# Arenadata™ Hadoop

*Версия - v2.1.4*

## Инструкция по установке кластера ADH

*2021 Arenadata™*

# Оглавление


1 Руководство по планированию кластера 3

2 Подготовка к установке кластера 12

3 Инструкция по установке ADCM 17

4 Загрузка бандла хостпровайдера ADCM 19

5 Выгрузка бандла хостпровайдера в ADCM 20

6 Добавление хостов в ADCM 22

7 Загрузка бандла кластера 23

8 Установка ADH в ADCM 26

Инструкция может быть полезна администраторам, программистам, разработчикам и сотрудникам подразделений информационных технологий, осуществляющих внедрение и эксплуатацию кластера.

Кластер Arenadata Hadoop (ADH) устанавливается при помощи Arenadata Cluster Manager (ADCM) – отдельного продукта Arenadata, предназначенного для простого, удобного и быстрого развертывания и эксплуатации программного обеспечения.

С документацией ADCM возможно ознакомиться по следующей ссылке: https://docs.arenadata.io/adcm/

# Глава 1

# Руководство по планированию кластера

Руководство по планированию кластера включает в себя следующие главы необходимые для прочтения перед непосредственной установкой кластера:

- *Рекомендации по аппаратному обеспечению для Apache Hadoop*
- *Рекомендации по партиционированию файловой системы*

## 1.1 Рекомендации по аппаратному обеспечению для Apache Hadoop

Рабочие нагрузки **Hadoop** и **HBase**, как правило, сильно различаются, и необходим опыт для правильного прогнозирования объемов хранилища, вычислительной мощности и межузловой связи, которые потребуются для выполнения различных видов работ.

В главе содержится информация о выборе соответствующих аппаратных компонентов для оптимального баланса между производительностью и стабильностью работы для различных нагрузок:

- *Типичный кластер Hadoop*
- *Шаблоны рабочей нагрузки для Hadoop*
- *Early Deployments*
- *Аппаратное обеспечение для узлов кластера*

Краткое резюме приведено в разделе *Заключение*.

**Hadoop** – это программный фреймворк, поддерживающий распределенный анализ данных на стандартных серверах. **Arenadata** является участником инициатив с открытым исходным кодом (**Apache Hadoop**, **HDFS**, **Pig**, **Hive**, **HBase**, **ZooKeeper**) и имеет большой опыт управления кластерами **Hadoop** на производственном уровне. **Arenadata** рекомендует следовать принципам проектирования, которые обеспечивают надежное развертывание на различном оборудовании.

Для кластера **Hadoop** или **HBase** крайне важно точно спрогнозировать объем, тип, и количество задач. Начиная с **Hadoop** или **HBase**, есть возможность получения метрик, измеряя фактические рабочие нагрузки во время пилотного проекта. Таким образом, можно легко масштабировать пилотную среду, не внося существенных изменений в существующее оборудование, программное обеспечение, стратегии развертывания и сетевое подключение.

### 1.1.1 Типичный кластер Hadoop

Кластеры **Hadoop** и **HBase** имеют два типа машин:

- Masters – HDFS NameNode, YARN ResourceManager и HBase Master;

- Slaves – HDFS DataNodes, YARN NodeManagers и HBase RegionServers.

Узлы DataNodes, NodeManagers и HBase RegionServers размещаются и разворачиваются совместно для оптимальной локализации данных. Кроме того, **HBase** требует использования отдельного компонента (**ZooKeeper**) для управления кластером.

**Arenadata** рекомендует разделять master (основные) и slave (подчиненные) узлы, поскольку:

- Рабочие нагрузки задачи/приложения на подчиненных узлах должны быть изолированы от основных;
- Подчиненные узлы часто выводятся из эксплуатации для технического обслуживания.

В целях эксперимента или тестирования можно развернуть **Hadoop**, используя инсталляцию с одним узлом (при условии, что все основные и подчиненные процессы находятся на одной машине). Для небольшого двухузлового кластера можно установить NameNode и ResourceManager на master-узле, а DataNode и NodeManager на slave-узле.

Кластеры из трех или более машин обычно используют один NameNode и ResourceManager со всеми другими узлами в качестве подчиненных. Кластер High-Availability использует первичный и вторичный NameNode, а также может использовать первичный и вторичный ResourceManager.

Как правило, кластер **Hadoop** среднего и большого размера состоит из двухуровневой или трехуровневой архитектуры, построенной на монтируемых в стойку серверах. Каждая стойка серверов соединена с помощью коммутатора *1 Gigabit Ethernet* (GbE). Каждый коммутатор уровня стойки подключен к коммутатору уровня кластера (который, как правило, является коммутатором с большей плотностью портов *10GbE*). Коммутаторы кластерного уровня могут также соединяться с другими коммутаторами уровня кластера или даже принадлежать к другому уровню инфраструктуры.

### 1.1.2 Шаблоны рабочей нагрузки для Hadoop

Дисковое пространство, пропускная способность ввода-вывода (необходимая для **Hadoop**) и вычислительная мощность (необходимая для процессов **MapReduce**) являются наиболее важными параметрами для точного определения размера оборудования. Кроме того, при установке **HBase** также необходимо проанализировать приложение и его требования к памяти, поскольку **HBase** является компонентом, интенсивно использующим память. Исходя из типичных вариантов использования **Hadoop**, в рабочих средах наблюдаются следующие шаблоны нагрузки:

- **Balanced Workload** – сбалансированная

Если рабочие нагрузки распределены равномерно между различными типами заданий (привязка к процессору, к дисковому вводу/выводу или к сетевому вводу/выводу), кластер имеет сбалансированный шаблон рабочей нагрузки. Это хорошая конфигурация по умолчанию для неизвестных или растущих нагрузок.

- **Compute Intensive** – ресурсоемкая

Рабочие нагрузки связаны с ЦП и характеризуются необходимостью большого количества процессоров и большого объема памяти для хранения данных в процессе. Шаблон использования типичен для обработки естественного языка или рабочих нагрузок HPCC.

- **I/O Intensive** – интенсивный ввод/вывод

Типичное задание MapReduce (например, сортировка) требует очень мало вычислительной мощности. Вместо этого оно больше зависит от связанной способности ввода/вывода кластера (например, при большом объеме холодных данных). Для такого типа рабочей нагрузки рекомендуется закладывать больше дисков.

- **Unknown or evolving workload patterns** – неизвестные или растущие шаблоны рабочей нагрузки

Изначально можно не знать необходимые шаблоны рабочей нагрузки. И, как правило, первые задания, переданные в **Hadoop**, сильно отличаются от фактических, которые будут выполняться в производственной среде. По этим причинам **Arenadata** рекомендует либо использовать конфигурацию сбалансированной рабочей нагрузки, либо инвестировать в пилотный кластер **Hadoop** и планировать развитие его структуры по мере анализа шаблонов рабочей нагрузки в среде.

### 1.1.3 Early Deployments

При первом знакомстве с **Hadoop** или **HBase** рекомендуется начинать с относительно небольшого экспериментального кластера, рассчитанного на Balanced Workload и получать оценку путем измерения фактических рабочих нагрузок во время пилотного проекта.

Для пилотного развертывания можно начать с 1U-машины и использовать следующие рекомендации:

- Два четырехъядерных процессора;
- От *12* до *24 ГБ* памяти;
- От четырех до шести дисков емкостью *2 ТБ*.

Минимальное требование для сети составляет *1GbE* all-to-all и может быть легко достигнуто путем подключения всех узлов к коммутатору Gigabit Ethernet. Чтобы использовать запасной сокет для добавления процессоров в будущем, можно использовать шести или восьми ядерный процессор.

Для небольших и средних кластеров **HBase** необходимо предоставить каждому серверу **ZooKeeper** около *1 ГБ* RAM и желательно собственный диск.

#### Jump-start – Hadoop Cluster

Один из способов быстрого развертывания **Hadoop** – выбрать "cloud trials" или использовать виртуальную инфраструктуру. **Arenadata** обеспечивает доступность дистрибутива через платформу данных **Enterprise Data Platform** (**EDP**), которую можно установить в общедоступных и частных облаках.

Обратиться в службу технической поддержки **Arenadata** можно по адресу электронной почты info@arenadata.io или через окно консультации на сайте www.arenadata.io.

Однако, облачные сервисы и виртуальные инфраструктуры не предназначены для **Hadoop**. В этом случае развертывания **Hadoop** и **HBase** могут иметь низкую производительность из-за виртуализации и неоптимальной архитектуры ввода/вывода.

#### Контроль ресурсов при пилотном развертывании

**Arenadata** рекомендует контролировать пилотный кластер с помощью **Ganglia**, **Nagios** или других систем мониторинга производительности. При этом важно:

- Измерить использование ресурсов для CPU, RAM, операций дискового ввода/вывода в секунду (IOPS) и отправленных и полученных сетевых пакетов. Запустить актуальные виды запросов или аналитических заданий;
- Убедиться, что подмножество данных масштабируется до размера пилотного кластера;
- Проанализировать данные мониторинга на предмет насыщения ресурсов. На основе этого анализа можно классифицировать задания как связанные с процессором, с дисковым вводом/выводом или с сетевым вводом/выводом.

---

**Important:** Большинство приложений Java допускают использование RAM до максимально заданного значения

---

- Проанализировать **ZooKeeper** (так как в нем часто обнаруживаются проблемы, связанные с сетью и памятью для **HBase**).
- (Опционально) Настроить параметры работы и конфигурации оборудования или сети, чтобы сбалансировать использование ресурсов. Если задания попадают в разные шаблоны рабочей нагрузки, можно выбрать управление только параметрами задания, а для оборудования оставить "balanced".

#### Challenges – настройка job-характеристик на использование ресурсов

Способ разработки приложения или представления данных может оказать большое влияние на баланс ресурсов. Например, затраты ресурсов могут быть смещены между дисковым IOPS и CPU с учетом выбранной схемы сжатия или формата синтаксического анализа. Процессор для каждого узла и активность диска можно обменять на пропускную способность между узлами в зависимости от реализации стратегии Map/Reduce.

Кроме того, Amdahl's Law показывает, как требования к ресурсам могут меняться в значительной степени нелинейным образом при изменении требований: изменение, которое, как можно ожидать, приведет к снижению затрат на вычисления на 50%, может вместо этого привести к изменению чистой производительности на 10% или на 90%.

#### Повторное использование пилотных машин

Установив пилотный кластер, можно приступить к анализу шаблонов рабочих нагрузок для выявления узких мест CPU и I/O. Позже эти машины могут быть повторно использованы в производственных кластерах, даже если базовые характеристики изменятся.

Чтобы добиться положительного коэффициента возврата инвестиций (return on investment, ROI), рекомендуется убедиться, что машины в пилотных кластерах составляют менее *10%* от конечного производственного кластера.

### 1.1.4 Аппаратное обеспечение для узлов кластера

В разделе приведены рекомендации по аппаратному обеспечению узла сервера для выбора количества узлов, параметров хранилища на узел (количество дисков и их размер, MTBF и затраты на репликацию при сбоях дисков), вычислительной мощности на узел (сокеты, ядра, тактовая частота), RAM на узел и возможности сети (количество, скорость портов):

- *Slave узлы*
- *Master узлы*

#### Slave узлы

**Серверная платформа**

Как правило, серверы с двумя сокетами являются оптимальными для развертывания **Hadoop**. Для средних и больших кластеров их использование является лучшим выбором по сравнению с серверами начального уровня благодаря возможности балансировки нагрузки и распараллеливания. С точки зрения компактности выбирается серверное оборудование, которое подходит для небольшого количества стоек. Обычно серверы *1U* или *2U* используются в *19"* стойках или шкафах.

**Возможности хранения**

Для приложений общего назначения рекомендуется использовать относительно большое количество жестких дисков, обычно от *8* до *12* дисков SATA LFF на сервер. В настоящее время типичная емкость **Hadoop** в производственных средах составляет около *2 ТБ* на диск. Высокоинтенсивные среды ввода/вывода могут потребовать диски *12x2 ТБ* SATA. Оптимальный баланс между затратами и производительностью, как правило, достигается с помощью SATA дисков со скоростью вращения *7200 об/мин*. Если в хранилище прогнозируется значительный рост, следует рассмотреть возможность использования дисков *3 ТБ*.

SFF-диски используются в некоторых конфигурациях для лучшей пропускной способности диска. Рекомендуется отслеживать кластер на предмет возможных сбоев дисков, так как увеличение их количества приводит и к повышению частоты сбоев. Если количество дисков на сервере велико, следует использовать два дисковых контроллера, чтобы нагрузка ввода/вывода могла распределяться между несколькими ядрами. Настоятельно рекомендуется использовать только SATA или SAS.

В **HDFS**, в котором используется недорогая надежная система хранения, данные остаются неограниченное время и потребности в хранилище быстро растут. С 12-дисковыми системами обычно получается

*24* или *36 ТБ* на узел. Использование такой емкости хранилища в узле целесообразно с версией **Hadoop 2.0** и выше.

**Hadoop** – это интенсивное и эффективное хранилище, не требующее при этом быстрых и дорогих жестких дисков. Если шаблон рабочей нагрузки не является I/O Intensive, можно добавить только четыре или шесть дисков на узел. Важно понимать, что затраты на электроэнергию пропорциональны количеству дисков, а не емкости хранилища.

**Important:** RAID vs. JBOD: не рекомендуется использовать RAID на slave-машинах Hadoop. Кластер допускает вероятность сбоя диска и обеспечивает избыточность данных на всех подчиненных узлах

**Important:** Диски должны иметь хорошие значения MTBF, так как подчиненные узлы в Hadoop подвержены сбоям

Подчиненные узлы не нуждаются в дорогостоящей поддержке, предлагающей услуги замены дисков в течение двух часов или меньше. **Hadoop** адаптирован к отказам slave-узлов, и поэтому следует относиться к работам по обслуживанию подчиненных узлов как к постоянной задаче, а не как к чрезвычайной ситуации.

Хорошо иметь возможность замены дисков, без изъятия сервера из стойки, хотя при этом непродолжительное отключение стойки является недорогой операцией в кластере.

**Размер памяти**

Крайне важно обеспечить достаточную память, чтобы процессоры были заняты без подкачки и без чрезмерных затрат на нестандартные материнские платы. В зависимости от количества ядер подчиненным узлам обычно требуется от *24* до *48 ГБ* оперативной памяти. Для больших кластеров этот объем памяти обеспечивает достаточно дополнительной RAM (приблизительно *4 ГБ*) для платформы **Hadoop** и для процессов запросов и анализа (**HBase** и/или Map/Reduce).

Для обнаружения и исправления случайных нерегулярных ошибок настоятельно рекомендуется использовать память с error correcting code (ECC). Исправление ошибок RAM позволяет доверять качеству вычислений. Доказано, что некоторые детали (chip-kill/chip spare) обеспечивают лучшую защиту с меньшим повторением битовых ошибок, чем традиционные конструкции.

