

Система управления подписками на
ресурсы виртуальной
инфраструктуры Digital Energy
(«СУП-РВИ-DE»)

Инструкция по установке

3.1

Версия документа IG-01.3.1.1 • 21 Января 2019

Digital Energy Cloud Solutions

Россия, г. Москва, ул. Рочдельская, 15, корп. 15

+7 (499) 649 09 90

www.digitalenergy.online

Отказ от ответственности

Информация, представленная в данном документе, предоставляется "как есть", в существующем состоянии и без каких-либо гарантий. Digital Energy Cloud Solutions отказывается от всех гарантий, явных или скрытых, включая гарантии коммерческой ценности и пригодности для определенной цели. Ни при каких обстоятельствах Digital Energy Cloud Solutions не несет ответственности за любой ущерб, включая прямой, косвенный, случайный, последующий, за потерю коммерческой прибыли или понесенные специальные убытки, даже в том случае, если Digital Energy Cloud Solutions или ее поставщики были предупреждены о возможности таких убытков.

Период действия документа

Digital Energy Cloud Solutions может периодически обновлять документацию в промежутках между выпусками соответствующих программных обеспечений. Следовательно, если данный документ не был загружен в течение недавнего времени, в нем может отсутствовать последняя обновленная информация. Пожалуйста, обратитесь в отдел технической поддержки сайта <https://support.digitalenergy.online>, для получения самой последней информации.

Где можно получить помощь

Техническую поддержку, программный продукт и информацию о лицензировании Digital Energy Cloud Solutions можно получить одним из следующих способов.

Информация о программном продукте — сведения о продуктах размещены на веб-сайте Digital Energy Cloud Solutions по следующему адресу: <https://digitalenergy.online>

Техническая поддержка — Перейдите по ссылке <http://support.digitalenergy.online> и выберите вкладку "Открыть новую заявку". На странице технической поддержки, вы увидите несколько вариантов, в том числе варианты для создания запроса на обслуживание. Обратите внимание, что для открытия запроса на обслуживание необходимо иметь действующее соглашение о поддержке.

Ваши замечания

Ваши предложения и замечания помогут нам и дальше продолжать повышать общее качество публикаций, поддержки и, самое главное, качество конечного продукта. Пожалуйста, присылайте ваши замечания по данному документу на адрес электронной почты infobox@digitalenergy.online

Если у вас есть предложения, замечания или вопросы по поводу конкретной информации, пожалуйста, укажите название, редакцию, номера страниц (если таковые имеются) и любые другие детали, которые помогут нам найти тему, на которую вы ссылаетесь.

Содержание

1	Введение	4
2	Установка платформы	5
2.1	Установка ОС на управляющих узлах (контроллеры).	5
2.2	Развертывание управляющего кластера на основе kubernetes	10
2.3	Установка ОС на узлы платформы	14
2.4	Установка СХД OpenvStorage	15
2.5	Установка СУП-РВИ-DE на узлы платформы	18
3	Авторизация в платформе	19

1 Введение

Документ «Инструкция по установке» является частью документации к продукту «Система управления подписками на ресурсы виртуальной инфраструктуры Digital Energy» (СУП-РВИ-DE)», в котором описываются пошаговые действия процедуры установки платформы.

2 Установка платформы

2.1 Установка ОС на управляющих узлах (контроллеры).

Стек контроллера состоит из трех узлов, запущенных под управлением ОС Ubuntu 16.04

