinfra service compute key list
+-----+-----+-----+
| name      | description | created_at |
+-----+-----+-----+
| testkey   | test key    | 2019-04-24T13:41:05.209837+00:00 |
| publickey | public key  | 2019-04-25T13:41:14.241736+00:00 |
+-----+-----+-----+
```

Эта команда выводит список всех SSH-ключей.

3.9.3 vinfra service compute key show

Отображение сведений о SSH-ключе вычислительного кластера:

```
usage: vinfra service compute key show <ssh-key>
```

<ssh-key>

Имя SSH-ключа

Пример:

```
# vinfra service compute key show publickey
+-----+-----+
| Field          | Value |
+-----+-----+
| created_at     | 2019-04-25T13:41:14.241736+00:00 |
| description    | public key |
| name           | publickey |
| public_key_fingerprint | 1a:fb:de:d8:1e:0a:84:30:fc:ff:e4:fd:89:e7:96:a9 |
+-----+-----+
```

Эта команда отображает сведения о SSH-ключе publickey.

3.9.4 `vinfra service compute key delete`

Удаление SSH-ключа вычислительного кластера:

```
usage: vinfra service compute key delete <ssh-key>
```

<ssh-key>

Имя SSH-ключа

Пример:

```
# vinfra service compute key delete publickey
Operation successful
```

Эта команда удаляет SSH-ключ `publickey`.

3.10 Управление виртуальными сетями

3.10.1 `vinfra service compute network create`

Создание вычислительной сети:

```
usage: vinfra service compute network create [--dhcp | --no-dhcp]
                                             [--dns-nameserver <dns-nameserver>]
                                             [--allocation-pool <allocation-pool>]
                                             [--gateway <gateway> | --no-gateway]
                                             [--ip-version <ip-version>]
                                             [--physical-network <physical-network>]
                                             [--cidr <cidr>] <network-name>
```

`--dhcp`

Включение DHCP.

`--no-dhcp`

Отключение DHCP.

`--dns-nameserver <dns-nameserver>`

IP-адрес сервера DNS. Этот параметр можно использовать несколько раз.

`--allocation-pool <allocation-pool>`

Пул IP-адресов для создания внутри сети в формате: `ip_addr_start-ip_addr_end`. Этот параметр можно использовать несколько раз.

--gateway <gateway>

IP-адрес шлюза

--no-gateway

Не настраивать шлюз для этой сети.

--ip-version <ip-version>

Версия IP-адресов сети

--physical-network <physical-network>

Физическая сеть для связи с внешней сетью

--cidr <cidr>

Маска подсети в нотации CIDR

<network-name>

Имя сети

Пример:

```
# vinfra service compute network create myprivnet --type vxlan \
--cidr 192.128.128.0/24 --gateway 192.128.128.1
```

```
+-----+-----+
| Field           | Value                                     |
+-----+-----+
| id              | 3848fb5d-bc98-4320-acd0-cde2df7c5bdd    |
| name            | myprivnet                               |
| physical_network|                                           |
| project_id      | 72a5db3a033c403a86756021e601ef34      |
| subnet          | allocation_pools:                       |
|                 | - end: 192.128.128.254                 |
|                 |   start: 192.128.128.2                 |
|                 | cidr: 192.128.128.0/24                 |
|                 | dns_nameservers: []                    |
|                 | enable_dhcp: true                       |
|                 | gateway_ip: 192.128.128.1              |
|                 | ip_version: 4                           |
| type            | vxlan                                    |
+-----+-----+
```

Эта команда создает частную сеть myprivnet с определенным методом CIDR и шлюзом.

3.10.2 vinfra service compute network list

Вывод списка вычислительных сетей:

```
usage: vinfra service compute network list
```

Пример:

```
# vinfra service compute network list -c id -c name -c cidr -c type -c allocation_pools
+-----+-----+-----+-----+-----+
| id          | name      | type  | cidr          | allocation_pools |
+-----+-----+-----+-----+-----+
| 1bf2c9da-... | private  | vxlan | 192.168.128.0/24 | - end: 192.168.128.254 |
|              |          |       |                 | start: 192.168.128.2 |
| 3848fb5d-... | myprivnet | vxlan | 192.128.128.0/24 | - end: 192.128.128.254 |
|              |          |       |                 | start: 192.128.128.2 |
| 417606ac-... | public   | flat  | 10.94.0.0/16   | - end: 10.94.129.79 |
|              |          |       |                 | start: 10.94.129.64 |
+-----+-----+-----+-----+-----+
```

Эта команда выводит список сетей, используемых в вычислительном кластере. (Список сокращен для экономии места на странице.)

3.10.3 vinfra service compute network show

Отображение сведений о вычислительной сети:

```
usage: vinfra service compute network show <network>
```

<network>

Идентификатор или имя сети

Пример:

```
# vinfra service compute network show 417606ac-1dbe-426a-844d-e047831ddce9
+-----+-----+
| Field          | Value |
+-----+-----+
| allocation_pools |       |
| cidr            |       |
| dns_nameservers |       |
| enable_dhcp     |       |
| gateway_ip      |       |
| id              | 417606ac-1dbe-426a-844d-e047831ddce9 |
| ip_version      |       |
| name            | public |
| physical_network | Public |
+-----+-----+
```

```
| project_id      | 72a5db3a033c403a86756021e601ef34 |
| type           | flat                               |
+-----+-----+
```

Эта команда отображает сведения о сети с идентификатором 417606ac-1dbe-426a-844d-e047831ddce9.

3.10.4 vinfra service compute network set

Изменение параметров вычислительной сети:

```
usage: vinfra service compute network set [--dhcp | --no-dhcp]
      [--dns-nameserver <dns-nameserver>]
      [--allocation-pool <allocation-pool>]
      [--gateway <gateway> | --no-gateway]
      [--name <name>] <network>
```

`--dhcp`

Включение DHCP.

`--no-dhcp`

Отключение DHCP.

`--dns-nameserver <dns-nameserver>`

IP-адрес сервера DNS. Этот параметр можно использовать несколько раз.

`--allocation-pool <allocation-pool>`

Пул IP-адресов для создания внутри сети в формате: ip_addr_start-ip_addr_end. Этот параметр можно использовать несколько раз.

`--gateway <gateway>`

IP-адрес шлюза

`--no-gateway`

Не настраивать шлюз для этой сети.

`--name <name>`

Новое имя для сети

`<network>`

Идентификатор или имя сети

Пример:

```
# vinfra service compute network set myprivnet --no-dhcp
+-----+-----+
| Field          | Value                                     |
+-----+-----+
| id             | 3848fb5d-bc98-4320-acd0-cde2df7c5bdd    |
| name           | myprivnet                                |
| physical_network |                                           |
| project_id     | 72a5db3a033c403a86756021e601ef34       |
| subnet         | allocation_pools:                        |
|                | - end: 192.128.128.254                  |
|                |   start: 192.128.128.2                  |
|                | cidr: 192.128.128.0/24                  |
|                | dns_nameservers: []                     |
|                | enable_dhcp: false                      |
|                | gateway_ip: 192.128.128.1              |
|                | ip_version: 4                           |
| type           | vxlan                                    |
+-----+-----+
```

Эта команда отключает DHCP для частной сети myprivnet.

3.10.5 vinfra service compute network delete

Удаление вычислительной сети:

```
usage: vinfra service compute network delete <network>
```

<network>

Идентификатор или имя сети

Пример:

```
# vinfra service compute network delete myprivnet
Operation successful
```

Эта команда удаляет частную сеть myprivnet.

3.11 Управление виртуальными маршрутизаторами

3.11.1 vinfra service compute router create

Создание виртуального маршрутизатора:


```
usage: vinfra service compute router create [--external-gateway <network>]
                                           [--enable-snat | --disable-snat]
                                           [--fixed-ip <fixed-ip>]
                                           [--internal-interface <network=network,
ip-addr=ip-addr>|<network>] <router-name>
```

`--external-gateway <network>`

Указывает внешнюю сеть для использования в качестве внешнего шлюза маршрутизатора (имя или идентификатор)

`--enable-snat`

Включение SNAT на внешнем шлюзе

`--disable-snat`

Отключение SNAT на внешнем шлюзе

`--fixed-ip <fixed-ip>`

Нужный IP-адрес на внешнем шлюзе

`--internal-interface <network=network, ip-addr=ip-addr>|<network>`

Указывает внутренний интерфейс. Этот параметр можно использовать несколько раз.

- `network`: имя частной виртуальной сети.
- `ip-addr`: неиспользуемый IP-адрес из выбранной частной сети для назначения интерфейсу; укажите, если шлюз по умолчанию для выбранной частной сети уже используется.

`<router-name>`

Имя виртуального маршрутизатора

Пример:

```
# vinfra service compute router create myrouter --external-gateway public \
--internal-interface private --enable-snat
+-----+-----+
| Field          | Value                                     |
+-----+-----+
| external_gateway_info | enable_snat: true                       |
|                   | ip_addresses:                            |
|                   | - 10.94.129.76                           |
|                   | network_id: 720e45bc-4225-49de-9346-26513d8d1262 |
| id              | b9d8b000-5d06-4768-9f65-2715250cda53     |
| name            | myrouter                                  |
| project_id      | 894696133031439f8aaa7e4868dcbd4d        |
| routes          | []                                        |
| status          | ACTIVE                                    |
+-----+-----+
```

Эта команда создает маршрутизатор `myrouter` между внешней сетью `public` и частной сетью `private` с включенным преобразованием SNAT на внешнем шлюзе.

3.11.2 `vinfra service compute router list`

Вывод списка виртуальных маршрутизаторов:

```
usage: vinfra service compute router list
```

Пример:

```
# vinfra service compute router list -c id -c external_gateway_info -c name -c status
+-----+-----+-----+-----+
| id          | external_gateway_info          | name      | status |
+-----+-----+-----+-----+
| b9d8b000-5d06-<...> | enable_snat: true              | myrouter  | ACTIVE |
|                | ip_addresses:                  |           |        |
|                | - 10.94.129.76                 |           |        |
|                | network_id: 720e45bc-4225-<...> |           |        |
+-----+-----+-----+-----+
```

Эта команда выводит список виртуальных маршрутизаторов, используемых в вычислительном кластере. (Список сокращен для экономии места на странице.)

3.11.3 `vinfra service compute router show`

Отображение информации о виртуальном маршрутизаторе:

```
usage: vinfra service compute router show <router>
```

<router>

Имя виртуального маршрутизатора

Пример:

```
# vinfra service compute router show myrouter
+-----+-----+
| Field          | Value                               |
+-----+-----+
| external_gateway_info | enable_snat: true                 |
|                  | ip_addresses:                     |
|                  | - 10.94.129.76                     |
|                  | network_id: 720e45bc-4225-49de-9346-26513d8d1262 |
| id             | b9d8b000-5d06-4768-9f65-2715250cda53 |
| name           | myrouter                           |
+-----+-----+
```

```
| project_id      | 894696133031439f8aaa7e4868dcbd4d |
| routes         | []                                 |
| status        | ACTIVE                             |
+-----+-----+
```

Эта команда отображает сведения о виртуальном маршрутизаторе `myrouter`.

3.11.4 `vinfra service compute router set`

Изменение параметров виртуального маршрутизатора:

```
usage: vinfra service compute router set [--name <name>] [--external-gateway <network> |
--no-external-gateway] [--fixed-ip <fixed-ip>]
[--enable-snat | --disable-snat]
[--route <destination=destination, nexthop=nexthop>
--no-route] <router>
```

`--name <name>`

Имя виртуального маршрутизатора

`--external-gateway <network>`

Указывает внешнюю сеть для использования в качестве внешнего шлюза маршрутизатора (имя или идентификатор)

`--no-external-gateway`

Удаление внешнего шлюза с маршрутизатора

`--enable-snat`

Включение SNAT на внешнем шлюзе

`--disable-snat`

Отключение SNAT на внешнем шлюзе

`--fixed-ip <fixed-ip>`

Нужный IP-адрес на внешнем шлюзе

`--route <destination=destination, nexthop=nexthop>`

Статический маршрут для маршрутизатора. Этот параметр можно использовать несколько раз.

- `destination`: целевая маска подсети в нотации CIDR.
- `nexthop`: IP-адрес следующего транзитного участка в одной из сетей, к которым подключен маршрутизатор.

```
--no-route
```

Очистка маршрутов, связанных с маршрутизатором

```
<router>
```

Имя или идентификатор виртуального маршрутизатора

Пример:

```
# vinfra service compute router set myrouter --disable-snat --external-gateway public
+-----+-----+
| Field          | Value                                     |
+-----+-----+
| external_gateway_info | enable_snat: false                       |
|                  | ip_addresses:                            |
|                  | - 10.94.129.76                          |
|                  | network_id: 720e45bc-4225-49de-9346-26513d8d1262 |
| id             | b9d8b000-5d06-4768-9f65-2715250cda53    |
| name           | myrouter                                 |
| project_id     | 894696133031439f8aaa7e4868dcbd4d       |
| routes         | []                                       |
| status         | ACTIVE                                  |
+-----+-----+
```

Эта команда отключает SNAT на внешнем шлюзе виртуального маршрутизатора myrouter.

3.11.5 vinfra service compute router iface add

Добавление интерфейса для виртуального маршрутизатора:

```
usage: vinfra service compute router iface add [--ip-address <ip-address>]
        --interface <network> router
```

```
--ip-address <ip-address>
```

IP-адрес

```
--interface <network>
```

Имя или идентификатор сети

```
router
```

Имя или идентификатор виртуального маршрутизатора

Пример:

```
# vinfra service compute router iface add myrouter --interface private2 \
--ip-address 192.168.30.3
```

```

| network_id | is_external | ip_addresses | status |
+-----+-----+-----+-----+
| 720e45bc-4225-49de-9346-26513d8d1262 | True | - 10.94.129.76 | ACTIVE |
| e6f146ce-a6d0-48b2-9e4f-64a128ce97ae | False | - 192.168.128.1 | ACTIVE |
| 86803e07-a6d7-4809-9566-1cbe4a89adfd | False | - 192.168.30.3 | DOWN |
+-----+-----+-----+-----+

```

Эта команда добавляет интерфейс из виртуальной сети `private2` на виртуальный маршрутизатор `myrouter` с IP-адресом `192.168.30.3`.

3.11.6 `vinfra service compute router iface list`

Вывод списка интерфейсов маршрутизатора:

```
usage: vinfra service compute router iface list router
```

`router`

Имя или идентификатор виртуального маршрутизатора

Пример:

```

# vinfra service compute router iface list myrouter
+-----+-----+-----+-----+
| network_id | is_external | ip_addresses | status |
+-----+-----+-----+-----+
| 720e45bc-4225-49de-9346-26513d8d1262 (public) | True | - 10.94.129.76 | ACTIVE |
| e6f146ce-a6d0-48b2-9e4f-64a128ce97ae (private) | False | - 192.168.128.1 | ACTIVE |
| 86803e07-a6d7-4809-9566-1cbe4a89adfd (private2) | False | - 192.168.30.3 | ACTIVE |
+-----+-----+-----+-----+

```

Эта команда выводит список интерфейсов виртуального маршрутизатора `myrouter`.

3.11.7 `vinfra service compute router iface remove`

Удаление интерфейса из виртуального маршрутизатора:

```
usage: vinfra service compute router iface remove --interface <network> router
```

`--interface <network>`

Имя или идентификатор сети

`router`

Имя или идентификатор виртуального маршрутизатора

Пример:

```
# vinfra service compute router iface remove myrouter --interface private2
+-----+-----+-----+-----+
| network_id          | is_external | ip_addresses        | status |
+-----+-----+-----+-----+
| 720e45bc-4225-49de-9346-26513d8d1262 | True        | - 10.94.129.76     | ACTIVE |
| e6f146ce-a6d0-48b2-9e4f-64a128ce97ae | False       | - 192.168.128.1   | ACTIVE |
+-----+-----+-----+-----+
```

Эта команда удаляет интерфейс виртуальной сети `private2` из виртуального маршрутизатора `myrouter`.

3.11.8 `vinfra service compute router delete`

Удаление виртуального маршрутизатора:

```
usage: vinfra service compute router delete <router>
```

<router>

Идентификатор или имя виртуального маршрутизатора

Пример:

```
# vinfra service compute router delete myrouter
Operation successful
```

Эта команда удаляет виртуальный маршрутизатор `myrouter`.

3.12 Управление плавающими IP-адресами

3.12.1 `vinfra service compute floatingip create`

Создание плавающего IP-адреса:

```
usage: vinfra service compute floatingip create [--floating-ip-address <floating-ip-address>
                                                [--port-id <port-id>]
                                                [--fixed-ip-address <fixed-ip-address>]
                                                [--description description] --network <netw
```

`--floating-ip-address <floating-ip-address>`

Плавающий IP-адрес

`--port-id <port-id>`

Идентификатор порта, который будет связан с плавающим IP-адресом. Чтобы узнать идентификатор порта выбранного сервера, используйте команду `vinfra service compute server iface list` (страница 64).

`--fixed-ip-address <fixed-ip-address>`

IP-адрес порта (требуется, только если у порта несколько IP-адресов)

`--description <description>`

Описание плавающего IP-адреса

`--network <network>`

Идентификатор или имя сети, из которой будет выделен плавающий IP-адрес

Пример:

```
# vinfra service compute floatingip create 720e45bc-4225-49de-9346-26513d8d1262 \
--port-id 418c8c9e-aaa5-42f2-8da7-24bfead6f28b --fixed-ip-address 192.168.128.5
+-----+-----+
| Field          | Value                                     |
+-----+-----+
| attached_to    | a172cb6a-1c7b-4157-9e86-035f3077646f    |
| description    |                                           |
| fixed_ip_address | 192.168.128.5                            |
| floating_ip_address | 10.94.129.72                             |
| floating_network_id | 720e45bc-4225-49de-9346-26513d8d1262    |
| id             | a709f884-c43f-4a9a-a243-a340d7682ef8    |
| port_id        | 418c8c9e-aaa5-42f2-8da7-24bfead6f28b    |
| project_id     | 894696133031439f8aaa7e4868dcbd4d        |
| router_id      | f7f86029-a553-4d61-b7ec-6f581d9c5f5f    |
| status         | DOWN                                     |
+-----+-----+
```

Эта команда создает плавающий IP-адрес из внешней сети с идентификатором 720e45bc-4225-49de-9346-26513d8d1262 и назначает его серверу на порт с идентификатором 418c8c9e-aaa5-42f2-8da7-24bfead6f28b и частным IP-адресом 192.168.128.5.

3.12.2 vinfra service compute floatingip list

Вывод списка плавающих IP-адресов:

```
usage: vinfra service compute floatingip list
```

Пример:

```
# vinfra service compute floatingip list -c id -c fixed_ip_address -c port_id \
-c floating_ip_address -c floating_network_id
+-----+-----+-----+-----+
| id          | fixed_ip_address | port_id          | floating_ip_address | floating_netwo
+-----+-----+-----+-----+
| a709f884-... | 192.168.128.5   | 418c8c9e-...   | 10.94.129.72       | 720e45bc-...>
```

Эта команда выводит список плавающих IP-адресов, используемых в вычислительном кластере. (Список сокращен для экономии места на странице.)

3.12.3 vinfra service compute floatingip show

Отображение информации о плавающем IP-адресе:

```
usage: vinfra service compute floatingip show <floatingip>
```

<floatingip>

Идентификатор плавающего IP-адреса

Пример:

```
# vinfra service compute floatingip show a709f884-c43f-4a9a-a243-a340d7682ef8
+-----+-----+
| Field          | Value |
+-----+-----+
| attached_to    | a172cb6a-1c7b-4157-9e86-035f3077646f |
| description    | |
| fixed_ip_address | 192.168.128.5 |
| floating_ip_address | 10.94.129.72 |
| floating_network_id | 720e45bc-4225-49de-9346-26513d8d1262 |
| id             | a709f884-c43f-4a9a-a243-a340d7682ef8 |
| port_id        | 418c8c9e-aaa5-42f2-8da7-24bfead6f28b |
| project_id     | 894696133031439f8aaa7e4868dcbd4d |
| router_id      | f7f86029-a553-4d61-b7ec-6f581d9c5f5f |
| status         | ACTIVE |
+-----+-----+
```

Эта команда отображает сведения о плавающем IP-адресе с идентификатором a709f884-c43f-4a9a-a243-a340d7682ef8.

3.12.4 vinfra service compute floatingip set

Изменение параметров плавающего IP-адреса:

```
usage: vinfra service compute floatingip set [--port-id <port-id>]
                                           [--fixed-ip-address <fixed-ip-address>]
                                           [--description <description>] <floatingip>
```

--port-id <port-id>

Идентификатор порта, который будет связан с плавающим IP-адресом

--fixed-ip-address <fixed-ip-address>

IP-адрес порта (требуется, только если у порта несколько IP-адресов)

--description <description>

Описание плавающего IP-адреса

<floatingip>

Идентификатор плавающего IP-адреса

Пример:

```
# vinfra service compute floatingip set a709f884-c43f-4a9a-a243-a340d7682ef8 \
--description "Floating IP for myvm"
+-----+-----+
| Field          | Value                                     |
+-----+-----+
| attached_to    | a172cb6a-1c7b-4157-9e86-035f3077646f    |
| description    | Floating IP for myvm                    |
| fixed_ip_address | 192.168.128.5                            |
| floating_ip_address | 10.94.129.72                             |
| floating_network_id | 720e45bc-4225-49de-9346-26513d8d1262    |
| id             | a709f884-c43f-4a9a-a243-a340d7682ef8    |
| port_id        | 418c8c9e-aaa5-42f2-8da7-24bfead6f28b    |
| project_id     | 894696133031439f8aaa7e4868dcbd4d        |
| router_id      | f7f86029-a553-4d61-b7ec-6f581d9c5f5f    |
| status         | ACTIVE                                   |
+-----+-----+
```

Эта команда добавляет описание для плавающего IP-адреса с идентификатором a709f884-c43f-4a9a-a243-a340d7682ef8.

3.12.5 `vinfra service compute floatingip delete`

Удаление плавающего IP-адреса:

```
usage: vinfra service compute floatingip delete <floatingip>
```

<floatingip>

Идентификатор плавающего IP-адреса

Пример:

```
# vinfra service compute floatingip delete a709f884-c43f-4a9a-a243-a340d7682ef8
Operation successful
```

Эта команда удаляет плавающий IP-адрес с идентификатором `a709f884-c43f-4a9a-a243-a340d7682ef8`.

3.13 Управление балансировщиками нагрузки

3.13.1 `vinfra service compute load-balancer create`

Создание балансировщика нагрузки:

```
usage: vinfra service compute load-balancer create [--description <description>]
          [--enable | --disable]
          [--address <address>]
          [--floating-ip <floating-ip>]
          [--pools-config <pools>]
          <name> <network>
```

`--description <description>`

Описание балансировщика нагрузки

`--enable`

Включение балансировщика нагрузки.

`--disable`

Отключение балансировщика нагрузки.

`--address <address>`

IP-адрес, который балансировщик нагрузки попытается взять в сети.

--floating-ip <floating-ip>

Плавающий IP-адрес, который будет использоваться для подключения к балансировщику нагрузки из внешних сетей.

--pools-config <pools>

Файл конфигурации пулов

Ниже приведен пример файла конфигурации пулов в формате YAML.

```
- backend_protocol: HTTPS
  backend_protocol_port: 443
  healthmonitor: {delay: 5, max_retries: 3, max_retries_down: 3, timeout: 5, type: PING,
    url_path: /}
  lb_algorithm: ROUND_ROBIN
  members:
  - {address: 192.168.30.49, compute_server_id: 5fd82e2a-3fef-4171-bfa4-67daa99ae64f}
  - {address: 192.168.30.15, compute_server_id: 4f7262d0-2031-4772-9984-2f1066ac166b}
  name: pool1
  protocol: HTTPS
  protocol_port: 443
  sticky_session: False
```

<name>

Имя балансировщика нагрузки

<network>

Идентификатор или имя сети, в которой будет работать балансировщик нагрузки.

Пример:

```
# vinfra service compute load-balancer create mylbaas private1 --floating-ip 10.94.129.70
+-----+-----+
| Field      | Value                                     |
+-----+-----+
| address    | 192.168.30.230                           |
| amphorae   |                                           |
| created_at | 2019-11-18T12:59:08.243413                |
| description |                                           |
| enabled    | True                                       |
| floating_ip | 10.94.129.70                              |
| ha_enabled  |                                           |
| id         | 941bf637-2d55-40f0-92c0-e65d6567b468     |
| members_count | 0                                         |
| name       | mylbaas                                    |
| network_id | 2b821d00-e428-4a76-b1ae-d181c9f5ae7f     |
| pools     | []                                         |
| port_id   | 2d8ab88a-847c-4396-857e-11eaa80e1b24     |
| project_id | e4e059c67dee4736851df14d4519a5a5        |
| status    | CREATING                                  |
+-----+-----+
```

```
| updated_at | |
+-----+-----+
```

Эта команда создает балансировщик нагрузки mylbaas без пулов балансировки, который будет работать в сети private с плавающим IP-адресом 10.94.129.70.

3.13.2 vinfra service compute load-balancer list

Вывод списка балансировщиков нагрузки:

```
usage: vinfra service compute load-balancer list
```

Пример:

```
# vinfra service compute load-balancer list
+-----+-----+
| id | name |
+-----+-----+
| 941bf637-2d55-40f0-92c0-e65d6567b468 | mylbaas |
+-----+-----+
```

Эта команда выводит список балансировщиков нагрузки в вычислительном кластере.

3.13.3 vinfra service compute load-balancer show

Отображение сведений о балансировщике нагрузки:

```
usage: vinfra service compute load-balancer show <load-balancer>
```

<load-balancer>

Идентификатор или имя балансировщика нагрузки

Пример:

```
# vinfra service compute load-balancer show mylbaas
+-----+-----+
| Field | Value |
+-----+-----+
| address | 192.168.30.230 |
| amphora | - active: true |
| | compute_id: b0c4793f-e1b1-4251-91c2-94e34787f537 |
| | created_at: '2019-11-18T12:59:12.742446' |
| | id: b7b23106-a87b-412d-9ce6-7c69b5594342 |
| | image_id: 6d1ba6f9-cf86-4ea4-a32d-f138868a9742 |
| | role: STANDALONE |
+-----+-----+
```

```

|          | status: ALLOCATED |
|          | updated_at: '2019-11-18T13:01:07.601184' |
| created_at | 2019-11-18T12:59:08.243413 |
| description | |
| enabled | True |
| floating_ip | 10.94.129.70 |
| ha_enabled | False |
| id | 941bf637-2d55-40f0-92c0-e65d6567b468 |
| members_count | 0 |
| name | mylbaas |
| network_id | 2b821d00-e428-4a76-b1ae-d181c9f5ae7f |
| pools | [] |
| port_id | 2d8ab88a-847c-4396-857e-11eaa80e1b24 |
| project_id | e4e059c67dee4736851df14d4519a5a5 |
| status | ACTIVE |
| updated_at | 2019-11-18T13:01:10.983144 |
+-----+-----+

```

Эта команда отображает сведения о балансировщике нагрузки mylbaas.

3.13.4 vinfra service compute load-balancer stats

Отображение статистики для балансировщика нагрузки:

```
usage: vinfra service compute load-balancer stats <load-balancer>
```

<load-balancer>

Идентификатор или имя балансировщика нагрузки

Пример:

```

# vinfra service compute load-balancer stats mylbaas
+-----+-----+
| Field | Value |
+-----+-----+
| stats | active_connections: 0 |
|       | bytes_in: 0 |
|       | bytes_out: 0 |
|       | listeners: null |
|       | loadbalancer_id: 17cfa86f-c374-4ca3-8cd6-f638a5234fe7 |
|       | request_errors: 0 |
|       | total_connections: 0 |
+-----+-----+

```

Эта команда отображает статистику для балансировщика нагрузки mylbaas.

3.13.5 vinfra service compute load-balancer set

Изменение балансировщика нагрузки:

```
usage: vinfra service compute load-balancer set [--description <description>]
        [--enable | --disable] [--name <name>]
        <load-balancer>
```

`--description <description>`

Описание балансировщика нагрузки

`--enable`

Включение балансировщика нагрузки.

`--disable`

Отключение балансировщика нагрузки.

`--name <name>`

Имя балансировщика нагрузки

`<load-balancer>`

Идентификатор или имя балансировщика нагрузки

Пример:

```
# vinfra service compute load-balancer set mylbaas --disable \
--description "Disabled load balancer"
+-----+
| Field      | Value                                     |
+-----+
| address    | 192.168.30.230                           |
| amphorae   |                                           |
| created_at | 2019-11-18T12:59:08.243413                |
| description | Disabled load balancer                   |
| enabled    | False                                     |
| floating_ip |                                           |
| ha_enabled |                                           |
| id         | 941bf637-2d55-40f0-92c0-e65d6567b468     |
| members_count | 0                                         |
| name       | mylbaas                                   |
| network_id | 2b821d00-e428-4a76-b1ae-d181c9f5ae7f     |
| pools      | []                                        |
| port_id    | 2d8ab88a-847c-4396-857e-11eaa80e1b24     |
| project_id | e4e059c67dee4736851df14d4519a5a5        |
| status     | DISABLED                                  |
| updated_at | 2019-11-18T13:09:09.151442                |
+-----+
```

Эта команда отключает балансировщик нагрузки `mu1baas` и добавляет к нему описание.

3.13.6 `vinfra service compute load-balancer pool create`

Создание пула балансировщика нагрузки:

```
usage: vinfra service compute load-balancer pool create --name <name>
                                                --protocol {HTTP,HTTPS}
                                                --port <port>
                                                --algorithm <algorithm>
                                                --backend-protocol {HTTP,HTTPS}
                                                --backend-port <backend_port>
                                                [--certificate-file <cert_file>]
                                                [--connection-limit <limit>]
                                                [--description <description>]
                                                [--healthmonitor type=<type>,
url_path=<url>[,key=value,...]]
                                                [--member address=<ip>
[,key=value,...]]
                                                [--privatekey-file <key>]
                                                [--enable-sticky-session |
--disable-sticky-session]
                                                [--enable | --disable]
                                                <load-balancer>
```

`--name <name>`

Имя пула

`--protocol {HTTP,HTTPS}`

Протокол для входящих подключений

`--port <port>`

Порт для входящих подключений

`--algorithm <algorithm>`

Алгоритм балансировки нагрузки (LEAST_CONNECTIONS, ROUND_ROBIN или SOURCE_IP)

`--backend-protocol {HTTP,HTTPS}`

Протокол для целевых подключений

`--backend-port <backend_port>`

Порт для целевых подключений

`--certificate-file <cert_file>`

Файл сертификата x.509 в формате PEM. Требуется для TLS-терминированных балансировщиков нагрузки HTTPS->HTTP.

`--connection-limit <limit>`

Максимально разрешенное количество подключений для этого пула. Значение по умолчанию: -1 (неограниченные подключения).

`--description <description>`

Описание пула

`--healthmonitor type=<type>,url_path=<url>[,key=value,...]`

Параметры монитора состояния:

- `type`: тип монитора состояния (HTTP, HTTPS, PING или TCP)
- `url_path`: URL-путь к монитору состояния
- разделенные запятыми пары `key=value` с ключами (необязательно):
 - `delay`: время в секундах между отправками запросов участникам.
 - `enabled`: указывает, включен монитор состояния или нет (`true` или `false`).
 - `max_retries`: количество успешных проверок, необходимых для смены статуса участника на HEALTHY. Значение в диапазоне от 1 до 10.
 - `max_retries_down`: количество неуспешных проверок, необходимых для смены статуса участника на UNHEALTHY. Значение в диапазоне от 1 до 10.
 - `timeout`: максимальное время в секундах, в течение которого монитор ожидает подключения. Это значение должно быть меньше значения `delay`.

`--member address=<ip>[,key=value,...]`

Параметры участников:

- `address`: адрес IPv4 вычислительного сервера
- разделенные запятыми пары `key=value` с ключами (необязательно):
 - `enabled`: указывает, включен участник или нет (`true` или `false`).
 - `weight`: определяет долю подключений, которые обслуживает участник по сравнению с другими участниками пула. Например, `weight 10` означает, что этот участник обрабатывает в пять раз больше подключений, чем участник с `weight 2`. Значение 0 показывает, что этот участник не получает новых подключений, но продолжает обслуживать существующие. Значение может быть в диапазоне от 0 до 256. Значение по умолчанию: 1. Этот параметр можно использовать несколько раз.

`--privatekey-file <key>`

Закрытый TLS-ключ в формате PEM. Требуется для TLS-терминированных балансировщиков нагрузки HTTPS->HTTP.

`--enable-sticky-session`

Включить сохранение сеанса.

`--disable-sticky-session`

Отключить сохранение сеанса.

`--enable`

Включить пул.

`--disable`

Отключить пул.

`<load-balancer>`

Идентификатор или имя балансировщика нагрузки

Пример:

```
# vinfra service compute load-balancer pool create mylbaas --name mypool --protocol HTTP \
--port 80 --backend-protocol HTTP --backend-port 80 --algorithm LEAST_CONNECTIONS \
--member address=192.168.31.153, compute_server_id=d51c10a7-6187-4a5a-a838-de5fc78a688a \
--member address=192.168.31.22, compute_server_id=54603109-8963-49f2-8c49-332537c57e90 \
--enable-sticky-session
+-----+-----+
| Field          | Value                                |
+-----+-----+
| backend_protocol | HTTP                                 |
| backend_protocol_port | 80                                   |
| certificate      |                                       |
| connection_limit | -1                                   |
| created_at      | 2019-11-18T13:11:27.982129          |
| description     |                                       |
| enabled         | True                                 |
| healthmonitor   |                                       |
| id              | fa40e282-b29a-465a-afaa-2c702d2bde17 |
| lb_algorithm    | LEAST_CONNECTIONS                  |
| listener_id     | 66cc714e-af7f-40eb-9db8-67b8b6b6d23c |
| loadbalancer_id | 941bf637-2d55-40f0-92c0-e65d6567b468 |
| members         | []                                   |
| name            | mypool                              |
| private_key     |                                       |
| project_id      | e4e059c67dee4736851df14d4519a5a5   |
| protocol        | HTTP                                 |
| protocol_port   | 80                                   |
| status          | CREATING                            |
+-----+-----+
```

```
| sticky_session      | True      |
| updated_at         |           |
+-----+-----+
```

Эта команда добавляет пул балансировки `mypool` для балансировщика нагрузки `mylbaas` со следующими параметрами:

- правило перенаправления «HTTP на порту 80 -> HTTP на порту 80»
- алгоритм балансировки `LEAST_CONNECTIONS`
- два участника в пуле
- сохранение сеанса включено

3.13.7 `vinfra service compute load-balancer pool list`

Вывод списка пулов балансировщика нагрузки:

```
usage: vinfra service compute load-balancer pool list
```

Пример:

```
# vinfra service compute load-balancer pool list
+-----+-----+-----+
| id              | name      | loadbalancer_id |
+-----+-----+-----+
| fa40e282-b29a-465a-afaa-2c702d2bde17 | mypool   | 941bf637-2d55-40f0-92c0-e65d6567b468 |
+-----+-----+-----+
```

Эта команда выводит список пулов балансировщиков нагрузки в вычислительном кластере.

3.13.8 `vinfra service compute load-balancer pool show`

Отображение сведений о пуле балансировщика нагрузки:

```
usage: vinfra service compute load-balancer pool show <pool>
```

<pool>

Идентификатор или имя пула балансировщика нагрузки

Пример:

```
# vinfra service compute load-balancer pool show mypool
+-----+-----+
```

Field	Value
backend_protocol	HTTP
backend_protocol_port	80
certificate	
connection_limit	-1
created_at	2019-11-18T13:11:27.982129
description	
enabled	True
healthmonitor	
id	fa40e282-b29a-465a-afaa-2c702d2bde17
lb_algorithm	LEAST_CONNECTIONS
listener_id	66cc714e-af7f-40eb-9db8-67b8b6b6d23c
loadbalancer_id	941bf637-2d55-40f0-92c0-e65d6567b468
members	- address: 192.168.31.153 compute_server_id: d51c10a7-6187-4a5a-a838-de5fc78a688a created_at: '2019-11-18T13:11:59.681101' enabled: true id: 3fd5dcc5-6e2c-4e22-8d0a-8e94e20a122f name: '' pool_id: null status: HEALTHY updated_at: '2019-11-18T13:12:01.467306' weight: 1 - address: 192.168.31.22 compute_server_id: 54603109-8963-49f2-8c49-332537c57e90 created_at: '2019-11-18T13:12:10.176853' enabled: true id: ccb645b3-63c7-44f8-b861-b197c85506d4 name: '' pool_id: null status: HEALTHY updated_at: '2019-11-18T13:12:12.281578' weight: 1
name	mypool
private_key	
project_id	e4e059c67dee4736851df14d4519a5a5
protocol	HTTP
protocol_port	80
status	ACTIVE
sticky_session	True
updated_at	2019-11-18T13:12:12.305509

Эта команда отображает сведения о пуле балансировщика нагрузки mypool.

3.13.9 vinfra service compute load-balancer pool set

Изменение пула балансировщика нагрузки:

```
usage: vinfra service compute load-balancer pool set --name <name> --protocol {HTTP,HTTPS}
--port <port> --algorithm <algorithm>
--backend-protocol {HTTP,HTTPS}
--backend-port <backend_port>
[--certificate-file <cert_file>]
[--connection-limit <limit>]
[--description <description>]
[--healthmonitor type=<type>,
url_path=<url>[,key=value,...]]
[--member address=<ip>,
compute_server_id=<id>
[,key=value,...]]
[--privatekey-file <key>]
[--enable-sticky-session |
--disable-sticky-session]
[--enable | --disable] <pool>
```

--name <name>

Имя пула

--protocol {HTTP,HTTPS}

Протокол для входящих подключений

--port <port>

Порт для входящих подключений

--algorithm <algorithm>

Алгоритм балансировки нагрузки (LEAST_CONNECTIONS, ROUND_ROBIN или SOURCE_IP)

--backend-protocol {HTTP,HTTPS}

Протокол для целевых подключений

--backend-port <backend_port>

Порт для целевых подключений

--certificate-file <cert_file>

Файл сертификата x.509 в формате PEM. Требуется для TLS-терминированных балансировщиков нагрузки HTTPS->HTTP.

--connection-limit <limit>

Максимально разрешенное количество подключений для этого пула. Значение по умолчанию: -1 (неограниченные подключения).

`--description <description>`

Описание пула

`--healthmonitor type=<type>,url_path=<url>[,key=value,...]`

Параметры монитора состояния:

- `type`: тип монитора состояния (HTTP, HTTPS, PING или TCP)
- `url_path`: URL-путь к монитору состояния
- разделенные запятыми пары `key=value` с ключами (необязательно):
 - `delay`: время в секундах между отправками запросов участникам.
 - `enabled`: указывает, включен монитор состояния или нет (`true` или `false`).
 - `max_retries`: количество успешных проверок, необходимых для смены статуса участника на HEALTHY. Значение в диапазоне от 1 до 10.
 - `max_retries_down`: количество неуспешных проверок, необходимых для смены статуса участника на UNHEALTHY. Значение в диапазоне от 1 до 10.
 - `timeout`: максимальное время в секундах, в течение которого монитор ожидает подключения. Это значение должно быть меньше значения `delay`.

`--member address=<ip>,compute_server_id=<id>[,key=value,...]`

Параметры участников:

- `address`: адрес IPv4 вычислительного сервера
- `compute_server_id`: UUID вычислительного сервера
- разделенные запятыми пары `key=value` с ключами (необязательно):
 - `enabled`: указывает, включен участник или нет. Может иметь значение `true` или `false`.
 - `weight`: определяет долю подключений, которые обслуживает участник по сравнению с другими участниками пула. Например, `weight 10` означает, что этот участник обрабатывает в пять раз больше подключений, чем участник с `weight 2`. Значение 0 показывает, что этот участник не получает новых подключений, но продолжает обслуживать существующие. Значение может быть в диапазоне от 0 до 256. Значение по умолчанию: 1. Этот параметр можно использовать несколько раз.

`--privatekey-file <key>`

Закрытый TLS-ключ в формате PEM. Требуется для TLS-терминированных балансировщиков нагрузки HTTPS->HTTP.

`--enable-sticky-session`

Включить сохранение сеанса.

`--disable-sticky-session`

Отключить сохранение сеанса.

`--enable`

Включить пул.

`--disable`

Отключить пул.

`<pool>`

Идентификатор или имя пула балансировщика нагрузки

Пример:

```
# vinfra service compute load-balancer pool set mypool --algorithm ROUND_ROBIN \
--member address=192.168.31.153,compute_server_id=d51c10a7-6187-4a5a-a838-de5fc78a688a \
--member address=192.168.31.22,compute_server_id=54603109-8963-49f2-8c49-332537c57e90 \
--member address=192.168.31.51,compute_server_id=5eb49a67-f0ff-44ec-b4e6-105e16e2db07 \
--disable-sticky-session
Operation accepted.
```

Эта команда изменяет параметры пула балансировки `mypool` следующим образом:

- устанавливает алгоритм балансировки `ROUND_ROBIN`
- добавляет третьего участника в пул
- отключает сохранение сеанса

3.13.10 `vinfra service compute load-balancer pool delete`

Удаление пула балансировщика нагрузки:

```
usage: vinfra service compute load-balancer pool delete <pool>
```

`<pool>`

Идентификатор или имя пула балансировщика нагрузки

Пример:

```
# vinfra service compute load-balancer pool delete mypool
Operation successful.
```

Эта команда удаляет пул балансировщика нагрузки mypool.

3.13.11 vinfra service compute load-balancer delete

Удаление балансировщика нагрузки:

```
usage: vinfra service compute load-balancer delete <load-balancer>
```

<load-balancer>

Идентификатор или имя балансировщика нагрузки

Пример:

```
# vinfra service compute load-balancer delete mylbaas
Operation accepted.
```

Эта команда удаляет балансировщик нагрузки mylbaas.

3.14 Управление томами

3.14.1 vinfra service compute volume create

Создание нового вычислительного тома:

```
usage: vinfra service compute volume create [--description <description>]
                                           [--network-install <network_install>]
                                           [--image <image>] [--snapshot <snapshot>]
                                           --storage-policy <storage_policy>
                                           --size <size-gb> <volume-name>
```

--description <description>

Описание тома

--network-install <сетевая_установка>

Выполнение установки по сети (true — да или false — нет).

--image <образ>

Идентификатор или имя исходного образа вычислений

--snapshot <снимок>

Идентификатор или имя исходного снимка образа вычислений

--storage-policy <политика_хранилища>

Идентификатор или имя политики хранилища

--size <размер_в_ГБ>

Размер тома в гигабайтах

<имя_тома>

Имя тома

Пример:

```
# vinfra service compute volume create myvolume --storage-policy default --size 8
+-----+-----+
| Field          | Value                                     |
+-----+-----+
| attachments    | []                                        |
| availability_zone | nova                                     |
| bootable       | False                                    |
| consistencygroup_id |                                           |
| created_at     | 2018-09-12T12:30:12.665916              |
| description    |                                           |
| encrypted      | False                                    |
| id             | c9c0e9e7-ce7a-4566-99d5-d7e40f2987ab   |
| imageRef       |                                           |
| migration_status |                                           |
| multiattach    | False                                    |
| name           | myvolume                                 |
| network_install | False                                    |
| os-vol-host-attr:host |                                           |
| os-vol-mig-status-attr:migstat |                                           |
| os-vol-mig-status-attr:name_id |                                           |
| project_id     | 72a5db3a033c403a86756021e601ef34       |
| replication_status |                                           |
| size           | 8                                        |
| snapshot_id    |                                           |
| source_volid   |                                           |
| status         | creating                                 |
| storage_policy_name | default                                  |
| updated_at     |                                           |
| user_id        | 98bf389983c24c07af9677b931783143     |
| volume_image_metadata |                                           |
+-----+-----+
```

Эта команда создает том myvolume размером 8 ГБ и задает для него политику хранилища по умолчанию.

3.14.2 vinfra service compute volume list

Список вычислительных томов:

```
usage: vinfra service compute volume list
```

Пример:

```
# vinfra service compute volume list -c id -c name -c size -c status
+-----+-----+-----+-----+
| id           | name       | size | status  |
+-----+-----+-----+-----+
| c9c0e9e7-ce7a-4566-99d5-d7e40f2987ab | myvolume  | 8    | available |
+-----+-----+-----+-----+
```

Эта команда выводит список томов, доступных для вычислительного кластера. (Выходные данные усекаются, чтобы они поместились на странице.)

3.14.3 vinfra service compute volume show

Вывод подробных сведений о вычислительном томе:

```
usage: vinfra service compute volume show <volume>
```

<ТОМ>

Идентификатор или имя тома

Пример:

```
# vinfra service compute volume show myvolume
+-----+-----+
| Field           | Value                                     |
+-----+-----+
| attachments     | []                                       |
| availability_zone | nova                                    |
| bootable        | False                                   |
| consistencygroup_id |                                         |
| created_at      | 2018-09-12T12:30:12.665916             |
| description     |                                         |
| encrypted       | False                                   |
| id              | c9c0e9e7-ce7a-4566-99d5-d7e40f2987ab  |
| imageRef        |                                         |
| migration_status |                                         |
| multiattach     | False                                   |
| name            | myvolume                               |
| network_install | False                                   |
| os-vol-host-attr:host | stor-1.example.com.vstoragedomain@vstorage#vstorage |
+-----+-----+
```

```

| os-vol-mig-status-attr:migstat |
| os-vol-mig-status-attr:name_id |
| project_id | 72a5db3a033c403a86756021e601ef34
| replication_status |
| size | 8
| snapshot_id |
| source_volid |
| status | available
| storage_policy_name | default
| updated_at | 2018-09-12T12:30:33.167654
| user_id | 98bf389983c24c07af9677b931783143
| volume_image_metadata |
+-----+

```

Эта команда отображает подробные данные о томе myvolume.

3.14.4 vinfra service compute volume set

Изменение параметров тома:

```

usage: vinfra service compute volume set [--description <description>]
                                          [--network-install <network_install>]
                                          [--storage-policy <storage_policy>]
                                          [--bootable <bootable>]
                                          [--name <name>] <volume>

```

`--description <description>`

Описание тома

`--network-install <сетевая_установка>`

Выполнение сетевой установки (true — да или false — нет)

`--storage-policy <политика_хранилища>`

Идентификатор или имя политики хранилища

`--bootable <загрузочный>`

Сделать том загрузочным (true — да или false — нет)

`--name <name>`

Новое имя для тома

`<том>`

Идентификатор или имя тома

Пример:

```
# vinfra service compute volume set myvolume --storage-policy mystorpolicy
+-----+-----+
| Field          | Value                                     |
+-----+-----+
| attachments    | []                                       |
| availability_zone | nova                                    |
| bootable       | False                                  |
| consistencygroup_id |                                         |
| created_at     | 2018-09-12T12:30:12.665916            |
| description    |                                         |
| encrypted      | False                                  |
| id             | c9c0e9e7-ce7a-4566-99d5-d7e40f2987ab  |
| imageRef       |                                         |
| migration_status |                                         |
| multiattach    | False                                  |
| name           | myvolume                               |
| network_install | False                                  |
| os-vol-host-attr:host | stor-1.example.com.vstoragedomain@vstorage#vstorage |
| os-vol-mig-status-attr:migstat |                                         |
| os-vol-mig-status-attr:name_id |                                         |
| project_id     | 72a5db3a033c403a86756021e601ef34     |
| replication_status |                                         |
| size           | 8                                       |
| snapshot_id    |                                         |
| source_volid   |                                         |
| status         | available                               |
| storage_policy_name | mystorpolicy                           |
| updated_at     | 2018-09-12T12:55:29.298717            |
| user_id        | 98bf389983c24c07af9677b931783143     |
| volume_image_metadata |                                         |
+-----+-----+
```

Эта команда изменяет политику хранилища для тома myvolume на политику mystorpolicy.

3.14.5 vinfra service compute volume extend

Увеличение размера вычислительного тома:

```
usage: vinfra service compute volume extend --size <size_gb> <volume>
```

<том>

Идентификатор или имя тома

Пример:

```
# vinfra service compute volume extend myvolume --size 16
Operation successful
```

Эта команда увеличивает размер тома myvolume до 16 ГБ.

3.14.6 vinfra service compute volume delete

Удаление вычислительного тома:

```
usage: vinfra service compute volume delete <volume>
```

<том>

Идентификатор или имя тома

Пример:

```
# vinfra service compute volume delete myvolume2
Operation successful
```

Эта команда удаляет том myvolume2.

3.15 Управление моментальными снимками ТОМОВ

3.15.1 vinfra service compute volume snapshot create

Создание снимка тома:

```
usage: vinfra service compute volume snapshot create [--description <description>]
           --volume <volume>
           <volume-snapshot-name>
```

--description <description>

Описание снимка тома

--volume <том>

Идентификатор или имя тома

<имя_снимка_тома>

Имя снимка тома

Пример:

```
# vinfra service compute volume snapshot create mysnapshot --volume myvolume
+-----+-----+
| Field      | Value                                |
+-----+-----+
```

```

| created_at | 2019-04-30T13:12:54.297629+00:00 |
| description | |
| id | 3fdfe5d6-8bd2-4bf5-8599-a9cef50e5b71 |
| metadata | {} |
| name | mysnapshot |
| project_id | fd0ae61496d04ef6bb637bc3167b7eaf |
| size | 8 |
| status | creating |
| volume_id | 92dc3bd7-713d-42bf-83cd-4de40c24fed9 |
+-----+-----+

```

Эта команда инициирует создание снимка mysnapshot тома myvolume.

3.15.2 vinfra service compute volume snapshot list

Вывод списка снимков томов:

```
usage: vinfra service compute volume snapshot list
```

Пример:

```

# vinfra service compute volume snapshot list -c id -c name -c size -c status
+-----+-----+-----+-----+
| id | name | status |
+-----+-----+-----+-----+
| 3fdfe5d6-8bd2-4bf5-8599-a9cef50e5b71 | mysnapshot | available |
+-----+-----+-----+-----+

```

Эта команда выводит список снимков томов, доступных для вычислительного кластера. (Выходные данные усекаются, чтобы они поместились на странице.)

3.15.3 vinfra service compute volume snapshot show

Отображение подробных данных о снимке тома:

```
usage: vinfra service compute volume snapshot show <volume-snapshot>
```

<снимок_тома>

Идентификатор или имя снимка тома

Пример:

```

# vinfra service compute volume snapshot show mysnapshot
+-----+-----+
| Field | Value |
+-----+-----+

```

```

| created_at | 2019-04-30T13:12:54.297629+00:00 |
| description | |
| id | 3fdfe5d6-8bd2-4bf5-8599-a9cef50e5b71 |
| metadata | {} |
| name | mysnapshot |
| project_id | fd0ae61496d04ef6bb637bc3167b7eaf |
| size | 8 |
| status | available |
| volume_id | 92dc3bd7-713d-42bf-83cd-4de40c24fed9 |
+-----+-----+

```

Эта команда отображает подробные данные о снимке тома mysnapshot.

3.15.4 vinfra service compute volume snapshot set

Изменение параметров снимка тома:

```
usage: vinfra service compute volume snapshot set [--description <description>]
                                                [--name <name>] <volume-snapshot>
```

`--description <description>`

Описание снимка тома

`--name <name>`

Новое имя для снимка тома

`<снимок_тома>`

Идентификатор или имя снимка тома

Пример:

```

# vinfra service compute volume snapshot set mysnapshot --name mynewsnapshot
+-----+-----+
| Field      | Value |
+-----+-----+
| created_at | 2019-04-30T13:12:54.297629+00:00 |
| description | |
| id         | 3fdfe5d6-8bd2-4bf5-8599-a9cef50e5b71 |
| metadata   | {} |
| name       | mynewsnapshot |
| project_id | fd0ae61496d04ef6bb637bc3167b7eaf |
| size       | 8 |
| status     | available |
| volume_id  | 92dc3bd7-713d-42bf-83cd-4de40c24fed9 |
+-----+-----+

```

Эта команда меняет имя снимка тома mysnapshot на mynewsnapshot.

3.15.5 vinfra service compute volume snapshot upload-to-image

Создание образа вычислений из снимка вычислительного тома:

```
usage: vinfra service compute volume snapshot upload-to-image [--name <name>]
      <volume-snapshot>
```

--name <name>

Имя образа

<снимок_тома>

Идентификатор или имя снимка тома

Пример:

```
# vinfra service compute volume snapshot upload-to-image --name myvm-image \
  mynewsnapshot
+-----+-----+
| Field          | Value                                |
+-----+-----+
| checksum       |                                       |
| container_format | bare                                  |
| created_at     |                                       |
| disk_format    | qcow2                                 |
| id             | 6a7a78c1-7168-4387-9b55-23fd477fdaa0 |
| min_disk       |                                       |
| min_ram        |                                       |
| name           | myvm-image                           |
| os_distro      | linux                                 |
| os_type        | linux                                 |
| project_id     |                                       |
| protected      | False                                 |
| public         | False                                 |
| size           | 1                                     |
| status         | uploading                             |
| tags           |                                       |
| updated_at     | 2019-06-07T12:30:43.462707          |
| virtual_size   |                                       |
+-----+-----+
```

Эта команда создает образ вычислений myvm-image из снимка тома mynewsnapshot.

3.15.6 vinfra service compute volume snapshot revert

Возврат тома к состоянию указанного снимка:

```
usage: vinfra service compute volume snapshot revert <volume-snapshot>
```

<снимок_тома>

Идентификатор или имя снимка тома

Пример:

```
# vinfra service compute volume snapshot revert mynewsnapshot
+-----+-----+
| Field      | Value                                |
+-----+-----+
| created_at | 2019-04-30T13:12:54.297629+00:00    |
| description |                                       |
| id         | 3fdfe5d6-8bd2-4bf5-8599-a9cef50e5b71 |
| metadata   | {}                                    |
| name       | mynewsnapshot                        |
| project_id | fd0ae61496d04ef6bb637bc3167b7eaf    |
| size       | 8                                     |
| status     | available                             |
| volume_id  | 92dc3bd7-713d-42bf-83cd-4de40c24fed9 |
+-----+-----+
```

Эта команда возвращает том к состоянию, определенному в его снимке mynewsnapshot.

3.15.7 vinfra service compute volume snapshot reset-state

Сброс снимка тома, застрявшего в состоянии «Ошибка» или в одном из переходных состояний, в состояние «Доступно»:

```
usage: vinfra service compute volume snapshot reset-state <volume-snapshot>
```

<снимок_тома>

Идентификатор или имя снимка тома

Пример:

```
# vinfra service compute volume snapshot reset-state mynewsnapshot
+-----+-----+
| Field      | Value                                |
+-----+-----+
| created_at | 2019-04-30T13:12:54.297629+00:00    |
| description |                                       |
+-----+-----+
```



```

| id          | 3fdfe5d6-8bd2-4bf5-8599-a9cef50e5b71 |
| metadata   | {}                                     |
| name       | mynewsnapshot                         |
| project_id | fd0ae61496d04ef6bb637bc3167b7eaf    |
| size       | 8                                      |
| status     | available                             |
| volume_id  | 92dc3bd7-713d-42bf-83cd-4de40c24fed9 |
+-----+-----+

```

Эта команда сбрасывает состояние снимка тома mysnapshot.

3.15.8 vinfra service compute volume snapshot delete

Удаление снимка тома:

```
usage: vinfra service compute volume snapshot delete <volume-snapshot>
```

<снимок_тома>

Идентификатор или имя снимка тома

Пример:

```
# vinfra service compute volume snapshot delete mynewsnapshot
Operation successful
```

Эта команда удаляет снимок тома mynewsnapshot.

3.16 Управление политиками хранилища

Управлять политиками хранилища можно только после создания вычислительного кластера.

3.16.1 vinfra cluster storage-policy create

Создание новой политики хранилища:

```
usage: vinfra cluster storage-policy create --tier {0,1,2,3}
      (--replicas <norm>[:<min>] |
      --encoding <M>+<N>) --failure-domain
      {disk,host,rack,row,room} <name>
```

--tier {0,1,2,3}

Уровень хранилища

```
--replicas <norm>[:<min>]
```

Схема репликации хранилища в формате:

- `norm`: количество сохраняемых реплик
- `min`: минимально требуемое количество реплик (необязательно)

```
--encoding <M>+<N>
```

Схема помехоустойчивого кодирования хранилища в формате:

- `M`: число блоков данных
- `N`: число паритетных блоков

```
--failure-domain {disk,host,rack,row,room}
```

Область отказа хранилища

```
<name>
```

Имя политики хранилища

Пример:

```
# vinfra cluster storage-policy create mystorpolicy --tier 3 \
--encoding 3+2 --failure-domain host
+-----+-----+
| Field          | Value                                |
+-----+-----+
| failure_domain | host                                  |
| id              | 2199e71e-ce8a-4ba9-81cd-75502f0344ca |
| name            | mystorpolicy                          |
| redundancy      | encoding=3+2                           |
| tier             | 3                                       |
+-----+-----+
```

Эта команда создает политику хранилища `mystorpolicy` и задает в ней уровень 3, схему избыточности — избыточное кодирование 3+2 и область отказа — хост.

3.16.2 vinfra cluster storage-policy list

Вывод списка имеющихся политик хранилищ:

```
usage: vinfra cluster storage-policy list
```

Пример:

```
# vinfra cluster storage-policy list
+-----+-----+-----+-----+-----+
| id          | name          | tier | redundancy | failure_domain |
+-----+-----+-----+-----+-----+
| 2199e71e-... | mystorpolicy  | 3   | encoding=3+2 | host           |
| 4274d6fd-... | default       | 0   | replicas=3   | host           |
+-----+-----+-----+-----+-----+
```

Эта команда выводит список политик хранилищ, доступных для вычислительного кластера.

3.16.3 vinfra cluster storage-policy show

Отображение подробных данных для политики хранилища:

```
usage: vinfra cluster storage-policy show <storage-policy>
```

<политика_хранилища>

Идентификатор или имя политики хранилища

Пример:

```
# vinfra cluster storage-policy show mystorpolicy
+-----+-----+-----+-----+-----+
| Field      | Value                                               |
+-----+-----+-----+-----+-----+
| failure_domain | host                                               |
| id          | 2199e71e-ce8a-4ba9-81cd-75502f0344ca             |
| name        | mystorpolicy                                       |
| redundancy   | encoding=3+2                                       |
| tier         | 3                                                  |
+-----+-----+-----+-----+-----+
```

Эта команда отображает подробные данные о политике хранилища mystorpolicy.

3.16.4 vinfra cluster storage-policy set

Изменение параметров политики хранилища:

```
usage: vinfra cluster storage-policy set [--name <name>] [--tier {0,1,2,3}]
      [--replicas <norm>[:<min>] |
      --encoding <M>+<N>] [--failure-domain
      {disk,host,rack,row,room}] <storage-policy>
```

--name <name>

Новое имя для политики хранилища

```
--tier {0,1,2,3}
```

Уровень хранилища

```
--replicas <norm>[:<min>]
```

Схема репликации хранилища в формате:

- `norm`: количество сохраняемых реплик
- `min`: минимально требуемое количество реплик (необязательно)

```
--encoding <M>+<N>
```

Схема помехоустойчивого кодирования хранилища в формате:

- `M`: число блоков данных
- `N`: число паритетных блоков

```
--failure-domain {disk,host,rack,row,room}
```

Область отказа хранилища

```
<политика_хранилища>
```

Идентификатор или имя политики хранилища

Пример:

```
# vinfra cluster storage-policy set mystorpolicy --encoding 5+2
+-----+-----+
| Field          | Value                                     |
+-----+-----+
| failure_domain | host                                     |
| id             | 2199e71e-ce8a-4ba9-81cd-75502f0344ca   |
| name           | mystorpolicy                             |
| redundancy     | encoding=5+2                             |
| tier            | 3                                         |
+-----+-----+
```

Эта команда меняет тип избыточности для политики хранилища `mystorpolicy` с избыточного кодирования 3+2 на 5+2.

3.16.5 vinfra cluster storage-policy delete

Политику по умолчанию удалить невозможно.

Удаление существующей политики хранилища:

```
usage: vinfra cluster storage-policy delete <storage-policy>
```

<политика_хранилища>

Идентификатор или имя политики хранилища

Пример:

```
# vinfra cluster storage-policy delete mystorpolicy
Operation successful
```

Эта команда удаляет политику хранилища mystorpolicy.

3.17 Управление кластерами Kubernetes

3.17.1 vinfra service compute k8saas create

Создание нового кластера Kubernetes:

```
usage: vinfra service compute k8saas create [--master-node-count <count>]
                                           [--node-count <count>]
                                           [--volume-storage-policy <policy>]
                                           [--kubernetes-version <version>]
                                           --master-flavor <flavor> --flavor
<flavor> [--volume-size <size>]
                                           --external-network <network>
                                           [--network <network>] --key-name
<key-name>
                                           [--use-floating-ip <use-floating-ip>]
<name>
```

<name>

Имя кластера Kubernetes

--master-node-count <count>

Количество мастер-серверов в кластере Kubernetes

--node-count <count>

Количество рабочих серверов в кластере Kubernetes

--volume-storage-policy <policy>

Идентификатор или имя политики хранилища для тома, на котором будут расположены контейнеры.

`--kubernetes-version <version>`

Версия Kubernetes

`--master-flavor <flavor>`

Тип VM, который следует использовать для мастер-серверов Kubernetes.

`--flavor <flavor>`

Тип VM, который следует использовать для рабочих серверов Kubernetes.

`--volume-size <size>`

Размер тома хранилища на каждом сервере Kubernetes

`--external-network <network>`

Идентификатор или имя внешней сети, которая будет обеспечивать доступ к Интернету для серверов Kubernetes.

`--network <network>`

Идентификатор или имя частной сети, которая будет обеспечивать взаимодействие между серверами Kubernetes.

`--key-name <key-name>`

Пара ключей, которую следует использовать для доступа к серверам Kubernetes.

`--use-floating-ip <use-floating-ip>`

Назначение плавающих IP-адресов мастер-серверам и рабочим серверам (true или false).

Необходимые условия для создания кластера Kubernetes:

- Компонент «Kubernetes как услуга». Его можно развернуть одновременно с вычислительным кластером или позже (см. раздел [Creating the Compute Cluster](#) или [Managing Add-On Services](#)).
- Частная сеть, которая будет соединять серверы Kubernetes. Для нее должны быть указаны шлюз и DNS-сервер.
- Внешняя сеть с доступом к Интернету (для обращения к сервису обнаружения etcd) и назначенными типами трафика **API вычислений** и **VM внешн.**
- SSH-ключ, который будет установлен на рабочих и мастер-серверах.
- Достаточно ресурсов для всех серверов Kubernetes с учетом их типов.

Пример:

```
# vinfra service compute k8saas create --master-node-count 1 --node-count 3 \
--volume-storage-policy default --kubernetes-version v1.15.6 --master-flavor medium \
--flavor small --volume-size 10 --external-network public1 --network private1 \
--key-name key1 --use-floating-ip true k8s1 --vinfra-username user1 \
--vinfra-password password --vinfra-domain domain1 --vinfra-project project1
```

Field	Value
boot_volume_size	10
boot_volume_storage_policy	default
containers_volume_size	10
containers_volume_storage_policy	default
create_timeout	60
external_network_id	7006065f-9067-4aed-b888-d89baa7004b8
id	c0754d99-6066-4675-8062-e62602939cf3
key_name	key1
master_flavor	medium
master_node_count	1
name	k8s1
network_id	d037623b-0db7-40c2-b38a-9ac34fbd1cc5
project_id	c734b9832e9540bd8f79bc2272c167e6
status	CREATING
user_id	c2cba773dc824125b07720744d0e49e2
worker_pools	- flavor: small node_count: 3

Эта команда, выполняемая от имени пользователя `user1` из `domain1 > project1`, запускает создание кластера Kubernetes `k8s1` с этими параметрами:

- версия Kubernetes 1.15.6
- 1 мастер-сервер на базе типа VM `medium` и 3 рабочих узла на базе типа VM `small`
- тома хранилища размером 10 ГБ с применением политики хранилища по умолчанию
- частная сеть `private1`, которая будет подключаться к Интернету через внешнюю сеть `public1`
- плавающие IP-адреса для каждого сервера, взятые из указанной внешней сети
- открытый SSH-ключ `key1`

3.17.2 vinfra service compute k8saas list

Вывод списка кластеров Kubernetes:

```
usage: vinfra service compute k8saas list
```

Пример:

```
# vinfra service compute k8saas list
+-----+-----+-----+
| id           | name | status |
+-----+-----+-----+
| f3e71ee8-8583-4b6a-abce-0132818f5108 | k8s1 | ACTIVE |
+-----+-----+-----+
```

Эта команда выводит список кластеров Kubernetes.

3.17.3 vinfra service compute k8saas config

Печать конфигурации кластера Kubernetes (должна выполняться от имени пользователя, создавшего этот кластер Kubernetes):

```
usage: vinfra service compute k8saas config <cluster>
```

cluster

Идентификатор или имя кластера

Пример:

```
# vinfra service compute k8saas config k8s1 --vinfra-domain domain1 \
--vinfra-project project1 --vinfra-username user1 --vinfra-password password \
> kubeconfig
```

Эта команда печатает конфигурацию кластера Kubernetes k8s1 в файл kubeconfig.

3.17.4 vinfra service compute k8saas show

Отображение сведений о кластере Kubernetes:

```
usage: vinfra service compute k8saas show <cluster>
```

cluster

Идентификатор или имя кластера

Пример:

```
# vinfra service compute k8saas show k8s1
+-----+-----+-----+
| Field           | Value |
+-----+-----+-----+
| boot_volume_size | 10    |
+-----+-----+-----+
```



```

| boot_volume_storage_policy      | default      |
| containers_volume_size         | 10           |
| containers_volume_storage_policy | default      |
| create_timeout                 | 60           |
| external_network_id            | 7006065f-9067-4aed-b888-d89baa7004b8 |
| id                              | c0754d99-6066-4675-8062-e62602939cf3 |
| key_name                       | key1         |
| master_flavor                  | medium       |
| master_node_count              | 1            |
| name                           | k8s1         |
| network_id                     | d037623b-0db7-40c2-b38a-9ac34fbd1cc5 |
| project_id                     | c734b9832e9540bd8f79bc2272c167e6   |
| stack_id                       | 3ef9ec9d-fde4-4358-bdb6-91205cd8ca52 |
| status                         | ACTIVE       |
| user_id                       | c2cba773dc824125b07720744d0e49e2   |
| version                        | v1.15.6     |
| worker_pools                   | - flavor: small
|                               |   node_count: 3
+-----+-----+

```

Эта команда отображает сведения о кластере Kubernetes k8s1.

3.17.5 vinfra service compute k8saas set

Изменение параметров кластера Kubernetes (должно выполняться от имени пользователя, создавшего этот кластер Kubernetes):

```

usage: vinfra service compute k8saas set [--node-count <count>]
                                           [--volume-storage-policy <policy>]
                                           <cluster>

```

cluster

Идентификатор или имя кластера

--node-count <count>

Количество рабочих серверов в кластере Kubernetes

--volume-storage-policy <policy>

Идентификатор или имя политики хранилища для тома, на котором будут расположены контейнеры.

Пример:

```

# vinfra service compute k8saas set --node-count 5 k8s1 \
--vinfra-domain domain1 --vinfra-project project1 \
--vinfra-username user1 --vinfra-password password
+-----+-----+

```

Field	Value
boot_volume_size	10
boot_volume_storage_policy	default
containers_volume_size	10
containers_volume_storage_policy	default
create_timeout	60
external_network_id	7006065f-9067-4aed-b888-d89baa7004b8
id	c0754d99-6066-4675-8062-e62602939cf3
key_name	key1
master_flavor	medium
master_node_count	1
name	k8s1
network_id	d037623b-0db7-40c2-b38a-9ac34fbd1cc5
project_id	c734b9832e9540bd8f79bc2272c167e6
stack_id	3ef9ec9d-fde4-4358-bdb6-91205cd8ca52
status	ACTIVE
user_id	c2cba773dc824125b07720744d0e49e2
version	v1.15.6
worker_pools	- flavor: small node_count: 3

Эта команда запускает настройку кластера Kubernetes k8s1, чтобы число рабочих серверов равнялось 5.

3.17.6 vinfra service compute k8saas delete

Удаление кластера Kubernetes:

```
usage: vinfra service compute k8saas delete <cluster>
```

cluster

Идентификатор или имя кластера

Пример:

```
# vinfra service compute k8saas delete k8s1
Operation accepted.
```

Эта команда удаляет кластер Kubernetes k8s1.

3.18 Управление квотами вычислительных ресурсов

3.18.1 `vinfra service compute quotas show`

Вывод списка квот вычислительных ресурсов:

```
usage: vinfra service compute quotas show [--usage] <project_id>
```

`--usage`

Включить использование квот.

`<project_id>`

Идентификатор проекта

Пример:

```
# vinfra service compute quotas show 6ef6f48f01b640ccb8ff53117b830fa3 --usage
+-----+-----+
| Field                | Value |
+-----+-----+
| compute.cores.limit  | 20    |
| compute.cores.used   | 2     |
| compute.ram.limit    | 40960 |
| compute.ram.used     | 4096  |
| k8saas.cluster.limit | 10    |
| k8saas.cluster.used  | 0     |
| lbaas.loadbalancer.limit | 10   |
| lbaas.loadbalancer.used  | 0     |
| network.floatingip.limit | 10   |
| network.floatingip.used  | 0     |
| storage.gigabytes.default.limit | 1024 |
| storage.gigabytes.default.used  | 66   |
+-----+-----+
```

Эта команда отображает квоты вычислительных ресурсов и их использование для проекта с идентификатором `6ef6f48f01b640ccb8ff53117b830fa3`.

3.18.2 `vinfra service compute quotas update`

Обновление квот вычислительных ресурсов:

```
usage: vinfra service compute quotas update [--cores <cores>] [--ram <ram>]
                                           [--floatingip <floatingip>]
                                           [--gigabytes <storage_policy>:<size>]
                                           [--k8saas-cluster <cluster>]
                                           <project_id>
```

`--cores <cores>`

Количество ядер

`--ram <ram>`

Объем ОЗУ в мегабайтах

`--floatingip <floatingip>`

Количество плавающих IP-адресов

`--gigabytes <storage_policy>:<size>`

Список <storage_policy>:<size> через запятую

`--k8saas-cluster <cluster>`

Количество кластеров Kubernetes

`<project_id>`

Идентификатор проекта

Пример:

```
# vinfra service compute quotas update 6ef6f48f01b640ccb8ff53117b830fa3 --cores 10 \
--ram 20480 --gigabytes default:512
Operation successful.
```

Эта команда обновляет квоты вычислительных ресурсов до 10 виртуальных ЦП, 20 ГБ ОЗУ и 512 ГБ дискового пространства для политики хранилища default.

3.19 Управление обновлениями

3.19.1 vinfra software-updates check-for-updates

Проверка наличия обновлений для программного обеспечения:

```
usage: vinfra software-updates check-for-updates
```

Пример:

```
# vinfra software-updates check-for-updates
+-----+-----+
| Field          | Value          |
+-----+-----+
| available_storage_release |
| last_check_datetime |
| nodes          | - current_storage_release:
|                 |   release: '756'
|                 |   version: 3.5.0
|                 |   downloaded_storage_release: null
|                 |   host: node1.vstoragedomain
|                 |   id: 51cc14d4-eec6-433e-a7b1-e1c5c7f9555e
|                 |   orig_hostname: node1
|                 |   status: download_completed
| services       | []
| status         | uptodate
| tasks         |
+-----+-----+
```

3.19.2 vinfra software-updates eligibility-check

Проверка того, подлежат ли серверы обновлению:

```
usage: vinfra software-updates eligibility-check
```

Пример:

```
# vinfra software-updates eligibility-check
+-----+-----+
| Field  | Value          |
+-----+-----+
| task_id | 0143aec7-f9ce-4654-ad48-edb6f4104e22 |
+-----+-----+
```

Эта команда создает задачу для проверки того, подлежат ли серверы в кластере хранилища установке обновлений.

Результат выполнения задания:

```
# vinfra task show 0143aec7-f9ce-4654-ad48-edb6f4104e22
+-----+-----+
| Field  | Value          |
+-----+-----+
| details |
| name    | backend.presentation.software_updates.tasks.EligibilityCheckTask |
| result  | cluster_has_releasing_nodes:
|         |   details: null
|         |   exception: null
|         |
+-----+-----+
```

```

|         | message: null
|         | passed: true
|         | severity: critical
|         | cluster_unhealthy:
|         |   details: null
|         |   exception: null
|         |   message: null
|         |   passed: true
|         |   severity: critical
|         | not_enough_space_on_agents:
|         |   details: null
|         |   exception: null
|         |   message: null
|         |   passed: true
|         |   severity: critical
|         | not_enough_space_on_mn:
|         |   details: null
|         |   exception: null
|         |   message: null
|         |   passed: true
|         |   severity: critical
|         | postgres_not_running:
|         |   details: null
|         |   exception: null
|         |   message: null
|         |   passed: true
|         |   severity: critical
| state   | success
| task_id | 0143aec7-f9ce-4654-ad48-edb6f4104e22
+-----+

```

3.19.3 vinfra software-updates download

Загрузка обновлений для программного обеспечения:

```
usage: vinfra software-updates download
```

Пример:

```

# vinfra software-updates download
+-----+-----+
| Field  | Value
+-----+-----+
| task_id | 2f930030-22de-4ce5-bf00-05328ee672f0 |
+-----+-----+

```

Эта команда создает задачу загрузки обновлений для программного обеспечения.

Результат выполнения задания:

```
# vinfra task show 2f930030-22de-4ce5-bf00-05328ee672f0
+-----+-----+
| Field  | Value |
+-----+-----+
| details |       |
| name    | backend.business.models.software_updates.tasks.DownloadSoftwareUpdatesTask |
| result  |       |
| state   | success |
| task_id | 2f930030-22de-4ce5-bf00-05328ee672f0 |
+-----+-----+
```

3.19.4 vinfra software-updates start

Запуск процедуры обновления программного обеспечения:

```
usage: vinfra software-updates start [--skip] [--force]
```

`--skip`

Пропустить и не обновлять серверы, которые не могут перейти в режим обслуживания.

`--force`

Принудительно обновить и перезагрузить (при необходимости) все серверы, даже если они не могут перейти в режим обслуживания. Использование этого параметра может привести к простоям.

Если ни один из этих двух параметров не указан, обновление будет останавливаться в случае, когда сервер не может перейти в режим обслуживания. При этом серверы, уже обновленные на текущий момент, остаются обновленными.

Пример:

```
# vinfra software-updates start
+-----+-----+
| Field  | Value |
+-----+-----+
| task_id | 0eae9159-7595-42a7-8feb-d04df3e295c7 |
+-----+-----+
```

Эта команда создает задачу для запуска обновлений.

Результат выполнения задания:

```
# vinfra task show 0eae9159-7595-42a7-8feb-d04df3e295c7
+-----+-----+
| Field  | Value |
+-----+-----+
```

```

| details |
| name    | backend.business.models.software_updates.tasks.StartSoftwareUpdateTask |
| result  |
| state   | running
| task_id | 0eae9159-7595-42a7-8feb-d04df3e295c7
+-----+

```

3.19.5 vinfra software-updates pause

Вывод состояния обновлений программного обеспечения:

```
usage: vinfra software-updates pause
```

Пример:

```

# vinfra software-updates pause
+-----+
| Field  | Value
+-----+
| task_id | b02a686b-3214-447e-a9b4-43698aa9388b |
+-----+

```

Эта команда создает задачу для приостановки обновлений.

Результат выполнения задания:

```

# vinfra task show b02a686b-3214-447e-a9b4-43698aa9388b
+-----+
| Field  | Value
+-----+
| details |
| name    | backend.presentation.software_updates.tasks.PauseSoftwareUpdateTask |
| result  |
| state   | success
| task_id | b02a686b-3214-447e-a9b4-43698aa9388b
+-----+

```

3.19.6 vinfra software-updates resume

Возобновление процедуры обновления ПО:

```
usage: vinfra software-updates resume [--skip] [--force]
```

`--skip`

Пропустить и не обновлять серверы, которые не могут перейти в режим обслуживания.

--force

Принудительно обновить и перезагрузить (при необходимости) все серверы, даже если они не могут перейти в режим обслуживания. Использование этого параметра может привести к простоям.

Если ни один из этих двух параметров не указан, обновление будет останавливаться в случае, когда сервер не может перейти в режим обслуживания. При этом серверы, уже обновленные на текущий момент, остаются обновленными.

Пример:

```
# vinfra software-updates resume
+-----+-----+
| Field | Value |
+-----+-----+
| task_id | 35323989-bdbc-4826-94c3-70ed7d06969d |
+-----+-----+
```

Эта команда создает задачу для возобновления обновления.

Результат выполнения задания:

```
# vinfra task show 35323989-bdbc-4826-94c3-70ed7d06969d
+-----+-----+
| Field | Value |
+-----+-----+
| details | |
| name | backend.presentation.software_updates.tasks.ResumeSoftwareUpdateTask |
| result | |
| state | success |
| task_id | 35323989-bdbc-4826-94c3-70ed7d06969d |
+-----+-----+
```

3.19.7 vinfra software-updates cancel

Отмена обновлений для программного обеспечения:

```
usage: vinfra software-updates cancel
```

Пример:

```
# vinfra software-updates cancel
+-----+-----+
| Field | Value |
+-----+-----+
```

```
| task_id | 7aeb20ba-1f9f-4f28-9790-086428d3e18e |
+-----+-----+
```

Эта команда создает задачу для отмены обновления.

Результат выполнения задания:

```
# vinfra task show 7aeb20ba-1f9f-4f28-9790-086428d3e18e
+-----+-----+
| Field | Value |
+-----+-----+
| details |
| name | backend.presentation.software_updates.tasks.CancelSoftwareUpdateTask |
| result |
| state | success |
| task_id | 7aeb20ba-1f9f-4f28-9790-086428d3e18e |
+-----+-----+
```

3.19.8 vinfra software-updates status

Проверка состояния обновлений программного обеспечения:

```
usage: vinfra software-updates status
```

Пример:

```
# vinfra software-updates status
+-----+-----+
| Field | Value |
+-----+-----+
| available_storage_release | release: '758' |
| | version: 3.5.0 |
| last_check_datetime | 2019-12-17T13:25:41.991763+00:00 |
| nodes | - current_storage_release: |
| | release: '758' |
| | version: 3.5.0 |
| | downloaded_storage_release: null |
| | host: man-hci7-1.vstoragedomain |
| | id: 51cc14d4-eec6-433e-a7b1-e1c5c7f9555e |
| | orig_hostname: man-hci7-1 |
| | status: uptodate |
| services | [] |
| status | uptodate |
| tasks | - errors: |
| | message: None |
| | nodes: [] |
| | id: 8cf880d1-6648-450b-b96c-c87c27b9e181 |
| | name: StartSoftwareUpdateTask |
| | params: |
+-----+-----+
```

```
|           | force: false |
|           | skip: false  |
+-----+-----+
```

Эта команда показывает состояние обновления сервера.

ГЛАВА 4

Управление резервным кластером

4.1 Создание, отображение и удаление резервного кластера

4.1.1 `vinfra service backup cluster create`

Создание резервного кластера

```
usage: vinfra service backup cluster create --nodes <nodes> --domain <domain>
      --reg-account <reg-account>
      --reg-server <reg-server> --tier {0,1,2,3}
      --encoding <M>+<N> --failure-domain
      {disk,host,rack,row,room}
      --storage-type {local,nfs,s3,swift,azure,
      google} [--stdin]
      [--nfs-host <host>]
      [--nfs-export <export>]
      [--nfs-version <version>]
      [--s3-flavor <flavor>]
      [--s3-region <region>]
      [--s3-bucket <bucket>]
      [--s3-endpoint <endpoint>]
      [--s3-access-key-id <access-key-id>]
      [--s3-secret-key-id <secret-key-id>]
      [--s3-cert-verify <cert-verify>]
      [--swift-auth-url <auth-url>]
      [--swift-auth-version <auth-version>]
      [--swift-user-name <user-name>]
```

```

[--swift-api-key <api-key>]
[--swift-domain <domain>]
[--swift-domain-id <domain-id>]
[--swift-tenant <tenant>]
[--swift-tenant-id <tenant-id>]
[--swift-tenant-domain <tenant-domain>]
[--swift-tenant-domain-id <tenant-domain-id>]
[--swift-trust-id <trust-id>]
[--swift-region <region>]
[--swift-internal <internal>]
[--swift-container <container>]
[--swift-cert-verify <cert-verify>]
[--azure-endpoint <endpoint>]
[--azure-container <container>]
[--azure-account-name <account-nameE>]
[--azure-account-key <account-key>]
[--google-bucket <bucket>]
[--google-credentials <credentials>]

```

--nodes <nodes>

Список имен хостов или идентификаторов серверов через запятую

--domain <domain>

Имя домена для резервного кластера

--reg-account <reg-account>

Партнерская учетная запись в облаке или учетная запись администратора организации на локальном сервере управления

--reg-server <reg-server>

URL-адрес портала управления облаком или имя хоста/IP-адрес и порт локального сервера управления

--tier {0,1,2,3}

Уровень хранилища

--encoding <M>+<N>

Схема помехоустойчивого кодирования хранилища в формате:

- M: число блоков данных
- N: число паритетных блоков

--failure-domain {disk,host,rack,row,room}

Область отказа хранилища

```
--storage-type {local,nfs,s3,swift,azure,google}
```

Тип хранилища

```
--stdin
```

Запрашивать ввод пароля регистрации в консоли.

Параметры хранилища типа nfs:

```
--nfs-host <host>
```

Имя хоста или IP-адрес NFS

```
--nfs-export <export>
```

Полный путь к экспорту NFS

```
--nfs-version <version>
```

Версия NFS (3 или 4)

Параметры хранилища типа s3:

```
--s3-flavor <flavor> (необязательно)
```

Имя типа VM

```
--s3-region <region> (необязательно)
```

Задайте регион для Amazon S3.

