

АО «РТ Лабс»

ИНСТРУКЦИЯ ПО УСТАНОВКЕ

«RTL BIOMETRICS PLATFORM»

АО «РТ ЛАБС»

На 46 листах

Москва 2018

ОГЛАВЛЕНИЕ

АННОТАЦИЯ	4
ТЕРМИНЫ И ОПРЕДЕЛЕНИЯ.....	5
1. ОБЩИЕ СВЕДЕНИЯ	6
1.1. Назначение документа	6
1.2. Уровень подготовки пользователей	6
1.3. Связанные документы.....	6
2. УСТАНОВКА И НАСТРОЙКА HADOOP (AMBARI)	6
2.1. Основные рекомендации по преднастройке серверов	6
2.2. Подготовка серверов	6
2.3. Включение и синхронизация ntpd-сервиса	6
2.4. Разметка диска для метрик сервера управления	6
2.5. Разметка диска для серверов hbase, fdfs, datanode	7
2.6. JAVA (JDK1.8.181).....	9
2.7. Установка AMBARI-AGENT	9
2.8. Установка AMBARI-SERVER.....	10
2.9. Администрирование Ambari Server	10
3. УСТАНОВКА И НАСТРОЙКА POSTGRESQL 10.....	20
3.1. Подготовка серверов PGSQL	20
3.1.1. Разметка диска по стандарту	20
3.1.2. Установка пакетов и настройка базы данных.....	21
3.2. Установка postgresql 10.....	21
3.3. Конфигурация базы данных	23
3.4. Перезапуск сервиса для применения конфигураций	26
3.5. Скрипт для создания системных и прикладных пользоателей	26
3.5.1. Создание базы данных bsb, faceis_db	27
4. УСТАНОВКА И НАСТРОЙКА RABBITMQ.....	27
4.1. Подготовка серверов	27
4.2. Разметка диска	27
4.3. Установка Rabbitmq.....	28
4.4. Подключение LUNA	28
5. УСТАНОВКА И НАСТРОЙКА LUNA STACK.....	29
5.1. Подготовка серверов	29
5.2. Разметка диска	29
5.3. Установка LUNA	30

5.4.	Установка лицензии	30
5.5.	Конфигурирование Aerospike DB	30
5.6.	Настройка LUNA	31
5.6.1.	LUNA-API	31
5.6.2.	LUNA-ADMIN	33
5.6.3.	LUNA-UI	33
5.7.	Регистрации в системе LUNA-UI и получение токена выполняется двумя способами:	34
6.	УСТАНОВКА И НАСТРОЙКА JAVA STACK.....	36
6.1.	Разметка диска	36
6.2.	создание системного пользователя BSB	37
6.3.	JAVA (JDK1.8.181).....	37
6.4.	Установка КриптоПро CSP 4.0 для Linux (x64, rpm)	37
6.5.	JCSP 5.0.39267	38
6.6.	JCSP: Установка ключей УКЭП Банка для СМЭВ	39
6.7.	FRONTEND (BSB-WEB.JAR)	41
6.7.1.	Подготовка OS (Под пользователем root)	41
6.7.2.	Установка FRONTEND (BSB-WEB.WAR)	41
6.7.3.	Конфигурация FRONTEND (BSB-WEB.WAR)	41
6.7.4.	Настройки логирования	42
6.8.	BACKEND (BSB-BKG.WAR)	42
6.8.1.	Подготовка OS (Под пользователем root)	42
6.8.2.	Установка BACKEND (bsb-bkg.war)	43
6.8.3.	Конфигурация BACKEND (bsb-bkg.war)	43
6.8.4.	Использование встроенного механизма подписания.....	43
6.8.5.	Использование внешнего сервера подписания HSM	43
6.8.6.	Настройки логирования	44
7.	УСТАНОВКА И НАСТРОЙКА NGINX (РАЗДАЧА СТАТИКИ, БАЛАНСИРОВКА)	44
7.1.	Подготовка O.S.	44
7.2.	Разметка диска	44
7.3.	Установка NGINX	45
7.3.1.	Конфигурации NGINX.....	45
7.4.	Статика (BSB-FRONT.TAR.GZ)	46

АННОТАЦИЯ

В настоящем документе представлено руководство системного администратора для эксплуатации Информационной системы «Биометрическая система банка».

Руководство системного администратора предназначено для работников Банка, задействованных в процессе:

- установка, настройка и эксплуатация серверного и прикладного программного обеспечения.

В рамках настоящего руководства приведено описание пошаговой установки и настройки серверного программного обеспечения Информационной системы «Биометрическая система банка» в Standalone конфигурации, описана и проиллюстрирована последовательность действий для выполнения вышеприведённых процессов.

Настоящее руководство состоит из следующих основных частей:

1. Установка и настройка Hadoop/Ambari (Хранилище данных).
2. Установка и настройка PostgreSQL 10 (РБД).
3. Установка и настройка RabbitMQ (Сервер очередей).
4. Установка и настройка Luna Stack (Luna-api, Luna-admin, Luna-ui).
5. Установка и настройка Java, JCSP и приложений BSB.
6. Установка и настройка NGINX (Балансировка компонентов, раздача статического контента).

ТЕРМИНЫ И ОПРЕДЕЛЕНИЯ

ТЕРМИН	ОПРЕДЕЛЕНИЕ
Авторизация	Предоставление определённому лицу или системе прав на выполнение определённых действий, а также процесс проверки (подтверждения) данных прав при попытке выполнения этих действий. Перед авторизацией выполняется аутентификация
Аутентификация	Процесс сопоставления предъявленных неизвестным лицом или системой сведений с имеющимися (полученными ранее из доверенного источника) сведениями зарегистрированных в ИС пользователей или систем с целью подтверждения факта принадлежности учётной записи лицу или системе и, как следствие, установления доверенных отношений в контексте сессии информационного взаимодействия. Аутентификация предшествует авторизации
Библиотека контроля качества	Предназначена для автоматического первичного контроля качества снимаемых биометрических данных клиента в процессе биометрической регистрации. Является модулем Системы
ИС	Информационная Система
Методические рекомендации по работе со СМЭВ	Методические рекомендации по работе с Единой системной межведомственного электронного взаимодействия. Описываемые в документе правила являются обязательными к применению участниками информационного обмена с использованием системы межведомственного электронного взаимодействия. Версия документа доступна для загрузки с официального портала СМЭВ версии 3: https://smev3.gosuslugi.ru/portal/
Микросервис	Комплекс программных компонент, реализованных на единой программной платформе и подлежащих повторному использованию
Системный пользователь	Лицо, имеющее системную учетную запись в системе Linux
Система межведомственного электронного взаимодействия	Часть инфраструктуры электронного правительства, отвечает за выполнение транспортных функций при выполнении информационных взаимодействий информационных систем органов власти (федерального, регионального уровня), порталом государственных услуг РФ, и информационных систем других организаций (в т.ч. коммерческих), которым предоставлен доступ к системе
УКЭП Банка для СМЭВ	Усиленная квалифицированная электронная подпись, используемая в автоматизированном режиме Системы (Коннектор СМЭВ) для доступа к видам сведений СМЭВ. Тип электронной подписи в терминологии СМЭВ: «ЭП-ОВ» (см. раздел №4 «Электронные подписи» Методических рекомендаций по работе со СМЭВ, версии 3.X)

1. ОБЩИЕ СВЕДЕНИЯ

1.1. НАЗНАЧЕНИЕ ДОКУМЕНТА

Документ «Руководство системного администратора» предназначен для системных инженеров и системных администраторов.

1.2. УРОВЕНЬ ПОДГОТОВКИ ПОЛЬЗОВАТЕЛЕЙ

Системный администратор должен иметь знания и опыт в области настройки и администрирования Microsoft Windows и Linux, настройки и администрирования СУБД Oracle и PostgreSQL, создания/ хранения/ восстановления резервных копий данных Системы, установки и настройки СКЗИ «КриптоПро CSP» и «КриптоПРО JavaCSP», установки обновлений и настройки операционной системы, установки оборудования и настройки рабочего места для снятия биометрии (веб-камера, микрофон).

1.3. СВЯЗАННЫЕ ДОКУМЕНТЫ

Инструкция по настройке масштабирования и отказоустойчивости БСБ приведена в документе «Масштабирование и отказоустойчивость конфигурации БСБ».

2. УСТАНОВКА И НАСТРОЙКА HADOOP (AMBARI)

2.1. ОСНОВНЫЕ РЕКОМЕНДАЦИИ ПО ПРЕДНАСТРОЙКЕ СЕРВЕРОВ

- Для каждого сервера в роли DataNode следует подключать по отдельному HDD. Предпочтительно обычные (не SSD) диски
- Размечать следует только основной раздел без использования LVM
- Наиболее предпочтительно использовать файловые системы ext4, xfs

2.2. ПОДГОТОВКА СЕРВЕРОВ

На сервере управления d00bsb-adm01 необходимо создать ssh ключи (ssh-keygen -t rsa) и выполнить авторизацию (ssh-copy-id) на всех серверах кластера hdfs для беспарольного входа по SSH (hd01, hd02, hd03).

Подключить репозиторий на всех серверах

```
wget -nv http://public-repo-1.hortonworks.com/ambari/sles12/2.x/updates/2.6.2.0/ambari.repo -O /etc/zypp/repos.d/ambari.repo
```

2.3. ВКЛЮЧЕНИЕ И СИНХРОНИЗАЦИЯ NTPD-СЕРВИСА

Должен быть включён и синхронизирован ntpd сервис

```
systemctl enable ntpd  
systemctl restart ntpd
```

2.4. РАЗМЕТКА ДИСКА ДЛЯ МЕТРИК СЕРВЕРА УПРАВЛЕНИЯ

```
pvccreate /dev/sdb  
vgcreate vg_data /dev/sdb
```

Создание раздела для метрик Ambari необходимо выполнять на мастер-сервере

1. Задать размер логического раздела диска более 15 GB

```
mkdir /ambari
lvcreate --name lv_ambari --size 10G vg_data
mkfs.xfs /dev/vg_data/lv_ambari
```

2. Прописать в /etc/fstab точку монтирования

```
/dev/vg_data/lv_ambari  /ambari          xfs      defaults      1 1
```

3. Примонтировать

```
mount -a
```

2.5. РАЗМЕТКА ДИСКА ДЛЯ СЕРВЕРОВ HBASE, FDFS, DATANODE

1. Выполнить команду для начала разметки выделенного диска /dev/sdb

```
fdisk /dev/sdb

Welcome to fdisk (util-linux 2.29.2).
Changes will remain in memory only, until you decide to write them.
Be careful before using the write command.

Device does not contain a recognized partition table.
...
```

2. Выполнить команду для получения информации о диске

```
Command (m for help): p
Disk /dev/sdb: 35 GiB, 37580963840 bytes, 73400320 sectors
Units: sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
Disklabel type: dos
Disk identifier: 0xa9c86609
```

3. Выполнить следующие команды в консоли fdisk для задания размера и создания Primary раздела

```
- Ввести: n
Command (m for help): n
Partition type
  p   primary (0 primary, 0 extended, 4 free)
  e   extended (container for logical partitions)
Select (default p):
  - Ввести: <<Enter>>

Using default response p.
  - Ввести: <<Enter>>

Partition number (1-4, default 1):
```

- Ввести: <<Enter>>

First sector (2048-73400319, default 2048):

- Ввести: <<Enter>>

Last sector, +sectors or +size{K,M,G,T,P} (2048-73400319, default 73400319):

- Ввести: <<Enter>>

Created a new partition 1 of type 'Linux' and of size 35 GiB.

При наличии уже созданной файловой системы, будет предложено переписать данные. Соглашаемся

Partition #1 contains a xfs signature.

Do you want to remove the signature? [Y]es/[N]o: Y

The signature will be removed by a write command.

- Запишем изменения
- Ввести: w

Command (m for help): w

The partition table has been altered.

