

# **Программный обеспечение «Нестор»**

Инструкция по установке

Санкт-Петербург, 2024

## **Введение.**

Настоящий документ содержит инструкцию по развертыванию программного обеспечения «Нестор» («NeStore») (далее, Платформа) под операционной системой Astra Linux SE 1.7.5. в конфигурации CE.

## Содержание

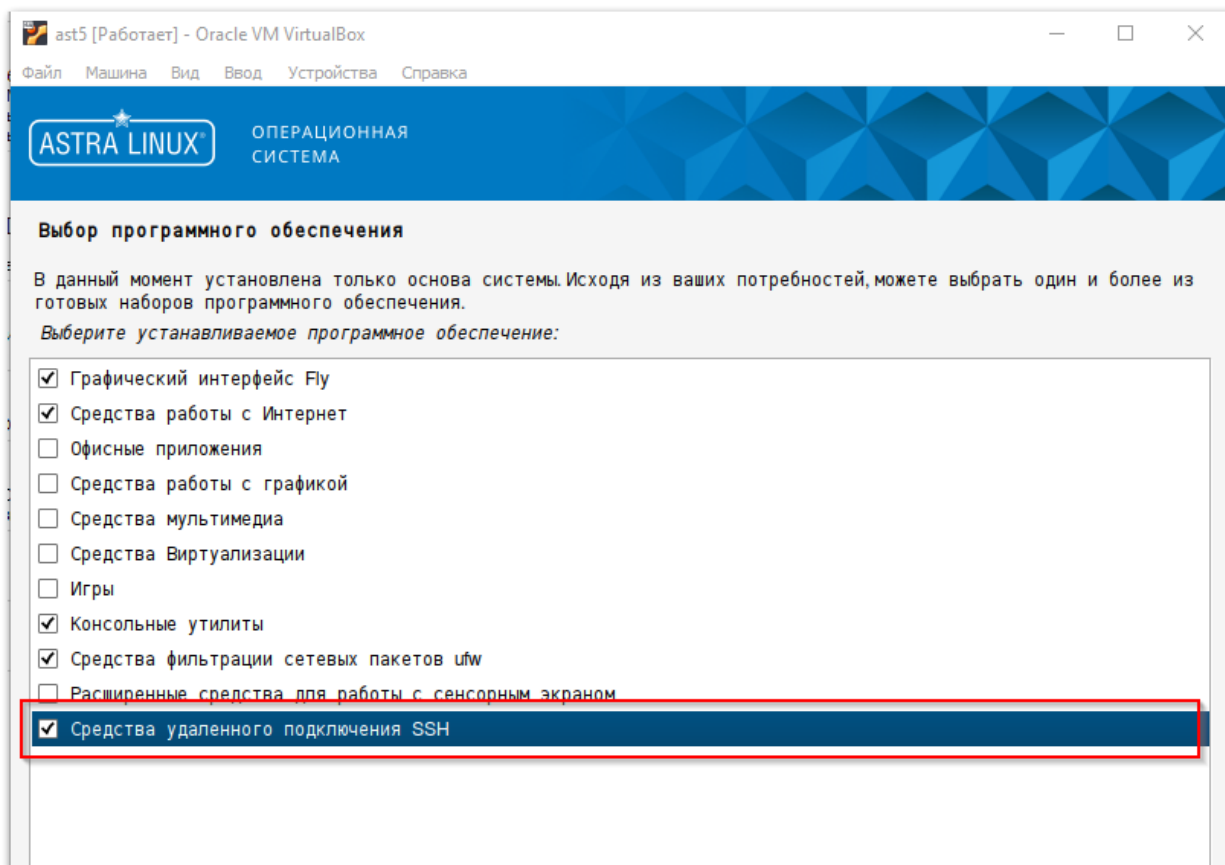
1. Установка Платформы на Astra Linux SE 1.7.5 .....	4
1.1. Общие условия успешной установки .....	4
1.2. Создание пользователя и пар SSH ключей .....	5
2. Процедура инсталляции .....	5
2.1. Установка Docker .....	5
2.2. Подключение стандартного внешнего репозитория «Debian».....	6
2.3. Добавление ключей для верификации внешнего репозитория .....	6
2.4. Установка пакета «software-properties-common».....	6
2.5. Добавление ключа для «uq» репозитория .....	6
2.6. Установка jq .....	7
2.7. Конфигурирование параметров прокси для docker .....	7
2.8. Перезапуск сервисов .....	7
2.9. Клонирование основного репозитория проекта .....	7
2.10. Установка поддержки языка Go .....	7
2.11. Компиляция утилиты stool .....	8
2.12. Развертывание SE кластера .....	8
3. Проверка успешности развертывания SE кластера .....	8