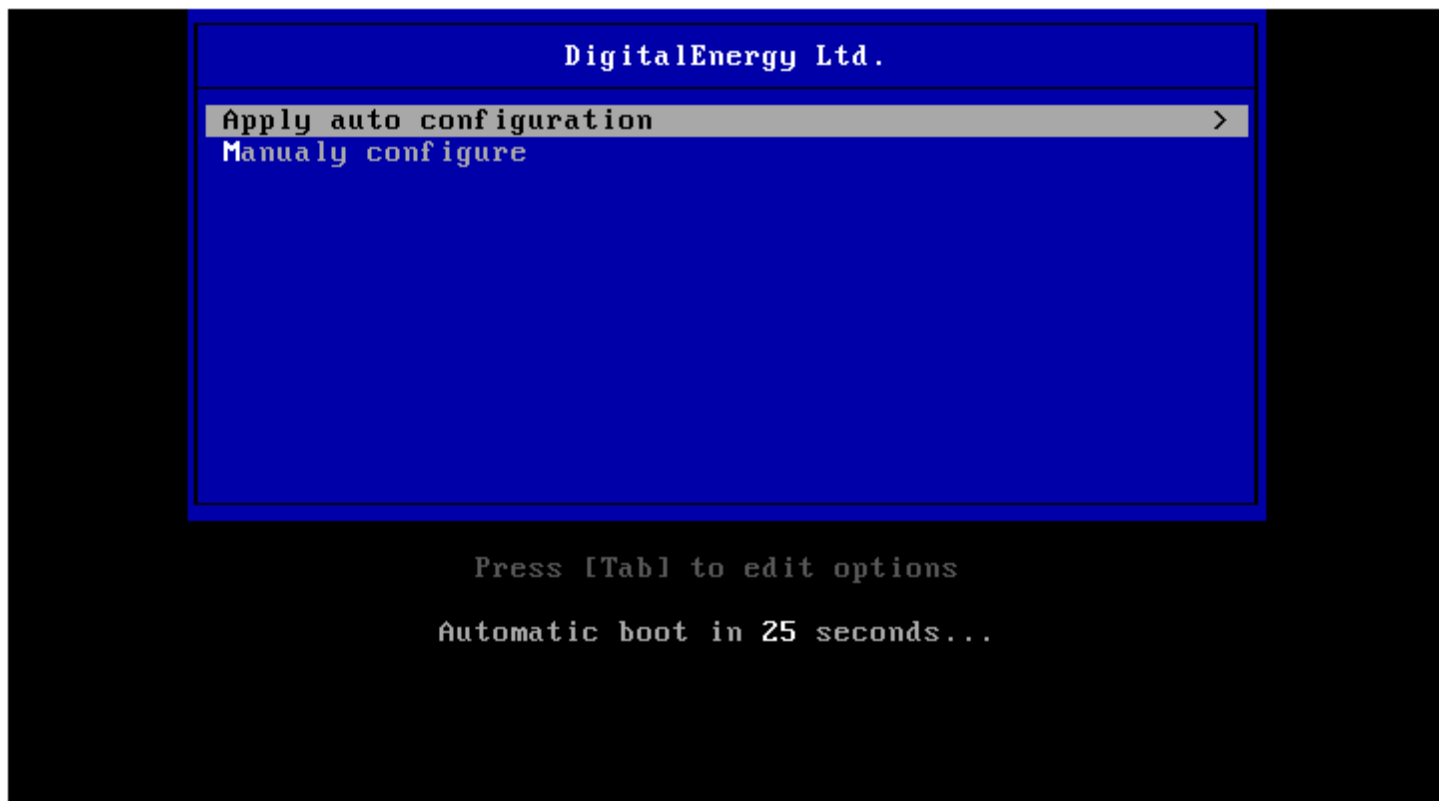
Каждый из узлов имеет специфические настройки. Среди прочих, особое внимание следует уделить настройкам сети и устройств хранения. Кроме того, каждый из узлов хранения должен содержать по меньшей мере 32 ГБ оперативной памяти.

1. Для установки следует воспользоваться образом -

https://301a0bf636ce33a24659f5f8f43a61eb.digitalenergy.online/install_images/digitalenergy_911.iso

