
Astra Linux UBI

сент. 03, 2024

Оглавление

1	Классификация	3
1.1	Дистрибутивы Astra Linux	3
1.2	Окружения	3
1.3	Уровень защищённости ОС	4

UBI (Universal Base Image) Astra Linux - это Универсальные базовые образы, готовые к использованию в различных окружения виртуализации и контейнеризации.

Образы делятся на предназначенные для облачных сервисов, для локальных систем виртуализации и контейнеризации. Образы для облачных сервисов уже загружены в их реестры образов. Образы для локальных систем виртуализации и контейнеризации доступны для загрузки по ссылкам:

Образы виртуальных машин [Загрузить](#)¹

Образы контейнеров [Загрузить](#)²

Боксы Vagrant [Перейти к списку](#)

¹ <https://registry.astralinux.ru/images/>

² <https://registry.astralinux.ru/browse/library>

Классификация

Все образы классифицируются по следующим признакам:

- название дистрибутива Astra Linux, на основе которого создан образ;
- состав пакетов и дополнительные настройки;
- поддержка графического интерфейса пользователя (GUI);
- версия ОС, на основе которой создан образ;
- целевая система виртуализации или облачный сервис;
- уровень защищённости ОС (только для Astra Linux Special Edition 1.7 и Astra Linux Special Edition 1.8).

1.1 Дистрибутивы Astra Linux

При именовании образа для обозначения дистрибутива, на основе которого он собран, используются следующие обозначения:

- smolensk – Astra Linux Special Edition 1.6;
- alse – Astra Linux Special Edition 1.7 и Astra Linux Special Edition 1.8;
- ore1 – Astra Linux Common Edition 2.12.

1.2 Окружения

Поддерживаемые локальные системы виртуализации и контейнеризации:

- Docker / Podman / Kubernetes;
- QEMU;
- Vagrant;
- VirtualBox;
- VMware vSphere.

1.3 Уровень защищённости ОС

Для образов на основе Astra Linux Special Edition 1.7 и Astra Linux Special Edition 1.8 поддерживается дополнительный параметр – уровень защищённости ОС. В названии образа уровень защищённости обозначается следующим образом:

- base – базовый (Орёл);
- adv – усиленный (Воронеж);
- max – максимальный (Смоленск).

Примечание: В базовом уровне защищённости не используется мандатное управление и мандатный контроль целостности.

Подробности см. в статье Справочного центра Astra Linux Special Edition (очередное обновление x.7): [Ключевые изменения в комплексе средств защиты информации](#)³.

1.3.1 Как устроен проект

Основные понятия

Окружение – это конкретная система виртуализации или контейнеризации, для которой предназначен образ.

Версия дистрибутива – версия конкретного дистрибутива Astra Linux.

Уровень защищённости – один из трёх заданных режимов работы подсистемы безопасности, реализованный запатентованными средствами защиты информации в Astra Linux Special Edition 1.7 и Astra Linux Special Edition 1.8.

Версия образа – версия проекта, которая использовалась для создания данного образа.

Версионирование образов

Образы версионированы согласно правилу [семантического версионирования](#)⁴. Версия образа выглядит как:

```
<major_version>.<minor_version>.<micro_version>
```

где:

- <major_version> – номер мажорной версии. Версия 0 используется для начальной разработки. Номер мажорной версии увеличивается, если в образ добавляются изменения, несовместимые с предыдущей мажорной версией.
- <minor_version> – номер минорной версии. Увеличивается при добавлении новой функциональности или существенных изменениях в уже существующей функциональности без нарушения совместимости. Минорная версия обнуляется при увеличении мажорной версии.
- <patch> – номер срочного обновления при исправлении ошибок. Увеличивается после исправления ошибок в имеющейся функциональности без потери обратной совместимости. Новая функциональность не вносится, существенные изменения в уже существующей не производятся. Это поле обнуляется при увеличении мажорной и минорной версий.

³ <https://wiki.astralinux.ru/pages/viewpage.action?pageId=153485983>

⁴ <https://semver.org/>

Именованние образов

Формат именованния образов зависит от того, предназначен ли образ для систем виртуализации или контейнеризации.

Образы для систем виртуализации

Для именованния образов для систем виртуализации используется следующий шаблон:

```
<distro>[-gui][-rc]-<version>-<secure-level>-<env>-<build-version>-<arch>.<ext>
```

Здесь:

- <distro> – название дистрибутива (на данный момент Astra Linux Special Edition);
- -gui – если присутствует, то означает наличие пакетов, поддерживающих графическую оболочку;
- -rc – версия является релиз-кандидатом;
- <version> – версия ОС;
- <secure-level> – уровень защищённости:
 - base – базовый (Орёл);
 - adv – усиленный (Воронеж);
 - max – максимальный (Смоленск);
- <env> – окружение, для которого собран образ;
- <build-version> – версия сборки (она же mg-версия);
- <arch> – архитектура процессора;
- <ext> – расширение файла (qcow2, ova, ovf, raw и т. п.);

Примеры

```
a1se-gui-1.7.4uu1-max-qemu-mg13.0.0-amd64.qcow2
```

где:

- a1se – Astra Linux Special Edition;
- gui – поддержка GUI включена;
- 1.7.4uu1 – версия очередного оперативного обновления – 1.7.4uu1;
- max – максимальный (Смоленск) режим защищённости ОС;
- qemu – образ предназначен для системы виртуализации Qemu/KVM;
- mg13.0.0 – версия образа 13.0.0;
- amd64 – образ предназначен для архитектуры AMD64.

<build-version> может быть latest – назначается на самую новую версию ОС и самую новую версию образа.

```
a1se-1.7-adv-qemu-latest-amd64.qcow2
```

где:

- a1se – Astra Linux Special Edition;
- 1.7 – последняя версия Astra Linux Special Edition 1.7;
- adv – усиленный (Воронеж) режим защищённости ОС;

- `qemu` – образ предназначен для системы виртуализации Qemu/KVM;
- `latest` – самая последняя версия образа;
- `amd64` – образ предназначен для архитектуры AMD64.

Образы для систем контейнеризации

Для именования образов для систем контейнеризации используется следующий шаблон:

```
astra/ubi<major-version>[-dev]:<tag>
```

Здесь:

- `<major version>` – мажорная версия ОС без точки (17 или 18);
- `dev` – если присутствует, то означает среду выполнения для определённого языка программирования;
- `<tag>` – тег контейнера (`latest`, версия ОС).

Примеры

```
astra/ubi18:1.8.0
```

где:

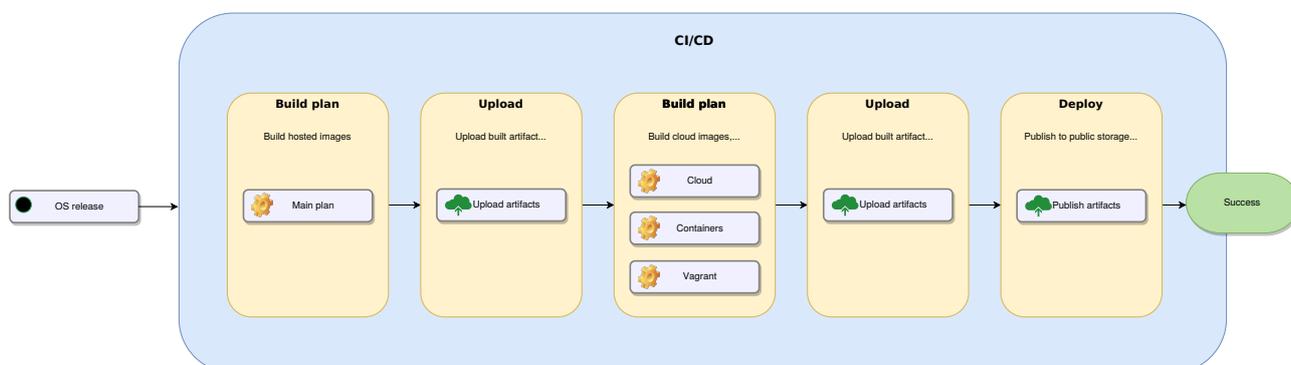
- 18 – Astra Linux Special Edition 1.8;
- 1.8.0 – версия Astra Linux Special Edition 1.8.0 без установленных оперативных обновлений.

```
astra/ubi17-openjdk:latest
```

где:

- 17 – мажорная версия Astra Linux Special Edition 1.7;
- `openjdk` – в образе установлен OpenJDK;
- `latest` – самая новая версия образа.

Схема сборки



1.3.2 Описания образов

Универсальный базовый образ – это образ диска с установленной Astra Linux в минимальной конфигурации. Включает в себя минимальный набор ПО, необходимый для запуска и работы образа в соответствующем окружении. Ниже представлены параметры установки образов.

Параметры установки

Включается интерфейс eth0 с настройкой получения IP-адреса по DHCP.

Параметры, используемые во время установки:

- Переключение раскладки – Alt+Shift.
- Имя компьютера – alse.
- Имя пользователя и пароли зависят от целевого окружения.

Важно: Логин/пароль во всех образах для локальных систем виртуализации, кроме Vagrant: astra / astra.

Логин/пароль в Vagrant боксах: vagrant / vagrant.

Логин и пароль для GRUB: astra / 12345678.

Для Cloud-образов логин и пароль определяется на этапе создания VM через механизмы cloud-init. Аутентификация производится по ключам SSH.

Для образов ПК СВ «Брест» логин и пароль определяется на этапе создания VM через механизмы one-context.

Для образов VMware Cloud Director логин пользователя: astra, пароль задаётся на этапе создания VM и её настройки.

- Часовой пояс – Europe/Moscow (UTC+3).
- Разметка дисков:
 - весь диск используется под корневую файловую систему;
 - раздел подкачки не создаётся.
- Выбор программного обеспечения — включен флаг Средства удаленного доступа SSH. Для образов с GUI дополнительно включен флаг Рабочий стол FLY.
- Дополнительные настройки ОС – по умолчанию (все флаги выключены).
- Во все образы добавлен файл /etc/mg_version с информацией о версии релиза и коммита, использованного для сборки.
- Установлен пакет dialog для корректной работы APT и других утилит.
- В файл /etc/hosts добавлена строка 127.0.0.1 alse.
- Размеры дисков для каждого окружения отличаются (см. в описании окружений).

После установки системы

- Образ очищается от временных файлов.
- Удаляется `machine-id`.
- Свободное место заполняется нулями (для лучшей компрессии).
- В образах для облачных систем виртуализации удаляются ключи SSH.

Образы для облачной виртуализации

ПК СВ «Брест»

Образы предназначены для работы в системе виртуализации ПК СВ «Брест».

Уровни защищённости ОС:

- базовый (Орёл);
- усиленный (Воронеж);
- максимальный (Смоленск).

Конфигурация образа

- Размер системного диска VM:
 - с GUI – 10 ГБ;
 - без GUI – 3 ГБ.

Размер диска можно изменить при создании VM.

- Дополнительно установлен пакет `one-context`.

Примечание: В образе отсутствует пользователь по умолчанию. Настройка пользователя производится через механизм `one-context`, инструкция доступна по [ссылке](#)⁵.

VMmanager

Образы предназначены для работы в системе виртуализации VMmanager.

Уровни защищённости ОС:

- базовый (Орёл);
- усиленный (Воронеж);
- максимальный (Смоленск).

⁵ <https://wiki.astralinux.ru/brest/latest/rasprostranenie-servisnoj-uchetnoj-zapisi-pol-zovatelya-na-gostevye-os-302023408.html>

Конфигурация образа

- Размер системного диска VM:
 - без GUI - 3 ГБ;
 - с GUI - 10 ГБ.

Размер диска можно изменить при создании VM.

- Дополнительно установлен пакет `qemu-guest-agent`.

Важно: По умолчанию в VM создается пользователь `root`. Использование учетной записи `root` для администрирования системы не рекомендуется.

Инструкция по использованию

Для создания учетной записи администратора выполните следующие действия:

1. Создайте скрипт для VM. Инструкция по созданию скрипта для VM доступна на [сайте с документацией](#)⁶ VMmanager .

Содержимое скрипта

```
#!/bin/bash

# Generate password hash for user
salt=$(echo $RANDOM | md5sum | head -c 20)
pass=$(openssl passwd -6 -salt "$salt" "$PASS")
root_ssh_keys=/root/.ssh/authorized_keys

# If username not provided default astra account will be used
if [ -n "${ASTRA_USERNAME}" ]; then
    ADMIN_USERNAME="${ASTRA_USERNAME}"
else
    ADMIN_USERNAME="astra"
fi

# Validate ssh key passed to script. If ssh key not provided root key will be used.
if [ -n "${SSH_PUBLIC_KEY}" ] && ssh-keygen -l -f - <<< "${SSH_PUBLIC_KEY}" > /dev/
  ↪null 2>&1; then
    SSH_AUTH_KEY="${SSH_PUBLIC_KEY}"
else
    if test -f "$root_ssh_keys"; then
        SSH_AUTH_KEY="$(cat /root/.ssh/authorized_keys)"
    else
        SSH_AUTH_KEY=""
    fi
fi

useradd -m -s /bin/bash -p "${pass}" "${ADMIN_USERNAME}"
USERHOME=$(getent passwd "${ADMIN_USERNAME}" | cut -d: -f6)
usermod -a -G astra-admin "${ADMIN_USERNAME}"

if [ -n "${SSH_AUTH_KEY}" ]; then
    if ! test -d "${USERHOME}/.ssh"; then
        mkdir "${USERHOME}/.ssh"
    fi
    chmod 0700 "${USERHOME}/.ssh"
    chown "${ADMIN_USERNAME}":"${ADMIN_USERNAME}" "${USERHOME}/.ssh"
```

(continues on next page)

⁶ <https://docs.ispsystem.ru/vmmanager-admin/skripty/sozdanie-skripta-dlya-vm>

(продолжение с предыдущей страницы)

```

echo "${SSH_AUTH_KEY}" >> "${USERHOME}/.ssh/authorized_keys"
chmod 0600 "${USERHOME}/.ssh/authorized_keys"
chown "${ADMIN_USERNAME}":"${ADMIN_USERNAME}" "${USERHOME}/.ssh/authorized_keys"
fi

ASTRA_MODE=$(astra-modeswitch get)
if test $ASTRA_MODE != "0"; then
  /usr/sbin/pdpl-user -i 63 "${ADMIN_USERNAME}"
  echo "$?"
fi

sed -i -e 's/#?\(PermitRootLogin\s*\).*$/\1 no/' /etc/ssh/sshd_config
systemctl restart sshd
passwd -l root

```

Скрипт использует следующие параметры:

- ASTRA_USERNAME – название учетной записи администратора.
Значение по умолчанию – astra.
- SSH_PUBLIC_KEY – публичный ключ SSH для учетной записи администратора.

