



Курсус

АСУДД «Атлас»

Руководство администратора

Октябрь 2018

Содержание

1. Введение	3
2. Системные требования	3
3. Архитектура АСУДД «Атлас»	4
3.1. Драйвер (Driver)	4
3.2. Встраиваемое ПО «Дорожная станция» (Roadstation).....	4
3.2.1. Системные требования к дорожному контроллеру	5
3.3. Сервис передачи сообщений (Messaging Service)	5
3.4. Ядро (Core).....	5
3.5. Конфигурационный каталог (Configuration Folder)	6
3.6. Операционная база данных (Operational DB).....	6
3.7. Фронтенд (Frontend).....	6
4. Установка	7
4.1. Предусловия	7
4.2. Шаги установки	7
5. Обновление.....	7
6. Запуск, остановка, рестарт.....	8
7. Управление учетными записями пользователей.....	8
8. Резервное копирование и восстановление	8
9. Мониторинг и устранение неисправностей.....	8
10. Квалификационные требования к обслуживающему персоналу	9

1. Введение

АСУДД «Атлас» - это специализированной прикладное программное обеспечение для автоматизированного управления дорожным движением на автомагистралях.

Основные функциональные возможности АСУДД «Атлас»:

- Взаимодействие с широким спектром полевого оборудования
- Гибкие механизмы автоматического и автоматизированного управления
- Простой и удобный пользовательский интерфейс диспетчера АСУДД
- Интегрированное оперативное видеонаблюдение
- Ведение карточки события
- Открытый программные интерфейсы (API) для интеграции с полевым оборудованием и внешними информационными системами

2. Системные требования

Серверная операционная система: Linux, 64 бита (CentOS 7, RHEL 7, Debian 8)

СУБД: PostgreSQL 9.5 или выше

Клиентская среда: веб-браузер Google Chrome версии 50 и выше

Аппаратные требования к серверному оборудованию зависят от количества полевого оборудования, находящегося под управлением АСУДД, сложности алгоритмов управления, а также от объемов и временной глубины сохраняемых данных о дорожной обстановке и журналов системы.

Для несложного участка дороги (3 ТПИ, 6 детекторов транспорта, 1 метеостанция, 2 светофора и 2 шлагбаума) система может успешно работать на сервере со следующими характеристиками:

- Intel Xeon E5-2600 4 Core, RAM 16GB, SATA HDD 1TB RAID 1

Для крупных проектов с количеством полевого оборудования порядка 1500 единиц и развитой логикой автоматического рекомендуется использовать решения на основе инфраструктуры виртуализации со специализированными системами хранения данных со следующими совокупными аппаратными ресурсами:

- Intel Xeon 96 Core, RAM 256 GB, HDD 10 TB

3. Архитектура АСУДД «Атлас»

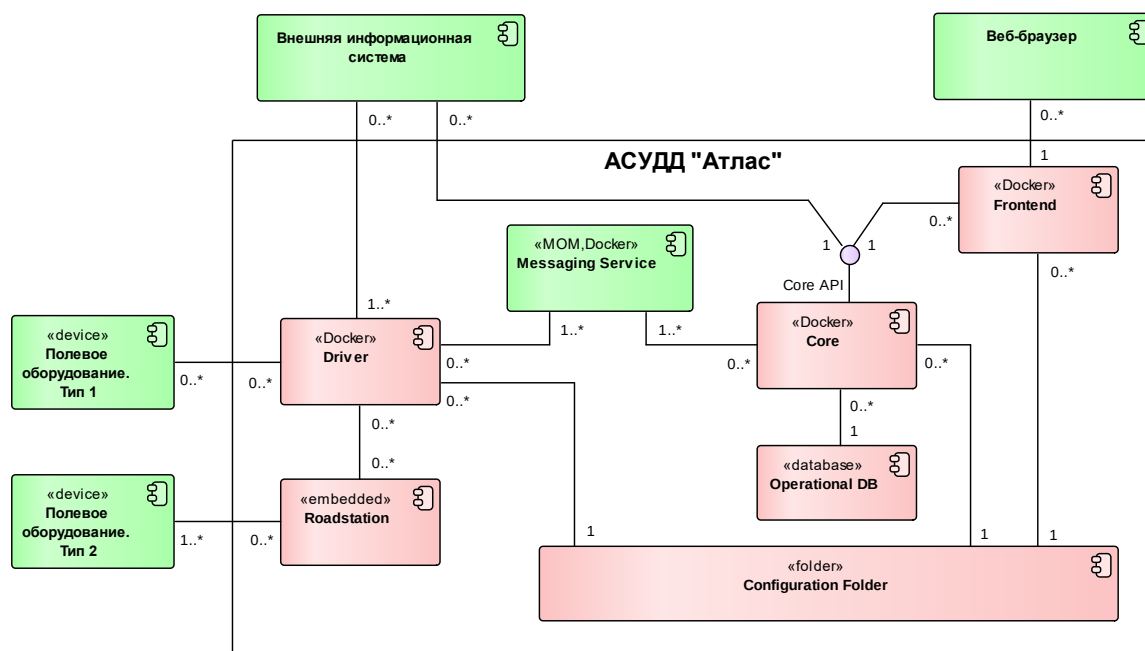


Рисунок 1. Архитектура АСУДД «Атлас»

3.1. Драйвер (Driver)

Драйвер (Driver) – это компонент центральной системы, осуществляющий взаимодействие с полевым оборудованием. В драйвере, как правило, реализуется конкретный программный интерфейс (API), предлагаемый производителем оборудования.

