

Инструкция по установке АСУМ «Алмаз»

1. Подготовка окружения

Требования:

- Процессор с поддержкой **AVX**
- ОС **Astra Linux 1.7.7.9 SE** и выше
- Установленный **Docker Engine**
- Установленный **Docker Compose 2.x**

2. Настройка сети

1. Настройте статическую IPv4-конфигурацию через NetworkManager:

- **IP-адрес:** 192.168.1.60
- **Маска подсети:** 255.255.255.0
- **Шлюз по умолчанию:** 192.168.1.1
- **DNS-серверы:** 10.0.2.3 , 8.8.8.8

Эти параметры используются в примерах далее. Замените параметры на актуальные в вашей сети.

2. Добавьте запись с IP-адресом и именем сервера в `/etc/hosts` :

```
sudo nano /etc/hosts
```

Пример:

```
192.168.1.60      server1
```

3. Подготовка структуры каталогов и файлов

1. Создайте базовую структуру каталогов для АСУМ Алмаз:

```
sudo mkdir -p /opt/ncw
```

2. Распакуйте архив с установочными файлами АСУМ Алмаз в созданную директорию.

Пример:

```
sudo tar -xzf ~/Загрузки/asum_almaz_v1.5sa_installer.tgz -C /opt/ncw
```

3. Создайте директорию для хранения данных MongoDB:

```
sudo mkdir -p /opt/ncw/db
```

4. Создайте ключ аутентификации для MongoDB и настройте корректные права доступа:

```
openssl rand -base64 768 | sudo tee /opt/ncw/mongo/mongokey > /dev/null
sudo chown 999:999 /opt/ncw/mongo/mongokey
sudo chmod 400 /opt/ncw/mongo/mongokey
```

5. Сгенерируйте самоподписанный SSL-сертификат и приватный ключ для Nginx:

```
sudo mkdir -p /opt/ncw/nginx/cert
sudo openssl req -x509 -nodes -days 1095 -newkey rsa:2048 \
  -keyout /opt/ncw/nginx/cert/private.key \
  -out /opt/ncw/nginx/cert/cert.crt
```

6. Убедитесь, что файлы и подкаталоги в директории `/opt/ncw` имеют следующую файловую структуру:

```
ncw
├─ config.json # конфиг-файл nsw
├─ docker-compose.yaml # файл-описание запуска стека контейнеров АСУМ Алмаз
├─ db # раздел в котором будет храниться БД
├─ mongo # раздел с конфигурационным файлом и ключем для mongodb
│   ├── mongo.conf
│   └─ mongokey
└─ nginx # раздел с конфигурацией и сертификатами для nginx
    ├── cert
    │   ├── cert.crt
    │   └─ private.key
    └─ nginx.conf
```

4. Настройка Docker

1. Создайте или отредактируйте конфигурационный файл Docker:

```
sudo mkdir -p /etc/docker
sudo nano /etc/docker/daemon.json
```

2. Добавьте следующую конфигурацию:

```
{
  "insecure-registries": ["192.168.29.100:5000"], // Разрешает подключение к
небезопасному (HTTP) репозиторию Docker
  "log-driver": "local", //Драйвер логирования – использует локальное
хранилище с ротацией
  "log-opts": { // Опции ротации логов контейнеров
    "max-size": "100m", // Максимальный размер одного файла лога
    "max-file": "5" // Максимальное количество хранимых файлов логов (на
каждый контейнер)
  }
}
```

Эти пояснения приведены для ознакомления. При использовании на сервере — удалите строки с комментариями, чтобы избежать ошибок парсинга JSON.

5. Доставка Docker-образов

1. Доставьте Docker-образы из архива `asum_almaz_1.5sa_containers.tgz`:

```
sudo docker load -i asum_almaz_1.5sa_containers.tgz
```

Архив содержит все необходимые образы: `mongo:5`, `nginx:latest`, `ncw`.

2. После доставки проверьте наличие образов:

```
sudo docker images
```

Важно: Следующие пункты выполняются если при доставке образов возникают ошибки, связанные с политиками безопасности (например: `permission denied`, `access denied`).