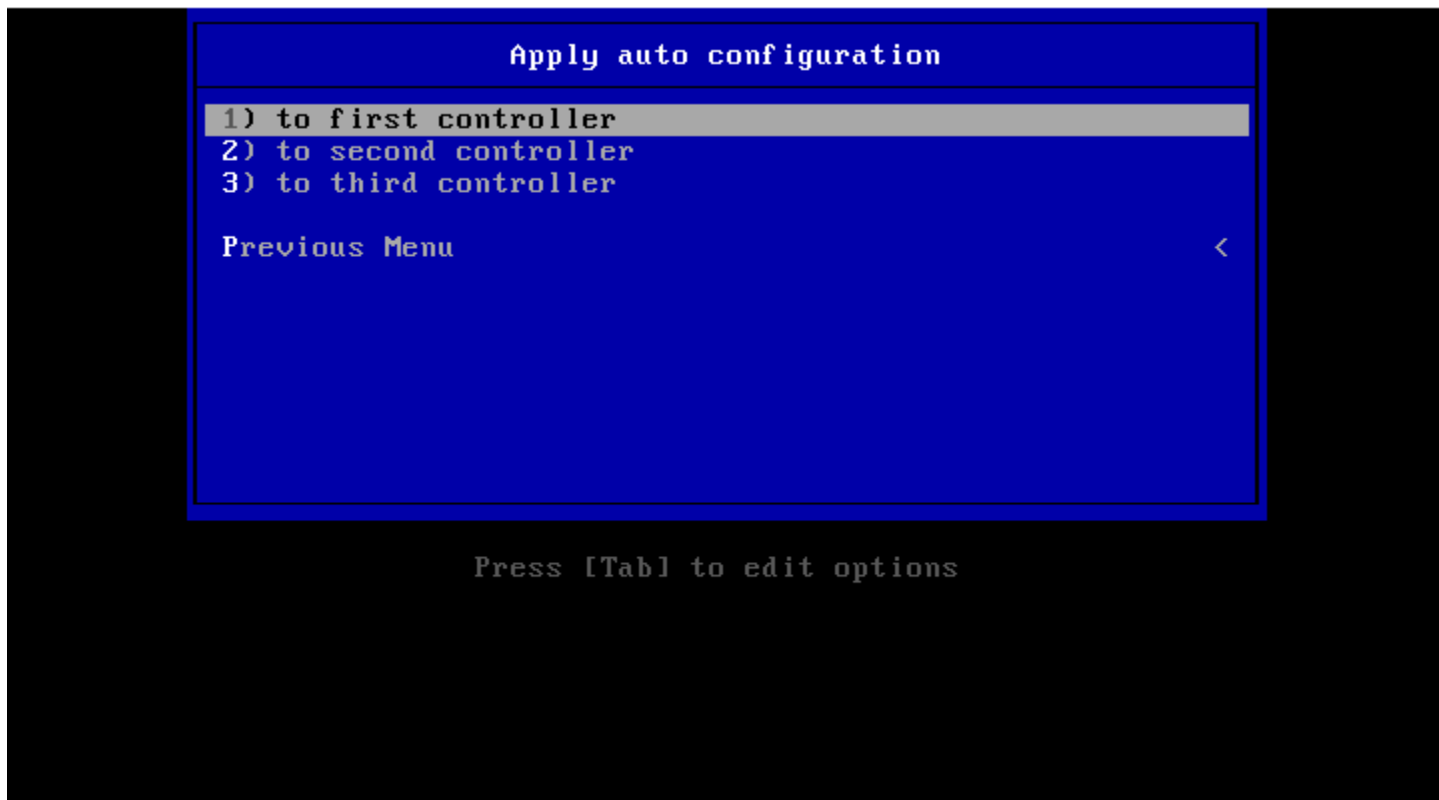
Образ может быть подключен к целевой машине любым удобным образом, после чего необходимо произвести загрузку машины с подготовленного образа.

Выберите один из пунктов, соответствующий тому, какие операции планируется выполнять. На выбор предоставляется два пункта: автоматически применить заранее предопределённую конфигурацию или сконфигурировать вручную.



В случае выбора ручной конфигурации загрузка продолжится запуском операционной системы с загрузочного устройства. В этом случае мы получим возможность отредактировать скрипт установки, расположенный в каталоге `/root/tools`

В том случае, если мы выбрали автоматическое применение предопределённой конфигурации, будет предложено уточнить выбор, указав, установка какого именно из трёх контроллеров производится в данный момент:



Помимо этого, существует также возможность отредактировать параметры загрузки ОС с загрузочного устройства через опции Grub. Для этого следует нажать Tab на этапе выбора контроллера

После запуска системы установка начнется автоматически и завершится сообщением.

```
Fri Oct 5 11:50:37 UTC 2018
Finished installing ctrl-01.
Please reboot without usb stick!!
```

2. Авторизуйтесь в системе с логином **root** и паролем **de_rooter**.

3. Авторизуйтесь пользователем root

```
sudo su -
```

4. Убедитесь, что два mdraid RAID1 собрались успешно.

```
cat /proc/mdstat
```

Также проверьте, что файловая система существует на обоих из них

```
fsck -N /dev/md*
```

Если это не так, необходимо отредактировать скрипт `~root/tools/CtrlInstall` между 36 и 59 строками (case) и запустить его вручную.

5. Перезагрузить машину без использования загрузочного образа - загрузить с жесткого диска.

6. Зайдите в систему с логином `d_energy` и паролем `d_energy_base` и получите полномочия администратора с помощью утилиты `sudo`

```
sudo su -
```

7. Смените пароль пользователя `d_energy` на более безопасный

8. Выполните команды

```
dpkg-reconfigure tzdata
```

```
dpkg-reconfigure openssh-server
```

9. Удалите имеющиеся `openvswitch` bridges.

```
ovs-vsctl del-br backplane1
```

```
ovs-vsctl del-br mgmt
```

```
ovs-vsctl del-br storage
```