# 1. Установка Платформы на Astra Linux SE 1.7.5

## 1.1. Общие условия успешной установки

Для успешной установки Платформы следует использовать «чистую» операционную систему Astra Linux Special Edition, версии 1.7.5.

В процессе инсталляции, на этапе выбора программного обеспечения, следует отметить к установке опцию «Средства удаленного доступа SSH».



## 1.2 Настройка прокси-сервера

Если для доступа в интернет используется прокси-сервер то, необходимо произвести следующие настройки (адрес прокси сервера в тексте документа указан произвольный. Его необходимо заменить на используемый в конкретных условиях):

### Установка глобальных переменных окружения

Для установки глобальных переменных окружения необходимо в файл «/etc/environment» добавить следующие строки:

```
https_proxy="https://proxy.office.sigma-soft.ru:8080/"
http_proxy="http://proxy.office.sigma-soft.ru:8080/"
ftp_proxy="ftp://proxy.office.sigma-soft.ru:8080/"
socks_proxy="socks://proxy.office.sigma-soft.ru:8080/"
```

```
no_proxy="127.0.0.1, localhost"
```

В переменную «no\_proxy» также необходимо прописать и IP-адрес локальной машины (его можно узнать командой «hostname -i»).

### **Настройка работы менеджера пакетов APT через прокси**

Для возможности работы менеджера пакетов APT через прокси необходимо в файле /etc/apt/apt.conf указать:

```
Acquire::http::proxy "http://proxy.office.sigma-soft.ru:8080/";  
Acquire::https::proxy "http://proxy.office.sigma-soft.ru:8080/";  
Acquire::ftp::proxy "http://proxy.office.sigma-soft.ru:8080/";  
Acquire::socks::proxy "http://proxy.office.sigma-soft.ru:8080/";  
Acquire:::Proxy "true";
```

### **Создание алиаса для возможности компиляции программы на Go**

Создайте алиас следующей командой:

```
alias go='http_proxy=http://proxy.office.sigma-soft.ru:8080/  
https_proxy=http://proxy.office.sigma-soft.ru:8080/ no_proxy=localhost,127.0.0.0/8,::1 go'
```

## **1.2. Создание пользователя и пар SSH ключей**

Создайте пользователя в ОС Astra Linux Special Edition (здесь и ниже в документации будем считать, что пользователь в системе создан и имеет имя «USER»).

Создайте для пользователя пару SSH ключей командой

```
ssh-keygen
```

## **2. Процедура инсталляции**

### **2.1. Установка Docker**

Выполните обновление пакетов командой:

```
sudo apt update
```

В Astra Linux Docker представлен пакетом docker.io и его следует установить командой:

```
sudo apt install docker.io
```

Установка должна выполняться от имени пользователя, являющегося администратором системы. После установки Docker необходимо предоставить администратору право работать с контейнерами не используя sudo. Для этого пользователя нужно включить в группу docker:

```
sudo usermod -aG docker $USER
```

Далее, необходимо установить docker-compose командой:

```
sudo apt install docker-compose
```

Необходимо обеспечить автоматический запуск служб командами:

```
sudo systemctl enable docker.service  
sudo systemctl enable containerd.service
```

## 2.2. Подключение стандартного внешнего репозитория «Debian»

Далее, следует добавить стандартный внешний репозиторий «Debian».