Calling ioctl() to re-read partition table.

Syncing disks.

4. Теперь можно создать файловую систему и отредактировать /etc/fstab для постоянного монтирования. Создать директорию /hadoop командой

```
mkdir -p /hadoop
```

5. Отформатировать новый раздел в XFS

```
mkfs.xfs /dev/sdb1
...
meta-data=/dev/sdb1      isize=256    agcount=4, agsize=2293696 blks
        =               sectsz=512   attr=2, projid32bit=1
        =               crc=0        finobt=0, sparse=0
data      =               bsize=4096   blocks=9174784, imaxpct=25
        =               sunit=0      swidth=0 blks
naming    =version 2      bsize=4096   ascii-ci=0 ftype=1
log       =internal log   bsize=4096   blocks=4479, version=2
        =               sectsz=512   sunit=0 blks, lazy-count=1
realtime  =none           extsz=4096   blocks=0, rtextents=0
```

6. Добавить следующую строчку в /etc/fstab

```
/dev/sdb1    /hadoop      xfs     defaults,noatime    0 0
```

7. Смонтировать командой

```
mount -a
```

8. Проверим доступность

```
df -h | grep hadoop
```

```
/dev/sdb1          35G  33M  35G  1% /hadoop
```

2.6. JAVA (JDK1.8.181)

Дистрибутив для установки доступен по ссылке: <https://portal.rtlabs.ru/~cl8Lr> (JDK1.8)

1. Создать директорию для установки: `mkdir /usr/java`
2. Положить установочный пакет в `/usr/java/jdk-8u181-linux-x64.rpm`
3. Выполнить стандартную установку на всех серверах

```
zypper install /usr/java/jdk-8u181-linux-x64.rpm -f
```

4. В ходе установке подтверждаем все положительно

```
jdk1.8-2000:1.8.0_181-fcs.x86_64 (Plain RPM files cache): Signature verification failed [4-Signatures public key is not available]
Abort, retry, ignore? [a/r/i] (a): i
```

Результат: JAVA установлена.

2.7. УСТАНОВКА AMBARI-AGENT

1. Выполнить на всех нодах hd01, hd02, hd03 следующие команды

```
zypper install snappy snappy-devel
zypper install ambari-agent
```

2. В файле конфигурации `/etc/ambari-agent/conf/ambari-agent.ini` на всех серверах указать

```
[server]
hostname=<fqdn сервера ambari (d00bsb-adm01)
```

3. В блоке `[security]` добавить строку:
`force_https_protocol=PROTOCOL_TLSv1_2.`
4. Запустить `ambari-agent` и включить в автозагрузку на всех серверах

```
systemctl enable ambari-agent
systemctl restart ambari-agent
```

5. Проверить лог `/var/log/ambari-agent/ambari-agent.log` и убедиться об успешном запуске

ВНИМАНИЕ

Сообщение вида «Server at `https://t00bsb-hd01:t00bsb-hd01:8440` is not reachable», говорит о том, что сервер `ambari` не доступен

2.8. УСТАНОВКА AMBARI-SERVER

1. Выполнить на нодe d00bsb-hd01 (мастер сервере) команду

```
zypper install ambari-server
```

2. Запустить первоначальную конфигурацию, выполнив команду

```
ambari-server setup -j /usr/java/default/jre
Customize user account for ambari-server daemon [y/n] (n)?
Ввести «n» и нажать клавишу «enter»

Enable Ambari Server to download and install GPL Licensed LZO packages [y/n] (n)?
Ввести «n» и нажать клавишу «enter»

Do you accept the Oracle Binary Code License Agreement [y/n] (y)?
Ввести «y» и нажать клавишу «enter»

Enter advanced database configuration [y/n] (n)?
Ввести «n» и нажать клавишу «enter»

Configuring database...
Default properties detected. Using built-in database.
Configuring ambari database...
Checking PostgreSQL...
Running initdb: This may take up to a minute.
About to start PostgreSQL
Configuring local database...
Configuring PostgreSQL...
```

3. Запустить ambari-server, выполнив команду:

```
systemctl restart ambari-server
```

4. Включить сервис в автозагрузку, выполнив команду

```
systemctl enable ambari-server
```

2.9. АДМИНИСТРИРОВАНИЕ AMBARI SERVER

Для продолжения установки и конфигурации открыть страницу администрирования Ambari Server <http://d00bsb-adm01:8080/>

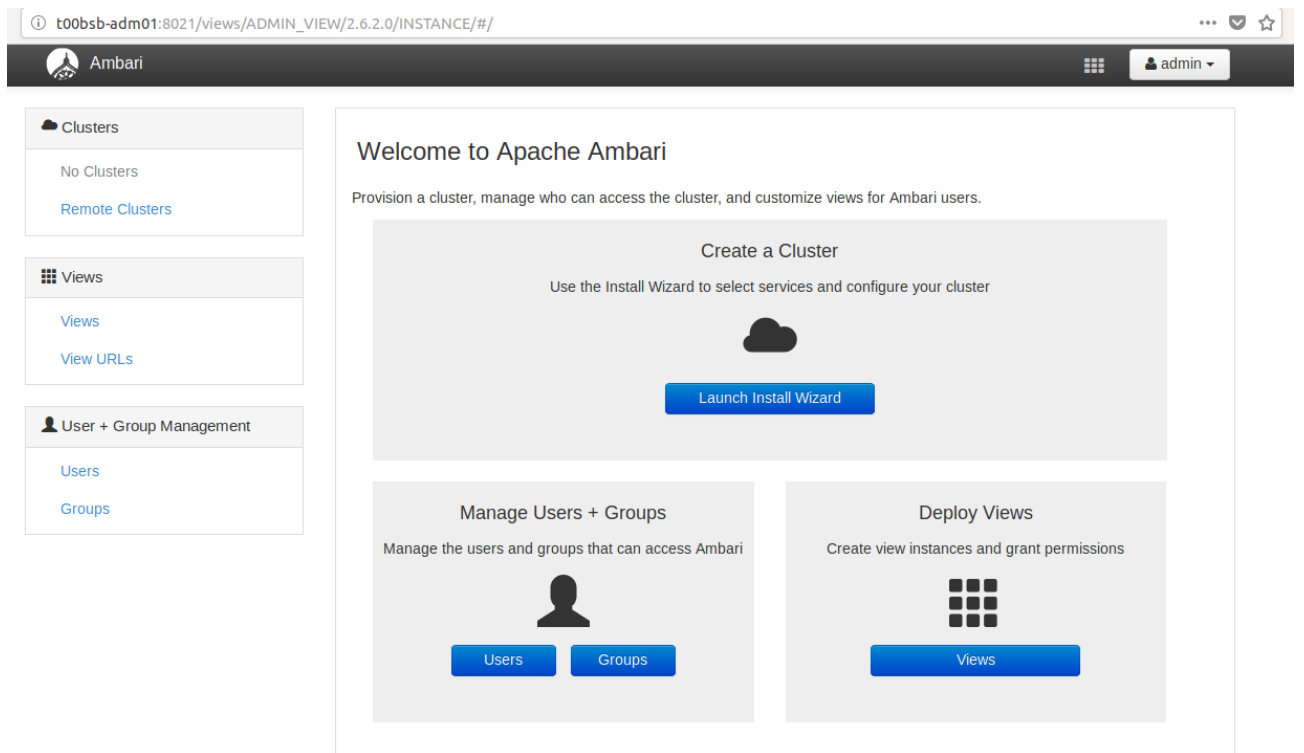


Рисунок 1.

Ввести данные для входа (login: admin, password: ***) и последовательно выполнить следующие действия:

1. Нажать кнопку «Launch Install Wizard»

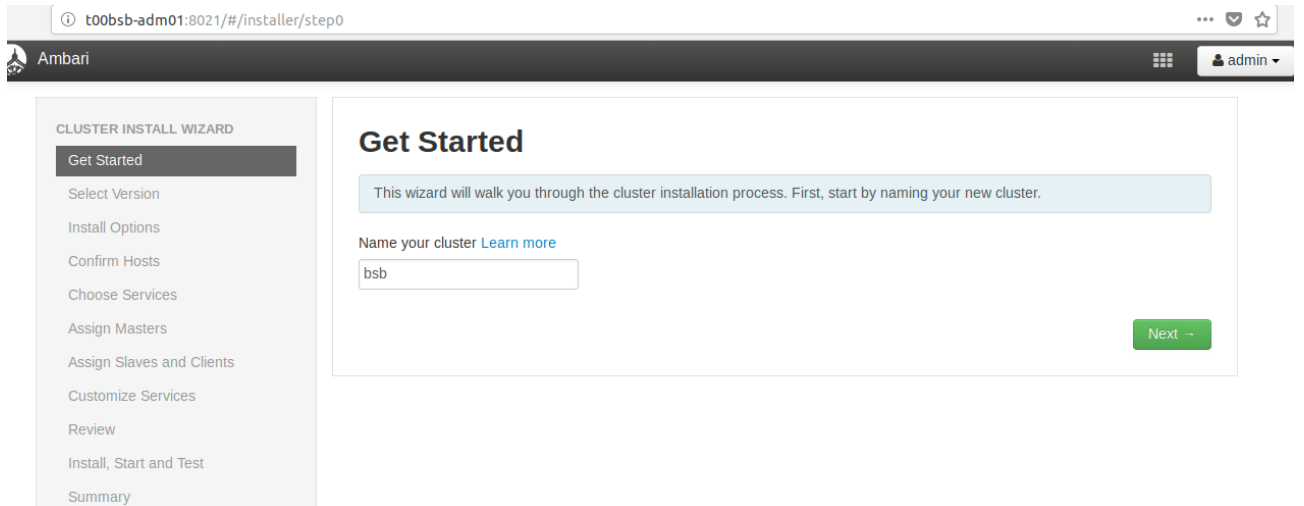


Рисунок 2.

2. Задать имя кластера bsb и нажать кнопку «Next»

Ambari

admin

CLUSTER INSTALL WIZARD

Get Started

Select Version

Install Options

Confirm Hosts

Choose Services

Assign Masters

Assign Slaves and Clients

Customize Services

Review

Install, Start and Test

Summary

Select Version

Select the software version and method of delivery for your cluster. Using a Public Repository requires Internet connectivity. Using a Local Repository requires you have configured the software in a repository available in your network.

HDP-2.6

HDP-2.5

HDP-2.4

HDP-2.6.5.0

Accumulo1.7.0

Ambari Infra0.1.0

Ambari Metrics0.1.0

Atlas0.8.0

Druid0.10.1

Falcon0.10.0

Flume1.5.2

Use Public Repository

Use Local Repository

Repositories

Provide Base URLs for the Operating Systems you are configuring.

OS	Name	Base URL	
suse12	HDP-2.6	http://public-repo-1.hortonworks.com/HDP/sles12/2.x/up	Remove
	HDP-UTILS-1.1.0.22	http://public-repo-1.hortonworks.com/HDP-UTILS-1.1.0.22	

☐ Skip Repository Base URL validation (Advanced)

☐ Use RedHat Satellite/Spacewalk

Back

Next

Рисунок 3.

3. Выбрать версию HDP-2.6, выбрать «Use Public Repository» и удалить иные репозитории, все кроме suse 12. Нажать кнопку «Next»

Ambari

CLUSTER INSTALL WIZARD

- Get Started
- Select Version
- Install Options**
- Confirm Hosts
- Choose Services
- Assign Masters
- Assign Slaves and Clients
- Customize Services
- Review
- Install, Start and Test
- Summary

Install Options

Enter the list of hosts to be included in the cluster and provide your SSH key.

