# Использование прокси-сервера при установке Nova Container Platform

В некоторых окружениях прямой доступ в сеть Интернет может быть ограничен, и подключение осуществляется только через HTTP или HTTPS прокси-сервер. Вы можете указать параметры подключения к прокси-серверу в Nova Container Platform на этапе установки платформы независимо от метода развертывания.

## 1. Предварительные действия

Ознакомьтесь со списком Интернет-ресурсов, которые используются во время установки Nova Container Platform, и определите, какие из ресурсов возможно исключить. По умолчанию весь трафик узлов платформы направляется через прокси-серверы. Однако использование прокси-сервера не влияет на конечные пользовательские сервисы. При необходимости вы можете добавить исключения для ресурсов в параметр `spec.noProxy` объекта Proxy.

При использовании прокси-сервера на узлах кластера автоматически добавляются следующие исключения `noProxy`:

- coreServer: Сервер управления Nova Universe при установке платформы в закрытом сетевом окружении.
- k8sDefaultDnsZone: Корневой DNS-домен кластера Kubernetes.
- k8sAPIDefaultFqdn: DNS-имя по умолчанию, используемое для доступа к серверу Kubernetes API.
- k8sAPIAdditionalSANs: Список дополнительных DNS-имен и IP-адресов (Subject Alternative Name) Kubernetes API.
- kubePodSubnet: Блок IP-адресов для pod'ов Kubernetes.
- kubeServiceAddresses: Блок IP-адресов для сервисной сети Kubernetes (Service Network).
- dnsBaseDomain: Базовый DNS-домен для настройки и публикации служебных веб-сервисов через Ingress-контроллер.
- **.svc**: Все ресурсы сервисов в служебном домене Kubernetes.
- **127.0.0.1**: Локальный хост.
- IP-адреса всех узлов платформы.
- Хостовые имена всех узлов платформы.

## 2. Настройка параметров прокси-сервера

В Nova Container Platform параметры прокси-сервера настраиваются через объект *Proxy* в Kubernetes. Перед установкой платформы вы можете сгенерировать необходимые конфигурационные манифесты с помощью `nova-ctl`.

По умолчанию `nova-ctl` предоставляет манифест *Proxy* с пустой спецификацией `spec`. Это означает, что прокси-сервер не будет использоваться при установке платформы. Пример данного манифеста представлен ниже.

YAML

```yaml
apiVersion: config.nova-platform.io/v1alpha1
kind: Proxy
metadata:
  name: cluster
spec: {}
```

Чтобы задать необходимые параметры воспользуйтесь спецификацией конфигурационного API.

*Пример подготовленного манифеста:*

YAML

```yaml
apiVersion: config.nova-platform.io/v1alpha1
kind: Proxy
metadata:
  name: cluster
spec:
  httpProxy:
    server: "http://172.31.100.100"
    port: 8080
    username: "changeme"
    password: "changeme"
  httpsProxy:
    server: "http://172.31.100.100"
    port: 8080
    username: "changeme"
    password: "changeme"
  noProxy: "hub.corp.internal,registry-1.corp.internal"
```

## 3. Настройка nova-ctl

При онлайн-установке платформы `nova-ctl` обращается как к публичным, так и внутренним ресурсам кластера. Поэтому если вы запускаете контейнер с утилитой `nova-ctl` в окружении, где требуется настройка прокси-сервера, вам необходимо установить исключения `noProxy`, аналогичные тем, что устанавливаются на узлах кластера автоматически, а также добавить дополнительные исключения при необходимости.

Вы можете настроить `nova-ctl`, используя команды ниже.

