

**Руководство по инсталляции**

**Программы для ЭВМ  
«Мониторинг знаний персонала»**

На 29 листах

## Оглавление

1. Общие положения.....	4
1.1 Полное наименование системы, обозначение.....	4
1.2 Название документа .....	4
1.3 Название документа .....	4
2. Требования к полигону развёртывания .....	4
2.1 Аппаратное обеспечение серверов.....	4
2.2 Требования к рабочим станциям.....	5
2.3 Программное обеспечение серверов и рабочих станций.....	5
3. Порядок установки .....	6
3.1 Содержимое установочного диска .....	6
3.2 Установка и настройка сервера баз данных .....	7
3.3 Установка и настройка Docker CE .....	21
3.4 Установка и настройка сервера управления очередями .....	22
3.5 Установка и настройка сервера приложений.....	23
3.6 Установка и настройка сервера синхронизации .....	24
3.7 Установка и настройка сервера интерфейсов управления .....	25
3.8 Установка и настройка сервера хранения .....	26
3.9 Установка и настройка сервера мониторинга.....	28
4. Проверка настройки приложения.....	29

### Обозначения и сокращения

Обозначение/ сокращение	Полное наименование
Веб-приложение	Подсистема «Работа с персоналом»
ПО	Программное обеспечение
Веб-браузер	Прикладное программное обеспечение для просмотра веб-страниц, содержания веб-документов, компьютерных файлов и их каталогов.
БД	База данных

## 1. Общие положения

### 1.1 Полное наименование системы, обозначение

Полное наименование системы: Подсистема «Работа с персоналом».

Краткое наименование (обозначение) системы: Система.

### 1.2 Название документа

Настоящий документ входит в комплект эксплуатационной документации по Подсистеме «Работа с персоналом» и предназначен для развертывания Подсистемы «Работа с персоналом».

### 1.3 Название документа

Документ «Руководство по инсталляции» входит в комплект отчетной документации.

## 2. Требования к полигону развёртывания

### 2.1 Аппаратное обеспечение серверов

Аппаратное обеспечение серверов описано в таблице 1.

Таблица 1 – Требования к аппаратному обеспечению серверов

№ п/п	Функциональное назначение	IP-адрес	Имя сервера	Кол-во ядер процессора	Объем оперативной памяти	Объем физической памяти
1	Сервер интерфейсов управления	<WEB IP>	<WEB HOSTNAME>	4	8 Гб	50 Гб
2	Сервер приложений – 1	<APP1 IP>	<APP1 HOSTNAME>	4	8 Гб	100 Гб
3	Сервер приложений – 2	<APP2 IP>	<APP2 HOSTNAME>	8	8 Гб	100 Гб
4	Сервер базы данных – узел 1	<DB1 IP>	<DB1 HOSTNAME>	4	8 Гб	500 Гб
5	Сервер базы данных – узел 2	<DB2 IP>	<DB2 HOSTNAME>	4	8 Гб	500 Гб
6	Сервер базы данных – узел 3	<DB3 IP>	<DB3 HOSTNAME>	4	8 Гб	500 Гб
7	Сервер управления очередями – узел 1	<SYNC1 IP>	<SYNC1 HOSTNAME>	4	8 Гб	100 Гб
8	Сервер управления очередями – узел 2	<SYNC2 IP>	<SYNC2 HOSTNAME>	4	8 Гб	100 Гб
9	Сервер управления очередями – узел 3	<SYNC3 IP>	<SYNC3 HOSTNAME>	4	8 Гб	100 Гб
10	Сервер синхронизации	<SYNC4 IP>	<SYNC4 HOSTNAME>	4	8 Гб	100 Гб
11	Сервер мониторинга	<MON IP>	<MON HOSTNAME>	4	8 Гб	50 Гб
12	Сервер хранения резервных копий	<BACKUP IP>	<BACKUP HOSTNAME>	1	4 Гб	1024 Гб
13	Сервер хранения – узел 1	<STORAGE1 IP>	<STOR1 HOSTNAME>	4	8 Гб	500 Гб
14	Сервер хранения – узел 2	<STORAGE2 IP>	<STOR2 HOSTNAME>	4	8 Гб	500 Гб
15	Сервер хранения – узел 3	<STORAGE3 IP>	<STOR3 HOSTNAME>	4	8 Гб	500 Гб

## 2.2 Требования к рабочим станциям

Требования к рабочим станциям пользователей описаны в таблице 2.

Таблица 2 – Требования к рабочим станциям пользователей

№ п/п	Функциональное назначение	Кол-во ядер процессора	Объем оперативной памяти	Объем физической памяти	Пропускная способность сетевого интерфейса	Размер и разрешающая способность монитора
1	Рабочая станция пользователя	2	4 ГБ	20 ГБ	100 Мбит	Не менее 1280x1024 пикселей

## 2.3 Программное обеспечение серверов и рабочих станций

Требования к программному обеспечению серверов и рабочих станций описаны в таблице 3.

Таблица 3 – Требования к программному обеспечению серверов и рабочих станций

№ п/п	Функциональное назначение	Системное программное обеспечение	Прикладное программное обеспечение	Дополнительное программное обеспечение
1	Сервер интерфейсов управления	Linux (Ubuntu 20.04 x64)	Подсистема «Работа с персоналом»	<ul style="list-style-type: none"><li>– Docker CE</li><li>– Docker-compose</li><li>– Nginx</li><li>– Cadvisor</li><li>– Nodeexporter</li></ul>
2	Сервер приложений – 1	Linux (Ubuntu 20.04 x64)	Подсистема «Работа с персоналом»	<ul style="list-style-type: none"><li>– Docker CE</li><li>– Docker-compose</li><li>– Cadvisor</li><li>– Nodeexporter</li><li>– Promtail</li></ul>
3	Сервер приложений – 2	Linux (Ubuntu 20.04 x64)	Подсистема «Работа с персоналом»	<ul style="list-style-type: none"><li>– Docker CE</li><li>– Docker-compose</li><li>– Cadvisor</li><li>– Nodeexporter</li><li>– Promtail</li></ul>
4	Сервер базы данных – узел 1	Linux (Ubuntu 20.04 x64)	-	<ul style="list-style-type: none"><li>– PostgreSQL</li><li>– etcd</li><li>– Patroni</li><li>– Keepalived</li><li>– HAProxy</li><li>– Cadvisor</li><li>– Nodeexporter</li></ul>
5	Сервер базы данных – узел 2	Linux (Ubuntu 20.04 x64)	-	<ul style="list-style-type: none"><li>– PostgreSQL</li><li>– etcd</li><li>– Patroni</li><li>– Keepalived</li><li>– HAProxy</li><li>– Cadvisor</li><li>– Nodeexporter</li></ul>
6	Сервер базы данных – узел 3	Linux (Ubuntu 20.04 x64)	-	<ul style="list-style-type: none"><li>– PostgreSQL</li><li>– etcd</li><li>– Patroni</li><li>– Keepalived</li><li>– HAProxy</li></ul>

№ п/п	Функциональное назначение	Системное программное обеспечение	Прикладное программное обеспечение	Дополнительное программное обеспечение
				– Cadvisor – Nodeexporter
7	Сервер управления очередями – узел 1	Linux (Ubuntu 20.04 x64)	-	– RabbitMQ – Redis – Cadvisor – Nodeexporter
8	Сервер управления очередями – узел 2	Linux (Ubuntu 20.04 x64)	-	– RabbitMQ – Redis – Cadvisor – Nodeexporter
9	Сервер управления очередями – узел 3	Linux (Ubuntu 20.04 x64)	-	– RabbitMQ – Redis – Cadvisor – Nodeexporter
10	Сервер синхронизации	Linux (Ubuntu 20.04 x64)	Подсистема «Работа с персоналом»	– Docker CE – Docker-compose – GlusterFS client – Cadvisor – Nodeexporter – Promtail
11	Сервер мониторинга	Linux (Ubuntu 20.04 x64)	-	– Grafana – Prometheus – Alertmanager – Loki – Cadvisor – Nodeexporter
12	Сервер хранения резервных копий	Linux (Ubuntu 20.04 x64)	-	– Cadvisor – Nodeexporter
13	Сервер хранения – узел 1	Linux (Ubuntu 20.04 x64)	-	– GlusterFS server – Cadvisor – Nodeexporter
14	Сервер хранения – узел 2	Linux (Ubuntu 20.04 x64)	-	– GlusterFS server – Cadvisor – Nodeexporter
15	Сервер хранения – узел 3	Linux (Ubuntu 20.04 x64)	-	– GlusterFS server – Cadvisor – Nodeexporter
16	Рабочая станция пользователя	Windows 7 и выше, Linux	Не требуется	Актуальные версии Google Chrome (56 и выше)

### 3. Порядок установки

Для выполнения действий установки учетная запись пользователя должна обладать правами администратора на серверах развертывания Системы. Все команды необходимо выполнять в командной строке терминала. Установка предполагает, что сервера имеют доступ в Интернет.