Target Hosts

Enter a list of hosts using the Fully Qualified Domain Name (FQDN), one per line. Or use [Pattern Expressions](#)

```
t00bsb-hd01
t00bsb-hd02
t00bsb-hd03
```

Host Registration Information

☒ Provide your [SSH Private Key](#) to automatically register hosts

No file selected.

```
L3ZQK8bLNRbyNE15LsXdaN7NRBhnmJYkJzg0YfnZrwqy+PBRy+xtmoQW7pzN
39tV
ry1dc0maEt0150LPK7R0xdYnUCDg6fGn6teqLf1s6tgvpkZ8mb5v
-----END RSA PRIVATE KEY-----
```

SSH User Account

SSH Port Number

☐ Perform [manual registration](#) on hosts and do not use SSH

Рисунок 4.

4. Вписать hostname всех серверов кластера hdfs в область «Target Hosts»
5. Получить приватный ключ с сервера управления Ambari (cat ~/.ssh/id_rsa) и вставить в область «SSH Private Key». Нажать кнопку «Register and Confirm»
6. Убедиться в том, что hostname присутствуют в /etc/hosts. Нажать кнопку «Ok».

Подтвердить согласием сообщение вида «The following hostnames are not valid FQDNs»
Начнётся установка и регистрация нод в ambari-server. Дождаться статуса «Success»

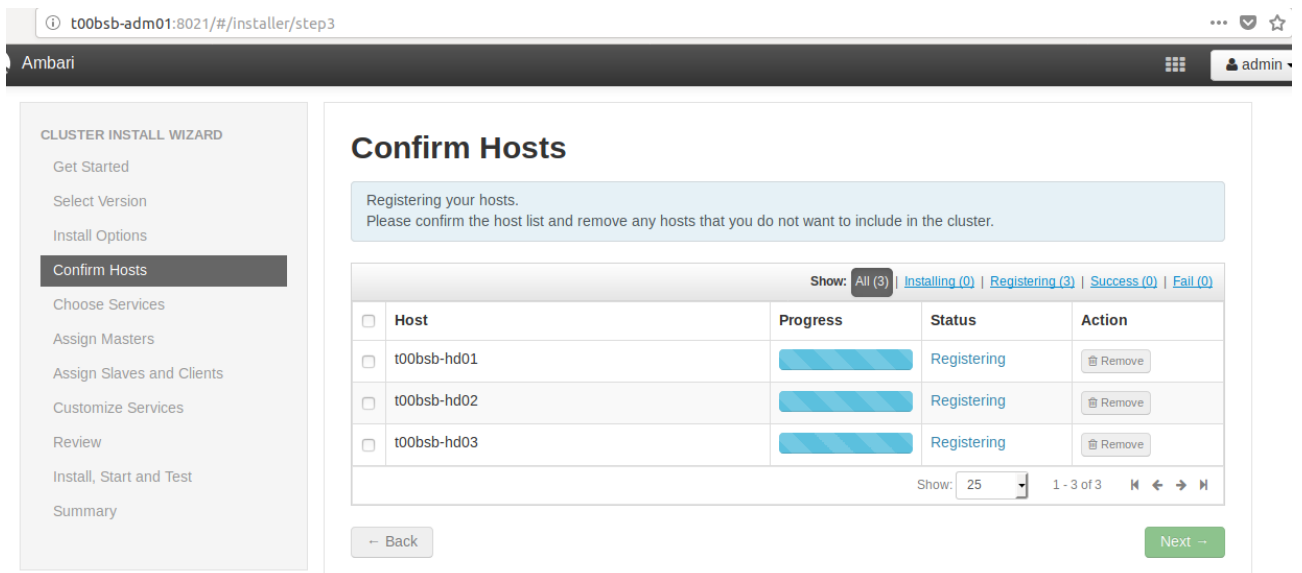


Рисунок 5.

7. Подтвердить добавление хостов и нажать кнопку «Next»

ВНИМАНИЕ

В случае наличия ошибок при добавлении хоста в кластер HDP необходимо:

- удостовериться, что включен Ambari Agent
- удостовериться, что включен NTPD
- отключены TNP и FireWall

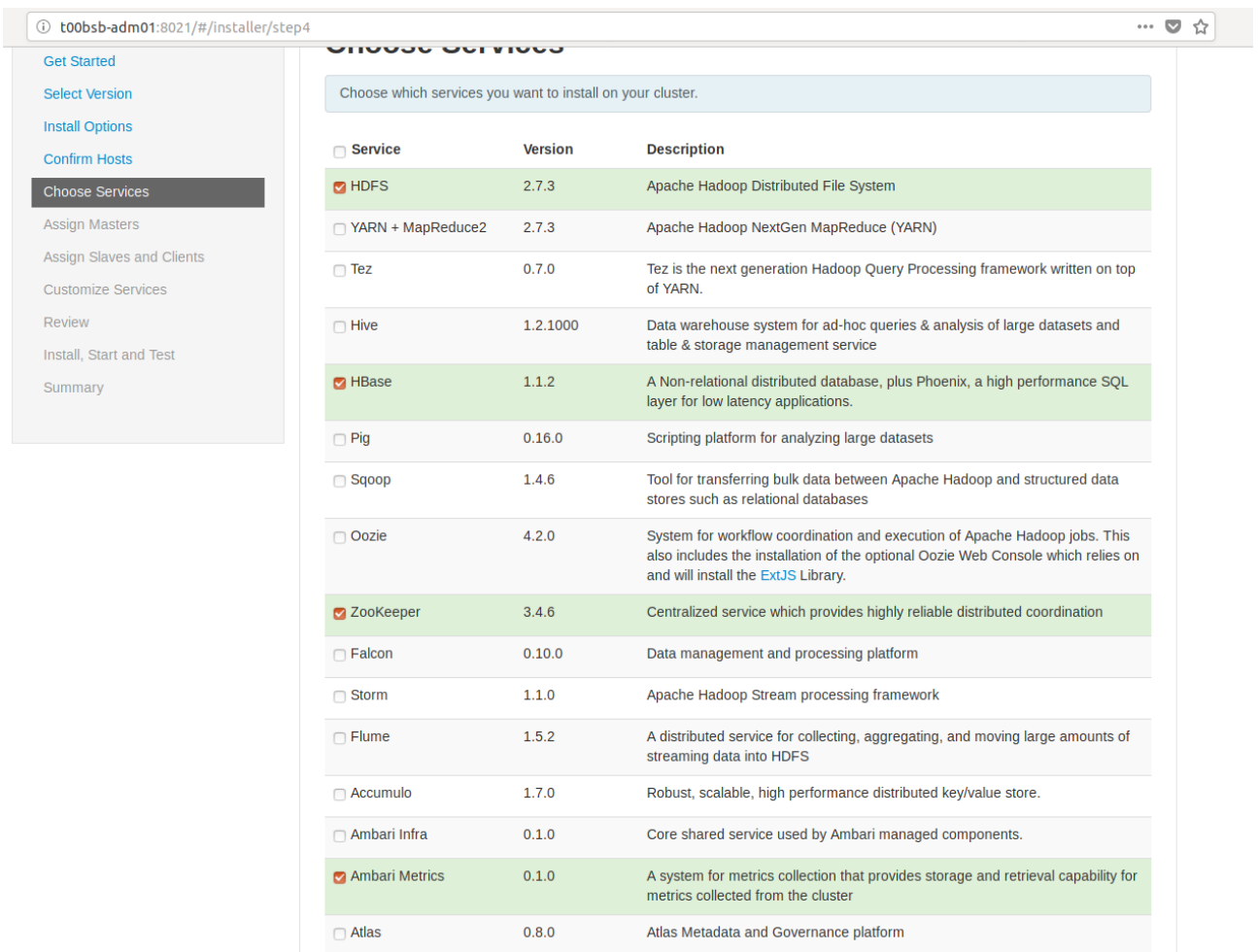


Рисунок 6.

8. Выбрать сервисы HDFS, Hbase, Zookeeper, Ambari Metrics. Нажать кнопку «Next»

Ambari

ad

CLUSTER INSTALL WIZARD

[Get Started](#)
[Select Version](#)
[Install Options](#)
[Confirm Hosts](#)
[Choose Services](#)
[Assign Masters](#)
[Assign Slaves and Clients](#)
[Customize Services](#)
[Review](#)
[Install, Start and Test](#)
[Summary](#)

Assign Masters

Assign master components to hosts you want to run them on.

NameNode: t00bsb-hd01 (3.5 GB, 2 cores)
SNameNode: t00bsb-hd02 (3.5 GB, 2 cores)
HBase Master: t00bsb-hd01 (3.5 GB, 2 cores)
ZooKeeper Server: t00bsb-hd01 (3.5 GB, 2 cores)
ZooKeeper Server: t00bsb-hd03 (3.5 GB, 2 cores)
ZooKeeper Server: t00bsb-hd02 (3.5 GB, 2 cores)
Grafana: t00bsb-hd01 (3.5 GB, 2 cores)
Metrics Collector: t00bsb-hd01 (3.5 GB, 2 cores)
Activity Analyzer: t00bsb-hd01 (3.5 GB, 2 cores)
Activity Explorer: t00bsb-hd01 (3.5 GB, 2 cores)
HST Server: t00bsb-hd01 (3.5 GB, 2 cores)

t00bsb-hd01 (3.5 GB, 2 cores)
NameNode
HBase Master
ZooKeeper Server
Grafana
Metrics Collector
Activity Analyzer
Activity Explorer
HST Server

t00bsb-hd02 (3.5 GB, 2 cores)
SNameNode
ZooKeeper Server

t00bsb-hd03 (3.5 GB, 2 cores)
ZooKeeper Server

Back
Next

Рисунок 7.

9. На странице «Assign Masters» убедиться в том, что Zookeeper Server отмечен на 3х разных серверах:

- NameNode должна быть расположена на ноде hd01
- SNameNode должна быть расположена на ноде hd02
- Для Metrics Colector выбирать ноду adm01

Нажать кнопку «Next»

Ambari

admin

CLUSTER INSTALL WIZARD

[Get Started](#)
[Select Version](#)
[Install Options](#)
[Confirm Hosts](#)
[Choose Services](#)
[Assign Masters](#)
[Assign Slaves and Clients](#)
[Customize Services](#)
[Review](#)
[Install, Start and Test](#)
[Summary](#)

Assign Slaves and Clients

Assign slave and client components to hosts you want to run them on.
Hosts that are assigned master components are shown with *.
"Client" will install HDFS Client, HBase Client and ZooKeeper Client.

Host	all none	all none	all none	all none	all none
t00bsb-hd01*	☐ DataNode	☐ NFSGateway	☐ RegionServer	☐ Phoenix Query Server	☐ Client
t00bsb-hd02*	☐ DataNode	☐ NFSGateway	☐ RegionServer	☐ Phoenix Query Server	☐ Client
t00bsb-hd03*	☒ DataNode	☐ NFSGateway	☒ RegionServer	☐ Phoenix Query Server	☒ Client

Show: 25
1 - 3 of 3

Back
Next

Рисунок 8.