*Процедура*

1. Запустите `nova-ctl`:

```
BASH
docker run --rm -it -v $PWD:/opt/nova hub.nova-platform.io/public/nova/nova-ctl:v6.0.1
```

2. Установите переменные окружения с параметрами прокси-сервера в контейнере `nova-ctl`, например:

```
BASH
export HTTP_PROXY=http://changeme:changeme@172.31.100.100:8080
export HTTPS_PROXY=http://changeme:changeme@172.31.100.100:8080
export NO_PROXY=.cluster.local,.svc,10.233.0.0/18,10.233.64.0/18
```

## 4. Рекомендуется в выполнению

- **Установка платформы**

# Подготовка к установке

Установку Nova Container Platform следует планировать с учетом **сетевого окружения** и **метода развертывания**:

## 1. Установка в открытом сетевом окружении

Если у узлов есть доступ в интернет, можно сразу перейти к выбору метода установки:

- Автоматизированная установка (IPI)
- Универсальная установка (UPI)

## 2. Установка в закрытом контуре (без доступа к сети интернет)

1. Для установки в закрытом контуре необходимо установить и инициализировать сервер управления **Nova Universe**. Он поставляется в виде образа виртуальной машины и содержит все необходимые компоненты для установки и функционирования платформы. Для установки Universe перейдите Руководству по установке сервера управления Nova Universe
2. После установки Universe выберите один из способов установки:
    - Автоматизированная установка (IPI): при использовании автоматизированного метода установки (IPI) достаточно создать и настроить шаблон узла, который будет применяться в процессе установки. Далее узел `nova-ctl` самостоятельно развернет кластер, используя API поддерживаемой платформы виртуализации и частного облака. Это значительно упрощает и ускоряет развертывание всей системы.
    - Универсальная установка (UPI): при использовании универсального метода установки (UPI) вы самостоятельно подготавливаете узлы платформы. Это позволяет использовать как виртуальные, так и физические сервера.

# Интеграция с vSphere

Данный раздел содержит информацию по интеграции Nova Container Platform с vSphere. Раздел содержит следующие основные шаги:

- Подготовка шаблонов виртуальных машин в среде vSphere
- Настройка пользователя vSphere

## 1. Подготовка шаблонов виртуальных машин в среде vSphere

В данном разделе документации описывается процесс подготовки виртуальной машины и создание из неё шаблона, который будет использоваться для развертывания всех виртуальных узлов кластера Nova Container Platform.

## 2. Создание виртуальной машины

Воспользуйтесь официальной документацией VMware, чтобы создать новую виртуальную машину. Выберите версию документации, соответствующую вашей версии VMware vSphere.

Проверьте совместимость вашей версии VMware vSphere с Nova Container Platform в разделе Перечень матриц совместимости и протестированных интеграций.

При создании виртуальной машины следуйте рекомендациям ниже:

- Используйте адаптеры VMware Paravirtual SCSI (PVSCSI) для виртуальных дисков машин
- Используйте виртуальные сетевые адаптеры VMware VMXNET3

## 3. Настройка виртуальной машины

Выполните стандартную установку ОС. После завершения установки выполните следующие команды для настройки:

1. Выполните обновление всех пакетов ОС:

```
dnf update -y
```

2. Установите необходимые пакеты:

```bash
dnf install -y cloud-utils-growpart \
  perl cloud-init git vim tar open-vm-tools \
  rsync nmap-ncat tcpdump vim wget sysstat unzip
```

3. Установите конфигурацию службы `cloud-init` в файле `/etc/cloud/cloud.cfg`:

   ▸ **AlmaLinux**

   ▸ **РЕДОС**

4. Добавьте в автозагрузку службу `vmtoolsd`:

```bash
systemctl enable vmtoolsd
```

5. Добавьте в автозагрузку службу `cloud-init`:

```bash
systemctl enable cloud-init
```

6. Разрешите запуск пользовательских скриптов с помощью Open VM Tools:

```bash
vmware-toolbox-cmd config set deployPkg enable-custom-scripts true
```

7. Выключите межсетевой экран:

```bash
systemctl disable firewalld
```

8. Подготовьте пользовательскую учетную запись согласно статье.

9. Выключите виртуальную машину:

```bash
shutdown -h now
```

## 4. Создание шаблона из виртуальной машины

Воспользуйтесь официальной документацией VMware, чтобы сконвертировать подготовленную виртуальную машину в шаблон. Выберите версию документации, соответствующую вашей версии VMware vSphere.

Проверьте совместимость вашей версии VMware vSphere с Nova Container Platform в разделе Перечень матриц совместимости и протестированных интеграций.

