Arenadata™ Hadoop

Версия - v1.6.1

Инструкция по установке кластера Arenadata Hadoop

2019 Arenadata™

Оглавление

1 Руководство по планированию кластера **3**
- 1.1 Рекомендации по аппаратному обеспечению для Apache Hadoop . . . 3
- 1.2 Рекомендации по партиционированию файловой системы . . . 11

2 Загрузка дистрибутива кластера **13**

3 Подготовка к установке кластера **17**
- 3.1 Системные требования . . . 17
- 3.2 Сбор информации . . . 21
- 3.3 Подготовка к установке Hadoop . . . 21

4 Установка сервера Ambari 2.6.2 **25**
- 4.1 Настройка репозитория . . . 25
- 4.2 Установка сервера Ambari . . . 27
- 4.3 Настройка сервера Ambari . . . 28
- 4.4 Запуск сервера Ambari . . . 28
- 4.5 Подготовка к установке основных компонентов ADH на кластер . . . 29

5 Запуск мастера установки **32**
- 5.1 Изменение URL-адресов репозиториев . . . 32
- 5.2 Ввод имен узлов и SSH-ключа . . . 32
- 5.3 Выбор компонентов . . . 35
- 5.4 Назначение мастер-узлов . . . 35
- 5.5 Назначение Slave и Client узлов . . . 36
- 5.6 Дополнительные настройки компонентов . . . 36
- 5.7 Установка, запуск и тестирование . . . 38

6 Панель кластера **41**

7 Ручная настройка подключения к базе данных **42**

В документации приведены необходимые сведения для подготовки к установке кластера Arenadata Hadoop, к его планированию, настройке и запуску.

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

Important: Контактная информация службы поддержки – e-mail: info@arenadata.io

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, свзязанные 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 коэффицента 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 дисков SATA емкостью *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/* – ...

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

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.

Глава 2

Загрузка дистрибутива кластера

Для загрузки дистрибутива кластера **Arenadata Hadoop** необходимо на сайте arenadata.io выбрать вкладку *"Скачать"*. При этом происходит переход на страницу *Магазин Программного Обеспечения Arenadata* (store.arenadata.io, Рис.2.3.), и открывается экранная форма заявки для загрузки продукта (Рис.2.1.).

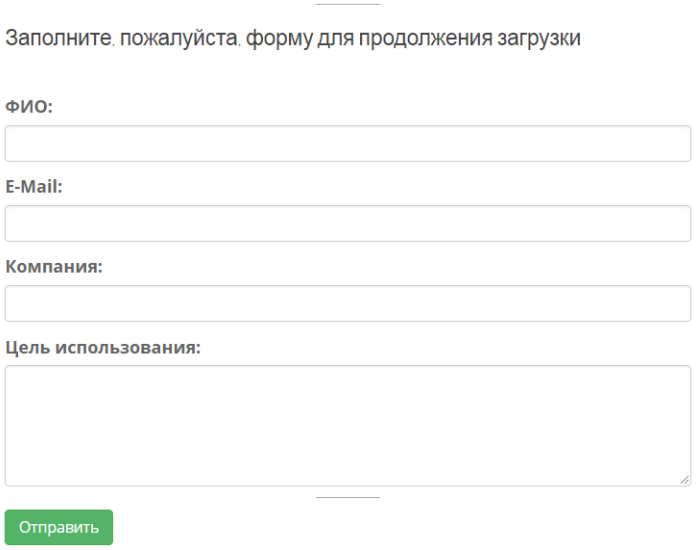


Рис.2.1.: Заявка для загрузки продукта

В экранной форме заявки следует заполнить следующие поля:

- *ФИО* – ФИО пользователя;
- *E-Mail* – адрес электронной почты пользователя;
- *Компания* – наименование компании пользователя;
- *Цель использования* – описание цели использования кластера.

Все поля являются обязательными для заполнения. После ввода данных необходимо нажать кнопку *Отправить*, в результате чего выдается соответствующее сообщение (Рис.2.2.).

При закрытии окна происходит переход на страницу *Магазин Приложений* с перечнем продуктов **Arenadata** (Рис.2.3.).

При наведении курсора на продукт **ARENADATA HADOOP** предоставляется выбор действий:



Рис.2.2.: Заявка принята

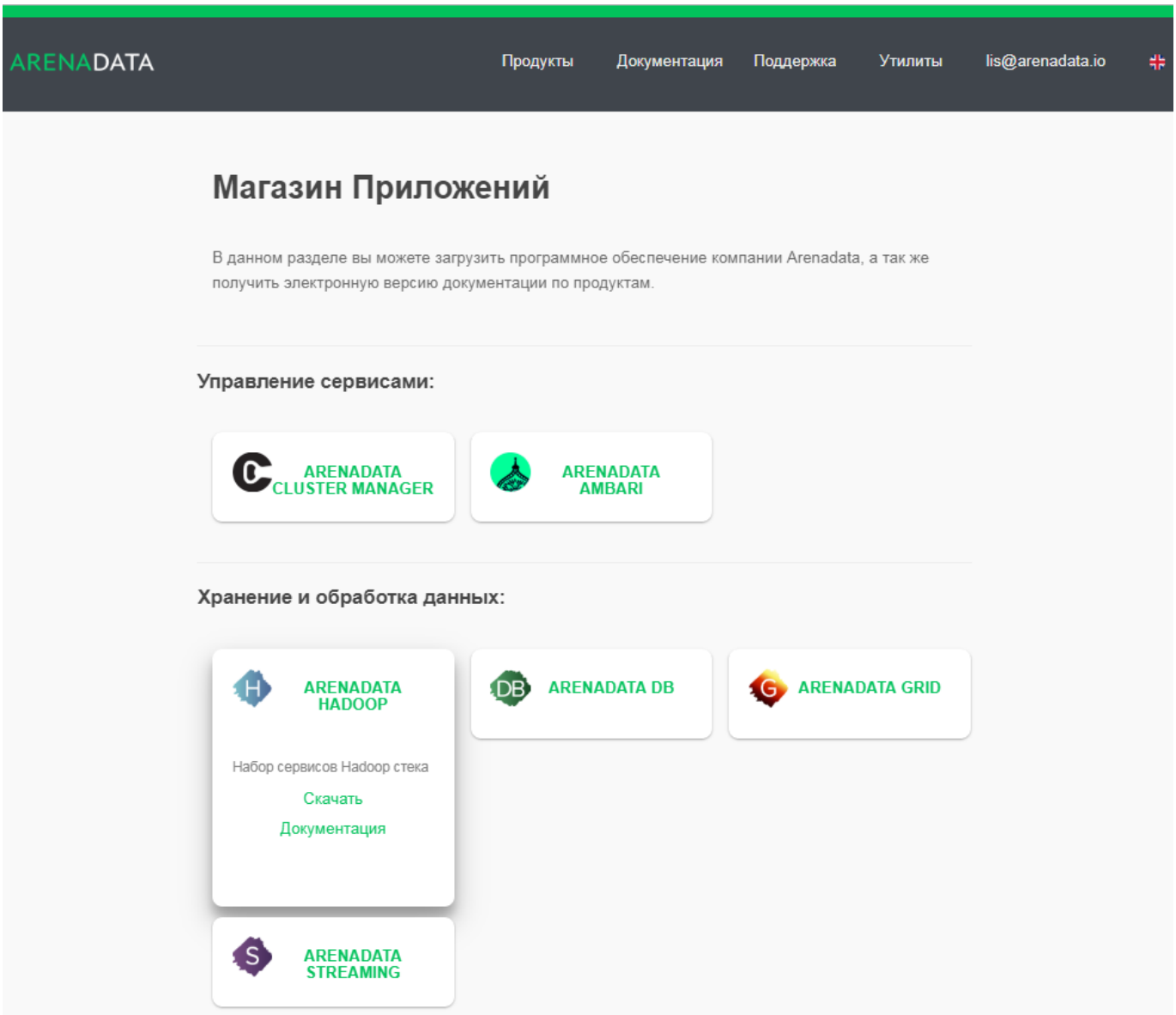


Рис.2.3.: Магазин Приложений

- *Скачать* – переход на страницу загрузки дистрибутива кластера **ADH**;
- *Документация* – переход на страницу он-лайн документации кластера **ADH**.

Для загрузки дистрибутива кластера следует выбрать пункт *Скачать*, при этом происходит переход на соответствующую страницу с кратким описанием продукта, ссылками на компоненты дистрибутива и информацией о релизе (Рис.2.4.).