Страница 16 из 46

10. На странице «Assign Slave and Clients» убедиться в том, что отмечены флажки для ноды hd03: DataNode, RegionServer, Client. Нажать кнопку «Next»

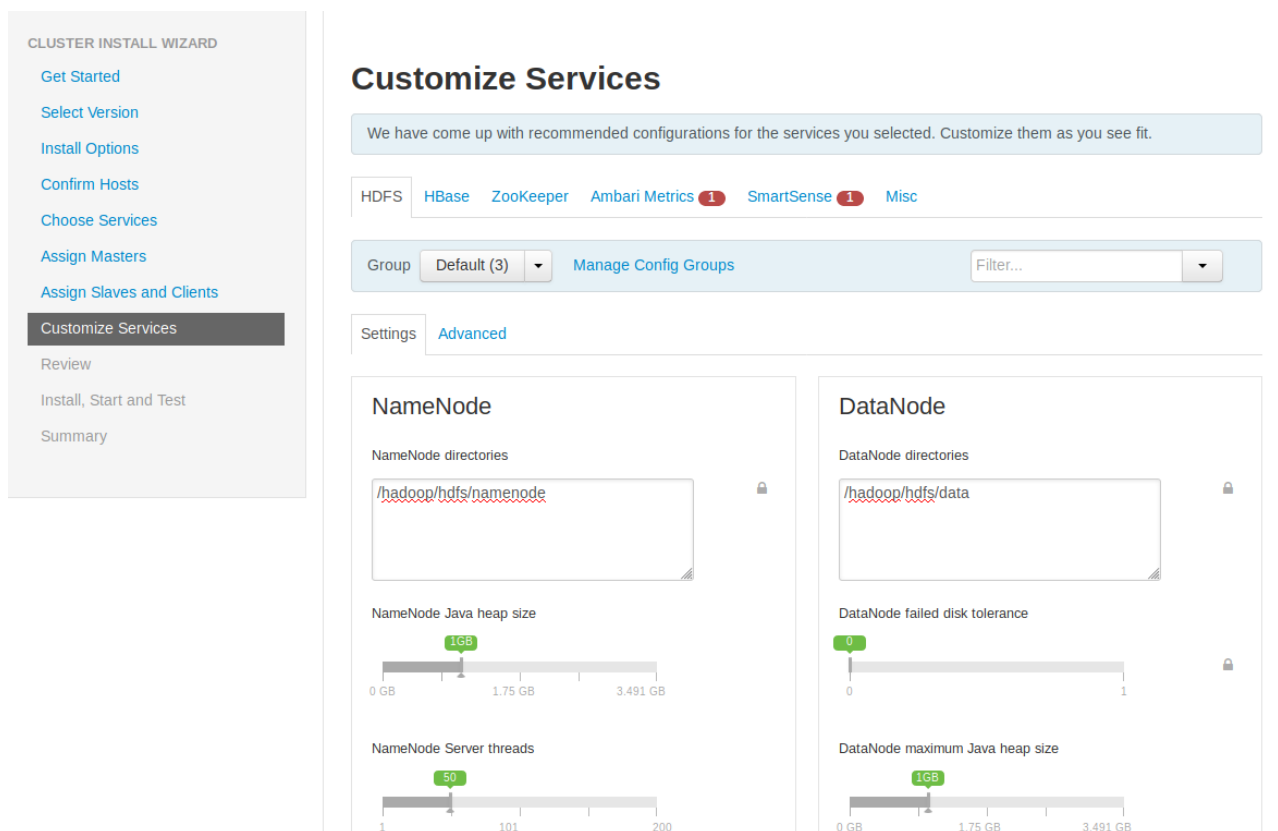


Рисунок 9.

11. Во вкладке HDFS убедиться, что:

- NameNode имеет путь: /hadoop/hdfs/namenode
- DataNode имеет путь: /hadoop/hdfs/data

12. На вкладке Ambari Metrics задать пароли для Ambari Metrics (вкладка General)

13. На странице «Ambari Metrics» вкладки «General» задать пути в

- Metrics Collector log dir (/ambari/ambari-metrics-collector)
- Metrics Monitor log dir (/ambari/ambari-metrics-monitor)

14. Развернуть «Advanced ams-hbase-site» и переопределить

```
hbase.rootdir= (file:///ambari/ambari-metrics-collector/hbase)
hbase.tmp.dir=/ambari/ambari-metrics-collector/hbase-tmp
Aggregator checkpoint directory (/ambari/ambari-metrics-collector/checkpoint)
```

15. Нажать There is 1 configuration change in 1 service [Show Details](#).

Подтвердить, кликнув кнопку «OK»

- На вкладке SmartSense (Basic) переопределить директорию /var/lib/smartsense > /ambari в трех полях согласно Рисунку 10

Group: Default (1) Manage Config Groups Filter...

Basic Data Capture Operations Gateway Activity Analysis Advanced

SmartSense Account

Customer account name
unspecified

SmartSense ID
unspecified

Notification Email
unspecified

Enable Flex Subscription
☐ No

Local Storage

Bundle storage directory
/ambari/hst-server/data

Server temporary data directory
/ambari/hst-server/tmp

Agent temporary data directory
/ambari/hst-agent/data/tmp

Рисунок 10.

16. Во вкладке Activity Analysis задать пароль для admin

Настройки можно оставить по умолчанию.

Конфигурация готова, можно запустить установку. Нажать кнопку «Next».

Подтвердить, проигнорировав предупреждения, нажав кнопку «Proceed Anyway».

Убедиться, что Review соответствует установкам, запустим инсталляцию «Deploy».

17. После завершения всех действий выполнить на серверах

```
python /usr/lib/ambari-agent/lib/ambari_agent/HostCleanup.py --silent --skip=users
```

ВНИМАНИЕ

Вероятный конфликт библиотек libsnappy1-1.1.3-1.2.x86_64. (DataNode Install)

Для решения выполнить на проблемных нодах

```
zypper remove snappy snappy-devel; zypper install snappy snappy-devel
```

Повторить установку с момента остановки. Нажать кнопку «Retry».

Нажать кнопку «Complete».

Summary

Here is the summary of the install process.

```
The cluster consists of 1 hosts
  Installed and started services successfully on 1 new host
Master services installed
  SNameNode installed on t00bsb-hd01
  NameNode installed on t00bsb-hd01
  HBase Master installed on t00bsb-hd01
All services started
All tests passed
Install and start completed in 10 minutes and 27 seconds
```

Complete →

Рисунок 11.

Результат: Установка завершена.

После установки перейти на главную страницу. На ней слева выбрать SmartSense, нажать кнопку «Service Actions» и выбрать «Stop», а затем – «Delete Service».

Осталось подготовить схему в HBASE. Для этого необходимо выполнить следующий скрипт на сервере HBase (d00bsb-hd01)

```
#!/bin/sh
## table names
declare -a table_names=("bioData;['bioFile']")

function tableExists {
    echo "Table <$1> already exists."
}

function createTable {
    echo "Table <$1> does not exist."
    echo "Creating the Table <$1>"

    echo "create '$1', '$2' | hbase shell > log 2>&1
    cat log | grep -q '0 row(s) in'

    if [ $? = 0 ]; then
        echo "Table <$1> created successfully."
    else
        echo "Table <$1> is not created."
    fi
}

## iterating table names and checking table already exists. if does not exist creating table.
for i in "${table_names[@]}"
do
    ## splitting the table based on ';'. First argument will be 'table_name' and second argument will be
    'column_families'
```

```

set -- "$@"
IFS=","; declare -a table_name_info=(*)

echo ""
echo "Checking the table <${table_name_info[0]}> exists or not"
echo ""

echo "exists '${table_name_info[0]}'" | hbase shell > log 2>&1
cat log | grep -q 'Table '${table_name_info[0]}' does exist'

if [ $? = 0 ]; then
    tableExists ${table_name_info[0]}
else
    createTable ${table_name_info[0]} ${table_name_info[1]}
fi
done

## deleting the log file.
rm -rf log
echo ""
echo "done."
exit

```

Дождаться вывода сообщения

```

Checking the table <bioData> exists or not

Table <bioData> does not exist.
Creating the Table <bioData>

Table <bioData> created successfully.

done.

```

3. УСТАНОВКА И НАСТРОЙКА POSTGRESQL 10

3.1. ПОДГОТОВКА СЕРВЕРОВ PGSQL

3.1.1. Разметка диска по стандарту

Необходимо выполнить разметку, форматирование диска и подключение к системе в следующем порядке:

1. Создать размеры и выполнить форматирование

```

pvcreate /dev/sdb ;
vgcreate vg_postgres /dev/sdb ;
lvcreate --name lv_u01 --size 5G vg_postgres ;
lvcreate --name lv_main -l 100%FREE vg_postgres ;
mkfs.xfs /dev/mapper/vg_postgres-lv_u01 ;
mkfs.xfs /dev/mapper/vg_postgres-lv_main ;

```

2. После создания файловой системы в конце прописываем постоянные записи в /etc/fstab

/dev/vg_postgres/lv_u01	/u01	xfs	defaults	1 1
/dev/vg_postgres/lv_main	/u01/postgres/main	xfs	defaults	1 1

3. Создать точки монтирования. Монтирование необходимо осуществлять последовательно

```
mkdir /u01 && mount /u01; mkdir /u01/postgres/main -p && mount -a
```

4. Проверить, что доступное место корректно определилось в разметке командой

```
df -h
...
/dev/mapper/vg_postgres-lv_u01 5,0G 33M 5,0G 1% /u01
/dev/mapper/vg_postgres-lv_main 59G 33M 59G 1% /u01/postgres/main
```

3.1.2. Установка пакетов и настройка базы данных

1. Перед установкой необходимо подключить репозитории пакетов

```
zypper ar -f https://download.postgresql.org/pub/repos/zypp/repo/pgdg-sles-10.repo
```

2. Для обеспечения всех зависимостей необходимо включить в yast2

“Suse Linux Enterprise Software Development Kit 12 SP3 x86_64, SUSE Linux Enterprise High Availability Extension 12 SP3 x86_64, SUSE Package Hub 12 SP3 x86_64

3. Выполнить

```
SUSEConnect -p sle-sdk/12.3/x86_64 ;
SUSEConnect -p PackageHub/12.3/x86_64 ;
```

4. Установить платные расширения

```
SUSEConnect -p sle-ha/12.3/x86_64 -r <регистрационный код> ;
```

3.2. УСТАНОВКА POSTGRESQL 10

Выполнить в консоли команду

```
zypper install postgresql10 postgresql10-contrib postgresql10-server postgresql10-devel
```

1. Перенести данные пользователя postgres в выделенную директорию

```
usermod -d /u01/postgres postgres
cp -f /etc/skel/.bash* /u01/postgres/
chown -R postgres: /u01/postgres
```

2. Для указания выделенного раздела при запуске для данных PGSQL следует создать файл /etc/systemd/system/postgresql-10.service.d/override.conf с содержанием

```
[Service]
Environment=PGDATA=/u01/postgres/main/pgdata
ExecStart=
ExecStart=/usr/pgsql-10/bin/postmaster -D ${PGDATA}
```

Для этого выполнить

```
mkdir /etc/systemd/system/postgresql-10.service.d;
cat <<EOF >/etc/systemd/system/postgresql-10.service.d/override.conf
[Service]
Environment=PGDATA=/u01/postgres/main/pgdata
ExecStart=
ExecStart=/usr/pgsql-10/bin/postmaster -D ${PGDATA}

EOF
;
```

3. Перечитать конфигурацию systemd

```
systemctl daemon-reload
```

4. Переключиться в пользователя postgres

```
su - postgres;
```

ВНИМАНИЕ

Перед созданием БД необходимо привести ~/.bash_profile в такой вид

```
[ -f /etc/profile ] && source /etc/profile
PGDATA=/u01/postgres/main/pgdata
export PGDATA
export PATH=$PATH:/usr/pgsql-10/bin
# If you want to customize your settings,
# Use the file below. This is not overridden
# by the RPMS.
[ -f /var/lib/pgsql/.pgsql_profile ] && source /var/lib/pgsql/.pgsql_profile
```

5. Создать пустую базу данных

Перечитать окружение source ~/.bash_profile

Проверить echo \$PGDATA

```
/u01/postgres/main/pgdata
```

Выполнить

```
initdb --locale=ru_RU.utf8 --data-checksums
```

3.3. КОНФИГУРАЦИЯ БАЗЫ ДАННЫХ

1. В первую очередь необходимо прописать все сетевые разрешения для подключения в файле /u01/postgres/main/pgdata/pg_hba.conf
2. Подсеть 192.168.3.0/24 необходимо заменить в соответствии с реальной
3. Строка разрешения подключения для пользователя приложения BSB и приложения luna

#	TYPE	DATABASE	USER	ADDRESS	METHOD
# IPv4 local connections:					
host		faceis_db	faceis	192.168.3.0/24	md5
host		faceis_ui	faceis	192.168.3.0/24	md5
host		admin_faceis_db	faceis	192.168.3.0/24	md5
host		bsb	bsb	192.168.3.0/24	md5

