

Поэтапная подготовка операционной системы

Если ваш дистрибутив не входит в список поддерживаемых программой предварительной настройки, вы можете выполнить ручную настройку системы, следуя шагам, описанным в данной инструкции.

Пакеты для оффлайн установки:

- [Debian.zip](#);
- [Ubuntu.zip](#);
- [Astra Linux.zip](#);
- [CentOS.zip](#);
- [Red OS.zip](#).

Настройка ОС

Ниже описаны шаги по подготовке окружения операционной системы, которые необходимо выполнить перед запуском инсталлятора wiSLA 5 CE.

1. Загрузка файлов на сервер

Загрузите архив [wisla-ce.tar](#) в домашний каталог текущего пользователя на вашем сервере с помощью утилиты `scp` или любым другим альтернативным способом, например с помощью SFTP.

Затем распакуйте архив `wisla-ce.tar` с инсталлятором и программой предварительной настройки:

```
tar -xvf wisla-ce.tar
```

После распаковки архива удалите файлы программы предварительной настройки узла, если данный способ установки вам не подходит по какой-либо причине.

```
rm -f preparing-os.start preparing.tar
```

2. Настройка имени узла

Для корректной работы инсталлятора wiSLA 5 CE необходимо скорректировать запись в файле `/etc/hosts`, обычно имя узла ссылается на `127.0.1.1` из-за чего может возникнуть ошибка.

Проверьте список сетевых адресов вашего сервера выполнив такую команду:

```
ip -o -4 addr show | grep -v '^.* lo ' | awk '{print $4}' | cut -d/ -f1
```

Скопируйте нужный IP-адрес и откройте файл `/etc/hosts` в привилегированном режиме с помощью любого текстового редактора.

```
sudo editor /etc/hosts
```

В открывшемся файле необходимо изменить запись в строке с именем сервера, в данном примере структура файла `/etc/hosts` выглядит следующим образом:

```
127.0.0.1    localhost
192.168.159.136 wisla

# The following lines are desirable for IPv6 capable hosts
::1    localhost ip6-localhost ip6-loopback
ff02::1 ip6-allnodes
ff02::2 ip6-allrouters
```

Здесь нас интересует данная строка:

```
192.168.159.136 wisla
```

- где `192.168.159.136` — сетевой адрес вашего сервера (IP-адрес), а `wisla` — имя узла (`hostname`).

Если вы решили изменить имя узла на другое не забудьте обновить запись в `/etc/hostname`!

Обратите внимание!

- ⚠ Если сетевые настройки на вашем сервере были получены по DHCP, в будущем могут возникнуть проблемы при изменении IP-адреса сервера, так как инсталлятор будет ссылаться на записи в файле `/etc/hosts`, который останется без изменений. Рекомендуется использовать статический IP-адрес на сетевом интерфейсе сервера.

- ✓ В файле `/etc/hosts` имя узла должно соответствовать IP-адресу, отличному от `127.0.0.1` и `127.0.1.1`.

3. Создание сервисной учётной записи

Если на вашем сервере присутствует только учётная запись суперпользователя `root`, тогда вам необходимо создать сервисную учётную запись для работы с системой. В данном примере будет создана учётная запись `wisla`:

```
sudo useradd -d /home/wisla -m wisla && sudo passwd wisla
```

В терминале сервера появится запрос на ввод пароля, задайте надёжный пароль для сервисной учётной записи.

4. Изменение привилегий для сервисной учётной записи

Чтобы все сервисы **WiSLA** работали корректно необходимо предоставить сервисной учётной записи привилегированный доступ без запроса пароля:

```
echo "wisla  ALL=(ALL:ALL) NOPASSWD:ALL" | sudo tee /etc/sudoers.d/wisla > /dev/null
```

Данное действие обязательно, иначе могут возникнуть проблемы из-за того, что группа, в которой состоит пользователь, не имеет `NOPASSWD` и будут унаследованы её права.

5. Создайте подкаталог `/opt/wisla5`

Для хранения файлов системы **wiSLA** необходимо создать подкаталог `/opt/wisla5`, сделать это можно такой командой:

```
sudo mkdir -p /opt/wisla5 && sudo chown -R wisla:wisla /opt/wisla5
```

6. Установка зависимостей

В зависимости от операционной системы на вашем сервере список необходимых пакетов может отличаться, нажмите на соответствующий блок для получения информации.