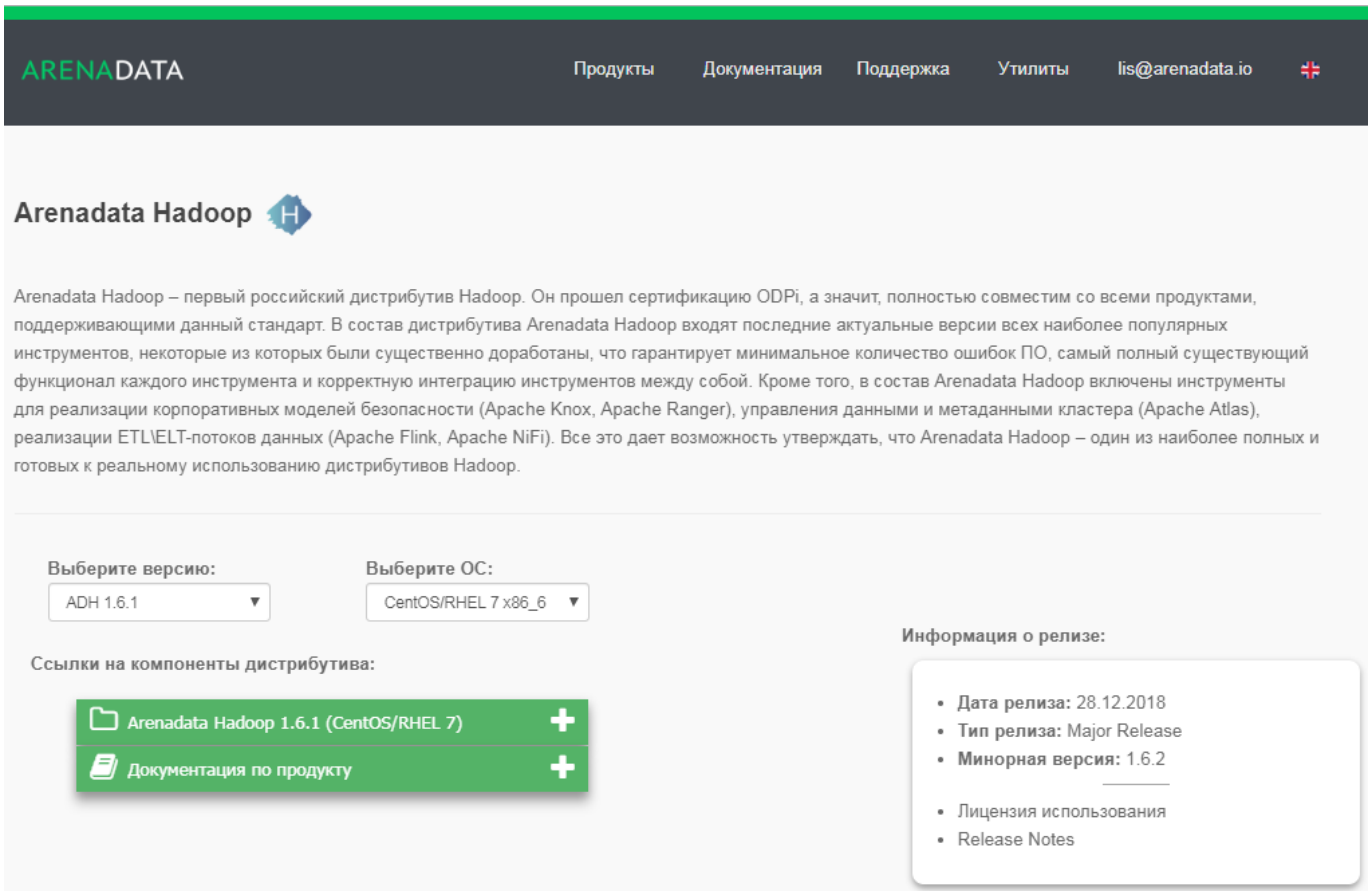


Рис.2.4.: Страница загрузки кластера Arenadata Hadoop

В разделе *Выберите версию продукта* следует указать интересующую версию кластера **ADH**, а в поле *Выберите ОС* – используемую операционную систему. При этом в разделе *Информация о релизе* отображается дата выхода и тип релиза выбранной версии, ее номер, ссылки на лицензию использования и Release Notes.

В разделе *Ссылки на компоненты дистрибутива* находятся ссылки для загрузки компонентов выбранной версии кластера. После загрузки компонентов следует установить их, действуя в соответствии с последующими разделами настоящего документа.

Так же в Магазине Приложений на вкладке *Утилиты* есть ссылка *Калькулятор Hadoop* – страница калькулятора для расчета необходимого количества серверов заданной конфигурации для хранения и обработки указанного объема пользовательских данных в **HDFS** (Рис.2.5.).

На странице калькулятора можно задать персональные параметры в соответствующих полях для расчета необходимого количества серверов заданной конфигурации для хранения и обработки указанного объема пользовательских данных в **HDFS**.

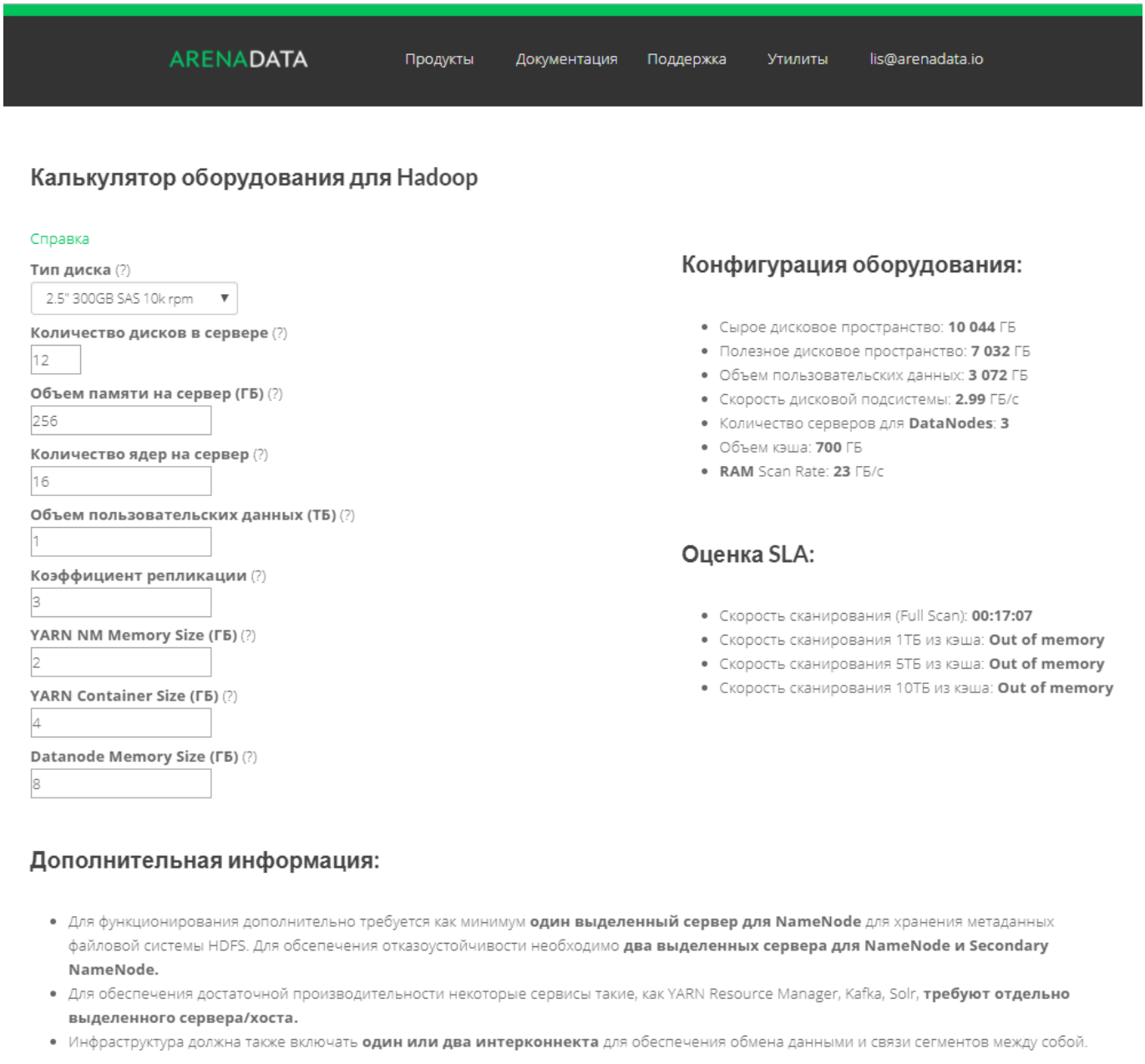


Рис.2.5.: Калькулятор оборудования для Hadoop

Глава 3

Подготовка к установке кластера

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

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

Ambari обеспечивает комплексное решение для управления и мониторинга кластера **ADH**. Используя API-интерфейс **Ambari Web UI** и **REST**, можно развертывать, разрабатывать, управлять изменениями конфигурации и контролировать службы всех узлов кластера с центральной точки.

3.1 Системные требования

Для корректного запуска **Hadoop** необходимо соблюдение минимальных системных требований, представленных в таблице, а также описанных в разделах *Требования к памяти*, *Требования к объему пакета и количеству индексных дескрипторов* и *Проверка максимального количества открытых файловых дескрипторов*.

Таблица3.1.: Минимальные системные требования для установки кластера Arenadata Ambari

Компонент	Требования
Платформа	• Intel x86_64 • IBM Power (Little-Endian)

Компонент	Требования
Операционная система	Поддерживаются следующие 64-разрядные ОС: • Red Hat Enterprise Linux (RHEL) v7.x; • CentOS v7.x; • Suse Enterprise Linux 12sp3 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. В случае возникновения проблем с недоступными базовыми хранилищами ОС, необходимо обратиться к системному администратору для обеспечения проксирования или зеркалирования этих дополнительных репозиториев
Браузер	Мастер установки Ambari работает как веб-приложение на основе браузера. Поэтому необходимо наличие машины, способной использовать графический браузер для применения данного веб-инструмента. Минимальные требуемые версии браузера: • Internet Explorer 9.0 (устаревшая версия); • Firefox 18; • Google Chrome 26; • Safari 5; На платформе рекомендуется обновить браузер до последней, стабильной версии (кроме Internet Explorer 9.0)
Программное обеспечение	На всех узлах кластера необходимо установить следующие компоненты: • YUM и rpm (RHEL / CentOS); • Zypper и php_curl (SLES); • Scp, curl, unzip, tar и wget; • OpenSSL (v1.01, build 16 или новее); • Python v2.7
Java	Поддерживаются следующие среды выполнения Java: • Oracle JDK 1.8 64-разрядный: минимум JDK 1.8.64;

