Программа «Электронная очередь»

**Руководство системного программиста**

**Листов \_12**

Инв. № подл.

Подп. и дата

Взам. инв. №

Инв. № дубл.

Подп. и дата

2023

**АННОТАЦИЯ**

Документ содержит описание структуры, процедуры установки, и проверки программы «Электронная очередь» и предназначен для системных программистов, осуществляющих подготовку программы к эксплуатации.

Настоящий документ предназначен для персонала, имеющего квалификацию системного администратора, обладающего знаниями и навыками в области системного администрирования, в том числе установки ОС Linux alpine 3.10, работы с СУБД PostgreSQL, Redis, создания и настройки кластеров Kubernetes.

Документ не является руководством пользователя по установке и настройке, операционной системы, и прочего системного и технологического программного обеспечения.

**Содержание**

[1. ОБЩИЕ СВЕДЕНИЯ О ПРОГРАММЕ 4](#_Toc133331122)

[2. СТРУКТУРА ПРОГРАММЫ 4](#_Toc133331123)

[3. УСТАНОВКА ПРОГРАММЫ 5](#_Toc133331124)

[3.1. Установка СУБД 5](#_Toc133331125)

[3.2. Установка сервисов в кластере Kubernetes. 6](#_Toc133331126)

[3.2.1. Настройка Kubernetes. 6](#_Toc133331127)

[3.2.2. Размещение сервисов в кластере Kubernetes. 9](#_Toc133331128)

[4. ПРОВЕРКА ПРОГРАММЫ 10](#_Toc133331129)

[5. ДОПОЛНИТЕЛЬНЫЕ ВОЗМОЖНОСТИ 11](#_Toc133331130)

[6. ПЕРЕЧЕНЬ СОКРАЩЕНИЙ 12](#_Toc133331131)

# 1. ОБЩИЕ СВЕДЕНИЯ О ПРОГРАММЕ

* 1. Программа «Электронная очередь» обеспечивает выполнение функций, описанных в документах:

-«Программа Электронная очередь. Руководство пользователя. Книга 1. Сервис предварительной записи».

-«Программа Электронная очередь. Руководство пользователя. Книга 2. Кабинет сотрудника».

Состав программы приведен в п. 2 настоящего документа.

* 1. Для обеспечения выполнения программы необходимы следующие технические средства:

Аппаратно-программные ресурсы выделенного сервера:

Операционная система: Linux alpine 3.10.

Используемая СУБД: PostgreSQL 15, Redis.

Развёрнутый Kubernetes-кластер.

# 2. СТРУКТУРА ПРОГРАММЫ

Структура программы «Электронная очередь» представлена в таблице 1.

Таблица 1.

|  |  |
| --- | --- |
| Название | Наименование |
| **Queue FRONT** | Программный сервис |
| **Queue Admin** | Программный сервис |
| **Queue API** | Программный сервис |

Вышеуказанные сервисы имеют следующее назначение:

**-Queue FRONT-**реализует пользовательский интерфейс, для сервиса предварительной записи.

**-Queue Admin-**реализует пользовательский интерфейс личного кабинета для сотрудников исправительных учреждений.

-**Queue API-**реализует основную логику сервиса предварительной записи и личного кабинета. Взаимодействует с сервисами **Queue FRONT, Queue Admin.** Обеспечивает взаимодействие с базой данных, ЕСИА, а также внешними сервисами для синхронизации списка учреждений, и отправки сообщений на электронную почту.

# УСТАНОВКА ПРОГРАММЫ

Все действия для установки компонентов программы выполняются от пользователя с правами «root».

## 3.1. Установка СУБД

Для функционирования программы необходимо установить СУБД PostgreSQL и Redis.

