Arenadata™ Hadoop

Версия - v2.0.0

Настройка безопасности для Ambari

2019 Arenadata™

Оглавление

1 Настройка Ambari и Hadoop для Kerberos **3**
- 1.1 Обзор Kerberos . . . 4
- 1.2 Принципалы Hadoop и Kerberos . . . 5
- 1.3 Установка и настройка KDC . . . 6
- 1.4 Подключение системы безопасности Kerberos в Ambari . . . 9
- 1.5 Клиентские пакеты Kerberos . . . 12
- 1.6 Отключение системы безопасности Kerberos в Ambari . . . 13
- 1.7 Настройка шаблона атрибута . . . 13
- 1.8 Перечень компонентов поддерживающих работу в Kerberos окружении . . . 14

2 Расширенные параметры безопасности для Ambari **15**
- 2.1 Настройка Ambari для аутентификации LDAP или Active Directory . . . 15
- 2.2 Настройка Ambari для Non-Root . . . 19
- 2.3 Шифрование базы данных и паролей LDAP (опционально) . . . 22
- 2.4 Настройка SSL для Ambari (опционально) . . . 24
- 2.5 Настройка Kerberos для сервера Ambari (опционально) . . . 24
- 2.6 Настройка Truststore для сервера Ambari . . . 25
- 2.7 Настройка двустороннего SSL между Ambari Server и Ambari Agents (опционально) . . . 26
- 2.8 Настройка шифров и протоколов для сервера Ambari (опционально) . . . 26

3 Настройка двухсторонней SSL аутентификации для Ambari **27**
- 3.1 Создание сертификатов . . . 27
- 3.2 Включение SSL на стороне ambari-server . . . 28
- 3.3 Включение двустороннего SSL шифрования . . . 28
- 3.4 Добавление новых нод . . . 29

4 Аутентификация SPNEGO для Hadoop **30**
- 4.1 Настройка сервера Ambari для HTTP с проверкой подлинности . . . 30
- 4.2 Настройка HTTP-аутентификации для HDFS, YARN, MapReduce2, HBase и Oozie . . . 30

В документе приведены сведения по настройке Ambari и Hadoop для Kerberos, расширенные параметры безопасности для Ambari и аутентификация SPNEGO для Hadoop.

Документ может быть полезен администраторам, программистам, разработчикам и сотрудникам подразделений информационных технологий, осуществляющих внедрение и сопровождение кластера.

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

Глава 1

Настройка Ambari и Hadoop для Kerberos

В данной главе описывается настройка **Kerberos** для надежной аутентификации пользователей и хостов **Hadoop** в кластере, управляемом **Ambari**:

- Обзор Kerberos;
- Принципалы Hadoop и Kerberos;
- Установка и настройка KDC;
- Подключение системы безопасности Kerberos в Ambari;
- Клиентские пакеты Kerberos;
- Отключение системы безопасности Kerberos в Ambari;
- Настройка шаблона атрибута;
- Перечень компонентов поддерживающих работу в Kerberos окружении.

Программное обеспечение Arenadata Hadoop включает возможность керберизации кластера внутренними средствами Ambari Server.

При использовании kerberos функции через Ambari Server:

- Сотрудники безопасности компании, не участвуют в настройке KDC.
- Администраторы Hadoop имеют полный контроль над установкой KDC. Управление keytab для кластера Hadoop полностью реализуется на стороне Ambari Server.
- Администраторы Hadoop несут дополнительную ответственность за управление KDC.
- Ответственность за любые уязвимости в системе безопасности несут администраторы Hadoop.
- Обеспечение высокой доступности KDC и аварийного восстановления - это ответственность администраторов Hadoop.
- Требуется ручная генерация keytab для любых разработчиков и стороннего программного обеспечения.

Important: Использование Kerberos может влиять на стабильность работы и производительность ряда сервисов кластера, ввиду необходимости использования дополнительных компонентов и параметров для аутентификации пользователей и взаимодействия сервисов внутри кластера.

Important: Некотрые сервисы требуют ручной настройки для работы с Kerberos и распространения тикетов внутри инфраструктуры для корректной работы.

Important: Перед активацией Kerberos на промышленном кластере, крайне рекомендуется провести опробацию функционала и работы всех приложений использующих сервисы Hadoop на тестовой среде.

Important: Для упрощения конфигурации и обеспечения более стабильной работы компонентов кластера Hadoop рекомендуется кербарезировать уровень операционной системы и не использовать Kerberos непосредственно для сервисов Hadoop.

1.1 Обзор Kerberos

Жесткая аутентификация и установление личности пользователя – основа безопасного доступа в **Hadoop**. Пользователи должны иметь возможность надежно "идентифицировать" себя, а затем использовать эту идентификацию во всем кластере **Hadoop**. Как только это будет сделано, данные пользователи могут получить доступ к ресурсам (например, к файлам или каталогам) или взаимодействовать с кластером (например, выполнять задания **MapReduce**). Помимо пользователей, сами ресурсы кластера **Hadoop** (такие как хосты и сервисы) должны проходить аутентификацию друг с другом, чтобы избежать потенциально опасных вредоносных систем или систем, "позиционирующих себя" как надежные компоненты кластера с целью получения доступа к данным.

Hadoop использует **Kerberos** в качестве основы для строгой аутентификации и обеспечения идентичности пользователям и сервисам. **Kerberos** является сторонним механизмом аутентификации, на который полагаются пользователи и сервисы для удостоверения подлинности друг друга. Сам сервер **Kerberos** известен как **Key Distribution Center** (**Центр распределения ключей**) или **KDC**. Он состоит из трех частей:

- База данных пользователей и сервисов (известных как принципалы), о которых он знает, и соответствующие пароли Kerberos;
- Сервер аутентификации (AS), который выполняет первоначальную проверку подлинности и выдает Ticket Granting Ticket (TGT);
- Ticket Granting Server (TGS) – сервер, который оформляет последующие билеты на основе начального TGT.

Пользователь-принципал запрашивает аутентификацию от **AS**. **AS** отправляет в ответ **TGT**, который зашифрован с использованием пароля пользователя-принципала **Kerberos**, известный только пользователю и **AS**. Пользователь-принципал расшифровывает **TGT** локально, используя свой пароль **Kerberos**, и с этого момента до истечения срока действия билета пользователь-принципал может использовать **TGT** для получения билетов от **TGS**. Данные билеты позволяют принципалу получить доступ к различным сервисам.

Поскольку ресурсы кластера (хосты или сервисы) не могут каждый раз предоставлять пароль для расшифровки **TGT**, они используют специальный файл *keytab*, который содержит учетные данные аутентификации ресурса. Набор хостов, пользователей и сервисов, над которыми сервер **Kerberos** имеет контроль, называется сферой.

Таблица1.1.: Термины и определения

Термин	Определение
Key Distribution Center, KDC	Надежный источник для аутентификации в экосистеме с поддержкой Kerberos
Сервер Kerberos KDC	Машина или сервер, который служит в качестве центра распределения ключей
Клиент Kerberos	Любая машина в кластере, которая аутентифицируется с KDC
Принципал	Уникальное имя пользователя или сервиса, который аутентифицируется с KDC
Keytab	Файл, содержащий один или несколько принципалов и их ключи
Сфера	Сеть Kerberos, включающая KDC и ряд клиентов
KDC Admin Account	Учетная запись администратора, используемая Ambari для создания принципов и генерации ключей в KDC

1.2 Принципалы Hadoop и Kerberos

Каждый сервис и под-сервис в **Hadoop** должны иметь своего принципала. Имя принципала в данной сфере состоит из основного имени и имени экземпляра – это полное доменное имя хоста, на котором работает сервер. Учетные данные серверов хранятся в файле *keytab*, который извлекается из базы данных **Kerberos** с помощью принципала сервера и хранится локально в защищенном каталоге на узле компонента сервера (Рис.1.1.).

