О К Т О П У С

СИСТЕМА КОМПЛЕКСНОЙ АНТИВИРУСНОЙ ПРОВЕРКИ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ

Руководство по развертыванию

Москва
2021г.

Контактная информация

127106, г. Москва, ул. Гостиничная, д.5
Тел.: +7 (495) 988-92-25
E-mail: office@avsw.ru
www.avsw.ru/about/contacts

Авторское право

Версия документа

Декабрь 20, 2021.

СОДЕРЖАНИЕ

1 Термины и определения

В настоящем документе используются термины и определения, представленные в таблице 1.

Таблица 1. Термины и определения

№	Термин	Определение
1.	Автоматический режим работы системы	Режим работы системы, в котором файлы и приложения, поступающие на интерфейс системы, автоматически загружаются в систему и анализируются на предмет нелегитимного поведения.
2.	Сессия исследования программного обеспечения	Последовательность действий, включающая в себя загрузку программного обеспечения в систему, анализ системой на предмет безопасности синтаксической структуры файла и фомирование отчета по проверке.
3.	Экспертный режим работы системы	Режим работы системы, в котором пользователь, имеющий роль аналитика, самостоятельно загружает файл или приложение для анализа в системе.
4.	Статический анализ	Проверка программного обеспечения локальными антивирусами, синтаксическим анализатором, внешними аналитическими ресурсами и нейронной сетью.

2 Сокращения и значения

В настоящем документе используется перечень сокращений, представленный в таблице 2.

Таблица 2. Сокращения и значения

№	Сокращение	Значение
1.	AD	Active Directory
2.	API	Application programming interface
3.	CPU	Central processing unit
4.	FTP	File transfer protocol
5.	KVM	Kernel-based virtual machine
6.	LDAP	Lightweight directory access protocol
7.	OSI	Open systems interconnection model
8.	RAID	Redundant array of independent disks
9.	SAS	Serial attached SCSI
10.	SIEM	Security information and event management
11.	SSD	Solid-state drive
12.	SSL	Secure sockets layer
13.	USB	Universal serial bus
14.	VNC	Virtual network computing
15.	БД	База данных
16.	ЛСС	Логическая схема сети
17.	ОС	Операционная система
18.	ПО	Программное обеспечение

3 Общие сведения о программе

Система комплексной антивирусной проверки программного обеспечения Октопус (далее – система Октопус) предназначена для усиления безопасности ИТ- инфраструктуры. Система Октопус представляет собой программный комплекс, который относится к классу систем антивирусных мультисканеров.

Основные технологии, используемые в системе Октопус, представлены в таблице 3.

Таблица 3. Программное обеспечение для системы Октопус

№	Наименование программного обеспечения	Версии
1.	ASP.NET Core Runtime	5.0
2.	Debian	9, 10
3.	Docker	20.10.7
4.	Flask	2.0.1
5.	Grafana	8.2.6 или 8.3.0-бета
6.	MongoDB	5.0.2
7.	MySQL	14.14
8.	MySQL Connector/Python 3	8.0.16
9.	PostgreSQL	9.4.21
10.	Prometheus	2.32.1
11.	Python	3.8.7
12.	Python 3 Pika	1.2.0
13.	RabbitMQ	3.7.17
14.	Nginx	1.14.12

3.1 Основные возможности

Система Октопус принимает на проверку файлы из различных типов источников, которые включают в себя:

- веб-трафик (ICAP)
- почтовый трафик (SMTP)
- сетевой трафик
- мессенджеры
- рабочие станции
- сервера
- ручная загрука
- API и др.

Система Октопус поддерживает проверку следующих типов объектов:

- файлы
- архивы
- многотомные архивы
- запароленные файлы и архивы
- веб-ссылки
- мобильные приложения

Система Октопус принимает на анализ любые типы файлов, примеры:

- исполняемые файлы (EXE, ELF, CMD)
- офисные документы (DOCX, XLSX, PPTX, PDF, RTF)
- мобильные приложения (APK)
- архивы, включая многотомные и запароленные (ZIP, JAR)
- скрипты (BAT, SH) и др.

3.2 Возможности развертывания

Программный комплекс Октопус поддерживает развертывание на следующих типах инфраструктур:

- Виртуальная
- Физическая
- Облачная

4 Требования

4.1 Квалификационные

Перед началом работы с настоящим документов рекомендуется ознакомиться с руководством пользователя системы Октопус.

Требования к специалистам, осуществляющим развертывание и администрирование системы Октопус:

- уверенное знание операционной системы (далее - ОС) на базе ядра Linux
- знание основ сетевого администрирования
- знание технологий контейнеризации (Docker)

4.2 Технологические

Для развёртывания системы Октопус необходимо использовать серверное оборудование с характеристиками не хуже, указанных в таблице 4.

Таблица 4. Характеристики оборудования

№	Параметр	Минимальные	1000 файлов в час
1.	Количество виртуалок	16	32
2.	Модель процессора	Intel(R) Xeon(R) CPU E5-2603 v4 @1.7GHz	Intel(R) Xeon(R) CPU E5-2603 v4 @1.7GHz
3.	Количество процессоров	2	4
4.	Количество ядер процессора	6	24
5.	Оперативная память	128 ГБ	256 ГБ
6.	Диск	SAS 1 TB x 8 RAID10	SAS 2 TB x 8 RAID10
7.	Сеть	10/100/1000 Мбит/с (2 шт.)	10/100/1000 Мбит/с (2 шт.)

4.3 Требования к браузерам

В таблице 5 представлены минимальные требования к версиям браузера, необходимые для функционирования веб-интерфейса системы Октопус.

Таблица 5. Минимальные версии браузера

№	Браузер	Версия браузера
1.	Chrome	80
2.	Edge	80
3.	Firefox	74
4.	Opera	67
5.	Safari	13.1

4.4 Требования к сетевым схемам

Для интеграции и координации обновлений программного комплекса Октопус в ИТ-инфраструктуру инженерам компании АВ Софт необходимо предоставить логические схемы сети (далее – ЛСС) уровней L1/2 и L3 модели взаимодействия открытых систем OSI и таблицу маршрутизации. ЛСС должна отображать компоненты сети и средства взаимодействия между ними. Подробное описание всех компонентов ЛСС представлено в таблице 6.

Таблица 6. Данные логической схемы сети

№	Параметры	Описание
1.	Подсети	LAN, VLAN
2.	Идентификаторы	Идентификаторы VLAN, маски и адреса
3.	Протоколы сетевой маршрутизации	IPv4, IPv6
4.	Сетевые устройства	Межсетевые экраны, маршрутизаторы, сетевые концентраторы

5 Установка

5.1 Подготовка сетевой инфраструктуры

Необходимо настроить сетевые интерфейсы в соответствии с согласованной схемой подключения стенда. Для этого необходимо изменить файл `/etc/network/interfaces` или воспользоваться NetworkManager-ом. Для внутренних нужд инфраструктуры необходимо назначить интерфейсу `lo` адрес `10.1.0.2/32`. Пример:

```
auto lo
iface lo inet loopback
iface lo inet static
    address 10.1.0.2/32
```

5.2 Подготовка репозиториев

Для функционирования системы необходимо установить на сервер программное обеспечение, указанное в пункте 3 настоящего документа.