При желании сохранения возможности добавления дополнительной памяти на серверы в будущем необходимо убедиться, что для этого есть место рядом с начальными модулями памяти.

**Подготовка памяти**

Память может представлять собой недорогие материнские платы на серверах низкого уровня, что типично для технологии Over-Provisioning. Неиспользуемая RAM в таком случае применяется либо приложениями **Hadoop** (обычно при параллельном запуске нескольких процессов), либо инфраструктурой (для кэширования данных на диске с целью повышения производительности).

**Процессоры**

Не смотря на то, что важно понимать шаблон рабочей нагрузки, рекомендуется использовать процессоры со средней тактовой частотой и менее, чем с двумя сокетами. Для большинства рабочих нагрузок дополнительная производительность на узел не является экономически эффективной. Следует использовать как минимум два четырехъядерных процессора для подчиненных машин больших кластеров.

**Мощность**

Мощность является главной задачей при проектировании кластеров **Hadoop**. Прежде, чем приобретать самые большие и быстрые узлы, советуется проанализировать расход энергии для имеющегося оборудования. Существует возможность огромной экономии в затратах и энергопотреблении путем избежания покупки самых быстрых процессоров, резервных источников питания и прочего.

Производители создают легковесные машины для облачных центров обработки данных с целью снижения затрат и энергопотребления. Например, **Supermicro**, **Dell** и **HP** имеют такие линейки продуктов для облачных провайдеров. Поэтому при необходимости закупки в большом объеме, рекомендуется оценить эти упрощенные "cloud servers".

Для подчиненных узлов достаточно одного блока питания (PSU), но для мастер-серверов необходимо использовать резервные блоки. Конструкции серверов, которые совместно используют PSU на смежных серверах, могут обеспечить повышенную надежность без увеличения затрат.

**Important:** Потребляемая мощность кластера: электричество и охлаждение составляют от *33,33%* до *50%* общей стоимости жизненного цикла оборудования в современных дата-центрах

**Сеть**

Сеть – это наиболее сложный параметр для предварительной оценки, поскольку рабочие нагрузки **Hadoop** сильно различаются. Ключевой позицией является покупка достаточной емкости сети при приемлемых затратах так, чтобы все узлы в кластере могли обмениваться данными с разумной скоростью. В больших кластерах обычно используются двойные каналы по *1 ГБ* для всех узлов в каждой 20-node стойке и соединительные каналы *2x10 ГБ*, доходящие до пары центральных коммутаторов.

Хороший расчет сети учитывает возможность перегрузки в критических точках при реальных нагрузках. Общепринятые коэффициенты превышения oversubscription составляют примерно 4:1 на уровне доступа к серверу и 2:1 между уровнем доступа и уровнем или ядром агрегации. Более низкие показатели превышения переподписки можно рассматривать, если требуется повышенная производительность. Кроме того, рекомендуется oversubscription *1 ГБ* между стойками.

Крайне важно иметь выделенные коммутаторы для кластера, а не пытаться назначать виртуальные каналы в существующих коммутаторах – нагрузка **Hadoop** воздействует на остальных пользователей коммутатора. Не менее важно работать с сетевой командой, чтобы гарантировать, что коммутаторы подходят как **Hadoop**, так и инструментам мониторинга.

Сеть необходимо разработать так, чтобы сохранить возможность добавления дополнительных стоек серверов **Hadoop**/**HBase**. В ином случае исправления в сети могут быть дорогостоящими. "Глубокая буферизация" ("Deep buffering") предпочтительнее низкой задержки в коммутаторах. Кроме того, включение Jumbo Frames в кластере совершенствует пропускную способность за счет улучшения контрольных сумм файлов, а также может обеспечить целостность пакета.

**Important:** Сетевая стратегия Hadoop: важно проанализировать соотношение стоимости network-to-computer и убедиться, что стоимость сети составляет около 20% от общей стоимости. Hadoop разработан с учетом аппаратного обеспечения, и сетевые затраты должны комплексно включать: коммутаторы ядра сети, коммутаторы стойки, любые необходимые сетевые карты и т.д.

### Master узлы

Главные узлы, будучи уникальными, предъявляют значительно иные требования к хранению и памяти, чем подчиненные узлы. Далее рассматриваются некоторые из компромиссов между памятью и хранилищем. Ориентир по сайзингу для небольших (5–50 узлов) и средних/больших (100-1000 узлов) кластеров приведены в главе *Заключение*.

Рекомендуется использовать два сервера NameNode – основной и вторичный. Оба сервера должны иметь высоконадежное хранилище для пространства имен и ведения журнала edit-log. Как правило, аппаратный RAID и/или надежное сетевое хранилище являются оправданными вариантами.

Главные серверы должны иметь как минимум четыре резервных тома хранения, несколько локальных и сетевых, но при этом каждый может быть относительно небольшим (обычно *1 ТБ*). RAID-диски на главных

узлах являются хорошим местом для контрактов поддержки при этом рекомендуется включить опцию замены диска on-site.

**Варианты хранения для ResourceManager**

На самом деле сервера ResourceManager не нуждаются в RAID-хранилище, поскольку они сохраняют свое состояние в **HDFS**. Сервер фактически может быть запущен на slave-узле с небольшим количеством дополнительной RAM. Однако использование тех же аппаратных спецификаций для ResourceManager, что и для NameNode, обеспечивает возможность переноса NameNode на тот же сервер, что и ResourceManager, в случае сбоя NameNode, и копию состояния NameNode может быть сохранена в сетевом хранилище.

**Память**

Объем памяти, необходимый для мастер-узлов, зависит от количества объектов файловой системы, которые создаются и отслеживаются с помощью NameNode. *64 ГБ* оперативной памяти поддерживает около *100* миллионов файлов. Некоторые сайты сейчас экспериментируют с *128 ГБ* RAM для еще больших пространств имен.

**Процессоры**

NameNodes активно взаимодействует с клиентами и компонентами кластера. Поэтому рекомендуется предоставлять *16* или даже *24* ядер CPU для обработки трафика обмена сообщениями для главных узлов.

**Сеть**

Кластер желательно обеспечение множеством сетевых портов и пропускной способностью *10 ГБ* для коммутатора.

### 1.1.5 Аппаратное обеспечение HBase

**HBase** использует различные типы кэшей для заполнения памяти, и, как правило, чем больше памяти у **HBase**, тем лучше он может кэшировать запросы на чтение. Каждый slave-узел в кластере **HBase** (RegionServer) поддерживает несколько областей (фрагментов данных в памяти). Для больших кластеров важно убедиться, что HBase Master и NameNode работают на отдельных серверах. Так же необходимо обратить внимание, что в крупномасштабных развертываниях узлы **ZooKeeper** не разворачиваются совместно с подчиненными узлами **Hadoop**/**HBase**.

**Варианты хранения**

При распределенной настройке **HBase** хранит данные в **HDFS**. Для получения максимальной локальности чтения/записи HBase RegionServers и DataNodes должны быть совместно развернуты на одних и тех же машинах. Поэтому все рекомендации по настройке оборудования DataNode и NodeManager также применимы к RegionServers. В зависимости от того, ориентированы ли приложения **HBase** на чтение/запись или на обработку, необходимо сбалансировать количество дисков с количеством доступных ядер процессора. Как правило, должно быть по крайней мере одно ядро на диск.

**Память**

Главные узлы **HBase** не так интенсивно используют вычислительные ресурсы, как типичный сервер RegionServer или NameNode. Поэтому для мастера **HBase** можно выбрать более скромную настройку памяти. Требования к памяти RegionServer сильно зависят от характеристик рабочей нагрузки кластера. Хотя избыточное выделение памяти выгодно для всех шаблонов рабочей нагрузки.

Кроме того, при работе кластера **HBase** с **Hadoop** необходимо обеспечить избыточную память для Hadoop MapReduce не менее, чем на *1–2 ГБ* для каждой задачи поверх памяти **HBase**.

### 1.1.6 Другие аспекты

**Вес**

Плотность хранения серверов последнего поколения подразумевает, что необходимо учитывать вес стоек. Следует убедиться, что вес стойки не превышает допустимую нагрузку пола центра обработки данных.

**Масштабируемость**

**Hadoop** легко масштабировать, добавляя в него новые сервера или целые серверные стойки и увеличивая объем памяти в главных узлах для работы с повышенной нагрузкой. Сначала это создает “перебалансировку трафика”, но в итоге обеспечивает дополнительное место для хранения и вычисления.

Для масштабирования кластера необходимо:

- Убедиться, что в центре обработки данных рядом с кластером **Hadoop** имеется потенциальное свободное пространство. Оно должно быть в состоянии вместить бюджет мощности для увеличенного количества стоек;
- Планировать сеть так, чтобы справиться с ростом количества серверов;
- Добавить больше дисков и RAM при наличии запасных сокетов в серверах, что позволит расширить существующий кластер без добавления дополнительных стоек или сетевых изменений;
- Планировать расширение по одному серверу за раз, так как обновление оборудования в работающем кластере может занять значительное время;
- При планировании дополнительного ЦП следует проконсультироваться с торговым представителем о сроках поставки, которые могут занять до 18 месяцев;
- Может потребоваться больше памяти на master-серверах.

**Support**

Концепция, которую следует рассмотреть, – это “заботиться о главных узлах, следить за подчиненными узлами”. Для большинства узлов кластера не нужны традиционные контракты на аппаратную поддержку корпоративного класса, поскольку их сбои являются скорее статистической проблемой, чем кризисом. Сэкономленные средства на поддержке оборудования могут пойти на большее количество подчиненных узлов.

**Ввод в эксплуатацию**

Важно обратить внимание, что “smoke tests”, которые поставляются с кластером **Hadoop**, являются хорошим начальным тестом.

### 1.1.7 Заключение

Достижение оптимальных результатов при установки кластера **Hadoop** начинается с выбора правильных аппаратных и программных стеков. Усилия, предпринимаемые на этапах планирования, могут значительно окупиться с точки зрения производительности и общей стоимости владения (TCO).

Следующие рекомендации составного системного стека могут помочь на этапах планирования:

Таблица1.1.: Рекомендации по составному системному стеку

| Сервер | Нагрузка | Хранение[1] | Процессор | Память | Сеть |
|---|---|---|---|---|---|
| Slaves | Balanced workload | Двенадцать дисков по 2-3 ТБ | 8 ядер | 128-256 ГБ | 1 GB onboard, 2x10 GbE mezzanine/external |
| | Compute-intensive workload | Двенадцать дисков по 1-2 ТБ | 10 ядер | 128-256 ГБ | 1 GB onboard, 2x10 GbE mezzanine/external |
| | Storage-heavy workload | Двенадцать дисков по 4+ ТБ | 8 ядер | 128-256 ГБ | 1 GB onboard, 2x10 GbE mezzanine/external |
| NameNode | Balanced workload | Четыре или более 2-3 ТБ RAID 10 with spares | 8 ядер | 128-256 ГБ | 1 GB onboard, 2x10 GbE mezzanine/external |
| Resource Manager | Balanced workload | Четыре или более 2-3 ТБ RAID 10 with spares | 8 ядер | 128-256 ГБ | 1 GB onboard, 2x10 GbE mezzanine/external |

## 1.2 Рекомендации по партиционированию файловой системы

**Настройка партиций файловой системы**

Базовая конфигурация для всех узлов кластера:

- Root-партиция: ОС и основные программные файлы;
- Swap: размер *2x* системной памяти.

**Партиционирование Slave-узлов:**

- Партиции Hadoop Slave: Hadoop должен иметь свои собственные партиции для файлов и журналов. Диски должны быть разделены с помощью XFS или ext4 (в соответствующем порядке предпочтения). Крайне не рекомендуется использовать LVM – создает задержку и узкие места;
- На slave-узлах все партиции Hadoop должны устанавливаться индивидуально для дисков как */hdfs/[0-n]*;
- Пример конфигурации Hadoop Slave Node Partitioning:
  - */root* – 20 ГБ: достаточно для существующих файлов, дальнейшего роста журнала и обновлений ОС;
  - */hdfs/0/* – полный диск ГБ: первая партиция Hadoop для локального хранилища;
  - */hdfs/1/* – вторая партиция Hadoop;
  - */hdfs/2/* – . . .

**Рекомендации по резервированию (RAID):**

- Master узлы – конфигурация на надежность (RAID 10, двойные карты Ethernet, двойные блоки электропитания);
- Slave узлы – RAID не требуется, сбои на узлах управляется автоматически кластером. Все данные хранятся по крайней мере на трех разных хостах.

[1] Дополнительно зарезервировать как минимум 2,5 ГБ на жестком диске для каждой устанавливаемой версии **Hadoop**.

# Глава 2

# Подготовка к установке кластера

В разделе приведена информация и даны материалы для подготовки к установке кластера **ADH**:

- *Системные требования*;
- *Сбор информации*;
- *Подготовка к установке Hadoop*.

## 2.1 Системные требования

Для корректного запуска **Hadoop** необходимо соблюдение минимальных системных требований, представленных в таблице.

Таблица2.1.: Минимальные системные требования для установки кластера Arenadata Hadoop

| Компонент | Требования |
|---|---|
| Платформа | • Intel x86_64 |
| Операционная система | Поддерживаются следующие 64-разрядные ОС:<br>• Red Hat Enterprise Linux (RHEL) v7.X;<br>• CentOS v7.X;<br>Установщик использует множество пакетов из базовых репозиториев ОС. Если нет полного набора базовых репозиториев ОС, доступных для всех компьютеров во время установки, могут возникнуть проблемы.<br>В случае возникновения проблем с недоступными базовыми хранилищами ОС, необходимо обратиться к системному администратору для обеспечения проксирования или зеркалирования этих дополнительных репозиториев |

| Компонент | Требования |
|---|---|
| Браузер | Arenadata Cluster Manager работает как веб-приложение на основе браузера. Поэтому необходимо наличие машины, способной использовать графический браузер для применения данного веб-инструмента.<br>Минимальные требуемые версии браузера:<br>• Internet Explorer 11.0 (устаревшая версия);<br>• Firefox 18;<br>• Google Chrome 26;<br>• Safari 10;<br>На платформе рекомендуется обновить браузер до последней, стабильной версии (кроме Internet Explorer 11.0) |
| Программное обеспечение | На всех узлах кластера необходимо установить следующие компоненты:<br>• YUM и rpm (RHEL / CentOS);<br>• Scp, curl, unzip, tar и wget; |
| Java | Поддерживаются следующие среды выполнения Java:<br>• Oracle JDK 1.8 64-разрядный: минимум JDK 1.8.64; |

## 2.2 Сбор информации

Перед развертыванием кластера **ADH** необходимо:

- Проверить полное доменное имя (FQDN) каждого хоста в кластере. Для проверки и установки FQDN необходимо воспользоваться командой:

```
hostname -f
```

Развертывание всех компонентов **ADH** на одном хосте возможно, но не рекомендуется для продуктивного использования. Как правило, для минимального кластера настраивается три узла – один главный и два подчиненных.