Этот сценарий выполняет следующие действия в ОС ВМ:

- Создание пользователя-администратора с именем, указанным в параметре ASTRA_USERNAME.
 - Установка пароля, заданного при создании ВМ.
 - Добавление публичного ключа SSH, указанного в параметре SSH_PUBLIC_KEY, в файл ~/.ssh/authorized_keys.
 - Блокировка аутентификации пользователя root.
2. При создании ВМ в пункте **Приложения и скрипты** выберите ранее созданный скрипт. Инструкция по созданию ВМ доступна на [сайте с документацией⁷ VMmanager](#).

VMware Cloud Director

Образы предназначены для работы в системе виртуализации VMware Cloud Director.

Уровни защищённости ОС:

- базовый (Орёл);
- усиленный (Воронеж).

Конфигурация образа

- Размер системного диска ВМ – 16 ГБ.
Размер диска можно изменить при создании ВМ.
- Установлены дополнительные пакеты:
 - cloud-init;
 - open-vm-tools.

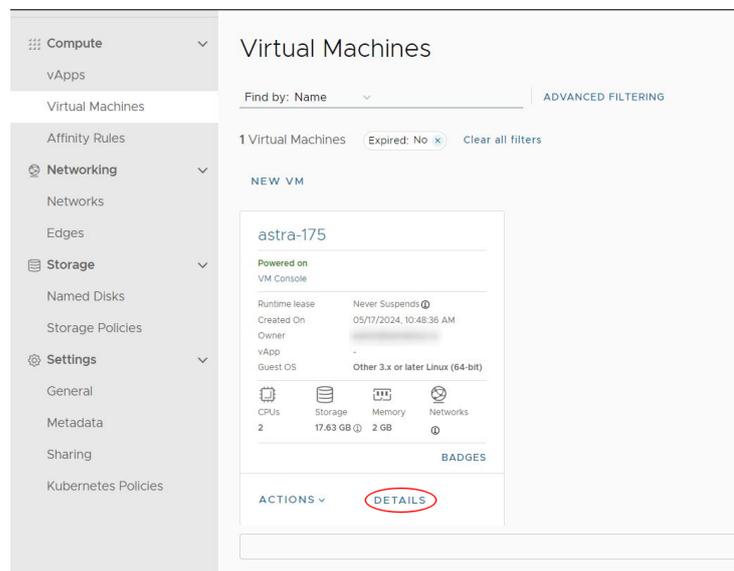
⁷ <https://docs.ispsystem.ru/vmmanager-admin/virtual-nye-mashiny/sozдание-virtual-nyh-mashin>

Инструкции по использованию

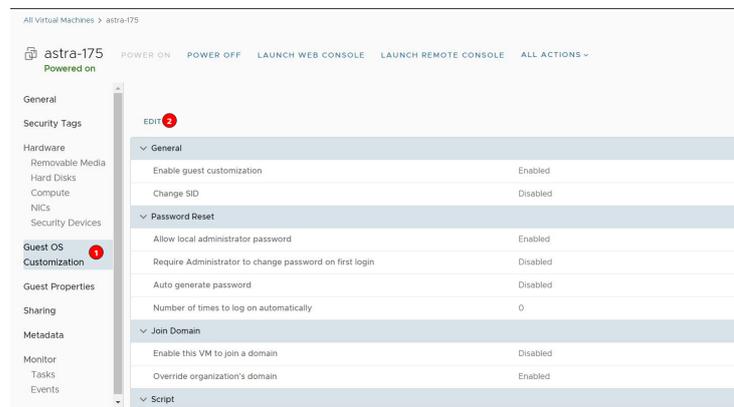
Предупреждение: Перед первым запуском VM необходимо задать пароль пользователя ОС, создаваемого по умолчанию!

Для создания и настройки VM выполните следующие действия в web-интерфейсе VMware Cloud Director:

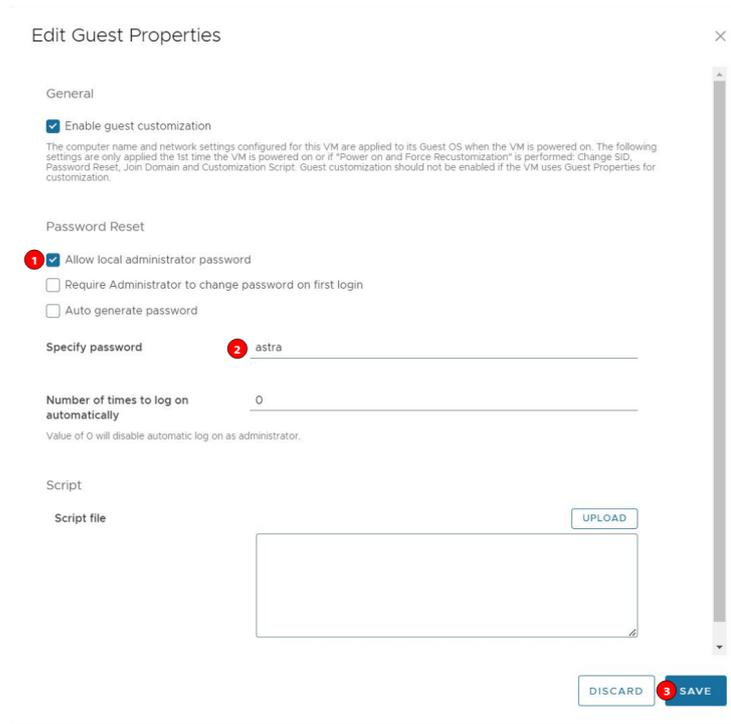
1. В разделе **Libraries** добавьте шаблон VM.
2. Для создания VM в нужном пуле нажмите кнопку *New VM*.
3. Заполните обязательные поля.
4. Выберите созданный ранее шаблон.
5. Нажмите кнопку *OK*.
6. В окне настройки VM нажмите кнопку *Details*.



7. Выберите вкладку **Guest OS Customization**.
8. Нажмите кнопку *EDIT*.

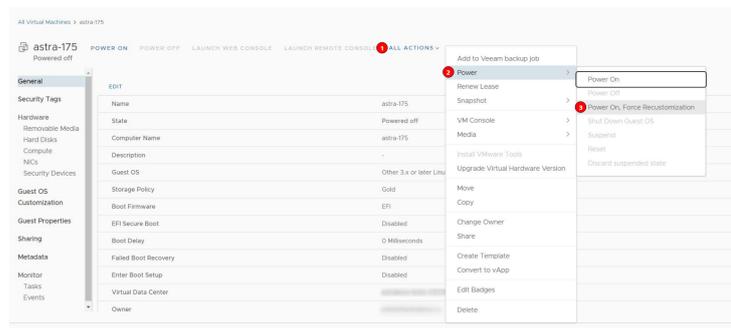


9. В открывшемся окне выключите флаг **Auto generate password**.
10. В поле **Specify password** введите пароль пользователя astra, по умолчанию создаваемого в VM.
11. Нажмите кнопку *Save*:



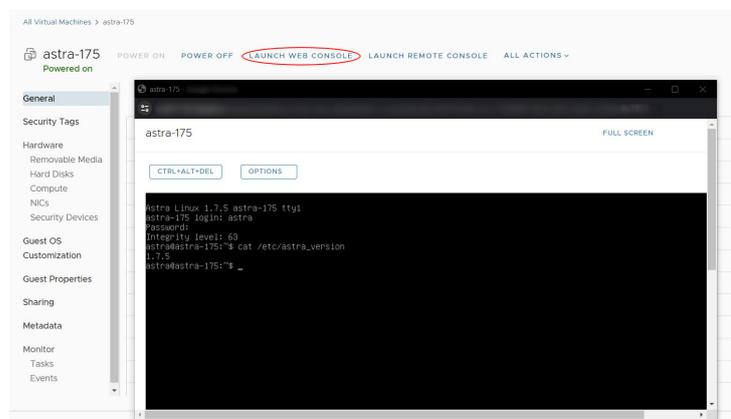
12. Нажмите кнопку *All Actions* и в открывшемся меню выберите *Power* ▶ *Power on, Force Recustomization*.

13. Дождитесь окончания загрузки ВМ.



14. Откройте консоль любым удобным способом, например, нажав кнопку *LAUNCH WEB CONSOLE*. Также доступно подключение по SSH.

15. В окне приглашения входа в ОС введите логин *astra* и заданный ранее пароль.



Образы для локальной виртуализации

Контейнеры

Поддерживаются образы следующих типов:

- **Standard** – базовая ОС Astra Linux Special Edition и стандартные утилиты из базовой системы.
- **Multi-service (init)** – базовая ОС Astra Linux Special Edition с системой инициализации systemd.
- **Dev** – базовая ОС Astra Linux Special Edition и окружение для одного из языков программирования.

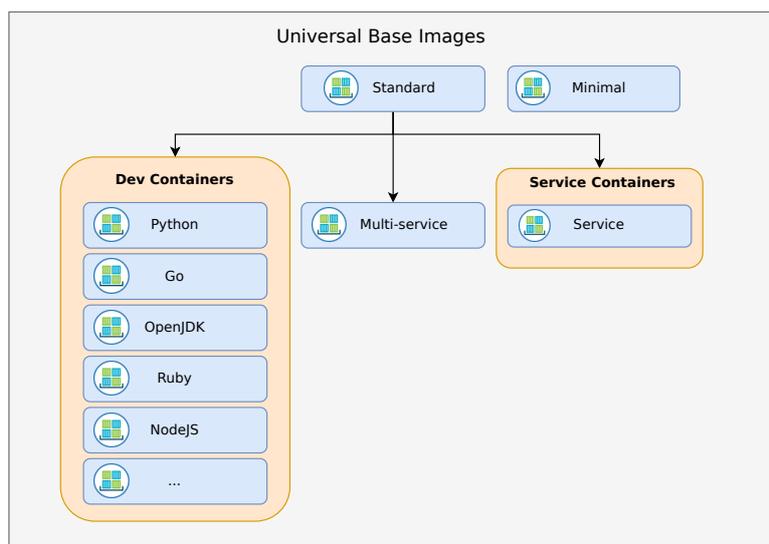
Поддерживаются следующие языки программирования (в примере указаны версии из базовых репозиториев):

Programmir language	ALSE 1.7.0	ALSE 1.7.1	ALSE 1.7.2	ALSE 1.7.3	ALSE 1.7.4	ALSE 1.7.5	ALSE 1.8.0
Golang	2:1.15~1					2:1.19~	2:1.21~2.astra1
NodeJS	10.24.0~dfsc-1~deb10u1	10.24.0~dfsg-1~deb10u1+ci202112021504+astra1			10.24.0~dfsg-1~deb10u3		18.13.0+dfsg1-1
OpenJDK	11.0.9.1+1-1~deb10u2	11.0.12+7-2~deb10u1	11.0.15+10-1~deb10u1	11.0.16+8-1~deb10u1	11.0.18+10-1~deb10u1+b1		17.0.10+7-1~deb12u1
Python	3.7.3-1						3.11.2-1
Ruby	1.2.5-1						1:3.1

Базовый образ Standard собирается с помощью [Packer](https://www.packer.io/)⁸ в один слой.

Все остальные образы собираются через заранее написанные Dockerfile на базе образа Standard.

Примечание: Образ Minimal находится в разработке.



⁸ <https://www.packer.io/>

Тегирование образов

Для образов определены теги вида:

- latest – назначается на самую новую версию ОС и самый новый релиз. Например, если тег не указан при выполнении команды `podman pull`, будет загружена версия с тегом latest.
- X.Y – назначается на самую новую мажорную версию ОС и самый новый релиз. Он соответствует тегу latest, если для ОС существует только одна мажорная версия. Если ОС имеет несколько мажорных версий, то тегов будет по количеству этих версий, и каждый из них будет указывать на самый новый релиз для этой версии ОС.
- X.Y.Z – назначается на самую новую версию ОС и самый новый релиз. Тег перемещается с очередным релизом.
- X.Y.Z-mgA.B.C – назначается на определенные версии ОС и релиз. Назначается единожды и всегда указывает на конкретный образ.