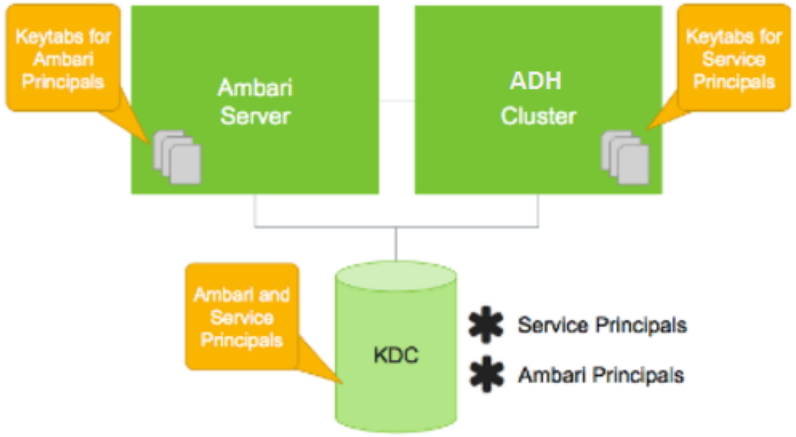


Рис.1.1.: Права доступа к кластеру

Пример условного обозначения имени принципалов и *Keytabs* приведен в таблице.

Таблица1.2.: Условное обозначение имени принципалов и Keytabs

	Условное обозначение	Пример
Principals	$service_component_name/$FQDN @EXAMPLE.COM	nn/c6401.ambari.apache.org@EXAMPLE.COM
Keytabs	$service_component_abbreviation. service.keytab	/etc/security/keytabs/nn.service.keytab

В дополнение к **Hadoop Service Principals**, сам **Ambari** также требует, чтобы набор Ambari-принципалов выполнял служебные “smoke” проверки и проверку работоспособности. Файлы *Keytab* для Ambari-принципалов, или “headless”, находятся на каждом хосте кластера, так же как и для принципалов сервиса.

В примере условного обозначения имени принципалов и *Keytabs* указано основное имя для каждого сервисного принципала. Основа имени, например, *nn* или *hive*, представляют собой соответственно сервис **NameNode** или **Hive**. К основному имени добавляется имя экземпляра и полное доменное имя хоста, на котором оно выполняется. Эта схема обеспечивает уникальное имя сервисам, которые работают на нескольких хостах, таких как **DataNodes** и **NodeManagers**. Добавление имени хоста служит для различия, например, запроса из **DataNode A** и запроса из **DataNode B**. Это важно по следующим причинам:

- Данные Kerberos для одного DataNode не подвергаются риску совпасть с данными других DataNodes;
- Если несколько DataNodes имеют одинаковый принципал и одновременно подключаются к одному NameNode, и если аутентификатор Kerberos имеет одинаковые временные метки, в таком случае аутентификация отклоняется как повторный запрос.

1.3 Установка и настройка KDC

Ambari может настроить **Kerberos** в кластере для работы с существующим **MIT KDC** или с существующей **Active Directory**. В данном разделе описываются шаги, необходимые для подготовки к интеграции.

Если у вас нет существующего **KDC** (**MIT** или **Active Directory**), необходимо установить новый **MIT KDC**.

Important: Установка KDC на узле кластера уже после установки клиента Kerberos может перезаписать созданный Ambari файл *krb5.conf*

При выборе автоматической настройки **Kerberos Ambari** самостоятельно подключается к **KDC**, создает необходимых принципалов, генерирует и распространяет *keytabs*. При выборе ручной настройки **Kerberos** необходимо вручную создавать принципалов, генерировать и распространять *keytabs*.

- Использование существующего MIT KDC;
- Использование существующей Active Directory;
- Ручная настройка Kerberos;
- Установка нового MIT KDC.

1.3.1 Использование существующего MIT KDC

Для использования существующего **MIT KDC** для кластера необходимо подготовить:

- Серверы Ambari и кластеры, имеющие сетевой доступ как к административным узлам KDC, так и к самому KDC;
- Учетные данные администратора KDC.

Дальнейшие действия описаны в разделе *Подключение системы безопасности Kerberos в Ambari*.

1.3.2 Использование существующей Active Directory

Для использования существующей **Active Directory** для кластера с автоматической установкой **Kerberos** необходимо подготовить:

- Серверы Ambari и кластеры, имеющие доступ к сети и DNS-именам Domain Controllers;
- Настроить конфигурацию LDAP (LDAPS) Active Directory;
- Пользовательскую Active Directory для принципалов. Например, *"OU = Hadoop, OU = People, dc = apache, dc = org"*;
- Учетные данные администратора Active Directory с с настроенным правом "Создание, удаление и управление учетными записями пользователей".

Дальнейшие действия описаны в разделе *Подключение системы безопасности Kerberos в Ambari*.

1.3.3 Ручная настройка Kerberos

Для ручной настройки **Kerberos** необходимо подготовить:

- Сетевой доступ узлов кластера к KDC;
- Установить утилиты клиента Kerberos (например, *kinit*) на каждом узле кластера;
- Установить расширения Java Cryptography (JCE) на хосте сервера Ambari Server и на всех узлах кластера;
- Вручную создать сервисные и Ambari принципалы в KDC перед выполнением мастера;
- Создать вручную и распространить ключи для принципалов сервисов и Ambari на узлы кластера перед выполнением мастера.

Дальнейшие действия описаны в разделе *Подключение системы безопасности Kerberos в Ambari*.

1.3.4 Установка нового MIT KDC

В данном разделе приведено подробное описание процесса установки **KDC**:

- Установка сервера KDC;
- Создание базы данных Kerberos;
- Запуск KDC;
- Создание администратора Kerberos.

Important: Поскольку Kerberos является точным к времени протоколом, все хосты в сфере должны синхронизироваться по времени, например, используя протокол сетевого времени (NTP)

Если локальное системное время клиента отличается от времени в **KDC** хотя бы на *5* минут, клиент не сможет аутентифицироваться.

Установка сервера KDC

Для установки сервера **KDC** необходимо выполнить следующие действия:

1. Установить новую версию сервера KDC:

- RHEL / CentOS:

```
yum install krb5-server krb5-libs krb5-workstation
```

- SLES:

```
Zypper install krb5 krb5-server krb5-client
```

2. Используя текстовый редактор, открыть файл конфигурации сервера KDC, расположенный по умолчанию в *Vi/etc/krb5.conf*;
3. Изменить раздел *[realms]* этого файла, заменив параметр *kerberos.example.com* для свойств *kdc* и *admin_server*, установленный по умолчанию с Fully Qualified Domain Name хоста сервера KDC, как показано в примере, где *kerberos.example.com* заменен на *my.kdc.server*:

```
[realms]
 EXAMPLE.COM = {
   kdc = my.kdc.server
   admin_server = my.kdc.server
}
```

Создание базы данных Kerberos

Для создания базы данных **Kerberos** необходимо использовать утилиту *kdb5_util*:

- RHEL / CentOS:

```
Kdb5_util create -s
```

- SLES:

```
Kdb5_util create -s
```

Запуск KDC

Для запуска сервера **KDC** и сервера администратора **KDC** необходимо выполнить команды:

- RHEL/CentOS 6:

```
/etc/rc.d/init.d/krb5kdc start
/etc/rc.d/init.d/kadmin start
```

- RHEL/CentOS 7:

```
systemctl start krb5kdc
systemctl start kadmin
```

- SLES 11:

```
rckrb5kdc start
rckadmind start
```

При установке и управлении собственным **MIT KDC** важно настроить сервер **KDC** на автоматический запуск при загрузке:

- RHEL/CentOS 6:

```
chkconfig krb5kdc on
chkconfig kadmin on
```

- RHEL/CentOS 7:

```
systemctl enable krb5kdc
systemctl enable kadmin
```

- SLES 11:

```
chkconfig rckrb5kdc on
chkconfig rckadmind on
```

Создание администратора Kerberos

Принципалы **Kerberos** могут быть созданы либо на самой машине **KDC**, либо через сеть, используя принципал *admin*. В последующей инструкции предполагается, что используется компьютер **KDC** и команда от утилиты администратора *kadmin.local*. Использование *kadmin.local* на машине **KDC** позволяет создавать принципалов без необходимости создания отдельного принципала-администратора перед началом работы.

При включении **Kerberos** для подключения **Ambari** к **KDC**, создания кластерных принципалов и генерации *keytabs* необходимо предоставить учетные данные администратора **Ambari**.