4. Следующие строки должны находиться в конце файла

# Replication:					
# IPv4 local connections:					
host	all	postgres		192.168.3.0/24	md5
# IPv4 local connections:					
host	postgres	db_backup		192.168.3.0/24	md5
# IPv4 local connections:					
host	replication	db_backup		192.168.3.0/24	trust
# IPv4 local connections:					
host	replication	db_replication		192.168.3.0/24	trust
# IPv4 local connections:					
host	replication	db_replication		192.168.3.0/24	trust
# IPv4 local connections:					
host	replication	db_replication		192.168.3.0/24	trust

5. Файл конфигурации /u01/postgres/main/pgdata/postgresql.conf следует привести к следующему виду

# Параметры PostgreSQL 9.6 (Соответствуют стандарту v1.5+)	
# Распределение памяти для 8GB	
# FILE LOCATIONS	
data_directory = '/u01/postgres/main/pgdata'	
# hba_file = '{{postgresql_hba_file}}'	
# ident_file = '{{postgresql_ident_file}}'	
# CONNECTIONS AND AUTHENTICATION	
# - Connection Settings -	
listen_addresses = '*'	
port = 5432	

```
max_connections = 500

# - Security and Authentication -

# - TCP Keepalives -

# RESOURCE USAGE (except WAL)

# - Memory -

shared_buffers = 2GB
work_mem = 16MB
maintenance_work_mem = 512MB
autovacuum_work_mem = 256MB

# - Kernel Resource Usage -

shared_preload_libraries = 'pg_stat_statements'

# WRITE AHEAD LOG

# - Settings -

wal_level = replica
fsync = on
synchronous_commit = on

# - Checkpoints -

checkpoint_timeout = 30min
max_wal_size = 4GB
min_wal_size = 1GB
checkpoint_completion_target = 0.9
checkpoint_warning = 60s

# - Archiving -

archive_mode = on
archive_command = '/bin/true'
archive_timeout = 600

# REPLICATION

# - Sending Server(s) -

max_wal_senders = 10
wal_keep_segments = 1000

# - Master Server -

# - Standby Servers -
```

```
hot_standby = on

# QUERY TUNING

# - Planner Method Configuration -

# - Planner Cost Constants -

effective_cache_size = 3GB

# ERROR REPORTING AND LOGGING

# - Where to Log -

log_destination = 'stderr'
log_directory = 'pg_log'
logging_collector = on
log_directory = '/var/log/postgres'
log_filename = 'postgresql-%Y-%m-%d_%H%M%S.log'
log_file_mode = '0600'
log_truncate_on_rotation = off
log_rotation_age = 1d
log_rotation_size = 0

# - When to Log -

log_min_duration_statement = '250'

# - What to Log -

log_checkpoints = on
log_connections = on
log_disconnections = on

log_line_prefix = '%t [%p] %q%h %a %u@%d '

log_lock_waits = on
log_statement = 'ddl'
log_replication_commands = on
log_temp_files = 0

# RUNTIME STATISTICS

# - Query/Index Statistics Collector -

track_io_timing = on

# AUTOVACUUM PARAMETERS

autovacuum = on
log_autovacuum_min_duration = 60000
```

```
autovacuum_max_workers = 3
autovacuum_vacuum_scale_factor = 0.01
autovacuum_analyze_scale_factor = 0.05
```

```
# CLIENT CONNECTION DEFAULTS
```

```
# - Locale and Formatting -
```

```
lc_messages = 'en_US.UTF-8'
```

```
# CUSTOMIZED OPTIONS
```

```
# Add settings for extensions here
```

```
# - pg_stat_statements -
```

```
pg_stat_statements.track = 'top'
```

```
pg_stat_statements.max = '10000'
```

```
pg_stat_statements.track_utility = 'off'
```

3.4. ПЕРЕЗАПУСК СЕРВИСА ДЛЯ ПРИМЕНЕНИЯ КОНФИГУРАЦИЙ

Перезагрузка сервиса для применения конфигураций осуществляется командой

```
systemctl restart postgresql-10;
```

либо, выполнив в консоли под пользователем postgres, команду

```
psql -c "SELECT pg_reload_conf();"
```

3.5. СКРИПТ ДЛЯ СОЗДАНИЯ СИСТЕМНЫХ И ПРИКЛАДНЫХ ПОЛЬЗОВАТЕЛЕЙ

Подключаемся к БД локально

Выполнить в консоли команду

```
psql
```

```
BEGIN;
```

```
-- Создание стандартных ролей
```

```
-- Роль для подключения к БД персонализированных пользователей
```

```
CREATE ROLE user_connect;
```

```
COMMENT ON ROLE user_connect
```

```
IS 'Разрешение на подключение к БД с адресов 0.0.0.0/0';
```

```
-- Роль для выдачи разрешений системам мониторинга
```

```
CREATE ROLE monitoring_role;
```

```
COMMENT ON ROLE monitoring_role
```

```
IS 'Разрешения для систем мониторинга';
```

```
-- Пользователь для резервного копирования
CREATE ROLE db_backup LOGIN
    ENCRYPTED PASSWORD 'md5c940ad9213634f663c1dc725367f5150' REPLICATION;
COMMENT ON ROLE db_backup IS 'Резервное копирование';

-- Пользователь для репликации
CREATE ROLE db_replication LOGIN
    ENCRYPTED PASSWORD 'md5f7cdb91bd94492806374fbd6c539d9d7' REPLICATION;
COMMENT ON ROLE db_replication IS 'Репликация';

-- Grant on database

REVOKE ALL ON DATABASE postgres FROM public;
GRANT CONNECT,TEMPORARY ON DATABASE postgres TO monitoring_role;
GRANT CONNECT,TEMPORARY ON DATABASE postgres TO db_backup;

-- Создание прикладных ролей
-- Пользователь Luna
create role faceis LOGIN PASSWORD 'faceis';
COMMENT ON ROLE faceis IS 'Пользователь Luna';

-- Пользователь bsb
create role bsb LOGIN PASSWORD 'bsb';
COMMENT ON ROLE bsb IS 'Пользователь BSB';

COMMIT;
```

Пароли faceis, bsb следует изменить на реальные

3.5.1. Создание базы данных bsb, faceis_db

Войти в БД под системным пользователем

```
psql -c "CREATE DATABASE bsb OWNER bsb; "
psql -c "CREATE DATABASE faceis_db OWNER faceis; "
psql -c "CREATE DATABASE admin_faceis_db OWNER faceis; "
psql -c "CREATE DATABASE faceis_ui OWNER faceis; "
```

4. УСТАНОВКА И НАСТРОЙКА RABBITMQ

Сервера очередей для обработки очередей заданий компонентов Fe и Be, так же обеспечивают взаимодействие luna.

4.1. ПОДГОТОВКА СЕРВЕРОВ

1. Предварительно необходимо подключить репозиторий erlang

```
zypper ar -f
http://download.opensuse.org/repositories/devel:/languages:/erlang/SLE_12/devel:languages:erlang.repo
```

2. Установить зависимости

```
zypper install erlang-20.3.8.6-12.1.x86_64
```

4.2. РАЗМЕТКА ДИСКА

```
pvccreate /dev/sdb;
vgcreate vg_data /dev/sdb;
```

```
lvcreate --name lv_log --size 10G vg_data;  
lvcreate --name lv_rabbit -l100%FREE vg_data;  
  
mkfs.xfs /dev/vg_data/lv_log;  
mkfs.xfs /dev/vg_data/lv_rabbitmq;
```

1. Создать соответствующие директории командами

```
mkdir /var/lib/rabbitmq  
mkdir /var/log/rabbitmq
```

2. Прописать записи для постоянного монтирования, добавив в конец файла /etc/fstab строки

```
/dev/vg_data/lv_rabbitmq /var/lib/rabbitmq xfs defaults 1 1  
/dev/vg_data/lv_log /var/log/rabbitmq xfs defaults 1 1
```

3. Смонтировать разделы командой

```
mount -a
```

4.3. УСТАНОВКА RABBITMQ

Установить ПО rabbitmq, выполнив в консоли

```
zypper install https://dl.bintray.com/rabbitmq/all/rabbitmq-server/3.7.8/rabbitmq-server-3.7.8-1.suse.noarch.rpm
```

1. Скачать плагин и распаковать в папку rabbitmq командами

```
wget https://dl.bintray.com/rabbitmq/community-plugins/3.7.x/rabbitmq_delayed_message_exchange/rabbitmq_delayed_message_exchange-20171201-3.7.x.zip  
  
unzip rabbitmq_delayed_message_exchange-20171201-3.7.x.zip -d /usr/lib/rabbitmq/lib/rabbitmq_server-3.7.8/plugins
```

2. Подключить плагины в брокере, запустить и включить автозагрузку

```
rabbitmq-plugins enable rabbitmq_management ;  
rabbitmq-plugins enable rabbitmq_delayed_message_exchange ;  
systemctl restart rabbitmq-server ;  
systemctl enable rabbitmq-server ;
```

4.4. ПОДКЛЮЧЕНИЕ LUNA

1. Получить расширенные права и доступ к утилите управления

```
find /var/lib/rabbitmq/ -name 'rabbitmqadmin' -exec chmod 0777 {} \;  
PATH="$PATH:/var/lib/rabbitmq/mnesia/rabbit@`hostname`-s`-plugins-expand/rabbitmq_management-`rpm -qa | grep rabbitmq | cut -d - -f 3`/priv/www/cli/";
```

2. Создавать пользователя luna и назначить ему расширенные права

```
rabbitmqctl add_user luna luna;  
rabbitmqctl set_user_tags luna administrator;  
rabbitmqctl set_permissions -p / luna ".*" ".*" ".*";
```

3. Создать в конфигурации брокера необходимые очереди для экстракции и матчинга, а также службы обмена для коммуникации между компонентами

```
rabbitmqadmin -u luna -p luna declare exchange name=luna.extract type=direct;  
rabbitmqadmin -u luna -p luna declare exchange name=luna.match type=direct;  
rabbitmqadmin -u luna -p luna declare queue name=extractor durable=false auto_delete=true;  
rabbitmqadmin -u luna -p luna declare queue name=matcher durable=false auto_delete=true;
```

4. Создадим пользователя bsb и назначим ему расширенные права

```
rabbitmqctl add_user bsb bsb;  
rabbitmqctl set_user_tags bsb administrator;  
rabbitmqctl set_permissions -p / bsb ".*" ".*" ".*";
```

5. УСТАНОВКА И НАСТРОЙКА LUNA STACK

Для полного функционирования данного решения необходима настроенная БД PGSQL, и сервера очередей RabbitMQ.

Исправленный дистрибутив luna_v.2.7.2.zip доступен по ссылке <https://portal.rtlabs.ru/~OfJ7k>. Пароль передается по запросу.

5.1. ПОДГОТОВКА СЕРВЕРОВ

1. Установить репозиторий epel

```
yum install epel-release
```

2. Далее необходимо установить зависимости

```
yum -y --nogpgcheck install boost-system boost-chrono jemalloc libexif qpidd-cpp-client boost-context boost-regex boost-thread double-conversion-devel glog libevent pyOpenSSL gcc ImageMagick-devel python36 python36-devel python36-setuptools python-virtualenv
```

5.2. РАЗМЕТКА ДИСКА

1. Выполнить команду для создания дискового раздела

```
pvccreate /dev/sdb;  
vgcreate vg_data;  
lv_create --name lv_luna -l100%FREE vg_data;  
mkfs.xfs /dev/vg_data/lv_luna;  
mkdir -p /var/lib/luna;
```

2. После создания файловой системы в конце прописать постоянные записи в /etc/fstab и монтировать

```
/dev/vg_data/lv_luna /var/lib/luna xfs defaults 0 0
```

3. Смонтировать

```
mount -a;
```

5.3. УСТАНОВКА LUNA

1. Перенести дистрибутив luna_v.2.7.2.zip на сервер d00bsb-vl01 и распаковать в каталоге /var/lib/luna

```
cd /var/lib/luna ;  
unzip luna_v.2.7.2.zip -d /var/lib/luna/
```

2. Создать символическую ссылку

```
ln -s /var/lib/luna/luna_v.2.7.2 /var/lib/luna/current
```

3. Скопировать systemd сервисы Luna

```
cp /var/lib/luna/current/extras/systemd/luna-*.service /etc/systemd/system/ ;  
systemctl daemon-reload ;
```

4. Установить ПО проверки лицензии:

```
yum -y install /var/lib/luna/current/extras/rpms/haspd-*.rpm /var/lib/luna/current/extras/rpms/tbb*.rpm  
/var/lib/luna/current/extras/rpms/aerospike*/*.rpm ;
```

5. Добавим сервисы systemd в автозагрузку

```
systemctl daemon-reload  
systemctl start aksusbd  
systemctl enable aksusbd
```

5.4. УСТАНОВКА ЛИЦЕНЗИИ

1. Файл лицензии Unlocked_*.v2c скопировать на сервер в директорию /root и выполнить команду

```
find /root/ -type f -name "Unlocked_*.v2c" -exec hasp_update u {} \;
```

5.5. КОНФИГУРИРОВАНИЕ AEROSPIKE DB

1. Скопируем подготовленный конфигурационный файл и подтверждаем замену

```
cp /var/lib/luna/current/extras/aerospike.conf /etc/aerospike/
```

2. Конфигурационный файл следует привести к такому виду, с учетом гибкого горизонтального масштабирования



aerospike.conf

3. Параметры в конфигурационном файле необходимо изменить в соответствии
 - address _hostname_ (Необходимо указать fqdn реального сервера на котором запускается aerospike
 - replication-factor 2 (Должен равняться количеству нод в кластере)
 - file /var/lib/luna/aerospike/data/luna.dat (Рекомендуется расположить на выделенном разделе диска)
4. Создать директорию для данных Aerospike командой

```
mkdir -p /var/lib/luna/aerospike/data
```

5. Создать директорию для логов

```
mkdir -p /var/log/aerospike
```

6. Запустить Aerospike DB, выполнив команды

```
systemctl restart aerospike ;  
systemctl enable aerospike ;
```

5.6. НАСТРОЙКА LUNA

5.6.1. LUNA-API

1. Установить pip, wheel, setuptools

```
python $(find /var/lib/luna/current/extras/requirements/common -name '*pip*')/pip install  
/var/lib/luna/current/extras/requirements/common/*
```

2. Создать виртуальное окружение

```
virtualenv /var/lib/luna/current/venv -p /usr/bin/python36
```

3. Установить зависимости для luna-api

```
source /var/lib/luna/current/venv/bin/activate ;  
cd /var/lib/luna/current/luna-api ;  
pip install -r requirements.txt /var/lib/luna/current/extras/requirements/luna-api/*
```

4. Перед заполнением структуры БД необходимо заменить конфигурационный файл

/var/lib/luna/current/luna-api/configs/config.conf
файлом /var/lib/luna/current/luna-api/configs/config.conf.short

Для этого выполнить команду

```
cat /var/lib/luna/current/luna-api/configs/config.conf.short > /var/lib/luna/current/luna-api/configs/config.conf
```

- DB_HOST > Указать реальный fqdn сервера БД postgres
- DB_NAME, PASSWORD_DB, USER_NAME (использовать указанные при создании БД faceis_db)
- SEND_TO_S3 > Выставить 0
- Удалить значения S3_BUCKET, S3_AWS_PUBLIC_ACCESS_KEY, S3_AWS_SECRET_ACCESS_KEY, S3_HOST
- SEND_ACCOUNT_STATS > Выставить 0

5. В конфигурационных файлах брокера

```
/var/lib/luna/current/conf/rabbitmq-matcher.conf
```

```
/var/lib/luna/current/conf/rabbitmq-extractor.conf
```

- В значении address указать реальный fqdn сервера rabbitmq

6. В файле /var/lib/luna/current/conf/extractor.conf удалить блок

```
"imageDump" : {  
  "skipErrors" : false,  
  "createBucket" : true,  
  "encoder" : "jpg",  
  "bucket" : "baguette",  
  "address" : "127.0.0.1",  
  "port" : 5020,  
  "protocol" : "http"  
},
```

7. Запустить создание БД

```
python bases_scripts/db_create.py
```

8. Заменить строку запуска в скрипте systemd

```
/etc/systemd/system/luna-api.service
```

```
sed -ie 's|^ExecStart.*$|ExecStart=/var/lib/luna/current/venv/bin/python  
/var/lib/luna/current/luna-api/run.py --workers=2 --port=5000|g' /etc/systemd/system/luna-  
api.service
```

9. Перечитать systemd и запустить CORE сервисы LUNA

```
systemctl start luna-broker ;  
systemctl enable luna-broker ;  
systemctl start luna-extractor ;  
systemctl enable luna-extractor ;  
systemctl start luna-matcher ;  
systemctl enable luna-matcher ;
```

```
systemctl start luna-api ;  
systemctl enable luna-api ;
```

10. Убедимся, что порты доступны командой

```
netstat -tulnap | egrep '5000'
```

```
tcp      0      0 0.0.0.0:5000        0.0.0.0:*          LISTEN    7990/python36  
tcp6     0      0 :::5000             :::*                LISTEN    7990/python36  
udp      0      0 0.0.0.0:50030       0.0.0.0:*          7656/hasplmd
```

5.6.2. LUNA-ADMIN

1. Установить зависимости для luna-admin

```
source /var/lib/luna/current/venv/bin/activate;  
cd /var/lib/luna/current/luna-admin ;  
pip install -r requirements.txt /var/lib/luna/current/extras/requirements/luna-admin/*
```

2. В файле /var/lib/luna/current/luna-admin/configs/config.conf

- LUNA_API_DB_HOST = Вписать реальный fqdn сервера postgres
- ADMIN_DB_HOST = Вписать реальный fqdn сервера postgres

3. Для заполнения БД выполнить команду

```
python bases_scripts/db_create.py
```

4. Исправить строку вызова в system скрипте
/etc/systemd/system/luna-admin.service

```
sed -i 's|^ExecStart.*$|ExecStart=/var/lib/luna/current/venv/bin/python /var/lib/luna/current/luna-admin/run.py --port=5010 --config=./configs/config.conf|g' /etc/systemd/system/luna-admin.service
```

5. Перечитать systemd и запустить LUNA-ADMIN командами

```
systemctl daemon-reload ;  
systemctl start luna-admin ;  
systemctl enable luna-admin ;
```

6. Убедиться, что порт 5010 доступен

```
(venv) [root@d00bsb-vl01 luna-admin]# netstat -tulnap | grep 5010  
tcp      0      0 0.0.0.0:5010        0.0.0.0:*          LISTEN    18103/python36  
tcp6     0      0 :::5010             :::*                LISTEN    18103/python36
```

5.6.3. LUNA-UI

Перед установкой необходимо завершить установку LUNA-CORE и LUNA-ADMIN

1. Установим зависимости для luna-admin

```
source /var/lib/luna/current/venv/bin/activate;  
cd /var/lib/luna/current/luna-ui ;
```

```
pip install -r requirements.txt /var/lib/luna/current/extras/requirements/luna-ui/*
```

2. В конфигурационном файле /var/lib/luna/current/luna-ui/config.py изменить

- DB_HOST = "d00bsb-db01"
- END_POINT_TO_LUNA_PYTHON_SERVER = "IP адрес доступный клиентской части"
- LUNA_PYTHON_SERVER = "IP адрес внутри сети" (d00bsb-vl01)

3. Запустим создание БД

```
python db_create.py
```

4. Исправить строку вызова python в system скрипте
/etc/systemd/system/luna-ui.service

```
sed -i 's|^ExecStart.*$|ExecStart=/var/lib/luna/current/venv/bin/python /var/lib/luna/current/luna-ui/runws.py 5005|g' /etc/systemd/system/luna-ui.service
```

5. Перечитать system и запустить LUNA-UI

```
systemctl daemon-reload ;  
systemctl start luna-ui ;  
systemctl enable luna-ui ;
```

6. Убедимся, что порты доступны

```
(venv) [root@d00bsb-vl01 luna-ui]# netstat -tulnap | grep 5005  
tcp      0      0 0.0.0.0:5005          0.0.0.0:*             LISTEN    18445/python36
```

5.7. РЕГИСТРАЦИИ В СИСТЕМЕ LUNA-UI И ПОЛУЧЕНИЕ ТОКЕНА ВЫПОЛНЯЕТСЯ ДВУМЯ СПОСОБАМИ:

1. Открыть интерфейс LUNA-UI, перейдя по ссылке: <http://d00bsb-vl01:5005>



Рисунок 12.

2. Заполнить поля для регистрации, нажать кнопку «Create account» для создания аккаунта
3. Выполнить вход в систему с указанными при регистрации учётной записи.

Если аккаунт корректно создан, будет доступно следующее меню, приведённое на Рисунке 3.

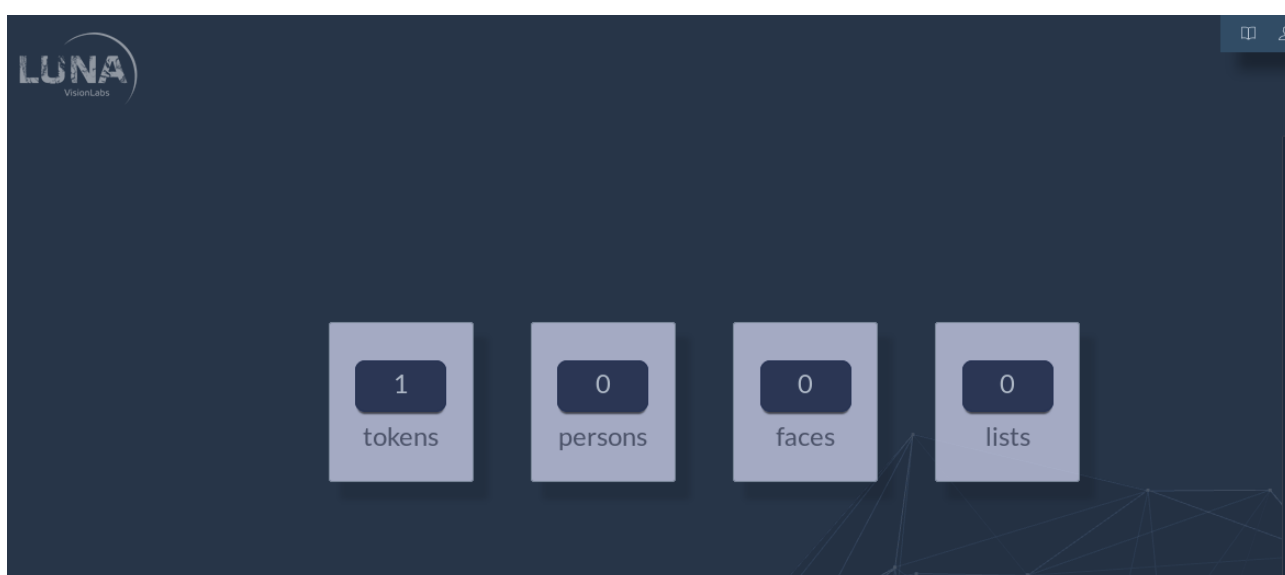


Рисунок 13.

4. Нажать tokens и сохранить значение для дальнейшего использования в настройках приложения.