Компонент	Требования
База данных	Ambari требует реляционную базу данных для хранения информации о конфигурации кластера и топологии. При установке ADH Stack с Hive или Oozie – для них также требуется реляционная база данных • Ambari: – PostgreSQL 8; – PostgreSQL 9.1.13+, 9.3; – MariaDB 5.5; – Oracle 11g R2, 12c. По умолчанию Ambari устанавливает инстанс PostgreSQL на хост сервера Ambari. При этом использование данного инстанса PostgreSQL, MySQL или Oracle необязательно. Дополнительная информация указана в главе Установка сервера Ambari 2.6.2 • Hive: – PostgreSQL 8; – PostgreSQL 9.1.13+, 9.3; – MariaDB 5.5; – Oracle 11g R2, 12c. По умолчанию (на RHEL/CentOS) Ambari устанавливает инстанс MySQL на хосте Hive Metastore. Также можно использовать указанный инстанс PostgreSQL, MySQL или Oracle. Дополнительная информация указана в главе Установка сервера Ambari 2.6.2 • Oozie: – PostgreSQL 8; – PostgreSQL 9.1.13+, 9.3; – MariaDB 5.5; – Oracle 11g R2, 12c. По умолчанию Ambari устанавливает инстанс Derby на хосте Oozie Server. При этом использование данного инстанса PostgreSQL, MySQL или Oracle необязательно. Использование Derby в качестве базы метаданных Oozie не рекомендуется для продуктивного использования. Дополнительная информация указана в главе Установка сервера Ambari 2.6.2. Инстанс Derby для рабочей среды не должен использоваться по умолчанию. Если планируется применение Derby для demo-версии, разработки или тестирования окружающей среды, перенос базы данных Oozie из Derby в новую базу данных доступен только в Community Edition • Ranger: – PostgreSQL 9.1.13+, 9.3; – MariaDB 5.5; – Oracle 11g R2, 12c. Для Ranger необходим указанный инстанс PostgreSQL, MySQL или Oracle. Для базы данных Ambari, если используется существующая база данных Oracle, необходимо убедиться, что инстанс Oracle работает на порте, отличном от 8080. Это необходимо, чтобы избежать конфликта с портом Ambari, установленном по умолчанию. Также следует обратиться к справочному руководству Ambari для получения информации о возможных изменениях порта по умолчанию сервера Ambari. Использование параметров базы данных Microsoft SQL Server или SQL Anywhere не поддерживается. Если используется существующая база данных, развернутая на том же сервере, что и Ambari Server, необходимо, чтобы эта база работала на порте, отличном от 8080

3.1.1 Требования к памяти

Хост **Ambari** должен иметь как минимум *1 ГБ* оперативной памяти и *500 МБ* свободной. На любом узле можно проверить доступную память по команде:

```
free -m
```

В общем случае хосту, на котором планируется запуск **Ambari Metrics Collector**, необходимо иметь доступную память и дисковое пространство в зависимости от размера кластера.

Таблица3.2.: Требования к объему памяти в зависимости от размера кластера

Количество хостов	Доступная память	Дисковое пространство
1	1024 MB	10 GB
10	1024 MB	20 GB
50	2048 MB	50 GB
100	4096 MB	100 GB
300	4096 MB	100 GB
500	8096 MB	200 GB
1000	12288 MB	200 GB
2000	16384 MB	500 GB

Сведения в таблице предлагаются в качестве рекомендаций. Необходимо проверять каждый частный случай.

3.1.2 Требования к объему пакета и количеству индексных дескрипторов

Таблица3.3.: Приблизительные значения размера пакета и количества индексных дескрипторов

	Размер пакета	Количество индексных дескрипторов
Ambari Server	100MB	5.000
Ambari Agent	8MB	1.000
Ambari Metrics Collector	225MB	4.000
Ambari Metrics Monitor	1MB	100
Ambari Metrics Hadoop Sink	8MB	100
After Ambari Server Setup	Данные отсутствуют	4.000
After Ambari Server Start	Данные отсутствуют	500
After Ambari Agent Start	Данные отсутствуют	200

3.1.3 Проверка максимального количества открытых файловых дескрипторов

Рекомендуемое максимальное количество открытых файловых дескрипторов – более *10000*. Для проверки текущего значения необходимо выполнить следующие команды на каждом хосте:

```
ulimit -Sn
ulimit -Hn
```

Если значение максимального количества открытых файловых дескрипторов не превышает *10000*, то следует установить подходящее значение, выполнив команду:

```
ulimit -n 10000
```

3.2 Сбор информации

Перед развертыванием кластера **ADH** необходимо:

- Проверить полное доменное имя (FQDN) каждого хоста в кластере. Мастер установки Ambari поддерживает использование IP-адресов. 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 Oozie;
  - 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

3.3 Подготовка к установке Hadoop

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

- *Настройка беспарольного SSH доступа к хостам*
- *Настройка учетных записей пользователей*
- *Синхронизация часов на всех узлах кластера*
- *Настройка DNS и NSCD*
- *Настройка IPTables*
- *Настройка SELinux, PackageKit и Umask*

3.3.1 Настройка беспарольного SSH доступа к хостам

Для **Ambari Server** необходимо, чтобы **Ambari Agents** были установлены на всех узлах кластера. Сервер **Ambari** связывается с агентами для выполнения установки и управления задачами кластера **Arenadata**.

Для автоматической установки Ambari-агентов на всех узлах кластера сервером **Ambari** необходимо для учетной записи, под которой производится установка (по умолчанию от *root*), настроить беспарольное SSH-соединение между хостом **Ambari Server** и всеми другими хостами в кластере.

Для настройки беспарольного **SSH** необходимо под выбранной учетной записью выполнить следующие инструкции (на примере *root*):

- Создать публичный и приватный ключи SSH на хосте сервера Ambari:

  ```
  ssh-keygen
  ```

- При запросе пароля для приватного ключа необходимо не задавая значения (поле для ввода пароля оставить пустым) нажать клавишу *Enter*;

- Скопировать публичный ключ SSH (*id_rsa.pub*) в учетную запись *root* на все узлы кластера:

  ```
  ssh-copy-id root@<remote.target.host>
  ```

  где <*remote.target.host*> – значение имени каждого хоста в кластере;

  В случае если во время первого подключения отображается предупреждение, необходимо ответить *yes*:

  ```
  Are you sure you want to continue connecting (yes/no)?
  ```

- Убедиться, что с сервера Ambari выполняется подключение к каждому хосту в кластере с помощью SSH без пароля:

  ```
  ssh root@<remote.target.host>
  ```

- (Опционально) Сохранить копию приватного ключа SSH на компьютере, с которого планируется запуск веб-мастера установки Ambari Install Wizard.

Хост сервера **Ambari** использует аутентификацию публичного ключа **SSH** для удаленного доступа и установки Ambari-агента.

Так же Ambari-агенты возможно установить вручную на каждом узле кластера. В этом случае не будет необходимости создавать и распространять ключи **SSH**.

3.3.2 Настройка учетных записей пользователей

Для каждого сервиса **ADH** необходима учетная запись пользователя ОС. Мастер установки **Ambari** создает новые и сохраняет все существующие учетные записи пользователей и использует их при настройке сервисов **Hadoop**.

3.3.3 Синхронизация часов на всех узлах кластера

Часам на всех узлах кластера и на машине, запускающей браузер для доступа к веб-интерфейсу **Ambari**, необходимо иметь возможность синхронизации друг с другом. Для этого следует включить службу **NTP** и убедиться, что синхронизация происходит автоматически:

- RHEL/CentOS 7 & SUSE/SLES 12:

  ```
  systemctl is-enabled ntpd
  ```

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

- RHEL/CentOS 7 & SUSE/SLES 12:

  ```
  systemctl enable ntpd
  ```

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

- RHEL/CentOS 7 & SUSE/SLES 12:

  ```
  systemctl start ntpd
  ```

3.3.4 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>
```

3.3.5 Настройка IPTables

Для взаимодействия во время установки **Ambari** 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
```

- SUSE/SLES 12:

```
rcSuSEfirewall2 stop
chkconfig SuSEfirewall2 off
```

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.

В процессе установки Ambari-сервера **Ambari** проверяет, работают ли **IPTables**. В случае если **IPTables** запущены, выдается предупреждение, напоминающее о необходимости проверить, что требуемые порты открыты и доступны.

В мастере установки на шаге подтверждения узла также выдается предупреждение для каждого хоста с запущенными **IPTables**.

3.3.6 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.

Ambari поддерживает 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 4

Установка сервера Ambari 2.6.2

Установка сервера **Ambari 2.6.2** проходит в несколько этапов:

- *Настройка репозитория*
- *Установка сервера Ambari*
- *Настройка сервера Ambari*
- *Запуск сервера Ambari*
- *Подготовка к установке основных компонентов ADH на кластер*

4.1 Настройка репозитория

Установка сервера **Ambari** выполняется с помощью пакетного менеджера из репозитория, содержащего соответствующий пакет. При этом допускается использование как удаленного репозитория, доступного через сеть Интернет, так и размещенного в локальной сети кластера (например, если из соображений безопасности доступ к сети Интернет ограничен).

Important: Репозиторий должен находиться на доступном со всех узлов кластера хосте

4.1.1 Настройка удаленного репозитория

Настройка удаленного репозитория не отличается от настройки любого дополнительного репозитория.

Для добавления репозитория необходимо выполнить от имени *root* команду:

- RHEL/CentOS 7:

```
yum-config-manager --add-repo <URL репозитория Ambari>/ambari.repo
```

- SUSE/SLES 12:

```
zypper addrepo <URL репозитория Ambari>/ambari.repo
```

4.1.2 Настройка локального репозитория

Для настройки нового сервера репозитория необходим веб-сервер *httpd*. Следует убедиться, что сервер *httpd* запускается на хосте, который служит в качестве репозитория:

- RHEL/CentOS 7:

```
systemctl status httpd
```

- SUSE/SLES 12:

```
systemctl status apache2.service
```

В случае если сервер не запущен, необходимо его установить и запустить:

- RHEL/CentOS 7:

```
systemctl start httpd
```

- SUSE/SLES 12:

```
systemctl start apache2.service
```

Создание промежуточного каталога

Для извлечения архивов для стеков **Ambari** и **ADH** рекомендуется использовать промежуточный каталог.

Каждый архив представляет собой архивный репозиторий и имеет скрипт *setup_repo.sh*, создающий ссылку из корня документа *httpd server/var/www/html* в каталог, из которого извлекается архив. Необходимо, чтобы промежуточный каталог и все верхнеуровневые каталоги были читаемыми и доступными пользователю, выполняющему процесс *httpd* (**apache**), а лучше сделать их доступными для всех пользователей кластера:

```
mkdir /staging
chmod a+rx /staging
```

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

Загрузка и распаковка архива Ambari

Arenadata Ambari поставляется как архив репозитория, который необходимо извлечь на сервер репозитория.

На узел, который используется в качестве репозитория, необходимо загрузить архив **Ambari 2.6.2** в ранее созданный промежуточный каталог или в каталог:

- RHEL/CentOS 7:

```
wget https://storage.googleapis.com/arenadata-repo/ambari/2.6.2/repos/ambari.repo -O /etc/yum.repos.d/
```

- SUSE/SLES 12:

```
wget https://storage.googleapis.com/arenadata-repo/ambari/2.6.2/repos/ambari-sles.repo - O /etc/zypp/
↪repos.d/
```

- PPC64le RHEL7:

```
wget https://storage.googleapis.com/arenadata-repo/ambari/2.6.2/repos/ambari-ppc64le.repo - O /etc/
↪zypp/repos.d/
```

Необходимо убедиться, что все родительские каталоги до промежуточного имеют доступ *"r + x"* для всех пользователей, поскольку данный каталог будет использоваться для создания локального репозитория.

После загрузки **Ambari 2.6.2** необходимо извлечь архив в промежуточный каталог. Например:

```
tar -xvf /staging/AMBARI-2.6.2.tar -C /staging/
```

Настройка локального репозитория

Для настройки локального репозитория необходимо на хосте, используемом в качестве репозитория, выполнить скрипт *setup_ repo.sh*, входящий в состав архива **Ambari**:

```
/staging/AMBARI-2.6.2/setup_repo.sh
```

В скрипте предполагается, что в корневом каталоге репозитория веб-сервер устанавливает */var/www/html* и создает ссылку *ambari-<версия>*, указывающую на извлеченный архив.

Необходимо убедиться, что репозиторий доступен на веб-сервере:

```
curl http://localhost/AMBARI-2.6.2/repodata/repomd.xml
```

Скрипт также создает определенный репозиторий **Ambari** и помещает его в файл:

- RHEL/CentOS 7:

```
/etc/yum.repos.d/ambari.repo
```

- SUSE/SLES 12:

```
/etc/zypp/repos.d/ambari.repo
```

Данный файл должен быть доступен на хосте администратора, где будет установлен сервер **Ambari**.

Important: Репозиторий Ambari должен быть доступен для всех узлов кластера

Необходимо проверить наличие доступа к следующему URL-адресу с хоста администратора и с узлов кластера:

```
http://<yum.repo.host.fqdn>/AMBARI-2.6.2
```

4.2 Установка сервера Ambari

Сервер **Ambari** устанавливается из RPM-пакета по команде:

- RHEL/CentOS 7:

```
yum install ambari-server
```

- SUSE/SLES 12:

```
zypper install ambari-server
```

Данная команда устанавливает сервер **Ambari**, являющийся сервером веб-приложений, на порт *8080*. Также устанавливает инстанс сервера **PostgreSQL** на порт *5432*.

4.3 Настройка сервера Ambari

Сервер **Ambari** необходимо настроить для корректной работы.

В случае если инстанс **PostgreSQL** настроен на порт по умолчанию, следует выполнить следующую команду:

```
ambari-server setup
```

В процессе настройки необходимо указать или принять по умолчанию параметры:

- *Учетная запись пользователя* – для запуска Ambari-сервера можно выбрать любую учетную запись (необязательно выполнять вход от *root*). В случае если пользователя не существует, он создается автоматически;
- *Java JDK* – для загрузки Oracle JDK 1.8 необходимо ввести значение *1* и принять лицензию Oracle JDK для загрузки файлов из Oracle. При этом установка JDK выполняется автоматически;
- *База данных* – выбор базы данных:

  ```
  Enter advanced database configuration
  ```

  В командной строке необходимо ответить *n* или *y*:

  – *n* – для использования с Ambari стандартной встроенной базы данных PostgreSQL. По умолчанию для базы данных PostgreSQL устанавливается имя “ambari” и логин / пароль принимают значения *ambari / bigdata*.

  – *y* – при необходимости использования с Ambari уже существующей базы данных PostgreSQL, MySQL или Oracle вместо предлагаемой по умолчанию. Далее для выбранной базы данных необходимо указать параметры подключения (см. Приложение 1).

4.4 Запуск сервера Ambari

После установки сервера **Ambari** запуск его осуществляется по команде:

```
ambari-server start
```

Для проверки статуса сервера необходимо использовать команду:

```
ambari-server status
```

Для остановки сервера необходимо использовать команду:

```
ambari-server stop
```

Сервер **Ambari** доступен на порту *8080*. По умолчанию для него установлена следующая учетная запись:

- User: *admin*
- Password: *admin*

Important: Рекомендуется сменить пароль после первого входа в систему

Для входа в веб-интерфейс **Ambari** необходимо в адресной строке браузера указать адрес сервера:

```
http://<адрес сервера>:8080
```

При этом запрашивается логин и пароль. После авторизации открывается веб-интерфейс **Ambari** (Рис.4.1.).

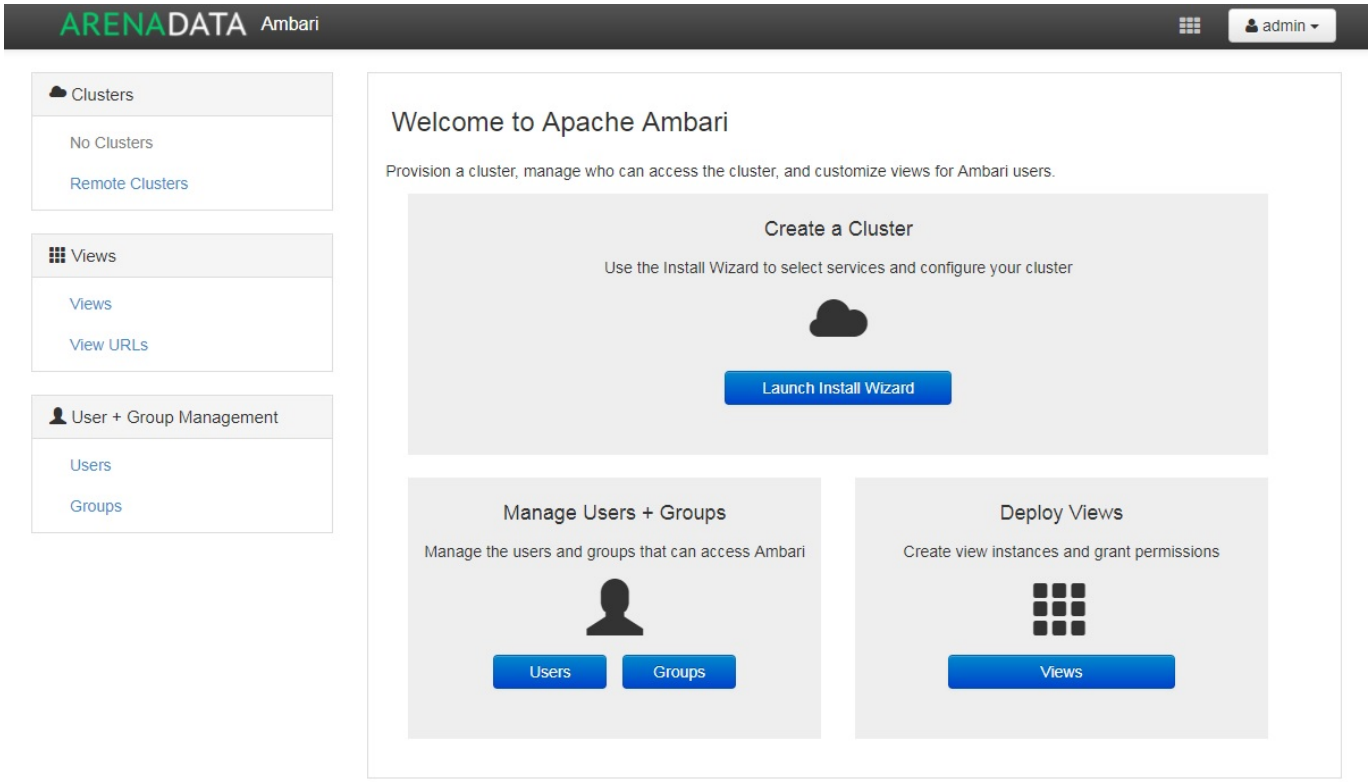


Рис.4.1.: Веб-интерфейс Ambari до настройки кластера

4.5 Подготовка к установке основных компонентов ADH на кластер

Основные компоненты **ADH** устанавливаются из репозиториев, которые определяются при первичной настройке кластера. Как и в случае репозитория **Ambari**, допускается использование удаленных и локальных репозиториев.

Удаленные репозитории уже заданы в **Ambari** как предлагаемые по умолчанию, для их настройки и использования не требуется дополнительных действий.

Для настройки локальных репозиториев необходимо выполнить действия, аналогичные настройке локального репозитория **Ambari**:

- Загрузить и извлечь архивы стека ADH;
- Настроить локальные репозитории.

4.5.1 Загрузка и извлечение архивов стека ADH

Архивы стека **ADH** необходимо установить на машине, где размещен репозиторий. В случае если для сервера репозитория используется выделенная машина, то архивы стека **ADH** следует установить на хосте администратора, использованном для установки сервера **Ambari**.

Необходимо загрузить и распаковать следующие архивы в выделенном для них месте (при этом следует избегать использования каталога */tmp*):

- *ADH-1.6.1* – RPM-пакеты для сервисов Hadoop, таких как HDFS, YARN, Hbase, Hive, Zookeeper;
- *ADH-UTILS-1.6.1* – дополнительные сервисы и библиотеки, используемые для мониторинга и оповещения серверов кластера.

В случае если архивы загружены в каталог */tmp*, то для их распаковки в каталоге, например, */staging* необходимо выполнить следующую команду:

```
tar -xvf /tmp/{stack}.tar -C /staging/
```

Для использования локальных репозиториев **ADH** и **ADH UTILS** необходимо выполнить настройки, описанные в пункте *Настройка локальных репозиториев*.