Для установки PostgreSQL необходимо выполнить:

**sudo apt install postgresql**

**sudo systemctl start postgresql**

**sudo systemctl enable postgresql**

**sudo -u postgres psql -c "CREATE USER username WITH PASSWORD 'qwerty';"**

**sudo -u postgres psql -c "CREATE DATABASE dbname OWNER username;"**

**sudo -u postgres psql -c "GRANT.......;"**

**sudo vi /etc/postgresql/<version>/main/pg\_hba.conf.**

Для установки базы Redis необходимо выполнить:

**sudo apt install lsb-release**

**curl -fsSL https://packages.redis.io/gpg | sudo gpg --dearmor -o /usr/share/keyrings/redis-archive-keyring.gpg**

**echo "deb [signed-by=/usr/share/keyrings/redis-archive-keyring.gpg] https://packages.redis.io/deb $(lsb\_release -cs) main" | sudo tee /etc/apt/sources.list.d/redis.list**

**sudo apt-get update**

**sudo apt-get install redis**

## 3.2. Установка сервисов в кластере Kubernetes.

### 3.2.1. Настройка Kubernetes.

На начальном этапе необходимо создать «мастер»-виртуальную машину и ноды. Далее необходимо установить ОС Linux alpine 3.10, sudo, docker и kubernetes, используя официальную документацию(<https://kubernetes.io/docs/setup/>).

Далее необходимо произвести настройку, для этого на мастере инициализируем kubeadm панель:

**# sudo kubeadm init --pod-network-cidr=10.244.0.0/16.**

cidr выставляется для конкретного плагина, в нашем случае это WeaveNet (взять его можно из официальной документации).

Затем необходимо выполнить:

**# mkdir -p $HOME/.kube**

**# sudo cp -i /etc/kubernetes/admin.conf $HOME/.kube/config**

**# sudo chown $(id -u):$(id -g) $HOME/.kube/config**

После этого нам нужно установить сетевой плагин:

**#kubectl apply -f "**[**https://cloud.weave.works/k8s/net?k8s-version=$**](https://cloud.weave.works/k8s/net?k8s-version=$)**(kubectl version | base64 | tr -d '\n')"**

Далее **каждую ноду** необходимо добавить в кластер. Пример команды (Для каждого кластера она разная, в целом Kubernetes при выполнении предыдущего пункта пишет ее в конце):

**# sudo kubeadm join 172.20.70.210:6443 --token ssot6s.vy0sld2u7kck5v1c --discovery-token-ca-cert-hash sha256:6b2ae90cb485e9f2d89c4b9a62e3e17fe57fc926652e681ce5aa917b68dffa3f**

На следующем шаге необходимо создать токен для Kubernetes dashboard, для этого нужно создать YAML-файл по подобию:

apiVersion: [rbac.authorization.k8s.io/v1](http://rbac.authorization.k8s.io/v1)  
kind:ClusterRoleBinding  
metadata:  
name:admin-user  
roleRef:  
apiGroup: [rbac.authorization.k8s.io](http://rbac.authorization.k8s.io/)  
kind:ClusterRole  
name:cluster-admin  
subjects:

kind:ServiceAccount  
name:admin-user  
namespace: kube-system

В трех местах, где прописано «admin-user» изменить запись на имя пользователя.

После того как скорректировали, выполнить:

**# kubectl apply -f 'имя\_файла'**

Далее выполнить:

**# kubectl -n kube-system get secret | grep admin-user**

(admin-user заменить на имя пользователя, которое указывали в YAML-файле).

**# kubectl -n kube-system describe secret admin-user-token-<id displayed by previous command>** (здесь указываем результат предыдущей команды). После этого в консоли отобразится токен и можно пробовать с ним заходить на dashboard.

Запустить kubectl proxy:

**# kubectl proxy.**

Далее необходимо перейти по ссылке <http://localhost:8001/api/v1/namespaces/kube-system/services/https:kubernetes-dashboard:/proxy/>.

В появившемся окне загрузить YAML-файл, как показано на рисунке 1.