10. Приведите конфигурацию сетевых интерфейсов к правильному виду, опираясь на следующий пример:

```
### backplane1
auto backplane1
iface backplane1 inet static
    address 10.211.1.1/24
    pre-up ip l set eno1 up
    pre-up sysctl -w net.ipv6.conf.eno1.disable_ipv6=1
    pre-up ip l set eno2 up
    pre-up sysctl -w net.ipv6.conf.eno2.disable_ipv6=1
    pre-up ovs-vsctl --may-exist add-br backplane1
    pre-up ovs-vsctl --may-exist add-bond backplane1 bond-backplane1 eno1 eno2
    pre-up ovs-vsctl set port backplane1 tag=2310

### mgmt
auto mgmt
iface mgmt inet static
    address 10.211.2.1/24
    pre-up ovs-vsctl --may-exist add-br backplane1
```

```
pre-up ovs-vsctl --may-exist add-port backplane1 mgmt -- set Interface mgmt type=internal
pre-up ip 1 set mgmt up
iface mgmt inet static
    address 10.211.4.1/24

### public
auto public
iface public inet static
    address 098.765.432.109/24
    gateway 098.765.432.1
pre-up ovs-vsctl --may-exist add-br backplane1
pre-up ovs-vsctl --may-exist add-port backplane1 public tag=2317 -- set Interface public
type=internal

### storage
auto storage
iface storage inet manual
pre-up ovs-vsctl --may-exist add-br storage
pre-up ovs-vsctl --may-exist add-br backplane1
pre-up ovs-vsctl --may-exist add-port backplane1 bkpln2stor tag=2323 -- set Interface bkpln2stor
type=patch options:peer=stor2bkpln
pre-up ovs-vsctl --may-exist add-port storage stor2bkpln -- set Interface stor2bkpln type=patch
options:peer=bkpln2stor

### reserve
auto reserve
iface reserve inet static
    address 123.456.789.012/30
    dns-nameservers 8.8.8.8 8.8.4.4
pre-up ovs-vsctl --may-exist add-port backplane1 wormhole tag=100 -- set Interface wormhole
type=internal

post-up ip route add 123.456.789.012/30 dev wormhole table wormhole
post-up ip route add default via 123.456.789.012/30 table wormhole
post-up ip rule add from 123.456.789.012/30 table wormhole
post-up ip rule add to 123.456.789.012/30 table wormhole
```


11. Проверьте, что сетевые интерфейсы подняты с корректными настройками, например:

```
ip a show backplane1  
ip l show backplane1
```

12. Теперь Вы можете подключиться к машине по ssh.

13. Отредактируйте hostname

```
hostnamectl set-hostname xxxxx  
vim /etc/hosts
```

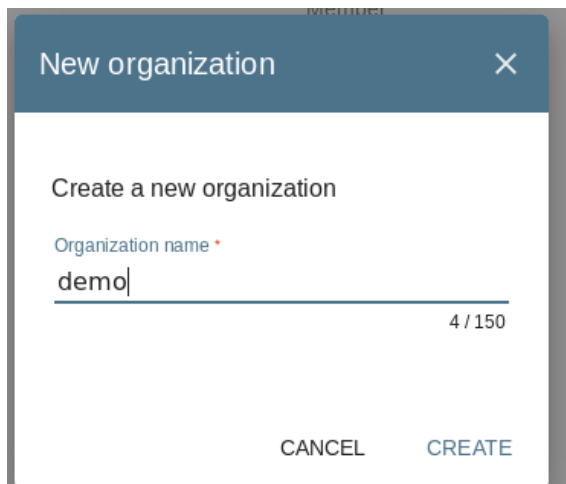
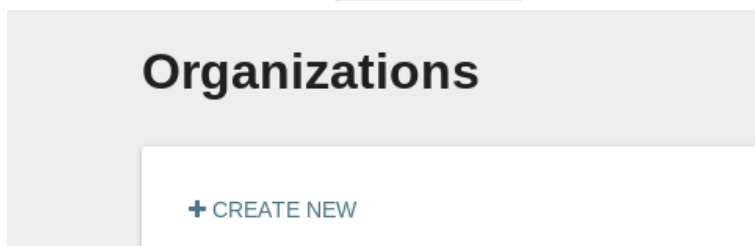
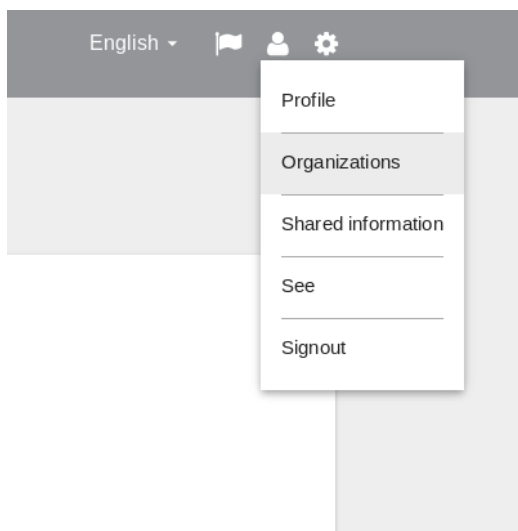
14. Повторите аналогичные действия для двух других контроллеров. После этого установка операционных систем на контроллеры будет завершена.