- Определить список компонентов, которые необходимо настроить на каждом узле;
- Установить базовые каталоги, которые будут определены в качестве точек для хранения:
  - Узла NameNode;
  - Узла (узлов) DataNode (в этих каталогах не должно содержаться никаких других данных);
  - Узла Secondary NameNode;
  - Узла YARN;
  - Узлов ZooKeeper, если устанавливается ZooKeeper;
  - Различных журналов, файлов *pid* и *db*, в зависимости от типа установки.

Необходимо использовать каталоги, которые предоставляют постоянные места хранения компонентов **ADH** и данных **Hadoop**.

**Important:** Не используйте */TMP* каталог для установки, так как файлы могут быть удалены в любое время

## 2.3 Подготовка к установке Hadoop

Для корректного развертывания **Arenadata Hadoop** необходимо выполнить действия, описанные в главах:

- *Синхронизация часов на всех узлах кластера*
- *Настройка DNS и NSCD*
- *Настройка IPTables*
- *Настройка SELinux, PackageKit и Umask*

### 2.3.1 Синхронизация часов на всех узлах кластера

Часам на всех узлах кластера необходимо иметь возможность синхронизации друг с другом. Для этого следует включить службу **NTP** и убедиться, что синхронизация происходит автоматически:

- RHEL/CentOS 7.X:

  ```
  systemctl is-enabled ntpd
  ```

Для настройки службы **NTP** на автоматический запуск при загрузке машины необходимо выполнить следующую команду на каждом хосте:

- RHEL/CentOS 7.X:

  ```
  systemctl enable ntpd
  ```

  Для запуска службы **NTP** необходимо выполнить команду на каждом хосте:

- RHEL/CentOS 7.X:

  ```
  systemctl start ntpd
  ```

### 2.3.2 Настройка DNS и NSCD

Все узлы в кластере должны быть настроены как для прямого, так и для обратного **DNS**.

В случае если не получается настроить **DNS**, необходимо отредактировать файл */etc/hosts* на каждом узле кластера так, чтобы он содержал IP-адрес и **FQDN** каждого узла.

Приведенные инструкции представлены в виде обзора и охватывают базовую настройку сети для общих узлов **Linux**. Разные версии и варианты **Linux** могут потребовать различные команды и процедуры.

**Hadoop** сильно зависит от **DNS** и выполняет многие DNS-запросы во время работы. Для того, чтобы снизить нагрузку на инфраструктуру **DNS**, настоятельно рекомендуется использовать **Name Service Caching Daemon** (**NSCD**) на узлах кластера под управлением **Linux**. **NSCD** кэширует запросы хоста, пользователя и группы, а также обеспечивает лучшую производительность и снижает нагрузку на инфраструктуру **DNS**.

#### Редактирование файла хоста

Для редактирования файла хоста необходимо, используя текстовый редактор, открыть файл *hosts* на каждом узле кластера командой:

```
vi/etc/hosts
```

И добавить строку для каждого хоста, состоящую из IP-адреса и **FQDN**, например:

```
1.2.3.4 <fully.qualified.domain.name>
127.0.0.1 localhost.localdomain localhost
::1 localhost6.localdomain6 localhost6
```

#### Проверка имени хоста

Проверка установленного имени хоста осуществляется командой:

```
hostname -f
```

В результате для имени хоста возвращается значение *fully.qualified.domain.name*. Для установки имени на каждом узле кластера необходимо использовать команду:

```
hostname <имя узла>
```

#### Редактирование файла конфигурации сети

С помощью текстового редактора необходимо открыть файл конфигурации сети на каждом узле и установить ему требуемую конфигурацию сети для каждого узла:

```
vi /etc/sysconfig/network
```

Следует изменить свойство *HOSTNAME*, чтобы задать **FQDN**:

```
NETWORKING=yes
HOSTNAME=<fully.qualified.domain.name>
```

### 2.3.3 Настройка IPTables

Для взаимодействия во время установки компонентов с развернутыми узлами необходимо, чтобы определенные компоненты были открыты и доступны. Самый простой способ сделать это – временно отключить **IPTables**:

- RHEL/CentOS 7:

```
systemctl disable firewalld
systemctl stop firewalld
```

По завершению установки следует перезапустить **IPTables**.

В случае если протоколы безопасности предотвращают отключение **IPTables**, можно продолжить работу с включенными **IPTables** при условии, что все необходимые порты открыты и доступны.

### 2.3.4 Настройка SELinux, PackageKit и Umask

Работа с **SELinux** не поддерживается, поэтому необходимо его отключить, выполнив команду:

- RHEL/CentOS 7:

```
setenforce 0
```

И установить значение */etc/selinux/config*.

На установочном узле, где запущен RHEL/CentOS с установленным **PackageKit**, с помощью текстового редактора необходимо открыть */etc/yum/pluginconf.d/refresh-packagekit.conf* и выполнить следующее изменение:

- RHEL/CentOS 7:

```
enabled=0
```

При создании нового файла или папки в системе **Linux UMASK** устанавливает разрешения по умолчанию или базовые разрешения. Большинство дистрибутивов **Linux** устанавливают **UMASK** значение по умолчанию *022*. Это разрешение допускает чтение, запись и выполнение *755* для новых файлов или папок. Значение **UMASK** равное *027* допускает чтение, запись и выполнение *750* для новых файлов или папок.

Поддерживается оба значения **UMASK**. Например, чтобы установить значение **UMASK** равное *022*, необходимо запустить команду в корне на всех хостах:

```
vi /etc/profile
```

Затем добавить следующую строку:

```
umask 022
```

# Глава 3

# Инструкция по установке ADCM

Для установки **ADCM** необходимо выделить отдельный хост. Перед установкой **ADCM** необходимо убедиться, что целевой для установки хост подходит под требования, описанные в документации продукта.

Также рекомендуется ознакомиться с базовыми понятиями, используемыми внутри **ADCM**.

Далее необходимо установить **ADCM**, руководствуясь инструкцией.

После установки **ADCM** его веб-интерфейс становится доступен на порте *8000* развернутого контейнера. Дальнейшая установка кластера **ADH** происходит через данный веб-интерфейс.

При установке **ADCM** по умолчанию создается пользователь с правами администратора – необходимо использовать эту учетную запись для первичного входа в **ADCM**:

- Имя пользователя: *admin*
- Пароль: *admin*

---

**Important:** Компоненты кластера имеют возможность отсылать статусную информацию о своем состоянии в ADCM. Для корректной работы этой функциональности могут потребоваться ручные корректировки автоматических настроек ADCM

---

Контейнер **ADCM** может оказаться в сети с использованием NAT, что исключает возможность автоматического определения адреса контейнера, видимого со стороны узлов кластера **ADH** (их установка описана далее). В таком случае может потребоваться указание адреса вручную, для этого необходимо перейти во вкладку “Settings” веб-интерфейса **ADCM** и проверить на корректность автоматически определившийся URL, при необходимости заменив его на исправный (Рис.3.1.).

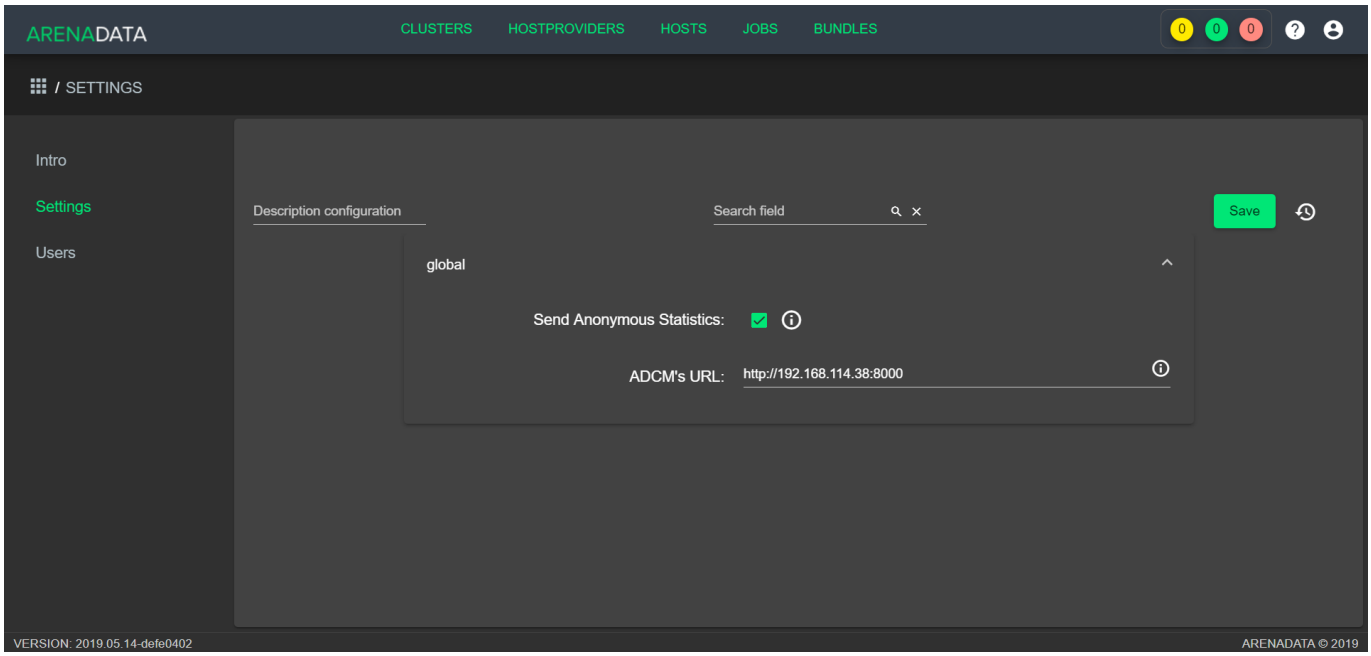



Рис.3.1.: Вкладка “Settings”

# Глава 4

# Загрузка бандла хостпровайдера ADCM

Для управления кластрером и его узлами через **ADCM** сначала необходимо подготовить хосты под кластер, а также связать эти хосты с **ADCM**. За эту функциональность отвечают хостпровайдеры **ADCM** – плагины **ADCM**, которые инкапсулируют сложный механизм взаимодействия **ADCM** и подконтрольных ему хостов.

Дистрибутивы хостпровайдеров для **ADCM** выполнены в виде так называемых бандлов. В случае простейшего хост провайдера ssh – это обычный архив, содержащий описание и программную логику подключения к хосту по протоколу ssh. Провайдер ssh в дальнейшем используется как пример.

Для загрузки бандла хостпровайдера необходимо перейти на сайт https://arenadata.tech/products/adcm и выбрать вкладку “Скачать”. В результате чего отражается краткое описание продукта, ссылки на компоненты дистрибутива и информация о релизе (Рис.4.1.).

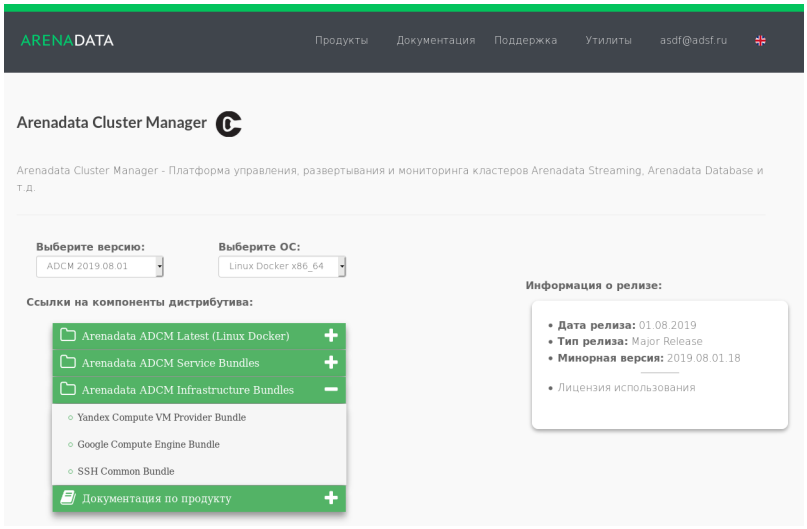



Рис.4.1.: Страница загрузки Arenadata Cluster Manager

После указания версии **ADCM** и операционной системы в раскрывающемся списке “Arenadata ADCM Infrastructure Bundles” предоставляется выбор вариантов загрузки бандлов хостпровайдеров – по нажатию на элемент списка браузер начинает загрузку. В блоке “Информация о релизе” отображается дата выхода, тип релиза выбранной версии и ссылки на лицензию.

Бандл хостпровайдера ssh называется *SSH Common Bundle*.

# Глава 5

# Выгрузка бандла хостпровайдера в ADCM

Для выгрузки бандла в **ADCM** следует выполнить следующие действия в веб-интерфейсе **ADCM**:

1. Открыть вкладку *“BUNDLES”* (Рис.5.1.).

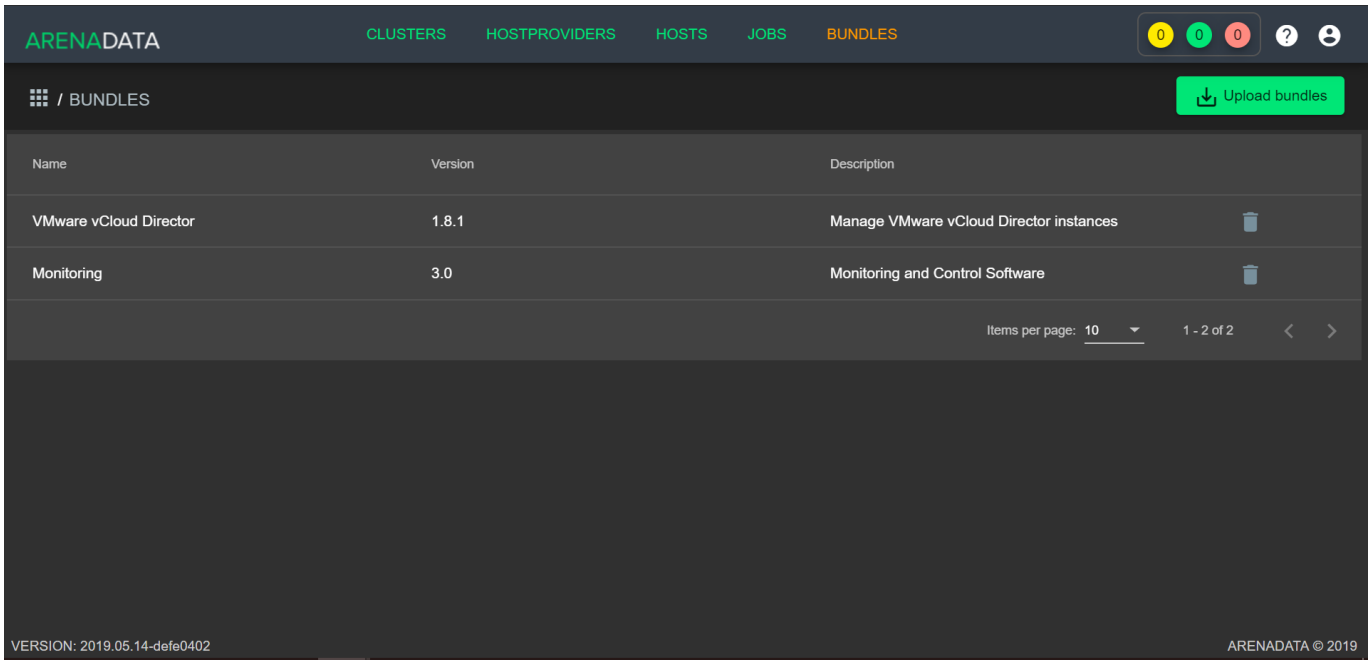



Рис.5.1.: Вкладка “BUNDLES”

2. Нажать *“Upload bundle”* и в открывшейся форме выбрать файл бандла (Рис.5.2.).
3. В результате выполненных действий факт успешной выгрузки отображается в общем списке бандлов на вкладке *“BUNDLES”* (Рис.5.3.).