Результат: Инсталляция завершена

ВНИМАНИЕ

В случае если интерфейс не доступен, регистрацию и получение токена можно провести вызовом с консоли на сервере d00bsb-vl01. Для этого необходимо выполнить, подставив свои значения

```
curl -X POST \
  http://d00bsb-vl01:5000/3/accounts \
  -H 'content-type: application/javascript' \
  -d '{
    "organization_name": "bsb_organization",
    "email": "bsb@domain.ru",
    "password": "SOME_PASSWORD"
  }'
```

В качестве ответа вернётся значение {"token":"*****"}

6. УСТАНОВКА И НАСТРОЙКА JAVA STACK

Дистрибутивы для установки доступны:

- <https://portal.rtlabs.ru/~cl8Lr> (JDK1.8)
- <https://portal.rtlabs.ru/~Dw1gO> (linux-amd64.tgz)
- <https://portal.rtlabs.ru/~ljyr1> (java-csp-5.0.39267.zip)
- <https://portal.rtlabs.ru/~a82gi> (bsb-web.tar.gz)
- <https://portal.rtlabs.ru/~SsTR8> (bsb-bkg.tar.gz)

На серверах d00bsb-app необходимо выполнить следующие действия:

- Подготовить OS
- Разметить диск
- Установить JDK1.8.141
- Установить JCSP
- Скопировать systemd скрипт управления приложением
- Сконфигурировать application.properties
- Сконфигурировать setenv.sh

6.1. РАЗМЕТКА ДИСКА

1. Убедимся, что дополнительный диск присутствует

```
fdisk -l | grep sdb
Disk /dev/sdb:
```

2. Выполнить

```
pvccreate /dev/sdb ;
vgcreate vg_data /dev/sdb ;
```

```
lvcreate --name lv_data --size 10G vg_data ;  
lvcreate --name lv_log -l 100%FREE vg_data ;  
mkfs.xfs /dev/mapper/vg_data-lv_data ;  
mkfs.xfs /dev/mapper/vg_data-lv_log ;
```

3. После создания файловой системы в конце прописываем постоянные записи в /etc/fstab

```
/dev/vg_data/lv_bsb /bsb xfs defaults 1 1  
/dev/vg_data/lv_log /var/log/bsb xfs defaults 1 1
```

4. Создать точки монтирования и монтируем

```
mkdir /bsb -p && mount /bsb  
mkdir /var/log/bsb -p && mount -a
```

6.2. СОЗДАНИЕ СИСТЕМНОГО ПОЛЬЗОВАТЕЛЯ BSB

1. Создать группу пользователей в системе

```
groupadd app_grp;
```

2. Создать пользователя bsb с группой app_grp и домашней директорией /bsb
3. Выдать права на директорию с логами

```
useradd -d /bsb -g app_grp bsb && chown -R bsb: /bsb /var/log/bsb;
```

6.3. JAVA (JDK1.8.181)

1. Создать директорию для установки

```
mkdir /usr/java
```

2. Положить установочный пакет в /usr/java/jdk-8u181-linux-x64.rpm и выполним стандартную установку

```
zypper install /usr/java/jdk-8u181-linux-x64.rpm ;
```

3. По ходу установки подтвердить все положительно

```
jdk1.8-2000:1.8.0_181-fcs.x86_64 (Plain RPM files cache): Signature verification failed [4-Signatures public  
key is not available]  
Abort, retry, ignore? [a/r/i] (a): i
```

Результат: Java установлена

6.4. УСТАНОВКА КРИПТОПРО CSP 4.0 ДЛЯ LINUX (X64, RPM)

Распаковать дистрибутив linux-amd64.tgz и выполнить установку в следующем порядке

```
tar xzf linux-amd64.tgz && cd linux-amd64;
```

```
zypper install lsb-cppcsp-base-*.noarch.rpm lsb-cppcsp-rdr-*.x86_64.rpm lsb-cppcsp-kc1-*.x86_64.rpm  
lsb-cppcsp-capilite-*.x86_64.rpm
```

Результат: Пакеты CSP установлены

6.5. JCSP 5.0.39267

1. Скопировать дистрибутив в /usr/java/java-csp-5.0.39267.zip на сервере d00bsb-app
2. Распаковать дистрибутив и выполнить установку JCSP командами

```
cd /usr/java;  
unzip java-csp-5.0.39267.zip && cd java-csp-5.0.39267 ;
```

3. Выполнить установку (с ключом -serial_jcp XXXXX-XXXXX-XXXXX-XXXXX-XXXXX-XXXXX если известен серийный номер, иначе в триальном режиме)

```
./setup_console.sh /usr/java/default/jre -force -en -install -jcp -jcsp
```

4. Об успешной установке отобразится сообщение вида

```
Executing: 100%...  
The operation complete.  
Congratulations! The operation complete.
```

5. В файле /usr/java/latest/jre/lib/security/java.security необходимо исправить порядок провайдеров

```
security.provider.11=ru.CryptoPro.JCP.JCP  
security.provider.12=ru.CryptoPro.JCSP.JCSP
```

на

```
security.provider.11=ru.CryptoPro.JCSP.JCSP  
security.provider.12=ru.CryptoPro.JCP.JCP
```

Для изменения порядков выполнить команду

```
sed -i 's|security.provider.11=ru.CryptoPro.JCP.JCP|security.provider.11=ru.CryptoPro.JCSP.JCSP|g'  
/usr/java/latest/jre/lib/security/java.security ;  
sed -i 's|security.provider.12=ru.CryptoPro.JCSP.JCSP|security.provider.12=ru.CryptoPro.JCP.JCP|g'  
/usr/java/latest/jre/lib/security/java.security ;
```

6. Проверить успешность через контрольную панель (необходимо выполнить подключение по ssh с пробросом X11. На сервере должен быть установлен пакет xauth)

```
./ControlPane.sh /usr/java/default/jre
```

Во вкладках должен присутствовать Java CSP Settings, Java JSP, JCP

7. Во вкладке Other необходимо установить флажок Suppress GOST R 34.10-2001 key usage warnings (не выдавать предупреждение по алгоритму GOST R 34.10.2001)

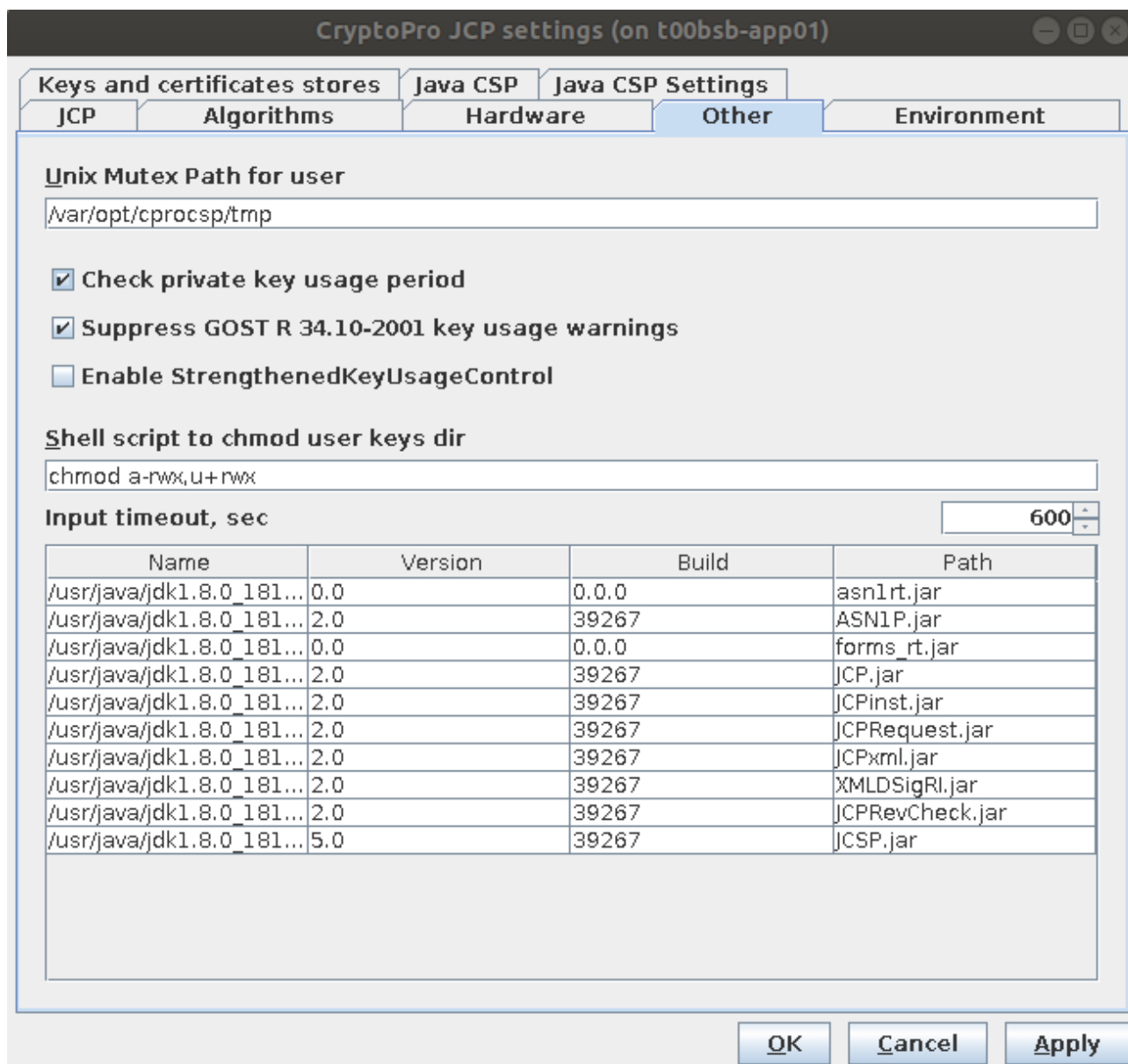


Рисунок 14.

6.6. JCSP: УСТАНОВКА КЛЮЧЕЙ УКЭП БАНКА ДЛЯ СМЭВ

При взаимодействии с видами сведений СМЭВ должна использоваться УКЭП Банка для СМЭВ, в соответствии с Федеральным законом от 06.04.2011 N 63-ФЗ (ред. от 23.06.2016) «Об электронной подписи», по рекомендациям, описанным в документе «Методические рекомендации по работе со СМЭВ».

Системный администратор получает действующую УКЭП Банка для СМЭВ у лица, наделенного полномочиями приказом по Банку, ответственным за обмен электронными документами с Государственной информационной системой о государственных и муниципальных платежах, Государственной информационной системой жилищно-коммунального хозяйства, Единой системой идентификации и аутентификации, системой

«Ключ Ростелеком» и Единым порталом государственных и муниципальных услуг» и являющегося ответственным за получение и передачу УКЭП Банка для СМЭВ.

Функциональный ключевой носитель, содержащий УКЭП Банка для СМЭВ, передается в опечатанном конверте с голограммой, подтверждающей целостность упаковки, с использованием специальной связи с оформлением Акта приема-передачи.