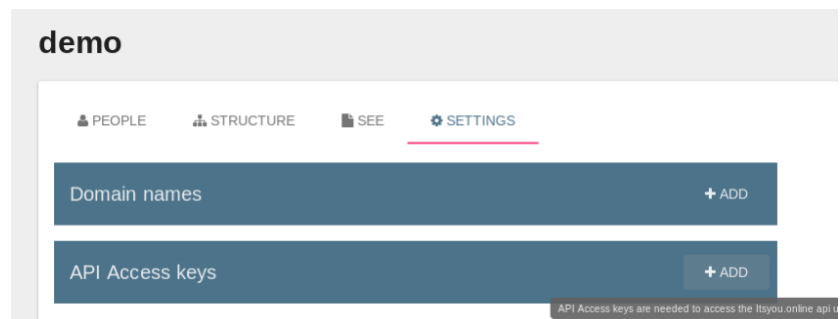
2.2 Развертывание управляющего кластера на основе kubernetes

2.2.1 Подготовка главного конфигурационного файла

1. Для установки используется внешний сервис авторизации - <https://itsyou.online>

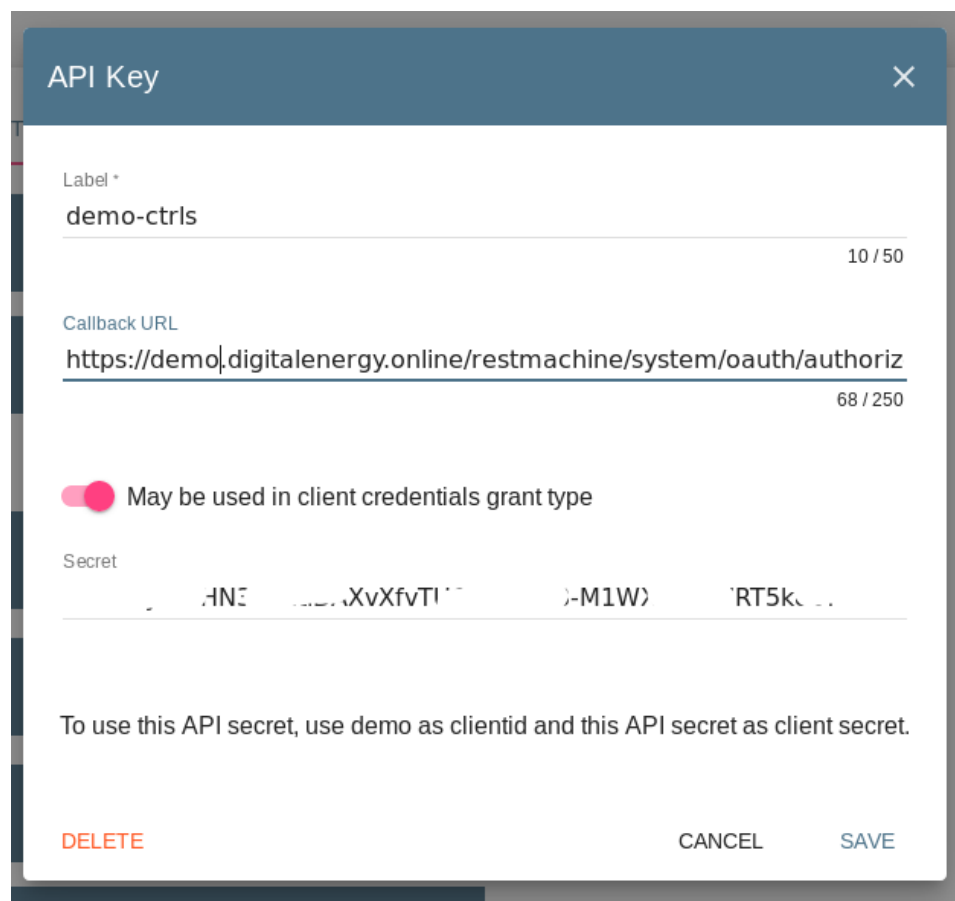
Перед началом установки необходимо зарегистрировать пользователя в этой системе и создать организацию.





Будьте внимательны при задании Callback URL. Доменное имя должно в точности совпадать с доменным именем, на котором отвечает портал. В противном случае авторизация работать не будет. Общий вид ссылки:

`https://domain.name/restmachine/system/oauth/authorize`



Запомним название организации и Secret.