4.5.2 Настройка локальных репозиториев

Стек **ADH** поставляется в виде архива репозитория, который необходимо развернуть на сервере репозитория так, чтобы при этом он был доступен серверу **Ambari** и всем узлам кластера.

Каждый репозиторий стека содержит скрипт *setup_repo.sh*, для которого необходимо выполнение следующих требований:

- Сервер репозитория доступен всем узлам кластера;
- Корень сервера репозитория находится в */var/www/html/*.

Скрипт каждого стека создает символическую ссылку в документе сервера репозитория, указывающую на местоположение извлеченного архива стека, и генерирует файл с местоположением репозитория в каталоге:

- RHEL/CentOS 7:

```
/etc/yum.repos.d/
```

- SUSE/SLES 12:

```
/etc/zypp/repos.d/
```

Для каждого стека необходимо запустить скрипт установки локального репозитория:

```
/staging/{stack}/setup_repo.sh
```

По завершению установки скрипт выводит URL-адрес репозитория. Данный URL потребуется при установке кластера **ADH** с использованием сервера **Ambari**.

В случае если сервер репозитория установлен не на хосте администратора (где установлен сервер **Ambari**), необходимо скопировать созданные файлы определения местоположения репозитория из папки репозитрия на хост администратора, где установлен сервер **Ambari**.

Затем необходимо проверить корректность настройки репозитория, выполнив две команды от узла администратора:

- RHEL/CentOS 7:

```
yum clean all
yum repolist
```

- SUSE/SLES 12:

```
zypper clean -a
zypper repos
```

При корректной настройке выдается список репозиториев стека.

Глава 5

Запуск мастера установки

Для создания кластера необходимо после входа в **Ambari** запустить мастер установки, нажав в главной экранной форме кнопку *Launch Install Wizard*. Мастер установки проводит по шагам, необходимым для создания нового кластера **ADH**. Некоторые этапы установки требуют особого внимания:

- Изменение URL-адресов репозиториев;
- Ввод имен узлов и SSH-ключа или ручная установка Ambari-агентов;
- Выбор компонентов;
- Назначение мастер-узлов для компонентов;
- Назначение Slave и Client узлов;
- Дополнительные настройки компонентов.

5.1 Изменение URL-адресов репозиториев

Для того чтобы открыть список репозиториев, необходимо в блоке "Stacks" установить флаг в поле *ADH 1.5* и раскрыть блок "Advanced Repository Options", при этом **Ambari** предлагает указать URL-адреса репозиториев (Рис.5.1.).

В полях "Base URL" необходимо указать URL-адреса репозиториев, которые были получены при запуске скрипта *setup_repo.sh*. Данные URL-адреса всегда можно уточнить в файлах *<имя репозитория>.repo*.

Для установки из публичного репозитория **Arenadata** необходимо выбрать соответствующий пункт меню "Use Public Repository" (Рис.5.2.).

Репозитории можно обновить после развертывания кластера через **Ambari UI** (*"Admin → Repositories"*).

5.2 Ввод имен узлов и SSH-ключа

В разделе "Install Options" следует указать **FQDN** для узлов, которые будут содержать кластер. Можно задавать диапазоны имен с помощью квадратных скобок, например, *host [01-10].domain* описывает *10* хостов. В случае если применяется **EC2**, необходимо использовать имена внутренних частных DNS-узлов.

Для автоматической регистрации Ambari-агентов на узлах кластера необходимо ввести закрытый ключ, который использовался для настройки беспарольного **SSH** для кластера. Можно передать сам файл *id_rsa* или скопировать и вставить его содержимое в экранную форму.

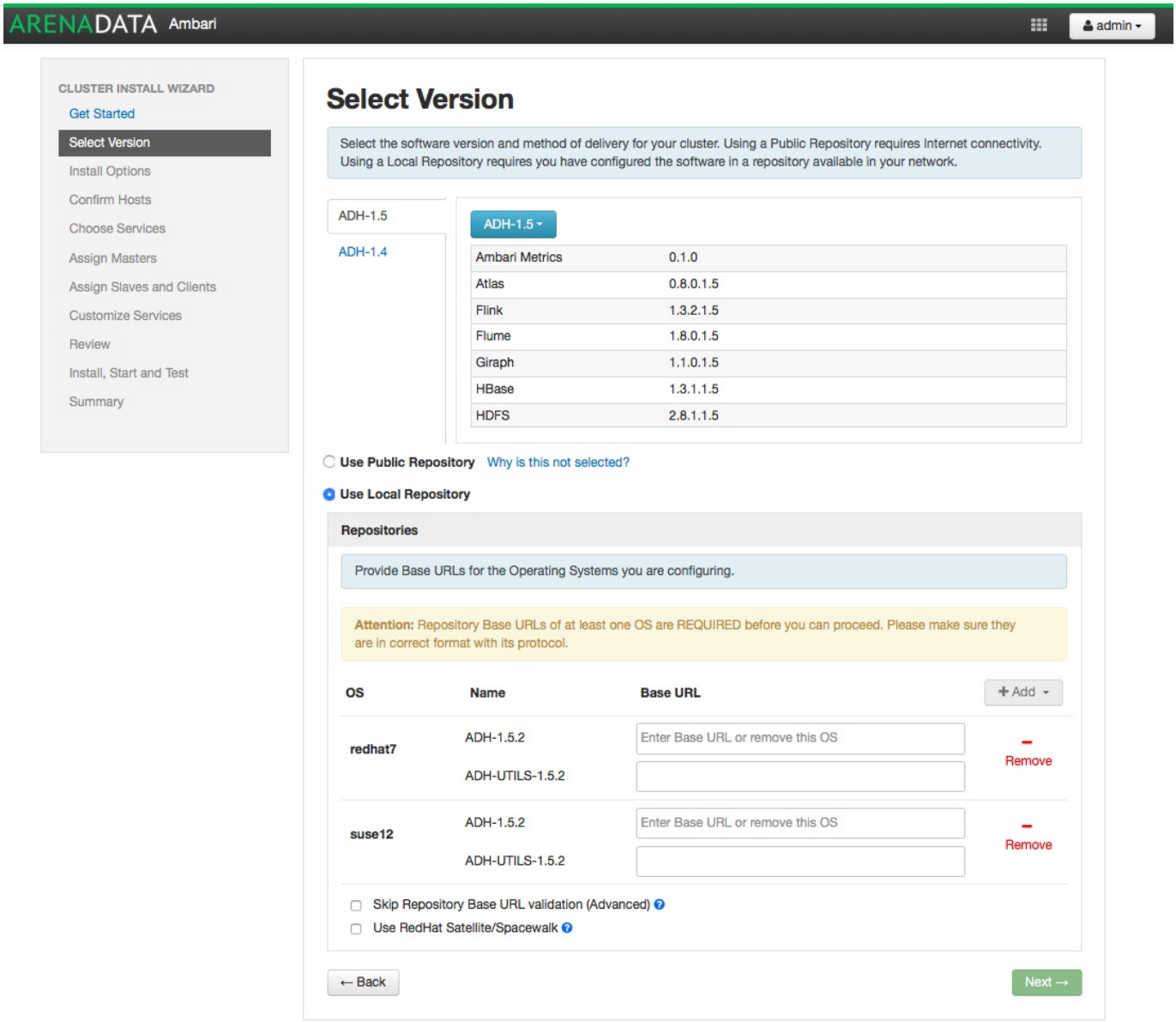


Рис.5.1.: Выбор стека

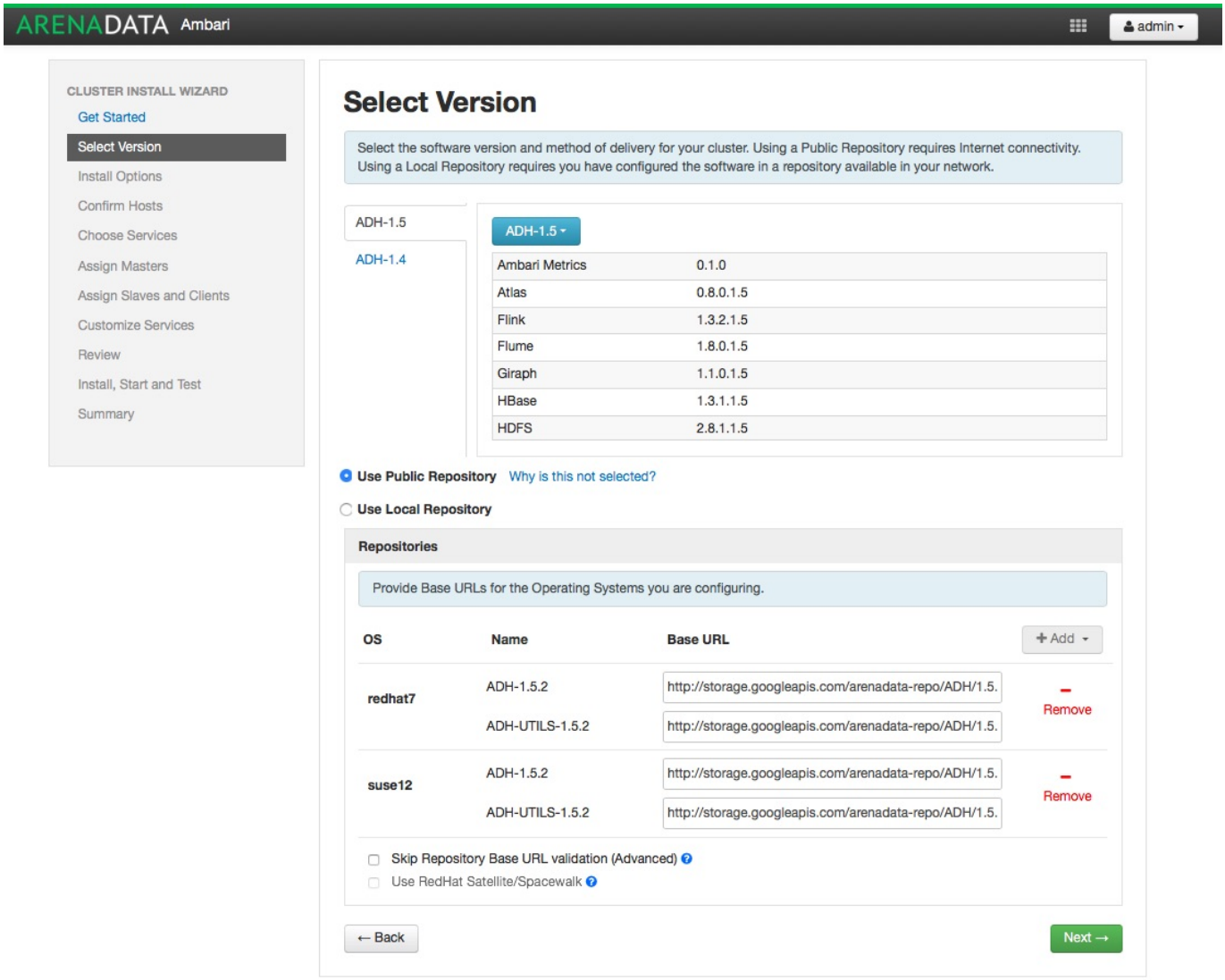


Рис.5.2.: Установка из публичного репозитория

5.2.1 Ручная установка Ambari-агентов

В случае если нет возможности предоставить закрытый ключ беспарольного **SSH**, следует произвести установку Ambari-агентов вручную. Для этого на каждом узле кластера необходимо выполнить следующие шаги:

- Установить репозиторий Ambari, скопировав файл *ambari.repo* с сервера репозитория;
- Установить Ambari-агент, выполнив команду:
  - RHEL/CentOS 7:

```
yum install ambari-agent
```

  - SUSE/SLES 12:

```
zypper install ambari-agent
```

- Изменить конфигурацию Ambari-агента */etc/ambari-agent/conf/ambari-agent.ini* для определения его на сервере Ambari:

```
[server]
hostname={ambari.server.hostname}
url_port=8440
secured_url_port=8441
```

- Запустить Ambari-агент, выполнив команду:

```
ambari-agent start
```

  Ambari-агент зарегистрируется на сервере при его запуске.

5.3 Выбор компонентов

На начальном этапе установки необходимо выбрать компоненты **ADH**, которые следует инсталлировать. При этом **HDFS** и **Zookeeper** всегда обязательны, остальные компоненты возможно установить позднее (Рис.5.3.).

В случае если выбирается компонент **Ambari Metrics**, то для контроля кластера можно использовать **Ambari**. Если данный компонент не выбирается, выдается предупреждение, которое можно игнорировать в случае, если кластер планируется контролировать с помощью других инструментов. При этом **Ambari Metrics** можно будет добавить в кластер позднее.

5.4 Назначение мастер-узлов

Необходимо назначить мастер-узлы компонентов кластера (Рис.5.4.).

Important: Если Hive Metastore использует новую базу данных *PostgreSQL*, компонент HIVE METASTORE не должен находиться на хосте AMBARI

Данное ограничение объясняется тем, что оба компонента будут пытаться использовать порт *5432*. В случае абсолютной необходимости совместного размещения указанных компонентов на одном и том же хосте, предварительно следует переконфигурировать базу данных **PostgreSQL** на порт, отличный от *5432*, и выбрать опцию “Existing PostgreSQL Database” для конфигурации **Hive Metastore**.

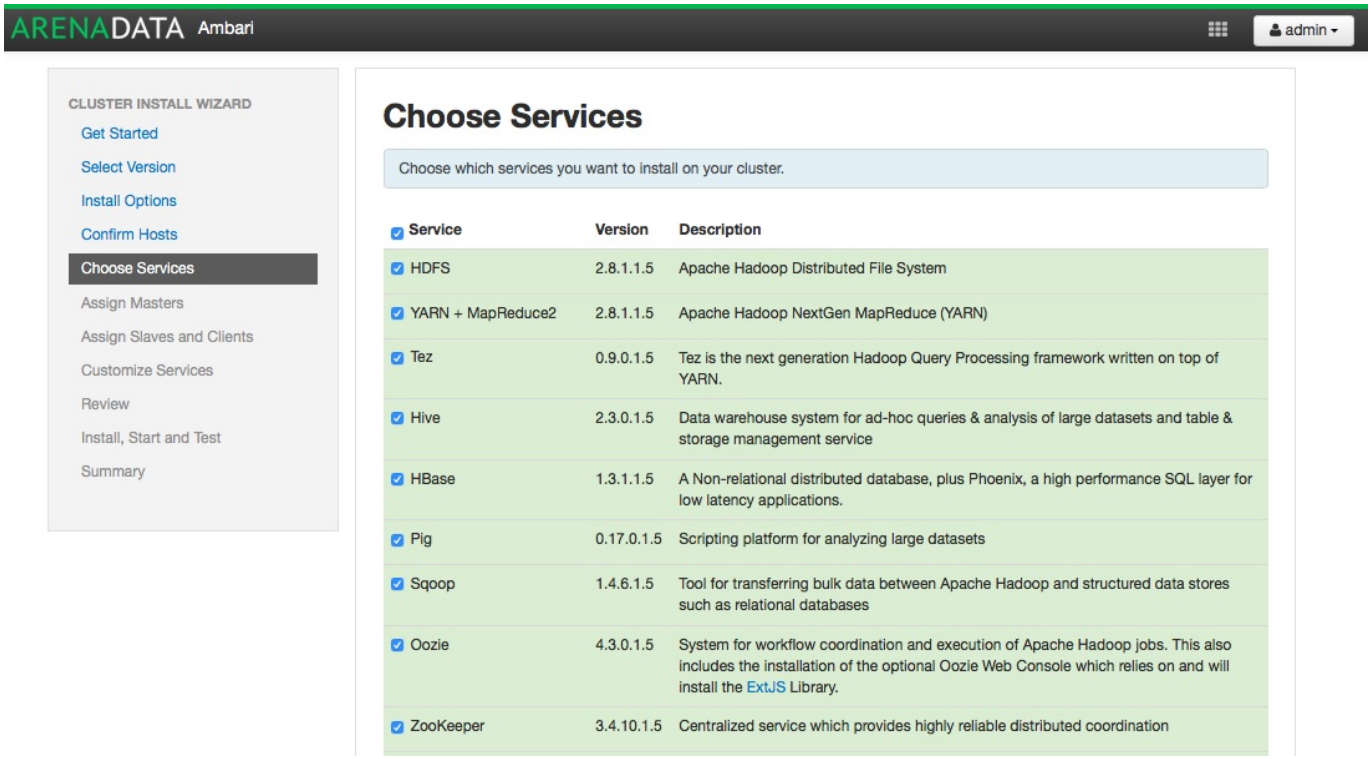


Рис.5.3.: Выбор компонентов

5.5 Назначение Slave и Client узлов

Необходимо назначить **Slave** и **Client** узлы, на которых будут разворачиваться соответствующие компоненты кластера (Рис.5.5.).

5.6 Дополнительные настройки компонентов

На экранной форме отображаются параметры конфигурации отдельных компонентов, автоматически сгенерированных установщиком **Ambari** на основе параметров кластера. Параметры каждого компонента можно менять по своему усмотрению в зависимости от планируемого использования того или иного компонента кластера.

В случае если для какого-либо обязательного параметра установщик не может предложить значение по умолчанию, перед продолжением установки данные параметры необходимо указать вручную (на Рис.5.6. приведен пример, когда для компонентов *Hive*, *Oozie*, *Ambari Metrics*, *Knox* необходимо указать пароли для внутренних баз данных).

Important: Каталоги для размещения данных HDFS (параметр “DataNode Directories” сервиса HDFS) не должны содержать никаких других данных, в том числе данных других компонентов. Это связано с тем, что при старте DataNode указанные каталоги очищаются, и может произойти потеря данных

Important: В случае если компонент DataNode сервиса HDFS устанавливается менее, чем на трех узлах кластера, необходимо задать соответствующее значение параметра *DFS Replication Factor* указанного