#### 3.1 Содержимое установочного диска

Содержимое установочного диска:

Руководство по установке	Номер страницы: 6 Версия 1.0
--------------------------	---------------------------------

- папка `config` – конфигурационные файлы компонентов Системы;
- папка `dist` – дистрибутивы микросервисов и пользовательского интерфейса Системы;
- папка `deploy` – вспомогательные скрипты для установки Системы.

### 3.2 Установка и настройка сервера баз данных

В данном разделе описаны условия и действия, необходимые для установки Базы данных.

Установка предполагает, что на (см. Таблица 1) еще не установлено никакое дополнительное программное обеспечение. Если PostgreSQL уже установлено, то можно сразу перейти к п.10 данного раздела. Установку PostgreSQL (п. 1 – п.9) необходимо провести на каждом из узлов «Сервер базы данных».

1. Добавьте репозиторий PostgreSQL:

```
sudo sh -c 'echo "deb http://apt.postgresql.org/pub/repos/apt $(lsb_release -cs)-pgdg main" > /etc/apt/sources.list.d/pgdg.list'
```

2. Добавьте ключ для репозитория:

```
wget --quiet -O - https://www.postgresql.org/media/keys/ACCC4CF8.asc | sudo gpg --dearmor -o /etc/apt/trusted.gpg.d/pgdg.gpg
```

3. Обновите индекс пакетов:

```
sudo apt -y update
```

4. Установите PostgreSQL:

```
sudo apt -y install postgresql-13
```

5. Включите прослушивание запросов к PostgreSQL на всех сетевых интерфейсах и разрешите доступ пользователю `postgres`, для этого отредактируйте файл `/etc/postgresql/13/main/pg_hba.conf` добавив следующие записи:

```
local all postgres trust
host all all 0.0.0.0/0 md5
```

6. Отредактируйте файл `/etc/postgresql/13/main/conf.d/postgresql.conf` оптимизировав настройки PostgreSQL под параметры виртуальной машины. Для виртуальной машины с параметрами из таблицы 1 настройки следующие:

```
listen_addresses = '*'
password_encryption = md5
max_connections = 500
shared_buffers = 1989MB
effective_cache_size = 5968MB
maintenance_work_mem = 398MB
checkpoint_completion_target = 0.9
wal_buffers = 62MB
default_statistics_target = 500
random_page_cost = 2
effective_io_concurrency = 100
work_mem = 20MB
min_wal_size = 1989MB
max_wal_size = 7957MB
max_worker_processes = 4
max_parallel_workers_per_gather = 2
max_parallel_workers = 4
max_parallel_maintenance_workers = 2
log_line_prefix = '%m [%p] %q%u@%d ' # special values:
log_timezone = 'Europe/Moscow'
datestyle = 'iso, mdy'
```

Руководство по инсталляции	Номер страницы: 7
	Версия 1.0

```
timezone = 'Europe/Moscow'
```

7. Активируйте и перезапустите сервис PostgreSQL и проверьте его статус:

```
sudo systemctl enable postgresql
```

```
sudo systemctl restart postgresql
```

```
sudo systemctl status postgresql
```

8. Создайте пользователей и базы данных приложения. Описание и конфигурация ниже предполагает, что БД будет созданы с именами administration, quiz, sync, а также будут созданы пользователи с такими же именами для доступа к этим БД. В процессе создания пользователей необходимо указать пароли пользователей и обязательно запомнить их.

```
sudo -i -u postgres
```

```
createuser --login --pwprompt administration
```

```
createdb --encoding UTF8 --owner administration administration
```

```
createuser --login --pwprompt quiz
```

```
createdb --encoding UTF8 --owner quiz quiz
```

```
createuser --login --pwprompt sync
```

```
createdb --encoding UTF8 --owner sync sync
```

```
exit
```

9. Создайте пользователей баз данных для управления кластером и репликацией:

```
sudo -i -u postgres
```

```
createuser --login --createrole --createdb --pwrompt admin
```

```
createuser --login --replication --pwprompt replicator
```

```
exit
```

10. Проверьте возможность подключения к созданным базам данных. После ввода команды нужно будет ввести пароль пользователя созданной базы данных. Вместо **<DB1\_IP>**, **<DB2\_IP>**, **<DB3\_IP>** указать IP адрес соответствующего «Сервера базы данных» (см. Таблица 1). Ответом на команду должна быть строка с номером версии PostgreSQL.

```
psql -h <DB1_IP> -U administration -W -c 'SHOW SERVER_VERSION;' -qt
```

```
psql -h <DB1_IP> -U quiz -W -c 'SHOW SERVER_VERSION;' -qt
```

```
psql -h <DB1_IP> -U sync -W -c 'SHOW SERVER_VERSION;' -qt
```

```
psql -h <DB2_IP> -U administration -W -c 'SHOW SERVER_VERSION;' -qt
```

```
psql -h <DB2_IP> -U quiz -W -c 'SHOW SERVER_VERSION;' -qt
```

```
psql -h <DB2_IP> -U sync -W -c 'SHOW SERVER_VERSION;' -qt
```

```
psql -h <DB3_IP> -U administration -W -c 'SHOW SERVER_VERSION;' -qt
```

```
psql -h <DB3_IP> -U quiz -W -c 'SHOW SERVER_VERSION;' -qt
```

```
psql -h <DB3_IP> -U sync -W -c 'SHOW SERVER_VERSION;' -qt
```

11. Для обеспечения отказоустойчивости узлы «Серверов базы данных» (см. Таблица 1) должны быть объединены в кластер под управлением Patroni. На узлах должны быть выставлены одинаковые временные зоны, определены имена и по этим именам они должны быть друг другу доступны

12. На каждом из узлов «Серверов базы данных» (см. Таблица 1) установить etcd:

```
sudo apt -y install etcd
```

```
sudo systemctl stop etcd
```

```
sudo systemctl disable etcd
```

```
sudo rm -rf /var/lib/etcd/default
```

13. На «Сервере базы данных» – узел 1 (см. Таблица 1) провести настройку etcd:

```
sudo mv /etc/default/etcd /etc/default/etcd-orig
```

```
sudo nano /etc/default/etcd
```

Добавить следующую конфигурацию:

```
ETCD_NAME=<DB1_HOSTNAME>
```

```
ETCD_DATA_DIR="/var/lib/etcd/<DB1_HOSTNAME>"
```



```
ETCD_LISTEN_PEER_URLS="http:// <DB1 IP>:2380"
ETCD_LISTEN_CLIENT_URLS="http://0.0.0.0:2379"
ETCD_INITIAL_ADVERTISE_PEER_URLS="http:// <DB1 IP>:2380"
ETCD_INITIAL_CLUSTER="<DB1 HOSTNAME>=http://<DB1 IP>:2380, <DB2
HOSTNAME>=http://<DB2 IP>:2380,<DB3 HOSTNAME>=http://<DB3 IP>:2380"
ETCD_INITIAL_CLUSTER_STATE="new"
ETCD_INITIAL_CLUSTER_TOKEN="etcd-cluster"
ETCD_ADVERTISE_CLIENT_URLS="http://0.0.0.0:2379"
ETCD_ENABLE_V2="true"
```

14. На «Сервере базы данных» – узел 2 (см. Таблица 1) провести настройку etcd:  
`sudo mv /etc/default/etcd /etc/default/etcd-orig`  
`sudo nano /etc/default/etcd`