Для установки вышеперечисленных пакетов необходимо подключить репозитории каждого. Для этого необходимо создать несколько файлов:

```
postgres.list
deb http://apt.postgresql.org/pub/repos/apt/ buster-pgdg main
mongo.list
deb https://repo.mongodb.org/apt/debian buster/mongodb-org/5.0 main
docker.list
deb [arch=amd64 signed-by=/usr/share/keyrings/docker-archive-keyring.gpg] https://download.docker.com/linux/debian   buster stable
```

Созданные файлы необходимо положить в папку следующей командой:

```
/etc/apt/sources.list.d.
```

Далее необходимо добавить в систему публичные ключи, которыми подписаны репозитории, выполнив следующую последовательность команд:

```
wget -qO - https://www.mongodb.org/static/pgp/server-5.0.asc | sudo apt-key add –

wget -qO - https://www.postgresql.org/media/keys/ACCC4CF8.asc | sudo apt-key add –

wget -qO - https://download.docker.com/linux/debian/gpg | sudo gpg --dearmor -o /usr/share/keyrings/docker-archive-keyring.gpg
```

После добавления репозиториев и добавления ключей необходимо выполнить команду обновления базы данных пакета следующей командой:

```
sudo apt update
```

Далее выполнить установку вышеперечисленных пакетов следующей командой:

```
sudo apt install имя пакета
```

5.3 База данных PostgreSQL

В конфигурационном файле БД PostgreSQL, находящимся по пути /etc/postgresql/12/main/postgresql.conf, необходимо изменить параметр listen_addresses, назначив ему значение 127.0.0.1,10.1.0.2.

Далее необходимо выполнить разрешение авторизации с указанных адресов при помощи изменения конфигурационного файла, находящегося по пути /etc/postgresql/12/main/pg_hba.conf, где необходимо добавить строку, разрешающую подключение по 10.1.0.2/32. Пример строки:

```
host   all    all      10.1.0.2/32     md5
```

После изменения конфигурационных файлов необходимо перезапустить сервис `PostgreSQL` следующей командой:

```
systemctl restart postgresql
```

Для работы необходимо создать БД под названием "sandBox" (соблюдая регистр). Для этого необходимо подключиться к PostgreSQL любым клиентом и выполнить следующий SQL запрос:

```
CREATE DATABASE "sandBox"
```

Учетные данные для БД передаются вендором в отдельном файле.

Для изменения пароля postgres необходимо подключиться к PostgreSQL любым клиентом и выполнить следующий SQL запрос:

```
ALTER USER postgres WITH PASSWORD 'TestPassV3TTR'
```

Для работы системы с первоначальными данными необходимо восстановить дамп БД, который поставляется отдельным файлом.

Дамп создаётся на работающей БД следующей командой:

```
pg_dumpall -c -f путь_к_файлу_дампа
```

Далее необходимо выполнить восстановление следующей командой:

```
pg_restore -d sandBox -C путь_к_файлу_дампа
```

Затем необходимо настроить автозапуск БД при помощи постановки сервиса systemd в состояние enabled следующей командой:

```
pg_dumpall -c -f путь_к_файлу_дампа
```

5.4 База данных MongoDB

1. В конфигурационном файле БД MongoDB, находящимся по пути /etc/mongod.conf, необходимо в секции net параметр bindIp задать адреса взаимодействия 127.0.0.1,10.1.0.2. (Обратите внимание на отсутствие пробела между адресами!)

2. Для автозапуска БД необходимо поставить сервис systemd в состояние enabled, выполнив следующую команду:

```
systemctl enable mongod
```

3. После изменения настроек необходимо перезапустить сервис MongoDB следующей командой:

```
systemctl restart mongod
```

5.5 Подготовка менеджера очередей RabbitMQ

1. Для менеджера очередей RabbitMQ в конфигурационном файле, находящемуся по пути /etc/rabbitmq/rabbitmq.conf, необходимо указать следующие адреса взаимодействия:

```
listeners.tcp.local = 127.0.0.1:5672

listeners.tcp.legacy = 10.1.0.2:5672
```

2. Далее необходимо выполнить настройку автозапуска при помощи постановки сервиса systemd в состояние enabled, выполнив следующую команду:

```
systemctl enable rabbitmq-server
```

3. После изменения настроек необходимо перезапустить службу rabbitmq-server следующей командой:

```
systemctl restart rabbitmq-server
```

4. Для авторизации в сервисе RabbitMQ необходимо воспользоваться выданными представителями компании АВ Софт идентификационными данными от учетных записей.

5. Добавление учётных записей осуществляется с помощью утилиты rabbitmqctl. Для добавления новой учётной записи необходимо выполнить следующую команду:

```
rabbitmqctl add_user логин "пароль"
```

6. Для выдачи прав администратора необходимо выполнить команду:

```
rabbitmqctl set_user_tags логин administrator
```

7. Для выдачи прав доступа в корень виртуальной файловой системы необходимо выполнить команду:

```
rabbitmqctl  set_permissions -p / логин ".*" ".*" ".*"
```

8. Для включения плагина менеджмента (открывает порт 0.0.0.0:15672 для доступа к веб интерфейсу менеджмента RabbitMQ) необходимо выполнить следующую команду:

```
rabbitmq-plugins enable rabbitmq_management
```

9. Далее необходимо выполнить установку ASP ASP.NET Core Runtime следующей последовательностью команд:

```
wget https://packages.microsoft.com/config/debian/11/packages-microsoft-prod.deb -O packages-microsoft-prod.deb
```

```
sudo dpkg -i packages-microsoft-prod.deb

rm packages-microsoft-prod.deb
```

10. Далее необходимо выполнить установку пакета SDK следующей последовательностью команд:

```
sudo apt-get update; \
  sudo apt-get install -y apt-transport-https && \
  sudo apt-get update && \
  sudo apt-get install -y dotnet-sdk-6.0
```

11. Далее необходимо выполнить установку MySQL Connector/Python 3 следующей последовательностью команд:

```
wget -q https://dev.mysql.com/get/Downloads/Connector-Python/mysql-connector-python-py3_8.0.26-1debian10_amd64.deb

sudo apt install ./mysql-connector-python-py3_8.0.26-1debian10_amd64.deb
```

12. Далее необходимо выполнить установку клиентской службы Python 3 Pika следующей последовательностью команд:

```
wget http://ftp.ru.debian.org/debian/pool/main/p/python-pika/python3-pika_1.2.0-1_all.deb

sudo apt install ./python3-pika_1.2.0-1_all.deb
```

5.6 Установка пакетов системы

Для установки и развертывания системы необходимо установить следующий список пакетов:

- octopus-interface
- octopus-service-agents
- octopus-service-analytic
- octopus-service-audit

- octopus-service-discovery
- octopus-service-linkanalytic
- octopus-service-maintance
- octopus-service-mobile
- octopus-service-networkanalytic
- octopus-service-reports
- octopus-service-researches
- octopus-service-researchevents
- octopus-service-securityalerts
- octopus-service-siem
- octopus-service-smtpnotifications
- octopus-service-staticanalytic
- octopus-service-useraccount
- octopus-service-userauth
- core- octopus-service-storagekeeper

Для установки пакетов необходимо выполнить следующую команду:

```
apt install имя-пакета
```

Пакеты зависят от aspnetcoreruntime v5.0. Для его установки необходимо выполнить следующую последовательность команд:

```
wget https://packages.microsoft.com/config/debian/10/packages-microsoft-prod.deb -O packages-microsoft-prod.deb
sudo dpkg -i packages-microsoft-prod.deb
rm packages-microsoft-prod.deb

sudo apt-get update; \
sudo apt-get install -y apt-transport-https && \
sudo apt-get update && \
sudo apt-get install -y aspnetcore-runtime-5.0

https://docs.microsoft.com/ru-ru/dotnet/core/install/linux-debian
```

5.7 Настройка nginx

1. Скопировать и установить nginx-athena-config_1.0.0_amd64.deb на нужный сервер.
2. Установить Nginx следующими командами:

```
sudo apt update
sudo apt install nginx
```

3. Сгенерировать SSL сертификат: Во время создания SSL-сертификата происходит последовательная обработка следующих видов ключей, описанных в таблице 7.