2. Скачайте пример главного конфигурационного файла -

<https://301a0bf636ce33a24659f5f8f43a61eb.digitalenergy.online/demo/system-config.yaml>

Заполнение разделов конфигурационного файла:

2.1 itsyouonline:

clientSecret - это Secret, который мы получили выше

clientId и environment - название организации

2.2 mailclient:

Укажите данные для доступа к Вашему SMTP-серверу. Это необходимо для того, чтобы система могла отправлять оповещения.

2.3 network:

В этом разделе важными являются подразделы management, public, backplane, storage.

Рассмотрим на примере одного из них:

```
management :  
  vlan: 2311  
  network: 10.109.2.0/24  
  gateway: 10.109.2.254
```

vlan - если не требуется, укажите 0

network - подсеть, из которой будут выдаваться адреса для данного интерфейса

gateway - шлюз для этого интерфейса

2.4 ssh: - закрытый ключ, с помощью которого можно подключиться по ssh на каждый из трех контроллеров

2.5 certificates: - закрытый ключ и сертификат TLS Web-сервера, который будет использоваться для портала управления платформой.

2.6 environment: - назначение идентификатором платформы

grid: id - произвольное десятичное число

subdomain: + basedomain: составляют доменное имя, на котором будет работать портал

ssl: используемые сертификаты

password: пароль, который будет назначен некоторым интерфейсам, например виртуальным микротикам.

2.7 nodes: - описание физических узлов. Рассмотрим на примере одного из них, прокомментируем только неочевидные:

```
- name: demo-ctrl-01
  roles:
    - controller - роли бывают controller, cpu и storage
  ip-lsb: 1 - последний октет IP-адреса данного физического узла в сетях backplane, management и storage. Например, если мы указали для management в разделе network значение 10.109.2.0/24, то данный узел получит адрес 10.109.2.1
  fallback:
    ipaddress: 10.101.109.1/16 - публичный IP-адрес узла
    gateway: 10.101.0.1
  ipmi:
    macaddress: 0C:C4:7A:AC:11:36 - MAC-адрес сетевого интерфейса IPMI
    username: ADMIN
    password: ADMIN
  management:
    macaddress: 0A:C4:7A:AC:11:36 - MAC-адрес сетевого интерфейса management
```

Составьте аналогичную секцию для других узлов. Следует иметь в виду, что CPU и STOR узлы обычно не имеют публичного IP-адреса, соответственно секцию fallback для них следует убрать.

2.2.2 Развертывание кластера

1. iptables -I INPUT -I public -p tcp --dport 3080 -j ACCEPT
 2. apt-get update
 3. apt-get install apt-transport-https
 4. curl -fsSL <https://download.docker.com/linux/ubuntu/gpg> | apt-key add -
 5. add-apt-repository "deb [arch=amd64] https://download.docker.com/linux/ubuntu \$(lsb_release -cs) stable"
 6. apt-get update
 7. apt-get install -y docker-ce=17.03.0~ce-0~ubuntu-xenial
 8. ssh-keygen
 9. eval `ssh-agent`
 10. ssh-add
 11. docker run -d -it --name mgmt openvcloud/management
 12. docker ps
 13. docker cp ~/.ssh/id_rsa* d3912fd14cc6:~/.ssh/
 14. docker cp ~/.ssh/id_rsa.pub d3912fd14cc6:/root/.ssh/authorized_keys
 15. Добавьте этот же открытый ключ в авторизованные для пользователя root также и на 2 и 3 контроллерах
 16. ssh -A 172.17.0.2
 17. Проверьте, что удается соединиться по ssh со всеми тремя контроллерами
- ```
ssh 10.211.2.1
ssh 10.211.2.2
ssh 10.211.2.3
```

18. Разместите сертификаты TLS Web-сервера в каталоге /certpath контейнера

```
docker cp certpath d3912fd14cc6:/
```

19. Разместите главный конфигурационный файл в /

```
docker cp system-config.yaml d3912fd14cc6:/
```

20. `docker exec -it d3912fd14cc6 bash`

21. `mkdir -p /opt/code/github/0-complexity`

22. `cd /opt/code/github/0-complexity`

23. `git clone https://github.com/0-complexity/openvcloud_installer/`

24. `cd /opt/code/github/0-complexity/openvcloud_installer/scripts/install`

25. `./installer --config /system-config.yaml --version 2.4.5 cluster deploy`

В случае неудачного завершения развертывания кластера, необходимо исправить ошибки и недостатки конфигурации систем или конфигурационного файла, а затем выполнить сброс кластера м изначальному состоянию на каждом узле командами:

```
#!/bin/bash
kubeadm reset
systemctl stop etcd teleport
rm -rf /var/lib/teleport /tmp/etcd* /tmp/teleport /etc/etcd
/etc/kubernetes /var/lib/etcd /var/lib/kubernetes /etc/teleport.yaml
/usr/local/bin/*
```

Только после этого имеет смысл повторять попытку развертывания.