Добавить следующую конфигурацию:

```
ETCD_NAME=<DB2 HOSTNAME>
ETCD_DATA_DIR="/var/lib/etcd/<DB2 HOSTNAME>"
ETCD_LISTEN_PEER_URLS="http://<DB2 IP>:2380"
ETCD_LISTEN_CLIENT_URLS="http://0.0.0.0:2379"
ETCD_INITIAL_ADVERTISE_PEER_URLS="http://<DB2 IP>:2380"
ETCD_INITIAL_CLUSTER="<DB1 HOSTNAME>=http://<DB1 IP>:2380,<DB2
HOSTNAME>=http://<DB2 IP>:2380,<DB3 HOSTNAME>=http://<DB3 IP>:2380"
ETCD_INITIAL_CLUSTER_STATE="new"
ETCD_INITIAL_CLUSTER_TOKEN="etcd-cluster"
ETCD_ADVERTISE_CLIENT_URLS="http://0.0.0.0:2379"
ETCD_ENABLE_V2="true"
```

15. На «Сервере базы данных» – узел 3 (см. Таблица 1) провести настройку etcd:  
`sudo mv /etc/default/etcd /etc/default/etcd-orig`  
`sudo nano /etc/default/etcd`

Добавить следующую конфигурацию:

```
ETCD_NAME=<DB3 HOSTNAME>
ETCD_DATA_DIR="/var/lib/etcd/<DB3 HOSTNAME>"
ETCD_LISTEN_PEER_URLS="http://<DB3 IP>:2380"
ETCD_LISTEN_CLIENT_URLS="http://0.0.0.0:2379"
ETCD_INITIAL_ADVERTISE_PEER_URLS="http://<DB3 IP>:2380"
ETCD_INITIAL_CLUSTER="<DB1 HOSTNAME>=http://<DB1 IP>:2380,<DB2
HOSTNAME>=http://<DB2 IP>:2380,<DB3 HOSTNAME>=http://<DB3 IP>:2380"
ETCD_INITIAL_CLUSTER_STATE="new"
ETCD_INITIAL_CLUSTER_TOKEN="etcd-cluster"
ETCD_ADVERTISE_CLIENT_URLS="http://0.0.0.0:2379"
ETCD_ENABLE_V2="true"
```

16. На каждом из узлов «Серверов базы данных» (см. Таблица 1) провести настройку переменных окружения:

```
cd ~
```

```
nano .profile
```

Добавить следующую конфигурацию:

```
export PGDATA="/var/lib/postgresql/13/data"
export ETCDCTL_API="3"
export PATRONI_ETCD_URL="http://127.0.0.1:2379"
export PATRONI_SCOPE="pg_cluster"
ENDPOINTS=<DB1 IP>:2379,<DB2 IP>:2379,<DB3 IP>:2379
```

17. На каждом из узлов «Серверов базы данных» (см. Таблица 1) одновременно запустить etcd для запуска кластера:

Руководство по установке	Номер страницы: 9
	Версия 1.0

```
sudo systemctl start etcd && sudo systemctl enable etcd
```

Статус проверить при помощи команды – состояние должно быть active (running)

```
sudo systemctl status etcd
```

18. С любого из узлов «Серверов базы данных» (см. Таблица 1) проверить состояние etcd кластера:

```
source ~/.profile
```

```
etcdctl endpoint status --write-out=table --endpoints=$ENDPOINTS
```

Вывод должен быть подобен этому:

```
+-----+-----+-----+-----+-----+-----+-----+
| ENDPOINT | ID | VERSION | DB SIZE | IS LEADER | RAFT TERM | RAFT INDEX |
+-----+-----+-----+-----+-----+-----+-----+
| <DB1 IP>:2379 | 9f2153c5a394bd0f | 3.2.26 | 25 kB | false | 1306 | 52709717 |
| <DB2 IP>:2379 | e86a34033259c511 | 3.2.26 | 25 kB | true | 1306 | 52709717 |
| <DB3 IP>:2379 | ffa44b073b20a4cd | 3.2.26 | 25 kB | false | 1306 | 52709717 |
+-----+-----+-----+-----+-----+-----+-----+
```

19. На каждом из узлов «Серверов базы данных» (см. Таблица 1) провести установку Patroni:

```
sudo apt -y install python3 python3-pip python3-dev libpq-dev patroni
```

```
sudo pip3 install --upgrade launchpadlib setuptools
```

```
sudo systemctl stop patroni
```

```
sudo systemctl disable patroni
```

20. На «Сервере базы данных» – узел 1 (см. Таблица 1) провести настройку patroni:

```
sudo nano /etc/patroni/config.yml
```

Добавить следующую конфигурацию:

```
scope: pg_cluster
```

```
namespace: /service/
```

```
name: <DB1 HOSTNAME>
```

```
restapi:
```

```
listen: <DB1 IP>:8008
```

```
connect_address: <DB1 IP>:8008
```

```
etcd:
```

```
hosts: <DB1 IP>:2379,<DB2 IP>:2379,<DB3 IP>:2379
```

```
bootstrap:
```

```
dc:
```

```
ttl: 30
```

```
loop_wait: 10
```

```
retry_timeout: 10
```

```
maximum_lag_on_failover: 1048576
```

```
postgresql:
```

```
use_pg_rewind: true
```

```
use_slots: true
```

```
parameters:
```

```
initdb:
```

```
- encoding: UTF8
```

```
- data-checksums
```

*pg\_hba:*

- *host replication replicator 127.0.0.1/32 md5*
- *host replication replicator <DB1 IP>/32 md5*
- *host replication replicator <DB2 IP>/32 md5*
- *host replication replicator <DB3 IP>/32 md5*
- *host all all 0.0.0.0/0 md5*

*users:*

*admin:*

*password: <ADMINISTRATOR PASSWORD>*

*options:*

- *createrole*
- *createdb*

*postgresql:*

*listen: <DB1 IP>:5432*

*connect\_address: <DB1 IP>:5432*

*data\_dir: /var/lib/postgresql/13/main*

*bin\_dir: /usr/lib/postgresql/13/bin*

*pgpass: /tmp/pgpass*

*authentication:*

*replication:*

*username: replicator*

*password: <REPLICATOR PASSWORD>*

*superuser:*

*username: postgres*

*password: <POSTGRES PASSWORD>*

*tags:*

*nofailover: false*

*noloadbalance: false*

*clonefrom: false*

*nosync: false*

Значения паролей должны быть заданы для своего окружения

21. На «Сервере базы данных» – узел 2 (см. Таблица 1) провести настройку  
*patroni:*

*sudo nano /etc/patroni/config.yml*

Добавить следующую конфигурацию:

*scope: pg\_cluster*

*namespace: /service/*

*name: <DB2 HOSTNAME>*

*restapi:*

*listen: <DB2 IP>:8008*

*connect\_address: <DB2 IP>:8008*

*etcd:*

*hosts: <DB1 IP>:2379,<DB2 IP>:2379,<DB3 IP>:2379*

*bootstrap:*

Руководство по инсталляции	Номер страницы: 11
	Версия 1.0

*dcs:*  
*ttl: 30*  
*loop\_wait: 10*  
*retry\_timeout: 10*  
*maximum\_lag\_on\_failover: 1048576*  
*postgresql:*  
*use\_pg\_rewind: true*  
*use\_slots: true*  
*parameters:*

*initdb:*  
*- encoding: UTF8*  
*- data-checksums*

*pg\_hba:*  
*- host replication replicator 127.0.0.1/32 md5*  
*- host replication replicator <DB1 IP>/32 md5*  
*- host replication replicator <DB2 IP>/32 md5*  
*- host replication replicator <DB3 IP>/32 md5*  
*- host all all 0.0.0.0/0 md5*

*users:*  
*admin:*  
*password: <ADMINISTRATOR PASSWORD>*  
*options:*  
*- createrole*  
*- createdb*

*postgresql:*  
*listen: <DB2 IP>:5432*  
*connect\_address: <DB2 IP>:5432*  
*data\_dir: /var/lib/postgresql/13/main*  
*bin\_dir: /usr/lib/postgresql/13/bin*  
*pgpass: /tmp/pgpass*  
*authentication:*  
*replication:*  
*username: replicator*  
*password: <REPLICATOR PASSWORD>*  
*superuser:*  
*username: postgres*  
*password: <POSTGRES PASSWORD>*