RedOS

С доступом к сети или внутреннему репозиторию:

```
sudo yum install ntp lzo dialog rsync uuid zip unzip wget tar python3 fontconfig curl pv uuid python3-paramiko
```

Без доступа к сети:

```
#Копируем архив на хост удобным способом
```

```
# Разархивируем
unzip redos.zip
cd redos
```

```
# Устанавливаем
```

```
#Все по очереди:
```

```
for i in $(ls *.rpm)
do
```

```
    rpm -i $i || exit
```

```
done
```

```
#Вручную:
```

```
rpm -i libtomcrypt-1.18.2-1.el7.x86_64.rpm libtommath-1.2.0-3.el7.x86_64.rpm dialog-1.3-14.20171209.el7.x86_64.rpm
```

```
pv-1.6.6-1.x86_64.rpm uuid-1.6.2-26.el7.x86_64.rpm
```

```
rpm -i python3-pynacl-1.5.0-1.el7.x86_64.rpm python3-bcrypt-3.2.2-1.el7.x86_64.rpm python3-paramiko-3.2.0-1.el7.noarch.rpm
```

CentOS

С доступом к сети:

```
sudo yum install ntp lzo dialog rsync uuid zip unzip wget tar python3 fontconfig curl
```

```
wget https://bootstrap.pypa.io/get-pip.py
```

```
python3 get-pip.py
```

```
python3 -m pip install --upgrade pip
pip install paramiko
sudo rpm -i http://www.ivarch.com/programs/rpms/pv-1.6.6-1.x86_64.rpm
```

Без доступа к сети:

```
#Копируем архив на хост удобным способом

# Разархивируем
unzip centos.zip
cd centos-pgks

# Устанавливаем
#Все по очереди:
for i in $(ls *.rpm)
do
    rpm -i $i || exit
done
#Вручную:
rpm -i wget-1.14-18.el7_6.1.x86_64.rpm
rpm -i uuid-1.6.2-26.el7.x86_64.rpm
rpm -i rsync-3.1.2-10.el7.x86_64.rpm
rpm -i python-crypto-2.6.1-1.el7.centos.x86_64.rpm
rpm -i python-paramiko-2.1.1-9.el7.noarch.rpm
rpm -i pv-1.6.6-1.x86_64.rpm
rpm -i dialog-1.2-5.20130523.el7.x86_64.rpm

# Возможно потребуется установить дополнительные пакеты из этого архива
```

Debian и Ubuntu

```
sudo apt install -y ntp pv liblz2-2 dialog rsync uuid zip unzip wget tar python3 python3-paramiko fontconfig curl language-pack-ru
```

Astra Linux 1.7

С доступом к сети:

```
sudo apt install -y ntp liblz2-2 dialog rsync zip unzip wget tar python3 python3-paramiko fontconfig curl
```

Обратите внимание!



В репозиториях **Astra Linux** нет пакетов `pv`, `paramiko` и `uuid`, поэтому их необходимо установить из исходников используя `.deb`-пакеты.

Следуйте приведённой ниже инструкции в секции "**Без доступа к сети**".

Без доступа к сети:

```
unzip astra.zip
cd astra-pkgs
#Все по очереди:
ls *.deb > /tmp/packages.list && sudo dpkg -i $(cat /tmp/packages.list) && rm -rf /tmp/packages.list
#Вручную:
sudo dpkg -i pv_1.6.6-1_amd64.deb
sudo dpkg -i rsync_3.1.3-6+ci202302061937+astra1_amd64.deb
sudo dpkg -i libossp-uuid16_1.6.2-1.5+b4_amd64.deb
sudo dpkg -i uuid_1.6.2-1.5+b4_amd64.deb
sudo dpkg -i wget_1.20.1-1.1_amd64.deb
sudo dpkg -i python3-paramiko_2.6.0-1~bpo10+1_all.deb
sudo dpkg -i python3-*
sudo dpkg -i dialog_1.3-20190211-1_amd64.deb
```

Astra Linux 1.8

С доступом к сети:

```
sudo apt install -y rsync libossp-uuid16 uuid wget python3 python3-paramiko dialog dialog fontconfig liblz2-2 python3-asn1crypto
python3-bcrypt python3-cffi-backend python3-cryptography python3-nacl rsync unzip zip
```

Обратите внимание!