```
--s3-bucket <bucket>
```

Имя корзины

```
--s3-endpoint <endpoint>
```

URL-адрес конечной точки

```
--s3-access-key-id <access-key-id>
```

Идентификатор ключа доступа

```
--s3-secret-key-id <secret-key-id>
```

Идентификатор секретного ключа

```
--s3-cert-verify <cert-verify> (необязательно)
```

Разрешить самозаверяющий сертификат конечной точки S3

Параметры хранилища типа swift:

```
--swift-auth-url <auth-url>
```

URL аутентификации (Keystone)

`--swift-auth-version <auth-version> (необязательно)`
Версия протокола проверки подлинности

`--swift-user-name <user-name>`
Имя пользователя

`--swift-api-key <api-key>`
Ключ API (пароль)

`--swift-domain <domain> (необязательно)`
Имя домена

`--swift-domain-id <domain-id> (необязательно)`
Идентификатор домена

`--swift-tenant <tenant> (необязательно)`
Имя тенанта

`--swift-tenant-id <tenant-id> (необязательно)`
Идентификатор тенанта

`--swift-tenant-domain <tenant-domain> (необязательно)`
Имя домена тенанта

`--swift-tenant-domain-id <tenant-domain-id> (необязательно)`
Идентификатор домена тенанта

`--swift-trust-id <trust-id> (необязательно)`
Идентификатор Trust

`--swift-region <region> (необязательно)`
Имя региона

`--swift-container <container> (необязательно)`
Имя контейнера

`--swift-cert-verify <cert-verify> (необязательно)`
Разрешить самозаверяющий сертификат конечной точки Swift (true или false)

Параметры хранилища типа azure:

`--azure-endpoint <endpoint>`
URL-адрес конечной точки

```
--azure-container <container>
```

Имя контейнера

```
--azure-account-name <account-name>
```

Имя учетной записи

```
--azure-account-key <account-key>
```

Ключ учетной записи

Параметры хранилища типа google:

```
--google-bucket <bucket>
```

Имя корзины Google

```
--google-credentials <credentials>
```

Путь к файлу с учетными данными Google

Пример 1. Создание резервного кластера в локальном хранилище.

```
# vinfra service backup cluster create --nodes 2f3f6091-0d44-45aa-94e3-ebc2b65c0eeb, \
74cbd22b-fb1b-4441-ae52-532078c54f9a, eeb06dce-4cfd-4c89-bc7f-4689ea5c7058 \
--storage-type local --domain dns.example.com --tier 0 --encoding 1+2 --failure-domain \
host --reg-account account@example.com --reg-server https://cloud.acronis.com/ --stdin
Password:
+-----+-----+
| Field  | Value                                     |
+-----+-----+
| task_id | ee7e60c5-5447-4177-8581-26657ac380c0 |
+-----+-----+
```

Эта команда создает задание для создания резервного кластера из трех серверов с указанными идентификаторами в локальном хранилище. Она также указывает доменное имя, уровень хранилища, область отказа, учетную запись и сервер регистрации.

Результат выполнения задания:

```
# vinfra task show ee7e60c5-5447-4177-8581-26657ac380c0
+-----+-----+
| Field  | Value                                     |
+-----+-----+
| details |                                           |
| name    | backend.presentation.abgw.tasks.RegisterAbgwTask |
| result  |                                           |
| state   | success                                     |
| task_id | ee7e60c5-5447-4177-8581-26657ac380c0 |
+-----+-----+
```

Пример 2. Создание резервного кластера в хранилище S3.


```
# vinfra service backup cluster create --nodes 2f3f6091-0d44-45aa-94e3-ebc2b65c0eeb,\
74cbd22b-fb1b-4441-ae52-532078c54f9a,eeb06dce-4cfd-4c89-bc7f-4689ea5c7058 \
--storage-type s3 --domain dns.example.com --tier 0 --encoding 1+2 --failure-domain \
host --s3-bucket mybucket --s3-endpoint s3.amazonaws.com --s3-access-key-id \
e302a06df8adbe9fAIF1 --s3-secret-key-id x1gXquRHQXuyiUJQoQMoAohA2TkYHer20o8tfPX7 \
--s3-cert-verify true --reg-account account@example.com --reg-server \
https://cloud.acronis.com/ --stdin
Password:
+-----+-----+
| Field   | Value                                     |
+-----+-----+
| task_id | 0fb53a6f-2bc4-410a-aa1c-b3cda6ca8570 |
+-----+-----+
```

Эта команда создает задание для создания резервного кластера из трех серверов с указанными идентификаторами в хранилище S3. Она также указывает доменное имя, уровень хранилища, область отказа, учетную запись и сервер регистрации, а также необходимые параметры S3.

Результат выполнения задания:

```
# vinfra task show 0fb53a6f-2bc4-410a-aa1c-b3cda6ca8570
+-----+-----+
| Field   | Value                                     |
+-----+-----+
| details |                                           |
| name    | backend.presentation.abgw.tasks.RegisterAbgwTask |
| result  |                                           |
| state   | success                                   |
| task_id | 0fb53a6f-2bc4-410a-aa1c-b3cda6ca8570 |
+-----+-----+
```

Пример 3. Создание резервного кластера в NFS-хранилище.

```
# vinfra service backup cluster create --nodes eeb06dce-4cfd-4c89-bc7f-4689ea5c7058 \
--storage-type nfs --domain dns.example.com --tier 0 --encoding 1+2 --failure-domain \
host --nfs-host nfs.example.com --nfs-export /myshare/myexport --nfs-version 4 \
--reg-account account@example.com --reg-server https://cloud.acronis.com/ --stdin
Password:
+-----+-----+
| Field   | Value                                     |
+-----+-----+
| task_id | d76ceb22-48e7-4eac-b04f-03d3aa3377b7 |
+-----+-----+
```

Эта команда создает задание для создания резервного кластера из одного сервера с идентификатором eeb06dce-4cfd-4c89-bc7f-4689ea5c7058 в хранилище NFS. Она также указывает доменное имя, уровень хранилища, область отказа, учетную запись и сервер регистрации, а также необходимые параметры NFS.

Результат выполнения задания:

```
# vinfra task show d76ceb22-48e7-4eac-b04f-03d3aa3377b7
+-----+-----+
| Field | Value |
+-----+-----+
| details | |
| name | backend.presentation.abgw.tasks.RegisterAbgwTask |
| result | |
| state | success |
| task_id | d76ceb22-48e7-4eac-b04f-03d3aa3377b7 |
+-----+-----+
```

4.1.2 vinfra service backup cluster show

Отображение сведений о резервном кластере:

```
usage: vinfra service backup cluster show
```

Пример:

```
# vinfra service backup cluster show
+-----+-----+
| Field | Value |
+-----+-----+
| abgw_address | dns.example.com |
| account_server | https://cloud.acronis.com |
| dc_uid | 44893a40296ecd9ae64567297a5b2b07-1577203369 |
| migration | dns: null |
| | ips: [] |
| | running: false |
| | time_left: 0.0 |
| reg_type | abc |
| storage_params | access_key_id: e302a06df8adbe9fAIF1 |
| | bucket: mybucket |
| | cert_verify: true |
| | endpoint: s3.amazonaws.com |
| | flavour: null |
| | region: null |
| | secret_key_id: x1gXquRHQXuyiUJQoQMoAohA2TkYHer20o8tfPX7 |
| storage_type | s3 |
+-----+-----+
```

Эта команда отображает доменное имя, сведения о регистрации и параметры хранилища для резервного кластера.

4.1.3 vinfra service backup cluster release

Удаление резервного кластера и всех его данных:

```
usage: vinfra service backup cluster release [--reg-account <reg-account>]
                                           [--force] [--stdin]
```

`--reg-account <reg-account>`

Партнерская учетная запись в облаке или учетная запись администратора организации на локальном сервере управления

`--force`

Освобождает ресурсы резервного кластера, но не отменяет его регистрацию в программе резервного копирования.

Примечание: Выбирайте этот вариант, только если уверены, что регистрация кластера уже удалена из программы резервного копирования.

`--stdin`

Запрашивать ввод пароля регистрации в консоли.

Пример:

```
# vinfra service backup cluster release --reg-account account@example.com --stdin
Password:
+-----+-----+
| Field | Value |
+-----+-----+
| task_id | cf270233-06d5-4a4a-8dea-443d6fb59b10 |
+-----+-----+
```

Эта команда создает задание для удаления резервного кластера со всеми данными и отмены его регистрации в программе резервного копирования.

Результат выполнения задания:

```
# vinfra task show cf270233-06d5-4a4a-8dea-443d6fb59b10
+-----+-----+
| Field | Value |
+-----+-----+
| details | |
| name | backend.presentation.abgw.tasks.ReleaseNodesTask |
| result | |
| state | success |
+-----+-----+
```


4.2.2 vinfra service backup node list

Вывод списка серверов в резервном кластере:

```
usage: vinfra service backup node list
```

Пример:

```
# vinfra service backup node list
+-----+-----+-----+
| id                | host                | is_online |
+-----+-----+-----+
| 2f3f6091-0d44-45aa-94e3-ebc2b65c0eeb | node003.vstoragedomain | True      |
| 74cbd22b-fb1b-4441-ae52-532078c54f9a | node001.vstoragedomain | True      |
| eeb06dce-4cfd-4c89-bc7f-4689ea5c7058 | node002.vstoragedomain | True      |
+-----+-----+-----+
```

Эта команда выводит список серверов в резервном кластере.

4.2.3 vinfra service backup node release

Освобождение списка серверов из резервного кластера:

```
usage: vinfra service backup node release --nodes <nodes>
```

--nodes <nodes>

Список имен хостов или идентификаторов серверов через запятую

Пример:

```
# vinfra service backup node release --nodes 2f3f6091-0d44-45aa-94e3-ebc2b65c0eeb
+-----+-----+
| Field  | Value |
+-----+-----+
| task_id | ea09642c-291c-4df8-87a5-a8958d6308c1 |
+-----+-----+
```

Эта команда создает задание для освобождения сервера с идентификатором 2f3f6091-0d44-45aa-94e3-ebc2b65c0eeb из резервного кластера.

Результат выполнения задания:

```
# vinfra task show ea09642c-291c-4df8-87a5-a8958d6308c1
+-----+-----+
| Field  | Value |
+-----+-----+
| details |       |
```

```

| name      | backend.presentation.abgw.tasks.ReleaseNodesTask |
| result    |                                                     |
| state     | success                                           |
| task_id   | ea09642c-291c-4df8-87a5-a8958d6308c1           |
+-----+-----+

```

4.3 Обновление сертификатов резервного кластера

Обновление сертификатов для резервного кластера:

```

usage: vinfra service backup cluster renew-certificates [--stdin]
                                                --reg-account <reg-account>
                                                --reg-server <reg-server>

```

`--stdin`

Запрашивать ввод пароля регистрации в консоли.

`--reg-account <reg-account>`

Партнерская учетная запись в облаке или учетная запись администратора организации на локальном сервере управления

`--reg-server <reg-server>`

URL-адрес портала управления облаком или имя хоста/IP-адрес и порт локального сервера управления

Пример:

```

# vinfra service backup cluster renew-certificates --reg-account account@example.com \
--reg-server https://cloud.acronis.com/ --stdin
Password:
+-----+-----+
| Field  | Value |
+-----+-----+
| task_id | 7f1873a7-cd9b-49f3-ae17-fb14ff08ddf5 |
+-----+-----+

```

Эта команда создает задание для обновления сертификатов резервного кластера.

Результат выполнения задания:

```

# vinfra task show 7f1873a7-cd9b-49f3-ae17-fb14ff08ddf5
+-----+-----+
| Field  | Value |
+-----+-----+

```

```
+-----+-----+
| details |                                     |
| name    | backend.presentation.abgw.tasks.RenewRegistrationAbgwTask |
| result  |                                     |
| state   | success                                     |
| task_id | 7f1873a7-cd9b-49f3-ae17-fb14ff08ddf5 |
+-----+-----+
```

4.4 Изменение параметров хранилища

4.4.1 vinfra service backup storage-params show

Отображение параметров хранилища:

```
usage: vinfra service backup storage-params show
```

Пример:

```
# vinfra service backup storage-params show
+-----+-----+
| Field          | Value                                     |
+-----+-----+
| abgw_address   | dns.example.com                         |
| account_server | https://cloud.acronis.com               |
| dc_uid         | 44893a40296ecd9ae64567297a5b2b07-1577264050 |
| migration      | dns: null                               |
|                | ips: []                                 |
|                | running: false                          |
|                | time_left: 0.0                          |
| reg_type       | abc                                      |
| storage_params | export: /myshare/myexport               |
|                | host: 10.94.129.70                      |
|                | version: 4                               |
| storage_type   | nfs                                      |
+-----+-----+
```

Эта команда отображает параметры хранилища для резервного кластера.

4.4.2 vinfra service backup storage-params change

Важно: Изменяйте параметры хранилища с осторожностью и только в рамках существующей конфигурации. Можно изменить IP-адрес внешнего хранилища или учетные данные для доступа к нему.

Изменение параметров хранилища:

```
usage: vinfra service backup storage-params change --storage-type {local,nfs,s3,swift,
        azure,google}
        [--nfs-host <host>]
        [--nfs-export <export>]
        [--nfs-version <version>]
        [--s3-flavor <flavor>]
        [--s3-region <region>]
        [--s3-bucket <bucket>]
        [--s3-endpoint <endpoint>]
        [--s3-access-key-id <access-key-id>]
        [--s3-secret-key-id <secret-key-id>]
        [--s3-cert-verify <cert-verify>]
        [--swift-auth-url <auth-url>]
        [--swift-auth-version <auth-version>]
        [--swift-user-name <user-name>]
        [--swift-api-key <api-key>]
        [--swift-domain <domain>]
        [--swift-domain-id <domain-id>]
        [--swift-tenant <tenant>]
        [--swift-tenant-id <tenant-id>]
        [--swift-tenant-domain <tenant-domain>]
        [--swift-tenant-domain-id <tenant-domain-id>]
        [--swift-trust-id <trust-id>]
        [--swift-region <region>]
        [--swift-internal <internal>]
        [--swift-container <container>]
        [--swift-cert-verify <cert-verify>]
        [--azure-endpoint <endpoint>]
        [--azure-container <container>]
        [--azure-account-name <account-name>]
        [--azure-account-key <account-key>]
        [--google-bucket <bucket>]
        [--google-credentials <credentials>]
```

```
--storage-type {local,nfs,s3,swift,azure,google}
```

Тип хранилища

Параметры хранилища типа nfs:

--nfs-host <host>

Имя хоста или IP-адрес NFS

--nfs-export <export>

Полный путь к экспорту NFS

--nfs-version <version>

Версия NFS (3 или 4)

Параметры хранилища типа s3:

--s3-flavor <flavor> **(необязательно)**

Имя типа VM

--s3-region <region> **(необязательно)**

Задайте регион для Amazon S3.

--s3-bucket <bucket>

Имя корзины

--s3-endpoint <endpoint>

URL-адрес конечной точки

--s3-access-key-id <access-key-id>

Идентификатор ключа доступа

--s3-secret-key-id <secret-key-id>

Идентификатор секретного ключа

--s3-cert-verify <cert-verify> **(необязательно)**

Разрешить самозаверяющий сертификат конечной точки S3

Параметры хранилища типа swift:

--swift-auth-url <auth-url>

URL аутентификации (Keystone)

--swift-auth-version <auth-version> **(необязательно)**

Версия протокола проверки подлинности

--swift-user-name <user-name>

Имя пользователя

```
--swift-api-key <api-key>  
    Ключ API (пароль)  
  
--swift-domain <domain> (необязательно)  
    Имя домена  
  
--swift-domain-id <domain-id> (необязательно)  
    Идентификатор домена  
  
--swift-tenant <tenant> (необязательно)  
    Имя тенанта  
  
--swift-tenant-id <tenant-id> (необязательно)  
    Идентификатор тенанта  
  
--swift-tenant-domain <tenant-domain> (необязательно)  
    Имя домена тенанта  
  
--swift-tenant-domain-id <tenant-domain-id> (необязательно)  
    Идентификатор домена тенанта  
  
--swift-trust-id <trust-id> (необязательно)  
    Идентификатор Trust  
  
--swift-region <region> (необязательно)  
    Имя региона  
  
--swift-container <container> (необязательно)  
    Имя контейнера  
  
--swift-cert-verify <cert-verify> (необязательно)  
    Разрешить самоверяющийся сертификат конечной точки Swift (true или false)
```

Параметры хранилища типа azure:

```
--azure-endpoint <endpoint>  
    URL-адрес конечной точки  
  
--azure-container <container>  
    Имя контейнера  
  
--azure-account-name <account-name>  
    Имя учетной записи
```

```
--azure-account-key <account-key>
```

Ключ учетной записи

Параметры хранилища типа google:

```
--google-bucket <bucket>
```

Имя корзины Google

```
--google-credentials <credentials>
```

Путь к файлу с учетными данными Google

Пример:

```
# vinfra service backup storage-params change --storage-type nfs --nfs-host \
10.94.129.71 --nfs-export /myshare/myexport --nfs-version 4
Operation successful.
```

Эта команда изменяет параметры хранилища NFS для резервного кластера.

4.5 Изменение параметров тома

4.5.1 vinfra service backup volume-params show

Отображение параметров тома:

```
usage: vinfra service backup volume-params show
```

Пример:

```
# vinfra service backup volume-params show
+-----+-----+
| Field      | Value      |
+-----+-----+
| failure_domain | host      |
| redundancy    | m: 1      |
|              | n: 2      |
|              | type: raid6 |
| tier         | 0         |
+-----+-----+
```

Эта команда отображает параметры тома для резервного кластера: область отказа, схему избыточности и уровень хранилища.

4.5.2 vinfra service backup volume-params change

Важно: Изменение схемы избыточности не рекомендуется, поскольку это может снизить производительность кластера. Причина в том, что перекодирование потребляет значительный объем ресурсов кластера в течение длительного времени. Если вы все равно хотите изменить схему избыточности, обратитесь в техническую поддержку.

Изменение параметров тома:

```
usage: vinfra service backup volume-params change [--tier {0,1,2,3}] [--encoding <M>+<N>]
        [--failure-domain {disk,host,rack,
        row,room}]
```

`--tier {0,1,2,3}`

Уровень хранилища

`--encoding <M>+<N>`

Схема помехоустойчивого кодирования хранилища в формате:

- M: число блоков данных
- N: число паритетных блоков

`--failure-domain {disk,host,rack,row,room}`

Область отказа хранилища

Пример:

```
# vinfra service backup volume-params change --tier 1 --encoding 1+0 \
--failure-domain host
+-----+-----+
| Field  | Value                                |
+-----+-----+
| task_id | 28ae19dc-51c9-49bf-bd93-51a763fa181b |
+-----+-----+
```

Эта команда создает задание для изменения параметров тома резервного кластера следующим образом:

- уровень хранилища — 1
- схема помехоустойчивого кодирования — 1+0
- область отказа — хост

Результат выполнения задания:

```
# vinfra task show 28ae19dc-51c9-49bf-bd93-51a763fa181b
+-----+-----+
| Field | Value |
+-----+-----+
| details |
| name | backend.presentation.abgw.tasks.ChangeVolumeParamsTask |
| result |
| state | success |
| task_id | 28ae19dc-51c9-49bf-bd93-51a763fa181b |
+-----+-----+
```

4.6 Управление георепликацией резервного кластера

Важно: Чтобы включить георепликацию, резервные кластеры должны иметь доступ друг к другу посредством доменных имен через TCP-порт 44445.

Чтобы включить георепликацию между двумя резервными кластерами, выполните следующие команды (см. примеры в разделах ниже):

1. На кластере, который будет настроен как подчиненный, выполните команду `vinfra service backup geo-replication show`, чтобы узнать его адрес и UID.
2. На кластере, который будет настроен как главный, выполните команду `vinfra service backup geo-replication master setup` с использованием адреса и UID подчиненного кластера.
3. На главном кластере выполните команду `vinfra service backup geo-replication master download-configs`, чтобы создать файл конфигурации главного кластера.
4. Переместите файл конфигурации главного кластера на подчиненный кластер с помощью стандартной программы командной строки Linux, например `scp`.
5. На подчиненном кластере выполните команду `vinfra service backup geo-replication slave setup`, чтобы загрузить файл конфигурации главного кластера.
6. На главном кластере выполните команду `vinfra service backup geo-replication master establish`, чтобы установить соединение между главным и подчиненным кластерами.

7. На одном из кластеров выполните команду `vinfra service backup geo-replication show`, чтобы проверить, что георепликация успешно включена.

4.6.1 `vinfra service backup geo-replication show`

Отображение конфигурации георепликации:

```
usage: vinfra service backup geo-replication show
```

Пример:

```
# vinfra service backup geo-replication show
+-----+-----+
| Field | Value |
+-----+-----+
| self  | address: slave.example.com |
|       | datacenter_uid: e63a67388deb3c99d044eecbd7b79ad3-1577275849 |
+-----+-----+
```

Эта команда отображает конфигурацию георепликации для подчиненного кластера.

4.6.2 `vinfra service backup geo-replication master setup`

Настройка георепликации для главного кластера:

```
usage: vinfra service backup geo-replication master setup --slave-cluster-address
<slave-cluster-address>
--slave-cluster-uid
<slave-cluster-uid>
```

```
--slave-cluster-address <slave-cluster-address>
```

Доменное имя подчиненного кластера

```
--slave-cluster-uid <slave-cluster-uid>
```

UID подчиненного кластера

Пример:

```
# vinfra service backup geo-replication master setup --slave-cluster-address \
slave.example.com --slave-cluster-uid e63a67388deb3c99d044eecbd7b79ad3-1577275849
+-----+-----+
| Field | Value |
+-----+-----+
| task_id | 07df4a57-704e-47de-b681-615ee0c26a21 |
+-----+-----+
```

Эта команда создает задание настройки георепликации для главного резервного кластера.

Результат выполнения задания:

```
# vinfra task show 07df4a57-704e-47de-b681-615ee0c26a21
+-----+-----+-----+-----+
| Field  | Value                                                                 |
+-----+-----+-----+-----+
| details |                                                                 |
| name    | backend.tasks.message_dispatcher.CommandDispatcher |
| result  |                                                                 |
| state   | success                                                                 |
| task_id | 07df4a57-704e-47de-b681-615ee0c26a21 |
+-----+-----+-----+-----+
```

4.6.3 vinfra service backup geo-replication master download-configs

Скачивание файла конфигурации георепликации для главного кластера:

```
usage: vinfra service backup geo-replication master download-configs
       [--conf-file-path <conf-file-path>]
```

`--conf-file-path <conf-file-path>`

Путь, где будет сохранен файл конфигурации

Пример:

```
# vinfra service backup geo-replication master download-configs \
  --conf-file-path master_dc.conf
```

Эта команда скачивает конфигурацию георепликации главного кластера в указанный файл.

4.6.4 vinfra service backup geo-replication slave setup

Настройка георепликации для подчиненного кластера:

```
usage: vinfra service backup geo-replication slave setup
       --dc-config-file <dc-config-file>
```

`--dc-config-file <dc-config-file>`

Путь к файлу конфигурации главного кластера на локальном сервере.

Пример:

```
# vinfra service backup geo-replication slave setup --dc-config-file master_dc.conf
+-----+-----+
| Field   | Value                                     |
+-----+-----+
| task_id | d34b3a4f-6e16-4e60-b20a-844052945d3e |
+-----+-----+
```

Эта команда создает задание настройки георепликации для подчиненного резервного кластера.

Результат выполнения задания:

```
# vinfra task show d34b3a4f-6e16-4e60-b20a-844052945d3e
+-----+-----+
| Field   | Value                                     |
+-----+-----+
| details |                                           |
| name    | backend.tasks.message_dispatcher.CommandDispatcher |
| result  |                                           |
| state   | success                                   |
| task_id | d34b3a4f-6e16-4e60-b20a-844052945d3e |
+-----+-----+
```

4.6.5 vinfra service backup geo-replication master establish

Установка соединения между главным и подчиненным кластерами для включения георепликации:

```
usage: vinfra service backup geo-replication master establish
```

Пример:

```
# vinfra service backup geo-replication master establish
+-----+-----+
| Field   | Value                                     |
+-----+-----+
| task_id | 014903e4-c2e6-4e03-b1af-06c28b672f6e |
+-----+-----+
```

Эта команда создает задание для соединения главного и подчиненного кластеров и включения георепликации.

Результат выполнения задания:

```
# vinfra task show 014903e4-c2e6-4e03-b1af-06c28b672f6e
+-----+-----+
| Field   | Value                                     |
+-----+-----+
| details |                                           |
| name    | backend.tasks.message_dispatcher.CommandDispatcher |
+-----+-----+
```



```

| result | |
| state | success |
| task_id | 014903e4-c2e6-4e03-b1af-06c28b672f6e |
+-----+-----+

```

4.6.6 vinfra service backup geo-replication slave update-certificates

Обновление конфигурации главного кластера на подчиненном кластере:

```
usage: vinfra service backup geo-replication slave update-certificates
      --dc-config-file <dc-config-file>
```

--dc-config-file <dc-config-file>

Путь к файлу конфигурации главного кластера

Пример:

```

# vinfra service backup geo-replication slave update-certificates \
--dc-config-file primary_dc_updated.conf
+-----+-----+
| Field | Value |
+-----+-----+
| task_id | 0ab89de1-b02d-426b-a03c-b1922e610594 |
+-----+-----+

```

Эта команда создает задание для обновления конфигурации главного резервного кластера.

Результат выполнения задания:

```

# vinfra task show 0ab89de1-b02d-426b-a03c-b1922e610594
+-----+-----+
| Field | Value |
+-----+-----+
| details | |
| name | backend.tasks.message_dispatcher.CommandDispatcher |
| result | |
| state | success |
| task_id | 0ab89de1-b02d-426b-a03c-b1922e610594 |
+-----+-----+

```

4.6.7 vinfra service backup geo-replication master disable

Отключение георепликации на главном кластере:

```
usage: vinfra service backup geo-replication master disable
```

Пример:

```
# vinfra service backup geo-replication master disable
+-----+-----+
| Field  | Value |
+-----+-----+
| task_id | dc2bb8ae-8e32-4d37-8d97-4c4c46189d27 |
+-----+-----+
```

Эта команда создает задание для отключения георепликации на главном кластере.

Результат выполнения задания:

```
# vinfra task show dc2bb8ae-8e32-4d37-8d97-4c4c46189d27
+-----+-----+
| Field  | Value |
+-----+-----+
| details |       |
| name    | backend.tasks.message_dispatcher.CommandDispatcher |
| result  |       |
| state   | success |
| task_id | dc2bb8ae-8e32-4d37-8d97-4c4c46189d27 |
+-----+-----+
```

4.6.8 vinfra service backup geo-replication slave promote-to-master

Преобразование подчиненного сервера в главный в конфигурации георепликации:

```
usage: vinfra service backup geo-replication slave promote-to-master
```

Пример:

```
# vinfra service backup geo-replication slave promote-to-master
+-----+-----+
| Field  | Value |
+-----+-----+
| task_id | 083a7d6e-3be8-490f-b468-a3f84abb3487 |
+-----+-----+
```

Эта команда создает задание для преобразования подчиненного сервера в главный в конфигурации георепликации.

Результат выполнения задания:

```
# vinfra task show 083a7d6e-3be8-490f-b468-a3f84abb3487
+-----+-----+
| Field  | Value |
+-----+-----+
| details |       |
+-----+-----+
```

```

| name      | backend.tasks.message_dispatcher.CommandDispatcher |
| result    | success                                             |
| state     | success                                             |
| task_id   | 083a7d6e-3be8-490f-b468-a3f84abb3487             |
+-----+-----+

```

4.6.9 vinfra service backup geo-replication slave cancel

Отмена георепликации для подчиненного кластера:

```
usage: vinfra service backup geo-replication slave cancel
```

Пример:

```

# vinfra service backup geo-replication slave cancel
+-----+-----+
| Field  | Value |
+-----+-----+
| task_id | ad977d03-995c-4677-9308-5e73ec8a2821 |
+-----+-----+

```

Эта команда создает задание отмены георепликации для подчиненного резервного кластера.

Результат выполнения задания:

```

# vinfra task show ad977d03-995c-4677-9308-5e73ec8a2821
+-----+-----+
| Field  | Value |
+-----+-----+
| details |
| name    | backend.tasks.message_dispatcher.CommandDispatcher |
| result  |
| state   | success |
| task_id | ad977d03-995c-4677-9308-5e73ec8a2821 |
+-----+-----+

```

4.6.10 vinfra service backup geo-replication master cancel

Отмена георепликации для главного кластера:

```
usage: vinfra service backup geo-replication master cancel
```

Пример:

```

# vinfra service backup geo-replication master cancel
+-----+-----+

```

```
| Field | Value |
+-----+-----+
| task_id | e1931274-24a5-491e-a5f8-d24fdf4385f7 |
+-----+-----+
```

Эта команда создает задание отмены георепликации для главного резервного кластера.

Результат выполнения задания:

```
# vinfra task show e1931274-24a5-491e-a5f8-d24fdf4385f7
+-----+-----+
| Field | Value |
+-----+-----+
| details |
| name | backend.tasks.message_dispatcher.CommandDispatcher |
| result |
| state | success |
| task_id | e1931274-24a5-491e-a5f8-d24fdf4385f7 |
+-----+-----+
```

ГЛАВА 5

Управление общими параметрами

5.1 Управление лицензиями

5.1.1 `vinfra cluster license load`

Загрузка лицензии из ключа.