## 2.3 Установка ОС на узлы платформы

На этапе создания конфигурационного файла для развертывания управляющего кластера заполнялся раздел `nodes`, где, в числе прочего, перечислялись адреса, MAC и роли серверов. Скрипт инсталляции будет работать как раз с этими данным

Для установки операционной системы на отдельно взятом сервере требуется зайти в `pod management` (можно использовать `ssh`-доступ или штатные средства `kubernetes`):

```
1. kubectl get pods -o wide | grep management | awk '{print $6}'
```

```
10.244.2.35
```

```
2. ssh 10.244.2.35 -p 2205
```

```
root@ga-ctrl-01:~# kubectl get pods -o wide | grep management
management-785c48f77d-nxh4x 1/1 Running 0 3d 10.244.2.35 ga-ctrl-03
root@ga-ctrl-01:~# ssh 10.244.2.35 -p2205
Last login: Tue Nov 27 10:10:54 2018 from 10.244.2.1
root@management-785c48f77d-nxh4x:/#
```

Далее требуется выполнить команду скрипта `installer`, которая иницирует установку ОС

```
3. installer node action install_os -name <hostname>
```

при этом `<hostname>` следует заменить на значение поля `name` из раздела `nodes` в файле конфигурации `system-config.yaml`.

Допускается использование списка в качестве значения для параметра `-name`,

```
/# installer node action install_os --name demo-cpu-02,,demo-stor-01,demo-stor-02
```

например

В результате запуска произойдёт одно из следующих событий:

- скрипт завершится успешно. В этом случае операционная система будет установлена на целевом сервере, доступ гарантированно будет возможен по `mgmt`-адресу ноды при использовании протокола `ssh-agent`. При этом не гарантируется работоспособность иных интерфейсов. После установки следует проверить наличие интерфейсов `backplane1` и `storage` и их работоспособность

- скрипт завершится с ошибкой. В этом случае на экран будет выведено сообщение о причинах неуспешного завершения. Исходя из описания ошибки следует изменить конфигурацию на рабочую и попробовать снова.

## 2.4 Установка СХД OpenvStorage

Для установки OVS следует использовать `pod ovs`

```
root@ga-ctrl-01:~# kubectl exec -it `kubectl get pods | grep ovs | awk '{print $1}'` bash
root@ovs-6f45f878d8-g7g48:/#
```

В первую очередь следует подготовить установочную среду.

1. Установите пакеты `openvstorage-dev-ops/unknown` и `rsyslog`

```
apt install openvstorage-dev-ops
```

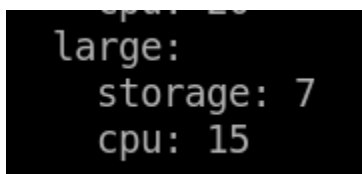
2. Перейдите в каталог `/opt/OpenvStorage/operations/Ansible/openvstorage/bin/` и выполните скрипт `pre-install.sh`

```
cd /opt/OpenvStorage/operations/Ansible/openvstorage/bin; bash pre-install.sh
```

Теперь следует подготовить конфигурационные файлы для установки OVS. Первым шагом на этом этапе будет подготовка шаблонов для генерации конфигурационных файлов

1. Перейдите в каталог `/opt/code/github/0-complexity/openvcloud_installer/scripts/ovs/`
2. Отредактируйте файл `specs.yaml` в соответствии с характеристиками используемого оборудования и типом инсталляции

- секция `nodes` содержит указание на тип инсталляции. В первую очередь следует отталкиваться от количества узлов хранения данных. До 4 узлов – `small`, до 6 – `medium`, до 8 – `large`. Отредактируйте значения полей `cpu` и `storage` в том типе, который соответствует планируемой конфигурации, внося актуальные значения. Например, при числе `cpu`-узлов 15 и `stor`-узлов 7 рекомендуется выставить значение так, как показано на рисунке



```
large:
 storage: 7
 cpu: 15
```

- секция `sizes` содержит указание объёмов применяемых в инсталляции дисков. Значения следует указывать в GiB

- секция `disks` описывает расположение и количество дисков, а также их тип, в нодах

3. Перейдите в каталог

`/opt/code/github/0-complexity/openvcloud_installer/scripts/ovs/templates`

В данном каталоге содержатся шаблоны для различных типов инсталляции – `small`, `medium`, `large`. Перейдите в нужный каталог (если нужный каталог отсутствует, создайте его путём копирования одного из существующих). Отредактируйте имеющиеся тут шаблоны конфигурационных файлов в соответствии с реализуемой архитектурой системы хранения данных