*tags:*  
*nofailover: false*  
*noloadbalance: false*  
*clonefrom: false*  
*nosync: false*

Значения паролей должны быть заданы для своего окружения

22. На «Сервере базы данных» – узел 3 (см. Таблица 1) провести настройку

patroni:

*sudo nano /etc/patroni/config.yml*

Руководство по инсталляции	Номер страницы: 12 Версия 1.0
----------------------------	----------------------------------

Добавить следующую конфигурацию:

*scope: pg\_cluster*

*namespace: /service/*

*name: <DB3 HOSTNAME>*

*restapi:*

*listen: <DB3 IP>:8008*

*connect\_address: <DB3 IP>:8008*

*etcd:*

*hosts: <DB1 IP>:2379,<DB2 IP>:2379,<DB3 IP>:2379*

*bootstrap:*

*dcs:*

*ttl: 30*

*loop\_wait: 10*

*retry\_timeout: 10*

*maximum\_lag\_on\_failover: 1048576*

*postgresql:*

*use\_pg\_rewind: true*

*use\_slots: true*

*parameters:*

*initdb:*

*- encoding: UTF8*

*- data-checksums*

*pg\_hba:*

*- host replication replicator 127.0.0.1/32 md5*

*- host replication replicator <DB1 IP>/32 md5*

*- host replication replicator <DB2 IP>/32 md5*

*- host replication replicator <DB3 IP>/32 md5*

*- host all all 0.0.0.0/0 md5*

*users:*

*admin:*

*password: <ADMINISTRATOR PASSWORD>*

*options:*

*- createrole*

*- createdb*

*postgresql:*

*listen: <DB3 IP>:5432*

*connect\_address: <DB3 IP>:5432*

*data\_dir: /var/lib/postgresql/13/main*

*bin\_dir: /usr/lib/postgresql/13/bin*

*pgpass: /tmp/pgpass*

*authentication:*

*replication:*

*username: replicator*

*password: <REPLICATOR PASSWORD>*

```
superuser:
  username: postgres
  password: <POSTGRES PASSWORD>
```

```
tags:
  nofailover: false
  noloadbalance: false
  clonefrom: false
  nosync: false
```

Значения паролей должны быть заданы для своего окружения

23. На каждом из узлов «Серверов базы данных» (см. Таблица 1) запустить `patroni` для запуска кластера:

```
sudo systemctl start patroni && sudo systemctl enable patroni
```

24. Статус кластера можно проверить при помощи команды

```
sudo systemctl status patroni
```

Состояние должно быть `active (running)`, один из узлов должен быть `leader`, остальные `secondary`

25. На каждом из узлов «Серверов базы данных» (см. Таблица 1) установить `keepalived`:

```
sudo apt -y install keepalived
sudo systemctl stop keepalived
sudo systemctl disable keepalived
```

26. На каждом из узлов «Серверов базы данных» (см. Таблица 1) провести настройку для `keepalived`.

```
sudo nano /etc/sysctl.conf
```

Добавить следующую конфигурацию:

```
net.ipv4.ip_nonlocal_bind = 1
```

```
net.ipv4.ip_forward = 1
```

Применить конфигурацию:

```
sudo sysctl --system
```

```
sudo sysctl -p
```

27. На «Сервере базы данных» – узел 1 (см. Таблица 1) провести настройку `keepalived`. Выбранный виртуальный IP адрес `<CLUSTER VIP>`.

```
sudo nano /etc/keepalived/keepalived.conf
```

Добавить следующую конфигурацию:

```
vrrp_script chk_haproxy {
    script "pkill -0 haproxy"
    interval 5
    weight -4
    fall 2
    rise 1
}
```

```
vrrp_script chk_lb {
    script "pkill -0 keepalived"
    interval 1
    weight 6
    fall 2
    rise 1
}
```

```

vrrip_script chk_servers {
    script "echo 'GET /are-you-ok' | nc 127.0.0.1 7000 | grep -q '200 OK'"
    interval 2
    weight 2
    fall 2
    rise 2
}

vrrip_instance vrrp_1 {
    interface ens160
    state MASTER
    virtual_router_id 51
    priority 101
    advert_int 6
    virtual_ipaddress {
        <CLUSTER_VIP>
    }
    track_interface {
        ens160 weight -2
    }
    track_script {
        chk_haproxy
        chk_lb
    }
}

```

28. На «Сервере базы данных» – узел 2 (см. Таблица 1) провести настройку keepalived. Выбранный виртуальный IP адрес <CLUSTER\_VIP>.

*sudo nano /etc/keepalived/keepalived.conf*

Добавить следующую конфигурацию:

```

vrrip_script chk_haproxy {
    script "pkill -0 haproxy"
    interval 5
    weight -4
    fall 2
    rise 1
}

vrrip_script chk_lb {
    script "pkill -0 keepalived"
    interval 1
    weight 6
    fall 2
    rise 1
}

```

```

vrrip_script chk_servers {
    script "echo 'GET /are-you-ok' | nc 127.0.0.1 7000 | grep -q '200 OK'"
    interval 2
    weight 2
    fall 2
    rise 2
}

```

```

}

vrrp_instance vrrp_1 {
    interface ens160
    state BACKUP
    virtual_router_id 51
    priority 100
    advert_int 6
    virtual_ipaddress {
        <CLUSTER_VIP>
    }
    track_interface {
        ens160 weight -2
    }
    track_script {
        chk_haproxy
        chk_lb
    }
}

```

29. На «Сервере базы данных» – узел 3 (см. Таблица 1) провести настройку keepalived. Выбранный виртуальный IP адрес <CLUSTER\_VIP>.

```
sudo nano /etc/keepalived/keepalived.conf
```

Добавить следующую конфигурацию:

```

vrrp_script chk_haproxy {
    script "pkill -0 haproxy"
    interval 5
    weight -4
    fall 2
    rise 1
}

vrrp_script chk_lb {
    script "pkill -0 keepalived"
    interval 1
    weight 6
    fall 2
    rise 1
}

vrrp_script chk_servers {
    script "echo 'GET /are-you-ok' | nc 127.0.0.1 7000 | grep -q '200 OK'"
    interval 2
    weight 2
    fall 2
    rise 2
}

vrrp_instance vrrp_1 {
    interface ens160
    state BACKUP
    virtual_router_id 51

```



```

priority 99
advert_int 6
virtual_ipaddress {
    <CLUSTER_VIP>
}
track_interface {
    ens160 weight -2
}
track_script {
    chk_haproxy
    chk_lb
}
}

```

30. На каждом из узлов «Серверов базы данных» (см. Таблица 1) запустить keepalived:

```
sudo systemctl start keepalived && sudo systemctl enable keepalived
```

На узле, который является leader кластера patroni проверить, что виртуальный IP адрес настроен на сетевом интерфейсе – при запуске команды ниже должны увидеть дополнительный IP адрес, в нашем случае <CLUSTER\_VIP>.

```
ip addr show ens160
```

31. На каждом из узлов «Серверов базы данных» (см. Таблица 1) установить HAProxy:

```

sudo apt -y install haproxy
sudo systemctl stop haproxy
sudo systemctl disable haproxy

```

32. На «Сервере базы данных» – узел 1 (см. Таблица 1) провести настройку HAProxy.

```

sudo mv /etc/haproxy/haproxy.cfg /etc/haproxy/haproxy.cfg.orig
sudo nano /etc/haproxy/haproxy.cfg

```

Добавить следующую конфигурацию:

```

global
    log 127.0.0.1 local2
    log /dev/log local0
    log /dev/log local1 notice
    chroot /var/lib/haproxy
    stats socket /run/haproxy/admin.sock mode 660 level admin expose-fd listeners
    stats timeout 30s
    user haproxy
    group haproxy
    maxconn 4000
    daemon

```

```

defaults
    mode                tcp
    log                 global
    option               tcplog
    retries              3
    timeout queue        3m
    timeout connect      10s
    timeout client        3m
    timeout server       3m

