Docker BPR (Build & Push & Run)

Выполнил: Гневнов А.Е.,ИВТ 2.1

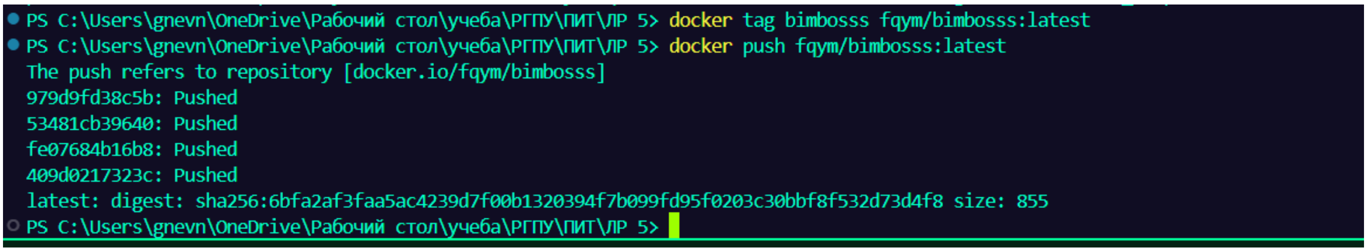
Конфиг Dockerfile

```
FROM golang:1.24.2-alpine AS builder

WORKDIR /app

COPY main.go .

RUN CGO_ENABLED=0 GOOS=linux GOARCH=amd64 go build -o app main.go

FROM alpine:latest

WORKDIR /app

COPY --from=builder /app/app .

EXPOSE 8089

CMD ["./app"]
```

Программа на Go

```
package main

import (
    "io"
    "log"
    "net/http"
)

func main() {
    http.HandleFunc("/", func(w http.ResponseWriter, r *http.Request) {
        w.Header().Set("Content-Type", "text/plain; charset=utf-8")
        w.WriteHeader(http.StatusOK)
        _, err := w.Write([]byte("Приложение go работает"))
        if err != nil {
            log.Println("Не удалось записать ответ:", err)
        }
    })

    http.HandleFunc("/echo", func(w http.ResponseWriter, r *http.Request) {
        if r.Method != http.MethodPost {
            w.Header().Set("Allow", http.MethodPost)
            http.Error(w, "Method Not Allowed", http.StatusMethodNotAllowed)
            return
        }

        w.Header().Set("Content-Type", "application/octet-stream")
        if _, err := io.Copy(w, r.Body); err != nil {
            http.Error(w, "Internal Server Error", http.StatusInternalServerError)
            log.Println("ошибка при копировании тела запроса:", err)
        }
    })

    if err := http.ListenAndServe(":8080", nil); err != nil {
        log.Fatalf("Ошибка запуска сервера: %v", err)
    }
}
```

Пушим Docker image в Docker Hub



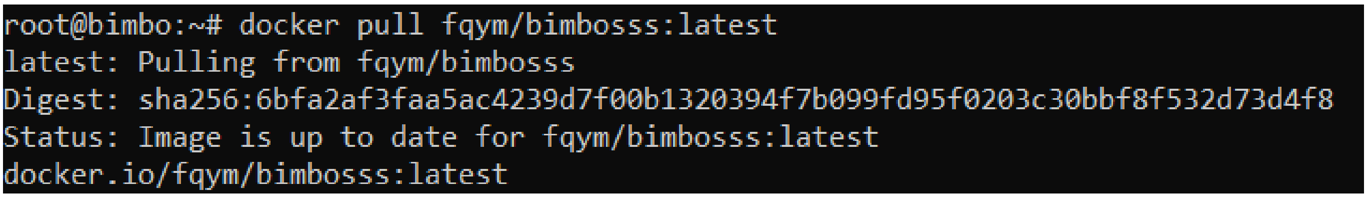
```
PS C:\Users\gnevn\OneDrive\Рабочий стол\учеба\РГПУ\ПИТ\ЛР 5> docker tag bimbosss fqym/bimbosss:latest
PS C:\Users\gnevn\OneDrive\Рабочий стол\учеба\РГПУ\ПИТ\ЛР 5> docker push fqym/bimbosss:latest
The push refers to repository [docker.io/fqym/bimbosss]
979d9fd38c5b: Pushed
53481cb39640: Pushed
fe07684b16b8: Pushed
409d0217323c: Pushed
latest: digest: sha256:6bfa2af3faa5ac4239d7f00b1320394f7b099fd95f0203c30bbf8f532d73d4f8 size: 855
PS C:\Users\gnevn\OneDrive\Рабочий стол\учеба\РГПУ\ПИТ\ЛР 5>
```

Устанавливаем docker на сервера

```
# Add Docker's official GPG key:
sudo apt-get update
sudo apt-get install ca-certificates curl
sudo install -m 0755 -d /etc/apt/keyrings
sudo curl -fsSL https://download.docker.com/linux/ubuntu/gpg -o
/etc/apt/keyrings/docker.asc
sudo chmod a+r /etc/apt/keyrings/docker.asc

# Add the repository to Apt sources:
echo \
  "deb [arch=$(dpkg --print-architecture) signed-by=/etc/apt/keyrings/docker.asc]
https://download.docker.com/linux/ubuntu \
  $(. /etc/os-release && echo "${UBUNTU_CODENAME:-$VERSION_CODENAME}") stable" | \
  sudo tee /etc/apt/sources.list.d/docker.list > /dev/null
sudo apt-get update
```

