

ООО «Астра Консалтинг»

**Астра ИС МД (Инфраструктурные Сервисы)**

**Модуль Астра ИС МД**

**Модуль Astra Redis**

**ИНСТРУКЦИЯ ПО УСТАНОВКЕ ЭКЗЕМПЛЯРА  
ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ, ПРЕДОСТАВЛЕННОГО ДЛЯ  
ПРОВЕДЕНИЯ ЭКСПЕРТНОЙ ПРОВЕРКИ**

Москва, 2026

# СОДЕРЖАНИЕ

<b>1. Область применения .....</b>	<b>3</b>
<b>2. Установка системы.....</b>	<b>3</b>
<b>2.1. Технические требования к оборудованию .....</b>	<b>3</b>
<b>2.2. Минимальные требования к сторонним компонентам .....</b>	<b>3</b>
<b>3. Архитектура системы.....</b>	<b>4</b>
<b>3.1. Характеристики системы.....</b>	<b>4</b>
<b>3.2. Структура docker-compose .....</b>	<b>4</b>
<b>4. Установка и запуск.....</b>	<b>5</b>
<b>4.1. Шаг 1: Подготовка директории.....</b>	<b>5</b>
<b>4.2. Шаг 2: Создание необходимых файлов.....</b>	<b>5</b>
<b>4.3. Шаг 3: Запуск контейнеров .....</b>	<b>6</b>
<b>4.4. Шаг 4: Проверка статуса.....</b>	<b>6</b>
<b>5. Доступ к сервисам.....</b>	<b>6</b>
<b>5.1. Точки доступа .....</b>	<b>6</b>
<b>5.2. Доступ к RedisInsight .....</b>	<b>7</b>
<b>5.3. Подключение к Redis .....</b>	<b>7</b>
<b>6. Проверка работы .....</b>	<b>7</b>
<b>6.1. Базовая проверка .....</b>	<b>7</b>
<b>6.2. Тестирование записи и чтения.....</b>	<b>7</b>
<b>6.3. Проверка статистики.....</b>	<b>8</b>
<b>7. Управление контейнерами.....</b>	<b>8</b>
<b>7.1. Основные команды .....</b>	<b>8</b>
<b>7.2. Управление Redis .....</b>	<b>8</b>
<b>8. Просмотр логов работы системы .....</b>	<b>9</b>
<b>8.1. Логи Redis.....</b>	<b>9</b>
<b>8.2. Логи всех сервисов.....</b>	<b>9</b>

## 1. Область применения

Настоящий документ предназначен для установки и запуска модуля Астра ИС МД и системы управления базами данных Модуль Astra Redis, развернутой через docker-compose с использованием системы контейнерной изоляции приложений Docker.

Данный docker-compose файл разворачивает следующие сервисы:

- redis — сервер базы данных Astra Redis
  - redisinsight — web-интерфейс для управления Astra Redis
  - redisinsight-proxu — nginx-прокси с HTTP-аутентификацией для RedisInsight
  - web — nginx web-сервер
- 

## 2. Установка системы

### 2.1. Технические требования к оборудованию

Для локальной инсталляции требуется виртуальный сервер (ВС) с минимальными техническими характеристиками, представленными в Таблица 1.

**Таблица 1** — Минимальные требования к ВС

Вид ресурса	Ед. измерения	Требуемый объем
Частота процессора	ГГц	2.3
vCPU количество процессоров и ядер	шт.	2
Объем оперативной памяти	Гб	4
Объем жесткого диска	Гб	20
Сетевой интерфейс	шт.	1
Скорость сетевого интерфейса	Гб/с	1

### 2.2. Минимальные требования к сторонним компонентам

Для установки Системы на стенде используется система контейнерной изоляции приложений Docker.