```

```
timeout check      10s
maxconn            3000
```

```
listen stats
mode http
bind *:7000
stats enable
stats uri /
```

```
listen primary
bind <CLUSTER_VIP>:5000
option httpchk OPTIONS /master
http-check expect status 200
default-server inter 3s fall 3 rise 2 on-marked-down shutdown-sessions
server <DB1 HOSTNAME> <DB1 IP>:5432 maxconn 100 check port 8008
server <DB2 HOSTNAME> <DB2 IP>:5432 maxconn 100 check port 8008
server <DB3 HOSTNAME> <DB3 IP>:5432 maxconn 100 check port 8008
```

```
listen standby
bind <CLUSTER_VIP>:5001
balance roundrobin
option httpchk OPTIONS /replica
http-check expect status 200
default-server inter 3s fall 3 rise 2 on-marked-down shutdown-sessions
server <DB1 HOSTNAME> <DB1 IP>:5432 maxconn 100 check port 8008
server <DB2 HOSTNAME> <DB2 IP>:5432 maxconn 100 check port 8008
server <DB3 HOSTNAME> <DB3 IP>:5432 maxconn 100 check port 8008
```

33. На «Сервере базы данных» – узел 2 (см. Таблица 1) провести настройку HAProxy.

```
sudo mv /etc/haproxy/haproxy.cfg /etc/haproxy/haproxy.cfg.orig
sudo nano /etc/haproxy/haproxy.cfg
```

Добавить следующую конфигурацию:

```
global
log 127.0.0.1 local2
log /dev/log local0
log /dev/log local1 notice
chroot /var/lib/haproxy
stats socket /run/haproxy/admin.sock mode 660 level admin expose-fd listeners
stats timeout 30s
user haproxy
group haproxy
maxconn 4000
daemon
```

```
defaults
mode tcp
log global
option tcplog
retries 3
timeout queue 3m
timeout connect 10s
```

```
timeout client      3m
timeout server      3m
timeout check       10s
maxconn             3000
```

```
listen stats
mode http
bind *:7000
stats enable
stats uri /
```

```
listen primary
bind <CLUSTER_VIP>:5000
option httpchk OPTIONS /master
http-check expect status 200
default-server inter 3s fall 3 rise 2 on-marked-down shutdown-sessions
server <DB1 HOSTNAME> <DB1 IP>:5432 maxconn 100 check port 8008
server <DB2 HOSTNAME> <DB2 IP>:5432 maxconn 100 check port 8008
server <DB3 HOSTNAME> <DB3 IP>:5432 maxconn 100 check port 8008
```

```
listen standby
bind <CLUSTER_VIP>:5001
balance roundrobin
option httpchk OPTIONS /replica
http-check expect status 200
default-server inter 3s fall 3 rise 2 on-marked-down shutdown-sessions
server <DB1 HOSTNAME> <DB1 IP>:5432 maxconn 100 check port 8008
server <DB2 HOSTNAME> <DB2 IP>:5432 maxconn 100 check port 8008
server <DB3 HOSTNAME> <DB3 IP>:5432 maxconn 100 check port 8008
```

34. На «Сервере базы данных» – узел 3 (см. Таблица 1) провести настройку HAProxy.

```
sudo mv /etc/haproxy/haproxy.cfg /etc/haproxy/haproxy.cfg.orig
sudo nano /etc/haproxy/haproxy.cfg
```

Добавить следующую конфигурацию:

```
global
log 127.0.0.1 local2
log /dev/log local0
log /dev/log local1 notice
chroot /var/lib/haproxy
stats socket /run/haproxy/admin.sock mode 660 level admin expose-fd listeners
stats timeout 30s
user haproxy
group haproxy
maxconn 4000
daemon
```

```
defaults
mode tcp
log global
option tcplog
retries 3
```

```

timeout queue      3m
timeout connect    10s
timeout client     3m
timeout server     3m
timeout check      10s
maxconn            3000

```

*listen stats*

```

mode http
bind *:7000
stats enable
stats uri /

```

*listen primary*

```

bind <CLUSTER_VIP>:5000
option httpchk OPTIONS /master
http-check expect status 200
default-server inter 3s fall 3 rise 2 on-marked-down shutdown-sessions
server <DB1 HOSTNAME> <DB1 IP>:5432 maxconn 100 check port 8008
server <DB2 HOSTNAME> <DB2 IP>:5432 maxconn 100 check port 8008
server <DB3 HOSTNAME> <DB3 IP>:5432 maxconn 100 check port 8008

```

*listen standby*

```

bind <CLUSTER_VIP>:5001
balance roundrobin
option httpchk OPTIONS /replica
http-check expect status 200
default-server inter 3s fall 3 rise 2 on-marked-down shutdown-sessions
server <DB1 HOSTNAME> <DB1 IP>:5432 maxconn 100 check port 8008
server <DB2 HOSTNAME> <DB2 IP>:5432 maxconn 100 check port 8008
server <DB3 HOSTNAME> <DB3 IP>:5432 maxconn 100 check port 8008

```

35. На каждом из узлов «Серверов базы данных» (см. Таблица 1) запустить HAProxy:

```
sudo systemctl start haproxy && sudo systemctl enable haproxy
```

36. Статус можно проверить при помощи команды

```
sudo systemctl status haproxy
```

Состояние должно быть active (running)

37. Отредактировать PostgreSQL конфигурацию в Patroni

```
sudo patronictl -c /etc/patroni/config.yml edit-config
```

Добавить следующую конфигурацию:

```

loop_wait: 10
maximum_lag_on_failover: 1048576
postgresql:
  parameters:
    checkpoint_completion_target: 0.9
    datestyle: iso, mdy
    default_statistics_target: 500
    effective_cache_size: 5968MB
    effective_io_concurrency: 100
    listen_addresses: '*'
    log_line_prefix: '%m [%p] %q%u@%d '

```

```
log_timezone: Europe/Moscow
maintenance_work_mem: 398MB
max_connections: 500
max_parallel_maintenance_workers: 2
max_parallel_workers: 4
max_parallel_workers_per_gather: 2
max_wal_size: 7957MB
max_worker_processes: 4
min_wal_size: 1989MB
password_encryption: md5
random_page_cost: 2
shared_buffers: 1989MB
timezone: Europe/Moscow
wal_buffers: 62MB
work_mem: 20MB
use_pg_rewind: true
use_slots: true
retry_timeout: 10
ttl: 30
```

Для применения настроек перезапустить кластер

```
sudo patronictl -c /etc/patroni/config.yml reload pg_cluster && sudo patronictl -c
/etc/patroni/config.yml restart pg_cluster
```

38. Подключиться к ранее созданным БД в кластере по виртуальному IP адресу можно при помощи следующих команд

```
psql -h <CLUSTER_VIP> -p 5000 quiz -d quiz
psql -h <CLUSTER_VIP> -p 5000 sync -d sync
psql -h <CLUSTER_VIP> -p 5000 administration -d administration
```

Подключение должно пройти успешно

39. Также можно проверить состояние кластера через интерфейс HAProxy [http://<CLUSTER\\_VIP>:7000/](http://<CLUSTER_VIP>:7000/). Должен быть доступен 1 primary и 2 standby узла.

Настройка сервера базы данных завершена.

### 3.3 Установка и настройка Docker CE

Приложение и часть инфраструктуры будут запущены в Docker CE, поэтому на «Сервере приложений – 1», «Сервере приложений – 2», «Сервере управления очередями – узел 1», «Сервере управления очередями – узел 2», «Сервере управления очередями – узел 3», «Сервере синхронизации – узел 1», «Сервере синхронизации – узел 2», «Сервере синхронизации – узел 3» (см. Таблица 1) должен быть установлен Docker CE и Docker-compose.

1. В случае отсутствия установите Docker CE согласно инструкции с официального сайта для Вашей операционной системы, ниже ссылка для Ubuntu: <https://docs.docker.com/engine/install/ubuntu/>

2. Добавьте пользователя, под которым происходит установка в группу docker:

```
$ sudo usermod -a -G docker $USER
```

Выйдите из системы и затем снова войти в систему для того, чтобы активировать принадлежность к группе.