1. Создать администратора KDC, путем создания принципала-администратора:

   `Kadmin.local -q "addprinc admin / admin"`

2. Убедиться, что созданный администратор имеет права в ACL KDC. Открыть файл ACL KDC, используя текстовый редактор:

- RHEL / CentOS:

  `Vi /var/kerberos/krb5kdc/kadm5.acl`

- SLES:

  `Vi /var/lib/kerberos/krb5kdc/kadm5.acl`

3. Убедиться, что файл ACL KDC содержит запись, позволяющую принципал-администратору управлять KDC в используемой конкретной сфере. При использовании сферы, отличной от *EXAMPLE.COM*, необходимо убедиться, что есть запись для конкретной сферы. Например, для принципала *admin/admin@HADOOP.COM* следующая запись:

   `*/admin@HADOOP.COM *`

4. После редактирования и сохранения файла *kadm5.acl* необходимо перезапустить процесс *kadmin*:

- RHEL/CentOS 6:

  `/etc/rc.d/init.d/kadmin restart`

- RHEL/CentOS 7:

  `systemctl restart kadmin`

- SLES 11:

  `rckadmind restart`

1.4 Подключение системы безопасности Kerberos в Ambari

Независимо от того, какая выбрана настройка **Kerberos** – автоматическая или ручная – **Ambari** предоставляет мастера установки, помогающего включить **Kerberos** в кластере. В данном разделе содержится информация о подготовке **Ambari** перед запуском мастера и о шагах для его запуска.

- Установка JCE;
- Запуск мастера Kerberos.

Important: Необходимым условием для включения Kererbos является установка JCE на всех узлах кластера (включая сервер Ambari), имеющих хост сервера Ambari как часть кластера. Это говорит о том, что на сервере Ambari Server должен быть запущен агент Ambari

1.4.1 Установка JCE

Перед включением **Kerberos** в кластере необходимо развернуть файлы безопасности **Java Cryptography Extension** (**JCE**) на сервере **Ambari** и на всех узлах кластера.

Important: Если используется Oracle JDK, необходимо распространять и устанавливать JCE на всех узлах кластера, включая сервер Ambari. Обязательно требуется перезапустить сервер Ambari после установки JCE

Если используется **OpenJDK**, дистрибутивы **OpenJDK** устанавливаются автоматически с неограниченной мощностью **JCE** и, следовательно, установка **JCE** не требуется.

1. Для установки JCE необходимо на сервере Ambari получить файл JCE, подходящий для версии JDK на вашем кластере:

- Для Oracle JDK 1.8
- Для Oracle JDK 1.7

2. Архив с полученным файлом необходимо сохранить во временной папке;
3. На сервере Ambari и на каждом узле кластера добавить неограниченные права безопасности JCE:

   `$JAVA_HOME/jre/lib/security/`

   Например, выполнить следующие действия для извлечения прав из JDK, установленном на хосте:

   `unzip -o -j -q jce_policy-8.zip -d /usr/jdk64/jdk1.8.0_40/jre/lib/security/`

4. Перезапустить сервер Ambari;
5. Перейти к началу работы мастера безопасности.

1.4.2 Запуск мастера Kerberos

Ambari предоставляет три варианта по установке **Kerberos**:

- Через существующий MIT KDC;
- Через существующую Active Directory;
- Ручная настройка принципалов и keytabs Kerberos.

При выборе автоматической установки **Kerberos** – через существующий **MIT KDC** или **Active Directory** – мастер **Kerberos** запрашивает информацию, связанную с **KDC**: учетную запись администратора **KDC** и принципалов **Ambari**. После предоставления сведений **Ambari** автоматически создает принципалов, генерирует *keytabs* и распространяет их на хосты в кластере. Сервисы настраиваются для **Kerberos**, и сервисные компоненты перезапускаются для аутентификации с **KDC**. Подробное описание автоматической установки **Kerberos** приведено в разделе «Автоматическая настройка Kerberos».

При выборе ручной настройки **Kerberos** необходимо самостоятельно создавать принципалов и генерировать и распространять *keytabs*. Подробное описание ручной установки приведено в разделе "Ручная настройка Kerberos".

Автоматическая настройка Kerberos

Для автоматической настройки **Kerberos** необходимо выполнить следующие действия:

1. Необходимо убедиться, что KDC установлен и настроен, а также подготовлен JCE на каждом хосте в кластере;
2. Войти в Ambari-Web и перейти на вкладку *"Admin → Kerberos"*;
3. Нажать *Enable Kerberos*, чтобы запустить мастер;
4. Выбрать тип KDC, который используется, и подтвердить, что необходимые условия выполнены;
5. Предоставить информацию о KDC и учетной записи администратора;
6. Далее приведен перечень необязательных настроек:

- В поле "Домены" указать список шаблонов для сопоставления хостов в кластере с соответствующей сферой. Например, если хосты имеют общий домен в своем "FQDN", таком как *host1.mycompany.local* и *host2.mycompany.local*, необходимо установить следующее:

  ```
  .mycompany.local,mycompany.local
  ```

- Чтобы управлять клиентом Kerberos *krb5.conf* вручную (вместо управления им Ambari), развернуть раздел "Advanced krb5-conf" и снять флажок "Manage". При этом *krb5.conf* должен быть настроен на каждом хосте.
- Чтобы Ambari не установил клиентские библиотеки Kerberos на всех хостах, развернуть раздел "Advanced kerberos-env" и снять флажок "Install OS-specific Kerberos client package(s)". При этом должны быть установлены утилиты клиента Kerberos на каждом хосте.
- Если клиентские библиотеки Kerberos находятся в нестандартных папках, развернуть раздел "Advanced kerberos-env" и настроить опцию *Executable Search Paths*.
- Если KDC имеет пароль безопасности, развернуть раздел "Advanced kerberos-env" и настроить параметры пароля.
- Ambari проверяет настройку Kerberos, создав для этого тестового принципала. Чтобы переименовать его необходимо развернуть раздел "Advanced kerberos-env" и изменить наименование. По умолчанию тестовое имя принципала представляет собой комбинацию имени и даты кластера (*$ {cluster_name} - $ {short_date}*). Данный принципал будет удален после завершения теста.
- Если необходимо настроить атрибуты для принципалов, которые Ambari создает при использовании Active Directory, следует обратиться к разделу "Настройка шаблона атрибута" для получения дополнительной информации. При использовании MIT KDC можно передать параметры атрибута в разделе "Advanced kerberos-env". Например, можно установить параметры, относящиеся к *pre-auth* или *max* и обновить их:

  ```
  -requires_preauth -maxrenewlife "7 days"
  ```

7. Продолжить установку;
8. Ambari устанавливает клиентов Kerberos на хостах и проверяет доступ к KDC и возможность создания принципалов, генерации *keytab* и их распространения;
9. Настроить идентификаторы Kerberos, используемые Hadoop, и перейти к керберизации кластера.

На шаге "Configure Identities" (настройка идентификаторов) обязательно посмотреть имена принципалов, в частности, *Ambari Principals* в таблице "General". Эти имена, по умолчанию, добавляют имя кластера каждому принципалу Ambari. Можно оставить значение по умолчанию или изменить его, удалив *- $ {имя-кластера}* из

строки имени принципала. Например, если кластер назван *ADH*, а сфера – *EXAMPLE.COM*, то hdfs принципала создается как *hdfs-ADH@EXAMPLE.COM*.

10. Подтвердить конфигурацию. По желанию можно загрузить CSV-файл с принципалами и ключами для их автоматической генерации Ambari;
11. Нажать *Next* для начала процесса;
12. После создания принципалов, генерации и распространения ключей Ambari обновляет конфигурации кластера, а затем запускает и тестирует сервисы в кластере;
13. Завершить работу мастера после окончания процесса.

Ручная настройка Kerberos

Для ручной настройки **Kerberos** необходимо выполнить следующие действия:

1. Убедиться, что KDC установлен и настроен, а также подготовлен JCE на каждом хосте в кластере;
2. Войти в Ambari-Web и перейти на вкладку *"Admin → Kerberos"*;
3. Нажать *Enable Kerberos*, чтобы запустить мастер;
4. Выбрать параметр "Manage Kerberos principals" и "keytabs manually" и убедиться, что выполнены необходимые условия;
5. Предоставить информацию о KDC и учетной записи администратора.

- Если клиентские библиотеки Kerberos находятся в нестандартных папках, развернуть раздел "Advanced kerberos-env" и настроить опцию *Executable Search Paths*.

6. Настроить идентификаторы Kerberos, используемые Hadoop, и перейти к керберизации кластера.