#### Требуемые компоненты:

- Docker Engine версии 20.10+
- Docker Compose версии 2.0+

Установка системы контейнерной изоляции приложений Docker осуществляется по инструкции: <https://docs.docker.com/engine/install/>

Для Ubuntu/Debian:

```
sudo apt update
sudo apt install docker.io docker-compose -y
sudo systemctl start docker
sudo systemctl enable docker
```

---

## 3. Архитектура системы

### 3.1. Характеристики системы

Таблица 2 — Характеристики демонстрационного стенда

Параметр	Значение
Redis версия	7 (alpine)
Порт	6379
Аутентификация	Включена
Пароль	mysecretpassword
Персистентность	AOF (Append Only File)
Сетевой драйвер	bridge

### 3.2. Структура docker-compose

Файл docker-compose.yml:

```
version: '3.9'

services:
  redis:
    image: redis:7-alpine
    container_name: redis
    ports:
      - "6379:6379"
    volumes:
      - redis_data:/data
    command: >
      sh -c "redis-server --appendonly yes --requirepass $$REDIS_PASSWORD"
    environment:
      REDIS_PASSWORD: mysecretpassword
    networks:
      - app-network

  redisinsight:
    image: redis/redisinsight:latest
    container_name: redisinsight
    depends_on:
      - redis
    expose:
      - "5540"
    environment:
      REDIS_HOSTS: "local:redis:6379"
    volumes:
      - redisinsight_data:/db
    networks:
      - app-network
```

```
redisinsight-proxy:
  image: nginx:alpine
  container_name: redisinsight-proxy
  ports:
    - "80:80"
  volumes:
    - ./redisinsight-nginx.conf:/etc/nginx/conf.d/default.conf:ro
    - ./nginx-auth/.htpasswd:/etc/nginx/.htpasswd:ro
  depends_on:
    - redisinsight
  networks:
    - app-network
```

```
web:
  image: nginx:alpine
  container_name: web
  ports:
    - "81:80"
  volumes:
    - ./nginx.conf:/etc/nginx/conf.d/default.conf:ro
    - ./usr/share/nginx/html:ro
  restart: always
  networks:
    - app-network
```

```
volumes:
  redis_data:
  redisinsight_data:
```

```
networks:
  app-network:
    driver: bridge
```

---

## 4. Установка и запуск

### 4.1. Шаг 1: Подготовка директории

Создайте рабочую директорию и перейдите в неё:

```
mkdir -p ~/redis-docker
cd ~/redis-docker
```

### 4.2. Шаг 2: Создание необходимых файлов

Для работы nginx-прокси с аутентификацией необходимо создать файл с пользователями и паролями.

Создание директории и файла nginx-auth/.htpasswd:

```
mkdir -p nginx-auth
htpasswd -bc nginx-auth/.htpasswd admin mysecretpassword
```

Создание файла redisinsight-nginx.conf:

```
server {
    listen 80;
    server_name localhost;

    location / {
        auth_basic "RedisInsight";
        auth_basic_user_file /etc/nginx/.htpasswd;

        proxy_pass http://redisinsight:5540;
        proxy_http_version 1.1;
        proxy_set_header Upgrade $http_upgrade;
        proxy_set_header Connection 'upgrade';
        proxy_set_header Host $host;
        proxy_set_header X-Real-IP $remote_addr;
        proxy_set_header X-Forwarded-For $proxy_add_x_forwarded_for;
        proxy_set_header X-Forwarded-Proto $scheme;
        proxy_cache_bypass $http_upgrade;
    }
}
```

### 4.3. Шаг 3: Запуск контейнеров

Для запуска Системы выполните команду:

```
docker compose up -d
```

Результат работы команды представлен на Рисунок 1.