Таблица 7. Описание видов ключей

№	Тип	Описание
1.	.key	Ключи шифрования, открытий и/или закрытый.
2.	.csr	Ключ, содержащий сформированный запрос для получения подписи сертификата от центра сертификации, а сам запрос — это открытый ключ и информация о домене и организации, связанной с ним.
3.	.crt, .cer, .pem	Сертификат, подписанный центром сертификации по запросу из файла .csr.

Для начала нужно создать закрытый ключ, выполнив следующую команду:

```
openssl genrsa -des3 -out server.key 2048
```

Команда genrsa генерирует RSA-ключ, опция -des3 указывает алгоритм шифрования ключа. А опция -out указывает, что ключ должен быть получен в виде файла server.key. При выполнении этой команды пользователю будет предложено ввести пароль для шифрования. Поскольку указан его алгоритм опцией -des3.

Далее необходимо создать запрос на подпись — CSR-файл, который будет включать только что сгенерированный ключ server.key:

```
openssl req -new -key server.key -out server.csr
```

При выполнении этой команды пользователю необходимо ввести информацию о домене и организации. Причём наименование домена следует вводить точно, например, если идентификатор URL сайта https://mycompany.com, то ввести нужно mycompany.com.

Для того чтобы получить подписанный сертификат, нужно подписать его тем же ключом, который использовался при создании файла CSR. Для этого следует выполнить следующую команду:

```
openssl x509 -signkey server.key -in server.csr -req -days 365 -out server.crt
```

Параметр -x509 задаёт формат генерируемого сертификата. Он является самым распространённым и используется в большинстве случаев. Опция -new позволяет запрашивать информацию для запроса у пользователя.

4. Настройка HTTPS на Nginx: Чтобы настроить HTTPS-сервер, необходимо включить параметр ssl на слушающих сокетах в блоке server, а также указать местоположение файлов с сертификатом сервера и секретным ключом:

```
server {
    listen              443 ssl;
    server_name         www.example.com;
    ssl_certificate     www.example.com.crt;
    ssl_certificate_key www.example.com.key;
    ssl_protocols       TLSv1 TLSv1.1 TLSv1.2;
    ssl_ciphers         HIGH:!aNULL:!MD5;
    ...
}
```

Сертификат сервера является публичным. Он посылается каждому клиенту, соединяющемуся с сервером. Секретный ключ следует

хранить в файле с ограниченным доступом (права доступа должны позволять главному процессу nginx читать этот файл).

Секретный ключ можно также хранить в одном файле с сертификатом:

```
ssl_certificate     www.example.com.cert;

ssl_certificate_key www.example.com.cert;
```

5.8 Модуль статического анализа

Для установки модуля статического анализа необходимо выполнить следующую последовательность команд:

1. Установка docker-ce docker-compose (оба находятся в зависимостях).
2. Создать сеть докера, используя следующую команду:

```
docker network create --driver bridge NETWORK --subnet 172.18.0.0/16
```

3. Добавить в /etc/hosts следующие адреса:

```
172.18.2.2      mysqlstatic.avsw.ru
172.18.2.4      rabmqstatic.avsw.ru
```

4. Скопировать архивы с антивирусами av_data.zip и av_containers.tar на нужный сервер.
5. Выполнить загрузку образов антивирусов, используя команду:

```
docker load < av_containers.tar
```

6. Установить основной установочный пакет модуля статики, используя следующую команду:

```
sudo apt install core-static-static
```

7. Добавить в iptables правила для доступа isolate в NETWORK
8. Добавить сервис `iptablesrestore.service'
9. Создать файл /etc/systemd/system/iptablesrestore.service
10. Добавить содежимое файла iptablesrestore.service

```
[Unit] Description=Restore iptables firewall rules After=docker.service #Befo
re=network-pre.target

[Service] Type=forking ExecStart=/sbin/iptables-restore -n /etc/iptables.conf

[Install] WantedBy=multi-user.target
```

11. Выполнить обновление службы systemd командой:

```
systemctl daemon-reload
```

12. Добавить сервис в автозагрузку командой:

```
systemctl enable iptablesrestore.service
```

13. Установить правила фильтрации:
    - Создать файл /etc/iptables.conf
    - Добавить содержимое файла iptables.conf

```
*filter -A INPUT -p tcp -m state --state NEW -m tcp --dport 21:56000 -j ACCEPT

:INPUT ACCEPT [0:0] :FORWARD ACCEPT [0:0] :OUTPUT ACCEPT [0:0]
:DOCKER - [0:0] :DOCKER-ISOLATION-STAGE-1 - [0:0] :DOCKER-
ISOLATION-STAGE-2 - [0:0] :DOCKER-USER - [0:0] :ATHENA-FW - [0:0]
:ATHENA-INPUT-MGMT - [0:0] :ATHENA-INPUT-INTERNET - [0:0]
:ATHENA-INPUT-TRAPS - [0:0]

-I FORWARD -j ATHENA-FW

-A ATHENA-FW -s 172.17.0.0/16 -j ACCEPT -A ATHENA-FW -d 172.17.0.0/16
-j ACCEPT
```

-A ATHENA-FW -s 10.230.0.0/16 -j ACCEPT -A ATHENA-FW -d 10.230.0.0/16 -j ACCEPT -A ATHENA-FW -s 172.18.1.124 -d 10.1.5.0/24 -j ACCEPT

-A ATHENA-FW -s 192.168.100.0/24 -d 10.1.0.5 -j ACCEPT -A ATHENA-FW -s 192.168.100.0/24 -d 172.18.0.0/16 -j ACCEPT -A ATHENA-FW -s 192.168.100.0/24 -d 172.17.0.0/16 -j ACCEPT

-A ATHENA-FW -s 192.168.100.0/24 -d 192.168.100.1/24 -j ACCEPT -A ATHENA-FW -s 192.168.100.0/24 -d 192.168.100.0/24 -j DROP -A ATHENA-FW -s 192.168.100.0/24 -d 192.168.0.0/16 -j DROP -A ATHENA-FW -s 192.168.100.0/24 -d 172.18.0.0/12 -j DROP -A ATHENA-FW -s 192.168.100.0/24 -d 10.0.0.0/8 -j DROP

-A ATHENA-FW -s 10.11.0.0/24 -j ACCEPT -A ATHENA-FW -d 10.11.0.0/24 -j ACCEPT -A ATHENA-FW -s 10.99.0.0/16 -d 172.18.0.0/16 -j ACCEPT -A ATHENA-FW -s 172.18.0.0/16 -d 10.99.0.0/16 -j ACCEPT -A ATHENA-FW -s 172.18.0.0/16 -d 10.1.1.0/24 -j ACCEPT -A ATHENA-FW -s 10.1.1.0/24 -d 172.18.0.0/16 -j ACCEPT -A ATHENA-FW -s 172.17.0.0/16 -d 172.18.0.0/16 -j ACCEPT

-A ATHENA-FW -s 172.18.0.0/16 -d 172.17.0.0/16 -j ACCEPT -A ATHENA-FW -s 10.1.0.0/24 -d 172.18.0.0/16 -j ACCEPT -A ATHENA-FW -s 172.18.0.0/16 -d 10.1.0.0/24 -j ACCEPT

-A ATHENA-FW -s 192.168.100.0/24 -j ACCEPT -A ATHENA-FW -d 192.168.100.0/24 -j ACCEPT

-A ATHENA-FW -j RETURN

-I INPUT -i eno2 -j ATHENA-INPUT-MGMT

-A ATHENA-INPUT-MGMT -p icmp -j ACCEPT -A ATHENA-INPUT-MGMT -p tcp -m tcp -j ACCEPT