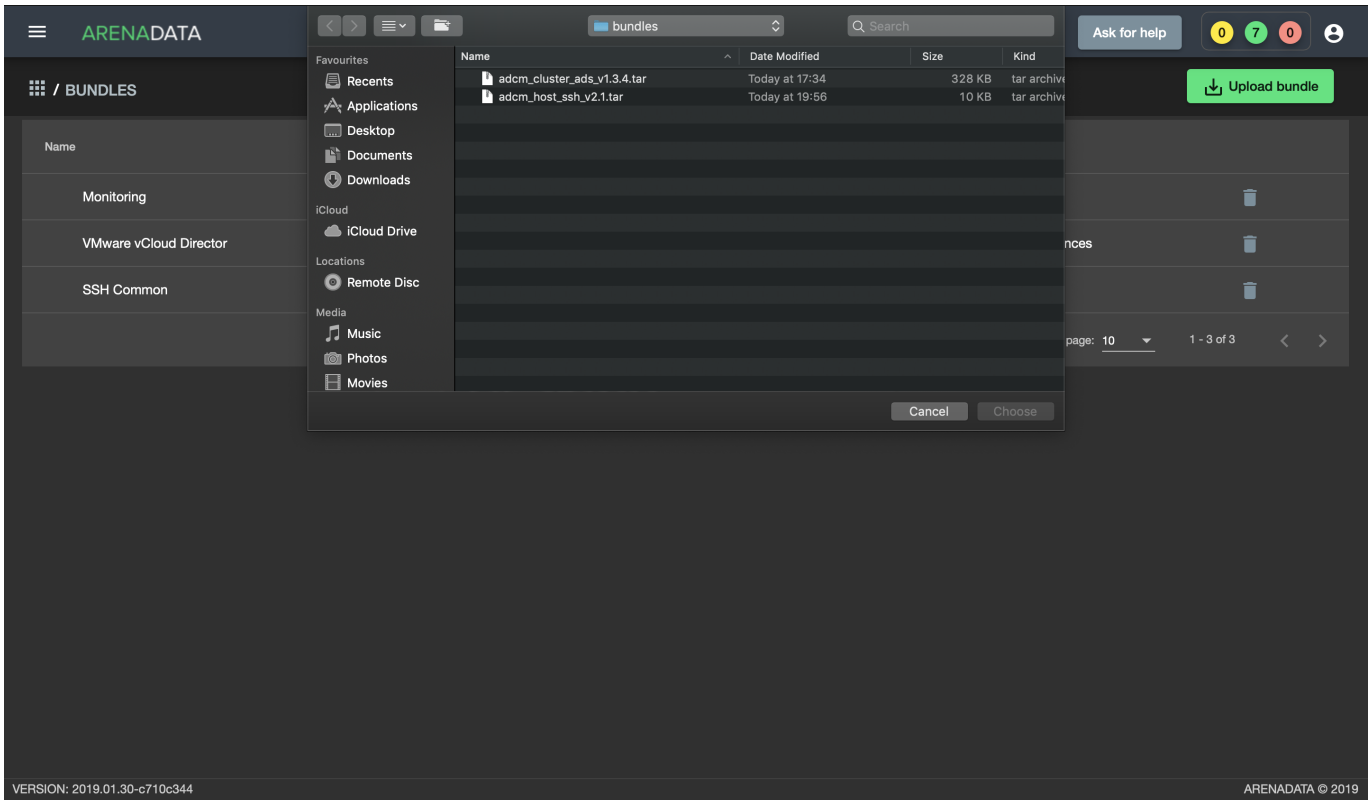



Рис.5.2.: Выбор файла бандла

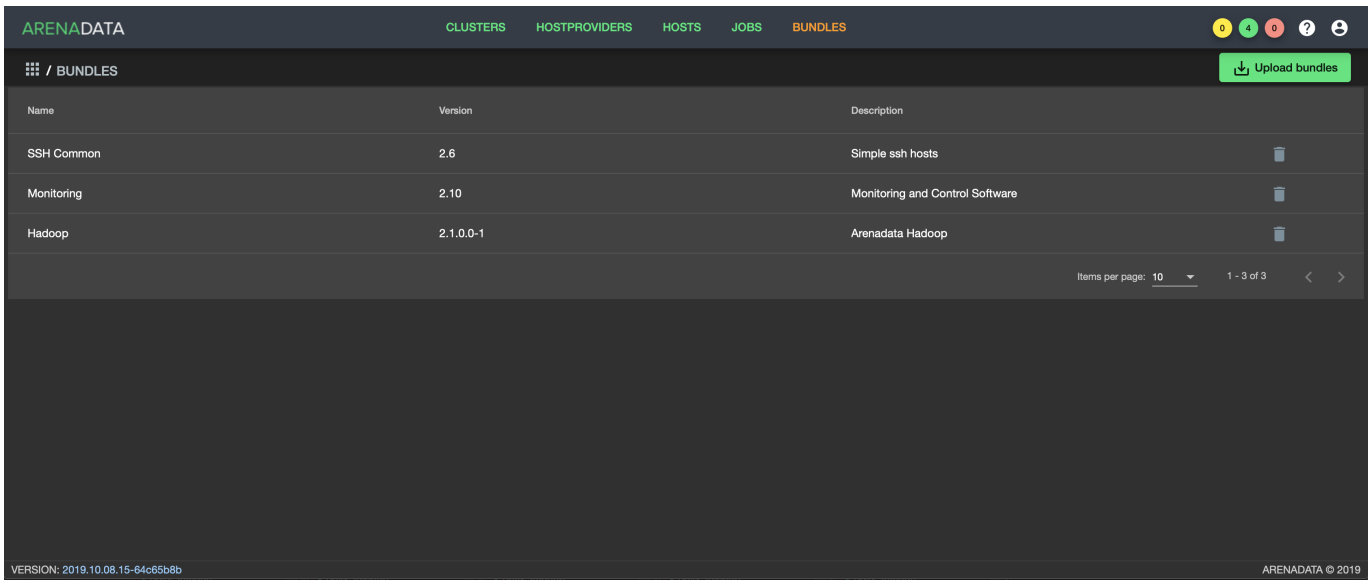



Рис.5.3.: Факт успешной выгрузки бандла

# Глава 6

# Добавление хостов в ADCM

В зависимости от выбранного хостпровайдера может потребоваться его изначальная настройка, а также настройка каждого добавленного хоста.

В простейшем случае при использовании хостпровайдера ssh (который является не более чем коннектором к уже созданным хостам) необходимо сначала создать хосты с возможностью подключения по протоколу ssh – эта операция производится вручную и данным хостпровайдером не поддерживается.

Далее необходимо создать хосты внутри **ADCM**:

- Загрузить выбранный бандл хостпровайдера как это описано в документации ADCM;
- На вкладке *“HOSTPROVIDERS”* добавить экземпляр загруженного бандла, воспользовавшись кнопкой *“Create provider”*;
- На вкладке *“HOSTS”* добавить хосты, воспользовавшись кнопкой *“Create host”*.

Настройки для каждого добавленного в **ADCM** хоста можно менять. Для этого на вкладке *“HOSTS”* необходимо нажать на иконку шестеренки в строке необходимого хоста. Так, в частности, при использовании хостпровайдера ssh в настройках хоста можно указать приватный ssh ключ в строке *ansible_ssh_private_key_file* (Рис.6.1.).

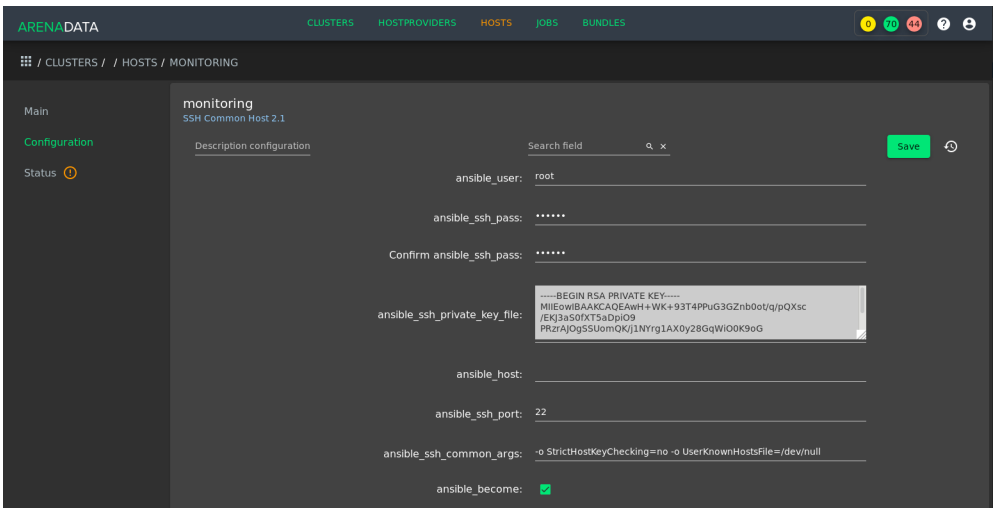



Рис.6.1.: Страница настройки для хоста в ADCM на примере хостпровайдера ssh

# Глава 7

# Загрузка бандла кластера

Дистрибутивы кластеров для **ADCM** выполнены в виде бандлов. В случае кластера **ADH** – это обычный архив, содержащий описание и программную логику развертывания и эксплуатации кластера.

Для загрузки бандла **ADH** необходимо перейти на сайте https://arenadata.io на вкладку *"Скачать"* продукта **Arenadata Hadoop**. При этом происходит переход на страницу Магазин Программного Обеспечения Arenadata и открывается форма заявки для загрузки продукта (Рис.7.1.).

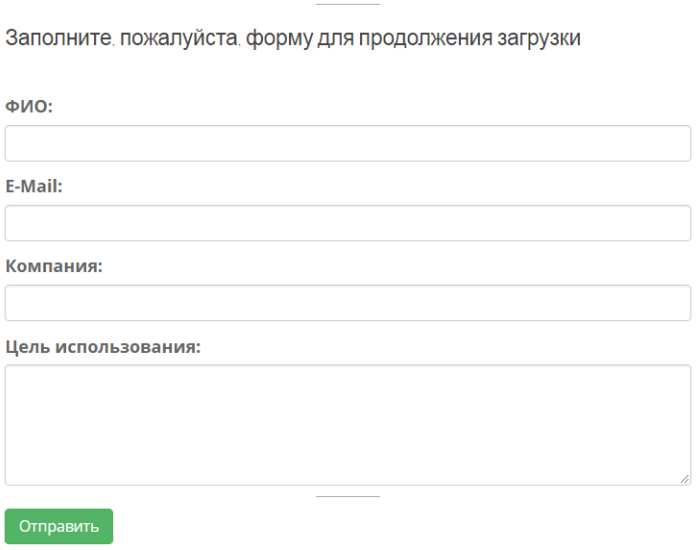



Рис.7.1.: Заявка для загрузки продукта

Все поля являются обязательными для заполнения. После отправки данных происходит переход на страницу *"Магазин Приложений"* с перечнем продуктов **Arenadata** (Рис.7.2.).

Для загрузки дистрибутива следует перейти на страницу продукта, кликнув на продукт **Arenadata Hadoop**. При этом происходит переход на соответствующую страницу с кратким описанием, ссылками на компоненты дистрибутива и информацией о релизе (Рис.7.3.).

После указания версии **ADH** и операционной системы в раскрывающемся списке *"Ссылки на компоненты дистрибутива"* предоставляется выбор вариантов загрузки дистрибутива – по нажатию на элемент списка браузер начинает загрузку. В блоке *"Информация о релизе"* отображается дата выхода, тип релиза выбранной версии, ссылки на лицензию и Release Notes.