В репозиториях **Astra Linux** нет пакета `pv`, поэтому его необходимо установить из исходников используя `.deb-пакет`.

Следуйте приведённой ниже инструкции в секции "**Без доступа к сети**".

Без доступа к сети:

```
unzip astra.zip
cd astra-pkgs
#Все по очереди:
ls *.deb > /tmp/packages.list && sudo dpkg -i $(cat /tmp/packages.list) && rm -rf /tmp/packages.list
#Вручную:
sudo dpkg -i pv_1.6.6-1_amd64.deb
sudo dpkg -i rsync_3.1.3-6+ci202302061937+astra1_amd64.deb
sudo dpkg -i libossp-uuid16_1.6.2-1.5+b4_amd64.deb
sudo dpkg -i uuid_1.6.2-1.5+b4_amd64.deb
sudo dpkg -i wget_1.20.1-1.1_amd64.deb
sudo dpkg -i python3-paramiko_2.6.0-1~bpo10+1_all.deb
sudo dpkg -i python3-*
sudo dpkg -i dialog_1.3-20190211-1_amd64.deb
```

Ручная установка пакета `pv`:

```
sudo dpkg -i pv_1.6.6-1_amd64.deb
```

Alt Linux (Simply Linux)

С доступом к сети:

```
sudo apt-get install -y ntp pv dialog rsync zip unzip wget tar python3 fontconfig curl python3-module-paramiko ossp-uuid
```

В ходе тестирования было замечено, что для установки **WiSLA** в **altLinux** необходимо подключаться по **ssh**, либо используйте `sudo su $(whoami)`, иначе будет возникать окно с авторизацией. Также необходимо увеличить размер `/tmp` в `/etc/fstab`, добавьте через запятую `size=4G` (**4G** указаны в качестве примера) и `mount -o remount, rw /tmp`.

7. Установить **python3** по умолчанию

Укажите системе использовать `python3` в качестве основной версии:

```
sudo update-alternatives --install /usr/bin/python python /usr/bin/python3 1
```

8. Настройка правил фильтрации

В зависимости от операционной системы на вашем сервере стандартные утилиты для управления сетевым фильтром будут отличаться, нажмите на соответствующий блок для получения информации.

Debian, Ubuntu и Astra Linux

Правила для UFW:

```
sudo ufw allow 8080/tcp
```

CentOS

Правила для firewalld:

```
sudo firewall-cmd --permanent --zone=public --add-port=8080/tcp
sudo firewall-cmd --reload
```

9. Настройка лимитов

Выполните команду ниже или создайте файл вручную как указано на примере ниже:

```
sudo tee /etc/security/limits.d/wisla > /dev/null << 'EOF'
wisla soft nofile 32768
wisla hard nofile 32768
```

```
wisla soft nproc 32768
wisla hard nproc 32768
EOF
```

10. Настройка локали

В зависимости от операционной системы на вашем сервере набор команд может отличаться, нажмите на соответствующий блок для получения информации.

Debian

Выполните команды приведённые ниже:

```
sudo locale-gen ru_RU.UTF-8
sudo dpkg-reconfigure locales
```

Затем заново авторизуйтесь на сервере.

Ubuntu

Выполните команды приведённые ниже:

```
sudo apt install -y locales
sudo sed -i 's|# ru_RU.UTF-8 UTF-8|ru_RU.UTF-8 UTF-8|g' /etc/locale.gen
sudo locale-gen ru_RU
sudo locale-gen ru_RU.UTF-8
sudo update-locale
localectl set-locale LANG=ru_RU.UTF-8
```

Затем заново авторизуйтесь на сервере.

CentOS

Выполните команды приведённые ниже:

```
sudo dnf install glibc-locale-source glibc-langpack-ru
sudo localectl set-locale LANG=ru_RU.UTF-8
```

Затем заново авторизуйтесь на сервере.

Обратите внимание!



Перед запуском программы установки следует выполнить команду `locale` и убедиться,

что активна `ru_RU.UTF-8`. При возникновении проблем необходимо обратиться к документации по дистрибутиву для установки нужной локали. Также следует проверить вывод `timedatectl`, часовой пояс должен иметь буквенное обозначение вместо `n/a`.