-A ATHENA-INPUT-MGMT -m conntrack --ctstate ESTABLISHED,RELATED -j ACCEPT -A ATHENA-INPUT-MGMT -j REJECT --reject-with icmp-port-unreachable -A ATHENA-INPUT-MGMT -j RETURN

-A ATHENA-INPUT-INTERNET -p icmp -j ACCEPT -A ATHENA-INPUT-INTERNET -m conntrack --ctstate ESTABLISHED,RELATED -j ACCEPT -A ATHENA-INPUT-INTERNET -j REJECT --reject-with icmp-port-unreachable -A ATHENA-INPUT-INTERNET -j RETURN

COMMIT

14. Перезапустить сервис командой:

```
systemctl restart iptablesrestore.service
```

15. Установить остальные пакеты командой:

```
sudo apt install core-static-*
```

16. Распаковать данные для антивирусов командой:

```
sudo 7z x av_data.zip -o/avsoft/static/
```

6 Vmware

Настройка на виртуальной инфраструктуре осуществляется посредством создания виртуальных машин со следующим набором модулей, описанным в таблице 8.

Таблица 8. Список модулей для установки

№	Модуль	Описание
1.	octopus-balancer	Модуль распределения внешних запросов к системе по модулям приема файлов.
2.	octopus-infrastructure	Модуль баз данных и менеджера очередей.
3.	octopus-kernel	Модуль обработки управления системой.
4.	octopus-static	Модуль сигнатурной и эвристической проверки объектов анализа.
5.	octopus-web	Модуль графического интерфейса.

6.1 Настройка сети

Необходимо выполнить настройку сети для виртуальных машин с типами сетей, которые описаны в таблице 9.

Таблица 9. Типы сетей

№	Типы	Описание
1.	octopus-internal	Сеть, объединяющая все виртуалки между собой.
2.	load-network	Сеть внешняя для загрузки файлов.
3.	mgmt-network	Сеть менеджмента для управления.
4.	db-replication	Сеть для кластера инфраструктуры.

Для настройки сети необходимо выполнить переход в VSphere Client «Host and Clusters» - выбрать хост - «Configure» - «Networking» - «Virtual switches» - «Add Networking» (Рисунок 1)

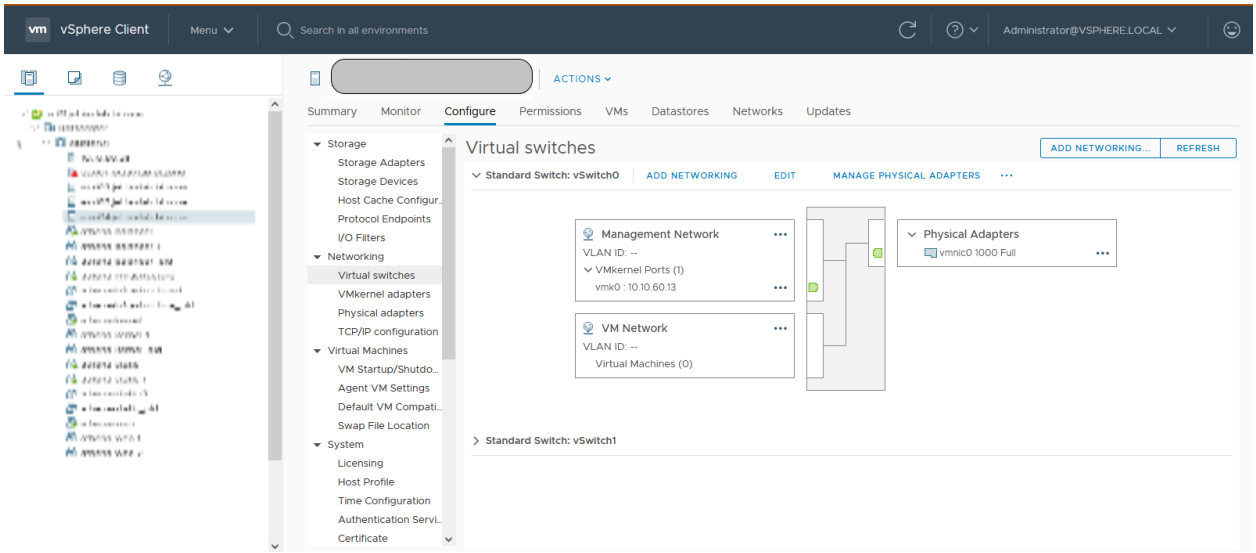

Рисунок 1. Настройка сети для хоста

Далее необходимо в мастере настройки в разделе «Select connection type» - выбрать «Virtual Machine Port Group for a Standard Switch» и нажать кнопку «Next» (Рисунок 2).

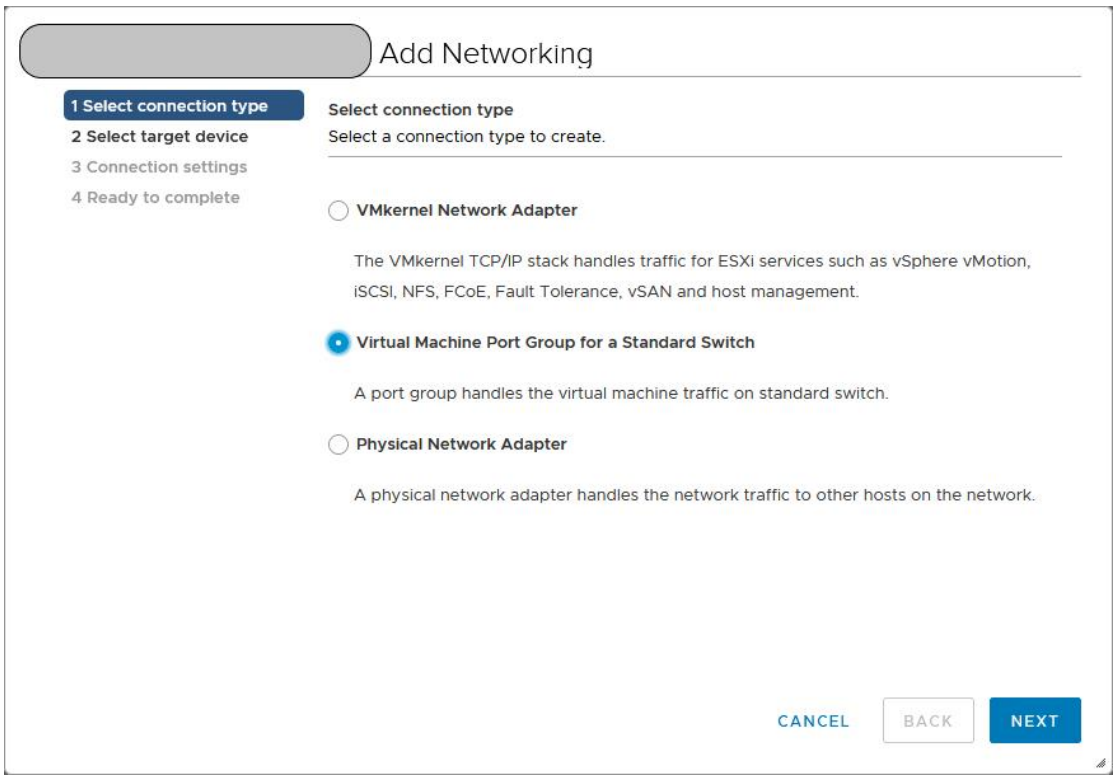

Рисунок 2. Выбор типа подключения

Далее необходимо в разделе «Select target device» указать switch и по окончании ввода данных нажать кнопку «Next» (Рисунок 3).