```
usage: vinfra cluster license load --key <license-key> --type <license-type>
```

`--key <лицензионный_ключ>`

Регистрируемый лицензионный ключ. Введите этот параметр несколько раз, чтобы зарегистрировать сразу несколько ключей.

`--type <тип_лицензии>`

Тип лицензии (`prolong` — продление или `upgrade` — обновление)

Пример:

```
# vinfra cluster license load --key A38600ML-3P6W746P-RZSK58BV-Y9ZH05Q5-2X7J48J6-KVRXRYPY-V
Z2FK7ZQ6-Y7FGZNYF --type upgrade
+-----+-----+
| Field      | Value                                |
+-----+-----+
| capacity   | 10995116277760                       |
| expiration | 2021-01-10T12:42:00                  |
| free_size  | 10973383165601                       |
| spla       | registered: false                    |
|            | registration_url: null                |
```

```
| status      | active      |
| total_size  | 1099511627760 |
| used_size   | 21733112159  |
+-----+-----+
```

Эта команда устанавливает лицензию из ключа A38600-3P6W74-RZSK58-Y9ZH05-2X7J48.

5.1.2 vinfra cluster license show

Отображение подробных данных по установленной лицензии:

```
usage: vinfra cluster license show
```

Пример:

```
# vinfra cluster license show
+-----+-----+
| Field      | Value      |
+-----+-----+
| capacity   | 1099511627760 |
| expiration | 2021-01-10T12:42:00 |
| free_size  | 10973383165601 |
| spla       | registered: false |
|            | registration_url: null |
| status     | active      |
| total_size | 1099511627760 |
| used_size  | 21733112159  |
+-----+-----+
```

Эта команда отображает подробные данные об установленной в настоящий момент лицензии.

5.2 Управление доменами

5.2.1 vinfra domain create

Создание нового домена:

```
usage: vinfra domain create [--description <description>] [--enable | --disable] <name>
```

--description <описание>

Описание домена

--enable

Включение домена

`--disable`

Отключение домена

<имя>

Имя домена

Пример:

```
# vinfra domain create mydomain
+-----+-----+
| Field          | Value                               |
+-----+-----+
| description    |                                     |
| enabled        | True                                |
| id             | ed408d00561c4a398f933c29e87cadab  |
| name           | domain1                             |
| projects_count | 0                                    |
+-----+-----+
```

Эта команда создает и включает домен `mydomain`.

5.2.2 vinfra domain list

Вывод списка всех доступных доменов:

```
usage: vinfra domain list
```

Пример:

```
# vinfra domain list
+-----+-----+-----+-----+
| id                | name      | enabled | description          |
+-----+-----+-----+-----+
| default           | Default   | True    | The default domain  |
| 24986479ee3246048d3ef2a065ea99f5 | mydomain  | True    |                       |
+-----+-----+-----+-----+
```

Эта команда выводит список доменов, используемых в вычислительном кластере.

5.2.3 vinfra domain show

Вывод сведений о домене:

```
usage: vinfra domain show <domain>
```

<домен>

Идентификатор или имя домена

Пример:

```
# vinfra domain show mydomain
+-----+-----+
| Field          | Value                               |
+-----+-----+
| description    |                                     |
| enabled        | True                                |
| id             | 24986479ee3246048d3ef2a065ea99f5 |
| name           | mydomain                            |
| projects_count | 0                                    |
+-----+-----+
```

Эта команда отображает подробные данные о домене mydomain.

5.2.4 vinfra domain set

Изменение существующего домена:

```
usage: vinfra domain set [--description <description>] [--enable | --disable]
                        [--name <name>] <domain>
```

--description <описание>

Описание домена

--enable

Включение домена

--disable

Отключение домена

--name <имя>

Имя домена

<домен>

Идентификатор или имя домена

Пример:

```
# vinfra domain set mydomain --description "A custom domain"
+-----+-----+
| Field          | Value                               |
+-----+-----+
```



```
| description | A custom domain |
| enabled    | True             |
| id         | 24986479ee3246048d3ef2a065ea99f5 |
| name       | mydomain        |
| projects_count | 0              |
+-----+-----+-----+-----+-----+
```

Эта команда добавляет описание для домена mydomain.

5.2.5 vinfra domain delete

Удаление домена:

```
usage: vinfra domain delete <domain>
```

<домен>

Идентификатор или имя домена

Пример:

```
# vinfra domain delete mydomain
Operation successful
```

Эта команда удаляет домен mydomain.

5.3 Управление пользователями домена

5.3.1 vinfra domain user list-available-roles

Вывод списка доступных ролей пользователей:

```
usage: vinfra domain user list-available-roles
```

Пример:

```
# vinfra domain user list-available-roles
+-----+-----+-----+-----+-----+
| id          | name          | description                                     | scope      |
+-----+-----+-----+-----+-----+
| abgw        | ABGW         | Can create and manage Acronis Backup          | - system  |
|             |              | Gateway.                                     |           |
| admin       | Administrator | Can perform all management operations.       | - system  |
| cluster     | Cluster      | Can create cluster, join nodes to cluster,   | - system  |
```

		and manage (assign and release) disks.	
compute	Compute	Can create and manage compute cluster.	- system
domain_admin	Domain Admin	Can manage users, projects and all resources in a domain.	- domain
image_upload	Image Upload	Can manage compute images.	- domain
iscsi	Block Storage	Can create and manage iSCSI targets, LUNs and CHAP users.	- system
login	Login	Can login in web UI.	[]
network	Network	Can modify network settings and roles.	- system
nfs	NFS	Can create and manage NFS.	- system
project_admin	Project Admin	Can manage virtual objects inside a project.	- project
s3	S3	Can create and manage S3 cluster.	- system
ssh	SSH	Can add and remove SSH keys for cluster nodes access.	- system
updates	Updates	Can install updates.	- system
viewer	Viewer	Viewer role (read only)	- system

Эта команда выводит список всех доступных ролей пользователей.

5.3.2 vinfra domain user create

Создание нового пользователя домена:

```
usage: vinfra domain user create [--email <email>] [--description <description>]
                                [--assign <project> <role>]
                                [--domain-permissions <domain_permissions>]
                                [--system-permissions <system_permissions>]
                                [--enable | --disable] --domain <domain> <name>
```

`--email <эл_почта>`

Адрес электронной почты пользователя

`--description <описание>`

Описание пользователя

`--assign <проект> <роль>`

Назначение пользователя проекту с одним или несколькими наборами разрешений. Укажите этот параметр несколько раз, чтобы назначить пользователя сразу нескольким проектам.

- `<проект>`: идентификатор или имя проекта
- `<роль>`: роль пользователя в проекте (`project_admin` — администратор проекта)

`--domain-permissions <разрешения_домена>`

Разделенный запятыми список разрешений домена. Чтобы просмотреть список доступных

разрешений домена, используйте команду `vinfra domain user list-available-roles | grep domain`.

`--system-permissions <системные_разрешения>`

Разделенный запятыми список системных разрешений. Чтобы просмотреть список доступных системных разрешений, используйте команду `vinfra domain user list-available-roles | grep system`.

`--enable`

Включение пользователя

`--disable`

Отключение пользователя

`--domain <домен>`

Имя или идентификатор домена

`<имя>`

Имя пользователя

Пример:

```
# vinfra domain user create --domain mydomain --name myuser \
--domain-permissions domain_admin
Password:
+-----+-----+
| Field          | Value                               |
+-----+-----+
| assigned_projects | []                                   |
| description      |                                     |
| domain_permissions | - domain_admin                     |
| email           |                                     |
| enabled         | True                                |
| id              | a9c67c6acf1f4df1818fdeeee0b4bd5e  |
| name            | myuser                              |
| role            | domain_admin                        |
| system_permissions | []                                   |
+-----+-----+
```

Эта команда создает и включает новую учетную запись администратора, `myuser`, в домене `mydomain`. Она также устанавливает пароль для этого нового пользователя.

5.3.3 vinfra domain user list

Вывод списка всех пользователей в домене:

```
usage: vinfra domain user list --domain <domain>
```

--domain <домен>

Имя или идентификатор домена

Пример:

```
# vinfra domain user list --domain mydomain -c id -c name -c enabled \
-c domain_permissions -c assigned_projects
+-----+-----+-----+-----+-----+
| id      | name  | enabled | domain_permissions | assigned_projects |
+-----+-----+-----+-----+-----+
| a9c6<...> | myuser | True    | - domain_admin    | []                |
+-----+-----+-----+-----+-----+
```

Эта команда выводит список всех пользователей в домене mydomain. (Выходные данные усекаются, чтобы они поместились на странице.)

5.3.4 vinfra domain user show

Вывод сведений о пользователе домена:

```
usage: vinfra domain user show --domain <domain> <user>
```

--domain <домен>

Идентификатор или имя домена

<пользователь>

Идентификатор или имя пользователя

Пример:

```
# vinfra domain user show myuser --domain mydomain
+-----+-----+
| Field          | Value                                     |
+-----+-----+
| assigned_projects | []                                       |
| description      |                                         |
| domain_permissions | - domain_admin                         |
| email           |                                         |
| enabled          | True                                    |
| id              | a9c67c6acf1f4df1818fdeeee0b4bd5e     |
+-----+-----+
```

```
| name           | myuser           |
| role           | domain_admin    |
| system_permissions | []              |
+-----+-----+
```

Эта команда отображает подробные данные о пользователе myuser из домена mydomain.

5.3.5 vinfra domain user set

Изменение параметров пользователя домена:

```
usage: vinfra domain user set [--password] [--email <email>]
                             [--description <description>]
                             [--assign <project> <role>]
                             [--domain-permissions <domain_permissions>]
                             [--system-permissions <system_permissions>]
                             [--enable | --disable] [--name <name>]
                             --domain <domain> <user>
```

`--password`

Запрос ввода пароля со стандартного входа (stdin)

`--email <эл_почта>`

Адрес электронной почты пользователя

`--description <описание>`

Описание пользователя

`--assign <проект> <роль>`

Назначение пользователя проекту с одним или несколькими наборами разрешений. Укажите этот параметр несколько раз, чтобы назначить пользователя сразу нескольким проектам.

- `<проект>`: идентификатор или имя проекта
- `<роль>`: роль пользователя в проекте (`project_admin` — администратор проекта)

`--domain-permissions <разрешения_домена>`

Разделенный запятыми список разрешений домена. Чтобы просмотреть список доступных разрешений домена, используйте команду `vinfra domain user list-available-roles | grep domain`.

`--system-permissions <системные_разрешения>`

Разделенный запятыми список системных разрешений. Чтобы просмотреть список доступных

системных разрешений, используйте команду `vinfra domain user list-available-roles | grep system`.

`--enable`

Включение пользователя

`--disable`

Отключение пользователя

`--name <имя>`

Имя пользователя

`--domain <домен>`

Имя или идентификатор домена

`<пользователь>`

Идентификатор или имя пользователя

Пример:

```
# vinfra domain user set myuser --domain mydomain \
--assign myproject project_admin
+-----+
| Field          | Value                                     |
+-----+-----+
| assigned_projects | []                                       |
| description      |                                         |
| domain_permissions | - domain_admin                         |
| email           |                                         |
| enabled         | True                                    |
| id              | a9c67c6acf1f4df1818fdeeee0b4bd5e     |
| name            | myuser                                  |
| role            | domain_admin                           |
| system_permissions | []                                       |
+-----+-----+
```

Эта команда назначает пользователя `myuser` из домена `mydomain` в проект `myproject` в качестве администратора проекта.

5.3.6 vinfra domain user delete

Удаление пользователя домена:

```
usage: vinfra domain user delete --domain <domain> <user>
```

```
--domain <домен>
```

Идентификатор или имя домена

```
<пользователь>
```

Идентификатор или имя пользователя

Пример:

```
# vinfra domain user delete myuser --domain mydomain
Operation successful
```

Эта команда удаляет пользователя myuser из домена mydomain.

5.4 Управление проектами домена

5.4.1 vinfra domain project create

Создание нового проекта домена:

```
usage: vinfra domain project create [--description <description>] [--enable | --disable]
      --name <name> --domain <domain>
```

```
--description <описание>
```

Описание проекта

```
--enable
```

Включение проекта

```
--disable
```

Выключение проекта

```
--name <имя>
```

Имя проекта

```
--domain <домен>
```

Имя или идентификатор домена

Пример:

```
# vinfra domain project create --domain mydomain --name myproject \
--description "A custom project"
+-----+-----+
| Field      | Value                                |
+-----+-----+
```

```
+-----+
| description | A custom project |
| domain_id   | 9f7e68938fe946a2a862e360bbe40d98 |
| enabled     | True              |
| id          | d1c4d6198fb940e6b971cf306571ebbd |
| name        | myproject         |
+-----+
```

Эта команда создает и включает проект `myproject` в домене `mydomain`, а также добавляет к нему описание.

5.4.2 `vinfra domain project list`

Вывод списка всех проектов в домене:

```
usage: vinfra domain project list --domain <domain>
```

`--domain <домен>`

Имя или идентификатор домена

Пример:

```
# vinfra domain project list --domain mydomain
+-----+-----+-----+-----+-----+
| id      | name      | enabled | description      | domain_id |
+-----+-----+-----+-----+-----+
| d1c4<...> | myproject | True   | A custom project | 9f7e<...> |
+-----+-----+-----+-----+-----+
```

Эта команда выводит список всех проектов в домене `mydomain`. (Выходные данные усекаются, чтобы они поместились на странице.)

5.4.3 `vinfra domain project show`

Отображение подробных данных о проекте домена

```
usage: vinfra domain project show --domain <domain> <project>
```

`--domain <домен>`

Имя или идентификатор домена

`<проект>`

Идентификатор или имя проекта

Пример:

```
# vinfra domain project show myproject --domain mydomain
+-----+-----+
| Field      | Value                               |
+-----+-----+
| description | A custom project                    |
| domain_id  | 9f7e68938fe946a2a862e360bbe40d98  |
| enabled    | True                                |
| id         | d1c4d6198fb940e6b971cf306571ebbd  |
| members_count | 0                                  |
| name       | myproject                           |
+-----+-----+
```

Эта команда отображает подробные данные о проекте myproject из домена mydomain.

5.4.4 vinfra domain project set

Изменение существующего проекта:

```
usage: vinfra domain project set [--description <description>] [--enable | --disable]
      [--name <name>] --domain <domain> <project>
```

--description <описание>

Описание проекта

--enable

Включение проекта

--disable

Выключение проекта

--name <имя>

Имя проекта

--domain <домен>

Имя или идентификатор домена

<проект>

Идентификатор или имя проекта

Пример:

```
# vinfra cluster domain project set myproject --domain mydomain --disable
+-----+-----+
| Field      | Value                               |
+-----+-----+
```

```
+-----+
| description | A custom project |
| domain_id   | 9f7e68938fe946a2a862e360bbe40d98 |
| enabled     | False              |
| id          | d1c4d6198fb940e6b971cf306571ebbd |
| name        | myproject          |
+-----+
```

Эта команда отключает проект myproject из домена mydomain.

5.4.5 vinfra domain project user list

Вывод списка пользователей проекта:

```
usage: vinfra domain project user list --domain <domain> <project>
```

--domain <домен>

Имя или идентификатор домена

<проект>

Идентификатор или имя проекта

Пример:

```
# vinfra domain project user list myproject --domain mydomain
+-----+-----+-----+-----+
| id          | name   | description | role          |
+-----+-----+-----+-----+
| eb0203e6b8a641d8be5b54b2f3fc9f47 | myuser |              | project_admin |
+-----+-----+-----+-----+
```

Эта команда выводит список пользователей проекта myproject в домене mydomain.

5.4.6 vinfra domain project user remove

Удаление пользователя из проекта:

```
usage: vinfra domain project user remove --user <user> --domain <domain> <project>
```

--user <пользователь>

Имя или идентификатор пользователя

--domain <домен>

Имя или идентификатор домена

<проект>

Идентификатор или имя проекта

Пример:

```
# vinfra domain project user remove myproject --domain mydomain --user myuser
Operation successful
```

Эта команда удаляет пользователя myuser из проекта myproject в домене mydomain.

5.4.7 vinfra domain project delete

Удаление проекта домена:

```
usage: vinfra domain project delete --domain <domain> <project>
```

--domain <домен>

Имя или идентификатор домена

<проект>

Идентификатор или имя проекта

Пример:

```
# vinfra domain project delete myproject --domain mydomain
Operation successful
```

Эта команда удаляет проект myproject из домена mydomain.

5.5 Управление SSH-ключами

5.5.1 vinfra cluster sshkey add

Добавление открытого ключа SSH из файла:

```
usage: vinfra cluster sshkey add <file>
```

<файл>

Файл открытого ключа SSH

Пример:

```
# vinfra cluster sshkey add id_rsa.pub
+-----+-----+
| Field   | Value                                     |
+-----+-----+
| task_id | 100a54ce-0bf5-4bc0-8e46-2e8b952343e6 |
+-----+-----+
```

Эта команда создает задачу для добавления открытого ключа SSH из файла `key.pub` в список доверенных ключей.

Результат выполнения задачи:

```
# vinfra task show 100a54ce-0bf5-4bc0-8e46-2e8b952343e6
+-----+-----+
| Field   | Value                                     |
+-----+-----+
| args    | - admin                                   |
|         | - 1                                       |
| kwargs  | key: ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAQACueW0956J/u5kjWnia7zePChoTMVBtsh1TDN |
|         | shfHWUzzfydi3/4sTrJ++6dtIoS1D3VVHvHBvp456PT5e/eVy7u0SipOPPoDY2vS2IEY+zjT6MYABi6 |
|         | Dbi7CsRL02HcTWzAkooZNLimWPggYaMT10BZ0KAvNB+Ctpkw8JaT5PRve8UVfjxIQIzL6pQ0f0CDeCH |
|         | xK7Srq0vBzT1F9mWkGdTgy+R0JrgGk+v9PvDXZwyeK+qS54uaGmpB6ZRkkMroIk3h+nZ4y/1eQ6m1C8 |
|         | nnaMaNK0twOibrd3MDroMcqkJWTTH/cukD3sB+MjL6nmFlrrAFRU6PBkwysIio6/XHS9jG+TI7NeRAp |
|         | vwIWEKSg6pqaiLUsMi/46KCHzde20zg08Hd0R5d7hNN/80mhD7b+bY9wig+VTMoQFQYSWrIy/qLL95v |
|         | nX0IksNFjffEE/+lMcZXt3j5kqjW70T2/xkqqWoumaM+FEPLNijL18yb29/XJr/cQZX5R9iXSk33DV |
|         | HG7xpHqAtrXbvKY8zI8t23otGT/rSvWRWV/wgPBZVSWtsE99FEMmwmkx/b3KuPhi0jk0IUKcv5UBL+ |
|         | rZRiYgw/fWXP03f6ZSLLJXtW4iW+BQL60qQWUNQ== |
|         | user@example.com                         |
| name    | backend.presentation.nodes.ssh.tasks.CreateSshKeyTask |
| result  | id: 6a2fb834-4bc6-4597-ae74-7cacf96b7c75 |
|         | key: ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAQACueW0956J/u5kjWnia7zePChoTMVBtsh1TDN |
|         | shfHWUzzfydi3/4sTrJ++6dtIoS1D3VVHvHBvp456PT5e/eVy7u0SipOPPoDY2vS2IEY+zjT6MYABi6 |
|         | Dbi7CsRL02HcTWzAkooZNLimWPggYaMT10BZ0KAvNB+Ctpkw8JaT5PRve8UVfjxIQIzL6pQ0f0CDeCH |
|         | xK7Srq0vBzT1F9mWkGdTgy+R0JrgGk+v9PvDXZwyeK+qS54uaGmpB6ZRkkMroIk3h+nZ4y/1eQ6m1C8 |
|         | nnaMaNK0twOibrd3MDroMcqkJWTTH/cukD3sB+MjL6nmFlrrAFRU6PBkwysIio6/XHS9jG+TI7NeRAp |
|         | vwIWEKSg6pqaiLUsMi/46KCHzde20zg08Hd0R5d7hNN/80mhD7b+bY9wig+VTMoQFQYSWrIy/qLL95v |
|         | nX0IksNFjffEE/+lMcZXt3j5kqjW70T2/xkqqWoumaM+FEPLNijL18yb29/XJr/cQZX5R9iXSk33DV |
|         | HG7xpHqAtrXbvKY8zI8t23otGT/rSvWRWV/wgPBZVSWtsE99FEMmwmkx/b3KuPhi0jk0IUKcv5UBL+ |
|         | rZRiYgw/fWXP03f6ZSLLJXtW4iW+BQL60qQWUNQ== |
|         | user@example.com                         |
|         | label: user@example.com                  |
| state   | success                                   |
| task_id | 100a54ce-0bf5-4bc0-8e46-2e8b952343e6 |
+-----+-----+
```

5.5.2 vinfra cluster sshkey list

Отображение списка добавленных открытых ключей SSH:

```
usage: vinfra cluster sshkey list
```

Пример:

```
# vinfra cluster sshkey list
+-----+-----+-----+
| id | key | label |
+-----+-----+-----+
| 8ccf7f1b-6a53-4d74-99ce-c410d51a9921 | ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAQCA
| | QCueW0956J/u5kjWnia7zePChoTMVBtsh1TDN
| | g0skMg5shfHWUzzfydi3/4sTrJ++6dtIoS1D3
| | VVHvHBvp456PT5e/eVy7u0Sip0PPoDY2vS2IE
| | Y+zjT6MYABi6oEYomIIDbi7CsRL02HcTWzAko
| | oZNIimWPggYaMT10BZOKAvNB+Ctpkw8JaT5PR
| | ve8UVfjxIQIzL6pQ0f0CDeCHgDsvwcmxK7Srq
| | 0vBzTlF9mWkGdTgy+R0JrgGk+v9PvDXZwyeK+
| | qS54uaGmpB6ZRkKMroIk3h+nZ4y/1eQ6m1C8A
| | spa0f5nnaMaNK0tw0ibrd3MDroMcqkJWTTH/c
| | ukD3sB+MjL6nmFlrrAfrU6PBkwysIio6/XHS9
| | jG+TI7NeRApkHnwi0vwIWEKSG6ppqaiLUsMi/
| | 46KCHzde20zg08Hd0R5d7hNN/80mhD7b+bY9w
| | ig+VTMoQFQYSWrIy/qLL95ws4amgAQnX0IksN
| | FjffEE/+lMcZXt3j5kqnjW70T2/xkqqWoumaM
| | +FEPLNijL18yb29/XJr/cQZX5R9iXSk33DVjh
| | ln/EyHG7xpHqAtrXbvKY8zI8t23otGT/rSvWR
| | WV/wgPBZVSWtsE99FEMmwmk/b3KuPhi0jK0
| | IUKcv5UBL+NLHw4gQrZriYgw/fWXP03f6ZSLL
| | JXtW4iW+BQL60qQWUNQ==
| | user@example.com
+-----+-----+-----+
```

Эта команда выводит список доверенных ключей SSH.

5.5.3 vinfra cluster sshkey delete

Удаление открытого ключа SSH с серверов кластера хранилища:

```
usage: vinfra cluster sshkey delete <sshkey>
```

<ключ_ssh>

Значение ключа SSH

Пример:

```
# vinfra cluster sshkey delete 8ccf7f1b-6a53-4d74-99ce-c410d51a9921
+-----+-----+
| Field   | Value |
+-----+-----+
| task_id | 053802b2-b4c3-454d-89e2-6d6d312dd2ed |
+-----+-----+
```

Эта команда создает задачу для удаления ключа SSH с идентификатором 8ccf7f1b-6a53-4d74-99ce-c410d51a9921.

Результат выполнения задачи:

```
# vinfra task show 053802b2-b4c3-454d-89e2-6d6d312dd2ed
+-----+-----+
| Field   | Value |
+-----+-----+
| args    | - admin |
|         | - 1     |
|         | - 8ccf7f1b-6a53-4d74-99ce-c410d51a9921 |
| kwargs  | {}      |
| name    | backend.presentation.nodes.ssh.tasks.RemoveSshKeyTask |
| state   | success |
| task_id | 053802b2-b4c3-454d-89e2-6d6d312dd2ed |
+-----+-----+
```

5.6 Управление внешними серверами DNS

5.6.1 vinfra cluster settings dns show

Отображение серверов DNS:

```
usage: vinfra cluster settings dns show
```

Пример:

```
# vinfra cluster settings dns show
+-----+-----+
| Field           | Value |
+-----+-----+
| dhcp_nameservers | 10.10.0.10,10.10.0.11,10.37.130.2 |
| nameservers      | 10.10.0.11,10.10.0.10 |
+-----+-----+
```

Эта команда выводит список используемых в настоящее время DNS-серверов (как внутренних, полученных по DHCP, так и внешних, из заданного пользователем статического набора).

5.6.2 vinfra cluster settings dns set

Установка серверов DNS:

```
usage: vinfra cluster settings dns set --nameservers <nameservers>
```

`--nameservers <сервера_имен>`

Разделенный запятыми список серверов DNS

Пример:

```
# vinfra cluster settings dns set --nameservers 8.8.8.8
+-----+-----+
| Field          | Value          |
+-----+-----+
| dhcp_nameservers | - 10.10.0.10  |
|                 | - 10.10.0.11  |
|                 | - 10.37.130.2 |
| nameservers     | - 8.8.8.8     |
+-----+-----+
```

Эта команда устанавливает внешний сервер DNS 8.8.8.8.

5.7 Настройка высокой доступности сервера управления

5.7.1 vinfra cluster ha create

Создание конфигурации высокой доступности:

```
usage: vinfra cluster ha create --virtual-ip <network:ip> --nodes <nodes> [--force]
```

`--virtual-ip <сеть:ip>`

Сопоставление конфигурации высокой доступности в формате:

- `сеть`: сеть, включаемая в конфигурацию высокой доступности (должна включать по меньшей мере один из следующих типов трафика: Управление системными сервисами, Панель администрирования, Панель самообслуживания или API вычислений).
- `ip`: виртуальный IP-адрес, который будет использоваться в конфигурации высокой доступности.

Укажите этот параметр несколько раз, чтобы создать конфигурацию высокой доступности сразу для нескольких сетей.

```
--nodes <серверы>
```

Разделенный запятыми список идентификаторов или имен серверов

```
--force
```

Пропустить проверки на соответствие минимальным аппаратным требованиям

Пример:

```
# vinfra cluster ha create --virtual-ip Private:10.37.130.200 \
--virtual-ip Public:10.94.129.79 --nodes 94d58604-6f30-4339-8578-adb7903b7277, \
f59dabdb-bd1c-4944-8af2-26b8fe9ff8d4,7d7d37b8-4c06-4f1a-b3a6-4b54257d70ce
+-----+-----+
| Field  | Value                                |
+-----+-----+
| task_id | 80a00e55-335d-4d41-bac4-5fee4791d423 |
+-----+-----+
```

Эта команда создает задачу для создания кластера высокой доступности сервера управления из серверов с идентификаторами 94d58604-6f30-4339-8578-adb7903b7277, f59dabdb-bd1c-4944-8af2-26b8fe9ff8d4 и 7d7d37b8-4c06-4f1a-b3a6-4b54257d70ce.

В команде должна быть указана сеть с типом трафика Управление системными сервисами, а также сеть с типом трафика Панель администрирования.

Важно: После создания кластера высокой доступности панель администрирования будет доступна только по заданному внешнему IP-адресу. Выполните вход на этот адрес по протоколу SSH, чтобы продолжить управление Acronis Инфраструктура с помощью инструмента командной строки `vinfra`. Также может потребоваться снова установить переменную среды `VINFRA_PASSWORD`, поскольку теперь вы при каждом входе в систему будете осуществлять доступ к различным серверам кластера высокой доступности, а на некоторых из них она может быть не установлена.

Результат выполнения задачи:

```
# vinfra task show 80a00e55-335d-4d41-bac4-5fee4791d423
+-----+-----+
| Field  | Value                                |
+-----+-----+
| details |                                       |
| name    | backend.presentation.ha.tasks.CreateHaConfigTask |
| result  | compute_task_id: c5125024-5472-4420-b8b6-e03971ab952c |
+-----+-----+
```



```

|         | ha_cluster_location: |
|         | - https://10.94.129.79:8888 |
|         | nodes: |
|         | - id: 94d58604-6f30-4339-8578-adb7903b7277 |
|         |   ipaddr: 10.37.130.118 |
|         |   is_primary: false |
|         | - id: f59dabdb-bd1c-4944-8af2-26b8fe9ff8d4 |
|         |   ipaddr: 10.37.130.134 |
|         |   is_primary: true |
|         | - id: 7d7d37b8-4c06-4f1a-b3a6-4b54257d70ce |
|         |   ipaddr: 10.37.130.246 |
|         |   is_primary: false |
|         | primary_node_location: https://10.94.62.243:8888 |
|         | virtual_ips: |
|         | - ip: 10.94.129.79 |
|         |   roles_set: 5f0adc1d-c10f-46c1-b7b8-dd1aacab613b |
|         | - ip: 10.37.130.200 |
|         |   roles_set: 5a0401b5-9b42-4d8b-8372-71c747230033 |
| state   | success |
| task_id | 80a00e55-335d-4d41-bac4-5fee4791d423 |
+-----+

```

5.7.2 vinfra cluster ha update

Обновление конфигурации высокой доступности:

```
usage: vinfra cluster ha update [--virtual-ip <network:ip>]
                                [--nodes <nodes>] [--force]
```

`--virtual-ip <сеть:ip>`

Сопоставление конфигурации высокой доступности в формате:

- `сеть`: сеть, включаемая в конфигурацию высокой доступности (должна включать по меньшей мере один из следующих типов трафика: **Управление системными сервисами, Панель администрирования, Панель самообслуживания** или **API вычислений**).
- `ip`: виртуальный IP-адрес, который будет использоваться в конфигурации высокой доступности.

Укажите этот параметр несколько раз, чтобы создать конфигурацию высокой доступности сразу для нескольких сетей.

`--nodes <серверы>`

Разделенный запятыми список идентификаторов или имен серверов

```
--force
```

Пропустить проверки на соответствие минимальным аппаратным требованиям

Пример:

```
# vinfra cluster ha update --nodes 94d58604-6f30-4339-8578-adb7903b7277,\
f59dabdb-bd1c-4944-8af2-26b8fe9ff8d4,4b83a87d-9adf-472c-91f0-782c47b2d5f1
+-----+
| Field | Value |
+-----+
| task_id | 565e9146-254b-4f7a-a2ff-b7119c95baa9 |
+-----+
```

Эта команда создает задачу для обновления конфигурации высокой доступности сервера управления, т. е. включения в нее серверов с идентификаторами 94d58604-6f30-4339-8578-adb7903b7277, f59dabdb-bd1c-4944-8af2-26b8fe9ff8d4 и 4b83a87d-9adf-472c-91f0-782c47b2d5f1.

Результат выполнения задачи:

```
# vinfra task show 565e9146-254b-4f7a-a2ff-b7119c95baa9
+-----+
| Field | Value |
+-----+
| details | |
| name | backend.presentation.ha.tasks.UpdateHaConfigTask |
| result | compute_task_id: 84994caf-3a02-43ea-b904-48632f0379c7 |
| | ha_cluster_location: |
| | - https://10.94.129.79:8888 |
| | nodes: |
| | - id: f59dabdb-bd1c-4944-8af2-26b8fe9ff8d4 |
| | ipaddr: 10.37.130.134 |
| | is_primary: true |
| | - id: 4b83a87d-9adf-472c-91f0-782c47b2d5f1 |
| | ipaddr: 10.37.130.127 |
| | is_primary: false |
| | - id: 94d58604-6f30-4339-8578-adb7903b7277 |
| | ipaddr: 10.37.130.118 |
| | is_primary: false |
| | primary_node_location: https://10.94.62.243:8888 |
| | virtual_ips: |
| | - ip: 10.94.129.79 |
| | roles_set: 5f0adc1d-c10f-46c1-b7b8-dd1aacab613b |
| | - ip: 10.37.130.200 |
| | roles_set: 5a0401b5-9b42-4d8b-8372-71c747230033 |
| state | success |
| task_id | 565e9146-254b-4f7a-a2ff-b7119c95baa9 |
+-----+
```

5.7.3 vinfra cluster ha show

Отображение конфигурации высокой доступности:

```
usage: vinfra cluster ha show
```

Пример:

```
# vinfra cluster ha show
+-----+-----+
| Field          | Value                                     |
+-----+-----+
| ha_cluster_location | - https://10.94.129.79:8888              |
| nodes           | - id: 94d58604-6f30-4339-8578-adb7903b7277 |
|                 | ipaddr: 10.37.130.118                   |
|                 | is_primary: false                       |
|                 | - id: f59dabdb-bd1c-4944-8af2-26b8fe9ff8d4 |
|                 | ipaddr: 10.37.130.134                   |
|                 | is_primary: true                        |
|                 | - id: 4b83a87d-9adf-472c-91f0-782c47b2d5f1 |
|                 | ipaddr: 10.37.130.127                   |
|                 | is_primary: false                       |
| primary_node_location | https://10.94.62.243:8888              |
| virtual_ips       | - ip: 10.37.130.200                     |
|                   | roles_set: 5a0401b5-9b42-4d8b-8372-71c747230033 |
|                   | - ip: 10.94.129.79                      |
|                   | roles_set: 5f0adc1d-c10f-46c1-b7b8-dd1aacab613b |
+-----+-----+
```

Эта команда отображает конфигурацию кластера высокой доступности сервера управления.

5.7.4 vinfra cluster ha delete

Удаление конфигурации высокой доступности:

```
usage: vinfra cluster ha delete
```

Пример:

```
# vinfra cluster ha delete
+-----+-----+
| Field  | Value                                     |
+-----+-----+
| task_id | c1f3e9c3-0a7b-455a-96d4-cef3b7e86e62 |
+-----+-----+
```

Эта команда создает задачу для удаления кластера высокой доступности сервера управления.

Результат выполнения задачи:

```
# vinfra task show c1f3e9c3-0a7b-455a-96d4-cef3b7e86e62
+-----+-----+
| Field  | Value                                     |
+-----+-----+
| details |                                           |
| name    | backend.presentation.ha.tasks.DeleteHaConfigTask |
| result  |                                           |
| state   | success                                   |
| task_id | c1f3e9c3-0a7b-455a-96d4-cef3b7e86e62       |
+-----+-----+
```

5.8 Управление резервными копиями кластера

5.8.1 vinfra cluster backup create

Создание резервной копии:

```
usage: vinfra cluster backup create
```

Пример:

```
# vinfra cluster backup create
+-----+-----+
| Field  | Value                                     |
+-----+-----+
| task_id | e4b4f891-cc3a-4308-a321-d76265ef7b5b |
+-----+-----+
```

Эта команда создает задачу для резервного копирования кластера хранилища данных.

Результат выполнения задачи:

```
# vinfra task show e4b4f891-cc3a-4308-a321-d76265ef7b5b
+-----+-----+
| Field  | Value                                     |
+-----+-----+
| details |                                           |
| name    | backend.presentation.backups.tasks.BackupManagedNodeTask |
| result  | status: finished                         |
| state   | success                                   |
| task_id | e4b4f891-cc3a-4308-a321-d76265ef7b5b       |
+-----+-----+
```

5.8.2 vinfra cluster backup show

Отображение информации о резервной копии:

```
usage: vinfra cluster backup show
```

Пример:

```
# vinfra cluster backup show
+-----+-----+
| Field | Value |
+-----+-----+
| last_backup_date | 2019-08-21T15:41:24+00:00 |
| last_backup_location | /mnt/vstorage/webcp/backup/ |
| ready | True |
| tasks | [] |
+-----+-----+
```

Эта команда выводит подробные данные о последней резервной копии кластера, а также идентификатор выполняемой в настоящее время задачи резервного копирования (если она имеется).

5.9 Управление шифрованием уровней хранилища

5.9.1 vinfra cluster settings encryption show

Отображение данных о шифровании уровней хранилища:

```
usage: vinfra cluster settings encryption show
```

Пример:

```
# vinfra cluster settings encryption show
+-----+-----+
| Field | Value |
+-----+-----+
| tier0 | False |
| tier1 | False |
| tier2 | False |
| tier3 | False |
+-----+-----+
```

Эта команда отображает состояние шифрования каждого из уровней хранилища.

5.9.2 vinfra cluster settings encryption set

Установка шифрования для уровня хранилища:

```
usage: vinfra cluster settings encryption set [--tier-enable {0,1,2,3}]
                                           [--tier-disable {0,1,2,3}]
```

`--tier-enable {0,1,2,3}`

Включение шифрования для определенных уровней хранилища. Этот параметр можно указывать несколько раз.

`--tier-disable {0,1,2,3}`

Отключение шифрования для определенных уровней хранилища. Этот параметр можно указывать несколько раз.

Пример:

```
# vinfra cluster settings encryption set --tier-enable 2
+-----+-----+
| Field | Value |
+-----+-----+
| tier0  | False |
| tier1  | False |
| tier2  | True  |
| tier3  | False |
+-----+-----+
```

Эта команда включает шифрование для уровня хранилища 2.

5.10 Управление автоматической конфигурацией дисков хранилища

5.10.1 vinfra cluster settings automatic-disk-replacement show

Отображение автоматической конфигурации дисков хранилища:

```
usage: vinfra cluster settings automatic-disk-replacement show
```

Пример:

```
# vinfra cluster settings automatic-disk-replacement show
+-----+-----+
```

```
| Field | Value |
+-----+-----+
| tier0 | True  |
| tier1 | False |
| tier2 | True  |
| tier3 | False |
+-----+-----+
```

Эта команда отображает статус автоматической конфигурации дисков для каждого из уровней хранилища.

5.10.2 vinfra cluster settings automatic-disk-replacement set

Изменение настройки автоматической конфигурации дисков хранилища:

```
usage: vinfra cluster settings automatic-disk-replacement set
       [--tier0 {on,off}] [--tier1 {on,off}] [--tier2 {on,off}]
       [--tier3 {on,off}]
```

`--tier0 {on,off}`

Включение (on) или выключение (off) автоматической конфигурации дисков хранилища для уровня 0

`--tier1 {on,off}`

Включение (on) или выключение (off) автоматической конфигурации дисков хранилища для уровня 1

`--tier2 {on,off}`

Включение (on) или выключение (off) автоматической конфигурации дисков хранилища для уровня 2

`--tier3 {on,off}`

Включение (on) или выключение (off) автоматической конфигурации дисков хранилища для уровня 3

Пример:

```
# vinfra cluster settings automatic-disk-replacement set \
--tier0 on --tier1 on --tier2 on --tier3 on
+-----+-----+
| Field | Value |
+-----+-----+
| tier0 | True  |
| tier1 | True  |
```

```
| tier2 | True |
| tier3 | True |
+-----+-----+
```

Эта команда включает автоматическую конфигурацию дисков хранилища для всех уровней хранилища.

5.11 Управление оповещениями

5.11.1 vinfra cluster alert list

Вывод списка записей в журнале оповещений:

```
usage: vinfra cluster alert list [--all]
```

--all

Показывать как включенные, так и отключенные оповещения

Пример:

```
# vinfra cluster alert list --all
+-----+-----+-----+-----+-----+-----+
| id | type           | datetime           | severity | enabled |
+-----+-----+-----+-----+-----+-----+
| 1 | Network warning | 2018-08-30T18:02:14 | warning  | True    |
| 2 | Network warning | 2018-08-30T18:02:14 | warning  | True    |
| 3 | Network warning | 2018-08-30T18:02:14 | warning  | True    |
| 4 | Network warning | 2018-08-31T13:02:15 | warning  | True    |
| 5 | Network warning | 2018-08-31T13:02:15 | warning  | True    |
| 6 | Network warning | 2018-08-31T13:02:15 | warning  | True    |
| 7 | Network warning | 2018-08-31T13:02:15 | warning  | True    |
| 8 | Network warning | 2018-08-31T13:02:15 | warning  | True    |
| 9 | Network warning | 2018-08-31T13:02:15 | warning  | True    |
| 10 | Network warning | 2018-08-31T13:02:15 | warning  | True    |
| 11 | Network warning | 2018-08-31T13:02:15 | warning  | True    |
| 12 | Network warning | 2018-08-31T13:02:15 | warning  | True    |
| 13 | Network warning | 2018-08-31T13:02:15 | warning  | True    |
| 14 | Network warning | 2018-08-31T13:02:15 | warning  | True    |
| 15 | Network warning | 2018-08-31T13:02:15 | warning  | True    |
+-----+-----+-----+-----+-----+-----+
```

Эта команда выводит список всех оповещений в журнале с указанием того, включены они или отключены.

5.11.2 vinfra cluster alert show

Вывод подробных данных по указанной записи журнала оповещений:

```
usage: vinfra cluster alert show <alert>
```

<оповещение>

Идентификатор оповещения

Пример:

```
# vinfra cluster alert show 1
+-----+-----+
| Field      | Value                                     |
+-----+-----+
| _type      | undefined_speed                         |
| cluster_id |                                           |
| cluster_name |                                           |
| datetime   | 2018-08-30T18:02:14.855302+00:00       |
| details    | host: stor-1.example.com.vstoragedomain. |
| enabled    | True                                     |
| group      | node                                     |
| host       | stor-1.example.com.vstoragedomain.     |
| id         | 1                                       |
| message    | Network interface "eth1" on node "stor-1.example.com.vstoragedomain." has |
|            | an undefined speed                       |
| node_id    | 4f96acf5-3bc8-4094-bcb6-4d1953be7b55  |
| object_id  | eth1                                     |
| severity   | warning                                  |
| suspended  |                                           |
| type       | Network warning                         |
+-----+-----+
```

Эта команда выводит подробные данные об оповещении с идентификатором 1.

5.11.3 vinfra cluster alert delete

Удаление записи из журнала оповещений:

```
usage: vinfra cluster alert delete <alert>
```

<оповещение>

Идентификатор оповещения

Пример:

```
# vinfra cluster alert delete 1
+-----+-----+
| Field      | Value
+-----+-----+
| _type      | undefined_speed
| cluster_id |
| cluster_name |
| datetime   | 2018-08-30T18:02:14.855302+00:00
| details    | host: stor-1.example.com.vstoragedomain.
| enabled    | True
| group      | node
| host       | stor-1.example.com.vstoragedomain.
| id         | 1
| message    | Network interface "eth1" on node "stor-1.example.com.vstoragedomain." has
|            | undefined speed
| node_id    | 4f96acf5-3bc8-4094-bcb6-4d1953be7b55
| object_id  | eth1
| severity   | warning
| suspended  |
| type       | Network warning
+-----+-----+
```

Эта команда удаляет оповещение с идентификатором 1 из журнала.

5.12 Управление журналом аудита

5.12.1 vinfra cluster auditlog list

Вывод списка всех записей в журнале аудита:

```
usage: vinfra cluster auditlog list
```

Пример:

```
# vinfra cluster auditlog list
+-----+-----+-----+-----+-----+
| id | username | type                | activity                | timestamp
+-----+-----+-----+-----+-----+
| 1  | admin   | LoginUser           | User login              | 2018-09-07T08:33:44
| 2  | admin   | ChangeNetwrokInterface | Configure network       | 2018-09-07T09:53:58
| 3  | admin   | UpInterface         | Bring up interface      | 2018-09-07T09:54:44
| 4  | admin   | ChangeNetwrokInterface | Configure network       | 2018-09-07T09:54:54
| 5  | admin   | CreateBonding        | Create bonding          | 2018-09-07T09:57:24
| 17 | admin   | RemoveNode           | Forget node             | 2018-09-07T12:21:59
| 14 | admin   | RemoveNetworkIface  | Delete interface       | 2018-09-07T12:17:14
| 15 | admin   | RemoveNode           | Forget node             | 2018-09-07T12:17:49
| 6  | admin   | UpInterface         | Bring up interface      | 2018-09-07T10:59:28
```

7	admin	ChangeNetwrokInterface	Configure network	2018-09-07T10:59:46
9	admin	UpInterface	Bring up interface	2018-09-07T11:42:29
10	admin	UpInterface	Bring up interface	2018-09-07T11:42:42
11	admin	CreateBonding	Create bonding	2018-09-07T11:43:46
12	admin	ChangeNetwrokInterface	Configure network	2018-09-07T11:52:17
13	admin	ChangeNetwrokInterface	Configure network	2018-09-07T11:52:44
16	admin	RemoveNode	Forget node	2018-09-07T12:21:51
8	admin	CreateBonding	Create bonding	2018-09-07T11:00:39
18	admin	RemoveNode	Forget node	2018-09-07T12:22:08
19	admin	UpInterface	Bring up interface	2018-09-07T12:33:16
20	admin	CreateVLAN	Create VLAN	2018-09-07T12:34:18
21	admin	RemoveNetworkIface	Delete interface	2018-09-07T13:26:40
22	admin	LoginUser	User login	2018-09-07T14:50:06
23	admin	LoginUser	User login	2018-09-07T14:51:34
24	admin	CreateNetworkRolesSet	Create custom role set	2018-09-07T15:06:03
25	admin	ChangeNetworkRolesSet	Configure custom role set	2018-09-07T15:37:50
26	admin	RemoveNetworkRolesSet	Remove custom role set	2018-09-07T15:39:31
27	admin	CreateNetworkRole	Create custom role	2018-09-07T15:58:50
28	admin	RemoveNetworkRole	Remove custom role	2018-09-07T16:20:22

Эта команда выводит список записей в журнале аудита.

5.12.2 vinfra cluster auditlog show

Вывод подробных данных по записи журнала аудита:

```
usage: vinfra cluster auditlog show <auditlog>
```

<журнал_аудита>

Идентификатор журнала аудита

Пример:

```
# vinfra cluster auditlog show 1
+-----+-----+
| Field      | Value                                |
+-----+-----+
| activity   | User login                           |
| cluster_id |                                       |
| cluster_name |                                       |
| component  | Users                                 |
| details    | []                                    |
| id         | 1                                     |
| message    | User "admin" login                   |
| node_id    |                                       |
| result     | success                               |
| session_id | 817a19beaf244f92604fbf4b40af2c29    |
| task_id    | 5686556295049300                    |
```

```
| timestamp | 2018-09-07T08:33:44.175797+00:00 |
| type     | LoginUser                       |
| username | admin                           |
+-----+-----+
```

Эта команда выводит подробные данные о записи журнала аудита с идентификатором 1.

5.13 Отправка отчетов о неполадках

Создание и отправка отчета о неполадке:

```
usage: vinfra cluster problem-report [--email <email>]
                                     [--description <description>] [--send]
```

`--email <эл_почта>`

Контактный адрес электронной почты

`--description <описание>`

Описание неполадки

`--send`

Создать архив с отчетом о неполадке и отправить его в службу технической поддержки

Пример:

```
# vinfra cluster problem-report --email test@example.com --description "Test report" --send
+-----+-----+
| Field  | Value |
+-----+-----+
| task_id | 8bcfb92f-f02b-4de8-8e44-3426047630e3 |
+-----+-----+
```

Эта команда создает задачу для отправки отчета о неполадке с описанием «Test report» в службу технической поддержки и указания в качестве ответного адреса для связи адреса: test@example.com. Укажите идентификатор отчета о неполадке в сведениях о задаче. Его нужно будет ввести в заявке на поддержку.

Результат выполнения задачи:

```
+-----+-----+
| Field  | Value |
+-----+-----+
| details |       |
| name    | backend.presentation.reports.tasks.ReportProblemTask |
| result  | id: '1001923113' |
+-----+-----+
```

```
|          | path: /var/cache/problem-reports/report-2018-12-10T15:33:23.391329.tar.gz |
| state   | success                                                                    |
| task_id | 37d5c13a-001c-4789-8242-96825a17deda                                       |
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
```

ГЛАВА 6

Мониторинг кластера хранилища

Мониторинг кластера хранилища данных очень важен, поскольку он позволяет проверить состояние и работоспособность всех компьютеров в кластере и соответствующим образом отреагировать.

Основная команда мониторинга — это `vstorage -c <имя_кластера> top`. Она вызывает текстовый пользовательский интерфейс, которым можно управлять с клавиатуры (чтобы получить справку, нажмите клавишу **h**).

6.1 Мониторинг общих параметров кластера хранилища данных

Отслеживая общие параметры, можно получать подробную информацию обо всех компонентах кластера хранения данных, а также его общем состоянии и работоспособности. Для вывода этой информации используйте команду `vstorage -c <имя_кластера> top`. Например:

```
Cluster 'stor1': healthy
Space: [OK] allocatable 1.32TB of 1.44TB, free 1.39TB of 1.44TB
MDS nodes: 3 of 3, epoch uptime: 19d 23h
CS nodes: 3 of 3 (3 avail, 0 inactive, 0 offline)
License: ACTIVE (expiration: 01/10/2021, capacity: 10TB, used: 20.3GB)
Replication: 1 norm, 1 limit
IO:      read  0B/s ( 0ops/s), write  0B/s ( 0ops/s)
```

MDSID	STATUS	%TIME	COMMITTS	%CPU	MEM	UPTIME	HOST
M 3	avail	0.0%	0/s	1.1%	192m	19d 23h	management.655c19da7e854d6f.nodes.svc.vstoragedomain:2510
1	avail	0.0%	0/s	0.2%	192m	20d 0h	management.b2823b72aeff4ddb.nodes.svc.vstoragedomain:2510
2	avail	0.0%	0/s	0.0%	192m	19d 23h	management.bda1f22b3a854b6c.nodes.svc.vstoragedomain:2510

CSID	STATUS	SPACE	AVAIL	REPLICAS	UNIQUE	IOWAIT	IOLAT (ms)	QDEPTH	HOST
1027	active	492.0G	451.4G	295	12	0%	0/0	0.0	management.655c19da7e854d6f.nodes.svc.v
1025	active	492.0G	449.5G	305	22	0%	0/0	0.0	management.b2823b72aeff4ddb.nodes.svc.v
1026	active	492.0G	453.0G	289	6	0%	0/0	0.0	management.bda1f22b3a854b6c.nodes.svc.v

CLID	LEASES	READ	WRITE	RD OPS	WR OPS	FSYNCS	IOLAT (ms)	HOST
2050	1/222	6B/s	6B/s	0ops/s	0ops/s	0ops/s	0.13/1	management.b2823b72aeff4ddb.nodes.svc.v
2226	1/2	0B/s	0B/s	0ops/s	0ops/s	0ops/s	0/0	management.bda1f22b3a854b6c.nodes.svc.v
2142	0/0	0B/s	0B/s	0ops/s	0ops/s	0ops/s	0/0	management.655c19da7e854d6f.nodes.svc.v

TIME	SYS	SEV	MESSAGE
21-12-18 12:06:24	MDS	INF	Add new MDS#3 at 10.37.130.79:2510 by request from 10.37.130.79:45672
21-12-18 12:06:24	MON	INF	MDS#3 was started
21-12-18 12:06:35	MON	INF	MDS#3 was stopped
21-12-18 12:06:35	MON	INF	CS#1027 was started
21-12-18 12:06:36	MDS	INF	New CS#1027 at 10.37.130.79:45742 (0.0.0.655c19da7e854d6f), tier=0
21-12-18 12:06:36	MON	INF	MDS#3 was started
21-12-18 12:06:38	MDS	INF	CS#1027 is active
21-12-18 12:06:45	MDS	INF	The cluster physical free space: 1.4Tb (99%), total 1.4Tb

Приведенная выше команда выводит подробную информацию о кластере stor1. Общие параметры (выделены красным) следующие:

Cluster (Кластер)

Общее состояние кластера:

Исправен

Все серверы фрагментов данных в этом кластере активны.

Unknown (Неизвестно)

Недостаточно информации о состоянии этого кластера (например, потому, что главный сервер MDS был выбран уже некоторое время назад).

Деградировал

Часть серверов фрагментов данных в кластере неактивна.

Failure (Отказ)

В кластере слишком много неактивных серверов фрагментов данных; автоматическая репликация отключена.

SMART warning (Предупреждение SMART)

У одного или нескольких физических дисков, подключенных к серверам кластера,

близится аппаратный отказ. Подробные сведения см. в *Мониторинг физических дисков* (страница 215).

Space (Объем)

Количество дискового пространства в кластере:

Свободно

Свободное дисковое пространство в кластере.

Allocatable (Доступно для распределения)

Объем логического дискового пространства, доступного для клиентов. Доступное для выделения дисковое пространство рассчитывается на основе текущих параметров репликации и объема свободного дискового пространства на серверах фрагментов данных. Он также может ограничиваться лицензией.

Примечание: Дополнительные сведения о мониторинге, а также описание использования дискового пространства в кластерах см. в разделе *Объяснение данных об использовании дискового пространства* (страница 208).

MDS nodes (Серверы MDS)

Число активных серверов MDS по сравнению с общим числом серверов MDS, настроенных для кластера.

Epoch time (Время эпохи)

Время, прошедшее с момента выбора главного сервера MDS.

CS nodes (Серверы фрагментов данных)

Число активных серверов фрагментов данных по сравнению с общим числом серверов фрагментов данных, настроенных для кластера.

В скобках отображается дополнительная информация об этих серверах фрагментов данных.

- Активные серверы фрагментов данных (avail.), которые в настоящее время запущены и работают на сервере.
- Неактивные серверы фрагментов данных (inactive), которые в настоящее время недоступны. Сервер фрагментов данных помечается как неактивный в течение первых 5 минут его неактивности.

- Отключенные серверы фрагментов данных (offline), которые были неактивны в течение более 5 минут. Сервер фрагментов данных меняет свое состояние на отключенный, если он неактивен больше 5 минут. После изменения состояния сервера на отключенный кластер начинает репликацию данных, чтобы восстановить фрагменты, которые хранились на отключенном сервере фрагментов данных.

License (Лицензия)

Номер ключа, под которым лицензия зарегистрирована на сервере аутентификации с ключом, и состояние лицензии.

Replication (Репликация)

Параметры репликации. Нормальное число реплик фрагментов и предельное число, ниже которого фрагмент блокируется до момента, когда он будет восстановлен.

IO (Ввод-вывод)

Активность дискового ввода-вывода в кластере:

- Скорость операций ввода-вывода при чтении и записи в байтах в секунду.
- Число операций ввода-вывода (чтения и записи) в секунду.

6.2 Мониторинг серверов метаданных

Серверы MDS (метаданных) являются критически важным компонентом любого кластера хранения данных, поэтому контроль состояния и работоспособности серверов MDS — это крайне важная задача. Для мониторинга серверов MDS используйте команду `vstorage -с <имя_кластера> top`. Например:

```
Cluster 'stor1': healthy
Space: [OK] allocatable 1.32TB of 1.44TB, free 1.39TB of 1.44TB
MDS nodes: 3 of 3, epoch uptime: 19d 23h
CS nodes: 3 of 3 (3 avail, 0 inactive, 0 offline)
License: ACTIVE (expiration: 01/10/2021, capacity: 10TB, used: 20.3GB)
Replication: 1 norm, 1 limit
IO:      read    0B/s ( 0ops/s), write    0B/s ( 0ops/s)
```

MDSID	STATUS	%CTIME	COMMITTS	%CPU	MEM	UPTIME	HOST
M 3	avail	0.0%	0/s	1.1%	192m	19d 23h	management.655c19da7e854d6f.nodes.svc.vstoragedomain:2510
1	avail	0.0%	0/s	0.2%	192m	20d 0h	management.b2823b72aeff4ddb.nodes.svc.vstoragedomain:2510
2	avail	0.0%	0/s	0.0%	192m	19d 23h	management.bda1f22b3a854b6c.nodes.svc.vstoragedomain:2510

CSID	STATUS	SPACE	AVAIL	REPLICAS	UNIQUE	IOWAIT	IOLAT (ms)	QDEPTH	HOST
1027	active	492.0G	451.4G	295	12	0%	0/0	0.0	management.655c19da7e854d6f.nodes.svc.v
1025	active	492.0G	449.5G	305	22	0%	0/0	0.0	management.b2823b72aeff4ddb.nodes.svc.v
1026	active	492.0G	453.0G	289	6	0%	0/0	0.0	management.bda1f22b3a854b6c.nodes.svc.v

CLID	LEASES	READ	WRITE	RD OPS	WR OPS	FSYNCS	IOLAT (ms)	HOST
2050	1/222	6B/s	6B/s	0ops/s	0ops/s	0ops/s	0.13/1	management.b2823b72aeff4ddb.nodes.
2226	1/2	0B/s	0B/s	0ops/s	0ops/s	0ops/s	0/0	management.bda1f22b3a854b6c.nodes.
2142	0/0	0B/s	0B/s	0ops/s	0ops/s	0ops/s	0/0	management.655c19da7e854d6f.nodes.

TIME	SYS	SEV	MESSAGE
21-12-18 12:06:24	MDS	INF	Add new MDS#3 at 10.37.130.79:2510 by request from 10.37.130.79:45672
21-12-18 12:06:24	MON	INF	MDS#3 was started
21-12-18 12:06:35	MON	INF	MDS#3 was stopped
21-12-18 12:06:35	MON	INF	CS#1027 was started
21-12-18 12:06:36	MDS	INF	New CS#1027 at 10.37.130.79:45742 (0.0.0.655c19da7e854d6f), tier=0
21-12-18 12:06:36	MON	INF	MDS#3 was started
21-12-18 12:06:38	MDS	INF	CS#1027 is active
21-12-18 12:06:45	MDS	INF	The cluster physical free space: 1.4Tb (99%), total 1.4Tb

Приведенная выше команда выводит подробную информацию о кластере stor1. Параметры мониторинга для серверов MDS (выделены красным) следующие:

MDSID

Идентификатор (ID) сервера MDS.

Если перед идентификатором добавлена буква M, она указывает, что этот сервер является главным (master) сервером MDS.

STATUS

Статус сервера MDS.

%CTIME

Суммарное время, в течение которого сервер MDS осуществлял запись в локальный журнал.

COMMITTS

Интенсивность фиксации в локальный журнал.

%CPU

Время активности сервера MDS.

MEM

Объем физической памяти, используемой сервером MDS.

UPTIME

Время, прошедшее с момента последнего запуска сервера MDS.

HOST

Имя хоста или IP-адрес сервера MDS.

6.3 Мониторинг серверов фрагментов данных

Отслеживая серверы фрагментов данных, можно контролировать объем доступного в кластере хранения данных пространства. Для отслеживания серверов фрагментов данных используйте команду `vstorage -c <имя_кластера> top`. Например:

```
Cluster 'stor1': healthy
Space: [OK] allocatable 1.32TB of 1.44TB, free 1.39TB of 1.44TB
MDS nodes: 3 of 3, epoch uptime: 19d 23h
CS nodes: 3 of 3 (3 avail, 0 inactive, 0 offline)
License: ACTIVE (expiration: 01/10/2021, capacity: 10TB, used: 20.3GB)
Replication: 1 norm, 1 limit
IO:      read    0B/s ( 0ops/s), write    0B/s ( 0ops/s)
```

MDSID	STATUS	%CTIME	COMMITTS	%CPU	MEM	UPTIME	HOST
M 3	avail	0.0%	0/s	1.1%	192m	19d 23h	management.655c19da7e854d6f.nodes.svc.vstoragedomain:2510
1	avail	0.0%	0/s	0.2%	192m	20d 0h	management.b2823b72aeff4ddb.nodes.svc.vstoragedomain:2510
2	avail	0.0%	0/s	0.0%	192m	19d 23h	management.bd1f22b3a854b6c.nodes.svc.vstoragedomain:2510

CSID	STATUS	SPACE	AVAIL	REPLICAS	UNIQUE	IOWAIT	IOLAT(ms)	QDEPTH	HOST
1027	active	492.0G	451.4G	295	12	0%	0/0	0.0	management.655c19da7e854d6f.nodes.svc.v
1025	active	492.0G	449.5G	305	22	0%	0/0	0.0	management.b2823b72aeff4ddb.nodes.svc.v
1026	active	492.0G	453.0G	289	6	0%	0/0	0.0	management.bd1f22b3a854b6c.nodes.svc.v

CLID	LEASES	READ	WRITE	RD OPS	WR OPS	FSYNCS	IOLAT(ms)	HOST
2050	1/222	6B/s	6B/s	0ops/s	0ops/s	0ops/s	0.13/1	management.b2823b72aeff4ddb.nodes.
2226	1/2	0B/s	0B/s	0ops/s	0ops/s	0ops/s	0/0	management.bd1f22b3a854b6c.nodes.
2142	0/0	0B/s	0B/s	0ops/s	0ops/s	0ops/s	0/0	management.655c19da7e854d6f.nodes.

TIME	SYS	SEV	MESSAGE
21-12-18 12:06:24	MDS	INF	Add new MDS#3 at 10.37.130.79:2510 by request from 10.37.130.79:45672
21-12-18 12:06:24	MON	INF	MDS#3 was started
21-12-18 12:06:35	MON	INF	MDS#3 was stopped
21-12-18 12:06:35	MON	INF	CS#1027 was started
21-12-18 12:06:36	MDS	INF	New CS#1027 at 10.37.130.79:45742 (0.0.0.655c19da7e854d6f), tier=0
21-12-18 12:06:36	MON	INF	MDS#3 was started
21-12-18 12:06:38	MDS	INF	CS#1027 is active
21-12-18 12:06:45	MDS	INF	The cluster physical free space: 1.4Tb (99%), total 1.4Tb

Приведенная выше команда выводит подробную информацию о кластере stor1. Параметры мониторинга для серверов фрагментов данных (выделены красным) следующие:

CSID Идентификатор (ID) сервера фрагментов данных.

STATUS

Статус сервера фрагментов данных.

active

Сервер фрагментов данных запущен и работает.

failed

Процесс сервера фрагментов данных запущен, но возникла проблема с диском СФД.

inactive

Сервер фрагментов данных временно недоступен. Сервер фрагментов данных помечается как неактивный (inactive) в течение первых 5 минут неактивности.

offline

Сервер фрагментов данных неактивен в течение более чем 5 минут. Когда сервер фрагментов данных оказывается недоступен (offline), кластер начинает репликацию данных, чтобы восстановить те фрагменты, которые хранились на затронутом сервере фрагментов данных.

dropped

Сервер фрагментов данных удален администратором.

maintenance

Производится обслуживание сервера, на котором размещен сервер фрагментов данных.

SPACE

Общий объем дискового пространства на сервере фрагментов данных.

AVAIL

Объем доступного дискового пространства на сервере фрагментов данных.

REPLICAS

Число реплик, сохраненных на сервере фрагментов данных.

UNIQUE

Число фрагментов, у которых нет реплик.

IOWAIT

Процент времени, затраченный на ожидание выполнения операций ввода-вывода.

IOLAT

Среднее/максимальное время (в мс), которое требовалось клиенту для выполнения одной операции ввода-вывода за последние 20 секунд.

QDEPTH

Средняя глубина очереди ввода-вывода на сервере фрагментов данных.

HOST

Имя хоста или IP-адрес сервера фрагментов данных.

FLAGS

Для активных серверов фрагментов данных могут отображаться следующие флаги:

- J** СФД использует журнал записи.
- C** На СФД включено вычисление контрольной суммы. Расчет контрольных сумм позволяет вам получать уведомление, если третья сторона изменит данные на диске.
- D** Прямой ввод-вывод — нормальное состояние СФД без журнала записи.
- c** Журнал записи сервера фрагментов данных пуст: нет никаких данных, ожидающих фиксации с твердотельного накопителя журналирования записи на жесткий диск в месте расположения СФД.

6.3.1 Объяснение данных об использовании дискового пространства

Обычно сведения о том, как используется дисковое пространство в вашем кластере, можно получить с помощью команды `vstorage top`. Эта команда выдает следующую информацию о дисках: общий объем, свободный объем и подлежащий выделению объем. Например:

```
# vstorage -c stor1 top
connected to MDS#1
Cluster 'stor1': healthy
Space: [OK] allocatable 180GB of 200GB, free 1.6TB of 1.7TB
...
```

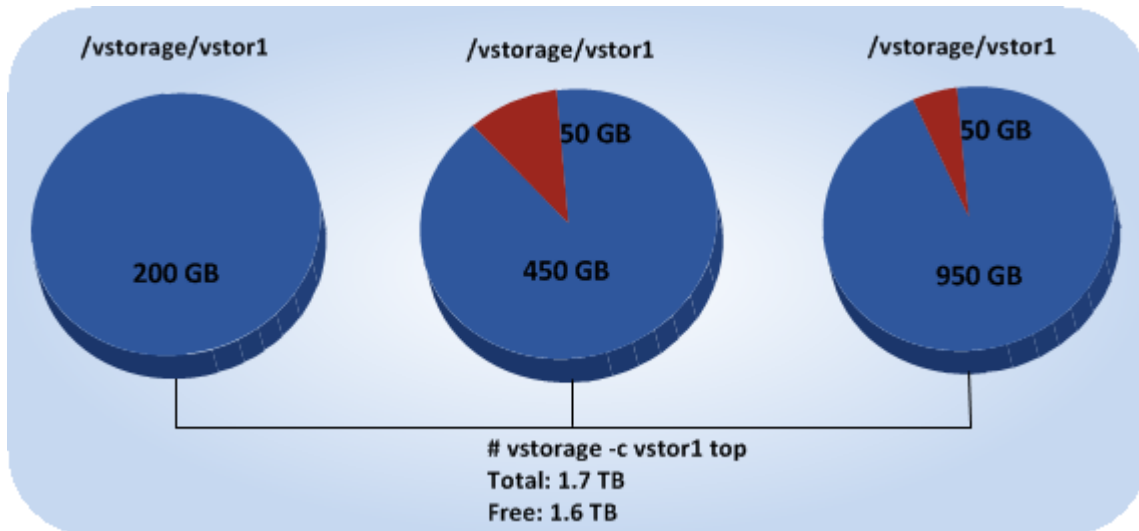
В выходных данных этой команды:

- 1,7 ТБ — это общий объем дисков в кластере `stor1`. Общий объем дисков рассчитывается на основе всего используемого и свободного пространства на всех разделах в кластере. Используемое дисковое пространство включает пространство, занятое всеми фрагментами данных и их репликами, а также пространство, занятое любыми другими файлами, которые хранятся на разделах кластера.

Допустим, что имеется раздел объемом в 100 ГБ и 20 ГБ на этом разделе занято теми или иными файлами. В случае если настроить на этом разделе сервер фрагментов данных, к кластеру будет добавлено 100 ГБ общего дискового пространства, хотя лишь 80 ГБ из этого пространства будет свободно и доступно для сохранения фрагментов данных.

- 1,6 ТБ — это свободное дисковое пространство в кластере stor1. Свободное дисковое пространство рассчитывается посредством вычета объема дискового пространства, занятого фрагментами данных и любыми другими файлами на разделах кластера, из объема общего дискового пространства.

Например, если объем свободного пространства составляет 1,6 ТБ, а общий объем дискового пространства равен 1,7 ТБ, это означает, что около 100 ГБ на разделах кластера уже занято теми или иными файлами.



- `allocatable 180GB of 200GB` (подлежат выделению 180 ГБ из 200 ГБ) — это объем свободного дискового пространства, который может использоваться для сохранения фрагментов данных. Более подробные данные см. ниже в разделе **Объяснение подлежащего выделению дискового пространства**.

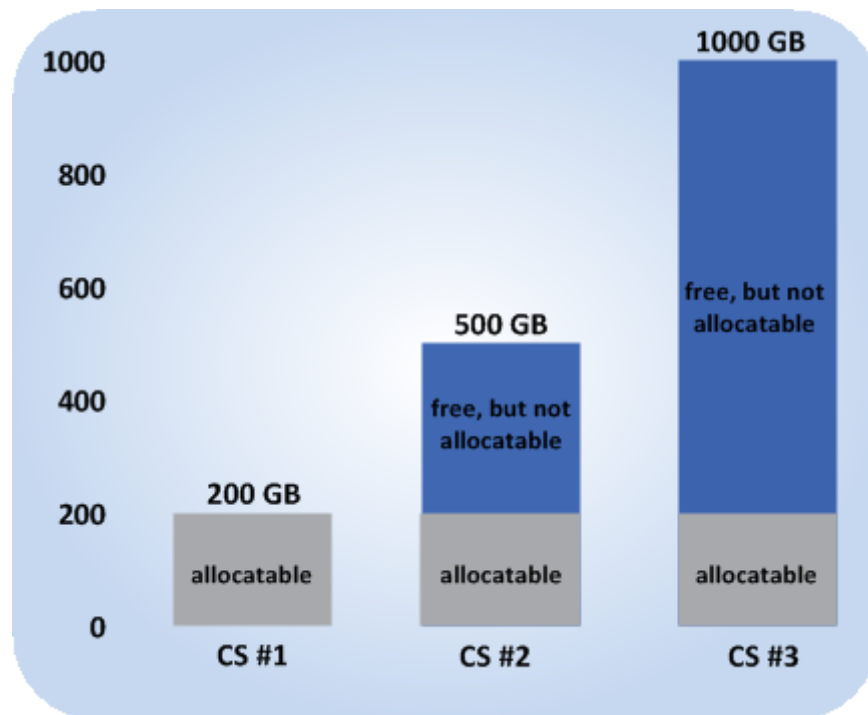
6.3.1.1 Объяснение подлежащего выделению дискового пространства

Контролируя данные о дисковом пространстве в кластере, следует также обращать внимание на пространство, о котором утилита `vstorage top` сообщает, как о подлежащем выделению (*allocatable*). Подлежащее выделению пространство — это объем дискового пространства, которое свободно и может использоваться для сохранения пользовательских данных. Когда это пространство закончится, записывать в кластер данные станет невозможно.

Расчет подлежащего выделению дискового пространства показан в следующем примере.

- Кластер содержит три сервера фрагментов данных. На первом сервере имеется 200 ГБ дискового пространства, на втором — 500 ГБ, а на третьем — 1 ТБ.

- В кластере используется коэффициент репликации по умолчанию 3: это означает, что у каждого фрагмента данных должно быть три реплики, которые должны храниться на трех различных серверах фрагментов данных.



В этом примере доступное дисковое пространство составляет 200 ГБ, то есть равно объему дискового пространства наименьшего из серверов фрагментов данных.

```
# vstorage -c stor1 top
connected to MDS#1
Cluster 'stor1': healthy
Space: [OK] allocatable 180GB of 200GB, free 1.6TB of 1.7TB
...
```

В этой конфигурации кластера одна реплика каждого из фрагментов данных должна сохраняться на каждом из серверов. Таким образом, после того как доступное дисковое пространство на наименьшем сервере фрагментов данных (200 ГБ) исчерпается, в кластере нельзя будет создавать новые фрагменты, пока в кластер не будет добавлен новый сервер фрагментов данных или не будет уменьшен коэффициент репликации.

Если изменить коэффициент репликации на 2, то команда `vstorage top` сообщит о доступном дисковом пространстве в 700 ГБ.

```
# vstorage set-attr -R /mnt/vstorage replicas=2:1
# vstorage -c stor1 top
connected to MDS#1
```

```
Cluster 'stor1': healthy
Space: [OK] allocatable 680GB of 700GB, free 1.6TB of 1.7TB
...
```

Доступное дисковое пространство увеличилось, поскольку теперь для каждого фрагмента данных создаются только две реплики и можно создавать новые фрагменты, даже когда на наименьшем из серверов фрагментов данных закончится место (в этом случае реплики будут сохраняться на одном из больших серверов фрагментов данных).

Доступное для выделения дисковое пространство также может ограничиваться лицензией.

6.3.1.2 Просмотр пространства, занятого фрагментами данных

Чтобы просмотреть общий объем дискового пространства, занятого всеми пользовательскими данными в кластере, запустите команду `vstorage top` и нажмите клавишу `V` на клавиатуре. После этого вывод команды должен выглядеть подобно следующему.

```
# vstorage -c stor1 top
Cluster 'stor1': healthy
Space: [OK] allocatable 1.32TB of 1.44TB, free 1.39TB of 1.44TB
MDS nodes: 3 of 3, epoch uptime: 19d 23h, cluster version: 128
CS nodes: 3 of 3 (3 avail, 0 inactive, 0 offline), storage version: 128
License: ACTIVE (expiration: 01/10/2021, capacity: 10TB, used: 20.3GB)
Replication: 1 norm, 1 limit
Chunks: [OK] 323 (100%) healthy, 0 (0%) standby, 0 (0%) degraded, 0 (0%) urgent,
        0 (0%) blocked, 0 (0%) pending, 0 (0%) offline, 0 (0%) replicating,
        0 (0%) overcommitted, 0 (0%) deleting, 0 (0%) void
FS: 20.3GB in 757 files, 757 inodes, 244 file maps, 323 chunks, 889 chunk replicas
IO:   read    0B/s ( 0ops/s), write  0B/s ( 0ops/s)
IO total: read 37.1GB ( 473Kops), write 133.7GB ( 4.7Mops)
Repl IO: read  0B/s, write:  0B/s
Sync rate:  0ops/s, datasync rate:  0ops/s
IO QDEPTH: 0.0 aver, 0.0 max
...
```

В поле **FS** отображается размер всех пользовательских данных в кластере без учета реплик.

6.3.2 Определение состояния фрагментов

Ниже приведен список всех возможных состояний фрагментов.

Исправен

Количество и процент фрагментов данных, у которых достаточно активных реплик. Это нормальное состояние фрагментов данных.

Офлайн

Количество и процент фрагментов данных, все реплики которых находятся в отключенном состоянии. Такие фрагменты данных полностью недоступны для кластера, невозможно их реплицировать, считывать или записывать в них данные. Все запросы к фрагменту данных, находящемуся в состоянии «Офлайн», замораживаются до тех пор, пока сервис CS, хранящий реплику соответствующего фрагмента, не станет активным.

Во избежание потери данных следует как можно быстрее вернуть серверы фрагментов данных, находящиеся в состоянии «Офлайн», в подключенное состояние.

Заблокировано

Количество и процент фрагментов данных, у которых число активных реплик меньше заданного минимального количества. Запросы на запись к заблокированному фрагменту данных замораживаются до тех пор, пока у него не будет по крайней мере заданного минимального количества реплик. В то же время запросы на чтение к заблокированным фрагментам выполняются, поскольку у них еще есть активные реплики. Заблокированные фрагменты имеют более высокий приоритет репликации, чем деградированные.

Наличие заблокированных фрагментов данных в кластере повышает риск потери данных, поэтому следует отложить все техническое обслуживание на рабочих серверах кластера и как можно быстрее вернуть недоступные серверы фрагментов данных в рабочее состояние.

Деградировал

Количество и процент фрагментов данных с небольшим числом активных реплик, но не меньшим установленного минимума. Для таких фрагментов данных возможно и чтение, и запись в них. Однако в последнем случае деградированный фрагмент данных становится срочным.

Replicating (Репликация)

Число и процент фрагментов, для которых осуществляется репликация. Операции записи в эти фрагменты заморожены до момента окончания репликации.

Void (Пустые)

Число и процент фрагментов, которые были выделены, но еще ни разу не использовались.

Такие фрагменты не содержат данных. Наличие в кластере некоторого числа пустых фрагментов — это нормальная ситуация.

Pending (Ожидают)

Число и процент фрагментов, которые необходимо реплицировать немедленно. Для выполнения запроса на запись от клиента во фрагмент у этого фрагмента должно быть не менее

заданного минимального числа реплик. Если их меньше, фрагмент блокируется и выполнить запрос невозможно. Поскольку заблокированные фрагменты необходимо реплицировать как можно быстрее, кластер помещает их в отдельную, высокоприоритетную очередь репликации и сообщает о них как об ожидающих репликации.

Urgent (Срочные)

Число и процент фрагментов, которые деградированы и имеют неидентичные реплики. Реплики деградированного фрагмента могут оказаться неидентичными, если некоторые из них окажутся недоступными во время операции записи. В таком случае часть реплик будет содержать новые данные, в то время как в других будут по-прежнему содержаться старые данные. Такие реплики удаляются кластером как можно быстрее. Срочные фрагменты не влияют на целостность информации, так как актуальные данные все равно хранятся не менее чем в заданном минимальном числе реплик.

Overcommitted (Фиксированные с избытком)

Число и процент фрагментов, у которых количество реплик больше нормального. Обычно такие фрагменты возникают после снижения нормального числа реплик или удаления большого объема данных. Со временем излишние реплики удаляются, однако во время репликации этот процесс может происходить медленнее.

Deleting (Удаление)

Число и процент фрагментов, поставленных в очередь на удаление.

6.4 Мониторинг клиентов

Посредством мониторинга клиентов можно контролировать состояние и работоспособность серверов, используемых для доступа к виртуальным машинам. Для мониторинга клиентов используйте команду `vstorage -c <имя_кластера> top`. Например:

```

Cluster 'stor1': healthy
Space: [OK] allocatable 1.32TB of 1.44TB, free 1.39TB of 1.44TB
MDS nodes: 3 of 3, epoch uptime: 19d 23h
CS nodes: 3 of 3 (3 avail, 0 inactive, 0 offline)
License: ACTIVE (expiration: 01/10/2021, capacity: 10TB, used: 20.3GB)
Replication: 1 norm, 1 limit
IO:      read      0B/s ( 0ops/s), write      0B/s ( 0ops/s)

```

MDSID	STATUS	%CTIME	COMMITTS	%CPU	MEM	UPTIME	HOST
M 3	avail	0.0%	0/s	1.1%	192m	19d 23h	management.655c19da7e854d6f.nodes.svc.vstoragedomain:2510
1	avail	0.0%	0/s	0.2%	192m	20d 0h	management.b2823b72aeff4ddb.nodes.svc.vstoragedomain:2510
2	avail	0.0%	0/s	0.0%	192m	19d 23h	management.bda1f22b3a854b6c.nodes.svc.vstoragedomain:2510

CSID	STATUS	SPACE	AVAIL	REPLICAS	UNIQUE	IOWAIT	IOLAT (ms)	QDEPTH	HOST
1027	active	492.0G	451.4G	295	12	0%	0/0	0.0	management.655c19da7e854d6f.nodes.svc.v
1025	active	492.0G	449.5G	305	22	0%	0/0	0.0	management.b2823b72aeff4ddb.nodes.svc.v
1026	active	492.0G	453.0G	289	6	0%	0/0	0.0	management.bda1f22b3a854b6c.nodes.svc.v

CLID	LEASES	READ	WRITE	RD OPS	WR OPS	FSYNCS	IOLAT (ms)	HOST
2050	1/222	6B/s	6B/s	0ops/s	0ops/s	0ops/s	0.13/1	management.b2823b72aeff4ddb.nodes.
2226	1/2	0B/s	0B/s	0ops/s	0ops/s	0ops/s	0/0	management.bda1f22b3a854b6c.nodes.
2142	0/0	0B/s	0B/s	0ops/s	0ops/s	0ops/s	0/0	management.655c19da7e854d6f.nodes.

TIME	SYS	SEV	MESSAGE
21-12-18 12:06:24	MDS	INF	Add new MDS#3 at 10.37.130.79:2510 by request from 10.37.130.79:45672
21-12-18 12:06:24	MON	INF	MDS#3 was started
21-12-18 12:06:35	MON	INF	MDS#3 was stopped
21-12-18 12:06:35	MON	INF	CS#1027 was started
21-12-18 12:06:36	MDS	INF	New CS#1027 at 10.37.130.79:45742 (0.0.0.655c19da7e854d6f), tier=0
21-12-18 12:06:36	MON	INF	MDS#3 was started
21-12-18 12:06:38	MDS	INF	CS#1027 is active
21-12-18 12:06:45	MDS	INF	The cluster physical free space: 1.4Tb (99%), total 1.4Tb

Приведенная выше команда выводит подробную информацию о кластере stor1. Параметры мониторинга для клиентов (выделены красным) следующие:

CLID Идентификатор (ID) клиента.

LEASES

Среднее число файлов, открытых клиентом на чтение/запись и все еще не закрытых, за последние 20 секунд.

READ

Средняя скорость (в байтах в секунду), с которой клиент считывает данные, за последние 20 секунд.

WRITE

Средняя скорость (в байтах в секунду), с которой клиент записывает данные, за последние 20 секунд.

RD_OPS

Среднее число операций чтения, выполняемых клиентом за секунду, в течение последних 20 секунд.

WR_OPS

Среднее число операций записи, выполняемых клиентом за секунду, в течение последних 20 секунд.

FSYNCS

Среднее число операций синхронизации, выполняемых клиентом за секунду, в течение последних 20 секунд.

IOLAT

Среднее/максимальное время (в мс), которое требовалось клиенту для выполнения одной операции ввода-вывода, за последние 20 секунд.

HOST

Имя хоста или IP-адрес клиента.

6.5 Мониторинг физических дисков

Состояние S.M.A.R.T. физических дисков отслеживается с помощью инструмента `smartctl`, устанавливаемого вместе с Acronis Инфраструктура. Чтобы он работал, необходимо включить поддержку S.M.A.R.T. в BIOS соответствующего сервера. Инструмент запускается каждые 10 минут по заданию `cron`, которое также добавляется в ходе установки. Инструмент `smartctl` опрашивает все физические диски, подсоединенные к серверам в кластере, в том числе твердотельные накопители кэширования и журналирования, и передает полученные результаты на сервер MDS.

Результаты опроса накопителей за последние 10 минут можно просмотреть в выходных данных команды `vstorage top`. Например:

```
Cluster 'stor1': healthy, SMART warning
Space: [OK] allocatable 100GB (+778GB unlicensed) of 926GB, free 924GB of 926GB
MDS nodes: 1 of 1, epoch uptime: 7d 22h
CS nodes: 2 of 2 (2 avail, 0 inactive, 0 offline)
Replication: 1 norm, 1 limit
IO:      read      0B/s ( 0ops/s), write      0B/s ( 0ops/s)
```

MDSID	STATUS	%CTIME	COMMITTS	%CPU	MEM	UPTIME	HOST
M 1	avail	0.0%	0/s	0.0%	48m	7d 22h	pcs36.qa.sw.ru:2510

CSID	STATUS	SPACE	AVAIL	REPLICAS	UNIQUE	IOWAIT	IOLAT(ms)	QDEPTH	HOST
1025	active	9.1GB	7.1GB	0	0	0%	0/0	0.0	pcs36.q
1026	active	916GB	870GB	0	0	0%	0/0	0.0	pcs36.q

CLID	LEASES	READ	WRITE	RD_OPS	WR_OPS	FSYNCS	IOLAT(ms)	HOST

TIME	SYS	SEV	MESSAGE
01-07-14 16:42:19	MDN	WRN	CS#1026 was stopped
01-07-14 16:42:26	JRN	INF	MDS#1 at 10.29.2.16:2510 became master
01-07-14 16:42:26	MDS	WRN	License not installed, please add license using comma
01-07-14 16:42:29	MDN	WRN	MDS#1 was stopped
01-07-14 16:42:44	MDS	INF	CS#1025, CS#1026 are active
01-07-14 16:42:53	MDS	INF	The cluster is healthy with 2 active CS
01-07-14 16:42:53	MDS	INF	The cluster physical free space: 925.0Gb (99%), total

Если в главной таблице отображается сообщение **SMART warning** (Предупреждение SMART), это означает, что для одного из физических дисков близок аппаратный отказ по данным S.M.A.R.T. Нажмите клавишу **d**, чтобы переключиться на таблицу дисков и просмотреть более подробные сведения. Например:

```
Cluster 'stor1': healthy, SMART warning
Space: [OK] allocatable 100GB (+778GB unlicensed) of 926GB, free 924GB of 926GB
MDS nodes: 1 of 1, epoch uptime: 7d 22h
CS nodes: 2 of 2 (2 avail, 0 inactive, 0 offline)
Replication: 1 norm, 1 limit
IO:      read      0B/s ( 0ops/s), write      0B/s ( 0ops/s)
```

DISK	SMART	TEMP	CAPACITY	SERIAL	MODEL	HOST
sdc	OK	27C	931GB	1374X80PS	TOSHIBA DT01ACA100	pcs36.qa
sde	Warn	31C	931GB	MSE5235U36ZHWJ	Hitachi HDS721010DLE630	pcs36.qa

В таблице дисков отображаются следующие параметры:

DISK Имя диска, присвоенное операционной системой.

SMART

Состояние S.M.A.R.T. диска:

OK Диск исправен.

Warn (Предупреждение)

Диск в состоянии, близком к аппаратному отказу. Близость к аппаратному отказу означает, что по меньшей мере один из следующих счетчиков S.M.A.R.T. не равен нулю:

- Reallocated Sector Count (Количество перераспределенных секторов)
- Reallocated Event Count (Количество событий перераспределения)
- Current Pending Sector Count (Текущее количество ожидающих секторов)
- Offline Uncorrectable (Отключено без возможности исправления)

TEMP

Температура диска в градусах Цельсия.

CAPACITY

Емкость диска.

SERIAL

Серийный номер диска.

MODEL

Модель диска.

HOST

Адрес хоста диска.

Чтобы отключить мониторинг S.M.A.R.T. дисков, удалите соответствующее задание cron.

6.6 Мониторинг журналов событий

С помощью утилиты `vstorage -c <имя_кластера> top` можно вести мониторинг важных событий, происходящих в кластере хранения данных. Например:

```

Cluster 'stor1': healthy
Space: [OK] allocatable 1.32TB of 1.44TB, free 1.39TB of 1.44TB
MDS nodes: 3 of 3, epoch uptime: 19d 23h
CS nodes: 3 of 3 (3 avail, 0 inactive, 0 offline)
License: ACTIVE (expiration: 01/10/2021, capacity: 10TB, used: 20.3GB)
Replication: 1 norm, 1 limit
IO:      read    0B/s ( 0ops/s), write    0B/s ( 0ops/s)

```

MDSID	STATUS	%TIME	COMMITTS	%CPU	MEM	UPTIME	HOST
M 3	avail	0.0%	0/s	1.1%	192m	19d 23h	management.655c19da7e854d6f.nodes.svc.vstoragedomain:2510
1	avail	0.0%	0/s	0.2%	192m	20d 0h	management.b2823b72aeff4ddb.nodes.svc.vstoragedomain:2510
2	avail	0.0%	0/s	0.0%	192m	19d 23h	management.bda1f22b3a854b6c.nodes.svc.vstoragedomain:2510

CSID	STATUS	SPACE	AVAIL	REPLICAS	UNIQUE	IOWAIT	IOLAT (ms)	QDEPTH	HOST
1027	active	492.0G	451.4G	295	12	0%	0/0	0.0	management.655c19da7e854d6f.nodes.svc.v
1025	active	492.0G	449.5G	305	22	0%	0/0	0.0	management.b2823b72aeff4ddb.nodes.svc.v
1026	active	492.0G	453.0G	289	6	0%	0/0	0.0	management.bda1f22b3a854b6c.nodes.svc.v

CLID	LEASES	READ	WRITE	RD OPS	WR OPS	FSYNCS	IOLAT (ms)	HOST
2050	1/222	6B/s	6B/s	0ops/s	0ops/s	0ops/s	0.13/1	management.b2823b72aeff4ddb.nodes.
2226	1/2	0B/s	0B/s	0ops/s	0ops/s	0ops/s	0/0	management.bda1f22b3a854b6c.nodes.
2142	0/0	0B/s	0B/s	0ops/s	0ops/s	0ops/s	0/0	management.655c19da7e854d6f.nodes.

TIME	SYS	SEV	MESSAGE
21-12-18 12:06:24	MDS	INF	Add new MDS#3 at 10.37.130.79:2510 by request from 10.37.130.79:45672
21-12-18 12:06:24	MON	INF	MDS#3 was started
21-12-18 12:06:35	MON	INF	MDS#3 was stopped
21-12-18 12:06:35	MON	INF	CS#1027 was started
21-12-18 12:06:36	MDS	INF	New CS#1027 at 10.37.130.79:45742 (0.0.0.655c19da7e854d6f), tier=0
21-12-18 12:06:36	MON	INF	MDS#3 was started
21-12-18 12:06:38	MDS	INF	CS#1027 is active
21-12-18 12:06:45	MDS	INF	The cluster physical free space: 1.4Tb (99%), total 1.4Tb

Приведенная выше команда отображает последние события, которые произошли в кластере stor1.

Сведения о событиях (выделены красным) отображаются в таблице со следующими столбцами:

TIME

Время события.

SYS Компонент кластера, в котором произошло событие (например, MDS для сервера MDS или JRN для локального журнала).

SEV Уровень серьезности события.

MESSAGE

Описание события.

В следующей таблице перечислены базовые события, отображаемые при запуске утилиты vstorage top.

Таблица 6.6.1: Базовые события

Событие	Серьезность	Описание
MDS#<N> (<адрес>:<порт>) lags behind for more than 1000 rounds (Сервер MDS № <N> отстает в течение более 1000 циклов)	JRN err (Ошибка журнала)	Создается главным сервером MDS, когда тот обнаруживает устаревание/отставание сервера MDS № <N>. Это сообщение может указывать на то, что какой-то сервер MDS работает очень медленно и не успевает обрабатывать операции.
MDS#<N> (<адрес>:<порт>) didn't accept commits for <i>M</i> sec (Сервер MDS № <N> не принимал фиксации в течение <i>M</i> с.)	JRN err (Ошибка журнала)	Создается главным сервером MDS, если сервер MDS № <N> не принимал фиксации в течение <i>M</i> секунд. Такой сервер MDS № <N> помечается как устаревший. Это сообщение может указывать, что на сервере MDS № <N> возникли проблемы со службой MDS. Такая проблема может быть критической, и ее следует решить как можно раньше.
MDS#<N> (<адрес>:<порт>) state is outdated and will do a full resync (Состояние сервера MDS № <N> устарело, будет произведена полная повторная синхронизация)	JRN err (Ошибка журнала)	Создается главным сервером MDS, если на сервере MDS № <N> будет проведена полная повторная синхронизация. Такой сервер MDS № <N> помечается как устаревший. Это сообщение может указывать, что определенный сервер MDS работал слишком медленно или соединение с ним нарушилось на такое продолжительное время, что он уже не поддерживает актуального состояния метаданных и его необходимо синхронизировать повторно. Такая проблема может быть критической, и ее следует решить как можно раньше.

Продолжается на следующей странице

Таблица 6.6.1 – продолжение с предыдущей страницы

Событие	Серьезность	Описание
MDS#<N> at <адрес>:<порт> became master (Сервер MDS № <N> стал главным сервером)	JRN info (Информация в журнале)	Создается каждый раз, когда в кластере выбирается новый главный сервер MDS. Частые изменения главного сервера MDS могут указывать на проблемы с сетевыми подключениями и негативно влиять на работу кластеров.
The cluster is healthy with <i>N</i> active CS (Кластер работоспособен с <i>N</i> активных СФД)	MDS info (Информация MDS)	Создается в случае, если состояние кластера меняется на работоспособное, и при выборе нового главного сервера MDS. Это сообщение указывает, что все серверы фрагментов в кластере активны и число реплик соответствует установленным в кластере требованиям.
The cluster is degraded with <i>N</i> active, <i>M</i> inactive, <i>K</i> offline CS (Кластер деградирован: <i>N</i> активных серверов фрагментов, <i>M</i> неактивных, <i>K</i> отключены)	MDS warn (Предупреждение MDS)	Создается, когда состояние кластера меняется на деградированное, а также при выборе нового главного сервера MDS. Это сообщение указывает, что некоторые серверы фрагментов в кластере <ul style="list-style-type: none"> • неактивны, то есть не посылают каких-либо сообщений о регистрации, или • отключены, то есть были неактивны дольше, чем задано в параметре <code>mds.wd.offline_tout</code> (по умолчанию 5 минут).

Продолжается на следующей странице

Таблица 6.6.1 – продолжение с предыдущей страницы

Событие	Серьезность	Описание
The cluster failed with N active, M inactive, K offline CS (Произошел отказ кластера: N активных серверов фрагментов данных, M неактивных, K отключены)	MDS err (Ошибка MDS)	Создается, когда состояние кластера меняется на отказ кластера, а также при выборе нового главного сервера MDS. Это сообщение указывает, что число отключенных серверов фрагментов данных превышает число, заданное в параметре <code>mds.wd.max_offline_cs</code> (по умолчанию равно 2). В случае отказа кластера планирование автоматической репликации прекращается. Поэтому администратору кластера необходимо принять меры: либо восстановить отказавшие серверы кластера, либо увеличить значение <code>mds.wd.max_offline_cs</code> . При установке значения 0 этого параметра перехода в режим отказа не происходит никогда.
The cluster is filled up to $\langle N \rangle\%$ (Кластер заполнен на $\langle N \rangle\%$)	MDS info/warn (Информация/предупреждение MDS)	Уведомляет о текущем использовании пространства в кластере. Если израсходовано 80% или более дискового пространства, формируется предупреждение. Важно иметь некоторый резервный объем дискового пространства для реплик данных на случай отказа одного из серверов фрагментов данных.
Replication started, N chunks are queued (Репликация начата, в очередь помещено N фрагментов)	MDS info (Информация MDS)	Создается, когда кластер начинает автоматическую репликацию данных для восстановления отсутствующих реплик.
Replication completed (Репликация завершена)	MDS info (Информация MDS)	Создается, когда кластер завершает автоматическую репликацию данных.

Продолжается на следующей странице

Таблица 6.6.1 – продолжение с предыдущей страницы

Событие	Серьезность	Описание
CS#<N> has reported hard error on <i>path</i> (СФД № <N> сообщил об аппаратной ошибке по пути <i>path</i>)	MDS warn (Предупреждение MDS)	Создается, когда СФД № <N> обнаруживает повреждение данных на диске. Рекомендуется проверить оборудование на предмет ошибок и заменить диски с поврежденными данными как можно быстрее.
CS#<N> has not registered during the last <i>T</i> sec and is marked as inactive/offline (СФД № <N> не проходил регистрацию в течение последних <i>T</i> секунд и помечен как неактивный/отключенный)	MDS warn (Предупреждение MDS)	Создается, когда СФД № <N> был недоступен в течение некоторого времени. Через 5 минут его состояние меняется на отключенный (offline), при этом запускается автоматическая репликация данных для восстановления реплик, хранившихся на сервере фрагментов данных, который оказался отключен.
Failed to allocate <i>N</i> replicas for „ <i>path</i> “ by request from <адрес>:<порт> - <i>K</i> out of <i>M</i> chunks servers are available (Не удалось распределить <i>N</i> реплик для пути „ <i>path</i> “ по запросу от <адрес>:<порт> - <i>K</i> из <i>M</i> серверов фрагментов данных недоступны)	MDS warn (Предупреждение MDS)	Создается, когда кластер не может выделить пространство для реплик фрагментов, например в случае, если на нем исчерпалось дисковое пространство.
Failed to allocate <i>N</i> replicas for „ <i>path</i> “ by request from <адрес>:<порт> since only <i>K</i> chunk servers are registered (Не удалось распределить <i>N</i> реплик для пути „ <i>path</i> “ по запросу от <адрес>:<порт>, поскольку зарегистрировано лишь <i>K</i> серверов фрагментов данных)	MDS warn (Предупреждение MDS)	Создается, когда кластер не может выделить пространство для реплик фрагментов, потому что в кластере зарегистрировано слишком мало серверов фрагментов данных.

6.7 Мониторинг параметров репликации

Настраивая параметры репликации, учитывайте, что новые параметры не вступают в силу немедленно. Например, при увеличении параметра числа реплик по умолчанию для фрагментов данных его отработка может занять некоторое время в зависимости от нового значения параметра и числа фрагментов данных в кластере.

Чтобы убедиться, что новые параметры репликации были успешно применены в кластере, выполните следующие действия.

1. Выполните команду `vstorage -c <имя_кластера> top`.
2. Нажмите клавишу **V**, чтобы отобразить дополнительные сведения о кластере. Типичный вывод этой команды может выглядеть следующим образом.

```
# vstorage -c stor1 top
Cluster 'stor1': healthy
Space: [OK] allocatable 448.6GB of 492.0GB, free 1.39TB of 1.44TB
MDS nodes: 3 of 3, epoch uptime: 20d 0h
CS nodes: 3 of 3 (3 avail, 0 inactive, 0 offline)
License: ACTIVE (expiration: 01/10/2021, capacity: 10TB, used: 20.3GB)
Replication: 3 norm, 2 limit
Chunks: [Warning] 187 (57%) healthy, 0 (0%) standby, 0 (0%) degraded, 135 (41%) urg
        0 (0%) blocked, 0 (0%) pending, 0 (0%) offline, 1 (0%) replicating,
        0 (0%) overcommitted, 0 (0%) deleting, 0 (0%) void
IO:      read      0B/s ( 0ops/s), write 106KB/s ( 7ops/s)
...
```

3. Проверьте значение в поле **Chunks** (Фрагменты), учитывая следующие моменты.
 - При уменьшении параметров числа реплик обращайте внимание на фрагменты в состоянии фиксации с избытком (*overcommitted*) и в состоянии удаления (*deleting*). После завершения процесса репликации в выходных данных команды не должны отображаться фрагменты ни в одном из этих состояний.
 - При увеличении параметров числа реплик обращайте внимание на фрагменты в заблокированном состоянии (*blocked*) и подлежащие срочной обработке (*urgent*). После завершения процесса репликации в выходных данных команды не должны отображаться фрагменты ни в одном из этих состояний. Кроме того, пока процесс продолжает выполняться, значение параметра работоспособности (*healthy*) будет ниже 100 %.

Дополнительные сведения о возможных состояниях фрагментов см. в разделе [Определение состояния фрагментов](#) (страница 211).

ГЛАВА 7

Доступ к кластерам хранилища через iSCSI

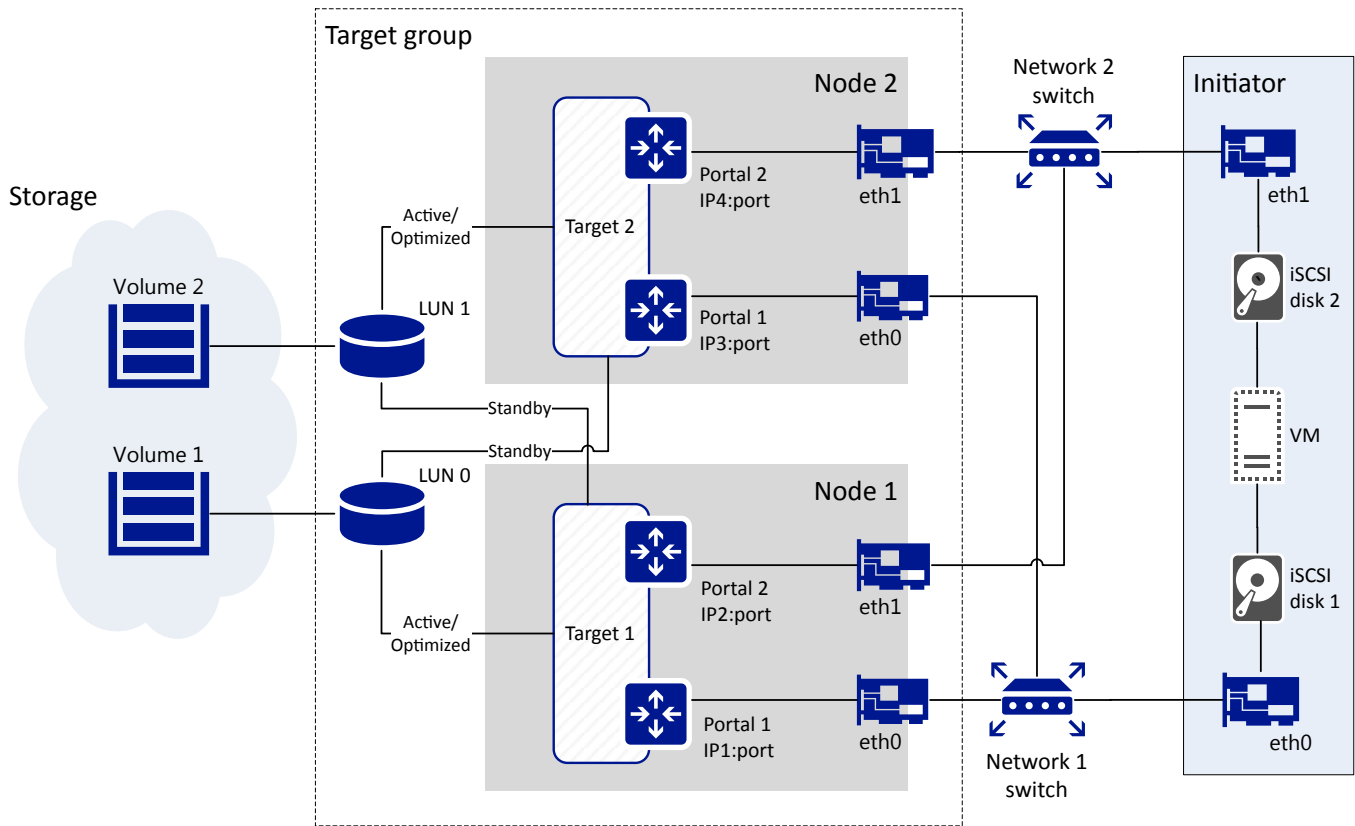
Acronis Инфраструктура позволяет экспортировать дисковое пространство кластера во внешние операционные системы и сторонние решения виртуализации в форме блочных устройств LUN по протоколу iSCSI подобно сети SAN.

В Acronis Инфраструктура можно создавать группы избыточных целевых устройств, работающих на разных серверах хранения данных. К каждой группе целевых устройств можно прикрепить множество томов хранения данных с собственной избыточностью, обеспечиваемой слоем хранения. Эти тома экспортируются целевыми устройствами как идентификаторы LUN.

На каждом сервере в группе целевых устройств может размещаться одно целевое устройство для данной группы при использовании Ethernet или по одному целевому устройству на порт FC при использовании Fibre Channel. Если один из серверов в группе выйдет из строя вместе со своими целевыми устройствами, то исправные целевые устройства из этой же группы продолжат предоставлять доступ к устройствам LUN, которые ранее обслуживались отказавшими целевыми устройствами.

Можно создать несколько групп целевых устройств на одних и тех же серверах. Однако том в любой момент времени можно присоединить только к одной группе целевых устройств.

На рисунке ниже показана типичная настройка для экспорта дискового пространства Acronis Инфраструктура через iSCSI.



На рисунке показаны два тома, расположенные в избыточном хранилище, предоставляемом продуктом Acronis Инфраструктура. Тома присоединены как LUN к группе из двух целевых устройств, работающих на серверах Acronis Инфраструктура. Каждое целевое устройство содержит два портала, по одному на каждый сетевой интерфейс с типом трафика iSCSI, что составляет в целом четыре обнаруживаемые конечные точки с разными IP-адресами. Каждое целевое устройство предоставляет доступ ко всем LUN, присоединенным к группе.

Целевые устройства работают в режиме ALUA, поэтому один путь к тому является предпочтительным и считается активным/оптимизированным, а другой — резервным. В норме активный/оптимизированный путь выбирается инициатором (явный режим ALUA). Если инициатор не поддерживает этот путь или истекает время ожидания, то путь выбирается самим хранилищем (неявный режим ALUA).

Сетевые интерфейсы eth0 и eth1 на каждом сервере подключены к разным коммутаторам для избыточности. Инициатор, например VMware ESXi, также подключается к обоим коммутаторам и предоставляет тома в качестве дисков iSCSI 1 и 2 виртуальной машине по разным сетевым путям.

Если активный/оптимизированный путь по какой-либо причине станет недоступен (например, при отказе сервера с целевым объектом или сетевого коммутатора), для подключения к тому вместо него

будет использоваться резервный путь через другое целевое устройство. После восстановления активного/оптимизированного пути он будет использоваться снова.

7.1 Обзор последовательности операций с iSCSI

Типичная последовательность операций при экспорте томов через iSCSI следующая.

1. Назначьте сеть с типом трафика **iSCSI** сетевому интерфейсу на каждом сервере, который будет добавлен в группу целевых устройств. См. раздел *Управление типами трафика и сетями* (страница 5).
2. Создайте группу целевых устройств на выбранных серверах, указав сведения о WWN и порталах целевых устройств. Целевые устройства будут созданы автоматически и добавлены в группу. Порталы целевых устройств будут созданы на указанных сетевых интерфейсах и портах. См. раздел *Создание групп целевых устройств* (страница 228).
3. Создайте тома и присоедините их к группе целевых устройств. См. раздел *Управление томами* (страница 114).
4. При необходимости включите авторизацию CHAP для группы целевых устройств: создайте учетные записи CHAP и назначьте их группе целевых устройств. См. раздел *Управление учетными записями CHAP* (страница 240).
5. При необходимости включите авторизацию ACL для группы целевых устройств: создайте список инициаторов, которым будет разрешен доступ только к определенным LUN. Инициаторы, не включенные в список, будут иметь доступ ко всем LUN в группе целевых устройств. См. раздел *Управление представлениями LUN* (страница 242).
6. Запустите группу целевых устройств. См. раздел *Запуск и остановка групп целевых устройств* (страница 230).
7. Подключите инициаторы к целевым устройствам с помощью стандартных средств операционной системы или используемого продукта, например `iscsiadm`. Используйте команду `vstorage-target session-list` для просмотра активных сеансов iSCSI на сервере в группе целевых устройств.

7.1.1 Управление устаревшими целевыми устройствами iSCSI

После обновления продукта Acronis Инфраструктура до версии 2.5 можно запускать старые целевые устройства iSCSI, созданные в версии 2.4, вместе с новыми. Старые целевые устройства iSCSI обнаруживаются командой `vstorage-iscsi list` и управляются только с помощью утилиты `vstorage-iscsi` (см. руководство администратора Acronis Storage 2.4 по командной строке).

Так как старые целевые устройства iSCSI не поддерживают режим ALUA, их LUN не обеспечивают высокой доступности. Чтобы включить для них высокую доступность, открепите устройство LUN от старой группы целевых устройств с помощью команды `vstorage-iscsi lun-detach -t <target_IQN> -l <lun_ID>` и присоедините к только что созданной, как описано в разделе [Присоединение томов iSCSI к группам целевых устройств](#) (страница 233).

7.2 Настройка инструмента командной строки

Перед тем как использовать инструмент командной строки `vstorage-target` для экспорта томов через iSCSI, настройте его, как описано далее. Выполните эти шаги на каждом сервере, где будут работать целевые устройства iSCSI.

1. Создайте файл конфигурации `/etc/vstorage/iscsi/config.json`, содержащий как минимум эти обязательные параметры:

```
{
  "ClusterName": "cluster1",
  "VolumesRoot": "/vols/iscsi/vols",
}
```

где `ClusterName` — имя кластера хранилища, а `VolumesRoot` — путь к каталогу для томов iSCSI.

Также можно задать эти дополнительные параметры:

- `"PcsLogLevel"`, уровень ведения журнала, в диапазоне от 1 (записываются только ошибки) до 7 (все сведения, включая сообщения отладки).
- `"LogPath"`, путь к файлам журнала, по умолчанию `"/var/log/vstorage"` (журнал будет сохраняться в файл `vstorage-target.log`).
- `"GetTimeout"`, время ожидания команды инициатора для чтения статуса группы портов целевых устройств, по умолчанию 3000 мс.

1. Включите сервис мониторинга целевых устройств.

```
# systemctl start vstorage-target-monitor.service
# systemctl enable vstorage-target-monitor.service
```

2. Создайте каталог томов iSCSI, если его не существует.

```
# mkdir -p /mnt/vstorage/vols/iscsi/
```

Если вы измените файл конфигурации позже, перезапустите сервис мониторинга TCM, чтобы применить изменения.

```
# systemctl restart vstorage-target-monitor.service
```

7.3 Управление группами целевых устройств

В этом разделе описано, как создавать группы целевых устройств iSCSI и управлять ими.

7.3.1 Создание групп целевых устройств

Перед созданием каких-либо групп целевых устройств назначьте сеть с типом трафика iSCSI сетевому интерфейсу на каждом сервере, который будет добавлен в группу целевых устройств.

Для создания группы целевых устройств потребуется файл конфигурации со списком всех серверов для добавления в группу, а также WWN и порталов целевых устройств. Например:

```
[
  {
    "NodeId": "01baeabee73e4a0d",
    "WWN": "iqn.2013-10.com.vstorage:test1",
    "Portals": [
      {
        "Addr": "192.168.10.11",
        "Port": 3025
      }
    ]
  },
  {
    "NodeId": "0d90158e9d2444e1",
    "WWN": "iqn.2013-10.com.vstorage:test2",
    "Portals": [
      {
```

```

        "Addr": "192.168.10.12",
        "Port": 3025
    }
  ],
},
{
  "NodeId": "a9eca47661a64031",
  "WWN": "iqn.2013-10.com.vstorage:test3",
  "Portals": [
    {
      "Addr": "192.168.10.13",
      "Port": 3025
    }
  ]
}
]
]

```

В этом файле конфигурации:

- `NodeId` — идентификатор сервера, который можно получить из каталога `/etc/vstorage/host_id` на сервере.
- `WWN` — уникальный идентификатор группы целевых устройств:
 - IQN, если используется протокол iSCSI, например `iqn.2013-10.com.vstorage:test1` (можно изменить только последнюю часть после двоеточия), или
 - WWPN в формате NAA, если используется протокол Fibre Channel, например `naa.21000024ff586d3b` (номер порта можно получить из каталога `/sys/class/fc_host/host6/port_name`).
- `Portals` — один или несколько порталов целевых устройств, комбинации IP-адреса и порта, через которые будет доступно целевое устройство. IP-адрес `Addr` принадлежит интерфейсу внешней сети на сервере, который обрабатывает тип трафика iSCSI. Порт `Port` указывать необязательно, по умолчанию используется 3260.

После создания файла конфигурации, например `tg1.json`, можно создать группу целевых устройств с помощью команды `vstorage-target tg-create`. Например, чтобы создать группу целевых устройств iSCSI, выполните команду:

```

# vstorage-target tg-create -name tg1 -targets tg1.json -type ISCSI
{
  "Id": "3d8364f5-b830-4211-85af-3a19d30ebac4"
}

```

В результате выполнения команды целевые устройства будут созданы на серверах, указанных в файле конфигурации, и присоединены к группе целевых устройств. Также будут созданы порталы целевых устройств на указанных сетевых интерфейсах и портах.

7.3.2 Запуск и остановка групп целевых устройств

При создании группы ее целевые устройства изначально остановлены. Их можно запустить с помощью команды `vstorage-target tg-start`. Например:

```
# vstorage-target tg-start -id 3d8364f5-b830-4211-85af-3a19d30ebac4
```

Эта команда запускает все целевые устройства в группе с идентификатором `3d8364f5-b830-4211-85af-3a19d30ebac4`.

Все целевые устройства в группе могут быть либо запущены, либо остановлены, поэтому при добавлении целевых устройств в группу уже запущенных новые устройства запускаются автоматически.

Чтобы остановить группу целевых устройств, используйте команду `vstorage-target tg-stop`. Например:

```
# vstorage-target tg-stop -id 3d8364f5-b830-4211-85af-3a19d30ebac4
```

7.3.3 Вывод списка групп целевых устройств

Чтобы вывести список групп целевых устройств, можно использовать команду `vstorage-target tg-list`, которая отображает базовую информацию о группах. Например:

```
# vstorage-target tg-list
[
  {
    "Id": "3d8364f5-b830-4211-85af-3a19d30ebac4",
    "Name": "tg1",
    "Type": "ISCSI",
    "Running": true,
    "ACL": false,
    "ChapAuth": false,
    "CHAP": {},
    "Mode": 0
  },
  {
    "Id": "78c3b51e-fd9a-485b-91ce-bc0a8171c89d",
    "Name": "tg2",
```

```

    "Type": "ISCSI",
    "Running": false,
    "ACL": false,
    "ChapAuth": false,
    "CHAP": {},
    "Mode": 0
  }
]

```

Чтобы напечатать полную информацию обо всех группах целевых устройств, используйте команду `vstorage-target tg-list -all`.

7.3.4 Печать сведений о группах целевых устройств

Чтобы напечатать сведения об определенной группе целевых устройств, используйте команду `vstorage-target tg-status`. Например:

```
# vstorage-target tg-status -id faeacacd-eba6-416c-9a7f-b5ba9e372e16
```

Эта команда выводит полные сведения о группе целевых устройств с идентификатором `faeacacd-eba6-416c-9a7f-b5ba9e372e16`. Обратите внимание на параметр `NodeState`. Он указывает, синхронизирован ли сервер с группой целевых устройств, то есть имеет сведения о ее текущей конфигурации. Могут отображаться следующие состояния:

- `synced`, сервер синхронизирован с группой целевых устройств.
- `syncing`, сервер синхронизируется с группой целевых устройств.
- `failed`, серверу не удалось синхронизироваться с группой целевых устройств (сведения см. в параметре `Error`).
- `offline`, сервер не в сети.
- `disabled`, сервер отключен, и его целевое устройство не в сети.

7.3.5 Управление постоянным резервированием групп целевых устройств

Резервирование SCSI-2 позволяет инициаторам получить эксклюзивный доступ к LUN и одновременно предотвращает изменения этого LUN другими инициаторами. Такое резервирование обычно снимается инициатором после внесения изменений в LUN. Однако оно также снимается при

сбоях инициатора или сбросе логических единиц. В SCSI-3 вводится постоянное резервирование, которое сохраняется в случае сбоя или сброса и снимается самим инициатором при необходимости. Оно также позволяет нескольким инициаторам организованно обращаться к LUN.

В продукте Acronis Инфраструктура постоянное резервирование используется в основном для поддержки серверов Microsoft Hyper-V, работающих в отказоустойчивых кластерах.

Постоянное резервирование SCSI включено по умолчанию. Его можно включить или отключить для всех томов в группе целевых устройств следующим образом:

```
# vstorage-target tg-pr -id <tg_ID> -enable  
# vstorage-target tg-pr -id <tg_ID> -disable
```

где <tg_ID> — идентификатор группы целевых устройств, для которой задается постоянное резервирование.

Примечание: Для работы постоянного резервирования сервис `vstorage-target-manager` должен быть запущен на всех серверах MDS.

7.3.6 Удаление групп целевых устройств

Чтобы удалить группу целевых устройств, используйте команду `vstorage-target tg-delete`. Например:

```
# vstorage-target tg-delete -id 3d8364f5-b830-4211-85af-3a19d30ebac4
```

Эта команда удаляет группу целевых устройств с идентификатором `3d8364f5-b830-4211-85af-3a19d30ebac4`.

7.4 Управление томами iSCSI

В этом разделе описано, как создать тома для экспорта через iSCSI и управлять ими.

7.4.1 Создание томов iSCSI

Чтобы создать том, используйте команду `vstorage-target vol-create`. Например:

```
# vstorage-target vol-create -name vol1 -size 1G \  
-vstorage-attr "replicas=3:2 failure-domain=host tier=0"  
{  
  "Id": "3277153b-5296-49c5-9b66-4c200ddb343d"  
}
```

Эта команда создает том размером 1 ГБ с именем `vol1` на уровне хранилища 0, репликацией 3:2 и хостом в качестве области отказа.

7.4.2 Вывод и печать сведений о томах iSCSI

Чтобы вывести список томов, используйте команду `vstorage-target vol-list`. Например:

```
# vstorage-target vol-list  
[  
  "3277153b-5296-49c5-9b66-4c200ddb343d",  
  "a12110d5-cbbc-498a-acdd-a8567286f927",  
  "d5cc3c13-cfb4-4890-a20d-fb80e2a56278"  
]
```

Используйте команду `vstorage-target vol-stat -all` для печати сведений обо всех томах. Чтобы напечатать сведения о конкретном томе, выполните команду `vstorage-target vol-stat -id <vol_ID>`.

7.4.3 Присоединение томов iSCSI к группам целевых устройств

Чтобы присоединить том к группе целевых устройств, используйте команду `vstorage-target tg-attach`. Том нельзя присоединить к нескольким группам целевых устройств одновременно. Например:

```
# vstorage-target tg-attach -id 3d8364f5-b830-4211-85af-3a19d30ebac4 \
-volume 3277153b-5296-49c5-9b66-4c200ddb343d -lun 0 -node bbfd0e7a26b1406d
```

Эта команда присоединяет том с идентификатором 3277153b-5296-49c5-9b66-4c200ddb343d к группе целевых устройств с идентификатором 3d8364f5-b830-4211-85af-3a19d30ebac4 как LUN 0. Нумерация идентификаторов LUN должна начинаться с 0. Эта же команда задает бит PREFERRED для сервера с идентификатором bbfd0e7a26b1406d. Активный/оптимизированный путь по умолчанию будет идти через этот сервер.

7.4.4 Установка активного/оптимизированного пути для томов iSCSI

Чтобы задать активный/оптимизированный путь для тома iSCSI, используйте команду `vstorage-target vol-set`. Она будет работать, только если указанный сервер находится в состоянии STABLE.

Примечание: Убедитесь, что новый предпочитаемый сервер доступен для инициатора.

```
# vstorage-target vol-set -id 3d8364f5-b830-4211-85af-3a19d30ebac4 \
-pref-node bbfd0e7a26b1406d
```

Эта команда устанавливает активный/оптимизированный путь для тома с идентификатором 3d8364f5-b830-4211-85af-3a19d30ebac4 через сервер с идентификатором bbfd0e7a26b1406d.

7.4.5 Просмотр информации ALUA для тома iSCSI

Чтобы просмотреть информацию ALUA для тома iSCSI, используйте команду `vstorage-target vol-info`. Например:

```
# vstorage-target vol-info -id 3d8364f5-b830-4211-85af-3a19d30ebac4
Volume ID:      3d8364f5-b830-4211-85af-3a19d30ebac4
Name:           vol1
Size:           1073741824
Used:           1073152
Serial:         d2be0e84fd7f
Attrs:          map[ ]
TG:             4708b908-8c2d-444c-91b1-a1e18a96d4fc
LUN:            0
```

```
*** Node #0 ***
-----
```

```

NodeId:                bbfd0e7a26b1406d
State:                  synced
TPGs:                   vstorage_tpg_0
ALUA:                   active
Preferred:              1
WWNs:                   iqn.2014-06.com.vstorage:target1 [2]
Portals:                10.37.130.61

```

Эта команда отображает сведения ALUA для тома с идентификатором 3d8364f5-b830-4211-85af-3a19d30ebac4.

7.4.6 Просмотр и настройка параметров тома iSCSI

Чтобы просмотреть и настроить параметры тома, такие как режим избыточности, область отказа или уровень хранилища, используйте команды `vstorage-target vol-attr get` и `vstorage-target vol-attr set` соответственно. Например:

```

# vstorage-target vol-attr get -id d5cc3c13-cfb4-4890-a20d-fb80e2a56278
{
  "chunk-size": "268435456",
  "client-ssd-cache": "1",
  "failure-domain": "host",
  "replicas": "3:2",
  "tier": "0"
}
# vstorage-target vol-attr set -id d5cc3c13-cfb4-4890-a20d-fb80e2a56278 \
-vstorage-attr "replicas=2:1 tier=1"

```

7.4.7 Увеличение размера тома iSCSI

Чтобы увеличить размер тома, используйте команду `vstorage-target vol-grow`. Например:

```
# vstorage-target vol-grow -id d5cc3c13-cfb4-4890-a20d-fb80e2a56278 -size 2G
```

7.4.8 Установка ограничений для тома iSCSI

Чтобы задать ограничения на чтение/запись для тома, используйте команду `vstorage-target vol-limits`. Например:

```
# vstorage-target vol-limits -id d5cc3c13-cfb4-4890-a20d-fb80e2a56278 -read-bps 10485760 \
-write-bps 10485760
```


Эта команда задает для тома с идентификатором d5cc3c13-cfb4-4890-a20d-fb80e2a56278 скорость чтения/записи 10 485 760 байт в секунду.

7.4.9 Отсоединение томов iSCSI от групп целевых устройств

Чтобы отсоединить том от группы целевых устройств, используйте команду `vstorage-target tg-detach`. LUN 0 должен быть отсоединен последним. Например:

```
# vstorage-target tg-detach -id 3d8364f5-b830-4211-85af-3a19d30ebac4 \  
-volume d5cc3c13-cfb4-4890-a20d-fb80e2a56278
```

Эта команда отсоединяет том с идентификатором d5cc3c13-cfb4-4890-a20d-fb80e2a56278 от целевой группы с идентификатором 3d8364f5-b830-4211-85af-3a19d30ebac4.

7.4.10 Удаление томов iSCSI

Для удаления тома используйте команду `vstorage-target vol-delete`. Тома, присоединенные к группам целевых устройств, удалить нельзя. Например:

```
# vstorage-target vol-delete -id d5cc3c13-cfb4-4890-a20d-fb80e2a56278
```

Эта команда удаляет том с идентификатором d5cc3c13-cfb4-4890-a20d-fb80e2a56278.

7.5 Управление серверами в группах целевых устройств

В этом разделе описывается управление серверами применительно к группам целевых устройств.

7.5.1 Добавление серверов в группы целевых устройств

Чтобы добавить сервер в группу целевых устройств, создайте файл конфигурации со сведениями о WWN и портале целевого устройства. Целевое устройство будет создано автоматически на добавленном сервере. Один сервер можно добавить в несколько групп целевых устройств, а его сетевые интерфейсы могут использоваться одновременно несколькими целевыми устройствами из разных групп.

Например:

```
# vstorage-target node-add -node bbfd0e7a26b1406d -tg 3d8364f5-b830-4211-85af-3a19d30ebac4
-targets target.json
```

Эта команда добавляет сервер с идентификатором `bbfd0e7a26b1406d` в группу целевых устройств с идентификатором `3d8364f5-b830-4211-85af-3a19d30ebac4`. Она также создает на нем целевое устройство в соответствии с файлом конфигурации `target.json`, который выглядит следующим образом.

```
[
  {
    "NodeId": "bbfd0e7a26b1406d",
    "WWN": "iqn.2013-10.com.vstorage:test2",
    "Portals": [
      {
        "Addr": "10.94.104.89",
        "Port": 3260
      }
    ]
  }
]
```

7.5.2 Вывод списка серверов в группах целевых устройств

Чтобы вывести список всех серверов во всех группах целевых устройств и подробную информацию о них, используйте команду `vstorage-target node-list`. Например:

```
# vstorage-target node-list
[
  {
    "ID": "bbfd0e7a26b1406d",
    "Status": "STABLE",
    "Enabled": true,
    "MonitorOnline": true,
    "Version": "7.10.32",
    "Address": "10.94.104.89:40135",
    "ActiveVolumes": [
      "0937f0e3-91a9-4dfc-8c10-6202bdc792c8"
    ]
  }
]
```

7.5.3 Включение и отключение серверов в группах целевых устройств

Чтобы включить или отключить сервер во всех группах целевых устройств, которым он принадлежит, используйте команду `vstorage-target node-set`. При включении сервера его целевые устройства запускаются, а при отключении они останавливаются и активный путь перемещается на другой сервер. Эти же операции выполняются, когда сервер выходит из режима обслуживания и входит в него.

Например, чтобы включить сервер с идентификатором `bbfd0e7a26b1406d`, выполните команду

```
# vstorage-target node-set -node bbfd0e7a26b1406d -enable
```

Перед отключением сервера убедитесь в наличии других серверов со статусом `STABLE`, куда можно переместить активный/оптимизированный путь. В противном случае произойдет ошибка ввода-вывода.

Чтобы отключить сервер с идентификатором `bbfd0e7a26b1406d`, выполните команду

```
# vstorage-target node-set -node bbfd0e7a26b1406d -disable
```

Статус узла можно проверить с помощью команды `vstorage-target node-list`, см. раздел [Вывод списка серверов в группах целевых устройств](#) (страница 237).

7.5.4 Удаление серверов из групп целевых устройств

Чтобы удалить сервер из группы целевых устройств, используйте команду `vstorage-target node-del`. Сервер можно удалить, только если он не находится на активном/оптимизированном пути. Иначе необходимо перенести путь на другой сервер с помощью отключения этого сервера (см. раздел [Включение и отключение серверов в группах целевых устройств](#) (страница 238)) или вручную (см. раздел [Установка активного/оптимизированного пути для томов iSCSI](#) (страница 234)).

```
# vstorage-target node-del -tg 3d8364f5-b830-4211-85af-3a19d30ebac4 -node bbfd0e7a26b1406d
```

Эта команда удаляет сервер с идентификатором `bbfd0e7a26b1406d` из целевой группы с идентификатором `3d8364f5-b830-4211-85af-3a19d30ebac4`.

7.6 Управление целевыми устройствами и порталами

В этом разделе описывается, как создавать целевые устройства и управлять ими.

Оптимальный способ — создать по одному целевому устройству на сервер при использовании протокола iSCSI и по одному целевому устройству на порт FC при использовании протокола FC.

7.6.1 Создание целевых устройств

Как правило, целевые устройства создаются автоматически при создании групп или добавлении в них серверов. Однако, поскольку можно удалить целевые устройства с сервера, не удаляя его из группы, на таком сервере можно снова создать целевые устройства. Используйте команду `vstorage-target target-create`. Например:

```
# vstorage-target target-create -tg 3d8364f5-b830-4211-85af-3a19d30ebac4 -json target.json
```

Эта команда создает целевое устройство на базе файла конфигурации `target.json` в группе целевых устройств с идентификатором `3d8364f5-b830-4211-85af-3a19d30ebac4`. В файле конфигурации перечислены сведения о целевом устройстве, такие как сервер для создания, WWN и портал.

Например:

```
{
  "NodeId": "bbfd0e7a26b1406d",
  "WWN": "iqn.2013-10.com.vstorage:test22",
  "Portals": [
    {
      "Addr": "10.94.104.90",
      "Port": 3260
    }
  ]
}
```

7.6.2 Добавление и удаление порталов целевых устройств

Чтобы добавить портал для целевого устройства, используйте команду `vstorage-target target-portal add`. Например:

```
# vstorage-target target-portal add -wwn iqn.2013-10.com.vstorage:test2 -addr 10.94.104.90  
-tg 3d8364f5-b830-4211-85af-3a19d30ebac4
```

Эта команда добавляет портал с IP-адресом 10.94.104.90 и стандартный порт 3260 для целевого устройства с IQN `iqn.2013-10.com.vstorage:test2` в группе целевых устройств с идентификатором `3d8364f5-b830-4211-85af-3a19d30ebac4`.

Чтобы удалить портал для целевого устройства, используйте команду `vstorage-target target-portal del`. Например:

```
# vstorage-target target-portal del -wwn iqn.2013-10.com.vstorage:test2 -addr 10.94.104.90  
-tg 3d8364f5-b830-4211-85af-3a19d30ebac4
```

Эта команда удаляет ранее созданный портал.

7.6.3 Удаление целевых устройств

Чтобы удалить целевое устройство из группы (а также с сервера), используйте команду `vstorage-target target-delete`. Например:

```
# vstorage-target target-delete -tg 3d8364f5-b830-4211-85af-3a19d30ebac4 \  
-wwn iqn.2013-10.com.vstorage:test22
```

Эта команда удаляет целевое устройство с IQN `iqn.2013-10.com.vstorage:test22` из группы с идентификатором `3d8364f5-b830-4211-85af-3a19d30ebac4`, а также с сервера, на котором оно расположено.

Сервер, на котором не осталось целевых устройств, удаляется из группы.

7.7 Управление учетными записями CHAP

Протокол CHAP (Challenge-Handshake Authentication Protocol) предоставляет способ ограничить доступ к целевым устройствам и их LUN, требуя от инициатора имя пользователя и пароль. Учетные записи CHAP применяются к целым группам целевых устройств. Группы целевых устройств Fibre Channel не используют CHAP.

Чтобы использовать авторизацию CHAP, включите ее для группы целевых устройств.

```
# vstorage-target tg-auth -enable-chap -id <tg_ID>
```

7.7.1 Создание и вывод списка учетных записей CHAP

Чтобы создать учетную запись CHAP, используйте команду `vstorage-target account-create`.

Например:

```
# vstorage-target account-create -user user1 -desc "User for TG1"  
Enter Password:
```

Пароль должен содержать от 12 до 16 символов.

Чтобы вывести список существующих учетных записей CHAP и сведения о них, используйте команду `vstorage-target account-list`.

7.7.2 Изменение сведений учетной записи CHAP

Чтобы изменить пароль или описание учетной записи CHAP, используйте команду `vstorage-target account-set`. Например:

```
# vstorage-target account-set description -user user1 -desc "A new description"  
# vstorage-target account-set password -user user1  
Enter Password:
```

7.7.3 Назначение учетных записей CHAP группам целевых устройств

Чтобы назначить учетную запись CHAP группе целевых устройств, используйте команду `vstorage-target tg-chap`. Например:

```
# vstorage-target tg-chap set -id faeacacd-eba6-416c-9a7f-b5ba9e372e16 -user user1
```

Чтобы удалить назначение, выполните команду

```
# vstorage-target tg-chap del -id faeacacd-eba6-416c-9a7f-b5ba9e372e16 -user user1
```

7.7.4 Удаление учетных записей CHAP

Чтобы удалить неиспользуемую учетную запись CHAP, используйте команду `vstorage-target account-delete`. Например:

```
# vstorage-target account-delete -user user1
```

7.8 Управление представлениями LUN

Представления LUN позволяют создавать и поддерживать список контроля доступа (ACL), который ограничивает доступ к выбранным LUN для определенных инициаторов. Инициаторы, отсутствующие в списке, имеют доступ ко всем LUN в группах целевых устройств iSCSI. Однако тома, экспортированные через группы целевых устройств Fibre Channel, доступны только инициаторам, добавленным в список ACL группы.

Чтобы использовать авторизацию на базе ACL, включите ее для группы целевых устройств.

```
# vstorage-target tg-auth -enable-acl -id <tg_ID>
```

7.8.1 Создание представлений LUN

Чтобы создать представление LUN для инициатора, используйте команду `vstorage-target tg-initiator add` или `vstorage-target view-add`. Первая команда добавляет инициатор в список ACL группы целевых устройств и создает для него представление. Вторая команда добавляет представления для инициаторов, которые уже есть в списке.

Например:

```
# vstorage-target tg-initiator add -alias initiator2 -luns 0,1 \  
-tg ee764519-80e3-406e-b637-8d63712badf1 -wwn iqn.1994-05.com.redhat:1535946874d
```

Эта команда добавляет инициатор с IQN `iqn.1994-05.com.redhat:1535946874d` в список ACL группы целевых устройств с идентификатором `ee764519-80e3-406e-b637-8d63712badf1` и создает представление, разрешающее инициатору доступ к LUN с идентификаторами 0 и 1.

Другой пример:

```
# vstorage-target view-add -tg faeacacd-eba6-416c-9a7f-b5ba9e372e16 -lun 2 -map 2 \  
-wwn iqn.1994-05.com.redhat:1535946874d
```

Эта команда добавляет для этого же инициатора представление, также разрешающее ему доступ к LUN 2.

7.8.2 Вывод списка представлений LUN

Чтобы вывести список представлений LUN для инициатора, используйте команду `vstorage-target view-list`. Например:

```
# vstorage-target view-list -tg ee764519-80e3-406e-b637-8d63712badf1 \  
-wwn iqn.1994-05.com.redhat:1535946874d
```

Эта команда выводит список представлений для инициатора с IQN `iqn.1994-05.com.redhat:1535946874d`.

7.8.3 Изменение сведений о представлениях LUN

Чтобы изменить представления LUN для инициатора, используйте команду `vstorage-target view-set`. Например:

```
# vstorage-target view-set -luns 1 -tg ee764519-80e3-406e-b637-8d63712badf1 \  
-wwn iqn.1994-05.com.redhat:1535946874d
```

Эта команда разрешает инициатору с IQN `iqn.1994-05.com.redhat:1535946874d` доступ только к LUN 1. По сути, она удаляет все представления LUN для инициатора, исключая указанное.

7.8.4 Удаление представлений LUN

Чтобы удалить представление LUN для инициатора, используйте команду `vstorage-target view-del`. Например:

```
# vstorage-target view-del -lun 1 -tg ee764519-80e3-406e-b637-8d63712badf1 \  
-wwn iqn.1994-05.com.redhat:1535946874d
```

Эта команда удаляет представление LUN 1 для инициатора с IQN `iqn.1994-05.com.redhat:1535946874d`.

ГЛАВА 8

Расширенные задачи

В этой главе описываются различные задачи по настройке и управлению.

8.1 Обновление ядра с помощью ReadyKernel

ReadyKernel — это сервис на базе `kpatch`, поставляемый с продуктом Acronis Инфраструктура, который доступен в готовом виде на физических серверах с активными лицензиями. По сравнению с обычным способом обновления ядра ReadyKernel предоставляет более удобную альтернативу без необходимости перезагрузки, что позволяет не ждать планового отключения сервера для применения критических обновлений безопасности. ReadyKernel позволяет получать накопительные исправления ядра, устраняющие критические проблемы безопасности, и применять их без перезагрузки сервера. Обновления ReadyKernel выпускаются для ядер младше 18 месяцев. Когда ядро становится старше 18 месяцев, необходимо перейти на более новое ядро, чтобы продолжить получать обновления ReadyKernel.

При установке исправления загружаются в ОЗУ сервера и сразу применяются к ядру. Если сервер перезагружается, эти исправления снова применяются к ядру при загрузке. Сведения о примененном исправлении ReadyKernel можно в любое время проверить с помощью команды `readykernel info`.

Если позже вы установите новое ядро или крупное обновление ядра, требующее перезагрузки, то скачанные исправления останутся на сервере, но не будут применены.

В продукте Acronis Инфраструктура функция ReadyKernel настроена на автоматическое скачивание и применение обновлений. Проверка наличия новых исправлений добавляется к каждой транзакции `um`, которая происходит на любом сервере в инфраструктуре.

Хотя по умолчанию ReadyKernel не требует никаких действий пользователя, вы можете ознакомиться со следующими подразделами, чтобы понять, как работает этот инструмент, и управлять им при необходимости.

8.1.1 Автоматическая установка исправлений ReadyKernel

По умолчанию функция ReadyKernel включена и проверяет наличие новых исправлений ежедневно в 12:00 по времени сервера посредством скрипта `cron.d`. Если исправление доступно, ReadyKernel скачает, установит и загрузит его для текущего ядра.

Чтобы отключить автоматическое обновление, выполните команду

```
# readykernel autoupdate disable
```

Автоматическое обновление можно снова включить позже с помощью следующей команды:

```
# readykernel autoupdate enable <hour>
```

Сервис будет проверять наличие исправлений ежедневно в указанное время `<hour>` (серверное время в 24-часовом формате).

8.1.2 Управление исправлениями ReadyKernel вручную

8.1.2.1 Скачивание, установка и загрузка исправлений ReadyKernel

Чтобы скачать, установить и сразу загрузить исправление ReadyKernel для текущего ядра, выполните следующие действия.

1. Проверьте наличие новых исправлений ReadyKernel.

```
# readykernel check-update
```

2. Если доступно новое исправление, скачайте, установите и сразу загрузите его для текущего ядра, выполнив команду:

```
# readykernel update
```

Исправления ReadyKernel являются накопительными, то есть последнее исправление включает в себя все предыдущие. Для поддержки защиты ядра достаточно установить и загрузить последнее исправление.

8.1.2.2 Загрузка и выгрузка исправлений ReadyKernel

Чтобы вручную загрузить в ядро последнее установленное исправление ReadyKernel, выполните одно из следующих действий.

- Если уже загружено более старое исправление, сначала выгрузите его, а затем загрузите последнее исправление, выполнив команду:

```
# readykernel load-replace
```

- Если более старые исправления отсутствуют, загрузите последнее исправление, выполнив команду:

```
# readykernel load
```

Чтобы выгрузить исправление из текущего ядра, выполните команду

```
# readykernel unload
```

8.1.2.3 Установка и удаление исправлений ReadyKernel для определенных ядер

Если на сервере несколько ядер, можно установить исправление ReadyKernel для определенного ядра.

```
# yum install readykernel-patch-<kernel_version>
```

Чтобы удалить определенное исправление ReadyKernel с сервера, выполните команду

```
# yum remove readykernel-patch-<kernel_version>
```

8.1.2.4 Понижение версии исправлений ReadyKernel

При возникновении проблем с последним исправлением ReadyKernel можно перейти на более раннюю версию, если она доступна.

Чтобы понизить версию исправления до предыдущей для текущего ядра, выполните команду

```
# yum downgrade readykernel-patch-$(uname -r)
```

Чтобы понизить версию исправления до предыдущей для определенного ядра, выполните команду

```
# yum downgrade readykernel-patch-<kernel_version>
```

Эти команды можно выполнить несколько раз, чтобы понизить версию исправления до нужной. Либо можно перейти на конкретную версию исправления, указав ее. Например:

```
# yum downgrade readykernel-patch-12.7-0.4-17.v17
```

8.1.2.5 Отключение загрузки исправлений ReadyKernel при загрузке машины

Если по какой-то причине вы не хотите, чтобы исправления ReadyKernel применялись во время загрузки, выполните следующую команду:

```
# readykernel autoload disable
```

Чтобы снова включить автоматическую загрузку исправлений ReadyKernel при загрузке машины, выполните команду

```
# readykernel autoload enable
```

8.1.2.6 Управление журналами ReadyKernel

ReadyKernel записывает информацию о событиях в `/var/log/messages` и `/var/log/kpatch.log`. Для последнего можно указать параметры ведения журнала в файле конфигурации `/etc/logrotate.d/kpatch`. Дополнительные сведения о параметрах, которые можно использовать, см. на справочной странице `logrotate`.

8.2 Управление дополнениями гостевой ОС

В этом разделе описывается, как установить и удалить дополнения гостевой ОС. Этих функций требует *Выполнение команд в виртуальных машинах без сетевого подключения* (страница 253).

8.2.1 Установка дополнений гостевой ОС

Чтобы установить дополнения гостевой ОС в виртуальные машины, сначала необходимо создать и загрузить вычислительные образы из прилагаемых ISO-файлов дополнений гостевой ОС, расположенных в каталоге `/usr/share/vz-guest-tools/`. Выполните следующие команды на одном из вычислительных серверов:

- Для дополнений гостевой ОС Linux:

```
# vinfra service compute image create vz-guest-tools-lin \
--file /usr/share/vz-guest-tools/vz-guest-tools-lin.iso --os-distro linux
Uploading image to server [Elapsed Time: 0:00:05] ...
```

- Для дополнений гостевой ОС Windows:

```
# vinfra service compute image create vz-guest-tools-win \
--file /usr/share/vz-guest-tools/vz-guest-tools-win.iso --os-distro windows
Uploading image to server [Elapsed Time: 0:00:09] ...
```

Далее необходимо присоединить созданный образ к ВМ и запустить установщик дополнений гостевой ОС. Эти шаги различаются для новых и уже существующих ВМ и описываются в следующих подразделах.

8.2.1.1 Установка дополнений гостевой ОС в новые ВМ

При создании новой ВМ к ней можно присоединить образ дополнений гостевой ОС и установить дополнения после операционной системы. Для этого выполните следующие шаги на вычислительном сервере.

1. Создайте новую ВМ с образом дополнений гостевой ОС. Например, чтобы создать ВМ Linux centos, выполните команду:

```
# vinfra service compute server create centos --network id=private --flavor medium \
--volume source=blank,size=64,boot-index=0,type=disk \
--volume source=image,id=centos7,size=3,boot-index=1,type=cdrom \
--volume source=image,id=vz-guest-tools-lin,size=1,boot-index=2,type=cdrom
```

Примечание: Округлите размер томов, создаваемых из образов, в большую сторону. Например, если образ дистрибутива ОС имеет размер 2,6 ГБ, используйте `size=3`.

В этом примере первый том — это пустой виртуальный жесткий диск, второй том — образ дистрибутива ОС centos7, а третий том — образ дополнений гостевой ОС vz-guest-tools-lin. Не забудьте указать правильный порядок загрузки с помощью параметра `boot-index`.

2. Выполните вход на виртуальную машину и установите на ней операционную систему.
3. Запустите установщик дополнений гостевой ОС внутри ВМ.
 - Внутри ВМ Linux создайте точку подключения для оптического диска с образом дополнений гостевой ОС и запустите установщик.

```
# mkdir /mnt/cdrom
# mount /dev/sr1 /mnt/cdrom
# bash /mnt/cdrom/install
```

- Внутри VM Windows запустите установщик в окне автозапуска, если он включен. В противном случае откройте оптический диск в проводнике и запустите файл `setup.exe`.
После установки дополнений гостевой ОС перезапустите VM.

Примечание: Guest tools rely on the QEMU guest agent that is installed alongside the tools. The agent service must be running for the tools to work.

8.2.1.2 Установка дополнений гостевой ОС в существующие VM

Шаги по установке дополнений в существующие VM зависят от типа гостевой ОС. Они описываются в следующих подразделах.

8.2.1.2.1 Установка дополнений гостевой ОС в существующие VM Linux

Чтобы установить дополнения гостевой ОС в существующую виртуальную машину Linux, выполните следующие действия.

1. Создайте том из загруженного образа дополнений гостевой ОС. Например:

```
# vinfra service compute volume create vz-guest-tools-lin-vol --storage-policy default
--size 1 --image vz-guest-tools-lin
```

2. Присоедините том дополнений гостевой ОС к виртуальной машине. Например:

```
# vinfra service compute server volume attach \
--server centos vz-guest-tools-lin-vol
+-----+-----+
| Field | Value |
+-----+-----+
| device | /dev/sr1 |
| id      | 1a40012a-7976-47a1-81f1-ff498cba90af |
+-----+-----+
```

3. Выполните вход на виртуальную машину, создайте точку подключения для оптического диска с образом дополнений гостевой ОС и запустите установщик.

```
# mkdir /mnt/cdrom
# mount /dev/sr1 /mnt/cdrom
# bash /mnt/cdrom/install
```

Примечание: Guest tools rely on the QEMU guest agent that is installed alongside the tools. The agent service must be running for the tools to work.

8.2.1.2.2 Установка дополнений гостевой ОС в существующие VM Windows

Чтобы установить дополнения гостевой ОС в существующую виртуальную машину Windows, выполните следующие действия.

1. Выключите VM Windows. Например, для остановки VM win10 выполните команду:

```
# vinfra service compute server stop win10
```

2. Преобразуйте системный диск VM в шаблон образа. Вам потребуется идентификатор тома, который можно получить с помощью команды `vinfra service compute volume list`. Например, чтобы использовать загрузочный том VM win10, выполните команду:

```
# vinfra service compute volume list | grep win10
| 7116d747-a1e1-4200-bd4a-25cc51ef006c | win10/windows_10_pro_x64.iso/Boot volume |
| ef2f1979-7811-4df6-9955-07e2fc942858 | win10/windows_10_pro_x64.iso/CD/DVD volume |
# vinfra service compute volume upload-to-image 7116d747-a1e1-4200-bd4a-25cc51ef006c |
| id | 79da5239-b2bb-4779-ada2-46cb8da8ba0e
```

3. Создайте новую VM Windows из шаблона, присоединив к ней образ дополнений гостевой ОС во время создания. Например:

```
# vinfra service compute server create newvm --network id=private --flavor medium \
--volume source=image,id=79da5239-b2bb-4779-ada2-46cb8da8ba0e,size=64,boot-index=0,type=
--volume source=image,id=vz-guest-tools-win,size=1,boot-index=1,type=cdrom
```

Примечание: Размер тома, создаваемого из шаблона образа, должен быть равен минимальному размеру тома, указанному в метаданных образа, или больше него. Чтобы узнать минимальный размер тома, используйте команду `vinfra service compute image show <image_id> | grep min_disk`.

В этом примере первый том — это шаблон системного диска исходной VM, а второй том — образ дополнений гостевой ОС. Не забудьте указать правильный порядок загрузки с помощью параметра `boot-index`.

4. После того как образ будет подключен внутри VM Windows, запустите установщик в окне автозапуска, если автозапуск включен. В противном случае откройте оптический диск в проводнике и запустите файл `setup.exe`.

После установки дополнений гостевой ОС перезапустите VM.

Примечание: Guest tools rely on the QEMU guest agent that is installed alongside the tools. The agent service must be running for the tools to work.

8.2.2 Удаление дополнений гостевой ОС

Шаги по удалению дополнений зависят от типа гостевой ОС и описываются в следующих разделах.

8.2.2.1 Удаление дополнений гостевой ОС из VM Linux

Чтобы удалить дополнения из гостевой ОС Linux, выполните вход в виртуальную машину, а затем выполните следующие действия.

1. Удалите пакеты.

- 1.1. В системах на базе RPM (CentOS и др.):

```
# yum remove dkms-vzvirtio_balloon prl_nettool qemu-guest-agent-vz vz-guest-udev
```

- 1.2. В системах на базе DEB (Debian и Ubuntu):

```
# apt-get remove vzvirtio-balloon-dkms prl-nettool qemu-guest-agent-vz vz-guest-ud
```

Если какие-либо из перечисленных выше пакетов не установлены в системе, выполнение команды завершится ошибкой. В этом случае исключите эти пакеты из команды и выполните ее снова.

2. Удалите файлы.


```
# rm -f /usr/bin/prl_backup /usr/share/qemu-ga/VERSION /usr/bin/install-tools \
/etc/udev/rules.d/90-guest_iso.rules /usr/local/bin/fstrim-static /etc/cron.weekly/fst
```

3. Перезагрузите правила udev.

```
# udevadm control --reload
```

После удаления дополнений гостевой ОС перезапустите виртуальную машину.

8.2.2.2 Удаление дополнений гостевой ОС из VM Windows

Чтобы удалить дополнения гостевой ОС для Windows, выполните вход в виртуальную машину, а затем выполните следующие действия.

1. Remove the QEMU device drivers from the device manager.

Важно: Не удаляйте драйвер жесткого диска VirtIO/SCSI и сетевой драйвер NetKVM. Без первого драйвера VM не будет загружаться, а без второго потеряет возможность подключения к сети.

2. Uninstall the QEMU guest agent and guest tools from the list of installed applications.
3. Остановите и удалите Guest Tools Monitor.

```
> sc stop VzGuestToolsMonitor
> sc delete VzGuestToolsMonitor
```

4. Отмените регистрацию Guest Tools Monitor в журнале событий.

```
> reg delete HKLM\SYSTEM\CurrentControlSet\services\eventlog\Application\
VzGuestToolsMonitor
```

5. Удалите раздел реестра для автозапуска RebootNotifier.

```
> reg delete HKLM\SOFTWARE\Microsoft\Windows\CurrentVersion\Run /v \
VzRebootNotifier
```

6. Удалите папку C:\Program Files\Qemu-ga\.

Если файл VzGuestToolsMonitor.exe заблокирован, закройте все окна средства просмотра событий. Если файл остается заблокированным, перезапустите службу eventlog.

```
> sc stop eventlog
> sc start eventlog
```

После удаления дополнений гостевой ОС перезапустите виртуальную машину.

8.3 Выполнение команд в виртуальных машинах без сетевого подключения

Если по какой-то причине у VM нет доступа к сети, вы все равно можете выполнять на ней команды с сервера, на котором она расположена. На этой VM должны быть установлены дополнения гостевой ОС (см. раздел *Управление дополнениями гостевой ОС* (страница 247)).

Вам потребуется идентификатор VM, который можно получить с помощью команды `vinfra service compute server list`. Вы также можете использовать имя домена `virsh`, которое можно получить с помощью команды `virsh list`.

8.3.1 Выполнение команд в виртуальных машинах Linux

Чтобы выполнить произвольную команду внутри VM Linux и получить вывод на консоль, используйте команду `virsh x-exec`. Например:

```
# virsh x-exec 1d45a54b-0e20-4d5e-8f11-12c8b4f300db /usr/bin/bash -c 'lsblk'
NAME          MAJ:MIN RM   SIZE RO TYPE MOUNTPOINT
loop0          7:0      0  945.9M 1 loop
loop1          7:1      0     5G  1 loop
  live-rw      253:0    0     5G  0 dm   /
  live-base    253:1    0     5G  1 dm
loop2          7:2      0    32G  0 loop
  live-rw      253:0    0     5G  0 dm   /
sda            8:0      0    64G  0 disk
sdc            8:32     0     1G  1 disk
sr0           11:0     1     2G  0 rom  /run/initramfs/live
```

Чтобы скопировать файл на VM Linux, используйте команды `virsh x-exec` и `cat`. Например:

```
# virsh x-exec 1d45a54b-0e20-4d5e-8f11-12c8b4f300db \
--shell 'cat > test.file' < /home/test.file
```

Чтобы получить файл с VM Linux, используйте те же команды `virsh x-exec` и `cat`. Например:

```
# virsh x-exec 1d45a54b-0e20-4d5e-8f11-12c8b4f300db \
--shell 'cat /home/test.file' > test.file
```

8.3.2 Выполнение команд в виртуальных машинах Windows

Чтобы выполнить произвольную команду внутри VM Windows и получить вывод на консоль, используйте команду `virsh x-exec`. Например:

```
# virsh x-exec bbf4a6ec-865f-4e2c-ac21-8639d1bfb85c --shell dir c:\\
Volume in drive C has no label.
Volume Serial Number is D0BE-A8D1

Directory of c:\

06/10/2009  01:42 PM                24 autoexec.bat
06/10/2009  01:42 PM                10 config.sys
07/13/2009  06:37 PM          <DIR>          PerfLogs
11/12/2018  07:45 AM          <DIR>          Program Files
11/12/2018  07:55 AM          <DIR>          test
11/12/2018  06:23 AM          <DIR>          Users
11/12/2018  07:53 AM          <DIR>          Windows
                2 File(s)                34 bytes
                5 Dir(s)  59,329,495,040 bytes free
```

Чтобы скопировать файл на VM Windows, используйте команды `virsh x-exec` и `prl_cat`. Например:

```
# virsh x-exec bbf4a6ec-865f-4e2c-ac21-8639d1bfb85c \
--shell '%programfiles%\qemu-ga\prl_cat' 'c:\test\test.file' < /home/test.file
```

Чтобы получить файл с VM Windows, используйте команды `virsh x-exec` и `type`. Например:

```
# virsh x-exec bbf4a6ec-865f-4e2c-ac21-8639d1bfb85c \
--shell type 'c:\test\test.file' > test.file
```

8.4 Настройка модели ЦП виртуальных машин

По умолчанию виртуальные машины создаются с моделью ЦП хоста. Если серверы в вычислительном кластере имеют разные ЦП, может не работать динамическая миграция VM либо могут неправильно работать приложения внутри VM, зависящие от конкретных ЦП. Для избежания этого можно определить, какая модель ЦП обеспечивает совместимость по всем серверам в вычислительном кластере, и вручную установить ее для кластера по умолчанию. Однако в этом случае моделью ЦП вычислительного кластера будет наименее продвинутая и вычислительные серверы потеряют возможности более производительного процессора.

Чтобы задать модель ЦП для вычислительного кластера, выполните следующие действия.

1. Выполните команду `virsh capabilities` на каждом сервере для вывода XML-документа с информацией о ЦП сервера. Объедините разделы `<cpu>` из всех XML-документов в один файл, например `cpu-compare.xml`.
2. Сравните характеристики ЦП с помощью команды `virsh cpu-baseline`. Например:

```
# virsh cpu-baseline cpu-compare.xml | grep model
<model fallback='allow'>IvyBridge</model>
```

Команда выведет наиболее совместимую модель ЦП для всех серверов.

3. Установите эту модель ЦП для вычислительного кластера. Например:

```
# vinfra service compute set --cpu-model IvyBridge
```

Обратите внимание на следующие моменты.

- Чтобы вывести список поддерживаемых моделей ЦП, выполните команду `vinfra service compute show`.
- Изменение модели ЦП влияет только на новые ВМ (то есть созданные после изменения).

8.5 Создание шаблонов Linux

Если у вас нет готового шаблона Linux, его можно создать с помощью инструмента `diskimage-builder`. Образ диска создается только с пользователем `root`, у которого нет ни пароля, ни SSH-ключей. Можно использовать методы `user data` и `cloud-init` для выполнения задач начальной конфигурации на ВМ, которые будут развернуты из этого образа, например для создания определенных учетных записей пользователей. Другие параметры для настройки ВМ во время загрузки см. в документации по `cloud-init`.

Чтобы создать шаблон и развернуть из него ВМ, выполните следующие действия.

1. Установите пакет `diskimage-builder`.

```
# yum install diskimage-builder
```

2. Для гостевой ОС RHEL 7 загрузите облачный образ с [клиентского портала Red Hat](#) (требуется вход) и выполните команду:

```
# export DIB_LOCAL_IMAGE=<path_to_rhel7_image>
```

3. Выполните следующую команду, чтобы создать образ диска с установленным пакетом `cloud-init` для нужной гостевой системы Linux. Например:

```
# disk-image-create vm centos7 -t qcow2 -o centos7
```

где

- `centos7` — имя гостевой ОС. Может иметь одно из следующих значений: `centos6`, `centos7`, `debian`, `rhel7` или `ubuntu`.

По умолчанию при использовании варианта `ubuntu` будет создан образ диска для Ubuntu 16.04. Чтобы создать образ для Ubuntu 18.04, добавьте в команду `DIB_RELEASE=bionic` следующим образом: `DIB_RELEASE=bionic disk-image-create vm ubuntu -t qcow2 -o ubuntu18`.

- `-o` задает имя итогового файла образа диска.

4. Загрузите созданный образ диска в вычислительный кластер с помощью инструмента `vinfra`.

```
# vinfra service compute image create centos7-image --os-distro centos7 \
--disk-format qcow2 --file centos7.qcow2
```

где

- `centos7-image` — имя нового образа.
- `centos7` — дистрибутив ОС. Может иметь одно из следующих значений: `centos6`, `centos7`, `debian9`, `rhel7`, `ubuntu16.04` и `ubuntu18.04`.
- `centos7.qcow2` — образ QCOW2, созданный на шаге 3.

5. Создайте файл конфигурации `user-data` с нужной учетной записью пользователя.

```
# cat <<EOF > user-data
#cloud-config
user: myuser
password: password
chpasswd: {expire: False}
ssh_pwauth: True
EOF
```

где `myuser` — имя пользователя, а `password` — пароль для учетной записи.

6. Запустите развертывание VM из образа диска с использованием файла конфигурации в качестве данных пользователя.

```
# vinfra service compute server create centos7-vm --flavor medium --network public \
--user-data user-data --volume source=image,id=centos7-image,size=10
```

где

- `centos7-vm` — имя новой VM.
- `user-data` — файл конфигурации, созданный на шаге 5.
- `centos7-image` — образ, добавленный в вычислительный кластер на шаге 4.

Дополнительные сведения об управлении вычислительными объектами через инструмент `vinfra` см. в разделе *Управление вычислительным кластером* (страница 47).

8.6 Подключение к интерфейсу командной строки OpenStack

Для управления вычислительным кластером можно также использовать клиент командной строки OpenStack, который автоматически устанавливается вместе с продуктом Acronis Инфраструктура.

Чтобы подключиться к интерфейсу командной строки OpenStack и работать с ним, выполните следующие действия.

1. На сервере управления создайте скрипт администратора OpenRC, который устанавливает переменные среды.

```
# kolla-ansible post-deploy
```

Команда создаст `bash`-скрипт `/etc/kolla/admin-openrc.sh`. Например:

```
export OS_PROJECT_DOMAIN_NAME=Default
export OS_USER_DOMAIN_NAME=Default
export OS_PROJECT_NAME=admin
export OS_USERNAME=vstorage-service-user
export OS_PASSWORD=<password>
export OS_AUTH_URL=https://<MN_IP_address>:5000/v3
export OS_IDENTITY_API_VERSION=3
export OS_AUTH_TYPE=password
export OS_INSECURE=true
export PYTHONWARNINGS="ignore:Unverified HTTPS request is being made"
export NOVACLIENT_INSECURE=true
export NEUTRONCLIENT_INSECURE=true
export CINDERCLIENT_INSECURE=true
export OS_PLACEMENT_API_VERSION=1.22
```

2. Запустите скрипт.

Важно: Скрипт необходимо запускать для каждого сеанса.

```
# source /etc/kolla/admin-openrc.sh
```

Теперь можно выполнять команды OpenStack с параметром `--insecure`. Например:

```
# openstack --insecure server list
+-----+-----+-----+-----+-----+-----+
| ID           | Name | Status | Networks          | Image | Flavor |
+-----+-----+-----+-----+-----+-----+
| 32b0f95d-477f-46b5-<...> | vm1  | ACTIVE | private=192.168.128.87 |      | tiny   |
+-----+-----+-----+-----+-----+-----+
```

Как защитить трафик OpenStack API, см. в разделе *Защита трафика API OpenStack с помощью SSL* (страница 259).

8.7 Установка доменного имени для API вычислений

С помощью типа трафика **API вычислений** Acronis Инфраструктура открывает внешнюю оконечную точку, которая прослушивает запросы API OpenStack. По умолчанию она указывает на IP-адрес сервера управления (или его виртуальный IP-адрес, если включена высокая доступность).

В некоторых случаях необходимо изменить все внешние оконечные точки так, чтобы использовалось доменное имя, преобразуемое в IP-адрес сервера управления (или его виртуальный IP), например, чтобы защитить трафик API OpenStack с помощью SSL-сертификата без поля `subjectAltName` или чтобы сервис Kubernetes обращался к API вычислений через доменное имя.

Чтобы настроить все внешние оконечные точки на использование доменного имени, выполните команду:

```
# openstack --insecure endpoint list | grep public
| 44aa0f53a40e4e52b1c7eeeb20c7811e | <...> | https://10.94.16.12:8774/v2.1/(tenant_id)s |
| 5a845b4b813047c292db73c42dad5efd | <...> | https://10.94.16.12:8780 |
| 0b906e518b1041c8b94af7f410403369 | <...> | https://10.94.16.12:9696 |
| d80af756adf1449f9237c3aeebc9206a | <...> | https://10.94.16.12:8004/v1/(tenant_id)s |
| d0e8c7da7d174e1f9aa4efbc6dff2113 | <...> | https://10.94.16.12:5000/v3 |
| 0e6d3a39d6c44aa883984a35dde434bb | <...> | https://10.94.16.12:9292 |
| 7d901686bca549f9b294e572f046f634 | <...> | https://10.94.16.12:8776/v2/(tenant_id)s |
| 1b68ac7c3f7949fbaeef4a815fe6f3b1 | <...> | https://10.94.16.12:8776/v3/(tenant_id)s |
```

```
# openstack --insecure endpoint set \
--url https://<DNS_name>:8774/v2.1/%(tenant_id)s 44aa0f53a40e4e52b1c7eeeb20c7811e
# openstack --insecure endpoint set \
--url https://<DNS_name>:8780 5a845b4b813047c292db73c42dad5efd
# openstack --insecure endpoint set \
--url https://<DNS_name>:9696 0b906e518b1041c8b94af7f410403369
# openstack --insecure endpoint set \
--url https://<DNS_name>:8004/v1/%(tenant_id)s d80af756adf1449f9237c3aeebc9206a
# openstack --insecure endpoint set \
--url https://<DNS_name>:5000/v3 d0e8c7da7d174e1f9aa4efbc6dff2113
# openstack --insecure endpoint set \
--url https://<DNS_name>:9292 0e6d3a39d6c44aa883984a35dde434bb
# openstack --insecure endpoint set \
--url https://<DNS_name>:8776/v2/%(tenant_id)s 7d901686bca549f9b294e572f046f634
# openstack --insecure endpoint set \
--url https://<DNS_name>:8776/v3/%(tenant_id)s 1b68ac7c3f7949fbaeef4a815fe6f3b1
```

8.8 Защита трафика API OpenStack с помощью SSL

Входящий и исходящий трафик внешней конечной точки, прослушивающей запросы API OpenStack, можно защитить с помощью сертификата SSL. Но поскольку доменные имена не используются по умолчанию, сертификат должен будет содержать поле `subjectAltName` с IP-адресом вышеупомянутого сервера управления. Если такого поля нет, потребуется изменить внешнюю конечную точку так, чтобы использовалось доменное имя, для которого у вас есть сертификат.

Чтобы защитить внешний трафик API OpenStack с помощью SSL, выполните следующие действия.

1. Загрузите сертификат, а затем закрытый ключ на панель администрирования, на экране **НАСТРОЙКИ > Сервер управления > ДОСТУП ПО SSL**.
2. Поместите файл сертификата ЦС в доверенную цепочку операционной системы.