3. Проверьте корректность установки запуском следующей команды:

```
docker --version
```

На экран должен быть выведен номер установленной версии

4. Скачайте и установите docker-compose

```
sudo curl -L "https://github.com/docker/compose/releases/download/v2.20.2/docker-
```

*compose-\$(uname -s)-\$(uname -m)" -o /usr/local/bin/docker-compose*

5. Добавьте необходимые права

*sudo chmod +x /usr/local/bin/docker-compose*

6. Проверьте корректность установки запуском следующей команды:

*docker-compose --version*

На экран должен быть выведен номер установленной версии

### 3.4 Установка и настройка сервера управления очередями

В данном разделе описаны условия и действия, необходимые для установки сервера управления очередями. Все действия нужно произвести на каждом из узлов «Сервер управления очередями – узел 1», «Сервер управления очередями – узел 2», «Сервер управления очередями – узел 3 (см. Таблица 1), если не указано иное.

Приложение будет запущено в Docker CE, поэтому на «Сервере управления очередями – узел 1», «Сервере управления очередями – узел 2», «Сервере управления очередями – узел 3 (см. Таблица 1) должен быть установлен Docker CE и Docker-compose. Описание установки описано в разделе 3.3

1. Скопируйте содержимое установочного диска в папку /tmp/rsk

2. Создайте каталоги, в которых будут размещены файлы сервиса управления очередями

*sudo mkdir /opt/rabbitmq/ && sudo chown -R \$USER:\$USER /opt/rabbitmq/*

3. Скопируйте файлы *docker-compose.yml* и *rabbitmq\_delayed\_message\_exchange-3.11.1.ez* с установочного диска из папки *deploy/rabbitmq* в каталог */opt/rabbitmq/* созданный ранее

4. В файле *docker-compose.yml* подставьте нужные значения *<RABBIT\_USER>* и *<RABBIT\_PASSWORD>* для доступа к серверу управления очередями

5. Перейдите в ранее созданный каталог */opt/rabbitmq* и запустите команду:

*docker-compose up -d*

6. Проверьте по логам контейнера, что его запуск прошел успешно и подключение к базе данных произошло. Для этого запустите команду:

*docker logs quiz\_rabbitmq*

7. После успешного выполнения п.1 – п.7 на всех узлах сервера управления очередями нужно объединить их в кластер. На узлах должны быть выставлены одинаковые временные зоны, определены имена и по этим именам они должны быть доступны друг другу.

8. Скопируйте значение из файла *.erlang.cookie* из контейнера на «Сервере управления очередями – узел 1» в контейнеры на «Сервер управления очередями – узел 2», «Сервер управления очередями – узел 3»

*docker exec -it quiz\_rabbitmq cat /var/lib/rabbitmq/.erlang.cookie* – посмотреть значение

*docker exec -it quiz\_rabbitmq bash* – зайти внутрь контейнера и отредактировать *nano /var/lib/rabbitmq/.erlang.cookie*

9. На «Сервер управления очередями – узел 1» выполните команды

*docker exec -it quiz\_rabbitmq rabbitmqctl stop\_app*

*docker exec -it quiz\_rabbitmq rabbitmqctl join\_cluster --disc rabbit@<SYNC1 HOSTNAME>*

*docker exec -it quiz\_rabbitmq rabbitmqctl start\_app*

10. На «Сервер управления очередями – узел 2» выполните команды

*docker exec -it quiz\_rabbitmq rabbitmqctl stop\_app*

*docker exec -it quiz\_rabbitmq rabbitmqctl join\_cluster --disc rabbit@<SYNC2 HOSTNAME>*

*docker exec -it quiz\_rabbitmq rabbitmqctl start\_app*



11. На «Сервере управления очередями – узел 3» выполните команды  
`docker exec -it quiz_rabbitmq rabbitmqctl stop_app`  
`docker exec -it quiz_rabbitmq rabbitmqctl join_cluster --disc rabbit@<SYNC3 HOSTNAME>`  
`docker exec -it quiz_rabbitmq rabbitmqctl start_app`
12. Проверьте видимость всех трех узлов, выполнив команду на любом из них:  
`docker exec -it quiz_rabbitmq rabbitmqctl cluster_status`  
 Должны быть видимы все три узла
13. Добавьте необходимые настройки репликации, выполнив команду на любом из узлов:  
`docker exec -it quiz_rabbitmq rabbitmqctl set_policy ha-all "" '{"ha-mode":"all","ha-sync-mode":"automatic"}'`
14. Создайте каталоги, в которых будут размещены файлы сервиса кэширования  
`sudo mkdir /opt/redis/ && sudo chown -R $USER:$USER /opt/redis/ && sudo mkdir /opt/redis/config`
15. Скопируйте файл `docker-compose.yml` с установочного диска из папки `deploy/redis` в каталог `/opt/redis/` созданный ранее
16. На «Сервере управления очередями – узел 1» скопируйте файл `config/redis-master.conf` с установочного диска из папки `deploy/redis` в `/opt/redis/config/redis.conf` созданный ранее (обратите внимание, что имя файла меняется)
17. На «Сервере управления очередями – узел 1» в файле `/opt/redis/config/redis.conf` подставьте нужное значение `<REDIS_PASSWORD>` для доступа к серверу кэширования
18. На «Сервере управления очередями – узел 2» и «Сервере управления очередями – узел 3» скопируйте файл `config/redis-slave.conf` с установочного диска из папки `deploy/redis` в `/opt/redis/config/redis.conf` созданный ранее (обратите внимание, что имя файла меняется)
19. На «Сервере управления очередями – узел 2» и «Сервере управления очередями – узел 3» в файле `/opt/redis/config/redis.conf` подставьте нужные значения `<REDIS_PASSWORD>` для доступа к серверу кэширования и `<MASTER_HOST>` - IP адрес «Сервера управления очередями – узел 1», который зарегистрирован в качестве мастера
20. Перейдите в ранее созданный каталог `/opt/redis` и запустите команду:  
`docker-compose up -d`
21. Проверьте по логам контейнера, что его запуск прошел успешно и подключение к базе данных произошло. Для этого запустите команду:  
`docker logs quiz_redis`  
 В логе должно быть сообщение, о том, что сервер запущен и не должно быть ошибок.  
 Настройка сервера управления очередями завершена

### 3.5 Установка и настройка сервера приложений

В данном разделе описаны условия и действия, необходимые для установки серверной части Веб-приложения.

Приложение будет запущено в Docker CE, поэтому на «Сервере приложений – 1» и «Сервере приложений – 2» (см. Таблица 1) должен быть установлен Docker CE и Docker-compose. Описание установки описано в разделе 3.3

1. Скопируйте содержимое установочного диска в папку `/tmp/rsk` на «Сервер приложений – 1» и «Сервер приложений – 2» (см. Таблица 1)

2. На «Сервере приложений – 1» (см. Таблица 1) создайте каталоги, в которых будут размещены файлы приложения

`sudo mkdir /opt/administration-service/ && sudo chown -R $USER:$USER /opt/administration-service/ && mkdir /opt/administration-service/service`

Руководство по инсталляции	Номер страницы: 23 Версия 1.0
----------------------------	----------------------------------

3. Скопируйте файл *docker-compose.yml* с установочного диска из папки *deploy/backend* в каталог */opt/administration-service/* созданный ранее на «Сервере приложений – 1» (см. Таблица 1).

4. Скопируйте файл *administration-service.jar* с установочного диска из папки *dist/backend* в каталог */opt/administration-service/service* созданный ранее на «Сервере приложений – 1» (см. Таблица 1).

5. На «Сервере приложений – 1» (см. Таблица 1) перейдите в ранее созданный каталог */opt/administration-service* и запустите команду:

```
docker-compose up -d
```

6. Проверьте по логам контейнера, что его запуск прошел успешно и подключение к базе данных произошло. Для этого запустите команду:

```
docker logs administration-service
```

7. На «Сервере приложений – 2» (см. Таблица 1) создайте каталоги, в которых будут размещены файлы приложения

```
8. sudo mkdir /opt/quiz-service/ && sudo chown -R $USER:$USER /opt/quiz-service/ && mkdir /opt/quiz-service/service
```

9. Скопируйте файл *docker-compose.yml* с установочного диска из папки *deploy/backend* в каталог */opt/quiz-service/* созданный ранее на «Сервере приложений – 2» (см. Таблица 1).

10. Скопируйте файл *quiz-service.jar* с установочного диска из папки *dist/backend* в каталог */opt/quiz-service/service* созданный ранее на «Сервере приложений – 2» (см. Таблица 1).

11. На «Сервере приложений – 2» (см. Таблица 1) перейдите в ранее созданный каталог */opt/quiz-service* и запустите команду:

```
docker-compose up -d
```

12. Проверьте по логам контейнера, что его запуск прошел успешно и подключение к базе данных произошло. Для этого запустите команду:

```
docker logs quiz-service
```

Если ошибок нет, то настройка серверов приложений завершена.

### 3.6 Установка и настройка сервера синхронизации

В данном разделе описаны условия и действия, необходимые для установки серверной части Веб-приложения, обеспечивающей синхронизацию с мобильными устройствами.

Приложение будет запущено в Docker CE, поэтому на «Сервере синхронизации» (см. Таблица 1) должен быть установлен Docker CE и Docker-compose. Описание установки приведено в разделе 3.3.

1. Скопируйте содержимое установочного диска в папку */tmp/rsk* на «Сервер синхронизации» (см. Таблица 1)

2. На «Сервере синхронизации» (см. Таблица 1) создайте каталоги, в которых будут размещены файлы приложения

```
sudo mkdir /opt/synchronization-service/ && sudo chown -R $USER:$USER /opt/synchronization-service/ && mkdir /opt/synchronization-service/config
```

3. Скопируйте файл *docker-compose.yml* с установочного диска из папки *deploy/mobile-sync* в каталог */opt/synchronization-service/* созданный ранее на «Сервере синхронизации» (см. Таблица 1).

4. Скопируйте файл *application-template.yml* с установочного диска из папки *config/mobile-sync* в каталог */opt/synchronization-service/config/* созданный ранее на «Сервере синхронизации» (см. Таблица 1), и переименуйте файл в *application.yml*.

5. Отредактируйте файл */opt/synchronization-service/config/application.yml* в соответствии с окружением, внесите следующие данные:

Руководство по установке	Номер страницы: 24
	Версия 1.0



<POSTGRESQL HOST> – адрес Сервера базы данных – следует указать виртуальный IP адрес кластера - <CLUSTER\_VIP> (см. раздел 3.2)

<POSTGRESQL PORT> – порт Сервера базы данных (см. раздел 3.2)

<POSTGRESQL\_USERNAME> – пользователь, созданный для синхронизации, обычно sync (см. раздел 3.2)

<POSTGRESQL PASSWORD> – пароль пользователя, созданного для синхронизации (см. раздел 3.2)

<RABBITMQ HOST> – адрес Сервера управления очередями – следует указать IP всех узлов в формате <RABBITMQ1 IP>:5672,<RABBITMQ2 IP>:5672,<RABBITMQ3 IP>:5672 (см. раздел 3.4)

<RABBITMQ PASSWORD> – пароль пользователя Сервера управления очередями (см. раздел 3.4 Установка и настройка сервера управления очередями)

<ADMINISTRATION HOST> – IP адрес Сервера приложений – 1 (см. раздел 3.5)

6. На «Сервере синхронизации» (см. Таблица 1) перейдите в ранее созданный каталог */opt/synchronization-service* и запустите команду:

*docker-compose up -d*

7. Проверьте по логам контейнера, что его запуск прошел успешно и подключение к базе данных произошло. Для этого запустите команду:

*docker logs synchronization-service*

Если ошибок нет, то настройка сервера синхронизации завершена.

### 3.7 Установка и настройка сервера интерфейсов управления

В данном разделе описаны условия и действия, необходимые для установки интерфейсной части Веб-приложения.

Приложение будет запущено в Docker CE, поэтому на «Сервере управления очередями – узел 1», «Сервере управления очередями – узел 2», «Сервере управления очередями – узел 3 (см. Таблица 1) должен быть установлен Docker CE и Docker-compose. Описание установки описано в разделе 3.3:

1. Создайте каталоги на «Сервере интерфейсов управления» (см. Таблица 1) в которых будут размещены файлы приложения

*sudo mkdir /opt/nginx/ && sudo chown -R \$USER:\$USER /opt/nginx*

*sudo mkdir /opt/quiz-front && sudo chown -R \$USER:\$USER /opt/quiz-front*

*sudo mkdir /opt/admin-front && sudo chown -R \$USER:\$USER /opt/admin-front*

*sudo mkdir /opt/quiz-employee-front && sudo chown -R \$USER:\$USER /opt/quiz-employee-front*

*sudo mkdir /opt/m-quiz-employee-front && sudo chown -R \$USER:\$USER /opt/m-quiz-employee-front*

2. Скопируйте файл *docker-compose.yml* с установочного диска из папки *deploy/frontend* в каталог */opt/nginx/* созданный ранее на «Сервере интерфейсов управления» (см. Таблица 1).

3. Создайте директорию */opt/nginx/config/conf.d/* на «Сервере интерфейсов управления» (см. Таблица 1) и скопируйте в нее все файлы с установочного диска из папки *config/frontend*. Обратите внимание, что нужно заменить значения **<FRONTEND1 HOSTNAME>**, **<FRONTEND2 HOSTNAME>**, **<FRONTEND3 HOSTNAME>**, **<FRONTEND4 HOSTNAME>**, **<BACKEND1 IP>**, **<BACKEND2 IP>** и **<BACKEND3 IP>** в соответствии с окружением:

**<FRONTEND1 HOSTNAME>** - доменное имя «Сервера интерфейсов управления» (см. Таблица 1) для Системы «Работа с персоналом» (предварительно, если еще не сделано, в DNS должна быть создана запись типа А соотносящая доменное имя и IP адрес «Сервера интерфейсов управления» (см. Таблица 1)).

<**FRONTEND2 HOSTNAME**> - доменное имя «Сервера интерфейсов управления» (см. Таблица 1) для Системы "Работа с персоналом. Администрирование" (предварительно, если еще не сделано, в DNS должна быть создана запись типа А соотносящая доменное имя и IP адрес «Сервера интерфейсов управления» (см. Таблица 1)).

<**FRONTEND3 HOSTNAME**> - доменное имя «Сервера интерфейсов управления» (см. Таблица 1) для Системы "Работа с персоналом. ЛК сотрудника." (предварительно, если еще не сделано, в DNS должна быть создана запись типа А соотносящая доменное имя и IP адрес «Сервера интерфейсов управления» (см. Таблица 1)).

<**FRONTEND4 HOSTNAME**> - доменное имя «Сервера интерфейсов управления» (см. Таблица 1) для Системы "Работа с персоналом. ЛК сотрудника. Мобильная версия" (предварительно, если еще не сделано, в DNS должна быть создана запись типа А соотносящая доменное имя и IP адрес «Сервера интерфейсов управления» (см. Таблица 1)).

<**BACKEND1 IP**> IP-адрес «Сервера приложений – 1» (см. Таблица 1)

<**BACKEND2 IP**> IP-адрес «Сервера приложений – 2» (см. Таблица 1)

<**BACKEND3 IP**> IP-адрес «Сервера синхронизации» (см. Таблица 1)

4. Скопируйте содержимое установочного диска в папку /tmp/quiz на «Сервере интерфейсов управления» (см. Таблица 1) и распакуйте при помощи следующей команды:

```
unzip /tmp/quiz/dist/frontend/employee-quiz.zip -d /opt/quiz-employee-front
unzip /tmp/quiz/dist/frontend/m-employee-quiz.zip -d /opt/m-quiz-employee-front
unzip /tmp/quiz/dist/frontend/quiz.zip -d /opt/quiz-front
unzip /tmp/quiz/dist/frontend/admin.zip -d /opt/quiz-admin
```

5. На «Сервере интерфейсов управления» (см. Таблица 1) скопируйте выпущенные SSL сертификаты с ключами в каталог /opt/nginx/config/ssl. Настройки путей задаются в конфигурационных файлах, откопированных в п.3

6. На «Сервере интерфейсов управления» (см. Таблица 1) перейдите в ранее созданный каталог /opt/nginx/ и запустите команду:

```
docker-compose up -d
```

7. Проверьте по логам контейнера Nginx, что его запуск прошел успешно. Для этого запустите команду:

```
docker logs nginx
```

Если ошибок нет, то настройка сервера интерфейсов управления завершена.

### 3.8 Установка и настройка сервера хранения

В данном разделе описаны условия и действия, необходимые для установки сервера хранения. Все действия нужно произвести на каждом из узлов «Сервер хранения – узел 1», «Сервер хранения – узел 2», «Сервер хранения – узел 3 (см. Таблица 1), если не указано иное. На всех узлах должны быть выставлены одинаковые временные зоны и время должно быть синхронизировано.

1. На каждом узле создать каталог /glusterfs/quiz:

```
sudo mkdir /glusterfs/quiz
```

2. На каждом узле выполнить установку GlusterFS и активировать сервер.

```
sudo apt-get install software-properties-common
sudo add-apt-repository -y ppa:gluster/glusterfs-7
sudo apt-get update
sudo apt-get install -y glusterfs-server
sudo systemctl enable glusterd
```

3. Для добавления узлов в пул узлов необходимо на любом из узлов выполнить команду:

```
sudo gluster peer probe <STORAGE1 IP>
```

```
sudo gluster peer probe <STORAGE2 IP>
```

```
sudo gluster peer probe <STORAGE3 IP>
```

где <STORAGE1 IP>, <STORAGE2 IP>, <STORAGE3 IP> - IP адрес соответствующего узла «Сервера хранения (см. Таблица 1)

4. Посмотреть добавленные в пул узлы можно командой:

```
sudo gluster pool list
```

Вывод должен быть примерно такой:

UUID	Hostname	State
c4beb94d-aca6-4a45-9d44-398e26fcba17	<STORAGE2 IP>	Connected
15e41d8c-d1d1-4329-9e48-997760816ff5	<STORAGE3 IP>	Connected
e8b71129-6d7c-4bd5-b032-198bce08915e	localhost	Connected

5. Создать том для хранения данных:

```
sudo gluster volume create quiz replica 3 server{1,2,3}.private:/glusterfs/quiz force
```

6. Убедиться в том, что том создан можно с помощью команды:

```
sudo gluster volume list
```

Вывод должен быть такой (имя было задано в п.5):

```
quiz
```

7. Для активации тома его необходимо запустить:

```
sudo gluster volume start quiz
```

8. Теперь можно запросить состояние тома:

```
sudo gluster volume status quiz
```

Вывод должен быть примерно такой:

```
Status of volume: quiz
```

Gluster process	TCP Port	RDMA Port	Online	Pid
Brick <STORAGE 1 IP>:/glusterfs/quiz	49152	0	Y	1747
Brick <STORAGE2 IP>:/glusterfs/quiz	49152	0	Y	1122
Brick <STORAGE3 IP>:/glusterfs/quiz	49152	0	Y	12872
Self-heal Daemon on localhost	N/A	N/A	Y	1758
Self-heal Daemon on <STORAGE2 IP>	N/A	N/A	Y	1137
Self-heal Daemon on <STORAGE3 IP>	N/A	N/A	Y	14948

```
Task Status of Volume quiz
```

```
There are no active volume tasks
```

9. Информацию о томе можно посмотреть следующей командой:

```
sudo gluster volume info quiz
```

Вывод должен быть примерно такой:

```
Volume Name: quiz
```

```
Type: Replicate
```

```
Volume ID: 28c0cafa-0ba5-43c8-8b24-45527bbd3530
```

```
Status: Started
```

```
Snapshot Count: 0
```

```
Number of Bricks: 1 x 3 = 3
```

```
Transport-type: tcp
```

```
Bricks:
```

```
Brick1: <STORAGE1 IP>:/glusterfs/quiz
```

```
Brick2: <STORAGE2 IP>:/glusterfs/quiz
```

```
Brick3: <STORAGE3 IP>:/glusterfs/quiz
```

```
Options Reconfigured:
```

```
transport.address-family: inet
```

```
storage.fips-mode-rchecksum: on
```

```
nfs.disable: on
```

```
performance.client-io-threads: off
```

10. Теперь на серверах «Сервер синхронизации» и «Сервер приложений 2» (см. Таблица 1) необходимо установить клиентскую часть GlusterFS. Для этого на каждом сервере создать каталог /storage:

```
sudo mkdir /storage
```

11. На каждом сервере «Сервер синхронизации» и «Сервер приложений 2» (см. Таблица 1) выполнить установку GlusterFS клиента:

```
sudo apt-get install software-properties-common
sudo add-apt-repository -y ppa:gluster/glusterfs-7
sudo apt-get update
sudo apt-get install -y glusterfs-client
```

12. Смонтировать реплицируемый том  
`sudo mount -t glusterfs <STORAGE1 IP>,<STORAGE2 IP>,<STORAGE3 IP>:quiz /storage`

где <STORAGE1 IP>, <STORAGE2 IP>, <STORAGE3 IP> - IP адрес соответствующего узла «Сервера хранения» (см. Таблица 1)

13. Проверить доступность смонтированного каталога на каждом сервере «Сервер синхронизации» и «Сервер приложений 2» (см. Таблица 1)

```
ls -al /storage
```

Каталог должен быть доступен и при создании файлов на одном из серверов, они должны быть доступны и на другом.

Настройка сервера хранения завершена.

### 3.9 Установка и настройка сервера мониторинга

В данном разделе описаны условия и действия, необходимые для установки сервера мониторинга.

Сервисы мониторинга будут запущены в Docker CE, поэтому на «Сервере мониторинга» (см. Таблица 1) должен быть установлен Docker CE и Docker-compose. Описание установки описано в разделе 3.3.

1. Создать каталог на «Сервере мониторинга» (см. Таблица 1), в котором будут размещены файлы мониторинга

```
sudo mkdir /opt/monitoring/ && sudo chown -R $USER:$USER /opt/monitoring
```

2. Откопировать все файлы и папки с установочного диска из папки deploy/monitoring в каталог /opt/monitoring/ созданный ранее на «Сервере мониторинга» (см. Таблица 1).

3. На «Сервере мониторинга» (см. Таблица 1) перейти в ранее созданный каталог /opt/monitoring/ и запустить команду:

```
docker-compose up -d
```

4. Проверить по логам контейнеров, что запуск прошел успешно. Для этого запустить команды:

```
docker logs grafana
docker logs loki
docker logs blackboxexporter
docker logs cadvisor
docker logs nodeexporter
docker logs prometheus
docker logs alertmanager
```

5. Для сбора логов с приложений на каждом из серверов «Сервер синхронизации», «Сервер приложений 1» и «Сервер приложений 2» (см. Таблица 1) создать каталог, в котором будут размещены файлы мониторинга

```
sudo mkdir /opt/monitoring/ && sudo chown -R $USER:$USER /opt/monitoring
```

6. Откопировать все файлы и папки с установочного диска из папки `deploy/promtail` в каталог `/opt/monitoring/` созданный ранее на каждом из серверов «Сервер синхронизации», «Сервер приложений 1» и «Сервер приложений 2» (см. Таблица 1).

7. На каждом из серверов «Сервер синхронизации», «Сервер приложений 1» и «Сервер приложений 2» (см. Таблица 1) перейти в ранее созданный каталог `/opt/monitoring/` и запустить команду:

```
docker-compose up -d
```

8. Проверить по логам контейнеров, что запуск прошел успешно. Для этого запустить команду:

```
docker logs promtail
```

Если ошибок нет, то настройка сервера мониторинга завершена.

#### **4. Проверка настройки приложения**

Ввести в адресной строке веб-браузера адрес Системы:

```
https://<FRONTEND1 HOSTNAME>
```

```
https://<FRONTEND2 HOSTNAME>
```

```
https://<FRONTEND3 HOSTNAME>
```

```
https://<FRONTEND4 HOSTNAME>
```

**<FRONTEND1 HOSTNAME>**, **<FRONTEND2 HOSTNAME>**, **<FRONTEND3 HOSTNAME>**, **<FRONTEND4 HOSTNAME>** – доменное имя сервера управления интерфейсов для соответствующей подсистемы, ранее прописанное в настройках Nginx. Для мобильной версии проверка должна осуществляться с мобильного устройства.

При успешном выполнении установки должен отобразиться интерфейс Системы Работа с Персоналом.