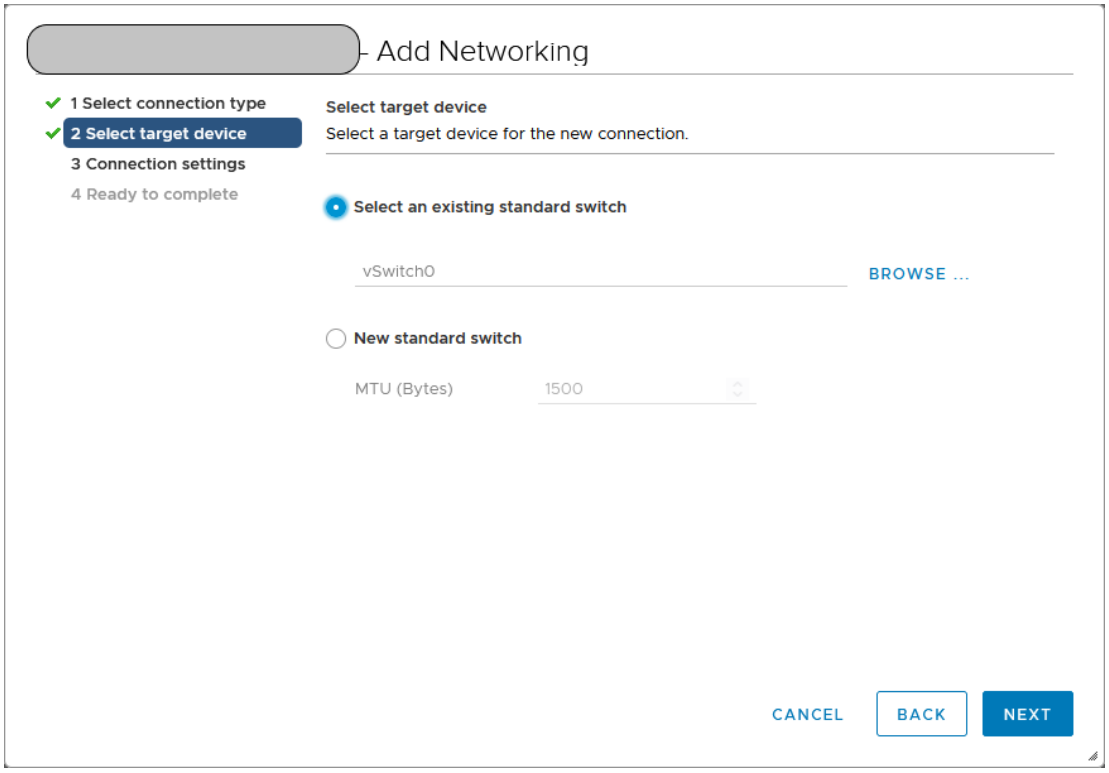

Рисунок 3. Выбор switch

Далее необходимо в разделе «Connection settings» указать Network label и VLAN ID, по окончании ввода данных нажать кнопку «Next» (Рисунок 4).

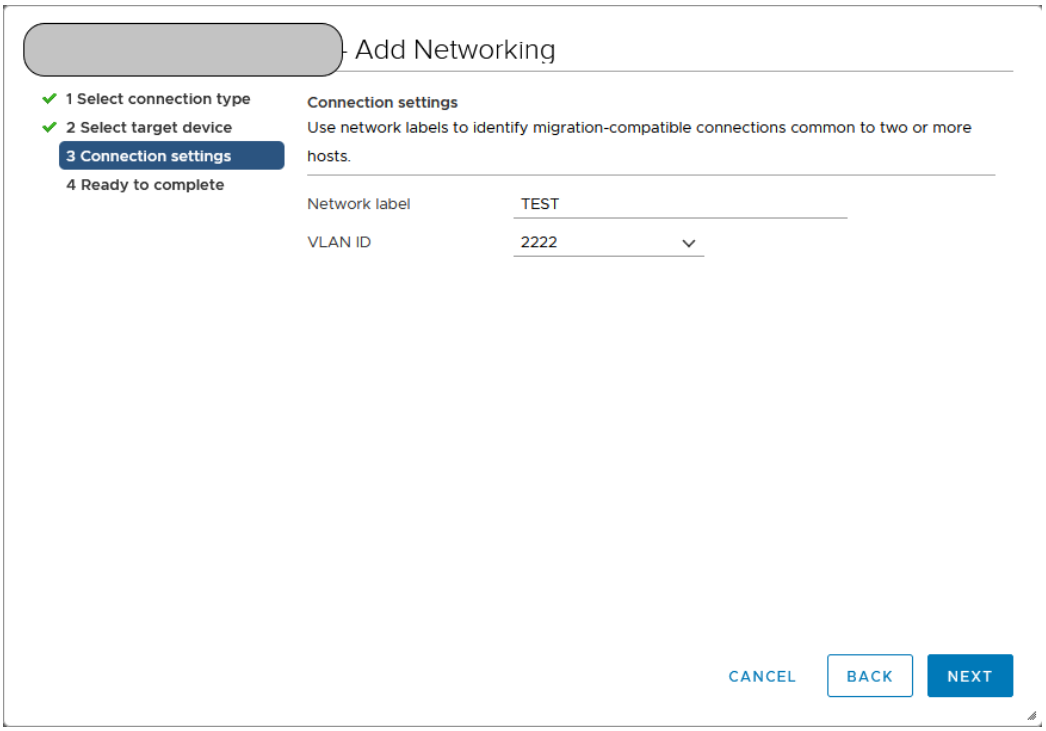

Рисунок 4. Настройки соединения

Далее необходимо завершить настройку, нажав кнопку «Finish» (Рисунок 5).

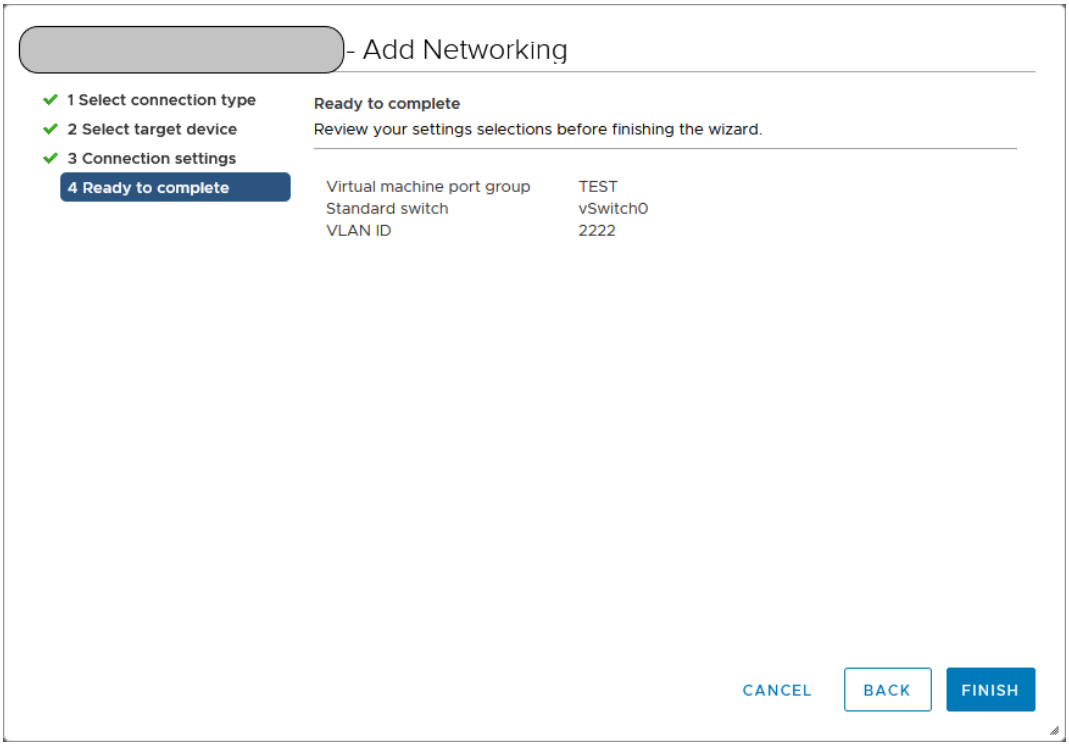

Рисунок 5. Завершение настройки

6.2 Импорт образов

Для импорта образов необходимо в разделе «Host and Clusters» выбрать хост и нажать на него правой кнопкой «мыши» - выбрать «Deploy OVF Template» (Рисунок 6).

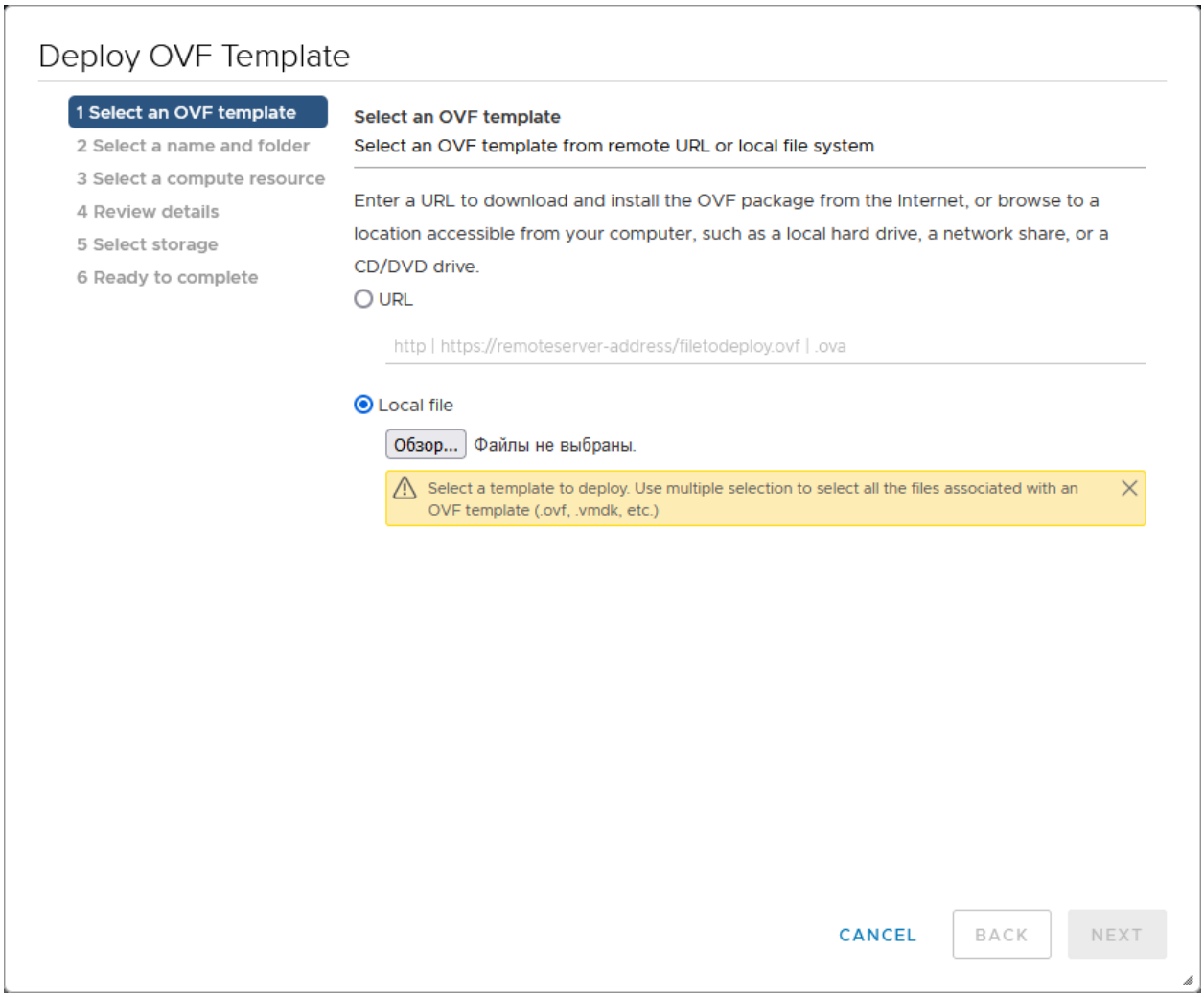

Рисунок 6. Выбор OVF файла

Загрузить файлы требуемого формата и нажать кнопку «Next». Далее необходимо в разделе «Select a name and folder» указать имя и папку и по окончании ввода данных нажать кнопку «Next» (Рисунок 7).

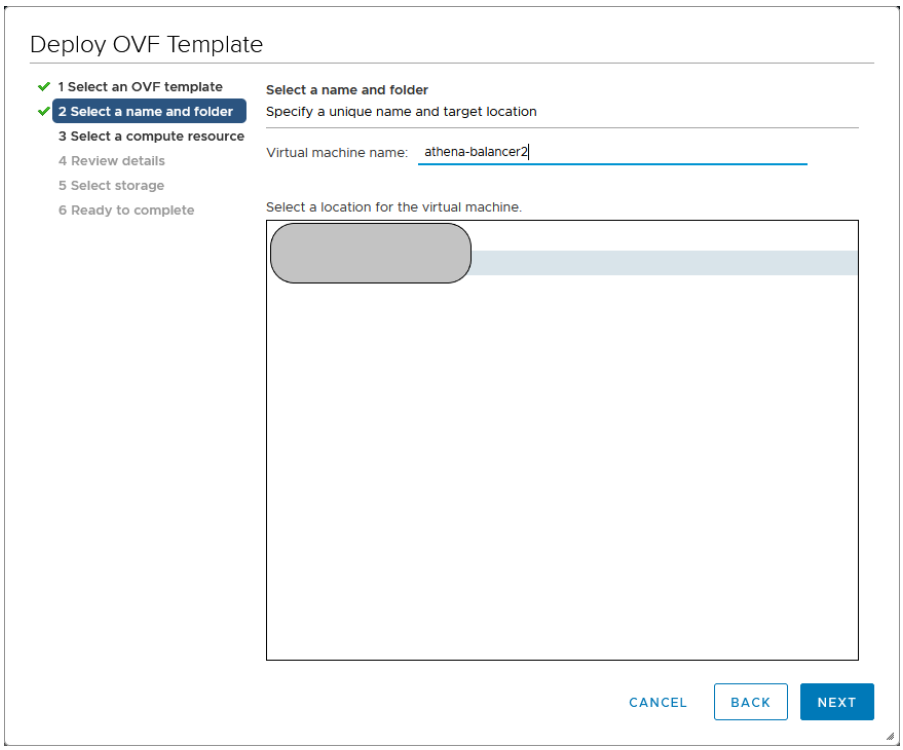

Рисунок 7. Указание имени и папки

Далее необходимо в разделе «Select a computer resource» указать хост и по завершении ввода данных нажать кнопку «Next» (Рисунок 8).