```
# cp ca.pem /etc/pki/ca-trust/source/anchors/
# update-ca-trust extract
```

Как вариант, можно добавить параметр `--os-cacert ca.pem` к каждому вызову клиента OpenStack.

3. Если в сертификате нет поля `subjectAltName`, измените все внешние конечные точки так, чтобы использовалось доменное имя, для которого у вас есть сертификат, как описано в разделе [Установка доменного имени для API вычислений](#) (страница 258). Это доменное имя должно

разрешаться в IP-адрес сервера управления (или его виртуальный IP-адрес, если включена высокая доступность).

4. В скрипте OpenRC измените OS_AUTH_URL на то же доменное имя и удалите все параметры, связанные с незащищенным доступом. Например:

```
export OS_PROJECT_DOMAIN_NAME=Default
export OS_USER_DOMAIN_NAME=Default
export OS_PROJECT_NAME=admin
export OS_USERNAME=admin
export OS_PASSWORD=<ADMIN_PASSWORD>
export OS_AUTH_URL=https://<DOMAIN_NAME>:5000/v3
export OS_IDENTITY_API_VERSION=3
```

Теперь можно выполнять команды OpenStack без параметра `--insecure`.

8.9 Использование учета для вычислительных ресурсов

Можно собирать данные об использовании вычислительных ресурсов с помощью [Gnocchi](#). Эта база данных временных рядов обрабатывает и хранит данные измерений вычислительных ресурсов и предоставляет доступ к ним через REST API или программу командной строки.

Выборка измерений может производиться из таких вычислительных ресурсов, как виртуальные машины, диски и интерфейсы VM, вычислительные сети, тома и т. д. Все ресурсы постоянно пересматриваются, и, если изменяется какой-либо атрибут ресурса, это изменение записывается в историю ресурса. Например, для VM можно измерить объем выделенной памяти и число виртуальных ЦП, а также потребление памяти и ресурсов ЦП.

Сущность, в которой хранятся агрегированные показатели, состоящие из метки времени и значения, называется метрикой. Метрика прикрепляется к определенному ресурсу и связывается с политикой архивирования. Политика определяет, как долго агрегированные показатели хранятся в метрике и как они вычисляются (минимум, максимум, среднее и т. д.).

Следующие метрики доступны для агрегирования.

Таблица 8.9.1: Метрики вычислительных ресурсов

Метрика	Тип	Тип ресурса	Описание
memory	gauge	instance	Объем ОЗУ, выделенный для VM, в мегабайтах
memory.usage	gauge	instance	Процент использования ОЗУ виртуальной машиной
vcpus	gauge	instance	Число виртуальных ЦП, выделенных для VM
cpu	cumulative	instance	Процессорное время, использованное виртуальной машиной, в наносекундах
disk.device.read.requests	cumulative	instance_disk	Число запросов на чтение
disk.device.write.requests	cumulative	instance_disk	Число запросов на запись
disk.device.read.bytes	cumulative	instance_disk	Объем прочитанных данных в байтах
disk.device.write.bytes	cumulative	instance_disk	Объем записанных данных в байтах
network.incoming.bytes	cumulative	instance_network_interface	Входящий сетевой трафик в байтах
network.outgoing.bytes	cumulative	instance_network_interface	Исходящий сетевой трафик в байтах
network.incoming.packets	cumulative	instance_network_interface	Входящий сетевой трафик в пакетах
network.outgoing.packets	cumulative	instance_network_interface	Исходящий сетевой трафик в пакетах
image.size	gauge	image	Размер загруженного образа в байтах
volume.size	gauge	volume	Размер тома в гигабайтах
snapshot.size	gauge	volume	Размер моментального снимка тома в гигабайтах

Продолжается на следующей странице

Таблица 8.9.1 – продолжение с предыдущей страницы

Метрика	Тип	Тип ресурса	Описание
magnum.cluster	gauge	coe_cluster	Число кластеров системы оркестрации контейнеров (COE), т. е. Kubernetes

Накопительные метрики опрашиваются каждые пять минут и увеличиваются с течением времени, а измерительные метрики обновляются при каждом событии и показывают переменные значения.

Для измерительных метрик политикой архивирования по умолчанию является `ceilometer-low`, а для накопительных — `ceilometer-low-rate`. Эти политики подразумевают, что все полученные агрегированные результаты хранятся один день с точностью до 5 минут и один месяц с точностью до 1 часа. Разница между ними состоит в используемых методах агрегирования: политика `ceilometer-low` сохраняет только средние значения, а политика `ceilometer-low-rate` сохраняет средние значения, а также среднее изменение значений на интервал.

8.9.1 Включение сервиса учета

Примечание: Если вы уже установили сервисы учета через панель администрирования, пропустите этот раздел.

Чтобы включить сервисы учета в вычислительном кластере, выполните одно из следующих действий.

- Если у вас еще нет вычислительного кластера, разверните его и включите учет, добавив параметр `--enable-metering` в команду `vinfra service compute cluster create`. Например:

```
# vinfra service compute create --nodes <node1_id>[,<node2_id>,...] --enable-metering
```

- Если вычислительный кластер уже был создан, используйте следующую команду:

```
# vinfra service compute cluster set --enable-metering
```

Примечание: Сервис будет учитывать только вычислительные объекты, созданные после его включения.

Эти команды открывают порт 8041 и включают два сервиса Gnocchi: HTTP-сервер `gnocchi-api` и демон метрик `gnocchi-metricd`.

8.9.2 Использование программы командной строки Gnocchi

После включения учета доступ к метрикам вычислительных ресурсов можно получить либо через REST API, либо с помощью программы командной строки Gnocchi. Чтобы использовать программу, выполните следующие действия.

1. Установите клиент Gnocchi.

```
# yum install python-gnocchiclient
```

2. Создайте файл администратора OpenRC.

```
# kolla-ansible post-deploy
# source /etc/kolla/admin-openrc.sh
```

Теперь вы можете использовать команду `gnocchi` с параметром `--insecure`. См. также раздел *Защита трафика API OpenStack с помощью SSL* (страница 259).

Например, можно выполнить следующие действия.

1. Просмотреть существующие ресурсы.

```
# gnocchi --insecure resource list -c id -c type -c project_id
+-----+-----+-----+
| id          | type                | project_id |
+-----+-----+-----+
| 238597c7<...> | volume              | c1bf1<...> |
| 3c78558f<...> | instance            | c1bf1<...> |
| 44f1896f<...> | instance_network_interface | c1bf1<...> |
| 880e9efc<...> | instance_disk       | c1bf1<...> |
+-----+-----+-----+
```

Выходные данные показывают, что в вычислительном кластере размещена одна виртуальная машина с одним сетевым адаптером и одним диском, который также присутствует как том.

2. Проверить все доступные метрики для ресурсов.

```
# gnocchi --insecure metric list
+-----+-----+-----+-----+-----+
| id          | archive_policy/name | name                | unit   | resou
+-----+-----+-----+-----+-----+
| 243c7ac<...> | ceilometer-low     | disk.root.size     | GB     | 3c785
| 365e454<...> | ceilometer-low-rate | network.outgoing.packets | packet | 44f18
| 4fbd3e8<...> | ceilometer-low-rate | disk.device.read.requests | request | 880e9
```

54519fa<...>	ceilometer-low	compute.instance.booting.time	sec	3c785
5e1406f<...>	ceilometer-low-rate	disk.device.write.bytes	B	880e9
66a96c2<...>	ceilometer-low	vcpus	vcpu	3c785
722ea97<...>	ceilometer-low	memory	MB	3c785
7c961ab<...>	ceilometer-low-rate	disk.device.write.requests	request	880e9
87e9fb7<...>	ceilometer-low-rate	network.incoming.packets	packet	44f18
9d56321<...>	ceilometer-low-rate	disk.device.read.bytes	B	880e9
b8be8f7<...>	ceilometer-low-rate	cpu	ns	3c785
c1961bb<...>	ceilometer-low	disk.ephemeral.size	GB	3c785
c9b61e0<...>	ceilometer-low	volume.size	GB	23859
d06a58c<...>	ceilometer-low-rate	network.outgoing.bytes	B	44f18
e2d9981<...>	ceilometer-low-rate	network.incoming.bytes	B	44f18
eaac2b5<...>	ceilometer-low	memory.usage	MB	3c785

3. Просмотреть измерения для определенной метрики. Например:

```
# gnocchi --insecure measures show cpu --resource-id 3c78558f-08bf-47e2-ba3e-bdb13e7b2
```

timestamp	granularity	value
2019-12-11T17:05:00+03:00	300.0	2.2756e+11
2019-12-11T17:10:00+03:00	300.0	2.8897e+11
2019-12-11T17:15:00+03:00	300.0	3.7367e+11
2019-12-11T17:20:00+03:00	300.0	4.64e+11
2019-12-11T17:25:00+03:00	300.0	7.6104e+11

По умолчанию используется метод агрегирования mean. Чтобы получить данные о потреблении процессорного времени на каждый интервал, используйте параметр `--aggregation rate:mean`.

```
# gnocchi --insecure measures show cpu --aggregation rate:mean \
--resource-id 3c78558f-08bf-47e2-ba3e-bdb13e7b25bb
```

timestamp	granularity	value
2019-12-11T17:10:00+03:00	300.0	6141000000.0
2019-12-11T17:15:00+03:00	300.0	8470000000.0
2019-12-11T17:20:00+03:00	300.0	9033000000.0
2019-12-11T17:25:00+03:00	300.0	2.9704e+11
2019-12-11T17:30:00+03:00	300.0	3.64e+11

Полный список команд `gnocchi` можно посмотреть в документации [OpenStack](#).

8.9.3 Просмотр использования ресурсов на уровне проекта

Чтобы получить данные об использовании вычислительных ресурсов, выделенных всем виртуальным машинам, принадлежащим к определенному проекту, используйте команду `vinfra service compute quotas show --usage <project_id>`. Например:

```
# vinfra service compute quotas show 6ef6f48f01b640ccb8ff53117b830fa3 --usage
+-----+-----+
| Field                | Value |
+-----+-----+
| compute.cores.limit  | 20    |
| compute.cores.used   | 2     |
| compute.ram.limit    | 40960 |
| compute.ram.used     | 4096  |
| k8saas.cluster.limit | 10    |
| k8saas.cluster.used  | 0     |
| lbaas.loadbalancer.limit | 10    |
| lbaas.loadbalancer.used  | 0     |
| network.floatingip.limit | 10    |
| network.floatingip.used  | 0     |
| storage.gigabytes.default.limit | 1024  |
| storage.gigabytes.default.used  | 66    |
+-----+-----+
```

Выходные данные показывают, что для VM, включенных в проект с идентификатором `62af79f31ae5488aa33077d02af48282`, было выделено 2 виртуальных ЦП, 4 ГБ ОЗУ и 66 ГБ дискового пространства.

8.10 Настройка политики памяти для сервисов хранилища

Ограничения и гарантированные объемы памяти для сервисов хранилища можно настроить во время выполнения с помощью команд `vinfra memory-policy vstorage-services`. Это можно сделать для всего кластера или отдельного сервера.

Следующие параметры памяти можно настроить вручную:

- гарантированный объем памяти,
- размер файла подкачки,
- кэш страниц (который в свою очередь задается с помощью минимального и максимального множителя).

Кэш страниц рассчитывается по следующей формуле:

```
$PAGE_CACHE = minimum <= ratio * $TOTAL_MEMORY <= maximum
```

Значения `minimum` и `maximum` — это жесткие ограничения, которые применяются, если значение `ratio * $TOTAL_MEMORY` выходит за эти пределы.

Чтобы лучше понять, как вычисляется размер кэша страниц, рассмотрим следующие примеры.

Таблица 8.10.1: Примеры кэша страниц

	Пример 1 (размер кэша в пределах ограничений)	Пример 2 (размер кэша равен минимальному)	Пример 3 (размер кэша равен максимальному)
Всего памяти	4 ГиБ	4 ГиБ	4 ГиБ
Множитель кэша	0,5	0,1	0,9
Минимальный кэш	1 ГиБ	2 ГиБ	1 ГиБ
Максимальный кэш	3 ГиБ	3 ГиБ	3 ГиБ
Размер кэша	2 ГиБ	2 ГиБ	3 ГиБ

Если параметры памяти заданы как для сервера, так и для кластера, применяются параметры для сервера. Если параметры памяти, настраиваемые вручную, отсутствуют, то управление памятью автоматически выполняется демоном `vcmmd` следующим образом.

- Каждый CS (например, диск хранилища) требует 512 МиБ ОЗУ для кэша страниц.
- Минимальный кэш страниц составляет 1 ГиБ.
- Если общий объем памяти меньше 48 ГиБ, то максимальный кэш страниц рассчитывается как две трети этого объема.
- Если общий объем памяти больше 48 ГиБ, то максимальный кэш страниц составляет 32 ГиБ.

Чтобы проверить текущие параметры памяти для сервисов хранилища, заданные демоном `vcmmd`, выполните команду

```
# vcmmdctl list
name                               type active guarantee    limit  swap  cache
```

```
<...>
vstorage.slice/vstorage-services.slice SRVC yes 1310720 24522132 0 1048576
```

8.10.1 vinfra memory-policy vstorage-services per-cluster change

Изменение параметров памяти для кластера:

```
usage: vinfra memory-policy vstorage-services per-cluster change
      [--guarantee <guarantee>] [--swap <swap>] [--cache-ratio <cache-ratio>]
      --cache-minimum <cache-minimum> --cache-maximum <cache-maximum>]
```

--guarantee <guarantee>

Гарантированный объем в байтах

--swap <swap>

Размер файла подкачки в байтах или -1, если размер неограничен

--cache-ratio <cache-ratio>

Множитель кэша от 0 до 1 включительно

--cache-minimum <cache-minimum>

Минимальный размер кэша в байтах

--cache-maximum <cache-maximum>

Максимальный размер кэша в байтах

Пример:

```
# vinfra memory-policy vstorage-services per-cluster change --guarantee 8796093022208 \
--swap 1099511627776 --cache-ratio 0.5 --cache-minimum 1099511627776 \
--cache-maximum 3298534883328
+-----+-----+
| Field   | Value                                     |
+-----+-----+
| cache   | maximum: 3298534883328 |
|         | minimum: 1099511627776 |
|         | ratio: 0.5              |
| guarantee | 8796093022208         |
| swap     | 1099511627776         |
+-----+-----+
```

Эта команда задает параметры памяти сервисов хранилища для всех серверов в кластере следующим образом:

- гарантируемый объем памяти — 8 ГБ

- размер файла подкачки — 1 ГБ
- ограничения кэша страниц: минимум — 1 ГБ, максимум — 3 ГБ, множитель кэша — 0,5

8.10.2 `vinfra memory-policy vstorage-services per-cluster show`

Отображение параметров памяти для кластера:

```
usage: vinfra memory-policy vstorage-services per-cluster show
```

Пример:

```
# vinfra memory-policy vstorage-services per-cluster show
+-----+-----+
| Field      | Value                               |
+-----+-----+
| cache      | maximum: 3298534883328             |
|             | minimum: 1099511627776             |
|             | ratio: 0.5                         |
| guarantee  | 8796093022208                     |
| swap       | 1099511627776                     |
+-----+-----+
```

Эта команда выводит параметры памяти сервисов хранилища для всех серверов в кластере.

8.10.3 `vinfra memory-policy vstorage-services per-cluster reset`

Сброс параметров памяти кластера до значений по умолчанию:

```
usage: vinfra memory-policy vstorage-services per-cluster reset [--guarantee]
      [--swap] [--cache]
```

`--guarantee`

Сброс только гарантированного объема.

`--swap`

Сброс только размера файла подкачки.

`--cache`

Сброс только значений кэша.

Пример:

```
# vinfra memory-policy vstorage-services per-cluster reset --cache
+-----+-----+
```

```

| Field      | Value          |
+-----+-----+
| cache     |                |
| guarantee | 8796093022208 |
| swap      | 1099511627776 |
+-----+-----+

```

Эта команда сбрасывает настроенные вручную ограничения кэша страниц до значений по умолчанию для всех серверов в кластере хранилища.

8.10.4 vinfra memory-policy vstorage-services per-node change

Изменение параметров памяти для сервера:

```

usage: vinfra memory-policy vstorage-services per-node change [--guarantee <guarantee>]
      [--swap <swap>] [--cache-ratio <cache-ratio> --cache-minimum <cache-minimum>
      --cache-maximum <cache-maximum>] --node <node>

```

`--guarantee <guarantee>`

Гарантированный объем в байтах

`--swap <swap>`

Размер файла подкачки в байтах или -1, если размер неограничен

`--cache-ratio <cache-ratio>`

Множитель кэша от 0 до 1 включительно

`--cache-minimum <cache-minimum>`

Минимальный размер кэша в байтах

`--cache-maximum <cache-maximum>`

Максимальный размер кэша в байтах

`--node <node>`

Идентификатор сервера или имя хоста

Пример:

```

# vinfra memory-policy vstorage-services per-node change --guarantee 8796093022208 \
--swap 1099511627776 --cache-ratio 0.5 --cache-minimum 1099511627776 \
--cache-maximum 3298534883328 --node 7ffa9540-5a20-41d1-b203-e3f349d62565
+-----+-----+
| Field      | Value          |
+-----+-----+
| cache     | maximum: 3298534883328 |

```

```
|           | minimum: 109951162776 |
|           | ratio: 0.5             |
| guarantee | 8796093022208         |
| swap      | 109951162776          |
+-----+-----+
```

Эта команда задает параметры памяти сервисов хранилища для сервера с идентификатором 7ffa9540-5a20-41d1-b203-e3f349d62565 следующим образом:

- гарантируемый объем памяти — 8 ГБ
- размер файла подкачки — 1 ГБ
- ограничения кэша страниц: минимум — 1 ГБ, максимум — 3 ГБ, множитель кэша — 0,5

8.10.5 `vinfra memory-policy vstorage-services per-node show`

Отображение параметров памяти для сервера:

```
usage: vinfra memory-policy vstorage-services per-node show --node <node>
```

`--node <node>`

Идентификатор сервера или имя хоста

Пример:

```
# vinfra memory-policy vstorage-services per-node show \
--node 7ffa9540-5a20-41d1-b203-e3f349d62565
+-----+-----+
| Field      | Value                               |
+-----+-----+
| cache      | maximum: 13194139533312            |
|             | minimum: 8796093022208             |
|             | ratio: 0.7                          |
| guarantee  | 8796093022208                       |
| swap       | 109951162776                        |
+-----+-----+
```

Эта команда выводит параметры памяти сервисов хранилища, заданные для сервера с идентификатором 7ffa9540-5a20-41d1-b203-e3f349d62565.

8.10.6 vinfra memory-policy vstorage-services per-node reset

Сброс параметров памяти сервера до значений по умолчанию:

```
usage: vinfra memory-policy vstorage-services per-node reset [--guarantee] [--swap]
      [--cache] --node <node>
```

--guarantee

Сброс только гарантированного объема.

--swap

Сброс только размера файла подкачки.

--cache

Сброс только значений кэша.

--node <node>

Идентификатор сервера или имя хоста

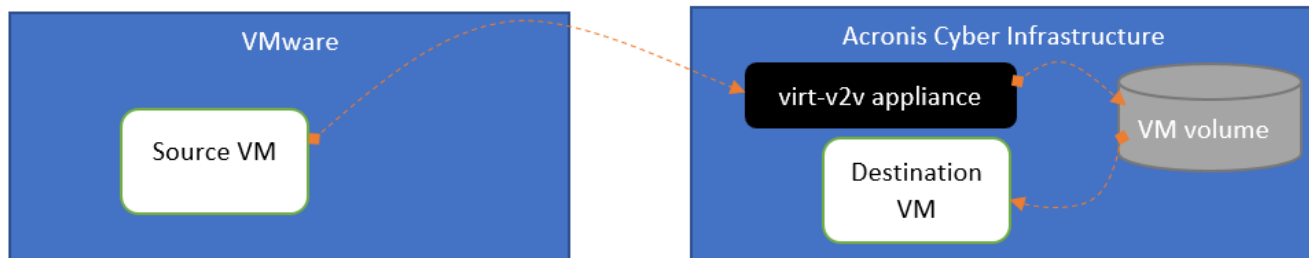
Пример:

```
# vinfra memory-policy vstorage-services per-node reset --cache \
--node 7ffa9540-5a20-41d1-b203-e3f349d62565
+-----+-----+
| Field   | Value           |
+-----+-----+
| cache   |                  |
| guarantee | 8796093022208 |
| swap    | 1099511627776  |
+-----+-----+
```

Эта команда сбрасывает настроенные вручную ограничения кэша страниц до значений по умолчанию для сервера с идентификатором 7ffa9540-5a20-41d1-b203-e3f349d62565.

8.11 Миграция виртуальных машин из VMware vCenter

Начиная с версии Acronis Инфраструктура 3.5, можно переносить виртуальные машины из VMware vCenter 5.0 или более новой версии в продукт Acronis Инфраструктура с помощью инструмента virt-v2v. Необходимо будет создать виртуальную машину устройства virt-v2v для переноса и преобразования данных.



8.11.1 Развертывание VM устройства

Чтобы создать VM устройства `virt-v2v`, выполните следующие действия.

1. Скачайте образ устройства `virt-v2v` из официального репозитория.
2. Загрузите образ в продукт Acronis Инфраструктура. Например:

```
# vinfra service compute image create virt-v2v-img \
--file vmware_to_acronis.qcow2
```

3. Создайте SSH-ключ для устройства, если у вас его нет. Например:

```
# vinfra service compute key create publickey \
--public-key virt-v2v-app-key.pub
```

4. Создайте виртуальную машину и разверните на ней загруженный образ. VM требуется как минимум два ЦП, 4 Гиб ОЗУ и достаточно дискового пространства для размещения самой большой VM, которая будет перенесена в продукт Acronis Инфраструктура. VM также должна быть подключена к сети, которая работает с типом трафика **API вычислений**, и сети с доступом к API VMware vCenter. Например:

```
# vinfra service compute server create virt-v2v-appliance \
--flavor medium --key-name <key>
--network id=<compute_API> --network id=<vcenter_API> \
--volume source=image,id=virt-v2v-img,size=<size>
```

где:

- `<key>` — SSH-ключ для авторизации в VM устройства.
- `<compute_API>` — сеть, которая работает с типом трафика **API вычислений**.
- `<vcenter_API>` — сеть с доступом к API VMware vCenter.

- `<size>` — размер диска. Для онлайн-миграции размер должен быть достаточным для размещения самой большой VM из тех, которые вы планируете перенести. Для офлайн-миграции размер должен быть еще в два раза больше.

8.11.2 Настройка проверки подлинности в VM устройства

1. Выполните вход в VM устройства как пользователь `admin` с SSH-ключом.
2. Получите привилегии пользователя `root`, например, с помощью команды `sudo -i`.
3. Создайте `bash`-скрипт, который экспортирует учетные данные OpenStack.

```
# cat > user-openrc.sh << EOF
export OS_PROJECT_DOMAIN_NAME=Domain_name
export OS_USER_DOMAIN_NAME=Domain_name
export OS_PROJECT_NAME=Project_name
export OS_USERNAME=user_name
export OS_PASSWORD=Password
export OS_AUTH_URL=https://<admin_panel_IP_addr>:5000/v3
export OS_IDENTITY_API_VERSION=3
export OS_INSECURE=true
export NOVACLIENT_INSECURE=true
export NEUTRONCLIENT_INSECURE=true
export CINDERCLIENT_INSECURE=true
export LIBGUESTFS_BACKEND=direct
EOF
```

Примечание: Вам потребуются учетные данные администратора для проекта, которому принадлежит VM устройства.

4. Скопируйте сертификат корневого ЦС OpenStack и ключи ЦС с сервера управления Acronis Инфраструктура.

```
# scp root@<MN_IP>:/usr/libexec/vstorage-ui-backend/ca/ca.* /etc/pki/ca-trust/source/a
# update-ca-trust extract
```

где `<MN_IP>` — IP-адрес сервера управления. Дополнительные сведения см. в разделе *Защита трафика API OpenStack с помощью SSL* (страница 259).

5. Создайте файл с паролем VMware vCenter для предоставления устройству `virt-v2v`. Например:

```
# echo $vCenterPass > password.txt
```

Как вариант, можно ввести пароль во время миграции или предоставить его устройству `virt-v2v` с помощью параметра `--password-file`.

8.11.3 Онлайн-миграция виртуальных машин в продукт Acronis Инфраструктура

Прежде чем продолжить, обратите внимание на следующее.

- Можно выполнять миграцию ВМ, созданных на vCenter 5.0 или более новой версии.
 - Удалите VMware Tools из виртуальных машин Windows перед миграцией, чтобы избежать проблем при загрузке позже. Вы не сможете сделать это после миграции. Утилиты VMware Tools будут удалены из гостевых ОС Linux автоматически.
1. Выполните вход в ВМ устройства как пользователь `admin` с SSH-ключом.
 2. Получите привилегии пользователя `root`, например, с помощью команды `sudo -i`.
 3. Задайте учетные данные OpenStack.

```
# source user-openrc.sh
```

4. Проверьте соединение между libvirt и VMware vCenter. Например:

```
# virsh -c 'vpx://<domain>%5c<user>@<hostname>?no_verify=1' list --all
Enter root's password for vcenter.example.com: ***
+-----+-----+-----+
| Id | Name          | <...> |
+-----+-----+-----+
| -  | Fedora 20     | <...> |
| -  | Windows 2008 | <...> |
+-----+-----+-----+
```

где `<hostname>` — имя хоста VMware ESXi, на котором работают виртуальные машины. Полный путь выглядит как `<vCenter_hostname>/<datacenter_name>/<cluster_name>/<server_hostname>`, его можно посмотреть в VMware vCenter.

Если имя пользователя VPX содержит обратную косую черту (например, `<domain>\<user>`), замените ее на `%5c`: `<domain>%5c<user>`. Аналогично замените пробелы на `%20`.

5. Проверьте подключение OpenStack и определите идентификатор устройства `virt-v2v`. Например:

```
# openstack --insecure server list
+-----+-----+-----+-----+
| ID | Name | Status | <...> |
+-----+-----+-----+-----+
| 635ae4cc-4c01-461a-ae63-91ca4187a7b1 | virt-v2v-appliance | ACTIVE | <...> |
+-----+-----+-----+-----+
```

6. Выключите VM. Виртуальные машины Windows следует выключать корректно, чтобы миграция прошла успешно.

7. Перенесите VM на том в продукте Acronis Инфраструктура. Например:

```
# virt-v2v -ip password.txt -ic 'vpx://<domain>%5c<user>@<hostname>?no_verify=1' \
'Windows 2008' -o openstack -oo server-id=635ae4cc-4c01-461a-ae63-91ca4187a7b1
```

8. Определите идентификатор или имя нового тома. Например:

```
# openstack --insecure volume list
+-----+-----+-----+-----+-----+
| ID | Name | Status | Size | Attached to |
+-----+-----+-----+-----+-----+
| 024b6843-2de3-4e25-a6e1-2b6ea2d601cf | sda1 | available | 64 | |
+-----+-----+-----+-----+-----+
```

9. Создайте виртуальную машину в продукте Acronis Инфраструктура на базе нового тома. Например:

```
# vinfra service compute server create migratedvm \
--network id=private --network id=public \
--volume source=volume,id=sda1,size=64 --flavor medium
```

8.11.4 Офлайн-миграция виртуальных машин в продукт Acronis Инфраструктура

Если сетевое подключение между VM устройства virt-v2v и VMware vCenter недостаточно быстрое, можно вручную скопировать виртуальные машины на USB-накопитель, подключить его к VM устройства virt-v2v и преобразовать их в VM продукта Acronis Инфраструктура. Для этого выполните следующие действия.

1. Скопируйте все файлы VM, включая vmdk и vmtx, на USB-накопитель.
2. Подключите USB-накопитель к хосту в той же локальной сети, где находится VM устройства.
3. Выполните вход в VM устройства как пользователь admin с SSH-ключом.

- Получите привилегии пользователя root, например, с помощью команды `sudo -i`.
- Скопируйте файлы виртуальных машин на VM устройства, например, с помощью `rsync` или `scp`.
- Задайте учетные данные OpenStack.

```
# source user-openrc.sh
```

- Перенесите VM на том в продукте Acronis Инфраструктура. Например:

```
# virt-v2v -i vmx <VM_config> -o openstack \
-oo server-id=635ae4cc-4c01-461a-ae63-91ca4187a7b1
```

где `<VM_config>` — файл конфигурации VM в формате `vmx`.

- Определите идентификатор или имя нового тома. Например:

```
# openstack --insecure volume list
+-----+-----+-----+-----+-----+
| ID                | Name | Status  | Size | Attached to |
+-----+-----+-----+-----+-----+
| 024b6843-2de3-4e25-a6e1-2b6ea2d601cf | sda1 | available | 64 |              |
+-----+-----+-----+-----+-----+
```

- Создайте виртуальную машину в продукте Acronis Инфраструктура на базе нового тома.

Например:

```
# vinfra service compute server create migratedvm \
--network id=private --network id=public \
--volume source=volume,id=sda1,size=64 --flavor medium
```

8.12 Изменение типа VM по умолчанию для балансировщика нагрузки

По умолчанию балансировщик нагрузки создается с частным типом VM `amphora`, которым нельзя управлять через `vinfra`. Однако тип VM можно изменить с помощью инструмента командной строки OpenStack следующим образом.

- Создайте файл администратора OpenRC.

```
# kolla-ansible post-deploy
# source /etc/kolla/admin-openrc.sh
```

- Проверьте, что стандартный тип VM `amphora` существует.

```
# openstack --insecure flavor list --all
```

ID	Name	RAM	Disk	Ephemeral	VCPUs	Is Public
100	tiny	512	0	0	1	True
101	small	2048	0	0	1	True
102	medium	4096	0	0	2	True
103	large	8192	0	0	4	True
104	xlarge	16384	0	0	8	True
amphora	amphora	1024	10	0	2	False

3. Удалите этот тип VM.

```
# openstack --insecure flavor delete amphora
```

4. Создайте новый тип VM amphora с нужными параметрами. Например:

```
# openstack --insecure flavor create amphora --id amphora --ram 4096 \
--vcpus 4 --disk 15 --private
```

Field	Value
OS-FLV-DISABLED:disabled	False
OS-FLV-EXT-DATA:ephemeral	0
disk	15
id	amphora
name	amphora
os-flavor-access:is_public	False
properties	
ram	4096
rxtx_factor	1.0
swap	
vcpus	4

5. Измените тип VM балансировщика нагрузки, выполнив переключение на резерв. Например:

```
# openstack --insecure loadbalancer failover mylbaas
```

Балансировщик нагрузки mylbaas будет создан заново с 4 виртуальными ЦП, 4 ГБ ОЗУ и 15 ГБ дискового пространства.

8.13 Configuring Retention Policy for Prometheus Metrics

The Prometheus service used for monitoring the cluster runs and stores its data on the management node. By default, Prometheus metrics are stored for 7 days. This retention period can be insufficient for troubleshooting purposes. You can increase this period in the Prometheus configuration file as follows:

1. On the management node, open the `/etc/sysconfig/prometheus` file to edit, set the needed retention period for the `STORAGE_RETENTION` option, and save your changes. For example:

```
STORAGE_RETENTION="--storage.tsdb.retention.time=30d"
```

2. Restart the Prometheus service:

```
systemctl restart prometheus.service
```

In case high availability is enabled in the storage cluster, repeat these steps for the other two management nodes.

However, with a long retention period, the root partition where the data is stored may run out of free space. To prevent this, you can define the maximum size for the Prometheus metrics. The oldest data will be removed first. To change the time retention policy to the size retention policy, do as follows:

1. On the management node, open the `/etc/sysconfig/prometheus` file to edit, change the flag for the `STORAGE_RETENTION` option, and save your changes. For example:

```
STORAGE_RETENTION="--storage.tsdb.retention.size=10GB"
```

2. Restart the Prometheus service:

```
systemctl restart prometheus.service
```

In case high availability is enabled in the storage cluster, repeat these steps for the other two management nodes.

8.14 Viewing Cluster Logs

When you encounter a problem in Acronis Инфраструктура, you can send a problem report as described in [Sending Problem Reports](#). The report will gather all the logs needed to troubleshoot the problem and send them to the technical support team.

Alternatively, you can investigate the root cause of your problem using the logs listed in table below.

Таблица 8.14.1: Cluster log location

Service	Log location	Описание
Metadata	/vstorage/mds/logs/mds.log.zst on the storage node hosting an MDS service	Storage metadata service events
Storage	/vstorage/<id>/cs/logs/cs.log.zst on the storage node hosting a CS service <hr/> Примечание: To find out the log location of a particular CS on the node, run <code>vstorage -c <cluster_name> list-services -C</code> . <hr/>	Chunk service events
Storage mountpoint	/var/log/vstorage/<cluster_name>/vstorage-mount.*.blog on any storage node	Software-defined storage mounting on each node
Management node	/var/log/vstorage-ui-backend/messages.log and /var/log/vstorage-ui-backend/celery*.log on the management node	Management node and admin panel events
	/var/log/vstorage-ui-agent/* on any storage node	Agent controller component events
Backup Gateway	/var/log/vstorage/abgw.log*zst on any node in the Backup Gateway cluster <hr/> Примечание: The latest log is <code>abgw.log.zst</code> , older ones are renamed to <code>abgw.log.0.zst</code> , <code>abgw.log.1.zst</code> , etc. <hr/>	Backup Gateway cluster deployment and management
iSCSI	/var/log/vstorage/iscsi/vstorage-target.log on any node in an iSCSI target group	iSCSI target management
	/var/log/vstorage/iscsi/vstorage-target-monitor.log on any node in an iSCSI target group	iSCSI target monitoring

Продолжается на следующей странице

Таблица 8.14.1 – продолжение с предыдущей страницы

Service	Log location	Описание
	/var/log/vstorage/iscsi/scst.log.zst on any node in an iSCSI target group	SCST service logs
S3	/var/log/ostor/NS-* on the S3 node with NS services	S3 name server events
	/var/log/ostor/OS-* on the S3 node with OS services	S3 object server events
	/var/log/ostor/S3GW-* on the S3 node with GW services	S3 gateway events
	/var/log/nginx/* on any node in the S3 cluster	nginx service logs
	/var/log/ostor/GR-* on the S3 node with GR services	S3 geo-replicator service events
NFS	/var/log/ganesha/ganesha.log and /var/log/ostor/ostorfs.log.gz on any node in the NFS cluster	NFS server events
	/var/log/vstorage/vstorage-nfsd.log on any node in the NFS cluster	NFS service events
	/var/log/ostor/FS-* on the node hosting an NFS share	FS service events
	/var/log/ostor/OS-* on the node hosting an NFS share	OS service events
Compute	/var/log/vstorage-ui-backend/ansible.log on the controller node	Compute cluster and add-on deployment
	/var/log/hci/beholder/beholder.log on the controller node	Notifications about all compute events, including VM placement
	/var/log/hci/nova/* on the compute node hosting a VM Примечание: In case of problems during VM migration, view /var/log/hci/nova/nova-compute.log on the source and destination compute nodes.	Virtual machine management

Продолжается на следующей странице

Таблица 8.14.1 – продолжение с предыдущей страницы

Service	Log location	Описание
	/var/log/hci/neutron/neutron-l3-agent.log on any compute node	Virtual routing events
	/var/log/hci/neutron/neutron-openvswitch-agent.log on the compute node hosting a VM	VM network interface management
	/var/log/hci/cinder/* on the controller node	Compute volume management
	/var/log/hci/glance/glance-api.log on the controller node	Image service API requests
	/var/log/hci/octavia/octavia-worker.log and /var/log/hci/octavia/octavia-api.log on the controller node	Load balancing service management
	/var/log/hci/magnum/magnum-conductor.log, /var/log/hci/magnum/magnum-api.log, and /var/log/hci/heat/heat-engine.log on the controller node	Kubernetes service and VM stack deployment and management
	/var/log/hci/gnocchi/* and /var/log/hci/ceilometer/* on any compute node	Billing metering service management
High availability	/var/log/vstorage-ui-backend/ha.log on all management nodes	High availability management
Updates	/var/log/vstorage-ui-backend/software-updates.log on the management node	Software update orchestration
	/var/log/vstorage-ui-agent/software-updates.log on any storage node	Software update downloading and installation on each node

To open log files, use the following commands:

- for log files:

```
# less <log_file>.log
```

- for blog files:

```
# blogcat <log_file>.blog | less
```

- for gz files:

```
# zless <log_file>.gz
```

- for zst files:

```
# zstdless <log_file>.zst
```