Примеры работы с образами UBI

Скачивание образа `astralinux.ru/library/astra/ubi17:1.7.3`:

```
podman pull registry.astralinux.ru/library/astra/ubi17:1.7.3
```

Запуск контейнера с `astralinux.ru/library/astra/ubi17:1.7.5` в фоновом режиме, с прикрепленным томом:

```
podman run -d \  
  --name else_volume \  
  --mount type=bind,source="$(pwd)"/data,target=/usr/data \  
  registry.astralinux.ru/library/astra/ubi17:1.7.5 \  
  /bin/sh -c 'while true; do sleep 10; done'
```

Примечание: Каталог `data/` должен существовать.

Запуск контейнера `astralinux.ru/library/astra/ubi18:latest` в интерактивном режиме. При завершении работы с оболочкой `Bash` контейнер автоматически удаляется:

```
podman run -it --rm \  
  --name else_mg \  
  registry.astralinux.ru/library/astra/ubi18:latest \  
  /bin/bash
```

Запуск контейнера `astralinux.ru/library/astra/ubi17:1.7.5-mg13.1.2` в фоновом режиме:

```
podman run -d \  
  --name else_mg \  
  registry.astralinux.ru/library/astra/ubi17:1.7.5-mg13.1.2 \  
  /bin/sh -c 'while true; do sleep 10; done'
```

QEMU

Образы предназначены для работы в системе виртуализации QEMU/KVM (libvirt).

Уровни защищённости ОС:

- базовый (Орёл);
- усиленный (Воронеж);
- максимальный (Смоленск).

Конфигурация образа

- Размер системного диска VM: 30 ГБ;
- Дополнительно установлен пакет `qemu-guest-agent`.

Примечание: Логин и пароль по умолчанию: `astra/astra`.

Инструкции по использованию

Для настройки libvirt можно воспользоваться данной [инструкцией](#)⁹.

Все образы libvirt доступны для загрузки по [ссылке](#)¹⁰.

В примере используется образ Astra Linux Special Edition 1.7.5uu1 в режиме базовый (Орёл).

Command line interface

1. Установите пакет `virt-viewer`.

```
sudo apt update && sudo apt install virt-viewer --yes
```

2. Для создания VM выполните следующую команду:

```
virt-install \
  --connect qemu:///system \
  --name alse-175 \
  --memory 2048 \
  --vcpus 2 \
  --disk alse-1.7.5uu1-base-qemu-mg13.1.1-amd64.qcow2 \
  --import \
  --os-variant debian11
```

Данная команда создаст виртуальную машину со следующими параметрами:

- название – `alse-175`;
- количество vCPU – 2;
- объём RAM, ГБ – 2;
- образ – `alse-1.7.5uu1-base-qemu-mg13.1.1-amd64.qcow2`.

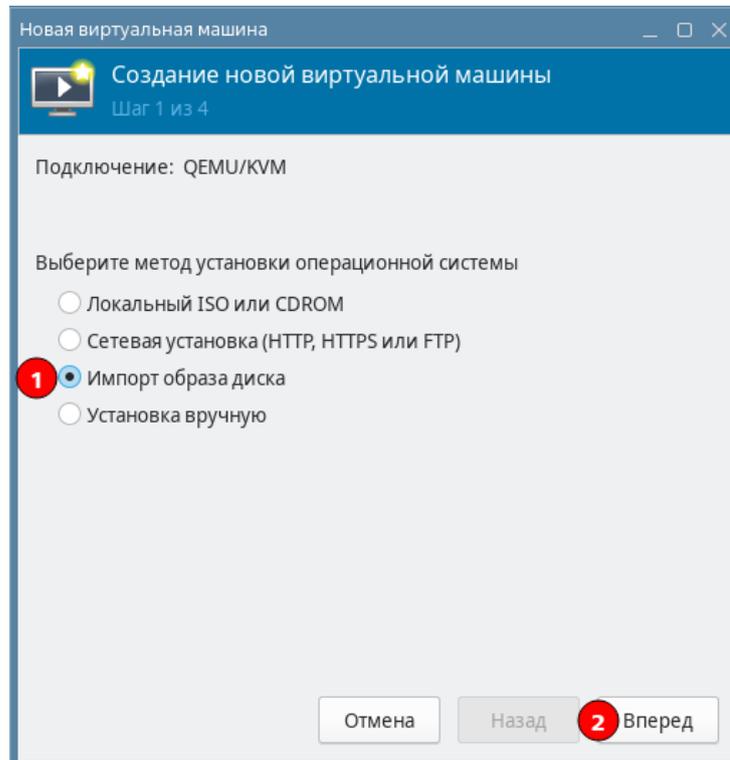
⁹ <https://wiki.astralinux.ru/x/cQIy>

¹⁰ <https://registry.astralinux.ru/images/alse/qemu/>

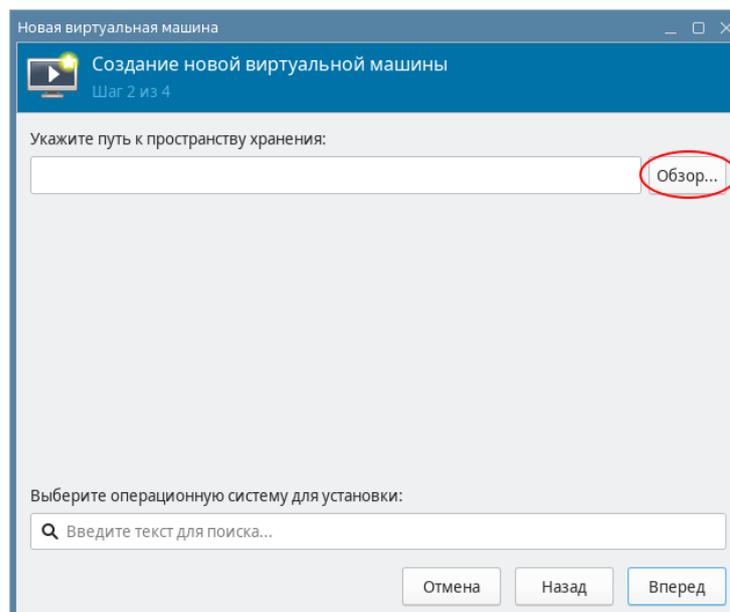
Virt-Manager

Чтобы с помощью Virt-Manager развернуть VM из образа, выполните следующие действия:

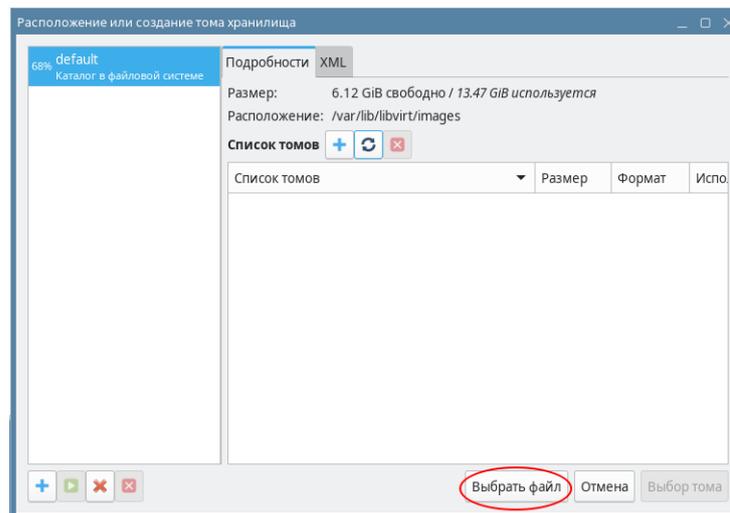
1. Загрузите образ и сохраните, например, в каталог `~/images`.
2. Запустите Virt-Manager: *Системные* ▶ *Менеджер виртуальных машин*.
3. Выберите в главном меню *Файл* ▶ *Создать виртуальную машину*.
4. Выберите метод установки операционной системы **Импорт образа диска** и нажмите кнопку *Вперед*.



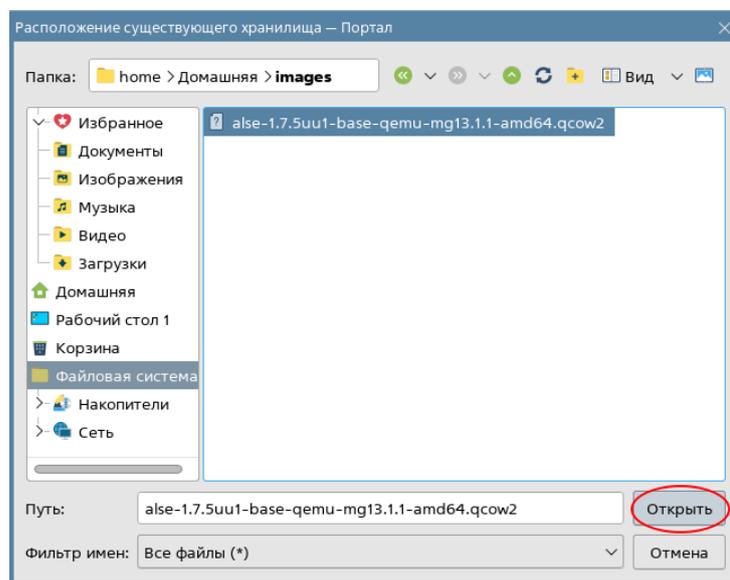
5. Нажмите кнопку *Обзор*.



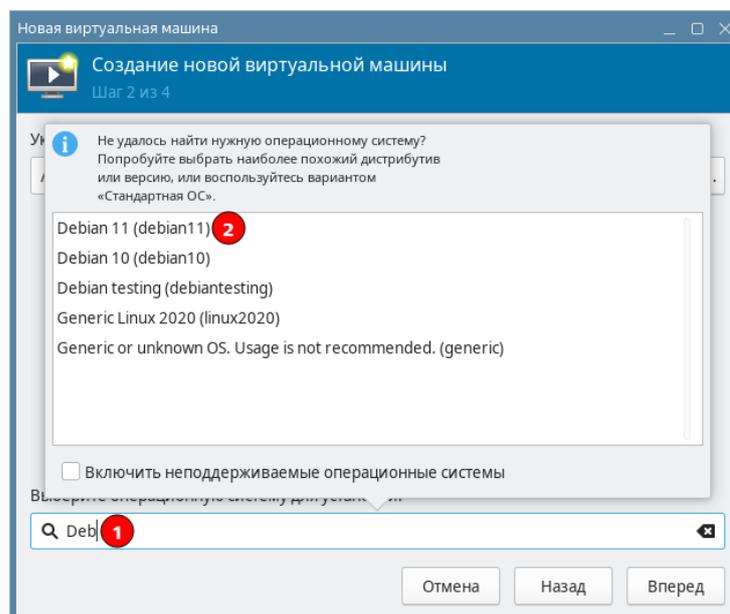
6. В открывшемся окне нажмите кнопку *Выбрать файл*.



7. Выберите файл образа и нажмите кнопку *Открыть*.

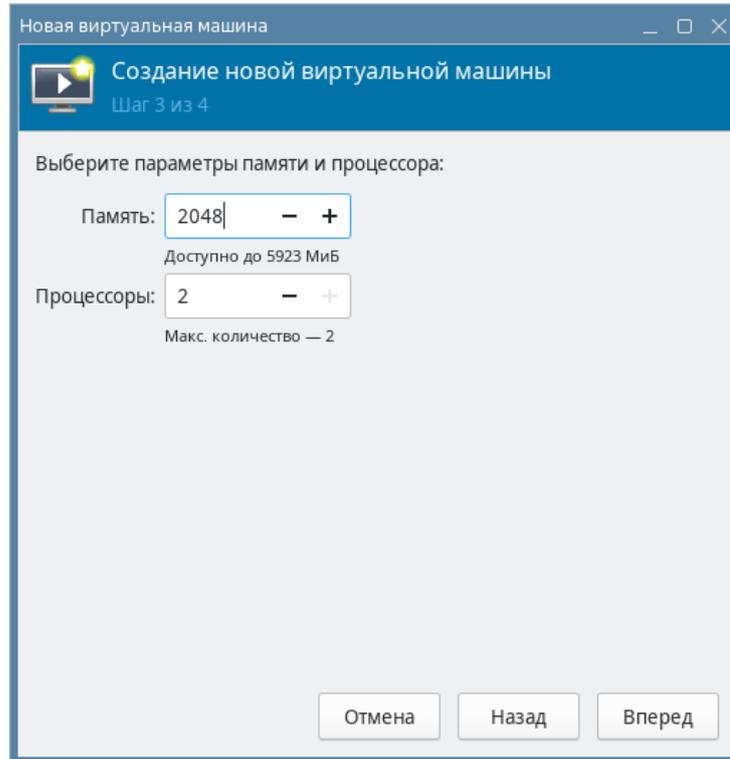


8. В поле **Выберите операционную систему для установки** введите «Deb» и выберите **Debian 11 (debian11)** из списка.



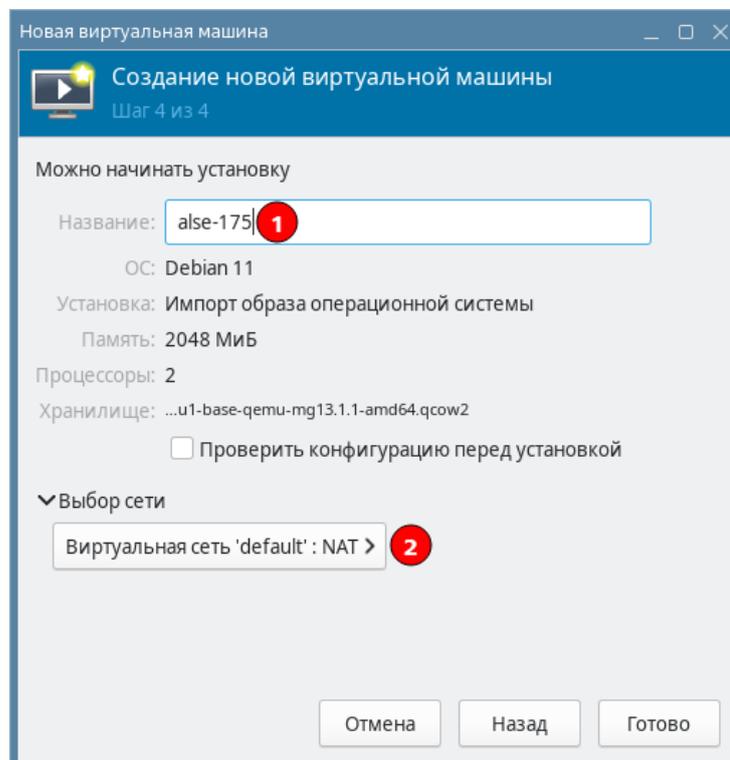
9. Нажмите кнопку *Вперед*.

10. Укажите нужное количество vCPU и RAM.



11. Нажмите кнопку *Вперед*.

12. Укажите название виртуальной машины и выберите сеть **default**.



13. Нажмите кнопку *Готово*.

Созданная VM будет запущена автоматически.

Vagrant

Боксы Vagrant предназначены только для использования совместно с VirtualBox.

Уровни защищённости ОС:

- базовый (Орёл);
- усиленный (Воронеж);
- максимальный (Смоленск).

Конфигурация бокса

- Размер системного диска VM – 30 ГБ.
- Установлен пакет расширений Virtual Box Guest Additions версии 6.1.38.

Примечание: Логин и пароль по умолчанию: vagrant/vagrant.

Инструкции по использованию

Инструкция по установке доступна в разделе [Vagrant](#).

VM на локальной машине

Для создания VM с установленной операционной системой Astra Linux выполните следующие действия:

1. Создайте пару ключей SSH, используемых для подключения к VM:

```
ssh-keygen -C "<comment>" -f ~/.ssh/deployment -N ""
```

В результате выполнения команды в каталоге ~/.ssh/ создаются два файла, содержащие приватный (deployment) и публичный (deployment.pub) ключи SSH соответственно.

2. Создайте файл Vagrantfile со следующим содержимым:

```
# frozen_string_literal: true

Vagrant.configure('2') do |config|
  config.vm.box = 'alse-vanilla-base/1.7.5uul'

  # Ссылка на бокс
  config.vm.box_url = 'https://registry.astralinux.ru/vagrant/alse-vanilla-base%2F1.7.5uul'

  config.vm.provision 'file',
    source: '~/.ssh/deployment.pub',
    destination: '/home/vagrant/.ssh/id_rsa.pub'

  config.vm.provision 'shell',
    inline: 'cat /home/vagrant/.ssh/id_rsa.pub >> /home/vagrant/.ssh/authorized_keys'

  # Настройка параметров VM
  config.vm.define 'host01' do |node|
    node.vm.hostname = 'host01' # Короткое имя хоста

    # Настройки, специфичные для VirtualBox
```

(continues on next page)

(продолжение с предыдущей страницы)

```

node.vm.provider 'virtualbox' do |vb|
  vb.cpus = 4 # Кол-во ядер CPU
  vb.memory = 4096 # Объем RAM, МБ
end

# Параметры сети
node.vm.network 'private_network',
  ip: '192.168.56.11', # Статический IP-адрес
  netmask: '255.255.255.0', # Маска подсети
  dhcp_enabled: false # Запрет использования DHCP
end
end

```

Название бокса Vagrant задаётся в строке:

```
config.vm.box = 'alse-vanilla-base/1.7.5uu1'
```

Ссылка на бокс задаётся в строке:

```
config.vm.box_url = 'https://registry.astralinux.ru/vagrant/alse-vanilla-base%2F1.7.5uu1'
```

Чтобы получить доступ к VM по SSH, необходимо в подкаталоге `~/ .ssh/` нужного пользователя VM разместить публичный ключ SSH, а также добавить его в список доверенных ключей:

```

config.vm.provision 'file',
  source: '~/ .ssh/deployment.pub',
  destination: '/home/vagrant/.ssh/id_rsa.pub'

config.vm.provision 'shell',
  inline: 'cat /home/vagrant/.ssh/id_rsa.pub >> /home/vagrant/.ssh/authorized_keys'

```

Короткое имя хоста (vm) задаётся в параметре `node.vm.hostname`:

```
node.vm.hostname = 'vm' # Короткое имя хоста
```

В этих строках задаётся количество доступных VM ядер CPU и МБ RAM:

```

vb.cpus = 4 # Кол-во ядер CPU
vb.memory = 4096 # Объем RAM, МБ

```

Настройки подключения VM к сети:

```

node.vm.network 'private_network',
  ip: '192.168.56.11', # Статический IP-адрес
  netmask: '255.255.255.0', # Маска подсети
  dhcp_enabled: false # Запрет использования DHCP

```

- Для проверки корректности синтаксиса созданного Vagrantfile выполните команду:

```
vagrant validate
```

При отсутствии ошибок в терминал выводится строка:

```
Vagrantfile validated successfully.
```

В случае наличия ошибок исправьте их и выполните повторную проверку синтаксиса Vagrantfile.

- Для создания и запуска VM выполните команду:

```
vagrant up
```

Дождитесь завершения выполнения команды, это может занять некоторое время.

5. Проверьте состояние VM:

```
vagrant status
```

Она должна быть активна (состояние running):

```
Current machine states:
```

```
vm                running (virtualbox)
```

```
The VM is running. To stop this VM, you can run `vagrant halt` to
shut it down forcefully, or you can run `vagrant suspend` to simply
suspend the virtual machine. In either case, to restart it again,
simply run `vagrant up`.
```

Если в процессе создания или запуска VM будут выведены сообщения об ошибках, выполните следующие действия:

1. Проверьте содержимое Vagrantfile на наличие ошибок в параметрах настройки VM.
2. Удалите существующую VM и создайте её заново:

```
vagrant destroy --force && vagrant up
```

Инфраструктура из нескольких VM

Следующий пример файла Vagrantfile демонстрирует способ создания инфраструктуры из двух VM:

Содержимое файла Vagrantfile

```
# frozen_string_literal: true

boxes = {
  'dc01' => {
    box: 'alse-vanilla-base/1.7.5uu1',
    box_url: 'https://registry.astralinux.ru/vagrant/alse-vanilla-base%2F1.7.5uu1',
    cpu_cores: '2',
    memory: '2048',
    ipv4: '192.168.56.10'
  },
  'dc02' => {
    box: 'alse-vanilla-base/1.7.5uu1',
    box_url: 'https://registry.astralinux.ru/vagrant/alse-vanilla-base%2F1.7.5uu1',
    cpu_cores: '2',
    memory: '2048',
    ipv4: '192.168.56.11'
  }
}

Vagrant.configure(2) do |config|
  boxes.each do |hostname, cfg|
    config.vm.define hostname do |host|
      host.vm.boot_timeout = 600
      host.vm.hostname = hostname
      host.vm.box = cfg[:box]
      host.vm.box_url = cfg[:box_url]
    end
  end
end
```

(continues on next page)

(продолжение с предыдущей страницы)

```
host.vm.provider 'virtualbox' do |v|
  v.cpus = cfg[:cpu_cores]
  v.memory = cfg[:memory]
end
host.vm.network 'private_network', ip: cfg[:ipv4]
end
end
end
```

Настройка состоит из двух частей:

- Первая часть `boxes` задаёт параметры двух VM.
- Вторая часть описывает создание этих VM с использованием параметров, заданных в структуре `boxes`.

Удаление VM

Для удаления неиспользуемых VM выполните следующие действия:

1. Определите список существующих VM:

```
vagrant global-status
```

2. Удалите ненужные VM с помощью команды:

```
vagrant destroy <VM>
```

где `<VM>` - названием удаляемой VM.

Удаление боксов

Для удаления неиспользуемых боксов выполните следующие действия:

1. Получите список боксов, загруженных на компьютер:

```
vagrant box list
```

2. Удалите ненужные боксы:

```
vagrant box remove <box>
```

где `<box>` - название бокса

Таблица боксов Vagrant

Astra Linux Special Edition 1.8

ОС	Версия ОС	Название бокса	URL
Astra Linux Special Edition базовый (Орёл)	1.8.1	alse-base/1.8.1	https://registry.astralinux.ru/vagrant/alse-base%2F1.8.1
Astra Linux Special Edition усиленный (Воронеж)	1.8.1	alse-adv/1.8.1	https://registry.astralinux.ru/vagrant/alse-adv%2F1.8.1
Astra Linux Special Edition максимальный (Смоленск)	1.8.1	alse-max/1.8.1	https://registry.astralinux.ru/vagrant/alse-max%2F1.8.1
Astra Linux Special Edition базовый (Орёл) + GUI	1.8.1	alse-gui-base/1.8.1	https://registry.astralinux.ru/vagrant/alse-gui-base%2F1.8.1
Astra Linux Special Edition усиленный (Воронеж) + GUI	1.8.1	alse-gui-adv/1.8.1	https://registry.astralinux.ru/vagrant/alse-gui-adv%2F1.8.1
Astra Linux Special Edition максимальный (Смоленск) + GUI	1.8.1	alse-gui-max/1.8.1	https://registry.astralinux.ru/vagrant/alse-gui-max%2F1.8.1

Astra Linux Special Edition 1.7

ОС	Версия ОС	Название бокса	URL
Astra Linux Special Edition базовый (Орёл)	1.7.5uu1	alse-base/1.7.5uu1	https://registry.astralinux.ru/vagrant/alse-base%2F1.7.5uu1
Astra Linux Special Edition усиленный (Воронеж)	1.7.5uu1	alse-adv/1.7.5uu1	https://registry.astralinux.ru/vagrant/alse-adv%2F1.7.5uu1
Astra Linux Special Edition максимальный (Смоленск)	1.7.5uu1	alse-max/1.7.5uu1	https://registry.astralinux.ru/vagrant/alse-max%2F1.7.5uu1
Astra Linux Special Edition базовый (Орёл) + GUI	1.7.5uu1	alse-gui-base/1.7.5uu1	https://registry.astralinux.ru/vagrant/alse-gui-base%2F1.7.5uu1
Astra Linux Special Edition усиленный (Воронеж) + GUI	1.7.5uu1	alse-gui-adv/1.7.5uu1	https://registry.astralinux.ru/vagrant/alse-gui-adv%2F1.7.5uu1
Astra Linux Special Edition максимальный (Смоленск) + GUI	1.7.5uu1	alse-gui-max/1.7.5uu1	https://registry.astralinux.ru/vagrant/alse-gui-max%2F1.7.5uu1

continues on next page

Таблица 1 - продолжение с предыдущей страницы

ОС	Версия ОС	Название бокса	URL
Astra Linux Special Edition базовый (Орёл)	1.7.5uu1	alse-vanilla-base/1.7.5uu1	https://registry.astralinux.ru/vagrant/alse-vanilla-base%2F1.7.5uu1
Astra Linux Special Edition усиленный (Воронеж)	1.7.5uu1	alse-vanilla-adv/1.7.5uu1	https://registry.astralinux.ru/vagrant/alse-vanilla-adv%2F1.7.5uu1
Astra Linux Special Edition максимальный (Смоленск)	1.7.5uu1	alse-vanilla-max/1.7.5uu1	https://registry.astralinux.ru/vagrant/alse-vanilla-max%2F1.7.5uu1
Astra Linux Special Edition базовый (Орёл) + GUI	1.7.5uu1	alse-vanilla-gui-base/1.7.5uu1	https://registry.astralinux.ru/vagrant/alse-vanilla-gui-base%2F1.7.5uu1
Astra Linux Special Edition усиленный (Воронеж) + GUI	1.7.5uu1	alse-vanilla-gui-adv/1.7.5uu1	https://registry.astralinux.ru/vagrant/alse-vanilla-gui-adv%2F1.7.5uu1
Astra Linux Special Edition максимальный (Смоленск) + GUI	1.7.5uu1	alse-vanilla-gui-max/1.7.5uu1	https://registry.astralinux.ru/vagrant/alse-vanilla-gui-max%2F1.7.5uu1
Astra Linux Special Edition усиленный (Воронеж)	1.7.5	alse-vanilla-adv/1.7.5	https://registry.astralinux.ru/vagrant/alse-vanilla-adv%2F1.7.5
Astra Linux Special Edition базовый (Орёл)	1.7.5	alse-vanilla-base/1.7.5	https://registry.astralinux.ru/vagrant/alse-vanilla-base%2F1.7.5
Astra Linux Special Edition максимальный (Смоленск)	1.7.5	alse-vanilla-max/1.7.5	https://registry.astralinux.ru/vagrant/alse-vanilla-max%2F1.7.5
Astra Linux Special Edition базовый (Орёл) + GUI	1.7.5	alse-vanilla-gui-base/1.7.5	https://registry.astralinux.ru/vagrant/alse-vanilla-gui-base%2F1.7.5
Astra Linux Special Edition усиленный (Воронеж) + GUI	1.7.5	alse-vanilla-gui-adv/1.7.5	https://registry.astralinux.ru/vagrant/alse-vanilla-gui-adv%2F1.7.5
Astra Linux Special Edition максимальный (Смоленск) + GUI	1.7.5	alse-vanilla-gui-max/1.7.5	https://registry.astralinux.ru/vagrant/alse-vanilla-gui-max%2F1.7.5
Astra Linux Special Edition базовый (Орёл)	1.7.4uu1	alse-vanilla-base/1.7.4uu1	https://registry.astralinux.ru/vagrant/alse-vanilla-base%2F1.7.4uu1
Astra Linux Special Edition усиленный (Воронеж)	1.7.4uu1	alse-vanilla-adv/1.7.4uu1	https://registry.astralinux.ru/vagrant/alse-vanilla-adv%2F1.7.4uu1

continues on next page

Таблица 1 - продолжение с предыдущей страницы

ОС	Версия ОС	Название бокса	URL
Astra Linux Special Edition максимальный (Смоленск)	1.7.4uu1	alse-vanilla-max/1.7.4uu1	https://registry.astralinux.ru/vagrant/alse-vanilla-max%2F1.7.4uu1
Astra Linux Special Edition базовый (Орёл) + GUI	1.7.4uu1	alse-vanilla-gui-base/1.7.4uu1	https://registry.astralinux.ru/vagrant/alse-vanilla-gui-base%2F1.7.4uu1
Astra Linux Special Edition усиленный (Воронеж) + GUI	1.7.4uu1	alse-vanilla-gui-adv/1.7.4uu1	https://registry.astralinux.ru/vagrant/alse-vanilla-gui-adv%2F1.7.4uu1
Astra Linux Special Edition максимальный (Смоленск) + GUI	1.7.4uu1	alse-vanilla-gui-max/1.7.4uu1	https://registry.astralinux.ru/vagrant/alse-vanilla-gui-max%2F1.7.4uu1
Astra Linux Special Edition базовый (Орёл)	1.7.4	alse-vanilla-base/1.7.4	https://registry.astralinux.ru/vagrant/alse-vanilla-base%2F1.7.4
Astra Linux Special Edition усиленный (Воронеж)	1.7.4	alse-vanilla-adv/1.7.4	https://registry.astralinux.ru/vagrant/alse-vanilla-adv%2F1.7.4
Astra Linux Special Edition максимальный (Смоленск)	1.7.4	alse-vanilla-max/1.7.4	https://registry.astralinux.ru/vagrant/alse-vanilla-max%2F1.7.4
Astra Linux Special Edition базовый (Орёл) + GUI	1.7.4	alse-vanilla-gui-base/1.7.4	https://registry.astralinux.ru/vagrant/alse-vanilla-gui-base%2F1.7.4
Astra Linux Special Edition усиленный (Воронеж) + GUI	1.7.4	alse-vanilla-gui-adv/1.7.4	https://registry.astralinux.ru/vagrant/alse-vanilla-gui-adv%2F1.7.4
Astra Linux Special Edition максимальный (Смоленск) + GUI	1.7.4	alse-vanilla-gui-max/1.7.4	https://registry.astralinux.ru/vagrant/alse-vanilla-gui-max%2F1.7.4
Astra Linux Special Edition базовый (Орёл)	1.7.3uu2	alse-vanilla-base/1.7.3uu2	https://registry.astralinux.ru/vagrant/alse-vanilla-base%2F1.7.3uu2
Astra Linux Special Edition усиленный (Воронеж)	1.7.3uu2	alse-vanilla-adv/1.7.3uu2	https://registry.astralinux.ru/vagrant/alse-vanilla-adv%2F1.7.3uu2
Astra Linux Special Edition максимальный (Смоленск)	1.7.3uu2	alse-vanilla-max/1.7.3uu2	https://registry.astralinux.ru/vagrant/alse-vanilla-max%2F1.7.3uu2

continues on next page

Таблица 1 - продолжение с предыдущей страницы

ОС	Версия ОС	Название бокса	URL
Astra Linux Special Edition базовый (Орёл) + GUI	1.7.3uu2	alse-vanilla-gui-base/1.7.3uu2	https://registry.astralinux.ru/vagrant/alse-vanilla-gui-base%2F1.7.3uu2
Astra Linux Special Edition усиленный (Воронеж) + GUI	1.7.3uu2	alse-vanilla-gui-adv/1.7.3uu2	https://registry.astralinux.ru/vagrant/alse-vanilla-gui-adv%2F1.7.3uu2
Astra Linux Special Edition максимальный (Смоленск) + GUI	1.7.3uu2	alse-vanilla-gui-max/1.7.3uu2	https://registry.astralinux.ru/vagrant/alse-vanilla-gui-max%2F1.7.3uu2
Astra Linux Special Edition базовый (Орёл)	1.7.3	alse-vanilla-base/1.7.3	https://registry.astralinux.ru/vagrant/alse-vanilla-base%2F1.7.3
Astra Linux Special Edition усиленный (Воронеж)	1.7.3	alse-vanilla-adv/1.7.3	https://registry.astralinux.ru/vagrant/alse-vanilla-adv%2F1.7.3
Astra Linux Special Edition максимальный (Смоленск)	1.7.3	alse-vanilla-max/1.7.3	https://registry.astralinux.ru/vagrant/alse-vanilla-max%2F1.7.3
Astra Linux Special Edition базовый (Орёл)	1.7.3uu1	alse-vanilla-base/1.7.3uu1	https://registry.astralinux.ru/vagrant/alse-vanilla-base%2F1.7.3uu1
Astra Linux Special Edition усиленный (Воронеж)	1.7.3uu1	alse-vanilla-adv/1.7.3uu1	https://registry.astralinux.ru/vagrant/alse-vanilla-adv%2F1.7.3uu1
Astra Linux Special Edition максимальный (Смоленск)	1.7.3uu1	alse-vanilla-max/1.7.3uu1	https://registry.astralinux.ru/vagrant/alse-vanilla-max%2F1.7.3uu1
Astra Linux Special Edition базовый (Орёл) + GUI	1.7.3	alse-vanilla-gui-base/1.7.3	https://registry.astralinux.ru/vagrant/alse-vanilla-gui-base%2F1.7.3
Astra Linux Special Edition усиленный (Воронеж) + GUI	1.7.3	alse-vanilla-gui-adv/1.7.3	https://registry.astralinux.ru/vagrant/alse-vanilla-gui-adv%2F1.7.3
Astra Linux Special Edition максимальный (Смоленск) + GUI	1.7.3	alse-vanilla-gui-max/1.7.3	https://registry.astralinux.ru/vagrant/alse-vanilla-gui-max%2F1.7.3
Astra Linux Special Edition базовый (Орёл) + GUI	1.7.3uu1	alse-vanilla-gui-base/1.7.3uu1	https://registry.astralinux.ru/vagrant/alse-vanilla-gui-base%2F1.7.3uu1

continues on next page

Таблица 1 - продолжение с предыдущей страницы

ОС	Версия ОС	Название бокса	URL
Astra Linux Special Edition усиленный (Воронеж) + GUI	1.7.3uu1	alse-vanilla-gui-adv/1.7.3uu1	https://registry.astralinux.ru/vagrant/alse-vanilla-gui-adv%2F1.7.3uu1
Astra Linux Special Edition максимальный (Смоленск) + GUI	1.7.3uu1	alse-vanilla-gui-max/1.7.3uu1	https://registry.astralinux.ru/vagrant/alse-vanilla-gui-max%2F1.7.3uu1
Astra Linux Special Edition базовый (Орёл)	1.7.2	alse-vanilla-base/1.7.2	https://registry.astralinux.ru/vagrant/alse-vanilla-base%2F1.7.2
Astra Linux Special Edition усиленный (Воронеж)	1.7.2	alse-vanilla-adv/1.7.2	https://registry.astralinux.ru/vagrant/alse-vanilla-adv%2F1.7.2
Astra Linux Special Edition максимальный (Смоленск)	1.7.2	alse-vanilla-max/1.7.2	https://registry.astralinux.ru/vagrant/alse-vanilla-max%2F1.7.2
Astra Linux Special Edition базовый (Орёл) + GUI	1.7.2	alse-vanilla-gui-base/1.7.2	https://registry.astralinux.ru/vagrant/alse-vanilla-gui-base%2F1.7.2
Astra Linux Special Edition усиленный (Воронеж) + GUI	1.7.2	alse-vanilla-gui-adv/1.7.2	https://registry.astralinux.ru/vagrant/alse-vanilla-gui-adv%2F1.7.2
Astra Linux Special Edition максимальный (Смоленск) + GUI	1.7.2	alse-vanilla-gui-max/1.7.2	https://registry.astralinux.ru/vagrant/alse-vanilla-gui-max%2F1.7.2
Astra Linux Special Edition базовый (Орёл)	1.7.1	alse-vanilla-base/1.7.1	https://registry.astralinux.ru/vagrant/alse-vanilla-base%2F1.7.1
Astra Linux Special Edition усиленный (Воронеж)	1.7.1	alse-vanilla-adv/1.7.1	https://registry.astralinux.ru/vagrant/alse-vanilla-adv%2F1.7.1
Astra Linux Special Edition максимальный (Смоленск)	1.7.1	alse-vanilla-max/1.7.1	https://registry.astralinux.ru/vagrant/alse-vanilla-max%2F1.7.1
Astra Linux Special Edition базовый (Орёл) + GUI	1.7.1	alse-vanilla-gui-base/1.7.1	https://registry.astralinux.ru/vagrant/alse-vanilla-gui-base%2F1.7.1
Astra Linux Special Edition усиленный (Воронеж) + GUI	1.7.1	alse-vanilla-gui-adv/1.7.1	https://registry.astralinux.ru/vagrant/alse-vanilla-gui-adv%2F1.7.1
Astra Linux Special Edition максимальный (Смоленск) + GUI	1.7.1	alse-vanilla-gui-max/1.7.1	https://registry.astralinux.ru/vagrant/alse-vanilla-gui-max%2F1.7.1

continues on next page

Таблица 1 - продолжение с предыдущей страницы

ОС	Версия ОС	Название бокса	URL
Astra Linux Special Edition базовый (Орёл)	1.7.0	alse-vanilla-base/1.7.0	https://registry.astralinux.ru/vagrant/alse-vanilla-base%2F1.7.0
Astra Linux Special Edition усиленный (Воронеж)	1.7.0	alse-vanilla-adv/1.7.0	https://registry.astralinux.ru/vagrant/alse-vanilla-adv%2F1.7.0
Astra Linux Special Edition максимальный (Смоленск)	1.7.0	alse-vanilla-max/1.7.0	https://registry.astralinux.ru/vagrant/alse-vanilla-max%2F1.7.0
Astra Linux Special Edition базовый (Орёл) + GUI	1.7.0	alse-vanilla-gui-base/1.7.0	https://registry.astralinux.ru/vagrant/alse-vanilla-gui-base%2F1.7.0
Astra Linux Special Edition усиленный (Воронеж) + GUI	1.7.0	alse-vanilla-gui-adv/1.7.0	https://registry.astralinux.ru/vagrant/alse-vanilla-gui-adv%2F1.7.0
Astra Linux Special Edition максимальный (Смоленск) + GUI	1.7.0	alse-vanilla-gui-max/1.7.0	https://registry.astralinux.ru/vagrant/alse-vanilla-gui-max%2F1.7.0

Astra Linux Common Edition 2.12

ОС	Версия ОС	Название бокса	URL
Орёл	2.12.46	orel-vanilla/2.12.4	https://registry.astralinux.ru/vagrant/orel-vanilla%2F2.12.46
Орёл + GUI	2.12.46	orel-vanilla-gui/2.12.46	https://registry.astralinux.ru/vagrant/orel-vanilla-gui%2F2.12.46
Орёл	2.12.45	orel-vanilla/2.12.4	https://registry.astralinux.ru/vagrant/orel-vanilla%2F2.12.45
Орёл + GUI	2.12.45	orel-vanilla-gui/2.12.45	https://registry.astralinux.ru/vagrant/orel-vanilla-gui%2F2.12.45
Орёл	2.12.44	orel-vanilla/2.12.4	https://registry.astralinux.ru/vagrant/orel-vanilla%2F2.12.44
Орёл + GUI	2.12.44	orel-vanilla-gui/2.12.44	https://registry.astralinux.ru/vagrant/orel-vanilla-gui%2F2.12.44
Орёл	2.12.42	orel-vanilla/2.12.4	https://registry.astralinux.ru/vagrant/orel-vanilla%2F2.12.42
Орёл + GUI	2.12.42	orel-vanilla-gui/2.12.42	https://registry.astralinux.ru/vagrant/orel-vanilla-gui%2F2.12.42
Орёл	2.12.40	orel-vanilla/2.12.4	https://registry.astralinux.ru/vagrant/orel-vanilla%2F2.12.40
Орёл + GUI	2.12.40	orel-vanilla-gui/2.12.40	https://registry.astralinux.ru/vagrant/orel-vanilla-gui%2F2.12.40
Орёл	2.12.29	orel-vanilla/2.12.2	https://registry.astralinux.ru/vagrant/orel-vanilla%2F2.12.29
Орёл + GUI	2.12.29	orel-vanilla-gui/2.12.29	https://registry.astralinux.ru/vagrant/orel-vanilla-gui%2F2.12.29

Astra Linux Special Edition 1.6

ОС	Версия ОС	Название бокса	URL
Смоленск	1.6.14	smolensk-vanilla/1.6.14	https://registry.astralinux.ru/vagrant/smolensk-vanilla%2F1.6.14
Смоленск + GUI	1.6.14	smolensk-vanilla-gui/1.6.14	https://registry.astralinux.ru/vagrant/smolensk-vanilla-gui%2F1.6.14
Смоленск	1.6.13	smolensk-vanilla/1.6.13	https://registry.astralinux.ru/vagrant/smolensk-vanilla%2F1.6.13
Смоленск + GUI	1.6.13	smolensk-vanilla-gui/1.6.13	https://registry.astralinux.ru/vagrant/smolensk-vanilla-gui%2F1.6.13
Смоленск	1.6.12	smolensk-vanilla/1.6.12	https://registry.astralinux.ru/vagrant/smolensk-vanilla%2F1.6.12
Смоленск + GUI	1.6.12	smolensk-vanilla-gui/1.6.12	https://registry.astralinux.ru/vagrant/smolensk-vanilla-gui%2F1.6.12
Смоленск	1.6.11	smolensk-vanilla/1.6.11	https://registry.astralinux.ru/vagrant/smolensk-vanilla%2F1.6.11
Смоленск + GUI	1.6.11	smolensk-vanilla-gui/1.6.11	https://registry.astralinux.ru/vagrant/smolensk-vanilla-gui%2F1.6.11
Смоленск	1.6.10	smolensk-vanilla/1.6.10	https://registry.astralinux.ru/vagrant/smolensk-vanilla%2F1.6.10
Смоленск + GUI	1.6.10	smolensk-vanilla-gui/1.6.10	https://registry.astralinux.ru/vagrant/smolensk-vanilla-gui%2F1.6.10
Смоленск	1.6.9	smolensk-vanilla/1.6.9	https://registry.astralinux.ru/vagrant/smolensk-vanilla%2F1.6.9
Смоленск + GUI	1.6.9	smolensk-vanilla-gui/1.6.9	https://registry.astralinux.ru/vagrant/smolensk-vanilla-gui%2F1.6.9
Смоленск	1.6.8	smolensk-vanilla/1.6.8	https://registry.astralinux.ru/vagrant/smolensk-vanilla%2F1.6.8
Смоленск + GUI	1.6.8	smolensk-vanilla-gui/1.6.8	https://registry.astralinux.ru/vagrant/smolensk-vanilla-gui%2F1.6.8
Смоленск	1.6.6	smolensk-vanilla/1.6.6	https://registry.astralinux.ru/vagrant/smolensk-vanilla%2F1.6.6
1.3. Уровень защищённости ОС			31
Смоленск + GUI	1.6.6	smolensk-	

VMware vSphere

Образы предназначены для работы в системе виртуализации VMware vSphere.

Уровни защищённости ОС:

- базовый (Орёл);
- усиленный (Воронеж);
- максимальный (Смоленск).

Конфигурация образа

- Образы виртуальной машины распространяются в формате OVA (Open Virtual Appliance) и содержат следующие настройки VM:
 - количество vCPU - 1;
 - объём RAM, ГБ - 2;
 - размер диска, ГБ - 16;
 - количество сетевых адаптеров - 1;
 - Hardware version - 11.
- Дополнительно установлен пакет `open-vm-tools`.

Примечание: Логин и пароль по умолчанию: `astra/astra`.

Необходимо изменить пароль при первом запуске VM.

Инструкции по использованию

В примере используется образ Astra Linux Special Edition 1.7.5uu1 в режиме базовый (Орёл). Этот и другие образы для VMware доступны по [ссылке](#)¹¹.

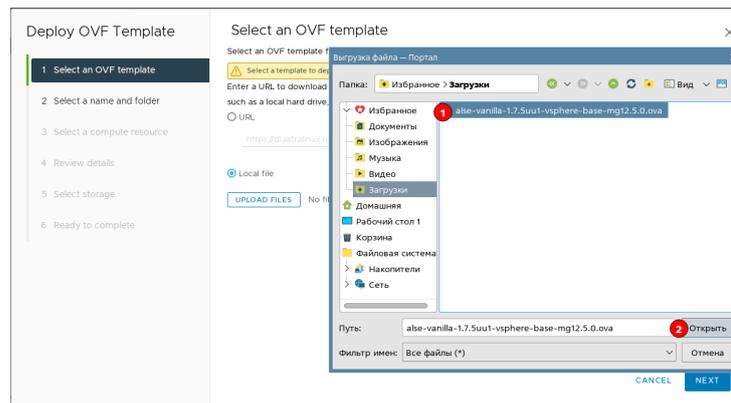
Пример запуска образа в VMware vSphere

1. Выберите *Actions* ▶ *Deploy OVF Template*.

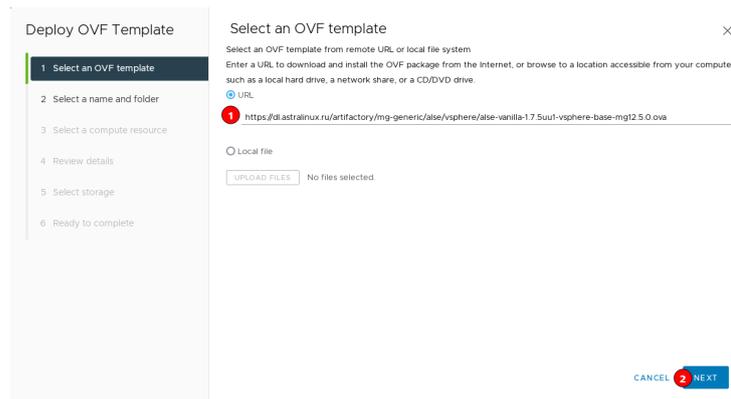


2. Если файл образа доступен локально, выберите необходимый образ через меню *Local file* ▶ *Upload files*.

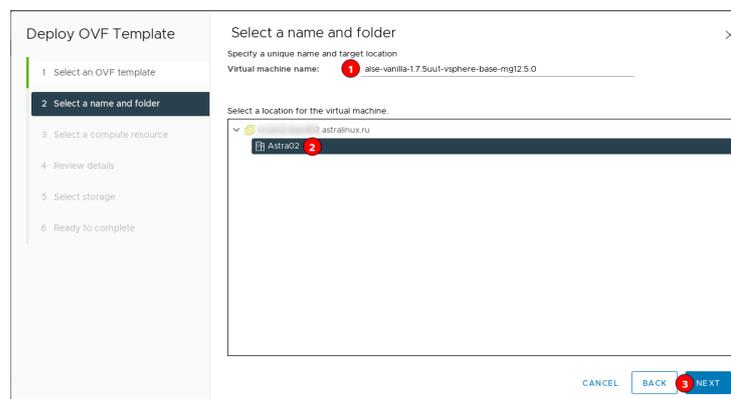
¹¹ <https://registry.astralinux.ru/images/alse/vsphere/>



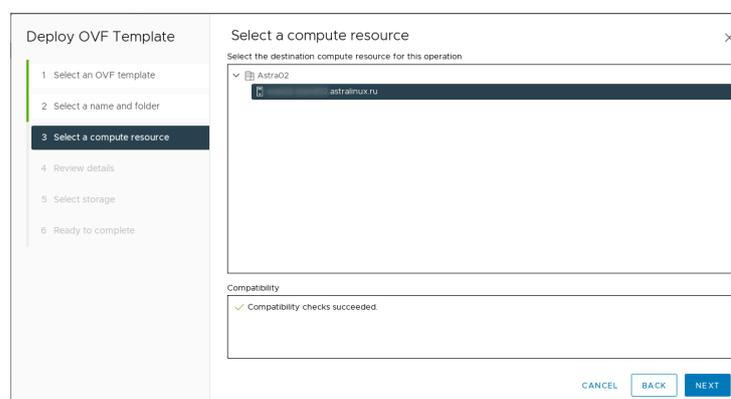
В противном случае укажите URL файла образа.



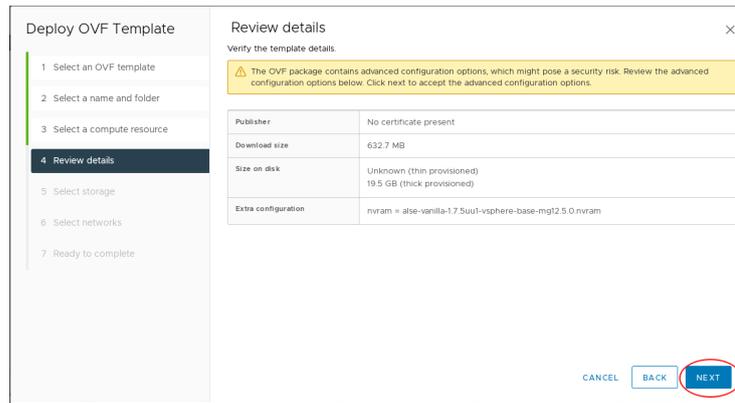
3. Укажите название VM и место её хранения.



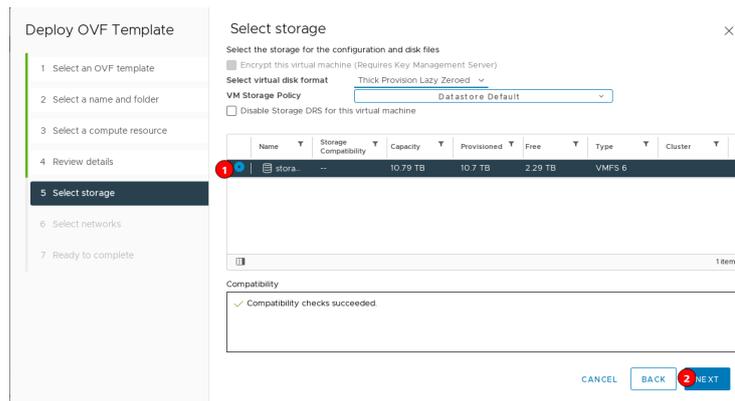
4. Выберите нужный узел ESXi для создания VM.



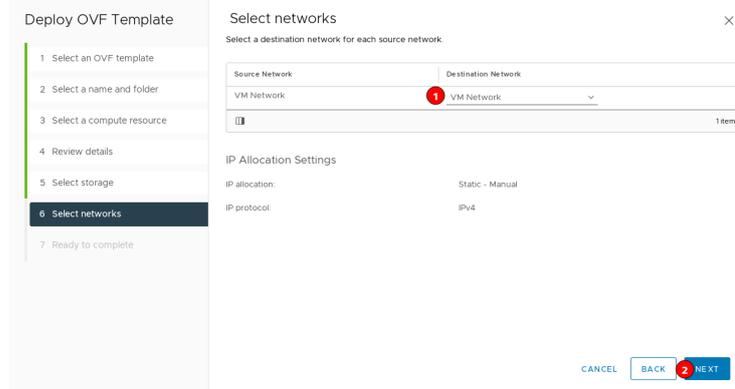
5. Нажмите кнопку *Next*.



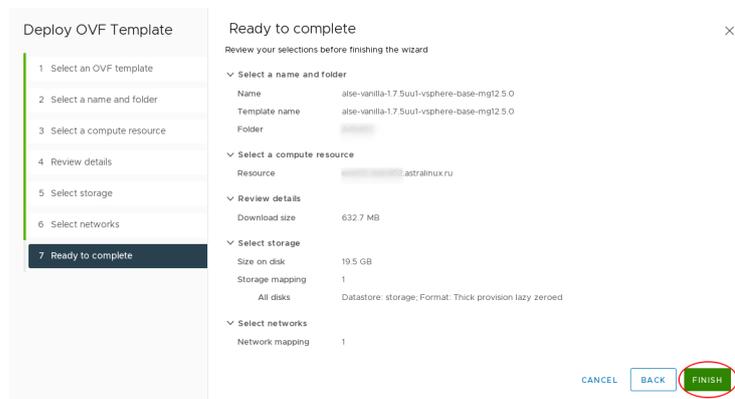
6. Выберите место хранения диска VM и нажмите кнопку *Next*.



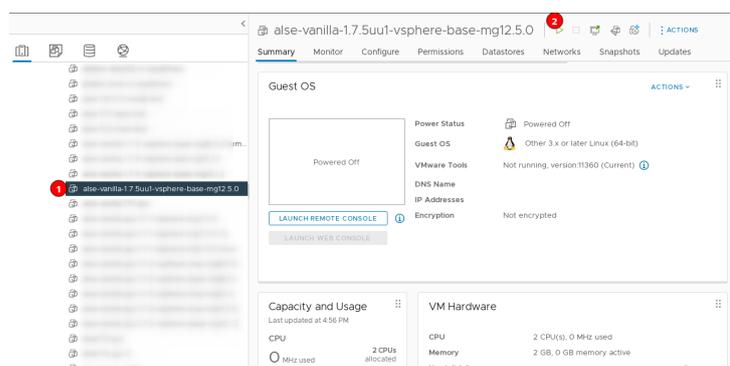
7. Выберите сеть.



8. Проверьте корректность данных и нажмите кнопку *Finish*.



9. Дождитесь импорта машины, выберите её в списке VM и нажмите кнопку запуска.



VirtualBox

Образы предназначены для работы в системе виртуализации VirtualBox.

Уровни защищённости ОС:

- базовый (Орёл);
- усиленный (Воронеж);
- максимальный (Смоленск).

Конфигурация

- Для сборки образов используется VirtualBox 6.1.38. Образы распространяются в формате OVA и содержат следующие настройки VM:
 - количество vCPU – 2;
 - объём RAM, ГБ – 2;
 - размер диска, ГБ – 30;
 - количество сетевых адаптеров – 1.
- Установлен пакет расширений Virtual Box Guest Additions версии 6.1.38.