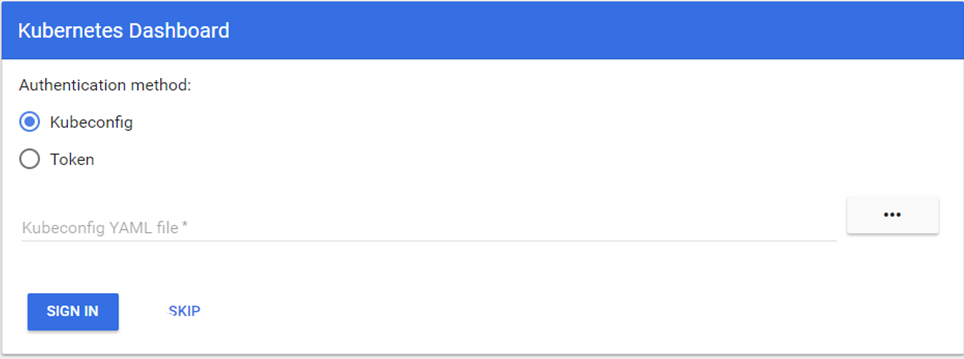


Рисунок 1 Внешний вид окна Dashboard

После этого выбрать вкладку «Token», подгрузить полученный токен и нажать «SIGN IN», после чего откроется окно Kubernetes Dashboard с правами суперадмина.

Далее необходимо установить пакетный менеджер Helm для Kubernetes.

Для этого следует скачать с официального сайта <https://github.com/helm/helm/releases> нужную версию и распаковать её, выполнив команду :

**# tar -zxvf helm-v3.0.0-linux-amd64.tar.gz**

Затем переместить бинарный файл в локацию, указанную в $PATH операционной системы:

**# mv linux-amd64/helm /usr/local/bin/helm**

### 3.2.2. Размещение сервисов в кластере Kubernetes.

Для первоначального размещения сервиса **«queue-admin»** необходимо выполнить:

**# git clone https://git.zonatelecom.ru/zonatelecom/fsin-queue/queue-admin.git**

**# cd queue-admin**

**# helm install queue-admin--set image.tag=тег\_желаемой\_версии ./**

Для первоначального размещения сервиса **«queue-front»** необходимо выполнить:

**# git clone https://git.zonatelecom.ru/zonatelecom/fsin-queue/queue-front.git**

**# cd queue-front**

**# helm install queue-front --set image.tag=тег\_желаемой\_версии ./**

Для первоначального размещения сервиса **«queue-api»** необходимо выполнить:

**# git clone https://git.zonatelecom.ru/zonatelecom/fsin-queue/queue-api.git**

**# cd queue-api**

**# helm install queue-api --set image.tag=тег\_желаемой\_версии./**

После успешного размещения сервисов в Kubernetes, сервисы должны отображаться с зелёным значком. Пример показан на рисунке 2.

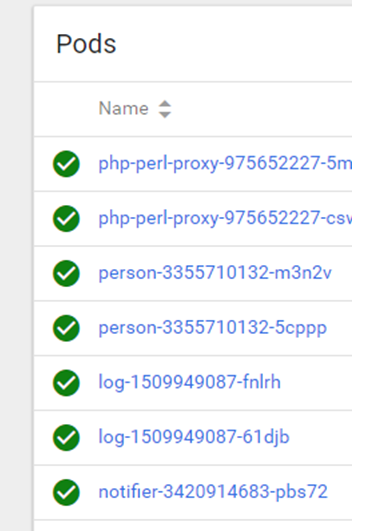


Рисунок 2. Окно отображения Pods

# ПРОВЕРКА ПРОГРАММЫ

Для проверки программы необходимо выполнить действия, описанные в документах:

-«Программа Электронная очередь. Руководство оператора Книга 1. Сервис предварительной записи».

-«Программа Электронная очередь. Руководство оператора. Книга 2. Кабинет сотрудника».

В случае корректного выполнения всех действий, установка программы считается завершённой успешно.

# 5. ДОПОЛНИТЕЛЬНЫЕ ВОЗМОЖНОСТИ

Дополнительные возможности, отличные от приведённых в документах «Программа Электронная очередь. Руководство оператора. Книга 1. Сервис предварительной записи», «Программа Электронная очередь. Руководство оператора. Книга 2. Кабинет сотрудника» не предусмотрены и могут быть реализованы разработчиком на основании анализа потребностей пользователей.

# 6. ПЕРЕЧЕНЬ СОКРАЩЕНИЙ

ОС-операционная система

СУБД-система управления базами данных

ЕСИА-единая система идентификации и авторизации

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Лист регистрации изменений | | | | | | | | | |
| Номера листов (страниц) | | | | | Всего листов (страниц)  в докум. | №  документа | Входящий № сопроводительного документа и дата | Подп. | Дата |
| Изм. | изменен-ных | заменен-ных | новых | аннули-рован-ных |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |