**Отключить SELinux**

setenforce 0

vim /etc/selinux/config

SELINUX=permissive

**Отключить firewalld**

systemctl stop firewalld.service

systemctl disable firewalld.service

**Приложение**

Устанавливаем git и клонируем репозиторий с приложением

dnf install -y git

git clone https://github.com/dockersamples/example-voting-app

cd example-voting-app

В приложении сервисы обращаются к redis и postgres по именам, поэтому нужно добавить их в hosts.

vim /etc/hosts

127.0.0.1 redis db localhost localhost.localdomain localhost4 localhost4.localdomain4

::1 localhost localhost.localdomain localhost6 localhost6.localdomain6

**Redis**

Устанавливаем Redis

dnf install -y epel-release

dnf install -y redis

Нужно убедиться, что redis слушает на 127.0.0.1

vim /etc/redis.conf

bind 127.0.0.1 # Uncomment

Запускаем redis и добавляем в автозагрузку

systemctl enable --now redis.service

**Postgres**

dnf install -y postgresql-server

postgresql-setup --initdb

systemctl enable --now postgresql.service

Подключаемся к postgres и ставим пароль для пользователя 'postgres'

sudo -u postgres psql

\du

ALTER USER postgres WITH PASSWORD 'postgres';

\l - список баз

\du - список пользователей

\c postgres - подключиться к базе 'postgres'

\dt \* - список таблиц

Разрешаем вход по паролю

vim /var/lib/pgsql/data/pg\_hba.conf

host all all 127.0.0.1/32 md5

Перезапускаем postgres

systemctl restart postgres.service

**Voting service**

Переходим в директорию vote

cd vote

Устанавливаем python, virtualenv и pip

dnf install python3 python3-virtualenv python3-pip

Создаём виртуальное окружение и устанавливаем зависимости сервиса

python3 -m venv .venv

source .venv/bin/activate

pip3 install -r requirements.txt

Запускаем и убеждаемся, что всё работает

gunicorn app:app -b 0.0.0.0:3000 --log-file - --access-logfile - --workers 4 --keep-alive 0

Проверяем, добавилась ли запись о нашем голосовании

#] redis-cli

127.0.0.1:6379> kyes \*

1) "votes"

127.0.0.1:6379> type votes

list

127.0.0.1:6379> lrange votes 0 10

1) "{\"voter\_id\": \"8fec9c55f129301\", \"vote\": \"a\"}"

Выходим из виртуального окружения python

deactivate

Теперь нужно создать systemd сервис

cd /etc/systemd/system/

cp /usr/lib/systemd/system/sshd.service vote.service

vim vote.service

[Unit]

Description=Vote app

After=network.target

[Service]

Type=simple

ExecStart=/root/example-voting-app/vote/.venv/bin/gunicorn app:app -b 0.0.0.0:3000 --log-file - --access-logfile - --workers 4 --keep-alive 0

WorkingDirectory=/root/example-voting-app/vote

Restart=on-failure

RestartSec=42s

[Install]

WantedBy=multi-user.target

Запускаем, добавляем в автозагрузку

systemctl daemon-reload

systemctl start vote.service

systemctl enable vote.service

**Worker service**

Переходим в директорию worker

cd /root/example-voting-app/worker

Устанавливаем dotnet и sdk

dnf install dotnet-sdk-3.1

dnf install dotnet-runtime-5.0

Запускаем и убеждаемся, что сервис работает (подключается к redis и db)

dotnet publish -c Release -o /out src/Worker/Worker.csproj

dotnet /root/example-voting-app/worker/src/Worker/bin/Release/netcoreapp3.1/Worker.dll

Создаём systemd service

cd /etc/systemd/system/

cp vote.service worker.service

vim worker.service

[Unit]

Description=Worker app

After=network.target

[Service]

Type=simple

ExecStart=/usr/bin/dotnet /root/example-voting-app/worker/src/Worker/bin/Release/netcoreapp3.1/Worker.dll

WorkingDirectory=/root/example-voting-app/worker

Restart=on-failure

RestartSec=42s

[Install]

WantedBy=multi-user.target

Запускаем и добавляем в автозагрузку

systemctl daemon-reload

systemctl start worker.service

systemctl enable worker.service

**Results app**

Переходим в директорию

cd /root/example-voting-app/result

Устанавливаем nodejs и npm

dnf module enable nodejs:10

dnf install nodejs

Устанавливаем зависимости и запускаем сервис

npm install

node server.js

Убеждаемся, что сервис работает и создаём systemd service

cd /etc/systemd/system/

cp vote.service results.service

vim results.service

[Unit]

Description=Results app

After=network.target

[Service]

Type=simple

ExecStart=node /root/example-voting-app/result/server.js

WorkingDirectory=/root/example-voting-app/result

Restart=on-failure

RestartSec=42s

[Install]

WantedBy=multi-user.target

Запускаем и добавляем в автозагрузку

systemctl daemon-reload

systemctl start results.service

systemctl enable results.service