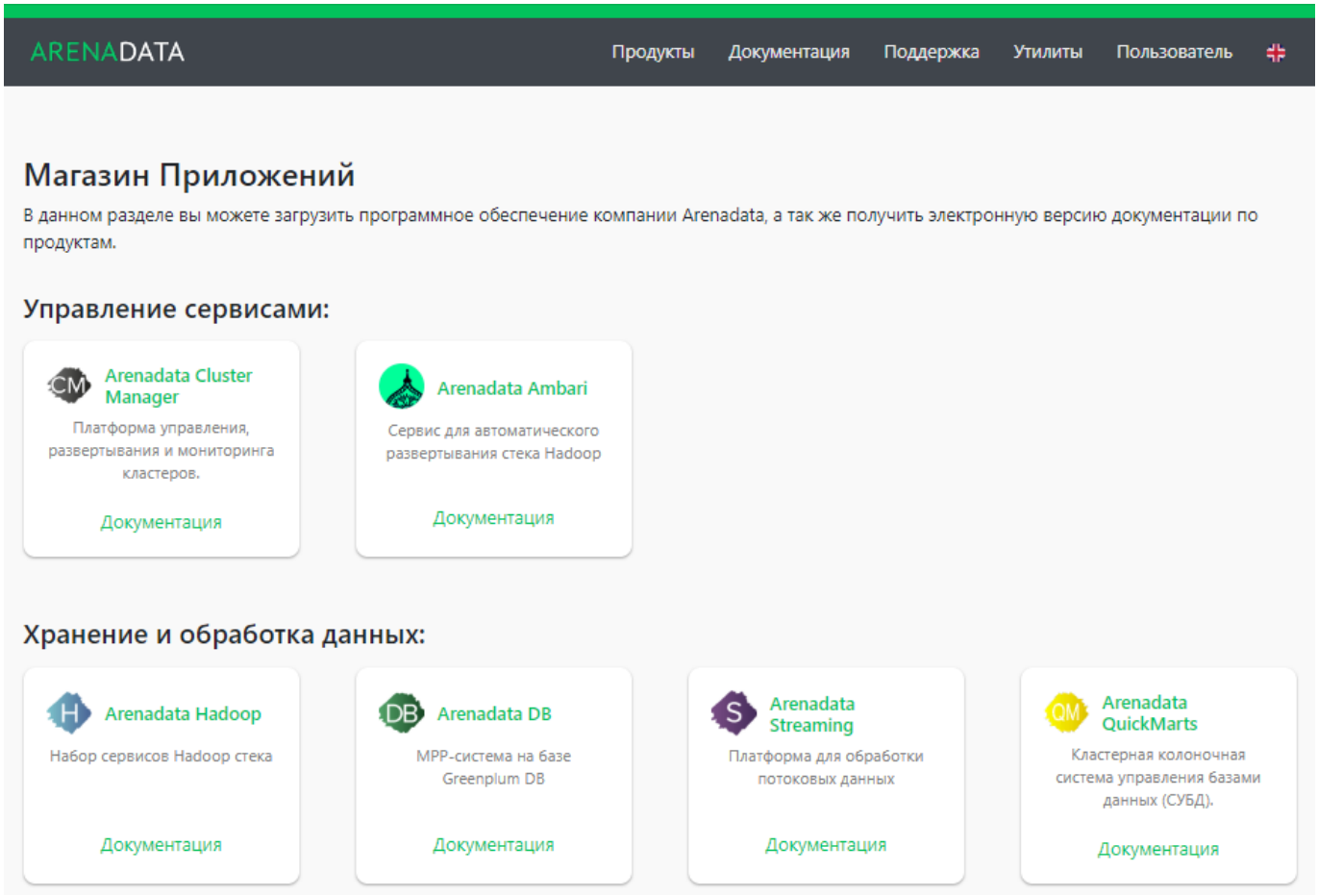



Рис. 7.2.: Магазин Приложений

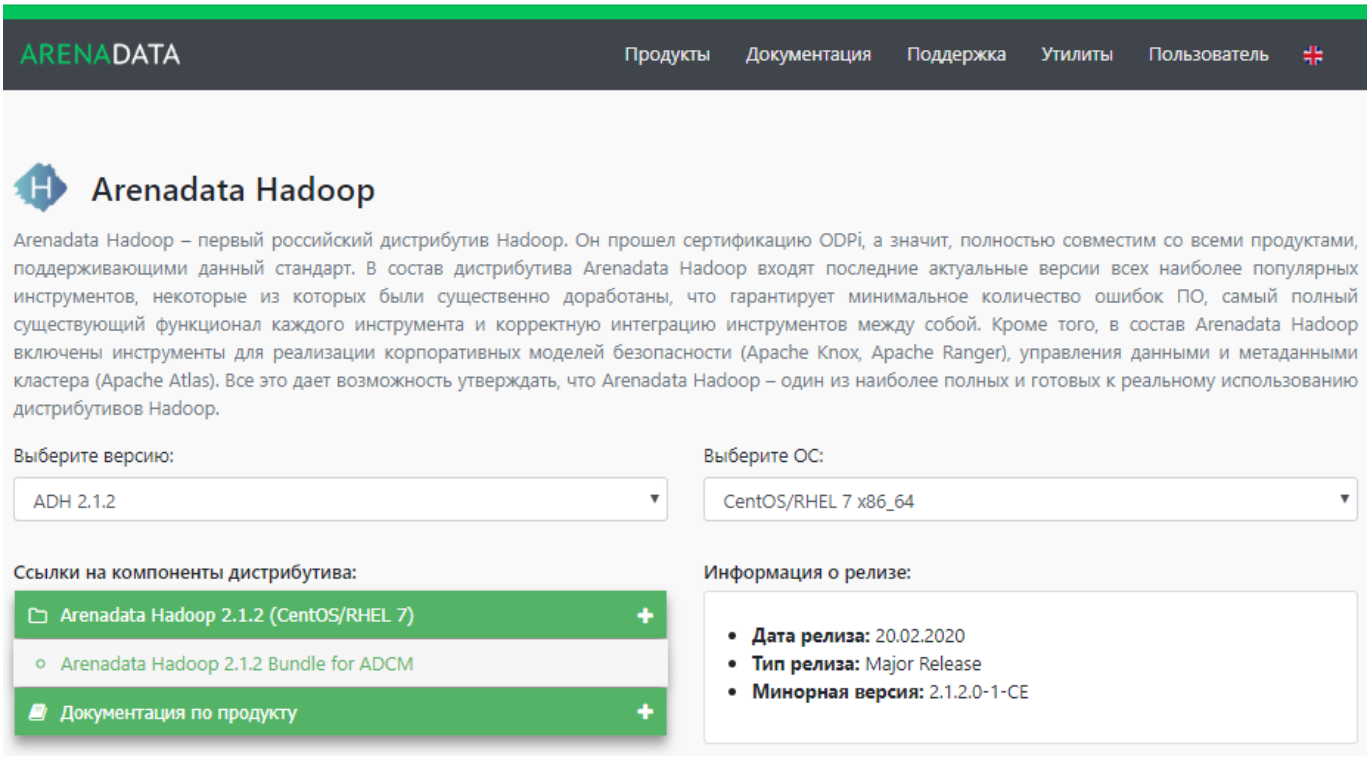



Рис. 7.3.: Страница загрузки Arenadata Hadoop

# Глава 8

# Установка ADH в ADCM

## 8.1 Выгрузка бандла ADH в ADCM

Выгрузка бандла **ADH** необходима для создания в **ADCM** прототипа кластера, из которого в дальнейшем становится возможным создание его экземпляров.

Для выгрузки бандла следует выполнить действия аналогичные *выгрузке бандла хостпровайдера*.

## 8.2 Создание экземпляра кластера

При создании кластера в веб-интерфейсе **ADCM** генерируется новый экземпляр кластера **ADH**, что означает только добавление данных о нем в базу данных **ADCM** – на этом этапе не производится установка **ADH** на хосты.

1. Открыть в ADCM вкладку *“CLUSTERS”* (Рис.8.1.).

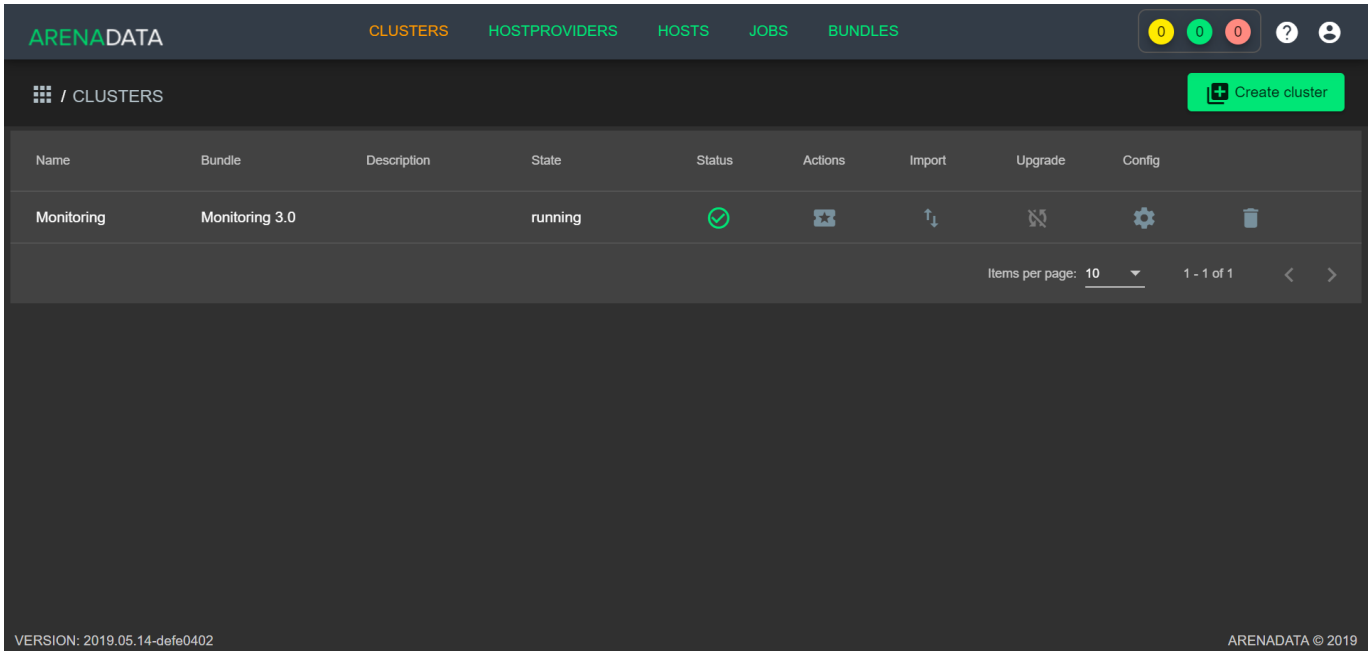



Рис.8.1.: Вкладка “CLUSTERS”

2. Нажать *“Add cluster”* и в открывшейся форме создать экземпляр кластера из прототипа ADH (Рис.8.2.).

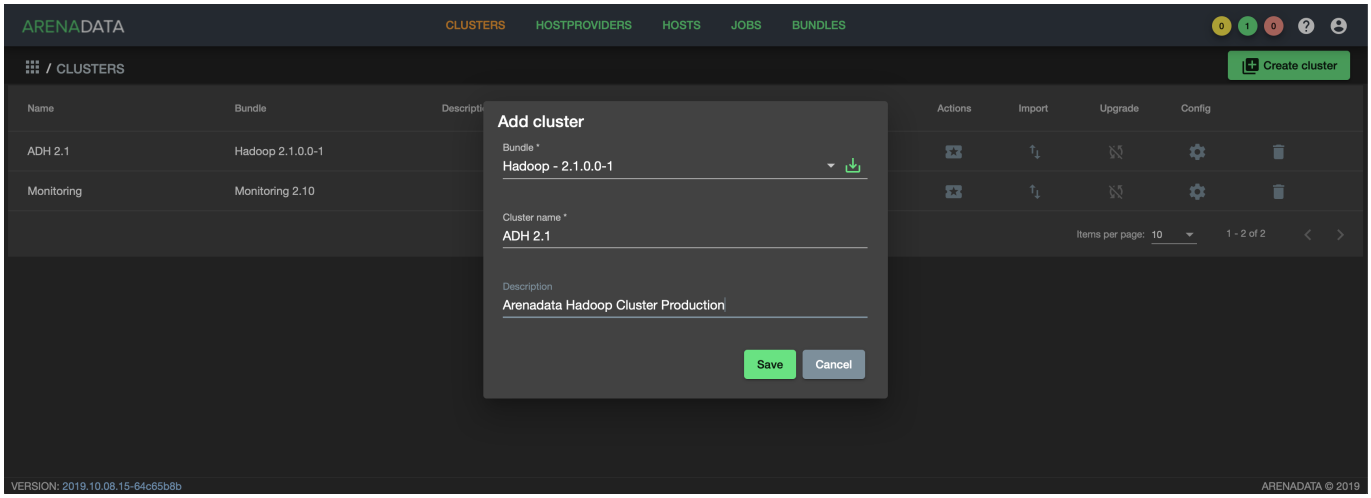



Рис.8.2.: Создание экземпляра кластера

3. В результате экземпляр кластера отражается в списке на вкладке *“CLUSTERS”* (Рис.8.3.).

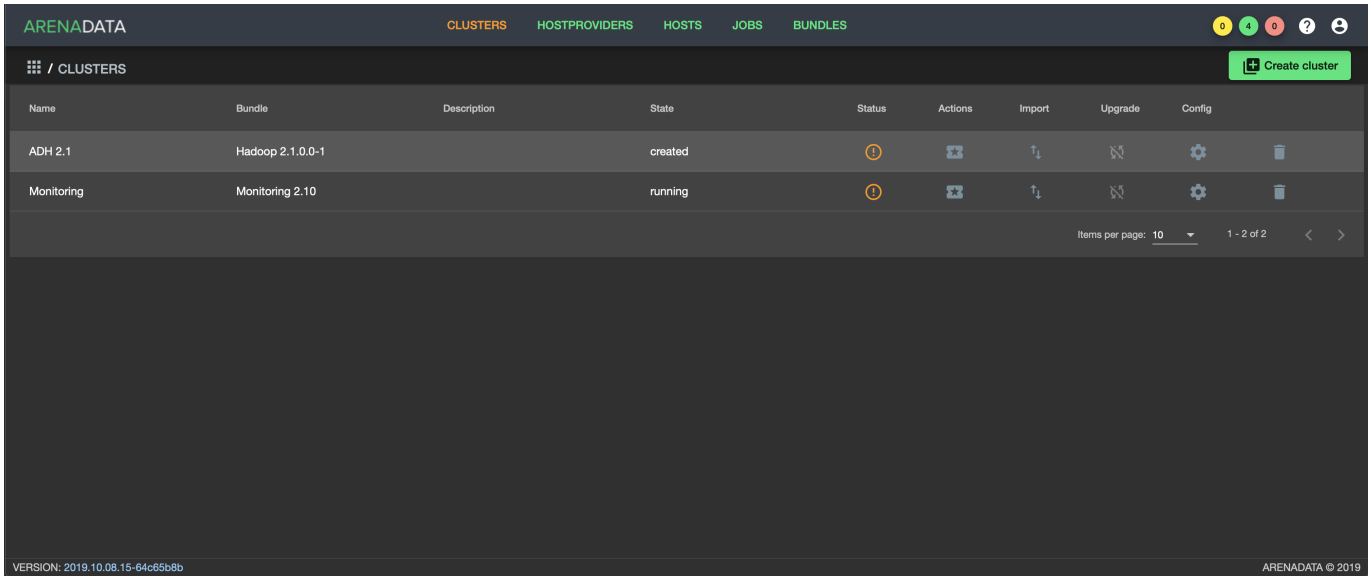



Рис.8.3.: Результат успешного создания экземпляра кластера

## 8.3 Добавление сервисов кластера

**Important:** На текущий момент невозможно удалить из кластера уже добавленный сервис

Для добавления сервисов в кластер **ADH** необходимо (Рис.8.4.):

1. В меню кластера *ADH* открыть вкладку *“Services”*;
2. Нажать *“Add service”*;

3. В открывшейся форме выбрать необходимые сервисы;
4. Нажать *“Save”*.