3. Временно добавьте в `daemon.json` параметр:

```
"astra-sec-level": 6
```

Пример итогового `daemon.json`:

```
{
  "insecure-registries": ["192.168.29.100:5000"],
  "log-driver": "local",
  "log-opts": {
    "max-size": "100m",
    "max-file": "5"
  },
  "astra-sec-level": 6
}
```

4. Перезапустите Docker и повторно доставьте Docker-образы из архива `asum_almaz_1.5sa_containers.tgz`:

```
sudo systemctl restart docker
sudo docker load -i asum_almaz_1.5sa_containers.tgz
```

5. После доставки проверьте наличие образов:

```
sudo docker images
```

6. Удалите параметр `"astra-sec-level": 6` после установки.

6. Подготовка `config.json` и `docker-compose.yml`

1. Откройте конфигурационный файл АСУМ АЛМАЗ для редактирования:

```
sudo nano /opt/ncw/config.json
```

2. Внесите в `config.json` актуальные значения: имя БД, IP-адрес сервера для подключения к MongoDB и адрес SWM

Пример содержимого:

```
{
  "database": {
    "type": "mongodb",
```

```

    "dbname": "ncw",
    "uri": "mongodb://dbadmin:dbadmin@server1:27017/?
replicaSet=ncwrs&authSource=admin", // адрес подключения к БД
    "topology_table": "topo",
    "topology_img_table": "topoimg",
    "websess_table": "websess",
    "webpages_table": "webpages",
    "websse_table": "websse",
    "inventory_table": "inv",
    "config_table": "conf",
    "logging_table": "logs",
    "pm_table": "pm",
    "auth_table": "auth",
    "ncw_table": "ncw",
    "schema_table": "schema",
    "tasks_table": "tasks",
    "archive_table": "archive",
    "scheduler_table": "scheduler"
  },
  "http": {
    "listen_ip": "0.0.0.0",
    "listen_port": "8080",
    "urlprefix": "http://server1:8080", // адрес для SWM
    "timeout": 1000000,
    "max_queued": 100,
    "session_timeout": 1800,
    "max_keepalive_requests": 0,
    "max_threads": 200,
    "web_static_path": "www/static",
    "web_dynamic_path": "www/dynamic"
  }
}

```

3. Обновите секцию **extra_hosts** в файле **docker-compose.yaml**, указав актуальное значение, соответствующее конфигурации хоста в **/etc/hosts**

Пример содержимого:

```

ncw:
  container_name: ncw
  image: ncw:v1.5sa
  volumes:
    - ./config.json:/ncw/config.json:ro
  extra_hosts:
    - "server1:192.168.1.60" #указать данные из /etc/hosts
  depends_on:
    mongo:
      restart: true

```

```
condition: service_healthy
restart: unless-stopped
```

7. Настройка MongoDB

1. Запустите контейнер MongoDB:

```
sudo docker-compose up -d mongo
```

2. Подключитесь к MongoDB и создайте пользователя:

```
sudo docker exec -it mongo mongosh
```

Внутри MongoDB:

```
use admin
db.createUser({user: "dbadmin", pwd: "dbadmin", roles: [{role: "root", db:
"admin"}]})
db.auth("dbadmin", "dbadmin")
```

Если возвращается 1 — пользователь создан успешно

8. Запуск Replicaset

1. Отредактируйте файл **mongo.conf**, раскомментировав строки настройки репликации:

```
replication:
  replSetName: ncwrs
```

2. Перезапустите контейнер **mongo**, выполнив команду:

```
sudo docker compose down -v
sudo docker compose up mongo -d
```

3. Подключитесь к MongoDB

```
sudo docker exec -it mongo mongosh
```

4. Перейдите в административную БД и авторизуйтесь:

```
use admin
db.auth("login", "password")
```

5. Инициализируйте replicaset в standalone:

```
rs.initiate({_id : "ncwrs", members:[{ _id: 0, host: "server1:27017" }]}))
```

6. Проверьте конфигурацию репликасета:

```
rs.conf()
```

9. Запуск АСУМ Алмаз

1. Выполните следующие команды для запуска всех сервисов в фоновом режиме:

```
sudo docker-compose down -v
sudo docker-compose up -d
sudo docker ps
```

2. Если в выводе `docker ps` контейнер `ncw` отображается как `(unhealthy)` и в логах наблюдаются ошибки подключения к MongoDB, выполните следующие действия:

```
sudo reboot
sudo docker-compose down -v
sudo docker-compose up -d
sudo docker ps
```

9. Проверка через браузер

1. Откройте:

```
https://192.168.1.60/login
```

2. Введите логин/пароль по умолчанию: `admin/admin`