11. Настройка беспарольного подключения

Инсталлятор wiSLA использует протокол SSH для работы с компонентами и обновления конфигурации системы, потому важно настроить беспарольный доступ для ранее созданного пользователя `wisla`.

Переключитесь на ранее созданную сервисную учётную запись `wisla`.

```
su -l wisla
```

После переключения на учётную запись `wisla` проверьте, чтобы вы находились в домашнем каталоге `/home/wisla`.

```
pwd
```

Затем сгенерируйте SSH-ключ **без пароля** для пользователя `wisla`:

```
ssh-keygen -P ""
```

Сгенерировав SSH-ключ скопируйте его на ваш сервер используя утилиту `ssh-copy-id` как показано на примере ниже.

```
ssh-copy-id wisla@$(hostname)
ssh-copy-id wisla@localhost
```

Затем проверьте беспарольный доступ по SSH, если система запросит пароль удалите все ключи для пользователя `wisla` и повторите настройку заново.

```
# Проверьте работу аутентификации по ключам
ssh $username@$hostname exit
ssh $username@localhost exit
```

Информация

- ❗ Если система запрашивает пароль для подключения, тогда требуется провести анализ для установления точной причины данного поведения. Если данный шаг проигнорировать, тогда в процессе установки будут происходить многократные запросы пароля. Возможной причиной может быть неразрешённый тип ключа или несоответствие имени сервера в `/etc/hosts` и записью в `/etc/hostname`.

12. Дополнительные настройки для Astra Linux и RHEL-подобных систем

Astra Linux

1. Настройка автозагрузки wiSLA 5 CE

В некоторых сценариях Astra Linux не создаёт systemd-юнит для автозапуска wiSLA, потому его необходимо создать вручную:


```
sudo tee /etc/systemd/system/wisla.service > /dev/null << 'EOF'
[Unit]
Description=Starts JBoss process with wiSLA 5 system
After=network-online.target
Requires=network-online.target

[Service]
Type=forking
RemainAfterExit=true
WorkingDirectory=/opt/wisla5
ExecStart=/opt/wisla5/scripts/wisla5.sh start
ExecStop=/opt/wisla5/scripts/wisla5.sh stop

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

Затем перечитайте конфигурацию демонов и добавьте `wisla.service` в автозагрузку:

```
sudo systemctl daemon-reload && sudo systemctl enable wisla.service
```

2. Отключение опции *KillUserProcesses*

Чтобы система не убивала процессы системы мониторинга wiSLA убедитесь, что опция ***KillUserProcesses*** выключена.

Отключить данную опцию можно выполнив такую команду:

```
sudo sed -i 's/#KillUserProcesses=yes/KillUserProcesses=no/g' /etc/systemd/logind.conf
```

Чтобы применить изменения выполните перезагрузку сервера:

```
sudo reboot
```

После перезагрузки сервера повторно подключитесь к нему по SSH, чтобы продолжить настройку.

RHEL-подобные дистрибутивы

Чтобы система не блокировала возможность записи в файлы журналов и не ломала работу файловой системы `HDFS` обязательно переключите ***SELinux*** в режим `Permissive` или отключите его полностью.

Откройте файл конфигурации в любом текстовом редакторе:

```
sudo editor /etc/selinux/config
```

И задайте нужное значение в строке с опцией `SELINUX`:

```
SELINUX=permissive
```

Чтобы не перезагружать сервер временно отключите SELinux:

```
sudo setenforce 0
```

13. Запуск инсталлятора wiSLA 5 CE

Переместите файл инсталлятора `wisla-5.2.10-2505210735.run` из домашнего каталога пользователя, под которым производилась загрузка архива `wisla-ce.tar` в каталог сервисной учётной записи `wisla`.

Например, если вы загрузили файлы на сервер под учётной записью пользователя `myuser`, тогда вам необходимо выполнить такую команду:

```
sudo mv ../myuser/wisla-5.2.10-2505210735.run /home/wisla
```

Дождитесь завершения перемещения инсталлятора, затем добавьте права на исполнение:

```
sudo chmod +x ./wisla-5.2.10-2505210735.run
```

Далее можно запустить инсталлятор wiSLA 5 CE и перейти к процессу установки системы.