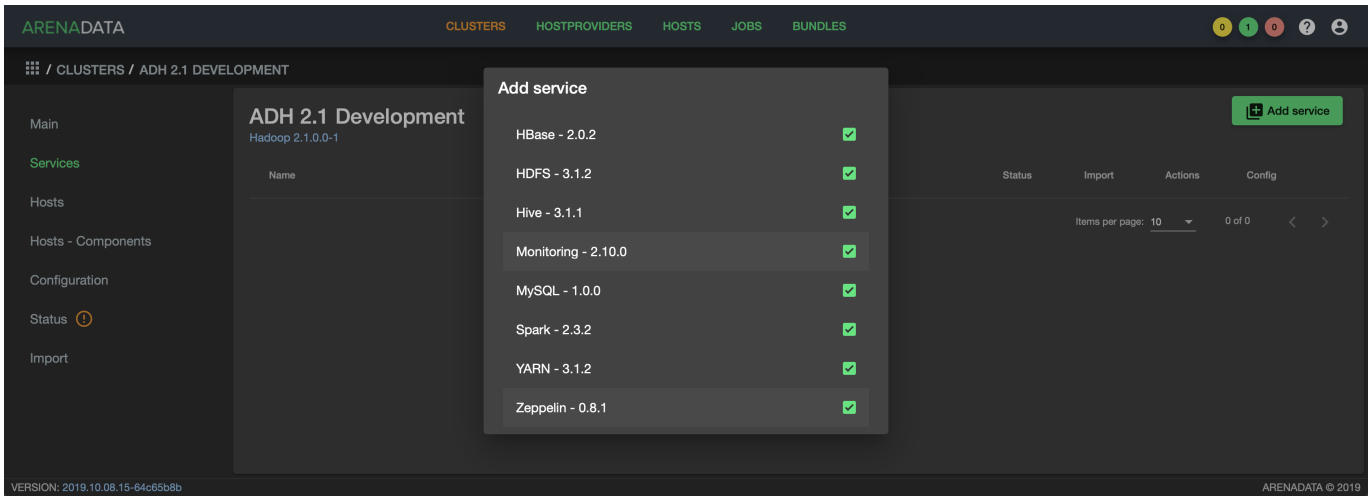



Рис.8.4.: Выбор сервисов

Возможность добавления нового сервиса в уже работающий кластер не отличается от установки сервиса с нуля.

## 8.4 Конфигурирование сервиса

Для перехода к настройкам сервиса кластера необходимо нажать кнопку с пиктограммой шестеренки в соответствующей строке вкладки пункта меню *“Services”* (Рис.8.5.).

По завершении конфигурирования сервиса необходимо нажать *“Save”*.

В случае, если сервису требуется изначальная настройка, то в поле *“Actions”* соответствующего сервиса вместо иконки отражается оранжевый восклицательный знак (Рис.8.6.).

Это означает, что необходимо открыть страницу конфигурирования сервиса и заполнить поля, выделенные оранжевым (Рис.8.7.).

## 8.5 Добавление хостов

Для добавления хостов в кластер **ADH** необходимо:

1. В меню кластера *ADH* открыть вкладку *“Hosts”* (Рис.8.8.).
2. Нажать *“Add hosts”* и в открывшейся форме выбрать необходимые хосты (Рис.8.9.).
3. В результате добавленные хосты отображаются в списке вкладки *“Hosts”* (Рис.8.10.).

## 8.6 Размещение компонентов сервисов на хостах

Изначально ни на одном из хостов нет компонентов – распределение компонентов осуществляется вручную.

Для размещения компонентов необходимо перейти на вкладку *“Hosts - Components”* (Рис.8.11.).

И распределить компоненты одним из двух способов:

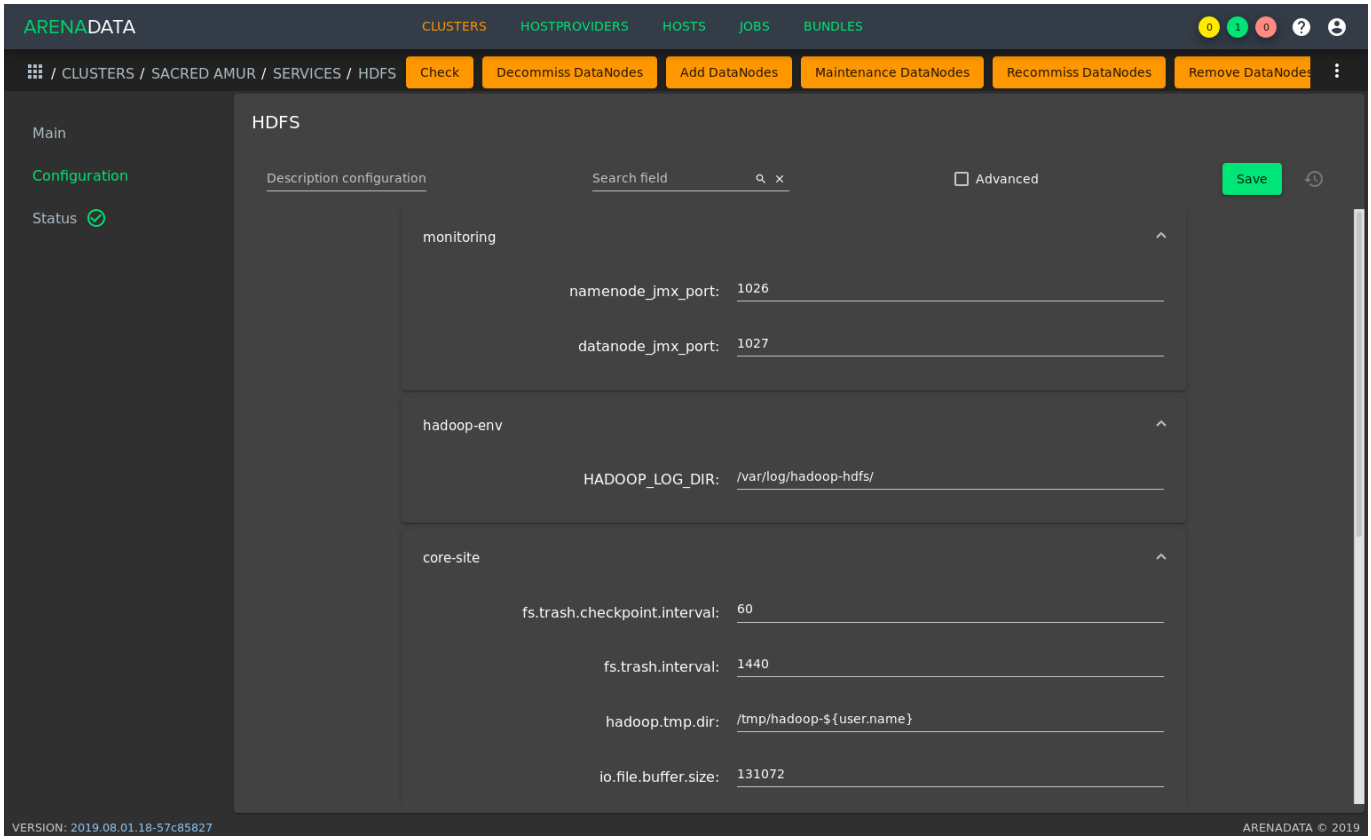



Рис.8.5.: Настройка сервиса

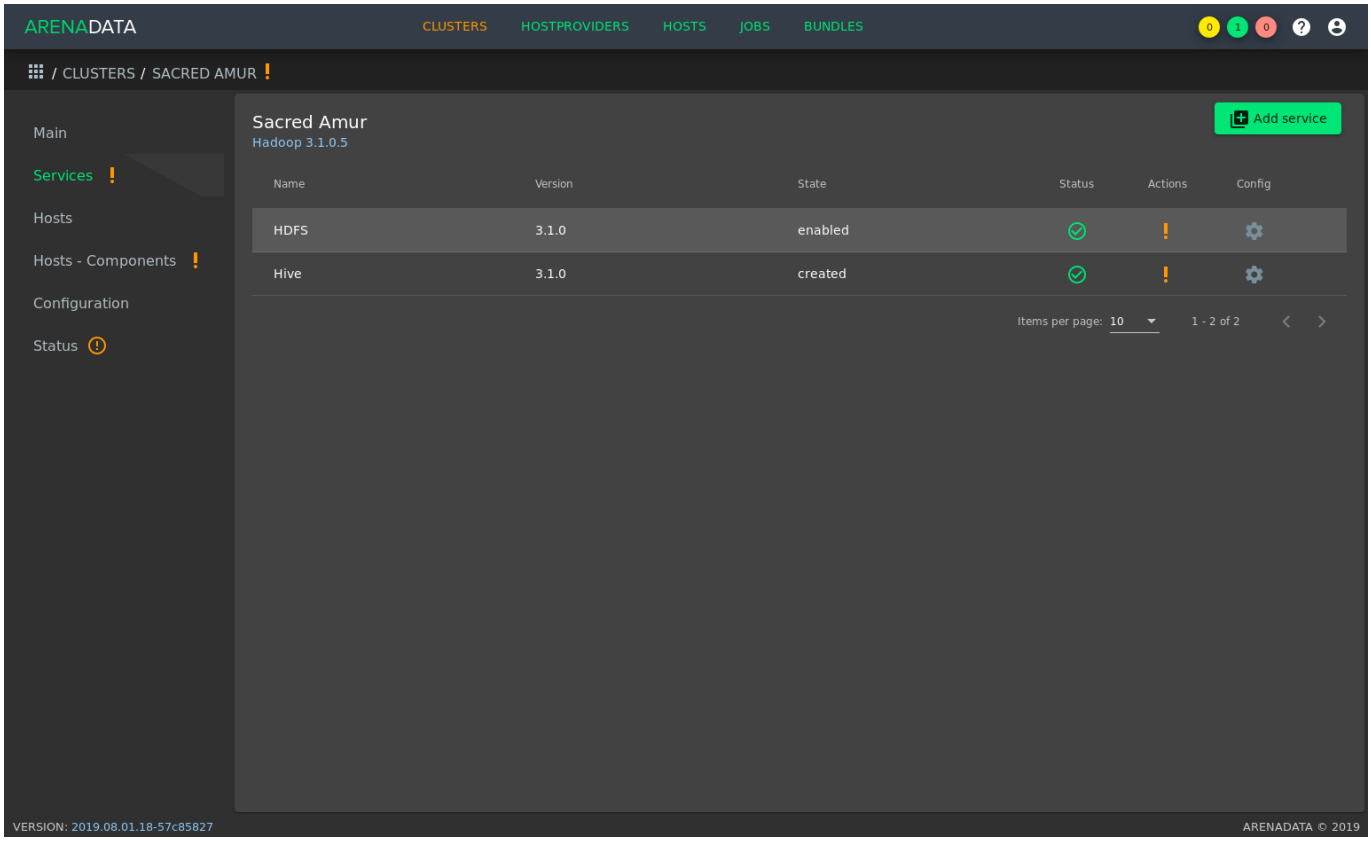



Рис.8.6.: Сервисы, требующие настройки

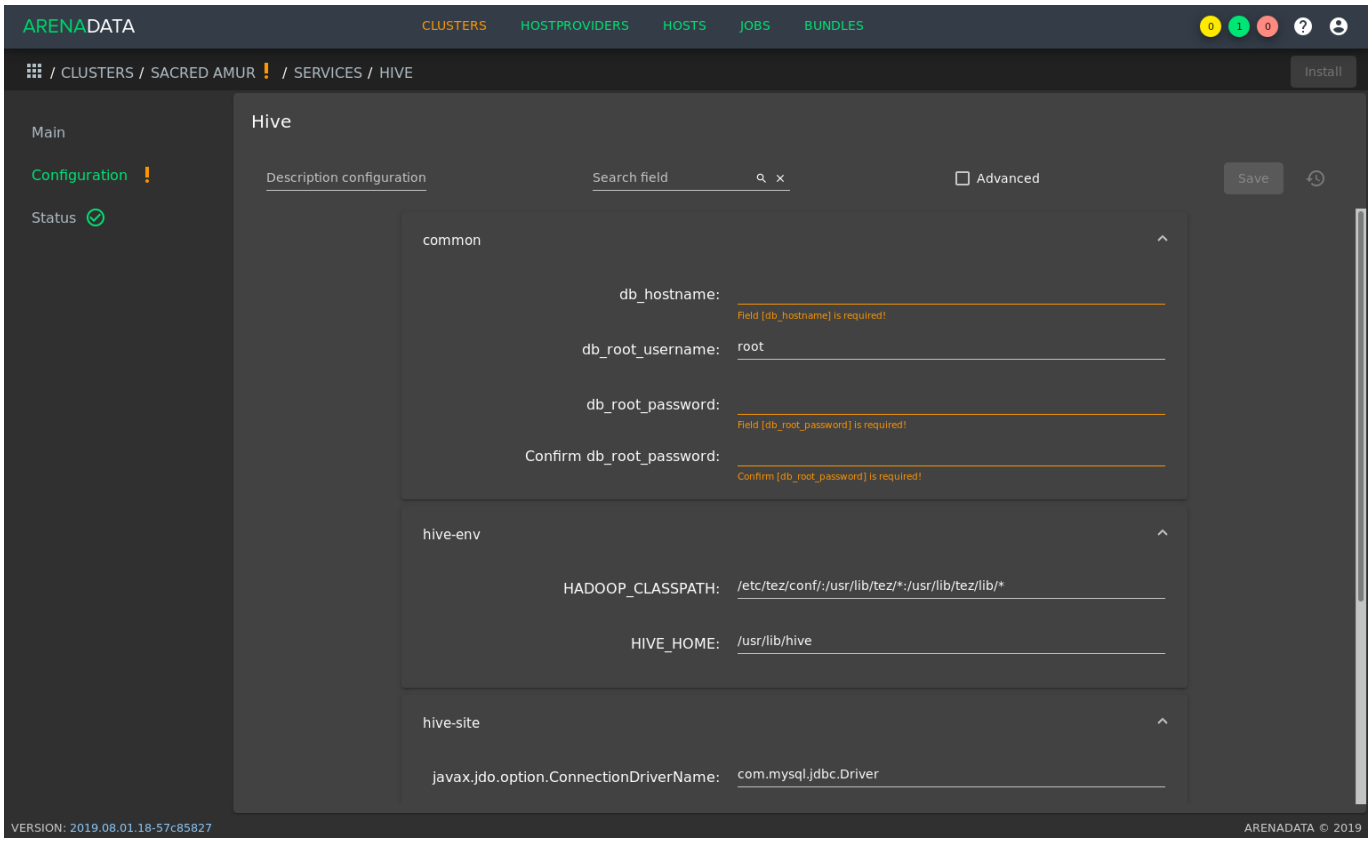



Рис.8.7.: Конфигурация сервиса, требующего настройки

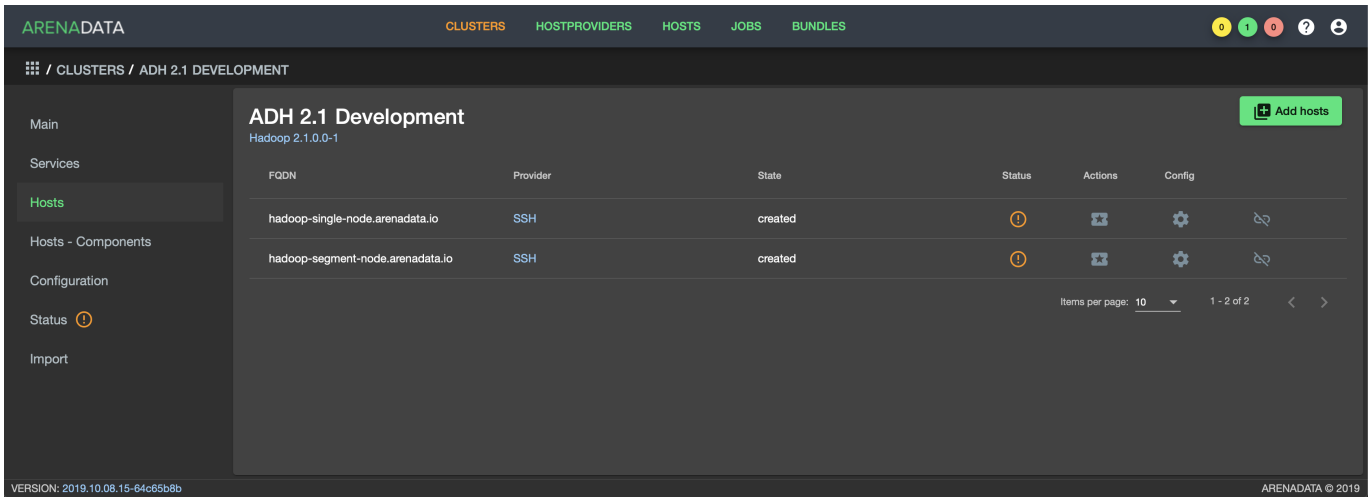



Рис.8.8.: Вкладка “Hosts” кластера ADH

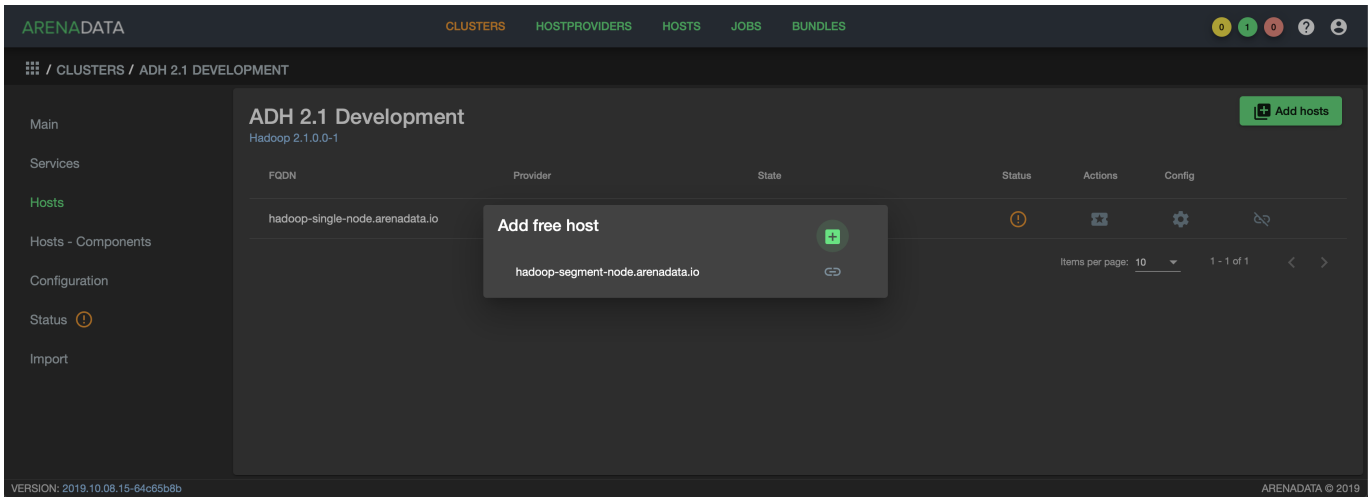



Рис.8.9.: Выбор хостов

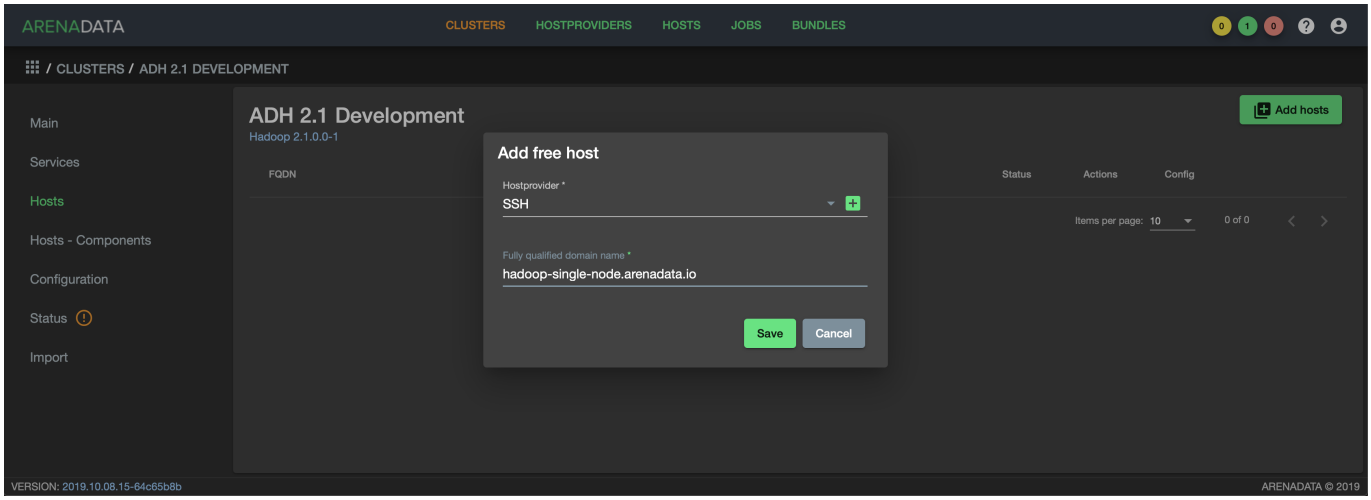



Рис.8.10.: Результат успешного добавления хостов

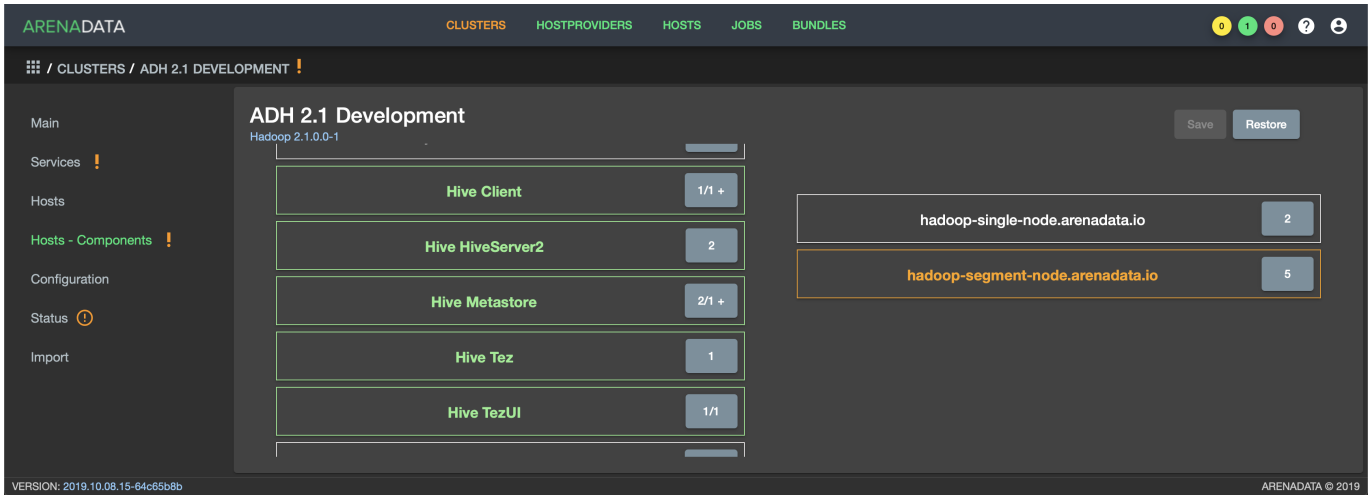



Рис.8.11.: Размещение компонентов сервисов на хостах

1. Выбрать компонент в колонке “Components” и определить для него хост в колонке “Hosts”;
2. Выбрать хост в колонке “Hosts” и определить для него компонент в колонке “Components”.

**Important:** В сервисе могут быть обязательные и необязательные компоненты. Если компонент обязательный, то в его количественном счетчике справа от названия присутствует символ ‘/’. Без назначения хоста обязательному компоненту карту размещения сервисов нельзя сохранить

По завершении распределения хостов необходимо нажать *“Save”*.

## 8.7 Конфигурирование кластера

Для перехода к настройкам экземпляра кластера **ADH** необходимо нажать кнопку с пиктограммой шестеренки в соответствующей строке вкладки *“CLUSTERS”* и перейти в раздел меню *“Configuration”*. При этом открывается окно конфигурации выбранного экземпляра (Рис.8.12.).

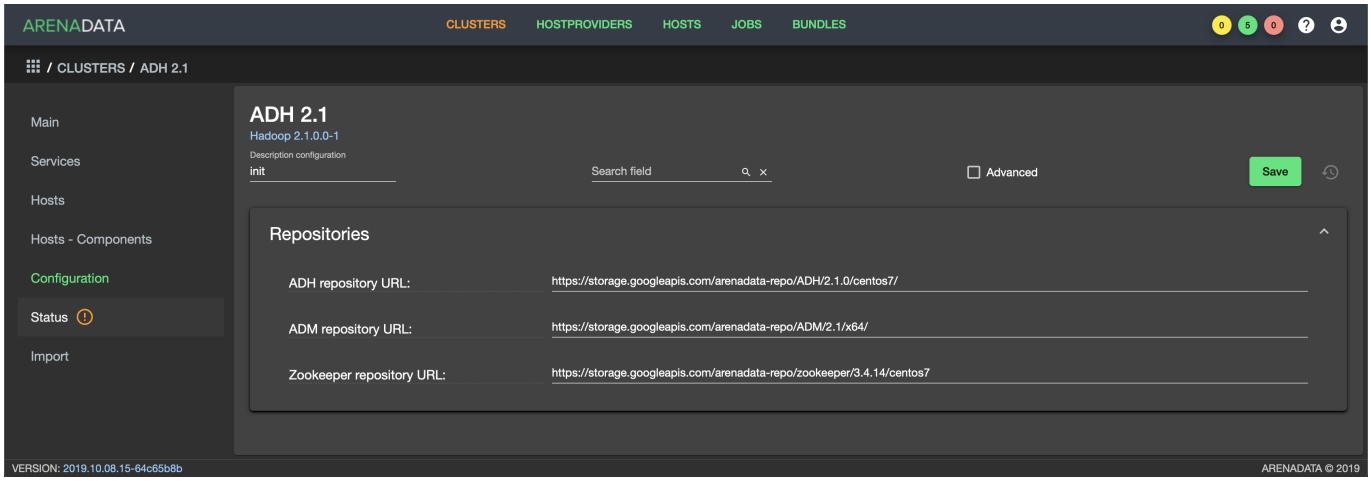



Рис.8.12.: Окно конфигурирования кластера

По завершении конфигурирования кластера необходимо нажать *“Save”*.

## 8.8 Установка сервисов кластера

**Important:** Порядок установки сервисов и зависимости между ними на данный момент не ограничивается со стороны ADCM

Устанавливать сервисы необходимо в следующем порядке:

1. Zookeeper
2. HDFS
3. YARN
4. HBase
5. Hive
6. Spark

7. Monitoring

Для установки добавленного сервиса необходимо в строке нужного сервиса нажать на пиктограмму в поле *“Actions”* и выбрать действие *Install*. После этого **ADCM** запускает задачу установки. Более конкретно о статусе и информации о задачах можно узнать на вкладке *“JOBS”* (Рис.8.13.).

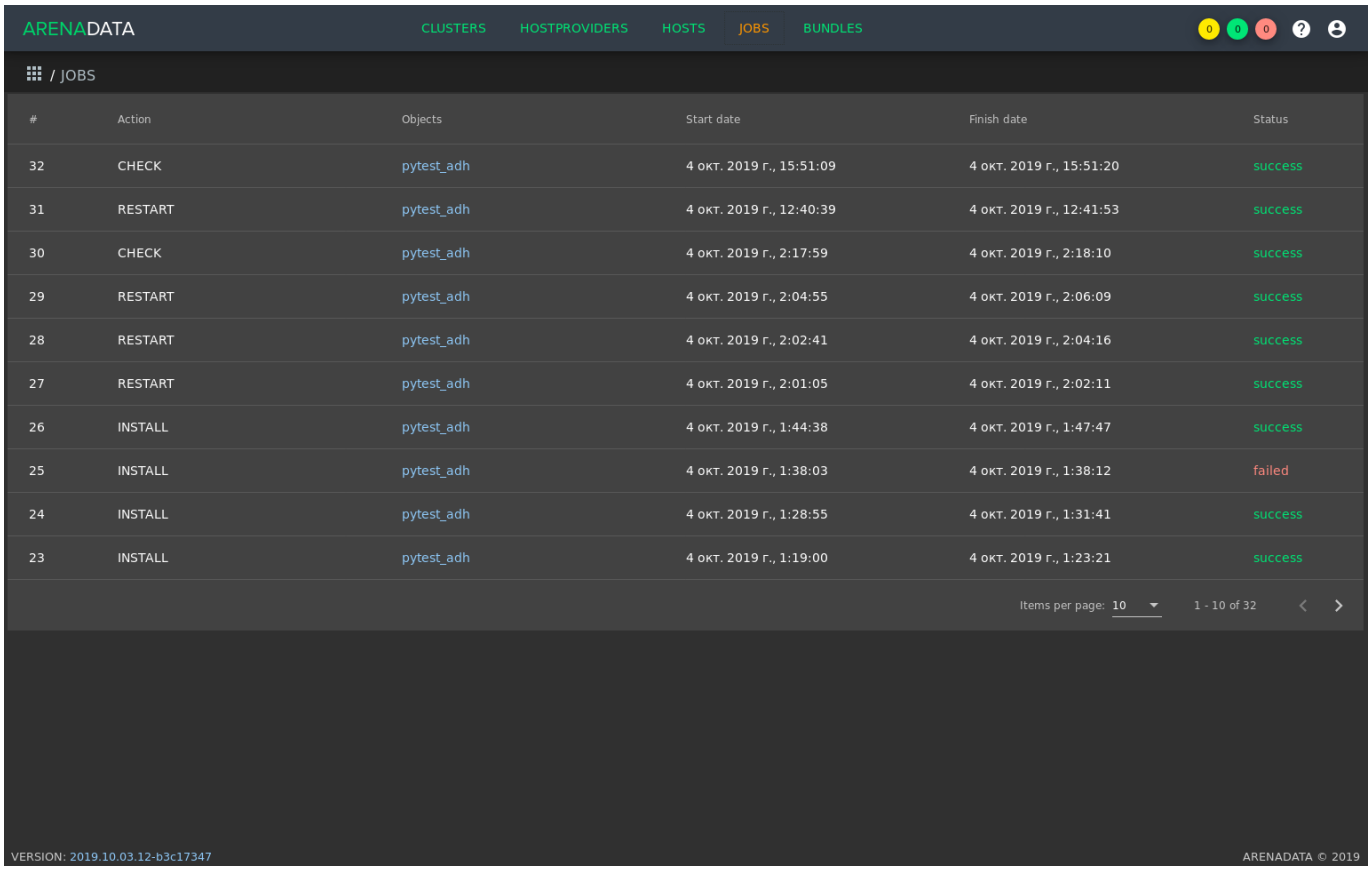



Рис.8.13.: Вкладка “JOBS”

**Important:** Статус задач ADCM отражается в правом верхнем углу web-интерфейса. Желтый круг отражает количество запущенных задач, а зеленый и красный – количество успешно и неуспешно завершенных задач соответственно

Успешное завершение установки сервиса определяется переходом задачи из статуса *running* в статус *success* на вкладке *“JOBS”*. При неудачном завершении задача переходит в статус *failed*. При таком исходе возможно нажать на строку задачи на странице вкладки *“JOBS”* для получения более подробной информации о возникших ошибках (Рис.8.14.).