Драйвер может работать с оборудованием напрямую (на рисунке 12 помечено как «Тип 1»), или опосредованно через ПО «Дорожная станция» – roadstation (на рисунке 12 помечено как «Тип 2»).

Драйвер отправляет полученные от оборудования данные в сервис передачи сообщений (Messaging Service) и принимает команды управления оборудованием.

В общем случае, каждый драйвер может работать с несколькими элементами полевого оборудования, равно как и конкретный элемент полевого оборудования может взаимодействовать с несколькими драйверами.

С программной точки зрения, частным случаем полевого оборудования может быть внешняя информационная система. Например, система автоматического выявления инцидентов может выступать поставщиком информации об остановившихся на автомагистрали автомобилях.

3.2. Встраиваемое ПО «Дорожная станция» (Roadstation)

ПО «Дорожная станция» (Roadstation) – это специальное ПО, входящее в границы АСУДД «Атлас», которое устанавливается на дорожный контроллер и выполняет следующие основные функции:

- непосредственное взаимодействие с полевым оборудованием;
- локальное хранение данных, полученных от оборудования, на время недоступности центральной системы;

- локальное управление полевым оборудованием в период недоступности центральной системы;
- взаимодействие с центральной системой: отправка данных и трансляция команд управления.

На стороне центральной системы этот компонент взаимодействует с соответствующим драйвером.

ПО «Дорожная станция» построено по модульному принципу и при необходимости интеграции с новым полевым оборудованием для него может быть разработан соответствующий модуль. Кроме того, в составе ПО «Дорожная станция» есть, так называемый, командный модуль, который является оберткой над любой внешней командой операционной системы. Это позволяет в простых случаях быстро интегрировать оборудование путем написания простых скриптов на любом языке программирования.

3.2.1. Системные требования к дорожному контроллеру

- CPU ARMv7 или x86/64
- RAM 256 MB
- HDD 1GB
- Linux, 32 или 64 бита
- Python 3.5 или выше

Рекомендуемые промышленные компьютеры: Муха UC-8112-LX, Муха V2201-E1-T, Муха MC-1121-E2-T

3.3. Сервис передачи сообщений (Messaging Service)

Сервис передачи сообщений (Messaging Service) реализует распределенную среду передачи сообщений между ядром системы (Core) и уровнем драйверов (Driver).

Такой способ организации взаимодействия позволяет разворачивать распределенные конфигурации системы на большом количестве узлов. Таким образом достигается высокая масштабируемость. А свойственная такой архитектуре слабая связанность компонент существенно облегчает администрирование системы.

В один момент времени в системе может быть развернуто несколько сервисов передачи сообщений на нескольких узлах для обеспечения высокой надежности и масштабируемости.

3.4. Ядро (Core)

Ядро (Core) реализует всю бизнес-логику работы АСУДД и, в свою очередь, содержит следующие основные модули:

- Объектное представление полевого оборудования. Каждый элемент полевого оборудования описывается объектом с атрибутами, определяющими его состояние.
- Модуль обработки правил. Обрабатывает правила на предмет выполнения условия и, в зависимости от настроек, формирует аварийный сигнал и/или команду на запуск сценария управления.
- Модуль исполнения сценариев управления.
- Модуль карточек событий. Обеспечивает ведение карточек событий.
- Подсистема безопасности. Обеспечивает аутентификацию пользователей и авторизацию любых действий в системе на базе ролевой модели.
- Core API. Полнофункциональный программный интерфейс ядра, реализованный на базе широко распространенных в индустрии технологий и подходов: HTTP, JSON, REST, SSE.

В один момент времени в системе может быть запущено несколько ядер на различных физических или виртуальных узлах. При этом ядра будут делить между собой нагрузку по обработке правил и выполнению сценариев. Таким образом достигается высокая масштабируемость системы.

Данные о состоянии оборудования и команды на его изменения ядро отправляет драйверам через сервис передачи сообщений.

Фронтенд (Frontend), реализующий пользовательских интерфейс или внешние информационные системы, взаимодействуют с ядром через Core API.

Для хранения данных ядро использует операционную базу данных (Operational DB).

Документация на Core API реализована на основе Swagger UI и доступна по адресу:

http://HOST_NAME/api/v1

где HOST_NAME – адрес хоста, на котором установлена система.