На шаге "Configure Identities" (настройка идентификаторов) обязательно посмотреть имена принципалов, в частности, *Ambari Principals* в таблице "General". Эти имена, по умолчанию, добавляют имя кластера каждому принципалу Ambari. Можно оставить значение по умолчанию или изменить его, удалив - *$ {имя-кластера}* из строки имени принципала. Например, если кластер назван *ADH*, а сфера – *EXAMPLE.COM*, то hdfs принципала создается как *hdfs-ADH@EXAMPLE.COM*.

7. Подтвердить конфигурацию. Поскольку выбран параметр ручной настройки "Manual Kerberos Setup", необходимо получить CSV-файл со списком принципалов и ключей, необходимых для работы кластера с Kerberos.

Important: Не продолжайте работу до тех пор, пока вручную не будут созданы и распределены узлам кластера принципалы и ключи

---

8. Нажать *Next* для продолжения;
9. Ambari обновляет конфигурации кластера, а затем запускает и тестирует сервисы в кластере;
10. Завершить работу мастера после окончания процесса.

1.5 Клиентские пакеты Kerberos

При автоматическом подключении **Kerberos Ambari** устанавливает клиенты **Kerberos** на узлах кластера. В зависимости от операционной системы устанавливаются следующие пакеты:

Таблица1.3.: Пакеты, устанавливаемые в зависимости от ОС

Операционная система	Пакет
RHEL/CentOS 7	krb5-workstation
RHEL/CentOS 6	krb5-workstation
SLES 11	krb5-client

1.6 Отключение системы безопасности Kerberos в Ambari

Для отключения системы безопасности **Kerberos** в **Ambari** необходимо выполнить следующие действия:

- Войти в Ambari-Web и перейти в *"Admin → Kerberos"*;
- Нажать *Disable Kerberos*, чтобы запустить мастер;
- Завершить работу мастера.

Если **Kerberos** был подключен путем автоматической настройки, **Ambari** попытается связаться с **KDC** и удалить созданных принципалов. Если **KDC** недоступен, мастер выводит ошибку на шаге "Unkerberize". Ее можно игнорировать и продолжить работу мастера, но удаление принципалов из **KDC** не будет выполнено.

1.7 Настройка шаблона атрибута

При автоматической настройке **Kerberos** с **Active Directory** в зависимости от прав **KDC** можно настроить атрибуты принципалов, устанавливаемые **Ambari** при их создании. На шаге мастера "Configure Kerberos" в разделе "Advanced kerberos-env" есть доступ к шаблону атрибутов **Ambari**. Этот шаблон (который основан на синтаксисе шаблонов **Apache Velocity**) можно изменить, чтобы установить, какие атрибуты назначаются принципалам, и как эти значения получаются.

В таблице приведен список доступных переменных атрибутов.

Таблица1.4.: Доступные переменные атрибутов

Переменные атрибута	Пример
$normalized_principal	nn/c6401.ambari.apache.org@EXAMPLE.COM
$principal_name	nn/c6401.ambari.apache.org
$principal_primary	nn
$principal_digest	[[MD5 hash of the $normalized_principal]]
$principal_instance	c6401.ambari.apache.org
$realm	EXAMPLE.COM
$password	[[password]]

1.8 Перечень компонентов поддерживающих работу в Kerberos окружении

Таблица1.5.: Поддеживаемые компоненты

Сервис	Поддержка
Apache HDFS	Поддерживается
Apache YARN	Поддерживается
Apache MapReduce	Поддерживается
Apache Zookeeper	Поддерживается
Apache Tez	Поддерживается
Apache Hive	Поддерживается
Apache Hive LLAP	Не поддерживается
Apache HBase	Поддерживается
Apache Phoenix	Поддерживается
Apache Pig	Поддерживается
Apache Sqoop	Поддерживается
Apache Flume	Поддерживается
Apache Oozie	Поддерживается
Apache Atlas	Поддерживается не во всех типах окружения
Apache NiFi	Поддерживается
Apache Apex	Не поддерживается
Apache Flink	Поддерживается не во всех типах окружения
Apache Kafka	Поддерживается
Apache Knox	Поддерживается
Apache Mahout	Поддерживается
Apache Ranger	Поддерживается не во всех типах окружения
Apache Ranger KMS	Поддерживается не во всех типах окружения
Apache Solr	Поддерживается не во всех типах окружения
Apache Spark	Поддерживается
Apache Zeppelin	Поддерживается не во всех типах окружения
Apache Giraph	Не поддерживается
Apache Slider	Не поддерживается
Kafka Manager	Поддерживается
Logsearch	Поддерживается

Глава 2

Расширенные параметры безопасности для Ambari

2.1 Настройка Ambari для аутентификации LDAP или Active Directory

По умолчанию **Ambari** использует внутреннюю базу данных в качестве хранилища пользователя для аутентификации и авторизации. При необходимости настройки внешней аутентификации **LDAP** или **Active Directory**, следует ознакомиться с разделом "Настройка аутентификации пользователя LDAP" и запустить команду настройки (см. раздел "Настройка Ambari для использования LDAP-сервера").

Также необходимо синхронизировать пользователей и группы **LDAP** с базой данных **Ambari**, чтобы иметь возможность управлять их авторизацией и правами (см. раздел "Синхронизация пользователей и групп LDAP").

При синхронизации пользователей и групп **LDAP Ambari** использует элементы поиска **LDAP** для синхронизации большого количества объектов **LDAP**. Большинство современных серверов **LDAP** поддерживают этот элемент управления. Но для неподдерживающих серверов, например, **Oracle Directory Server Enterprise Edition 11g**, **Ambari** вводит параметр конфигурации для отключения разбивки на страницы. Чтобы отключить элементы управления разбивкой на страницы, необходимо установить значение *false* свойству *authentication.ldap.pagination.enabled* в файле */etc/ambari-server/conf/ambari-properties*. Это ограничит максимальное количество объектов, которые могут быть импортированы в любой момент времени до максимального предела результатов на сервере **LDAP**. Чтобы избежать этого, необходимо импортировать пользователей и группы, используя параметры *-sers* и *-groups*, как описано в разделе "Определенный набор пользователей и групп".

2.1.1 Настройка аутентификации пользователя LDAP

В таблице описаны свойства и значения, которые необходимо знать для настройки аутентификации **LDAP**.

Important: Если оставить установленное по умолчанию значение *false* для *bindAnonymously*, необходимо убедиться, что у вас есть имя и пароль LDAP-менеджера. Если будет использоваться SSL, необходимо убедиться, что сертификат и ключи к нему установлены

Таблица2.1.: Свойства и значения для настройки аутентификации LDAP

Свойство	Значение	Описание
authentication.ldap. primaryUrl	server:port	Имя хоста и порт для LDAP или сервера Active Directory (AD). Пример: my.ldap.server: 389
authentication.ldap. secondaryUrl	server:port	Имя хоста и порт для вторичного сервера LDAP или AD. Пример: my.secondary.ldap.server: 389 Необязательный параметр
authentication.ldap. useSSL	true or false	Если значение true – использовать SSL при подключении к LDAP или серверу AD
authentication.ldap. usernameAttribute	[LDAP attribute]	Атрибут для имени пользователя. Пример: uid
authentication.ldap. baseDn	[Distinguished Name]	root Distinguished Name поиска в каталоге для пользователей. Пример: ou=people,dc=hadoop, dc=apache,dc=org
authentication.ldap. referral	[Referral method]	Определяет необходимость следовать рекомендациям LDAP
authentication.ldap. bindAnonymously	true or false	Если значение true – привязать сервер LDAP или AD анонимно
authentication.ldap. managerDn	[Full Distinguished Name]	Если для параметра «Bind anonymous» установлено значение «false», «Distinguished Name» («DN») для менеджера. Пример: uid=hdfs,ou=people,dc=hadoop, dc=apache,dc=org
authentication.ldap. managerPassword	[password]	Если для параметра «Bind anonymous» установлено значение «false», пароль для менеджера
authentication.ldap. userObjectClass	[LDAP Object Class]	Класс объекта для пользователей. Пример: organizationalPerson
authentication.ldap. groupObjectClass	[LDAP Object Class]	Класс объекта для групп. Пример: groupOfUniqueNames
authentication.ldap. groupMembershipAttr	[LDAP attribute]	Атрибут принадлежности к группе. Пример: uniqueMember
authentication.ldap. groupNamingAttr	[LDAP attribute]	Атрибут имени группы

2.1.2 Настройка Ambari для использования LDAP-сервера

Important: Если используется LDAPS, и сертификат сервера LDAPS подписывается доверенным центром сертификации, то нет необходимости импортировать сертификат в Ambari, и поэтому данный раздел следует игнорировать

Если сервер **LDAPS** – самоподписанный сертификат или подписан неопознанным центром сертификации, таким как внутренний центр сертификации, необходимо импортировать сертификат и создать файл хранилища ключей. В следующем примере создается файл хранилища ключей в файле */keys/ldaps-keystore.jks*, но его можно создать в любом месте файловой системы. Для этого следует запустить команду настройки **LDAP** на сервере **Ambari** и ответить на вопросы:

1. Место, где каталог ключей не существует, но в котором должен быть создан:

   ```
   mkdir /etc/ambari-server/keys
   ```

2. ```
   $JAVA_HOME/bin/keytool -import -trustcacerts -alias root -file $PATH_TO_YOUR_LDAPS_CERT -keystore /etc/
   ↪ambari-server/keys/ldaps-keystore.jks
   ```

3. Задать пароль, который будет использоваться при настройке *ambari-server-ldap*, при появлении запроса:

   ```
   ambari-server setup-ldap
   ```

4. В запросе “Primary URL*” ввести IP/URL-адрес и порт сервера *(Пример: myldap.com:389)*. Требуются значения, отмеченные звездочкой.
5. В запросе “Secondary URL*” ввести IP/URL-адрес вторичного сервера и порт *(Пример: myldap-secondary.com:389)*. Необязательный параметр.
6. В запросе “Use SSL*” ввести свое значение. При использовании LDAPS ввести значение *true*.
7. В запросе “User object class*” ввести класс объектов, используемый для пользователей *(Пример: user)*.
8. В запросе “User name attribute*” ввести свое значение *(Пример: sAMAccountName)*. По умолчанию устанавливается – *uid*.
9. В запросе “Group object class*” ввести класс объекта, используемый для групп *(Пример: group)*.
10. В запросе “Group name attribute*” ввести атрибут имени группы *(Пример: CN)*.
11. В запросе “Group member attribute*” ввести атрибут принадлежности к группе *(Пример: member)*.
12. В запросе “Distinguished name attribute*” ввести атрибут, используемый для *distinguished name (Пример: distinguishedName)*.
13. В запросе “Base DN*” ввести свое значение *(Пример: CN=Ambari,OU=Hadoop,DC=EXAMPLE,DC=COM)*.
14. В запросе “Referral method*” ввести *follow* или *ignore* передачи LDAP.
15. В запросе “Bind anonymously*” ввести свое значение.
16. В запросе “Manager DN*” ввести свое значение в том случае, если в запросе “Bind anonymously” установлено значение *false (Пример: CN=admin,OU=Hadoop,DC=EXAMPLE,DC=COM)*.
17. В запросе “Enter the Manager Password*” ввести пароль DN менеджера LDAP.
18. Если на шаге 6 установлено значение *Use SSL = true*, появится следующий запрос:

    ```
    Do you want to provide custom TrustStore for Ambari?
    ```

    Возможны следующие варианты ответов:

- **Более безопасный вариант**. Если используется самоподписанный сертификат, который вы не хотите импортировать в существующее хранилище ключей JDK, следует ввести значение *y*.

Например, в случае, если вы хотите, чтобы данный сертификат использовался только Ambari, без других приложений, запущенных JDK на том же узле.

Если выбирается эта опция, то появятся дополнительные запросы:

- В запросе “TrustStore type” ввести *jks*;
- В запросе “Path to TrustStore file” ввести */keys/ldaps-keystore.jks* (или фактический путь к файлу хранилища ключей);
- В запросе “Password for TrustStore” ввести пароль для хранилища ключей.
- **Менее безопасный вариант**. Если используется самоподписанный сертификат, который вы хотите импортировать и хранить в существующем ключевом хранилище JDK, введите значение *n*.
    ```

- Преобразовать сертификат SSL в формат *X.509*, если это необходимо, выполнив следующую команду, где *<slapd.crt>* – путь к сертификату *X.509*:

```
openssl x509 -in slapd.pem -out <slapd.crt>
```

- Импортировать сертификат SSL в существующее хранилище ключей, например, хранилище сертификатов *jre* по умолчанию, используя следующую команду:

```
/usr/jdk64/jdk1.7.0_45/bin/keytool -import -trustcacerts -file slapd.crt -keystore
/usr/jdk64/jdk1.7.0_45/jre/lib/security/cacerts
```

Где Ambari настроен для использования JDK 1.7, поэтому сертификат должен быть импортирован в хранилище ключей JDK 7.

19. Проверить все настройки, и если они верны, выбрать значение *y*.

20. Запустить или перезапустить сервер:

```
ambari-server restart
```

Импортированным пользователям назначаются права пользователя **Ambari**. Они могут читать метрики, просматривать статус и конфигурацию сервисов, а так же просматривать информацию о задании. Чтобы пользователи могли запускать или останавливать сервисы, менять конфигурации и запускать *smoke tests*, им необходимо назначить права администратора **Ambari**. Для внесения данных изменений следует перейти по пунктам меню *"Manage Ambari → Users → Edit"*.

Пример конфигурации Active Directory

В **Directory Server** используются специальные классы объектов и атрибуты для хранения идентификаторов. В данном разделе в качестве примера отображаются конфигурации, характерные для **Active Directory**.

Таблица2.2.: Пример конфигурации AD

Запрос (значение по умолчанию)	Пример значений для Active Directory
User object class* (posixAccount)	user
User name attribute* (uid)	cn
Group object class* (posixGroup)	group
Group member attribute* (memberUid)	Member
Distinguished name attribute* (dn)	distinguishedName

2.1.3 Синхронизация пользователей и групп LDAP

Для синхронизации **LDAP** необходимо запустить команду и ответить на запрос:

```
ambari-server sync-ldap [option]
```

Important: Для выполнения операции необходимо запустить сервер Ambari

- При появлении запроса необходимо предоставить учетные данные администратора Ambari;
- При синхронизации LDAP локальные учетные записи пользователей с совпадающими именами будут переключаться на тип LDAP, что означает, что их аутентификация будет действовать против внешнего LDAP, а не в локальном хранилище пользователей Ambari;

- LDAP синхронизирует до 1000 пользователей. Если вы планируете импортировать более 1000 пользователей, необходимо при синхронизации использовать опцию *–users* и указать отфильтрованный список пользователей для выполнения импорта пакетами.

Утилита предоставляет три варианта синхронизации:

- Определенный набор пользователей и групп;
- Синхронизация существующих пользователей и групп в Ambari с LDAP;
- Все пользователи и группы.

По завершению синхронизации необходимо проверить файлы журналов неудачных попыток импорта на */var/log/ambari-server/ambari-server.log* на хосте сервера **Ambari**.

2.1.4 Определенный набор пользователей и групп

Для синхронизации определенного набора пользователей и групп из **LDAP** в **Ambari** необходимо использовать параметр:

```
ambari-server sync-ldap --users users.txt --groups groups.txt
```

Далее следует предоставить текстовый файл пользователей и групп, разделенных запятыми. Записи в каждом из этих файлов должны основываться на значениях атрибутов в **LDAP**, выбранных во время установки. Для файла *users.txt* должен использоваться атрибут "User name attribute", а для файла *groups.txt* – "Group name attribute". Эта команда найдет, импортирует и синхронизирует соответствующие объекты **LDAP** с **Ambari**.

Членство в группе определяется с помощью атрибута "groupMembershipAttr", имя пользователя – с помощью атрибута "usernameAttribute", указанных во время настройки **LDAP**.

2.1.5 Существующие пользователи и группы

После синхронизации определенного набора пользователей и групп, следующий параметр используется для синхронизации только тех объектов, которые находятся в **Ambari** с **LDAP**:

```
ambari-server sync-ldap --existing
```

Несуществующие в **LDAP** пользователи удаляются из **Ambari**, а членство в группе **Ambari** обновляется до соответствия **LDAP** (членство в группе определяется с помощью атрибута "groupMembershipAttr", указанного во время настройки **LDAP**).

2.1.6 Все пользователи и группы

В случае необходимости синхронизации всех пользователей и групп с **LDAP** в **Ambari** используется следующий параметр:

```
ambari-server sync-ldap --all
```

Это действие импортирует все объекты с соответствующими классами пользователей и групп **LDAP** в **Ambari**.

2.2 Настройка Ambari для Non-Root

В целях безопасности экосистемы ограничение доступа и сервисов, выполняемых с правами *root*, является жестким требованием. Для этих сред **Ambari** может быть настроена для работы без доступа *root*. Компоненты **Ambari Server** и **Ambari Agent** обеспечивают работу без прав *root*:

- Настройка Ambari Server для Non-Root;
- Настройка Ambari Agent для Non-Root.

2.2.1 Настройка Ambari Server для Non-Root

Для настройки запуска **Ambari Server** от пользователя (без прав *root*) во время процесса настройки ambari-сервера необходимо выбрать значение у при запросе:

```
Customize user account for ambari-server daemon?
```

В процессе установки предлагается использовать для пользователя, не являющегося *root*, *Ambari Server*, например: *ambari*.

Пользователь без прав *root*, который выбран для запуска сервера **Ambari**, должен входить в группу **Hadoop**. Эта группа должна соответствовать учетным записям службы **Hadoop**, указанным на вкладке *"Customize Services → Misc tab"* во время этапа настройки мастера установки. Имя группы, задающееся по умолчанию – *hadoop*. Если во время установки кластера название группы было изменено, необходимо убедиться, что пользователь, не являющийся пользователем *root*, входит в данную группу.

Если **Ambari Server** работает как пользователь без прав *root*, например, *ambari*, и планируется использовать **Ambari Views**, необходимо добавить следующие свойства в *"Services → HDFS → Configs → Advanced core-site"*:

```
hadoop.proxyuser.ambari.groups=*
hadoop.proxyuser.ambari.hosts=*
```

2.2.2 Настройка Ambari Agent для Non-Root

Ambari Agent можно настроить для запуска от пользователя без прав *root*. Для этого требуется специальный доступ *sudo* к учетным записям сервиса **Hadoop** и выполнения определенных привилегированных команд. Настройка агентов **Ambari** для работы в качестве *non-root* требует ручной установки агентов на всех узлах кластера (см. руководство "Инструкция по установке кластера"). После установки каждого агента необходимо настроить агента для запуска как пользователя без права *root*. В данном примере используется пользователь *ambari*.

Необходимо изменить в файле */etc/ambari-agent/conf/ambari-agent.ini* свойство *run_as_user*:

```
run_as_user=ambari
```

Далее для старта работы от пользователя без полномочий *root* необходимо перезапустить **Ambari Agent**.

Для запуска определенных команд, требующих дополнительные права, устанавливаемые в конфигурации **Sudoer**, функция *non-root* основывается на *sudo*. Конфигурация *sudo* разделена на части: настраиваемые пользователи, ненастраиваемые пользователи, команды и значения *sudo* по умолчанию.

В последующих разделах описано как следует настраивать *sudo*, чтобы позволить **Ambari** запускаться от пользователя без прав *root*. Каждый из разделов включает определенные записи *sudo*, которые необходимо поместить в */etc/sudoers* и запустить команду:

```
visudo
```

Настраиваемые пользователи

Данный раздел содержит команды "su" и соответствующие учетные записи сервиса **Hadoop**, которые настраиваются при установке:

```
# Ambari Customizable Users
ambari ALL=(ALL) NOPASSWD:SETENV: /bin/su hdfs *,/bin/su ambari-qa *,/bin/su ranger *,/bin/su zookeeper *,/
↪bin/su knox *,/bin/su ams *,/bin/su hbase *,/bin/su spark *,/bin/su hive *,/bin/su hcat *,/bin/su mapred␣
↪*,/bin/su oozie *,/bin/su tez *,/bin/su atlas *,/bin/su yarn *,/bin/su kms *
```

Учетные записи пользователей должны соответствовать учетным записям серверов, указанным на вкладке *"Customize Services → Misc tab"* во время этапа настройки мастера установки. Например, если **YARN** настроен для запуска как *xyz_ yarn*, необходимо изменить команду *su* на */bin/su xyz_ yarn*.

Ненастраиваемые пользователи

Данный раздел содержит команды "su" для системных учетных записей, которые нельзя изменить, и которые требуются только в том случае, если используется **MySQL**, установленный и управляемый **Ambari** для **Hive Metastore**. Если используется существующая база данных **MySQL**, **PostgreSQL** или **Oracle** для **Hive Metastore**, включать данные команды нет необходимости.

```
# Ambari Non-Customizable Users
ambari ALL=(ALL) NOPASSWD:SETENV: /bin/su mysql *
```

Команды

Команды, которые должны входить в стандартные операции агента:

```
# Ambari Commands
ambari ALL=(ALL) NOPASSWD:SETENV: /usr/bin/yum,/usr/bin/zypper,/usr/bin/apt-get, /bin/mkdir, /usr/bin/test,␣
↪/bin/ln, /bin/chown, /bin/chmod, /bin/chgrp, /usr/sbin/groupadd, /usr/sbin/groupmod, /usr/sbin/useradd, /
↪usr/sbin/usermod, /bin/cp, /usr/sbin/setenforce, /usr/bin/test, /usr/bin/stat, /bin/mv, /bin/sed, /bin/rm,
↪ /bin/kill, /bin/readlink, /usr/bin/pgrep, /bin/cat, /usr/bin/unzip, /bin/tar, /usr/bin/tee, /bin/touch, /
↪usr/bin/distro-select, /usr/bin/conf-select, /usr/phd/current/hadoop-client/sbin/hadoop-daemon.sh, /usr/
↪lib/hadoop/bin/hadoop-daemon.sh, /usr/lib/hadoop/sbin/hadoop-daemon.sh, /sbin/chkconfig gmond off, /sbin/
↪chkconfig gmetad off, /etc/init.d/httpd *, /sbin/service phd-gmetad start, /sbin/service phd-gmond start,␣
↪/usr/sbin/gmond, /usr/sbin/update-rc.d ganglia-monitor *, /usr/sbin/update-rc.d gmetad *, /etc/init.d/
↪apache2 *, /usr/sbin/service phd-gmond *, /usr/sbin/service phd-gmetad *, /sbin/service mysqld *, /usr/
↪bin/python2.6 /var/lib/ambari-agent/data/tmp/validateKnoxStatus.py *, /usr/phd/current/knox-server/bin/
↪knoxcli.sh *
```

```
# Ambari Ranger Commands
ambari ALL=(ALL) NOPASSWD:SETENV: /usr/phd/*/ranger-usersync/setup.sh, /usr/bin/ranger-usersync-stop, /usr/
↪bin/ranger-usersync-start, /usr/phd/*/ranger-admin/setup.sh *, /usr/phd/*/ranger-knox-plugin/disable-knox-
↪plugin.sh *, /usr/phd/*/ranger-hbase-plugin/disable-hbase-plugin.sh *, /usr/phd/*/ranger-hdfs-plugin/
↪disable-hdfs-plugin.sh *,  /usr/phd/current/ranger-admin/ranger_credential_helper.py, /usr/phd/current/
↪ranger-kms/ranger_credential_helper.py
```

Important: Не изменяйте списки команд, только имена пользователей могут быть изменены в разделе «Customizable Users»

Для повторной итерации необходимо выполнить данную конфигурацию *sudo* на каждом узле кластера. Чтобы убедиться, что конфигурация выполнена правильно, следует выполнить "su" для пользователя *ambari* и запустить *sudo -l*. Там можно проверить, нет ли предупреждений, и убедиться, что результат конфигурации соответствует только что примененному.

Значения Sudo по умолчанию

Некоторые версии *sudo* имеют конфигурацию по умолчанию, которая предотвращает вызов *sudo* из не интерактивной оболочки. Чтобы агент выполнял команды не интерактивно, некоторые значения по умолчанию необходимо перенастроить.