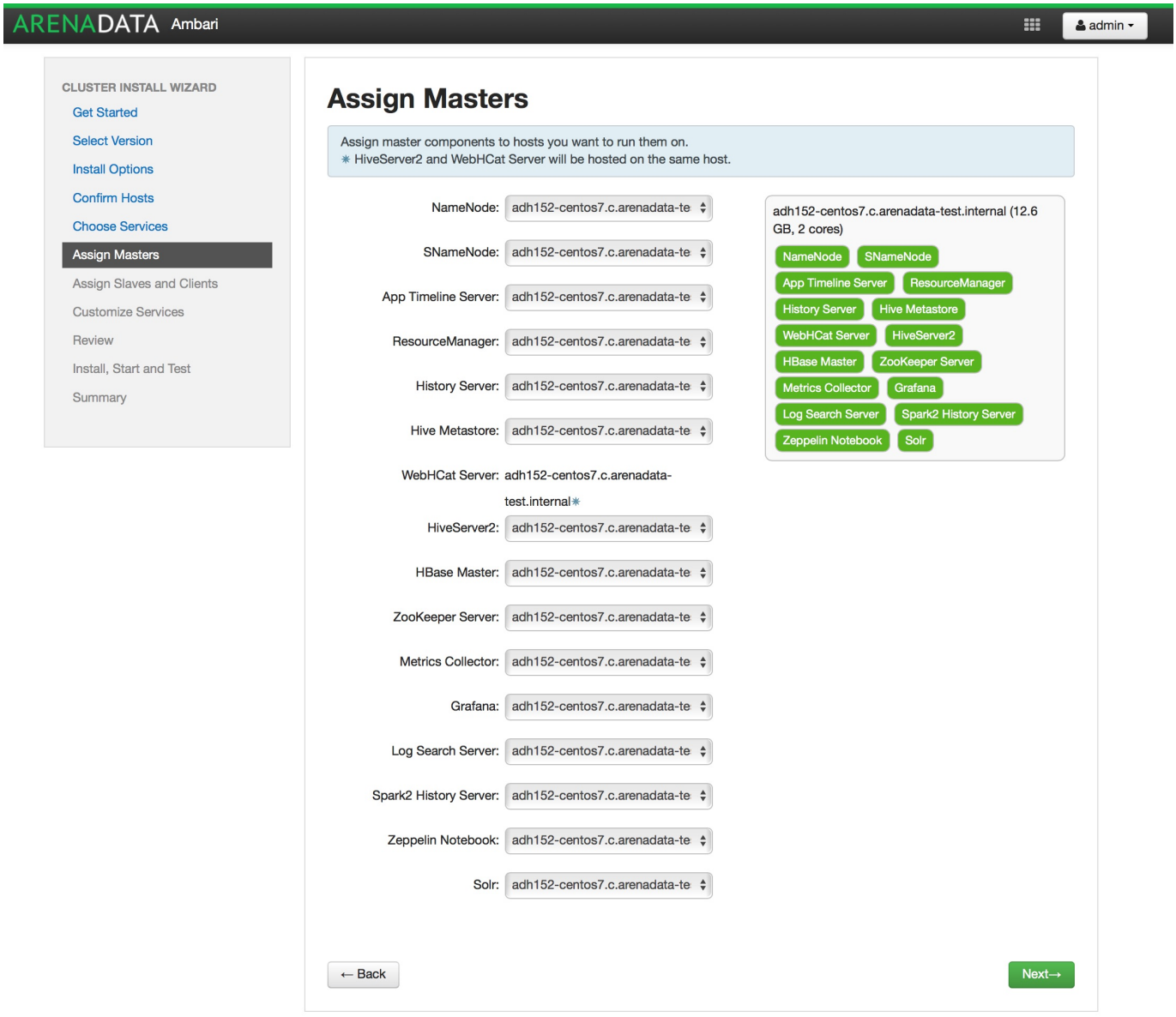


Рис.5.4.: Назначение мастер-узлов

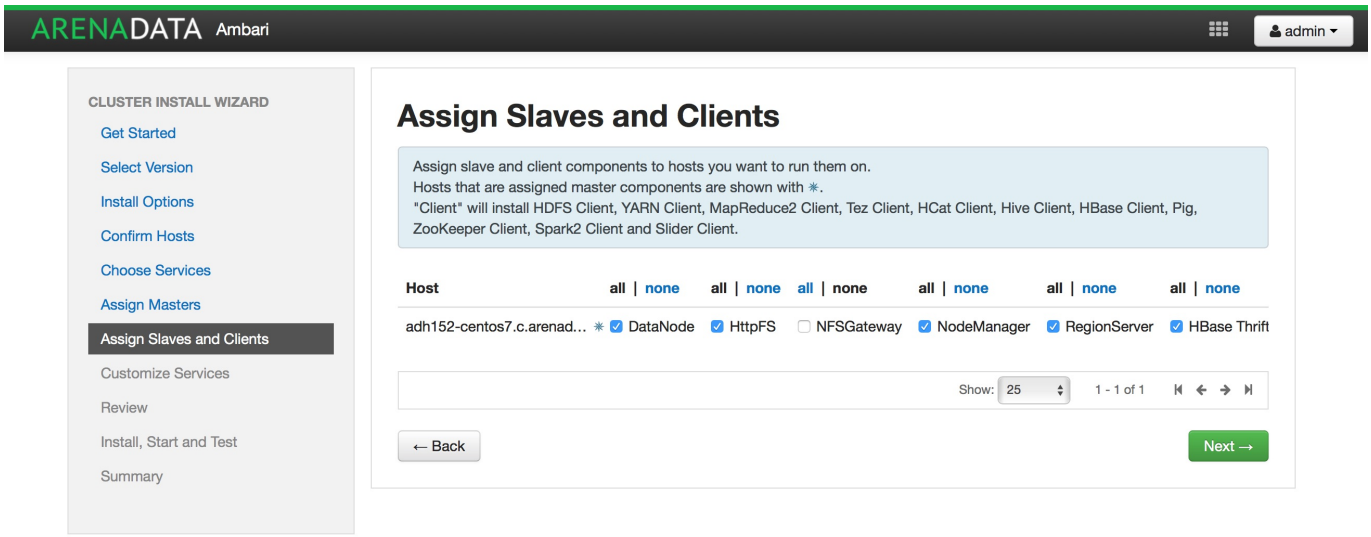


Рис.5.5.: Назначение Slave и Client узлов

компонента

5.7 Установка, запуск и тестирование

На экранной форме отображается ход развертывания кластера на каждом узле (Рис.5.7.).

Каждый компонент, который разворачивается вместе с хостом, устанавливается, запускается и проходит простой тест для проверки работоспособности.

При этом есть возможность просмотра подробной информации о завершенных и ожидающих задачах для каждого хоста (Рис.5.8.). Для этого необходимо нажать ссылку в столбце “Message” (см. Рис.5.7.).

По завершении установки компонентов появляется сообщение *Successfully installed and started the services*, в котором необходимо нажать кнопку *Next*.

Для окончания установки необходимо на странице “Summary” проверить список завершенных задач и нажать кнопку *Complete*. При этом открывается панель инструментов кластера.

