

ИНСТРУКЦИЯ ПО ПОДГОТОВКЕ СЕРВЕРА К РАБОТЕ (через SSH)

1. Введение и требования

1.1. Назначение документа

Настоящий документ содержит пошаговое руководство по развертыванию и настройке серверной инфраструктуры для работы системы на базе микросервисной архитектуры. Инструкция предназначена для системных администраторов и DevOps-инженеров.

1.2. Минимальные системные требования

Параметр	Требование
Операционная система	Linux
Дистрибутив	Debian 12
Оперативная память	16 Gb
Виртуальные ЦП	2
Сокеты	1
Кол-во ядер на сокет	1
Потоков на ядро	2

1.3. Условные обозначения

Обозначение	Описание
\$	Команда выполняется от обычного пользователя
#	Команда выполняется от суперпользователя (root)

2. Первоначальная настройка сервера

2.1. Подключение к серверу

Для подключения потребуется протокол SSH:

```
$ ssh <username>@<host> -p <port>
```

<username> — имя пользователя.

<host> — адрес сервера.

<port> — порт SSH-сервера.

2.2. Получение прав суперпользователя

После успешного подключения переключитесь на пользователя root для выполнения административных задач:

```
$ sudo su
```

2.3. Предварительная подготовка системы

Обновите репозитории, установите обновления и создайте базовые директории для хранения конфигураций и файлов Docker:

```
# apt update
```

```
# apt upgrade -y
# mkdir /opt/docker /opt/configs
```

2.4. Установка дополнительных пакетов

Установите утилиты для диагностики сети и мониторинга системы:

```
# apt install -y net-tools htop
```

3. Установка базового программного обеспечения

3.1. Установка Nginx

Установите веб-сервер для проксирования HTTP-запросов к сервисам и раздачи статики:

```
# apt install -y nginx
```

3.2. Установка Docker и Docker Compose

Docker является основой для запуска всех сервисов системы.

1. Установите зависимости и добавьте официальный GPG-ключ Docker:

```
# apt install -y ca-certificates curl
# install -m 0755 -d /etc/apt/keyrings
# curl -fsSL https://download.docker.com/linux/debian/gpg -o
/etc/apt/keyrings/docker.asc
# chmod a+r /etc/apt/keyrings/docker.asc
```
2. Добавьте репозиторий Docker:

```
# echo "deb [arch=$(dpkg --print-architecture) signed-by=/etc/apt/keyrings/docker.asc]
https://download.docker.com/linux/debian $(. /etc/os-release && echo
"$VERSION_CODENAME") stable" | sudo tee /etc/apt/sources.list.d/docker.list >
/dev/null
```
3. Установите пакеты Docker и плагин Compose:

```
# apt update
# apt install -y docker-ce docker-ce-cli containerd.io docker-buildx-plugin docker-
compose-plugin
```
4. Добавьте пользователя в группу docker (замените USERNAME на ваше имя пользователя):

```
# usermod -aG docker USERNAME
```

4. Настройка Docker для работы с внешним репозиторием

Если используется приватный репозиторий, необходимо добавить его в список доверенных.

1. Отредактируйте файл конфигурации daemon Docker:

```
# nano /etc/docker/daemon.json
```

Добавьте следующее содержимое (замените host:port на адрес вашего репозитория):

```
{  
  "insecure-registries": ["http://host:port"]  
}
```

2. Перезапустите Docker и выполните логин в репозиторий:

```
# systemctl restart docker  
# docker login http://host:port
```

5. Установка вспомогательных сервисов

Для установки сервисов используются файлы docker-compose.yml. Предполагается, что они уже подготовлены и скопированы в указанные директории.

5.1. Установка Grafana и Loki

```
# mkdir /opt/docker/grafana_loki  
# cd /opt/docker/grafana_loki  
# docker compose up -d
```

5.2. Установка PostgreSQL

Остановите системную службу, если она активна, и запустите контейнер:

```
# apt install -y postgresql-contrib  
# systemctl stop postgresql  
# systemctl disable postgresql  
# mkdir /opt/docker/postgresql  
# cd /opt/docker/postgresql  
# docker compose up -d
```

После запуска создайте базу данных platform:

```
# psql -U postgres -W -h localhost -p 5432 -c 'create database platform;'
```

5.3. Установка Redis

```
# mkdir /opt/docker/redis  
# cd /opt/docker/redis  
# docker compose up -d
```

5.4. Установка MinIO

```
# mkdir /opt/docker/minio  
# cd /opt/docker/minio  
# docker compose up -d
```

6. Установка микросервисов системы

6.1. Подготовка конфигурации

Создайте директории для файлов настроек (application-telros.properties) каждого сервиса:

```
# mkdir /opt/configs/Gateway-Server /opt/configs/Auth-Service /opt/configs/Company-Service /opt/configs/File-Service /opt/configs/Geo-Service /opt/configs/Itv-Service /opt/configs/Log-Service /opt/configs/Mail-Service /opt/configs/Monitor-Service /opt/configs/Faq-Service
```

Скопируйте подготовленные файлы конфигурации в соответствующие папки /opt/configs/.

6.2. Развертывание сервисов

Создайте директории для Compose-файлов и запустите сервисы в порядке, указанном в Таблице 3 (см. Приложение).

```
# mkdir /opt/docker/discovery /opt/docker/gateway /opt/docker/config /opt/docker/auth /opt/docker/monitor /opt/docker/log /opt/docker/mail /opt/docker/file /opt/docker/company /opt/docker/itv /opt/docker/geo /opt/docker/faq
```

Скопируйте файлы docker-compose.yml в соответствующие папки /opt/docker/.