**Рисунок 1** — Результат запуска Системы

NAME	IMAGE	STATUS	PORTS
redis	redis:7-alpine	Up	0.0.0.0:6379->6379/tcp
redisinsight	redis/redisinsight	Up	5540/tcp
redisinsight-proxy	nginx:alpine	Up	0.0.0.0:80->80/tcp
web	nginx:alpine	Up	0.0.0.0:81->80/tcp

### 4.4. Шаг 4: Проверка статуса

После запуска Системы необходимо убедиться, что контейнеры находятся в состоянии UP:

```
docker compose ps
```

---

## 5. Доступ к сервисам

### 5.1. Точки доступа

**Таблица 3** — Точки доступа к сервисам

Сервис	URL	Аутентификация
Redis CLI	localhost:6379	требуется (пароль: mysecretpassword)
RedisInsight	http://localhost	HTTP-авторизация (admin/mysecretpassword)
Web	http://localhost:81	нет

## 5.2. Доступ к RedisInsight

1. Откройте в браузере: `http://localhost`
2. Введите учетные данные:
  - Логин: `admin`
  - Пароль: `mysecretpassword`
3. После входа RedisInsight автоматически подключится к Redis

## 5.3. Подключение к Redis

Подключение через `redis-cli` (локально):

```
docker exec -it redis redis-cli -a mysecretpassword
```

Подключение с хост-машины:

```
sudo apt install redis-tools
redis-cli -h localhost -p 6379 -a mysecretpassword
```

Проверка подключения:

```
redis-cli -h localhost -p 6379 -a mysecretpassword ping
```

Ожидаемый ответ: PONG

---

## 6. Проверка работы

### 6.1. Базовая проверка

Проверка подключения:

```
redis-cli -h localhost -p 6379 -a mysecretpassword ping
```

Проверка версии сервера:

```
redis-cli -h localhost -p 6379 -a mysecretpassword INFO server
```

### 6.2. Тестирование записи и чтения

Запись данных:

```
redis-cli -h localhost -p 6379 -a mysecretpassword SET test "Hello Redis"
```

Чтение данных:

```
redis-cli -h localhost -p 6379 -a mysecretpassword GET test
```

Ожидаемый результат: Hello Redis

### **6.3. Проверка статистики**

Информация о сервере:

```
redis-cli -h localhost -p 6379 -a mysecretpassword INFO
```

Статистика:

```
redis-cli -h localhost -p 6379 -a mysecretpassword INFO stats
```

Использование памяти:

```
redis-cli -h localhost -p 6379 -a mysecretpassword INFO memory
```

Информация о персистентности:

```
redis-cli -h localhost -p 6379 -a mysecretpassword INFO persistence
```

---

## **7. Управление контейнерами**

### **7.1. Основные команды**

Запуск всех сервисов:

```
docker compose up -d
```

Остановка всех сервисов:

```
docker compose down
```

Перезапуск конкретного сервиса:

```
docker compose restart redis
```

Остановка с удалением volumes (удаление данных!):

```
docker compose down -v
```

### **7.2. Управление Redis**

Перезапуск Redis:

```
docker compose restart redis
```

Остановка Redis:

```
docker compose stop redis
```

Запуск Redis:

```
docker compose start redis
```

Выполнение команды внутри контейнера:

```
docker exec -it redis redis-cli -a mysecretpassword
```

---

## 8. Просмотр логов работы системы

Журналы дают возможность просматривать сообщения, выводимые программным обеспечением, работающим в ВС.

### 8.1. Логи Redis

Просмотр логов Redis:

```
docker compose logs redis
```

Просмотр логов в реальном времени:

```
docker compose logs -f redis
```

Просмотр последних 100 строк:

```
docker compose logs --tail=100 redis
```

### 8.2. Логи всех сервисов

Логи всех сервисов:

```
docker compose logs
```

Логи в реальном времени:

```
docker compose logs -f
```

### Рисунок 2 — Пример лога работы Системы

```
redis    | 1:C 19 Feb 2026 10:00:00.000 # oOoOoOoOoOoOo Redis is starting oOoOoOoOoOoOo
redis    | 1:C 19 Feb 2026 10:00:00.000 # Redis version=7.0.15, bits=64, commit=00000000
redis    | 1:C 19 Feb 2026 10:00:00.000 # Configuration loaded
redisinsight | [INFO] RedisInsight v2.54.0
```