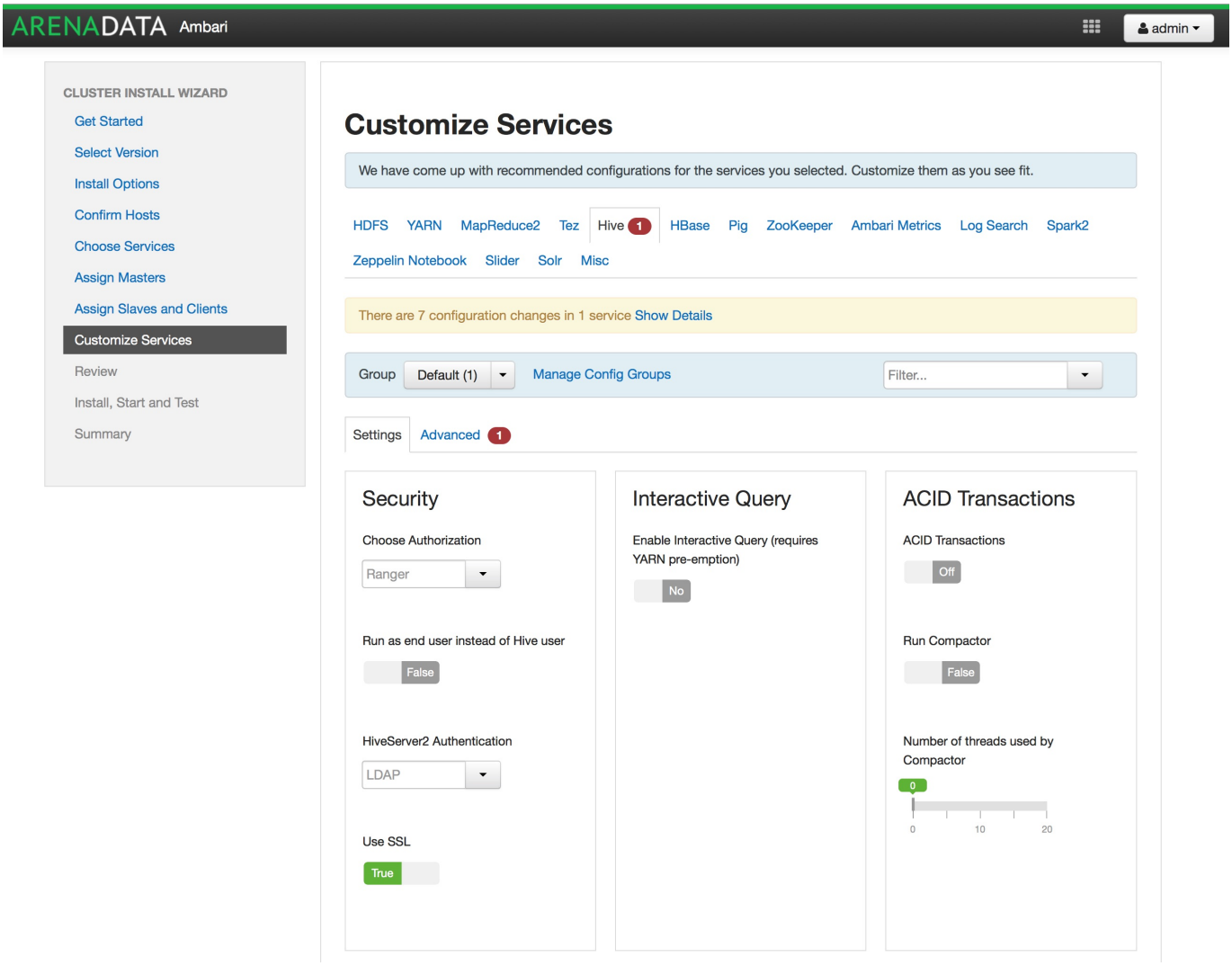


Рис.5.6.: Дополнительные настройки компонентов

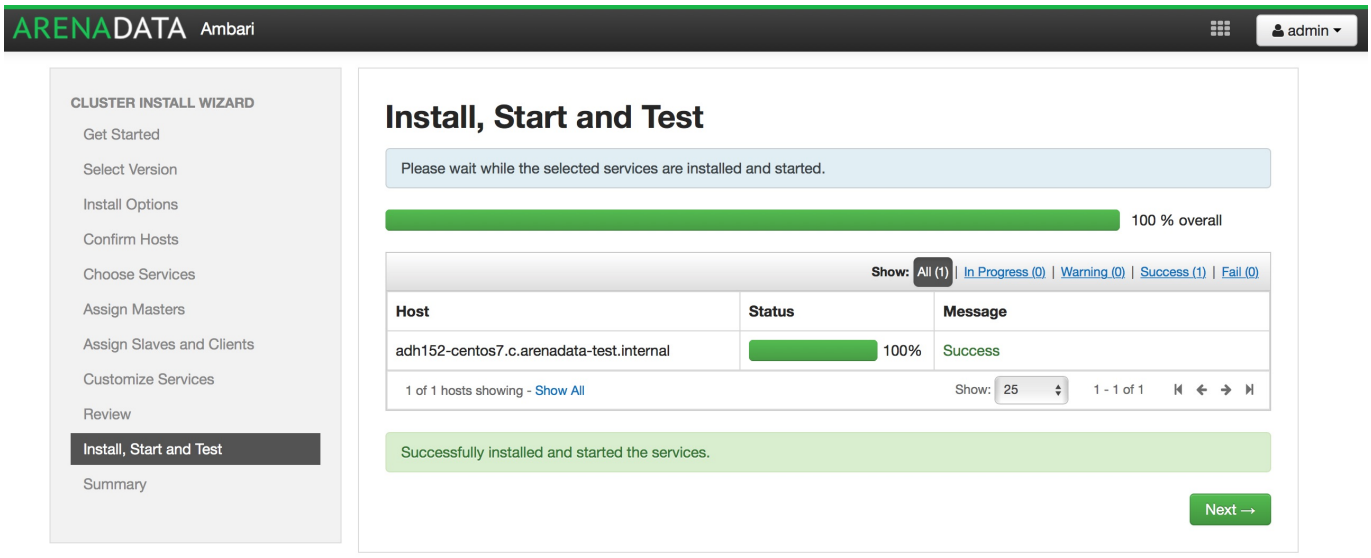


Рис.5.7.: Ход развертывания кластера

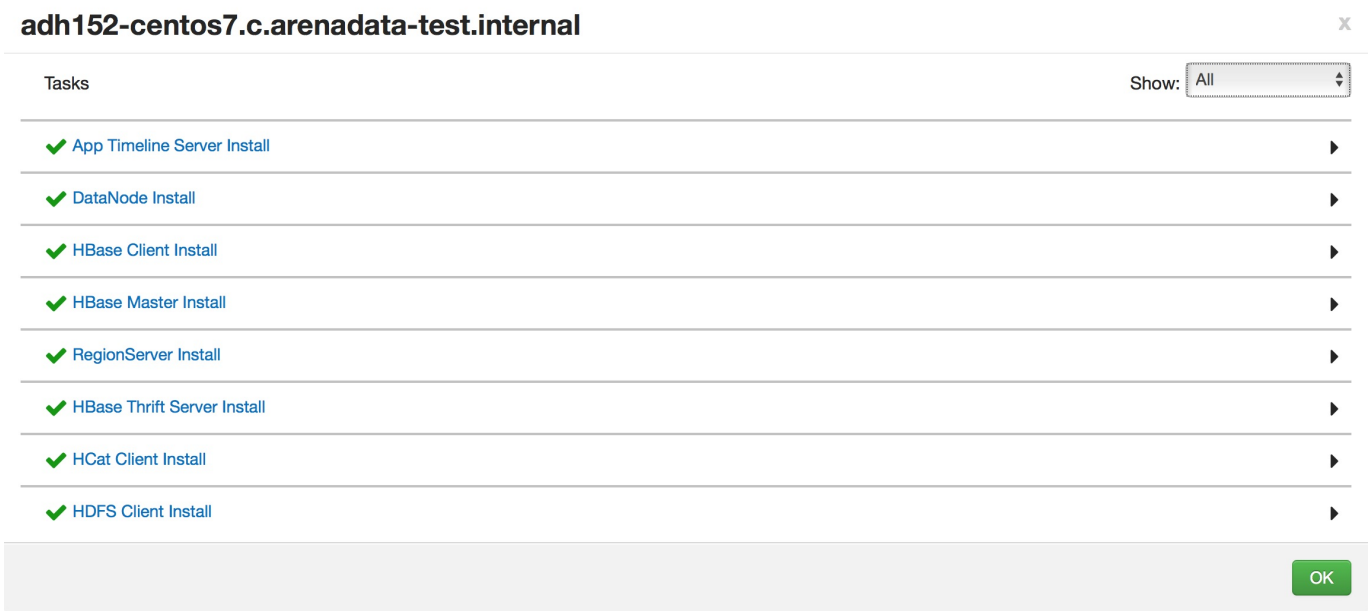


Рис.5.8.: Информация о задачах хоста

Глава 6

Панель кластера

Панель инструментов кластера является центральным местом и отображает развернутые сервисы и их статус. Здесь можно добавлять новые сервисы или хосты, останавливать и запускать сервисы и компоненты, анализировать показатели мониторинга и выполнять конкретные действия сервиса (Рис.6.1.).

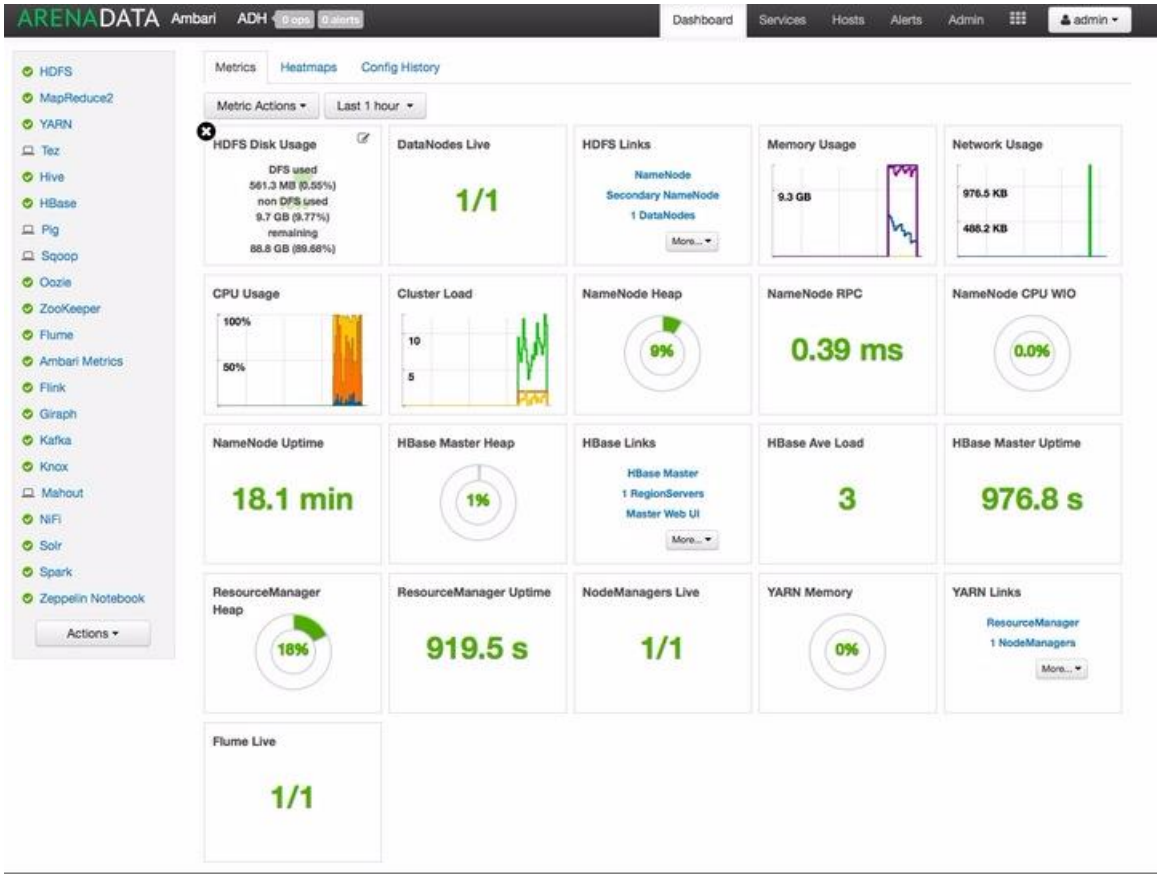


Рис.6.1.: Панель кластера

Глава 7

Ручная настройка подключения к базе данных

Если в процессе настройки сервера **Ambari** необходимо отличное от используемого по умолчанию подключение к базе данных следует в командной строке нажать клавишу *y*:

```
Enter advanced database configuration
```

Если инстанс **PostgreSQL** настроен на порт, отличный от предлагаемого по умолчанию, для настройки **Ambari** необходимо выполнить следующие шаги:

- Открыть в текстовом редакторе конфигурационный файл PostgreSQL */var/lib/pgsql/data/pg_hba.conf*. Чтобы позволить пользователю *ambari* подключиться к базе данных, необходимо в конце файла добавить следующие строки:

```
local all ambari md5
host all ambari 0.0.0.0/0 md5
host all ambari ::/0 md5
```

- Чтобы подключить порт, выбранный не по умолчанию, следует открыть файл */etc/sysconfig/pgsql/postgresql* и добавить в него строку с номером необходимого порта. Например, чтобы подключить порт *10432* следует указать:

```
PGPORT=10432
```

- Перезапустить базу данных PostgreSQL:

```
service postgresql restart
```

- Подключиться к базе данных под *postgres* (супер-пользователь) и выполнить следующие настройки:

```
psql -U postgres -p 10432;
postgres=# CREATE DATABASE ambari;
postgres=# CREATE USER ambari WITH ENCRYPTED PASSWORD 'bigdata';
postgres=# \c ambari;
ambari=# CREATE SCHEMA ambari AUTHORIZATION ambari;
ambari=# ALTER SCHEMA ambari OWNER TO ambari;
ambari=# ALTER ROLE ambari SET search_path to 'ambari','public';
ambari=# \q
```

- Выполнить команду установки Ambari:

```
ambari-server setup --database=postgres --databasehost=localhost--databaseport=10432 --
↪databasename=ambari --databaseusername=ambari--databasepassword=bigdata
```

- Чтобы убедиться, что *postgres* подключен к хосту *databasehost*, необходимо использовать следующую команду:

```
netstat -anp | egrep <port>
```

- Выполнить файл *Ambari-DDL-Postgres-CREATE.sql* в PostgreSQL для завершения настройки:

```
psql -f /var/lib/ambari-server/resources/Ambari-DDL-Postgres-CREATE.sql -U ambari -p 10432 -d ambari
```

- При запросе пароля необходимо ввести значение *bigdata*.