На странице задачи в левой части экрана обязательно содержится 2 раздела: *“№-ansible-out.txt”* и *“№-ansible-out.txt”*, где № является номером задачи. Это технические логи задачи, которые могут помочь в определении причины проблем.

Также может существовать опциональный третий раздел *“№-check-out.json”* – это логи проверок частых ошибок, описание этих ошибок более простое и конкретное, чем в случае первых двух технических логов.

Содержимое всех трех разделов подлежит изучению при возникновении ошибок.

После установки сервис запускается автоматически, кроме сервиса *Zookeeper* – его требуется запустить

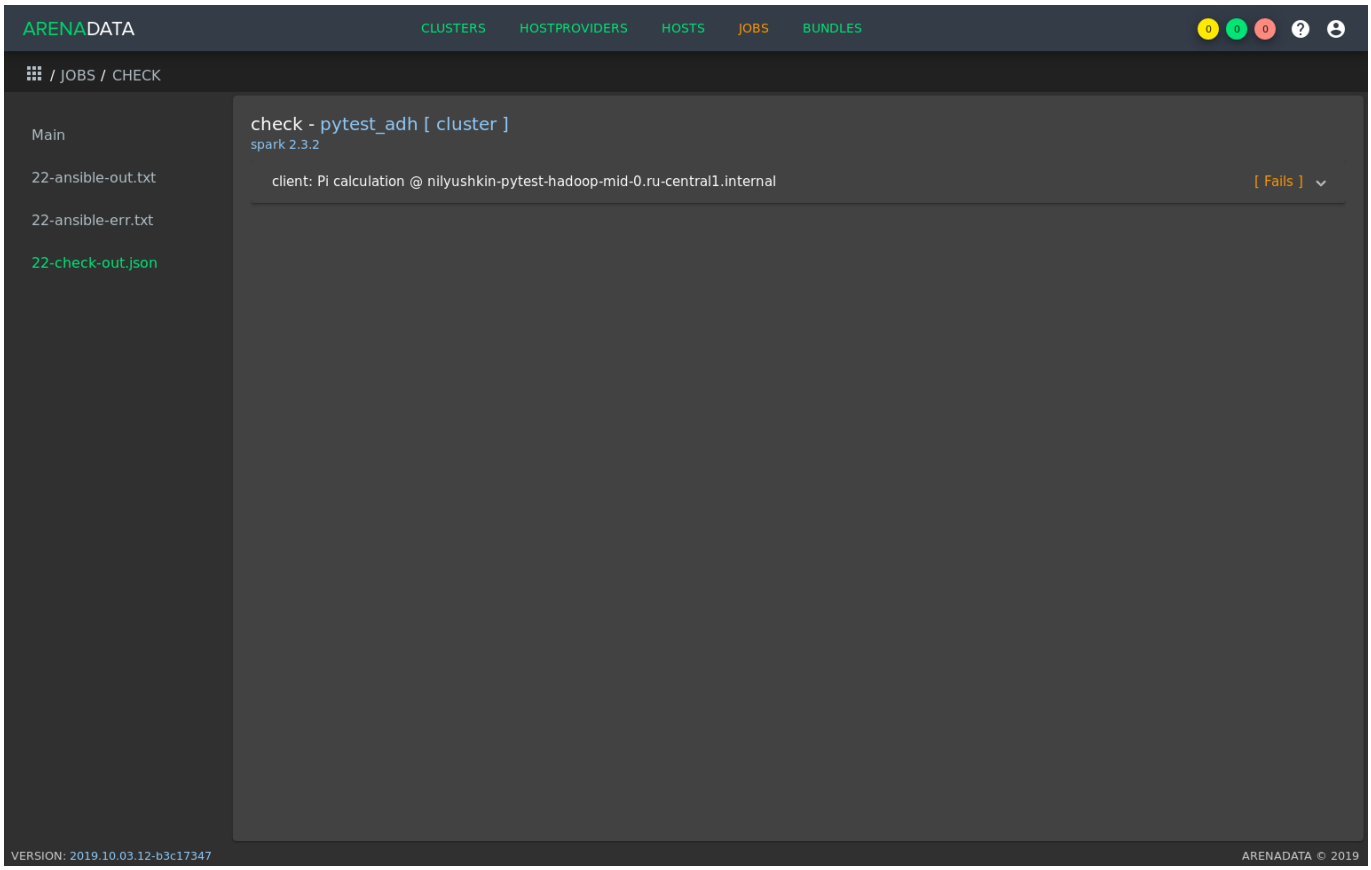



Рис.8.14.: Страница конкретной задачи

вручную нажатием кнопки *“Start”*.

**Important:** Сервис Zookeeper требует ручного запуска

По результатам инсталляции сервис меняет состояние (поле *“State”*) с *created* на *installed* (Рис.8.15.).

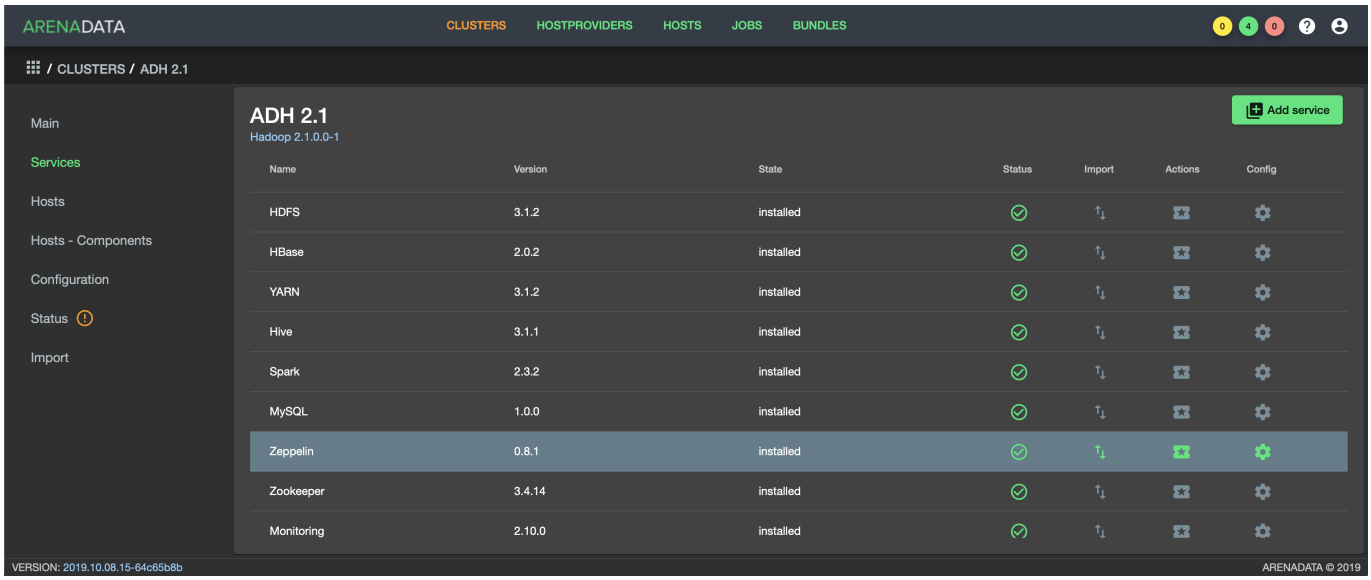



Рис.8.15.: Состояние сервисов кластера

### 8.8.1 Особенности установки сервисов

#### Hive

Компонент *Metastore* сервиса *Hive* требует наличия БД *MySQL/MariaDB* и доступов к ней.

Сервис *MySQL/MariaDB* в бандл **ADH** не входит, его необходимо устанавливать и настраивать другими средствами.

**Important:** Помимо настройки самого сервиса MySQL/MariaDB для использования его сервисом Hive необходимо также настроить соответствующие доступы (GRANT)

Так, самый простой случай выдачи доступов выглядит следующим образом:

```
MariaDB [(none)]> GRANT ALL ON *.* TO 'root'@'<metastore_fqdn>' identified by '<password>' WITH GRANT␣
↪OPTION;
```

Где `<metastore_fqdn>` и `<password>` необходимо заменить реальными данными конкретного сервера *MySQL/MariaDB*.

Указать сервису *Hive* адрес и учетные данные для подключения к БД можно в *настройках сервиса*.

#### Monitoring

Сервис *Monitoring* кластера **ADH** требует установки отдельного кластера Monitoring для целей интеграции в него кластера **ADH**.

После установки кластера *Monitoring* его необходимо интегрировать в кластер **ADH**. Для этого необходимо перейти в раздел меню *"Import"* кластера **ADH** (Рис.8.16.).

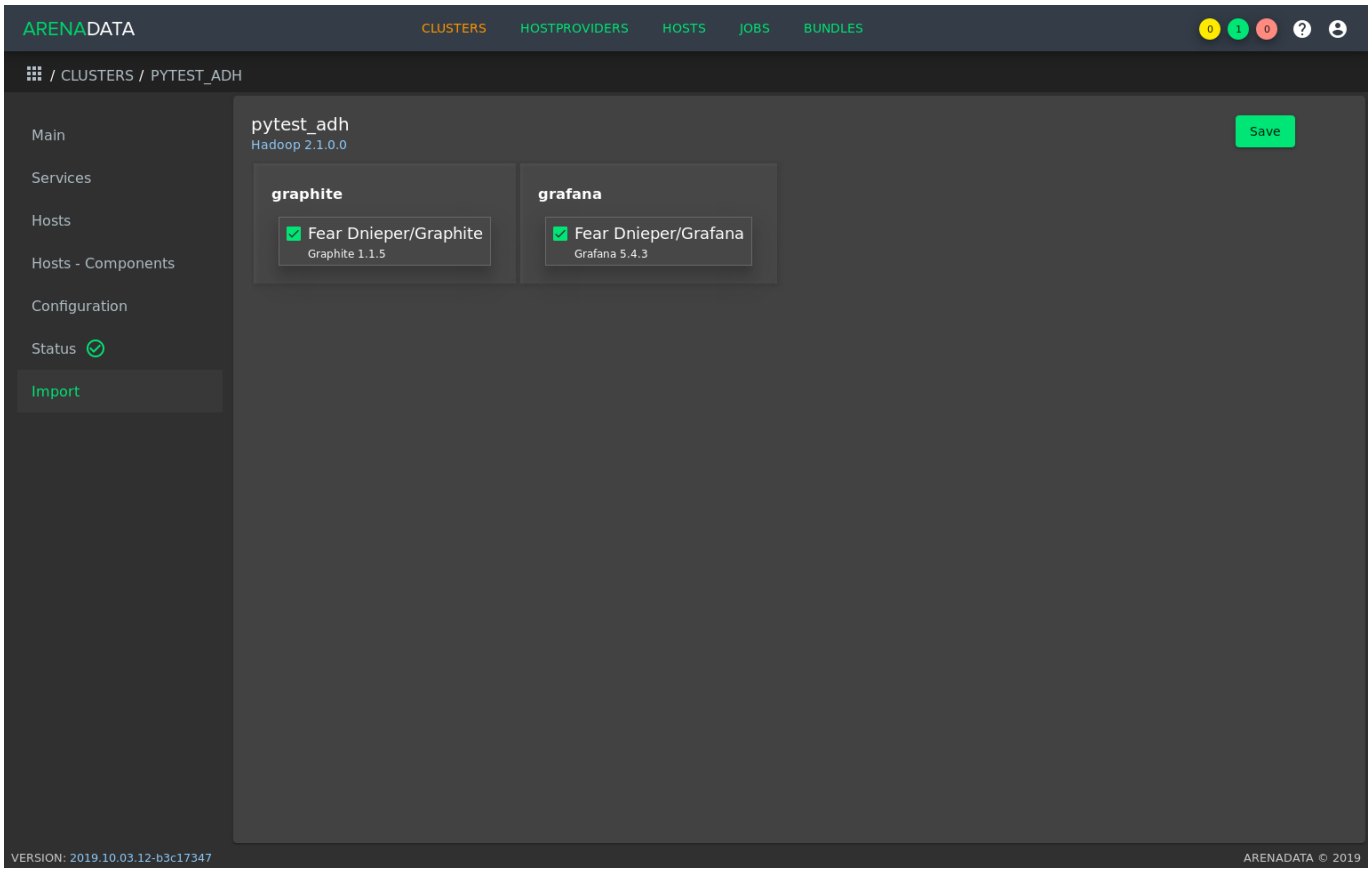



Рис.8.16.: Раздел меню "Import"

Далее необходимо отметить флагами оба импортируемых сервиса кластера *Monitoring*: *Graphite* и *Grafana*, и нажать *"Save"* в верхней правой части страницы.

После интеграции сервису *Monitoring* кластера **ADH** становятся доступными средства для мониторинга некоторых сервисов кластера:

- HDFS;
- YARN;
- HBase;
- Spark.

Для использования этих средств необходимо установить сервис, предназначенный для мониторинга, и сервис *Monitoring* кластера **ADH**.

**Important:** В случае, если сервис Monitoring установлен после сервиса, предназначенного для мониторинга, необходим рестарт этого сервиса