Пример запуска для Discovery-Server:

```
# cd /opt/docker/discovery
# docker compose up -d
```

Повторите эту операцию для остальных сервисов.

7. Настройка Nginx (Web-сервер)

Настройте виртуальный хост для проксирования запросов к бэкенду и раздачи фронтенда.

1. Отредактируйте конфигурацию Nginx:

```
# nano /etc/nginx/nginx.conf
```

(Либо создайте отдельный файл в /etc/nginx/sites-available/ и создайте симлинк).

2. Пример конфигурации (замените application-host.ru на ваш домен):

```
server {
    listen 80;
    server_name application-host.ru;

    gzip on;
    gzip_types text/plain text/css application/javascript application/json image/x-icon
application/octet-stream application/wasm;
    gzip_vary on;
    gzip_proxied no-cache no-store private expired auth;
    gzip_min_length 512;
    client_max_body_size 50m;

    # Раздача статики фронтенда
    location / {
```

```

    root /var/www/APPLICATION/public;
    index index.html index.htm;
    try_files $uri $uri/ /index.html @rsv;
}

# Проксирование на Gateway
location /server/ {
    proxy_pass http://localhost:21003/;
    proxy_set_header X-Real-IP $remote_addr;
    proxy_set_header X-Forwarded-Host $host;
    proxy_set_header X-Forwarded-Port $server_port;
    proxy_set_header Host $host;
    proxy_set_header Connection "Upgrade";
    proxy_set_header Upgrade $http_upgrade;
}

```

3. Перезагрузите конфигурацию Nginx:
nginx -s reload

8. Установка Web-приложения (Фронтенд)

1. Скопируйте папку с собранным веб-приложением в директорию, указанную в конфигурации Nginx:
cp -R /path/to/web/application/FOLDER /var/www/APPLICATION/public/
2. Отредактируйте файл конфигурации фронтенда, чтобы указать адрес вашего сервера:
nano /var/www/APPLICATION/public/config/1_main.js

Замените значения:

```

ENV_DECLARE_CONSTANT('MICROSERVICES_GATE', 'http://your-
domain.com');
ENV_DECLARE_CONSTANT('MICROSERVICES_GATE_WS', 'ws://your-
domain.com');
ENV_DECLARE_CONSTANT("SKADA_URL", "https://skada_url");

```

9. Управление Docker-контейнерами и образами

1. Просмотр списка контейнеров и образов:
docker ps
docker image ls
2. Перезапуск контейнера:
docker restart CONTAINER-NAME
3. Удаление контейнера и образа:
docker stop CONTAINER-NAME
docker rm CONTAINER-NAME
docker rmi repo.telros.ru:8083/microservices/services/APPLICATION:latest

4. Загрузка образа из файла .tar:
docker load -i APPLICATION.tar
После загрузки выполните docker compose up -d в директории с Compose-файлом сервиса.
5. Просмотр логов:
docker logs CONTAINER-NAME
Для сохранения логов в файл:
docker logs CONTAINER-NAME > ./FILENAME.log 2>&1
6. Просмотр информации о контейнере:
docker inspect CONTAINER_ID

10. Пример автоматизации обновления (Shell-скрипт)

Для упрощения обновления микросервисов рекомендуется использовать скрипты.

Пример скрипта обновления auth-service:

```
#!/bin/bash
RED='\033[0;31m'
GREEN='\033[0;32m'
YELLOW='\033[0;33m'
BLUE='\033[0;34m'
NC='\033[0m'

if [ "$EUID" -ne 0 ]; then
    echo -e "${RED}[$NC] Use: ${BLUE}sudo su$NC for root access"
    exit
fi

echo -e "${BLUE}[*]$NC Updating Auth-Service"
cd /home/telros/psim || { echo -e "${RED}[$NC] Directory not found"; exit 1; }

# Stop, remove container and image
docker stop auth-service 2>/dev/null
docker rm auth-service 2>/dev/null
docker rmi repo.telros.ru:8083/microservices/services/auth-service:latest 2>/dev/null

# Load new image
docker load -i ./auth.tar || { echo -e "${RED}[$NC] Error loading image"; exit 1; }
rm ./auth.tar

# Start container
cd /opt/composes/Auth || { echo -e "${RED}[$NC] Directory not found"; exit 1; }
docker compose up -d

Запуск скрипта:
# cd /home/USERNAME/shell
```

```
# chmod 755 ./FILENAME.sh
# ./FILENAME.sh
```

Приложение А: Порядок запуска приложений

Таблица 3 — Последовательность развертывания сервисов

Порядок	Приложение	Порт(ы)
<i>Вспомогательное ПО</i>		
1	Grafana + Loki	3000, 3100
2	PostgreSQL	5432
3	Redis	16379
4	MinIO	9000
<i>Микросервисы</i>		
5	Discovery-Server	21001
6	Config-Server	21002
7	Gateway-Server	21003
8	Auth-Service	21004
9	Monitor-Service	21005
10	Log-Service	21006
11	File-Service	21007
12	Mail-Service	21008
13	Itv-Service	21009
14	Geo-Service	21010
15	Company-Service	21011
16	Faq-Service	21012
17	Auth-Service (перезапуск)	21004