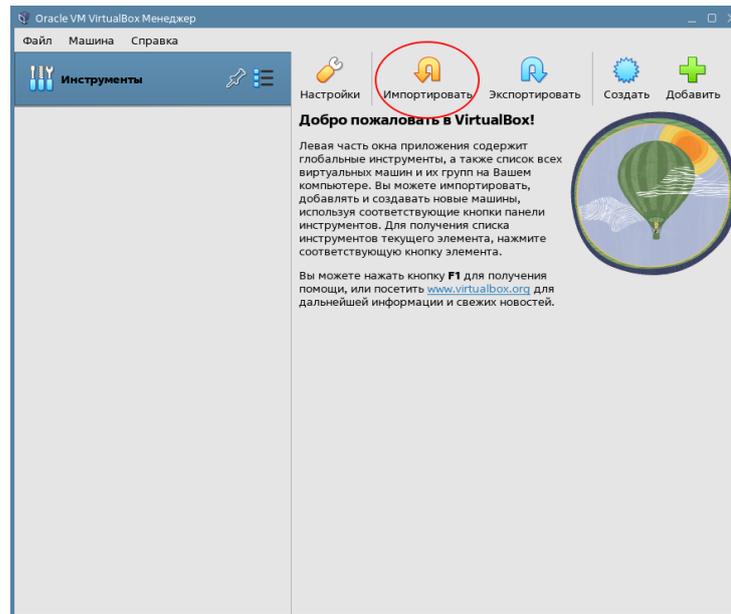
Примечание: Логин и пароль по умолчанию: astra/astra.

Инструкции по использованию

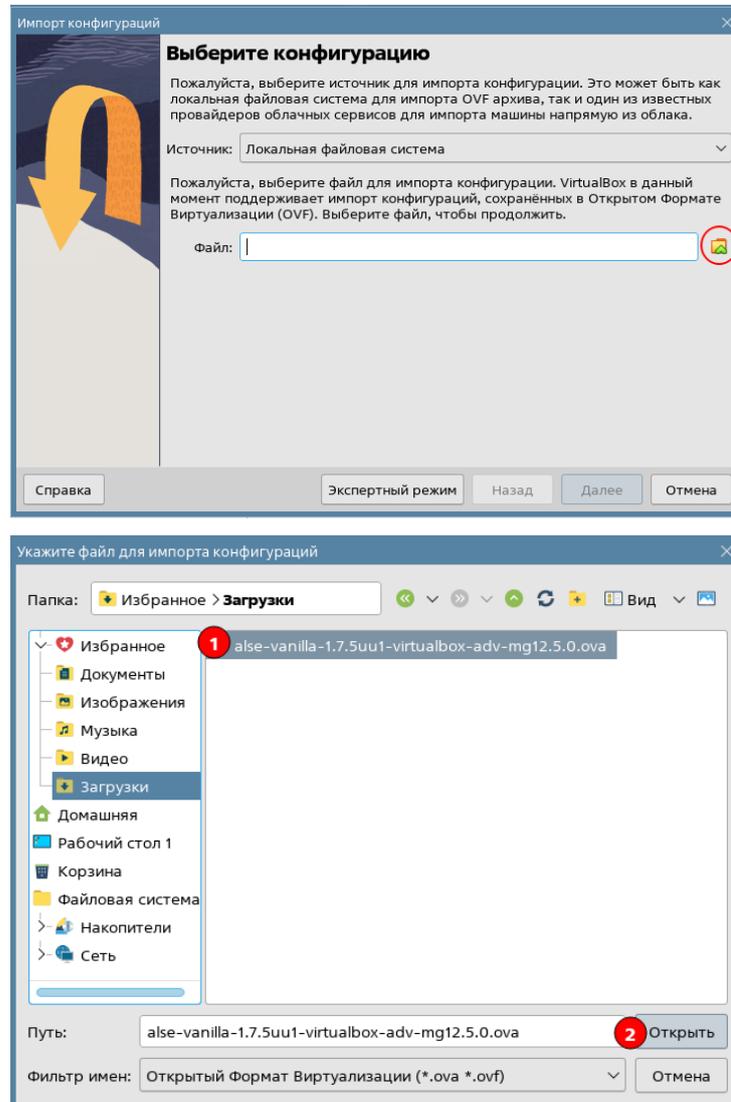
В примере используется образ Astra Linux Special Edition 1.7.5uu1 в режиме усиленный (Воронеж). Все образы для VirtualBox доступны по [ссылке](#)¹².

1. Установите VirtualBox версии не ниже 6.1 по [инструкции](#).
2. Запустите VirtualBox через меню *Системные* ▶ *Oracle VM VirtualBox*.
3. Нажмите кнопку *Импортировать*:

¹² <https://registry.astralinux.ru/images/alse/virtualbox/>

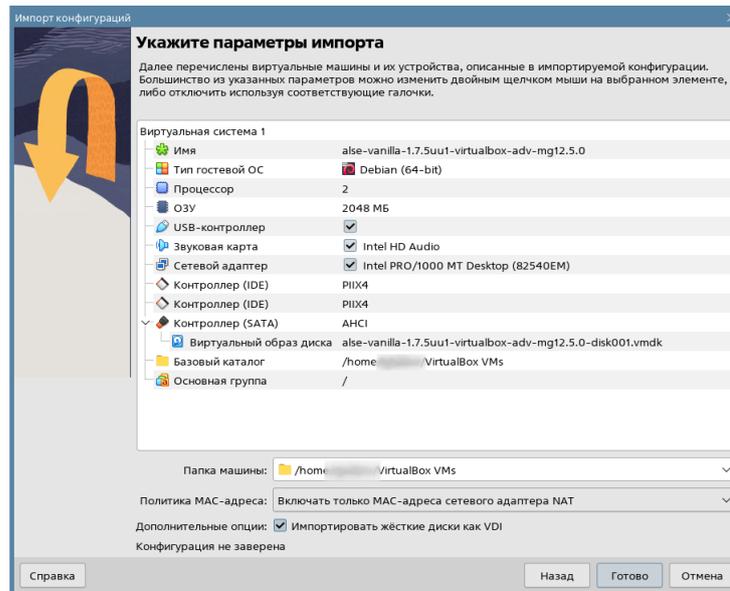


4. Выберите файл образа:



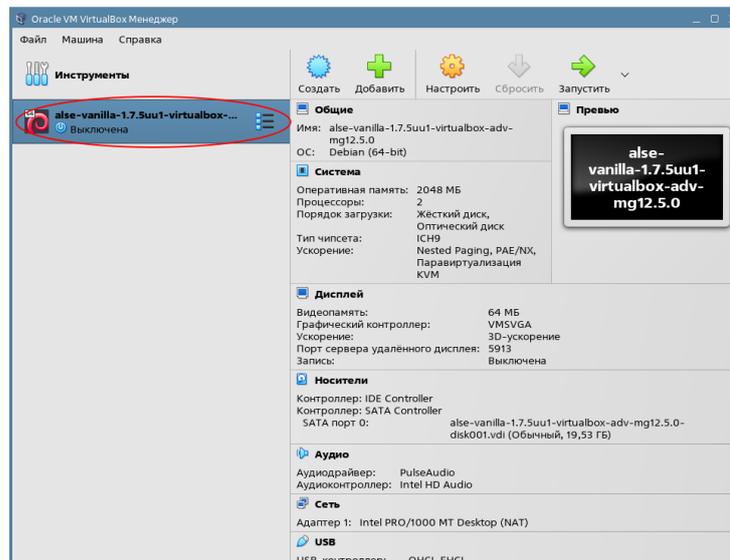
5. Нажмите кнопку *Далее*.

6. Укажите необходимые параметры:



7. Нажмите кнопку *Готово*.

8. Дождитесь импорта ВМ, затем выберите её из списка:



9. Нажмите кнопку *Запустить*.

1.3.3 Руководства пользователя

Раздел содержит материалы и пошаговые инструкции по инструментам виртуализации и автоматизации.

Cloud-init

Cloud-init — это инструмент для настройки облачного сервера, который позволяет передать пользовательские параметры (*user-data*) при запуске виртуальной машины. Это позволяет ускорить и автоматизировать процесс настройки серверов.

Дополнительно с работой и настройкой cloud-init можно ознакомиться на сайте с [официальной документацией](https://cloudinit.readthedocs.io/en/latest/index.html)¹³.

¹³ <https://cloudinit.readthedocs.io/en/latest/index.html>

Сценарии cloud-init

С помощью cloud-init вы можете настроить конфигурацию сервера, установить нужное ПО, создать пользователей и каталоги, назначить права доступа.

Сценарии cloud-init можно использовать при развёртывании инфраструктуры с помощью [Terraform](#).

Форматы

Настройки cloud-init могут быть описаны декларативно в формате YAML или императивно в виде сценариев оболочки (Shell-скриптов).

Содержимое файла интерпретируется в зависимости от заголовка:

- YAML:

```
#cloud-config
```

- сценарий оболочки:

```
#!/bin/sh
```

Создание пользователя

Пример создания пользователя со следующими свойствами:

- имя - administrator;
- использование механизма sudo без ввода пароля;
- аутентификация по SSH с помощью ключа;
- высокий уровень целостности;
- интерпретатор Bash в качестве оболочки командной строки.

```
#cloud-config
users:
- name: administrator
  groups: astra-admin
  shell: /bin/bash
  sudo: ['ALL=(ALL) NOPASSWD:ALL']
  parsec_user_max_ilev: high
  ssh-authorized-keys:
  - ssh-ed25519 AAAAC3...zvNm administrator@example.com
  - ssh-rsa AAAAB3...8Vc= administrator@example.com
```

Здесь:

- name - имя пользователя.
- groups - список дополнительных групп, в которые должен быть включен пользователь.
- parsec_user_max_ilev - уровень целостности.
- ssh-authorized-keys - публичные ключи SSH, которые можно будет использовать для доступа к VM.

Важно: Для администрирования системы необходимо, чтобы пользователь был членом группы astra-admin и имел высокий уровень целостности.

Инструкции по работе с пользователями доступны в разделе [Управление пользователями](#).

Изменение размера диска

Чтобы увеличить размер диска, используемого в libvirt, добавьте в настройки cloud-init следующие параметры:

```
#cloud-config
growpart:
  mode: auto
  devices: ['/']
```

Размер файловой системы будет увеличен до целевого размера диска.

Выполнение команд

В данном примере при первом запуске VM выполняется обновление ОС:

```
#cloud-config
runcmd:
  apt-get update
  apt-get -y upgrade
```

Установка пакетов

Установка пакетов при первом запуске VM:

```
#cloud-config
package_update: true
packages:
  - nginx
  - ntp
```

Изменение размера диска в образе Cloud

Для образов Cloud установлен размер диска по умолчанию 3000 МБ.

Для изменения размера диска VM в Yandex Cloud необходимо указать требуемый размер, не менее 3 ГБ. Размер указывается через [Консоль управления Yandex Cloud](#)¹⁴ или в конфигурационных файлах Terraform.

Образ Cloud возможно установить для провайдера libvirt с помощью Terraform. В этом случае необходимо выполнить следующие действия:

- указать требуемый размер диска (не менее 3 ГБ);
- добавить в конфигурацию cloud-init строки:

```
growpart:
  mode: auto
  devices: ['/']
```

Пример конфигурационного файла Terraform (фрагмент):

```
resource "libvirt_volume" "volume_orel_vanilla" {
  name = "orev-vanilla.qcow2"
  pool = "default"
  size = 5361393152
```

(continues on next page)

¹⁴ <https://console.cloud.yandex.ru/>

```

}

# Use CloudInit to add our ssh-key to the instance
resource "libvirt_cloudinit_disk" "cloudinit_orel_vanilla" {
  name = "cloudinit_orel_vanilla.iso"
  pool = "default"

  user_data = <<EOF
#cloud-config
growpart:
  mode: auto
  devices: ['/']
EOF
}

resource "libvirt_domain" "domain_orel_vanilla" {
  cloudinit = libvirt_cloudinit_disk.cloudinit_orel_vanilla.id
}

```

Terraform

Terraform представляет собой систему автоматизации развёртывания и обновления информационной инфраструктуры с использованием доступного для организации оборудования и программного обеспечения. Для описания требуемой инфраструктуры Terraform предлагает декларативный язык на базе HCL.

Создание пользователя в Yandex Cloud при помощи Terraform

В данном примере в Yandex Cloud создаётся пользователь, обладающий следующими свойствами:

- имя - administrator;
- использование механизма sudo без ввода пароля;
- аутентификация по SSH с помощью ключа;
- высокий уровень целостности;
- интерпретатор Bash в качестве оболочки командной строки.

1. В каталоге с описанием инфраструктуры создайте файл cloud-init.cfg со следующим содержимым:

```

#cloud-config
users:
  - name: administrator
    groups: astra-admin
    shell: /bin/bash
    sudo: ['ALL=(ALL) NOPASSWD:ALL']
    parsec_user_max_ilev: high
    ssh-authorized-keys:
      - ssh-ed25519 AAAAC3...zvNm administrator@example.com

```

2. В файле конфигурации Terraform добавьте к описанию VM следующие строки:

```

metadata = {
  user-data = "${file("./cloud-init.cfg")}"
}

```

Создание VM в libvirt при помощи Terraform

В данном примере создается VM в системе виртуализации libvirt:

Примечание: При использовании Astra Linux Special Edition 1.7.5 версия провайдера Terraform dmacvicar/libvirt должна быть не выше 0.7.1.