Для этого откройте на редактирование файл sources.list командой:

```
sudo apt edit-sources
```

и добавьте следующую строку в файл:

```
deb https://deb.debian.org/debian/ buster main contrib non-free
```

## 2.3. Добавление ключей для верификации внешнего репозитория

Для добавления ключей верификации внешнего репозитория необходимо выполнить команду:

```
sudo apt-key adv --keyserver keyserver.ubuntu.com --keyserver-options http-  
proxy=http://proxy.office.sigma-soft.ru:8080 --recv-keys 648ACFD622F3D138  
0E98404D386FA1D9 DCC9EFBF77E11517
```

В строке команды необходимо указать адрес прокси сервера (при необходимости).

## 2.4. Установка пакета «software-properties-common»

Пакет «software-properties-common» необходим для обеспечения абстракции используемых репозиториях apt. Он предоставляет скрипты для удаления и добавления персональных архивов пакетов (PPA).

Выполните команды:

```
sudo apt update  
sudo apt install software-properties-common
```

## 2.5. Добавление ключа для «uq» репозитория

Следующая команда обеспечит добавление ключа для «uq» репозитория:

```
sudo apt-key adv --keyserver keyserver.ubuntu.com --keyserver-options http-  
proxy=http://proxy.office.sigma-soft.ru:8080 --recv-keys  
9A2D61F6BB03CED7522B8E7D6657DBE0CC86BB64
```

```
sudo add-apt-repository 'deb
https://ppa.launchpadcontent.net/rmescandon/yq/ubuntu jammy main'
```

## 2.6. Установка jq

Для установки jq введите следующую команду:

```
sudo apt-get install jq
```

## 2.7. Конфигурирование параметров прокси для docker

Для конфигурирования параметров прокси необходимо создать каталог `docker.service.d` командой:

```
sudo mkdir -p /etc/systemd/system/docker.service.d
```

в каталоге создать файл `http-proxy.conf` и открыть его в редакторе командой:

```
sudo nano /etc/systemd/system/docker.service.d/http-proxy.conf
```

ввести следующий текст:

```
[Service]
Environment="HTTP_PROXY=http://proxy.office.sigma-soft.ru:8080"
Environment="HTTPS_PROXY=http://proxy.office.sigma-soft.ru:8080"
Environment="NO_PROXY=localhost,127.0.0.1,::1"
```

и сохранить файл.

## 2.8. Перезапуск сервисов

Далее, необходимо перезапустить сервисы командами:

```
sudo systemctl daemon-reload
```

```
sudo systemctl restart docker
```

## 2.9. Клонирование основного репозитория проекта

Клонирование основного репозитория проекта осуществляется командой:

```
git clone https://gitverse.ru/sigma-soft/voedger
```

## 2.10. Установка поддержки языка Go

Установка поддержки языка Go осуществляется командой:

```
curl -x http://proxy.office.sigma-soft.ru:8080 -L https://git.io/vQhTU | bash -s -- --
version 1.22.1
```

Выполните команду:

```
source /home/idd/.bashrc
```

## 2.11. Компиляция утилиты ctool

Конфигурируем git для работы через прокси

```
git config --global http.proxy http://proxy.office.sigma-soft.ru:8080
```

Компиляцию утилиты ctool необходимо выполнить командами:

```
cd ~/voedger/cmd/ctool
```

```
go build -o ctool
```

## 2.12. Развертывание CE кластера

Развертывание CE кластера выполняется командой:

```
./ctool init CE <адрес хоста>
```

Работа утилиты должна завершиться выводом сообщения:  
«CE cluster is deployed successfully»

## 3. Проверка успешности развертывания CE кластера

Проверка корректности функционирования Платформы осуществляется обращением с http запросом по адресу `http://host/static/sys/monitor/site/hello`, где «host» - IP-адрес хоста, где была развернута Платформа.

Получение кода возврата «200» говорит об успешности функционирования Платформы.

Можно проконтролировать состав и состояние работающих контейнеров в соответствии с рисунком:

voedger_alertmanager_1	running
voedger_cadvisor_1	healthy
voedger_grafana_1	running
voedger_node-exporter_1	running
voedger_prometheus_1	running
voedger_scylla_1	healthy
voedger_voedger_1	running