3.5. Конфигурационный каталог (Configuration Folder)

Конфигурационный каталог содержит артефакты, специфичные для конкретного проекта:

- Мнемосхема
- Конфигурация полевого оборудования
- Объектное представление
- Программируемые правила
- Программируемые сценарии
- Код адаптеров драйвера
- Конфигурация атрибутов карточки события
- Звуковые файлы
- И т.д.

Этот каталог подключается к всем драйверам, ядрам и фронтендам в окружении.

3.6. Операционная база данных (Operational DB)

В операционной БД сохраняются все данные, необходимые для текущей работы системы:

- экземпляры правил и сценариев;
- сигналы;
- изменения состояния полевого оборудования;
- карточки событий;
- системный журнал;
- пользовательский настройки;
- учетные записи пользователей;
- и т.д.

В качестве СУБД используется PostgreSQL 9.5 или выше.

3.7. Фронтенд (Frontend)

Фронтенд (Frontend) – это серверное приложение, генерирующее пользовательский интерфейс, доступный в обычном веб-браузере.

Фронтенд взаимодействует с ядром через Core API и «ретранслирует» Core API внешним потребителям.

4. Установка

4.1. Предусловия

Перед началом установки следует убедиться в том, что в целевой системе присутствуют следующие компоненты:

1. Docker CE
<https://docs.docker.com/engine/installation/>
2. Docker Compose
<https://docs.docker.com/compose/install/>
3. PostgreSQL 9.5 или выше
<https://www.postgresql.org/download/>

4.2. Шаги установки

1. Создать каталог для АСУДД «Атлас» и скопировать в него дистрибутив

```
mkdir /atms && cp ./altas-1.2.9-e99084a.zip /atms/
```

2. Распаковать дистрибутив и загрузить docker-image компонентов в хост-систему

```
cd /atms/ && unzip ./altas-1.2.9-e99084a.zip && ./install.sh
```

3. Создать конфигурационный каталог

```
mkdir /opt/atms
```

4. Распаковать архив с конфигурацией в конфигурационный каталог
5. Отредактировать YML-файл соответствующего окружения.
6. Запустить АСУДД «Атлас»

```
cd /atms/ && ./run.sh %env_name% kill && ./run.sh %env_name% up -d
```

здесь %env_name% - следует заменить на название окружения, например: demo.

После запуска веб-интерфейс АСУДД «Атлас» будет доступна в веб-браузере Google по адресу хоста, на котором она установлена.

5. Обновление

Процедура обновления полностью идентична процедуре установки. Новые файлы следует переписывать поверх существующих.

6. Запуск, остановка, рестарт

Запуск

```
cd /atms && sudo ./run.sh %env_name% up -d
```

Остановка

```
cd /atms && sudo ./run.sh %env_name% kill
```

Рестарт

```
cd /atms && sudo ./run.sh %env_name% kill && sudo ./run.sh %env_name% up -d
```

здесь %env_name% - следует заменить на название окружения, например: demo.

7. Управление учетными записями пользователей

Сразу после установки в системе присутствует учетная запись **admin / admin** с полным набором прав.

Создание новых учетных записей доступно в разделе «**Администрирование > Редактор пользователей**».

8. Резервное копирование и восстановление

Резервному копированию подлежат следующие компоненты системы:

1. Конфигурационная папка
2. База данных системы

Конкретное расписание резервного копирования зависит от соглашения об уровне обслуживания. Обычно вполне допустимо иметь 3 ежедневные копии и 3 еженедельные.

Процедура полного восстановления системы:

1. Установить АСУДД «Атлас», но не запускать.
2. Скопировать последнюю резервную копию конфигурационной папки.
3. Восстановить базу данных из последней резервной копии.
4. Запустить АСУДД «Атлас».

9. Мониторинг и устранение неисправностей

Мониторинг работоспособности АСУДД «Атлас» и его компонент может быть организован с помощью любых соответствующих инструментов, например: Zabbix или Nagios.

АСУДД «Атлас» сам по себе не ведет технического журнала. Однако сбор логов может быть организован на базе перенаправления стандартного вывода контейнеров Docker.

```
sudo docker logs -f --tail=50 atms_backend
```


Таблица 1. Устранение неисправностей

Неисправность	Способ устранения
Система не отвечает	Проверить исправность сети передачи данных. Проверить наличие свободного места на дисках сервера и в СУБД. Выполнить рестарт АСУДД «Атлас».
Большое время отклика	Проверить пропускную способность сети передачи данных. Проверить утилизацию системных ресурсов (процессор, память, дисковый ввод-вывод) узлов, на которых развернуты компоненты АСУДД «Атлас». Устранить проблемы, если они исходят от сторонних сервисов. Выполнить рестарт АСУДД «Атлас».
Система не работает и рестарт не помогает	Восстановите систему из последней работоспособной резервной копии. Если ничего не помогает, обратитесь в службу техническое поддержки компании «Курсус» по координатам на сайте http://cursusts.com .

10. Квалификационные требования к обслуживающему персоналу

- Среднее техническое образование.
- Знания основ операционных системы семейства Linux.
- Опыт администрирования операционных системы семейства Linux не менее 3 лет.