Пример кода

Список 1: cloud_config.cfg

```
---
#cloud-config
users:
  - name: astra
    groups: astra-admin
    shell: /bin/bash
    sudo: ["ALL=(ALL) NOPASSWD:ALL"]
    ssh_authorized_keys: "AAAAC3...zvNm administrator@example.com"

growpart:
  mode: auto
  devices: ["/"]
```

Список 2: network_config.cfg

```
---
version: 1
config:
  - type: physical
    name: eth0
    subnets:
      - type: static
        address: 192.168.122.22
        netmask: 255.255.255.0
        gateway: 192.168.122.1
        dns_nameservers:
          - 192.168.122.1
```

Список 3: main.tf

```
terraform {
  required_version = ">= 1.0"
  required_providers {
    libvirt = {
      source = "dmacvicar/libvirt"
      version = "0.7.1"
    }
  }
}

provider "libvirt" {
  uri = "qemu:///system"
}

variable "pool" {
  type = string
  default = "mg_test_pool"
}

variable "volume_id" {
  type = string
```

(continues on next page)

```
default = "images/alse-1.7.5uul-base-cloudinit-mg13.3.0-amd64.qcow2"
}

resource "libvirt_pool" "test" {
  name = var.pool
  type = "dir"
  path = "/media/mg_test_pool"
}

resource "libvirt_volume" "test" {
  name     = "testdisk"
  pool     = var.pool
  source   = var.volume_id
}

data "template_file" "user_data" {
  template = file("${path.module}/cloud_init.cfg")
}

data "template_file" "network_config" {
  template = file("${path.module}/network_config.cfg")
}

resource "libvirt_cloudinit_disk" "commoninit" {
  name           = "commoninit.test.iso"
  user_data      = data.template_file.user_data.rendered
  network_config = data.template_file.network_config.rendered
  pool           = var.pool
}

resource "libvirt_domain" "test" {
  name       = "test"
  memory    = "2048"
  vcpu      = "2"
  autostart = true
  qemu_agent = true

  cloudinit = libvirt_cloudinit_disk.commoninit.id

  cpu {
    mode = "host-passthrough"
  }

  disk {
    volume_id = libvirt_volume.test.id
  }

  boot_device {
    dev = ["hd"]
  }

  network_interface {
    hostname     = "test"
    bridge       = "virbr0"
    addresses    = ["192.168.122.22", "::1"]
    mac          = "52:54:00:8A:8B:8C"
    wait_for_lease = true
  }

  console {
    type           = "pty"
    target_type    = "virtio"
    target_port    = "1"
  }
}
```

(continues on next page)

(продолжение с предыдущей страницы)

```
graphics {
  type      = "spice"
  listen_type = "address"
  autoport  = true
}
}

output "name" {
  value = libvirt_domain.test.network_interface[0].addresses[0]
}
```

Управление пользователями

Вход в систему суперпользователя root по умолчанию заблокирован. Для администрирования ОС необходимо создать пользователя, который входит в группу `astra-admin` и имеет максимальный (63) уровень целостности. Пользователи, входящие в группу `astra-admin`, имеют право выполнять команды через механизм `sudo`.

В образах для локальной виртуализации такой пользователь уже создан. Название учётной записи и пароль этого пользователя см. в разделе [Описания образов](#).

В образах для облачных сервисов название учётной записи и учётные данные для входа в систему задаются на этапе создания VM с помощью механизмов `cloud-init` или `one-context`.

Создание пользователя

Чтобы создать пользователя выполните команду:

```
useradd -m <username>
```

Примечание: Домашний каталог пользователя в каталоге `/home` создаётся автоматически.

Добавление в группу

Чтобы добавить пользователя в группу выполните команду:

```
usermod -a -G <group> <username>
```

Здесь:

- `<group>` - название группы;
- `<username>` - название учётной записи пользователя.

Например, чтобы добавить пользователя `john` в группу `astra-admin` выполните команду:

```
usermod -a -G astra-admin john
```

Установка пароля

Чтобы установить пароль пользователю выполните команду:

```
passwd <username>
```

Установка целостности

Чтобы установить высокую целостность пользователю выполните команду:

```
pdpl-user -i 63 <username>
```

Чтобы проверить установку целостности выполните команду:

```
pdpl-user <username>
```

Vagrant

Vagrant¹⁵ относится к инструментам для создания VM (виртуальная машина) и объединения их в инфраструктуру в некотором виртуальном окружении, используя такие известные технологии, как VMware, VirtualBox и AWS. К такому виртуальному окружению относится, например, сервер с установленной на нем одной из упомянутых систем виртуализации.

Vagrant предоставляет развитый API, необходимый для автоматизации развёртывания систем. Следуя парадигме IAC, указания для Vagrant создают в файле Vagrantfile в виде директив, объединенных в общий поток последовательных действий.

Применение

Приложение Vagrant удобно использовать для отладки развёртывания пилотной инфраструктуры в локальной среде. Типовая последовательность использования Vagrant состоит из следующих шагов:

1. Создание и отладка потока управляющих директив на локальном узле с помощью VMware или VirtualBox.
2. Перенос полученного потока в производственную среду на базе облачных технологий или на собственные серверы.

В этом разделе рассматривается использование Vagrant совместно с VirtualBox.

Примечание: Приведенные ниже инструкции проверены в ОС Astra Linux Special Edition 1.7.4, 1.7.5 и 1.7.5uu1.

Установка

Инструкция по установке VirtualBox доступна в разделе [Virtualbox](#).

Приведенная ниже инструкция по установке и настройке ПО предназначена для ОС Astra Linux Special Edition 1.7. Пошаговые инструкции для других ОС доступны на [сайте разработчика](#)¹⁶.

Для установки Vagrant выполните следующие действия:

1. Загрузите DEB-пакет Vagrant с сайта разработчика, например:

¹⁵ <https://developer.hashicorp.com/vagrant>

¹⁶ <https://developer.hashicorp.com/vagrant/install>

```
wget https://releases.hashicorp.com/vagrant/2.4.1/vagrant_2.4.1-1_amd64.deb
```

Примечание: Если сайт разработчика по какой-либо причине недоступен, используйте для загрузки зеркало репозитория:

```
wget https://releases.comcloud.xyz/vagrant/2.4.1/vagrant_2.4.1-1_amd64.deb
```

- Установите загруженный пакет:

```
sudo dpkg -i vagrant_2.4.1-1_amd64.deb
```

- Определите путь к каталогу, в котором хранятся расширения Vagrant:

```
dpkg -L vagrant | egrep 'plugins/guests$'
```

В терминал выводится строка вида:

```
/opt/vagrant/embedded/gems/gems/vagrant-<version>/plugins/guests
```

где <version> – номер установленной версии Vagrant.

- Загрузите архив с расширением для Vagrant, позволяющим ему работать с образами Astra Linux:

```
wget https://dl.astralinux.ru/files/astra-vagrant.tar.gz
```

- Распакуйте загруженный архив в каталог с расширениями Vagrant, например:

```
sudo tar xf astra-vagrant.tar.gz -C <plugins_dir>
```

где <plugins_dir> – путь к каталогу с расширениями Vagrant.

Проверка корректности установки

Для проверки корректности установки Vagrant и VirtualBox выполните следующие действия:

- В любом каталоге создайте Vagrantfile с простейшей конфигурацией VM:

```
# frozen_string_literal: true

Vagrant.configure('2') do |config|
  config.vm.box = 'alse-vanilla-base/1.7.5uu1'
  config.vm.box_url = 'https://registry.astralinux.ru/vagrant/alse-vanilla-base%2F1.7.5uu1'

  config.vm.define 'VG' do |conf|
    conf.vm.hostname = 'VG'
  end
end
```

В этом файле описана VM с названием VG, использующая образ с Astra Linux Special Edition 1.7.5uu1.

Подробности об именовании и составе образов см. в разделе *Универсальные базовые образы Astra Linux*.

- Для создания и запуска VM выполните команду:

```
vagrant up
```

- Подключитесь к созданной VM по SSH:

```
vagrant ssh
```

При успешном подключении приглашение командной строки меняется на следующее:

```
vagrant@VG:~$
```

4. Для отключения от созданной ВМ выполните команду:

```
exit
```

Virtualbox

VirtualBox (Oracle VM VirtualBox) — программный продукт виртуализации для операционных систем Microsoft Windows, Linux, FreeBSD, macOS, Solaris/OpenSolaris, ReactOS, DOS и других.

Установка

Приведённая ниже инструкция по установке и настройке ПО предназначена для ОС Astra Linux Special Edition 1.7. Пошаговые инструкции для других ОС доступны на [сайте разработчика](#)¹⁷.

Подготовка к установке

Подготовьте ОС к установке и настройке VirtualBox.

1. Если установка производится на ВМ, включите в её настройках вложенную виртуализацию (nested virtualization). Для получения инструкций обратитесь к производителю ПО, используемого для виртуализации.
2. Получите номер установленного оперативного обновления ОС:

```
cat /etc/astra_version
```

Команда выводит в терминал строку вида:

```
1.7.5uu1
```

3. Добавьте в файл `/etc/apt/sources.list` ссылки на базовый (base) и расширенный (extended) репозитории Astra Linux Special Edition 1.7:

```
deb https://download.astralinux.ru/astra/frozen/1.7_x86-64/<version>/repository-  
↳base/      1.7_x86-64 main contrib non-free  
deb https://download.astralinux.ru/astra/frozen/1.7_x86-64/<version>/repository-  
↳extended/  1.7_x86-64 main contrib non-free
```

где `<version>` – версия установленного оперативного обновления Astra Linux Special Edition. Для примера выше указанные строки имеют вид:

```
deb https://download.astralinux.ru/astra/frozen/1.7_x86-64/1.7.5/uu/1/repository-  
↳base/      1.7_x86-64 main contrib non-free  
deb https://download.astralinux.ru/astra/frozen/1.7_x86-64/1.7.5/uu/1/repository-  
↳extended/  1.7_x86-64 main contrib non-free
```

4. Обновите список доступных пакетов:

```
sudo apt update
```

5. Получите номер версии и вариант сборки установленного ядра:

¹⁷ <https://www.virtualbox.org/>

```
uname -r
```

В терминал выводится строка вида:

```
6.1.50-1-generic
```

Здесь 6.1.50-1 – версия ядра, generic – вариант сборки.

- Установите менеджер загрузок `wget`, заголовочные файлы используемого ядра и утилиты, необходимые для сборки модулей ядра:

```
sudo apt install build-essential linux-headers-<version>-<build_variant> make wget -
↵-yes
```

где `<version>` и `<build_variant>` – версия ядра и вариант его сборки соответственно. Для примера выше нужная команда имеет вид:

```
sudo apt install build-essential linux-headers-6.1.50-1-generic make wget --yes
```

Совет: Шаги получения информации о версии ядра и установки его заголовочных файлов можно объединить в одну команду:

```
sudo apt install build-essential linux-headers-$(uname -r) make wget --yes
```

Установка VirtualBox

Для установки VirtualBox выполните следующие действия:

- Загрузите пакет `libvpx5` из репозитория Debian Linux, например:

```
wget http://security.debian.org/debian-security/pool/updates/main/libv/libvpx/
↵libvpx5_1.7.0-3+deb10u2_amd64.deb
```

Совет: Актуальная ссылка на загрузку доступна на странице пакета: <https://packages.debian.org/buster/amd64/libvpx5>

- Установите загруженный пакет:

```
sudo dpkg -i libvpx5_1.7.0-3+deb10u2_amd64.deb
```

- Импортируйте ключ репозитория VirtualBox:

```
wget https://www.virtualbox.org/download/oracle_vbox_2016.asc -O- | \
sudo gpg --dearmor --yes --output /etc/apt/trusted.gpg.d/oracle.gpg
```

- Создайте в каталоге `/etc/apt/sources.list.d/` файл `virtualbox.list` со следующим содержимым:

```
deb [arch=amd64] https://download.virtualbox.org/virtualbox/debian buster contrib
```

- Обновите список доступных пакетов:

```
sudo apt update
```

- Установите пакет `virtualbox-7.0`:

```
sudo apt install virtualbox-7.0 --yes
```

- Получите номер установленной версии VirtualBox:

```
apt policy virtualbox-7.0
```

В терминал выводятся строки вида:

```
virtualbox-7.0:
  Установлен: 7.0.16-162802~Debian~buster
  Кандидат:   7.0.16-162802~Debian~buster
  Таблица версий:
  *** 7.0.16-162802~Debian~buster 500
      500 https://download.virtualbox.org/virtualbox/debian buster/contrib amd64
↳ Packages
      100 /var/lib/dpkg/status
```

Здесь 7.0.16 – версия VirtualBox.

8. Загрузите [Extension Pack](#)¹⁸ для установленной версии VirtualBox:

```
wget https://download.virtualbox.org/virtualbox/<version>/Oracle_VM_VirtualBox_
↳Extension_Pack-<version>.vbox-extpack
```

где <version> – версия VirtualBox.

Для примера выше указанная команда имеет вид:

```
wget https://download.virtualbox.org/virtualbox/<version>/Oracle_VM_VirtualBox_
↳Extension_Pack-7.0.16.vbox-extpack
```

9. Установите Extension Pack:

```
sudo vboxmanage extpack install --replace <path>
```

где <path> – путь к загруженному ранее файлу.

Для примера выше указанная команда имеет вид:

```
sudo vboxmanage extpack install --replace ./Oracle_VM_VirtualBox_Extension_Pack-7.0.
↳16.vbox-extpack
```

1.3.4 Загрузка образов

Образы виртуальных машин [Загрузить](#)¹⁹

Образы контейнеров [Загрузить](#)²⁰

Боксы Vagrant [Перейти к списку](#)

¹⁸ <https://www.virtualbox.org/wiki/Downloads>

¹⁹ <https://registry.astralinux.ru/images/>

²⁰ <https://registry.astralinux.ru/browse/library>