Системный администратор выполняет установку УКЭП Банка для СМЭВ в соответствии со следующим порядком:

1. Под системным пользователем Linux bsb подключиться удаленно к серверам подписания ИС БСБ (в соответствии с п. 5.4 документа «Комплекс технических средств Биометрической системы банка») и поместить контейнер с ключами УКЭП Банка для СМЭВ в директорию /var/opt/cproscsp/keys/bsb/
2. Выдать права для системного пользователя bsb:app_grp

```
chown -R bsb: /var/opt/cproscsp/keys/bsb/
```

3. Задать ограничения на директорию и ключи УКЭП Банка для СМЭВ

```
chmod 755 /var/opt/cproscsp/keys/bsb/*  
chmod 644 /var/opt/cproscsp/keys/bsb/*/*
```

4. Выполнить команду для установки сертификата под системным пользователем Linux bsb

```
sudo -iu bsb  
  
/opt/cproscsp/bin/amd64/certmgr -inst -cont '\\.\HDIMAGE\<alias>'  
...  
Certmgr 1.1 (c) "CryptoPro", 2007-2010.  
program for managing certificates, CRLs and stores  
  
Installing:  
=====
```

```
1-----  
....
```

5. Проверить корректность установки под системным пользователем Linux bsb

```
/opt/cproscsp/bin/amd64/certmgr -list  
...  
Certmgr 1.1 (c) "CryptoPro", 2007-2010.  
program for managing certificates, CRLs and stores  
  
=====
```

```
1-----  
Issuer      :  
...
```

6.7. FRONTEND (BSB-WEB.JAR)

Дистрибутив bsb-web.tar.gz необходимо скопировать в директорию /bsb на сервере d00bsb-app01 и распаковать под пользователем bsb

```
sudo -iu bsb;  
tar xzvf bsb-web.tar.gz;
```

6.7.1. Подготовка OS (Под пользователем root)

1. Установить лимиты для группы пользователя. Для этого вставить следующее содержимое в конец файла /etc/security/limits.conf

```
...  
# End of file  
@app_grp soft core unlimited  
@app_grp soft nproc 147456  
@app_grp - nofile 65535
```

2. Разрешить управление сервисом под sudo для системного пользователя. Для этого выполнить следующую команду

```
cat <<EOF>/etc/sudoers.d/app_grp  
bsb ALL=(ALL) NOPASSWD:SETENV: /bin/systemctl status bsb-web, /bin/systemctl reload bsb-  
web, /bin/systemctl restart bsb-web, /bin/systemctl stop bsb-web , /bin/systemctl start bsb-web  
bsb ALL=(ALL) NOPASSWD:SETENV: /bin/systemctl status bsb-bkg, /bin/systemctl reload bsb-  
bkg, /bin/systemctl restart bsb-bkg, /bin/systemctl stop bsb-bkg , /bin/systemctl start bsb-bkg  
  
EOF
```

6.7.2. Установка FRONTEND (BSB-WEB.WAR)

1. Под пользователем root установить systemd скрипт запуска\останова сервиса

```
cp /bsb/bsb-web/systemd/bsb-web.service /etc/systemd/system/;
```

2. При необходимости выставить параметры JVM -xms, -mx в файле /bsb/bsb-web/bin/setenv.sh

6.7.3. Конфигурация FRONTEND (BSB-WEB.WAR)

1. Сконфигурировать в соответствии с реальными значениями
 - ip_adress_postgres_db > (Адрес базы данных PGSQL) d00bsb-db01
 - postgres_user > (Пользователь приложения в базе данных PGSQL)
 - postgres_password > (Пароль приложения в базе данных PGSQL)
 - ip_adress_zookeeper > (IP Адрес или список адресов через “;”) d00bsb-hd01:2181,d00bsb-hd02:2181,d00bsb-hd03:2181

- ip_address_luna > (IP Адрес Luna API) d00bsb-vl01
- luna_token > (Токен Luna полученный при установке Luna)
- ip_address_rammit_mq > (IP адрес Rabbitmq сервера) d00bsb-cmq01

6.7.4. Настройки логирования

2. Сконфигурировать параметры логирования в файле /bsb/bsb-web/logging.properties

- уровень логирования задаётся параметром в файле logging.properties (logging.logback.loglevel=INFO\ERROR\DEBUG\TRACE)
- можно изменить директорию для логов в параметре: logging.path=
- qradar.logserver = localhost > qradar.logserver = ip_address_qradar
- syslog.logserver = localhost > syslog.logserver = ip_address_syslogserver

3. Выполнить перезагрузку systemd, и включить автозапуск приложений

```
systemctl daemon-reload
systemctl enable bsb-web.service
systemctl restart bsb-web.service
```

6.8. BACKEND (BSB-BKG.WAR)

Дистрибутив bsb-bkg.tar.gz скопировать в директорию /bsb на сервере d00bsb-app01 и распаковать под системным пользователем Linux bsb

```
sudo -iu bsb;
tar xzvf bsb-bkg.tar.gz;
```

6.8.1. Подготовка OS (Под пользователем root)

1. Установить лимиты для группы пользователя. Для этого вставить следующее содержимое в конец файла /etc/security/limits.conf

```
...
# End of file
@app_grp soft core unlimited
@app_grp soft nproc 147456
@app_grp - nofile 65535
```

2. Разрешить управление сервиса под sudo для системного пользователя. Для этого выполнить следующую команду

```
cat <<EOF>/etc/sudoers.d/app_grp
bsb ALL=(ALL) NOPASSWD:SETENV: /bin/systemctl status bsb-web, /bin/systemctl reload bsb-web, /bin/systemctl restart bsb-web, /bin/systemctl stop bsb-web , /bin/systemctl start bsb-web
bsb ALL=(ALL) NOPASSWD:SETENV: /bin/systemctl status bsb-bkg, /bin/systemctl reload bsb-bkg, /bin/systemctl restart bsb-bkg, /bin/systemctl stop bsb-bkg , /bin/systemctl start bsb-bkg
```

6.8.2. Установка BACKEND (bsb-bkg.war)

1. Под пользователем root установить systemd скрипт запуска/останова сервиса

```
cp /bsb/bsb-web/systemd/bsb-web.service /etc/systemd/system/;
```

2. Под пользователем root установить systemd скрипт запуска/останова сервиса

```
cp /bsb/bsb-web/systemd/bsb-bkg.service /etc/systemd/system/;
```

6.8.3. Конфигурация BACKEND (bsb-bkg.war)

1. Сконфигурировать в соответствии с реальными значениями конфигурационный файл /bsb/bsb-bkg/application.properties
 - ip_address_postgres_db > (Адрес базы данных PGSQL) d00bsb-db01
 - postgres_user > (Пользователь приложения в базе данных PGSQL)
 - postgres_password > (Пароль приложения в базе данных PGSQL)
 - ip_address_zookeeper > (IP Адрес или список адресов через “;”) d00bsb-hd01:2181,d00bsb-hd02:2181,d00bsb-hd03:2181
 - ip_address_luna > (IP Адрес Luna API) d00bsb-vl01
 - luna_token > (Токен Luna полученный при установке Luna)
 - ip_address_rammit_mq > (IP адрес Rabbitmq сервера) d00bsb-cmq01
 - smev.registrar.mnemonic=RTK014 (Мнемоника полученная при регистрации ИС в СМЭВ)
 - smev.address=http://smev3-n0.test.gosuslugi.ru:7500/smev/v1.2/ws (Адрес СМЭВ3)
 - smev.ftp.address=smev3-n0.test.gosuslugi.ru (Адрес СМЭВ3)

6.8.4. Использование встроенного механизма подписания

1. Для этого необходимо выставить следующие значения
 - keystore_alias > (Алиас контейнера JCSP)
 - keystore_password > (Пароль от контейнера ключей JCSP)
 - crypto.provider.type > (Необходимо выставить значение JCSP)

6.8.5. Использование внешнего сервера подписания HSM

2. Для этого необходимо выставить следующие значения
 - crypto.provider.type=BSB
 - sign.server.address=http://d00bsb-cs01:8091/crypto/api (d00bsb-cs01 заменить на fqdn сервера подписания)
 - sign.server.token=38884** (Указанный token на этапе конфигурации сервера подписания, settings.xml)

6.8.6. Настройки логирования

3. Сконфигурировать параметры логирования в файле /bsb/bsb-bkg/logging.properties

- уровень логирования задать параметром (logging.logback.loglevel=INFO\ERROR\DEBUG\TRACE)
 - можно изменить директорию для логов в параметре: logging.path=
 - qradar.logserver = localhost > qradar.logserver = ip_address_qradar
 - syslog.logserver = localhost > syslog.logserver = ip_address_syslogserver
4. Выполнить перезагрузку systemd, и включить автозапуск приложений

```
systemctl daemon-reload
systemctl enable bsb-bkg.service
systemctl restart bsb-bkg.service
```

7. УСТАНОВКА И НАСТРОЙКА NGINX (РАЗДАЧА СТАТИКИ, БАЛАНСИРОВКА)

7.1. ПОДГОТОВКА O.S.

Дистрибутив с конфигурациями доступен по ссылке: <https://portal.rtlabs.ru/~aY4Nl>
Предварительно необходимо подключить официальный репозиторий nginx

```
zypper addrepo -G -t yum -c 'http://nginx.org/packages/sles/12' nginx;
wget http://nginx.org/keys/nginx\_signing.key ;
rpm --import nginx_signing.key ;
```

Перенести дистрибутив bsb-front.tar.gz на сервер в удобную директорию

Распаковать tar xzvf bsb-front.tar.gz

Для конфигурации параметров ядра необходимо

1. Скопировать файл

bsb-front/sysctl.d/99-sysctl.conf

в

/etc/sysctl.d/99-sysctl.conf

2. Применить изменения

```
sysctl -p
```

7.2. РАЗМЕТКА ДИСКА

1. Убедимся, что дополнительный диск присутствует

```
fdisk -l | grep sdb
Disk /dev/sdb:
Выполняем:
pvcreate /dev/sdb ;
vgcreate vg_data /dev/sdb ;
lvcreate --name lv_log -l 100%FREE vg_data ;
```

```
mkfs.xfs /dev/mapper/vg_data-lv_log ;
```

2. После создания файловой системы в конце прописываем постоянные записи в /etc/fstab

```
/dev/vg_data/lv_log /var/log/nginx xfs defaults 1 1
```

Далее, создать точки монтирования и монтируем

```
mkdir /var/log/nginx -p && mount -a
```

7.3. УСТАНОВКА NGINX

1. На сервере выполнить

```
zypper install nginx
```

2. Проверим версию

```
nginx -v  
nginx version: nginx/1.14.0
```

7.3.1. Конфигурации NGINX

Файлы необходимо скопировать

bsb-front/nginx/nginx.conf

в

/etc/nginx/nginx.conf на сервере

(Подтвердить с заменой)

```
worker_processes <Должно равняться количеству ЦПУ на сервере>
```

При использовании HTTPS необходимо положить *.key, *.pem в директорию на сервере /etc/nginx/ssl

Скопировать конфигурационный файл

bsb-front/nginx/conf.d/bsb_https.conf

в

/etc/nginx/conf.d/bsb_https.conf

В блоке upstream bsb-web значение server_web_fqdn > изменить на реальный сервер сервиса bsb-web.war.

Продублировать строчку и заменить server_fqdn на второй сервер в балансировке.

В блоке upstream bsb-bkg значение server_bkg_fqdn > изменить на реальный сервер сервиса bsb-bkg.war.

Продублировать строчку и заменить server_fqdn на второй сервер в балансировке.

- site_name > Внешний DNS системы
- ssl_key > Имя файла *.key в /etc/nginx/ssl/*.key
- ssl_cert > Имя открытой части сертификата в /etc/nginx/ssl/*.{pem\cert}

При использовании HTTP необходимо использовать bsb_http.conf

Скопировать nginx/conf.d/bsb_http.conf в /etc/nginx/conf.d/bsb_http.conf

В блоке upstream bsb-web значение server_web_fqdn > изменить на реальный сервер сервиса bsb-web.war

Продублировать строку и заменить server_fqdn на второй сервер в балансировке.

В блоке upstream bsb-bkg значение server_bkg_fqdn > изменить на реальный сервер сервиса bsb-bkg.war

Продублировать строку и заменить server_fqdn на второй сервер в балансировке

site_name > Внешний DNS системы

7.4. СТАТИКА (BSB-FRONT.TAR.GZ)

1. Дистрибутив со статикой bsb-front/bsb-front.tar.gz необходимо перенести в папку на сервере /usr/share/nginx/bsb-front.tar.gz и разархивировать

<pre>cd /usr/share/nginx ; tar xzvf bsb-front.tar.gz</pre>
--

2. Скорректировать пользователя и права на папку

<pre>chown -R nginx: /usr/share/nginx ; chmod -R 0755 /usr/share/nginx ;</pre>
--

3. Запустим nginx и добавим в автозагрузку сервис

<pre>systemctl enable nginx ; systemctl restart nginx ;</pre>