```
Defaults exempt_group = ambari
Defaults !env_reset,env_delete-=PATH
Defaults: ambari !requiretty
```

Для повторной итерации необходимо выполнить данную конфигурацию *sudo* на каждом узле кластера. Чтобы убедиться, что конфигурация выполнена правильно, следует выполнить "su" для пользователя *ambari* и запустить *sudo -l*. Там можно проверить, нет ли предупреждений, и убедиться, что результат конфигурации соответствует только что примененному.

2.3 Шифрование базы данных и паролей LDAP (опционально)

По умолчанию пароли доступа к базе данных **Ambari** и LDAP-серверу хранятся в простом текстовом файле. Для зашифровки паролей необходимо запустить специальную команду настройки.

Important: Во время шифрования паролей Ambari Server не должен быть запущен: либо внести изменения перед первым запуском сервера Ambari, либо остановить сервер для внесения изменений

1. На Ambari Server запустить команду настройки:

   `ambari-server setup-security`

2. При запросе "Choose one of the following options" выбрать вариант 2:

- [1] Включить HTTPS для сервера Ambari;
- [2] Шифровать пароли, хранящиеся в файле *ambari.properties*;
- [3] Настройка конфигурации JAAS Ambari kerberos;

3. Дважды ввести ключ для шифрования паролей (если пароли зашифрованы, необходим доступ к ключу, чтобы запустить Ambari Server).
4. Есть три варианта сохранения ключа:

- Перенесите его в файл на сервере, нажав *y* в строке;
- Создать переменную среду *AMBARI_SECURITY_MASTER_KEY* и установить на нее ключ;
- Вручную ввести ключ в командной строке при запуске сервера.

5. Запустить или перезапустить Ambari Server:

   `ambari-server restart`

2.3.1 Сброс шифрования

Сброс шифрования возможен в следующих случаях:

- Полное удаление шифрование;
- Изменение текущего мастер-ключа потому, что ключ забыт, либо для смены текущего ключа в целях безопасности.

Important: Во время сброса шифрования Ambari Server не должен быть запущен

Полное удаление шифрования

Для восстановления базы данных **Ambari** и паролей **LDAP** до полностью незашифрованного состояния необходимо выполнить следующие действия:

1. На хосте Ambari в текстовом редакторе открыть файл */etc/ambari-server/conf/ambari.properties* и установить свойство:

   `security.passwords.encryption.enabled=false`

2. Удалить:

   `/var/lib/ambari-server/keys/credentials.jceks`

3. Удалить:

   `/var/lib/ambari-server/keys/master`

4. Сбросить пароль базы данных и, при необходимости, пароль LDAP. Запустить настройку "ambari-server" (см. раздел "Шифрование базы данных и паролей LDAP (опционально)") и "setup-ldap ambari-server" (см. раздел "Настройка Ambari для использования LDAP-сервера").

Изменение текущего мастер-ключа

В случае если текущий мастер-ключ известен, для его изменения необходимо повторно запустить команду настройки шифрования и следовать инструкциям:

```
ambari-server setup-security
```

1. Из предложенных вариантов выбрать значение *2*:

- [1] Включить HTTPS для сервера Ambari;
- [2] Шифровать пароли, хранящиеся в файле *ambari.properties*;
- [3] Настройка конфигурации JAAS Ambari kerberos;

2. При запросе ввести текущий мастер-ключ;
3. В запросе "Do you want to reset Master Key" ввести значение *yes*;
4. В командной строке ввести новый мастер-ключ и подтвердить его.

В случае если текущий мастер-ключ неизвестен:

1. Полностью удалить шифрование (см. раздел "Полное удаление шифрования");
2. Произвести настройку мастер-ключа (как описано в начале текущего раздела):

   `ambari-server setup-security`

3. Запустить или перезапустить Ambari Server:

   `ambari-server restart`

2.4 Настройка SSL для Ambari (опционально)

Для ограничения доступа к серверу **Ambari** для соединений **HTTPS**, необходимо предоставить сертификат. Несмотря на то, что первоначально можно использовать самоподписанный сертификат, он не подходит для данной задачи. После того, как сертификат будет установлен, необходимо запустить специальную команду настройки.

Important: Во время настройки Ambari Server не должен быть запущен: либо внести изменения перед первым запуском сервера Ambari, либо остановить сервер для внесения изменений

1. Войти на хост Ambari Server;
2. Найти сертификат. Если создается временный самоподписанный сертификат, использовать его в качестве примера:

```
openssl genrsa -out $wserver.key 2048
openssl req -new -key $wserver.key -out $wserver.csr
openssl x509 -req -days 365 -in $wserver.csr -signkey $wserver.key -out $wserver.crt
```

Где *$wserver* – имя хоста сервера Ambari.

Используемый сертификат должен быть PEM-закодирован, а не DER-закодирован. Если использовать DER-закодированный сертификат, выдается следующая ошибка:

```
unable to load certificate 140109766494024:error:0906D06C:PEM routines:PEM_read_bio:no start line:pem_lib.c␣
↪:698:Expecting: TRUSTED CERTIFICATE
```

Для конвертации DER-закодированного сертификата в PEM-закодированный необходимо использовать следующую команду:

```
openssl x509 -in cert.crt -inform der -outform pem -out cert.pem
```

Где *cert.crt* – DER-закодированный сертификат и *cert.pem* – итоговый PEM-кодированный сертификат.

3. Запустить специальную команду настройки и ответить на запросы:

```
ambari-server setup-security
```

- Выбрать значение *1* для включения HTTPS для сервера Ambari;
- На запрос “Do you want to configure HTTPS?” ответить *y*;
- Выбрать порт для использования SSL. Номер порта, установленный по умолчанию – *8443*;
- Предоставить полный путь к файлу сертификата (*$wserver.crt from above*) и файлу закрытого ключа (*$wserver.key from above*);
- Ввести пароль для закрытого ключа;
- Запустить или перезапустить сервер:

```
ambari-server restart
```

2.5 Настройка Kerberos для сервера Ambari (опционально)

Когда кластер включен с **Kerberos**, конечные точки компонента **REST** (такие как компонент **YARN ATS**) требуют аутентификации **SPNEGO** (см. раздел “Аутентификация SPNEGO для Hadoop”).

В зависимости от сервисов в кластере **Ambari Web** нуждается в доступе к данным **API**. Также, такие представления, как **Tez View**, нуждаются в доступе к **ATS**. Поэтому сервер **Ambari** требует принципала **Kerberos** для аутентификации через **SPNEGO** в отношении этих **API**. В данном разделе описывается, как настроить сервер **Ambari** с помощью принципала **Kerberos** и *keytab*, чтобы позволить представлениям аутентифицироваться через **SPNEGO** по компонентам кластера.

1. Создать принципала в KDC для сервера Ambari. Например, используя *kadmin*:

   ```
   addprinc -randkey ambari-server@EXAMPLE.COM
   ```

2. Создать *keytab* для этого принципала:

   ```
   xst -k ambari.server.keytab ambari-server@EXAMPLE.COM
   ```

3. Поместить *keytab* на хост сервера Ambari. Обязательно установить права для файлов, чтобы запускающий Ambari Server пользователь, мог получить доступ к файлу *keytab*:

   ```
   /etc/security/keytabs/ambari.server.keytab
   ```

4. Остановить сервер Ambari:

   ```
   ambari-server stop
   ```

5. Запустить команду *setup-security*:

   ```
   ambari-server setup-security
   ```

6. Выбрать *3* для настройки Ambari kerberos JAAS;
7. Ввести имя принципала Kerberos для сервера Ambari, созданного на 1 шаге;
8. Ввести путь к *keytab* для принципала Ambari;
9. Перезапустить сервер Ambari:

   ```
   ambari-server restart
   ```

2.6 Настройка Truststore для сервера Ambari

При использовании шифрования для **Hadoop** необходимо настроить **Truststore Ambari** и добавить сертификаты.

Important: Во время настройки Ambari Server не должен быть запущен: либо внести изменения перед первым запуском сервера Ambari, либо остановить сервер для внесения изменений

1. Войти на хост Ambari Server;
2. Выбрать *4* для Setup truststore:

- На запрос “Do you want to import a certificate into Truststore?” ответить *y*;
- Ввести тип Truststore. Параметрами являются *jks*, *jceks* или *pks12*;
- Указать путь к файлу Truststore;
- Ввести пароль для Truststore и подтвердить его. Пароль должен содержать не менее 6 символов (Примечание: последние три шага требуются только при первичной настройке Truststore для Ambari);

3. Запустить или перезапустить Ambari Server:

   ```
   ambari -server restart
   ```

4. Выполнить настройку безопасности и выбрать пункт *5* для импорта сертификата в *truststore*.

2.7 Настройка двустороннего SSL между Ambari Server и Ambari Agents (опционально)

Двусторонний **SSL** обеспечивает шифрование связи между сервером **Ambari** и агентами **Ambari**. По умолчанию **Ambari** отправляет данные с отключенным двухсторонним **SSL**.

Important: Во время настройки Ambari Server не должен быть запущен: либо внести изменения перед первым запуском сервера Ambari, либо остановить сервер для внесения изменений

Для включения двухстороннего **SSL** необходимо:

1. На хосте сервера Ambari в текстовом редакторе открыть файл */etc/ambari-server/conf/ambari.properties*;
2. Добавить следующее свойство:

   ```
   security.server.two_way_ssl = true
   ```

3. Запустить или перезапустить Ambari Server:

   ```
   ambari -server restart
   ```

   Сертификаты агента автоматически загружаются во время регистрации агента.

2.8 Настройка шифров и протоколов для сервера Ambari (опционально)

Ambari обеспечивает контроль шифров и протоколов, которые доступны через **Ambari Server**.

Чтобы отключить определенные шифры, необходимо добавить список следующего формата в *ambari.properties* (при указании нескольких шифров, следует отделять каждый шифр с помощью нижнего подчеркивания):