- файл `all` содержит в себе переменные, которые будут использоваться `ansible` при установке СХД. Помимо этого, здесь также содержится описание кластерной базы данных `arakoop`, на которой реализовано управление СХД, начиная от конфигурации самого `arakoop` и заканчивая описанием `cachebackend`'ов и `backend`'ов хранения данных, а также системы передачи сообщений между управляющими нодами хранилища

В описании кластеров `arakoop`, `cachebackend`'ов и `backend`'ов следует указать контроллер-ноды и и ноды для обработки пространства имён для каждого из бэкендов, а также указать пресеты

- файл `inventory` не требует редактирования при стандартной инсталляции

- файл `setup.json` содержит продолжение описания `backend`'ов различных типов, а также конфигурацию `vPool`'ов и применяемых ими `volumedriver`'ов. Основное, что здесь следует отредактировать – размеры `SCO` (`sco_size`), кластера (`cluster_size`), объём буфера записи для каждого виртуального диска (`volume_write_buffer`), общий объём



буфера записи для всех виртуальных дисков (`global_write_buffer`), используемый тип транспорта для `dtl` (`dtl_transport`) и его режим работы (`dtl_mode`), а также размер и число SCO в каждом DTL (`number_of_scos_in_tlog` и `non_disposable_scos_factor`).

Помимо этого, следует указать, в каком `backend` и `cachebackend` будут храниться данные и кэш для данного `vPool`

После завершения редактирования шаблонов следует перейти в каталог `/opt/code/github/0-complexity/openvcloud_installer/scripts/ovs`

и запустить из него скрипт `ovs_configurator.py` с требуемыми параметрами:

```
python3 ovs_configurator.py --config_path /opt/cfg/system/system-
config.yaml
```

В результате выполнения скрипта будет создан каталог `output`, в котором будут расположены сгенерированные на основе шаблонов конфигурационные файлы.

Создайте каталог

```
/opt/OpenvStorage/operations/Ansible/openvstorage/playbooks/inventories/my
env/group_vars
```

Скопируйте файл `all` в этот каталог, а файлы `inventory` и `setup.json` в каталог на один уровень выше относительно `group_vars`

Непосредственно установка осуществляется двумя командами. Первая выполняет подготовку среды, в которую будет производиться установка, вторая запускает непосредственно автоматизированные инсталляционные процессы.

Перейдите в каталог:

```
1. cd /opt/OpenvStorage/operations/Ansible/openvstorage/playbooks
```

Запустите процесс подготовки среды для установки OVS:

```
2. ansible-playbook -i inventories/myenv/inventory preInstall.yml
```

После успешного выполнения подготовительного этапа запустите процесс автоматизированной установки:

```
3. ansible-playbook -i inventories/myenv/inventory full_setup.yml
```

В процессе выполнения п.3 будут произведены следующие действия:

- созданы необходимые пользователи и установлены ключи авторизации
- установлен и настроен кластер для обмена сообщениями между узлами СХД
- установлены все перечисленные кластеры `aragoon` (включая служебные)
- установлены `framework OVS`

- установлены и сконфигурированы alba и backend'ы
- созданы vPools, запущены volumedriver'ы для них
- установлено требуемое ПО на вычислительные узлы (исправленная версия cemu, AlbaStorageDriver и прочее)

Выполнение п.3 занимает длительное время (порядка нескольких часов) и не всегда завершается успешно

В случае неуспешного завершения п.3 следует действовать в зависимости от возвращённой ошибки: либо проводить полную очистку системы затем перезапускать установку, либо перезапускать установку сразу после исправления. В некоторых ситуациях возможно завершение установки в ручном режиме.

## 2.5 Установка СУП-РВИ-DE на узлы платформы

После успешного завершения установки СХД мы готовы приступить к финальному этапу – добавлению нод в систему оркестрации. Данное действие осуществляется из pod'a management.

1. Зайдём в pod:

```
ssh `kubectl get pods -o wide | grep management | awk '{print $6}'` -p 2205
```

2. Выполним команду, которая установит необходимое ПО на каждый из узлов, сконфигурирует его и добавит информацию об узле в систему оркестрации:

```
installer node jsaction install --name demo-cpu-01
```

Данную команду следует выполнить для всех узлов, которые предполагается использовать в платформе, после выполнения описанных действий. Также, как и для команды установки ОС на узлы аплаенса, а качестве значения поля name принимается список имён узлов через запятую, без пробелов.

Установка завершена.

## 3 Авторизация в платформе

После успешной установки перейдите по ссылке <https://domain.name/cbgrid>, вы попадёте на страницу авторизации.

После успешного прохождения авторизации вы попадёте на страницу административного портала.