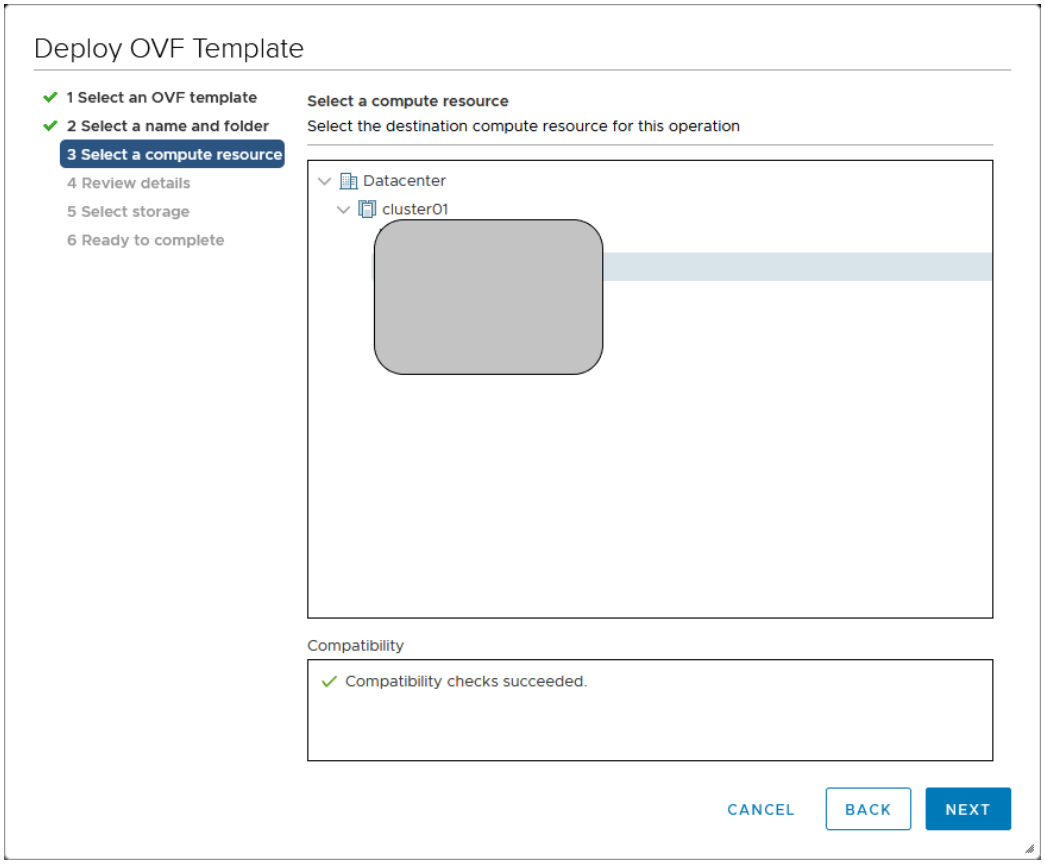

Рисунок 8. Указание хоста

Далее необходимо в разделе «Review details» удостовериться, что все параметры указаны верно и нажать кнопку «Next» (Рисунок 9).

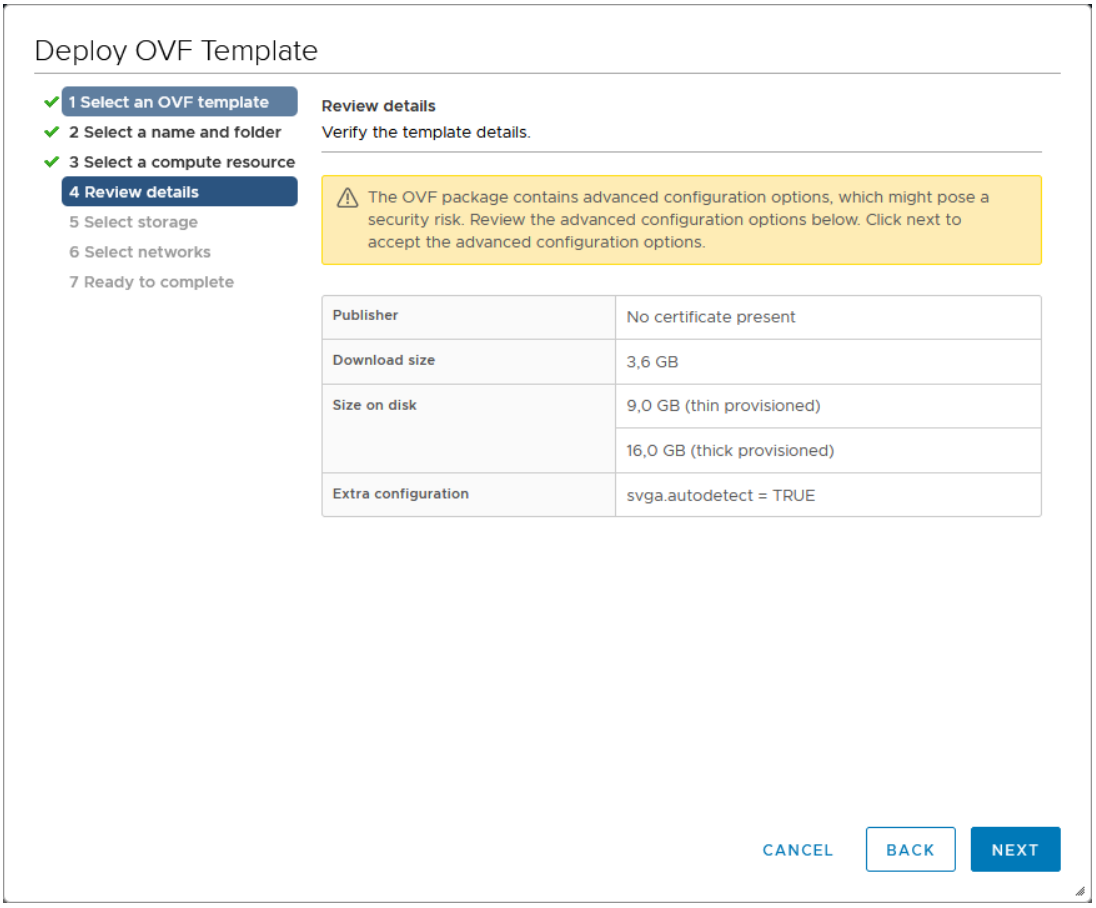

Рисунок 9. Проверка параметров

Далее в разделе «Select storage» необходимо указать хранилище и по завершении ввода данных нажать кнопку «Next» (Рисунок 10).