```
security.server.disabled.ciphers=TLS_ECDHE_RSA_WITH_3DES_EDE_CBC_SHA
```

Чтобы отключить определенные протоколы, необходимо добавить список следующего формата в *ambari.properties* (при указании нескольких протоколов, следует отделять каждый протокол с помощью вертикальной черты):

```
security.server.disabled.protocols=SSL|SSLv2|SSLv3
```

Глава 3

Настройка двухсторонней SSL аутентификации для Ambari

Настройка двухсторонней SSL аутентификации для **Ambari** описана на примере 3 узлов:

- *server.local* – ambari-server, изначальная установка;
- *agent.local* – ambari-agent, изначальная установка;
- *agent2.local* – ambari-agent, добавляется позже.

3.1 Создание сертификатов

Для создания сертификатов необходимо выполнить следующие действия:

1. Сгенерировать сертификаты для каждого узла:

```
> openssl genrsa -out ca.key 2048
> openssl genrsa -out server.local.key 2048
> openssl genrsa -out agent.local.key 2048
> openssl genrsa -out agent2.local.key 2048
```

2. Сгенерировать самоподписной сертификат для УЦ:

```
> openssl req -new -x509 -key ca.key -out ca.crt
```

Параметр *Common name* на усмотрение, не критичен для **ambari**.

3. Сгенерировать *csr* для каждого узла:

```
> openssl req -new -key server.local.key -out server.local.csr
> openssl req -new -key agent.local.key -out agent.local.csr
> openssl req -new -key agent2.local.key -out agent2.local.csr
```

Параметр *Common name* важен и должен соответствовать **FQDN** узла.

4. Выпустить сертификаты на основе сгенерированных *csr*:

```
> openssl x509 -req -in server.local.csr -CA ca.crt -CAkey ca.key -CAcreateserial -out server.
↪local.crt
> openssl x509 -req -in agent.local.csr -CA ca.crt -CAkey ca.key -CAcreateserial -out agent.local.
↪crt
> openssl x509 -req -in agent2.local.csr -CA ca.crt -CAkey ca.key -CAcreateserial -out agent2.
↪local.crt
```

3.2 Включение SSL на стороне ambari-server

Для включения SSL на стороне *ambari-server* неообходимо последовательно выполнить шаги:

1. Остановить сервер:

```
> ambari-server stop
```

2. Выполнить набор команд:

```
> scp ca.crt server.local.crt server.local.key root@server.local:/root/
> mv /var/lib/ambari-server/keys /var/lib/ambari-server/keys.bak
> mkdir /var/lib/ambari-server/keys
> ambari-server setup-security (option 1 -- use crt and key from scp step)
> openssl pkcs12 -export -in /root/server.local.crt -inkey /root/server.local.key -certfile /root/
↪server.local.crt -name 1 -out /var/lib/ambari-server/keys/keystore.p12
```

При этом утилита требует пароль для шифрования keystore. Пароль также необходимо поместить в файл */var/lib/ambari-server/keys/pass.txt*:

```
> echo "keystore_pass" > /var/lib/ambari-server/keys/pass.txt
> keytool -importcert -alias 2 -file /root/ca.crt -keystore /var/lib/ambari-server/keys/keystore.
↪p12 -storepass `cat /var/lib/ambari-server/keys/pass.txt`
```

Для того, чтобы удостовериться, что в keystore импортированы неообходимые сертификаты, а именно сертификаты *ambari-server* и УЦ, выполнить действие:

```
> keytool -list -v -keystore /var/lib/ambari-server/keys/keystore.p12 -storepass `cat /var/lib/
↪ambari-server/keys/pass.txt`
```

3. Запустить сервер:

```
> ambari-server start
```

3.3 Включение двустороннего SSL шифрования

Для включения двустороннего SSL шифрования необходимо:

1. На узле с *ambari-server* выполнить команду:

```
> ambari-server stop
```

После чего добавить в файл */etc/ambari-server/conf/ambari.properties* параметр: *security.server.two_way_ssl=true*.

2. На узлах *ambari-agent*:

```
> ambari-agent stop
> scp ca.crt agent.local.crt agent.local.key root@agent.local:/var/lib/ambari-agent/keys/
```

3. Стартовать *ambari-server* и *ambari-agent*:

```
> ambari-server start
> ambari-agent start
```

3.4 Добавление новых нод

Добавление новых нод осуществляется следующим образом:

1. На новых узлах *ambari-agent* выполнить:

```
> yum install ambari-agent
> scp ca.crt agent2.local.crt agent2.local.key root@agent2.local:/var/lib/ambari-agent/keys/
```

После чего в файле */etc/ambari-agent/conf/ambari-agent.ini* изменить параметр *hostname=server.local* и запустить *ambari-agent*:

```
> start ambari-agent
```

2. В UI *ambari-server* пройти процедуру добавления нового узла, выбрав опцию *Perform manual registration on hosts and do not use SSH* и подтвердив регистрацию кнопкой *Register and confirm*.

Глава 4

Аутентификация SPNEGO для Hadoop

По умолчанию доступ к HTTP-сервисам и пользовательскому интерфейсу кластера не настроен на необходимость аутентификации. Аутентификацию **Kerberos** можно настроить для веб-интерфейсов **HDFS**, **YARN**, **MapReduce2**, **HBase** и **Oozie**.

4.1 Настройка сервера Ambari для HTTP с проверкой подлинности

Для работы **Ambari** с кластером, требующим аутентифицированный HTTP-доступ к веб-интерфейсу, необходимо настроить сервер **Ambari** для **Kerberos**. Подробное описание настроек приведено в разделе "Настройка Kerberos для сервера Ambari (опционально)".

4.2 Настройка HTTP-аутентификации для HDFS, YARN, MapReduce2, HBase и Oozie

Для настройки HTTP-аутентификации для **HDFS**, **YARN**, **MapReduce2**, **HBase** и **Oozie** необходимо выполнить следующие действия:

1. Создать секретный ключ, используемый для подписания токенов аутентификации. Этот файл должен содержать случайные данные и размещаться на каждом узле кластера. Он также должен принадлежать пользователям и группам *hdfs*, входящим в группу *hadoop*. Права должны быть установлены на *440*. Например:

```
dd if=/dev/urandom of=/etc/security/http_secret bs=1024 count=1
chown hdfs:hadoop /etc/security/http_secret
chmod 440 /etc/security/http_secret
```

2. В Ambari Web перейти по вкладкам *"Services → HDFS → Configs"*;
3. Добавить или изменить свойства конфигурации в *Advanced core-site*, приведенные в таблице.

Таблица4.1.: Новые значения свойств конфигурации в Advanced core-site

Свойство	Новое значение
hadoop.http.authentication.simple. anonymous.allowed	false
hadoop.http.authentication. signature.secret.file	/etc/security/http_secret
hadoop.http.authentication.type	kerberos
hadoop.http.authentication. kerberos.keytab	/etc/security/keytabs/spnego.service.keytab
hadoop.http.authentication. kerberos.principal	HTTP/_HOST@ **EXAMPLE.COM**
hadoop.http.filter.initializers	org.apache.hadoop.security.AuthenticationFilterInitializer
hadoop.http.authentication. cookie.domain	**mycompany.local**

Выделенные в таблице записи зависят от сайта. Свойство *hadoop.http.authentication.cookie.domain* основано на полностью доменных именах серверов в кластере. Например, если **FQDN** вашего **NameNode** – *host1.mycompany.local*, то *hasoop.http.authentication.cookie.domain* должен быть установлен в *mycompany.local*.

4. Сохранить настройки и перезапустить соответствующие сервера.