# 5. Настройка пользователя в vSphere

Для автоматического создания ВМ, а также изменения их параметров и количества, вы можете использовать учетную запись администратора VMware vCenter. Однако рекомендуется создать служебного пользователя с ролью, имеющей ограниченный набор привилегий.

Воспользуйтесь официальной документацией VMware, чтобы подготовить необходимые учетные записи и роли. Выберите версию документации, соответствующую вашей версии VMware vSphere.

Проверьте совместимость вашей версии VMware vSphere с Nova Container Platform в разделе Перечень матриц совместимости и протестированных интеграций.

В таблице ниже представлен перечень привилегий в vSphere API, необходимых для установки и поддержки жизненного цикла кластеров Nova Container Platform.

| | **Привилегии в vSphere API** |
|---|---|
| vSphere vCenter | Cns.Searchable<br>InventoryService.Tagging.AttachTag<br>InventoryService.Tagging.CreateCategory<br>InventoryService.Tagging.CreateTag<br>InventoryService.Tagging.DeleteCategory<br>InventoryService.Tagging.DeleteTag<br>InventoryService.Tagging.EditCategory<br>InventoryService.Tagging.EditTag<br>Sessions.ValidateSession<br>StorageProfile.Update<br>StorageProfile.View |
| vSphere vCenter Cluster | Host.Config.Storage<br>Resource.AssignVMToPool<br>VApp.AssignResourcePool<br>VApp.ImportVirtualMachine.Config.AddNewDisk |
| vSphere vCenter Resource Pool | Host.Config.Storage<br>Resource.AssignVMToPool<br>VApp.AssignResourcePool<br>VApp.Import<br>VirtualMachine.Config.AddNewDisk |
| vSphere Datastore | Datastore.AllocateSpace<br>Datastore.Browse<br>Datastore.FileManagement<br>InventoryService.Tagging.ObjectAttachable |
| vSphere Port Group | Network.Assign |

| | **Привилегии в vSphere API** |
|---|---|
| Virtual Machine Folder | InventoryService.Tagging.ObjectAttachable<br>Resource.AssignVMToPool<br>VApp.Import<br>VirtualMachine.Config.AddExistingDisk<br>VirtualMachine.Config.AddNewDisk<br>VirtualMachine.Config.AddRemoveDevice<br>VirtualMachine.Config.AdvancedConfig<br>VirtualMachine.Config.Annotation<br>VirtualMachine.Config.CPUCount<br>VirtualMachine.Config.DiskExtend<br>VirtualMachine.Config.DiskLease<br>VirtualMachine.Config.EditDevice<br>VirtualMachine.Config.Memory<br>VirtualMachine.Config.RemoveDisk<br>VirtualMachine.Config.Rename<br>VirtualMachine.Config.ResetGuestInfo<br>VirtualMachine.Config.Resource<br>VirtualMachine.Config.Settings<br>VirtualMachine.Config.UpgradeVirtualHardware<br>VirtualMachine.Interact.GuestControl<br>VirtualMachine.Interact.PowerOff<br>VirtualMachine.Interact.PowerOn<br>VirtualMachine.Interact.Reset<br>VirtualMachine.Inventory.Create<br>VirtualMachine.Inventory.CreateFromExisting<br>VirtualMachine.Inventory.Delete<br>VirtualMachine.Provisioning.Clone<br>VirtualMachine.Provisioning.MarkAsTemplate<br>VirtualMachine.Provisioning.DeployTemplate |