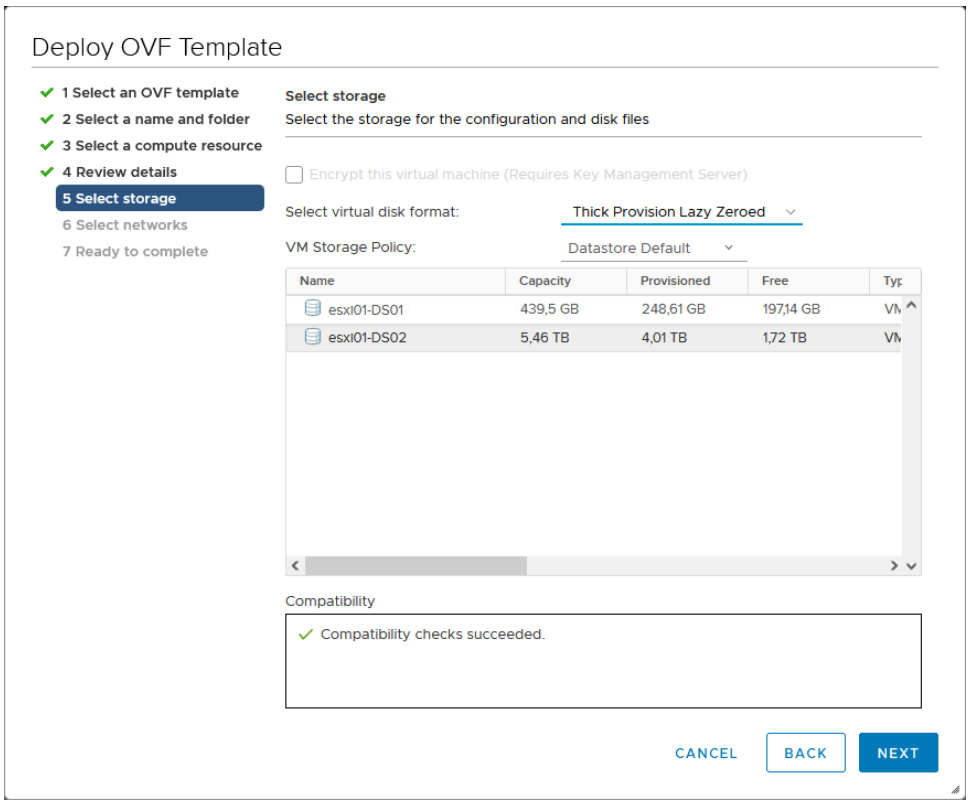

Рисунок 10. Указание хранилища

Далее в разделе «Select networks» необходимо выбрать сеть и по завершении ввода данных нажать кнопку «Next» (Рисунок 11).

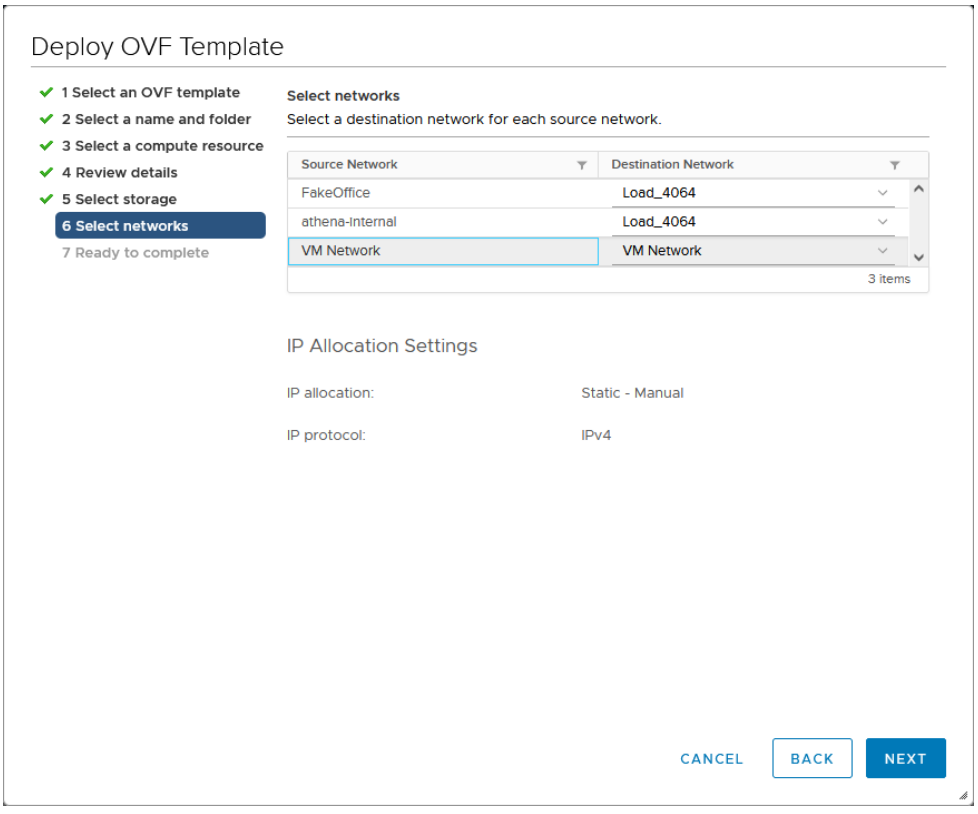

Рисунок 11. Указание сети

Далее в разделе «Ready to complete» необходимо проверить параметры и завершить установку кнопкой «Finish» (Рисунок 12).

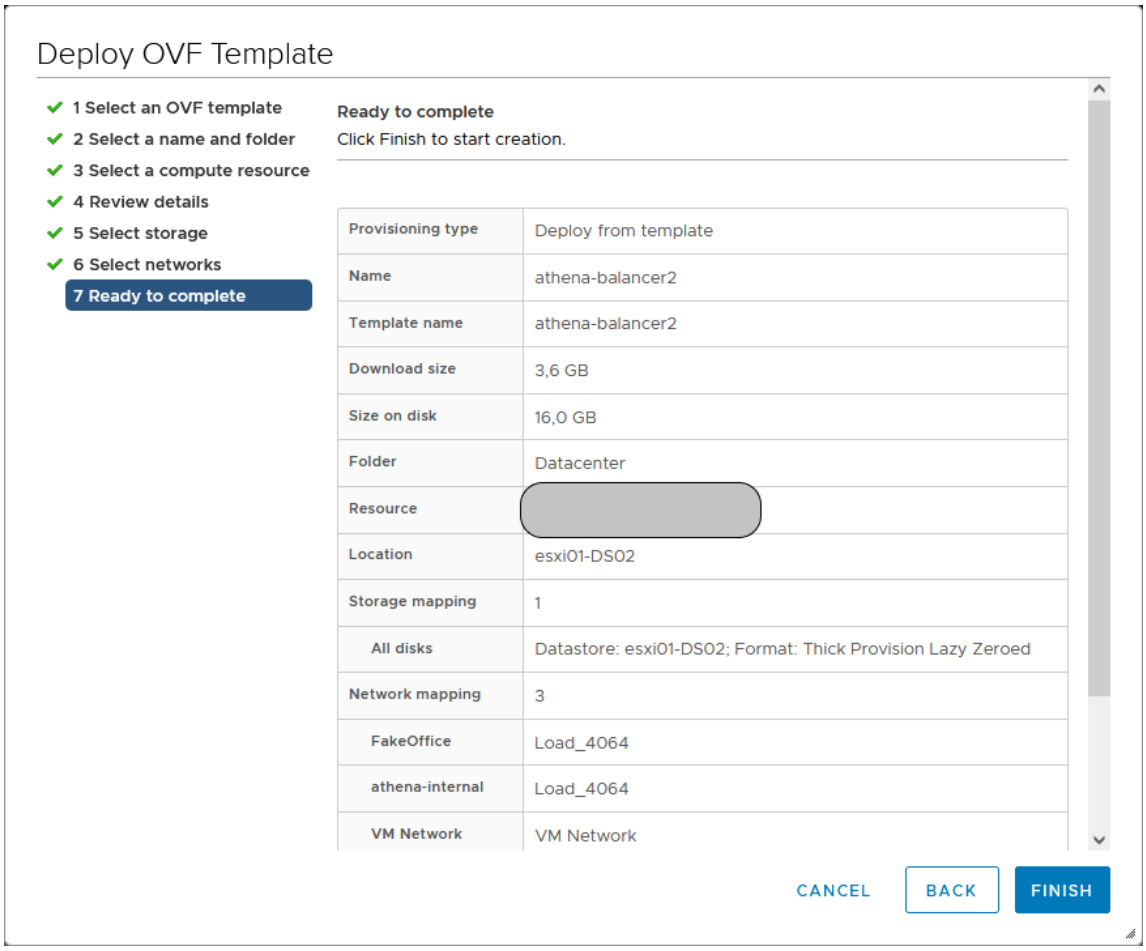

Рисунок 12. Завершение установки