| | **Привилегии в vSphere API** |
|---|---|
| vSphere<br>vCenter<br>Datacenter | InventoryService.Tagging.ObjectAttachable<br>Resource.AssignVMToPool<br>VApp.Import<br>VirtualMachine.Config.AddExistingDisk<br>VirtualMachine.Config.AddNewDisk<br>VirtualMachine.Config.AddRemoveDevice<br>VirtualMachine.Config.AdvancedConfig<br>VirtualMachine.Config.Annotation<br>VirtualMachine.Config.CPUCount<br>VirtualMachine.Config.DiskExtend<br>VirtualMachine.Config.DiskLease<br>VirtualMachine.Config.EditDevice<br>VirtualMachine.Config.Memory<br>VirtualMachine.Config.RemoveDisk<br>VirtualMachine.Config.Rename<br>VirtualMachine.Config.ResetGuestInfo<br>VirtualMachine.Config.Resource<br>VirtualMachine.Config.Settings<br>VirtualMachine.Config.UpgradeVirtualHardware<br>VirtualMachine.Interact.GuestControl<br>VirtualMachine.Interact.PowerOff<br>VirtualMachine.Interact.PowerOn<br>VirtualMachine.Interact.Reset<br>VirtualMachine.Inventory.Create<br>VirtualMachine.Inventory.CreateFromExisting<br>VirtualMachine.Inventory.Delete<br>VirtualMachine.Provisioning.Clone<br>VirtualMachine.Provisioning.DeployTemplate<br>VirtualMachine.Provisioning.MarkAsTemplate<br>Folder.Create<br>Folder.Delete |

# 6. Получение информации о VMware vSphere

## 6.1. Получение корневого сертификата vCenter

1. Скачайте корневые сертификаты vCenter с главной страницы его веб-интерфейса, как показано на рисунке ниже.

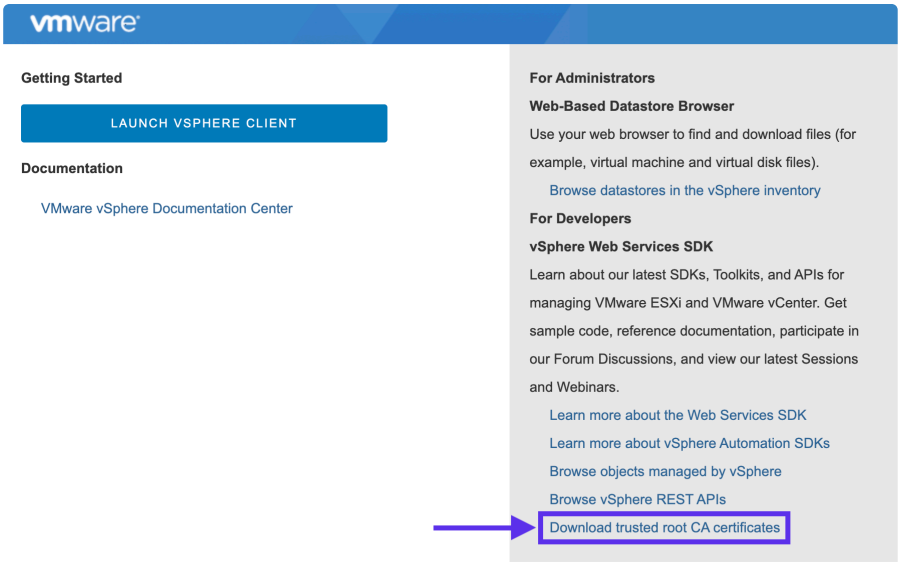


**Рисунок 1. Получение корневых сертификатов сервера VMware vCenter**

2. Распакуйте полученный архив, выполнив следующую команду:

```bash
unzip vcenter_certificates.zip

Archive:  vcenter_certificates.zip
  inflating: certs/lin/dbad4059.0
  inflating: certs/mac/dbad4059.0
  inflating: certs/win/dbad4059.0.crt
  inflating: certs/lin/8048c56c.r2
```

3. В директории `certs/lin` получите корневой CA-сертификат vCenter.

```bash
certs/lin
├── 5db219db.0
├── 5db219db.r1
├── 8048c56c.0
├── 8048c56c.1
├── 8048c56c.r0
├── 8048c56c.r2
├── dbad4059.0
├── dbad4059.1
├── dbad4059.r0
└── dbad4059.r1
```

Проверьте, что полученный сертификат действительно является корневым, выполнив команду:

```bash
openssl x509 -text -noout -in certs/lin/5db219db.0 | grep 'CA'

CA:TRUE, pathlen:0
```

где `5db219db.0` - имя проверяемого файла сертификата.

4. Закодируйте в `base64` полученный сертификат:

```bash
cat 5db219db.0 | base64 -w0
```

Закодированный сертификат потребуется вам в дальнейшем для установки платформы.

## 7. Рекомендуется к выполнению

- Подготовка пользовательской учетной записи