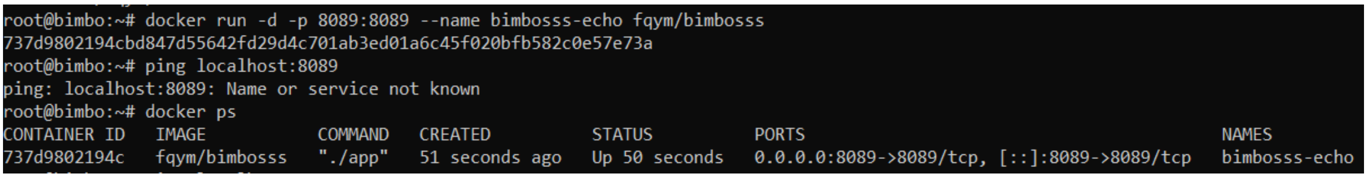
```
sudo apt-get install docker-ce docker-ce-cli containerd.io docker-buildx-plugin
docker-compose-plugin
```

Пулим докер image на сервера



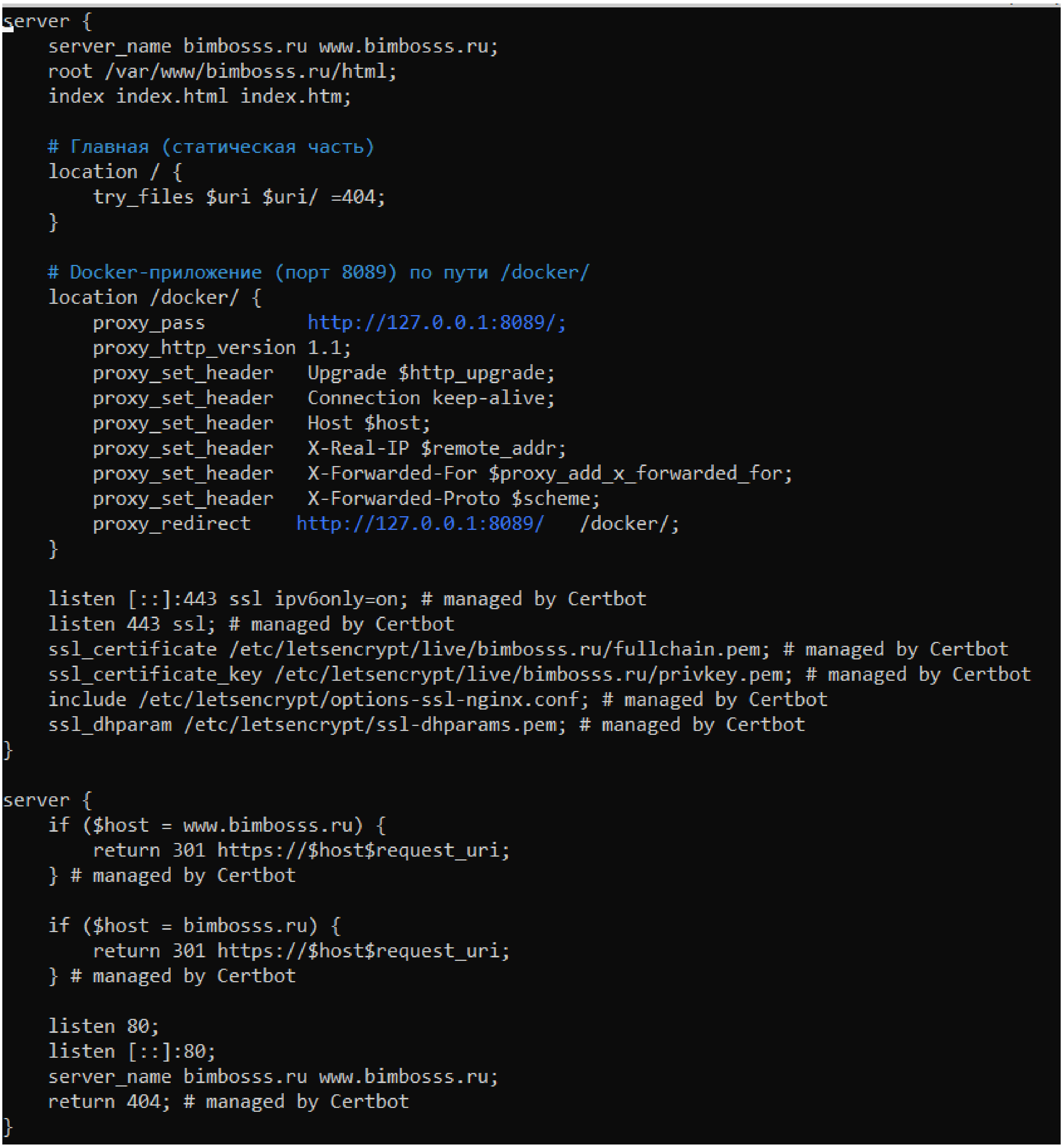
```
root@bimbo:~# docker pull fqym/bimbosss:latest
latest: Pulling from fqym/bimbosss
Digest: sha256:6bfa2af3faa5ac4239d7f00b1320394f7b099fd95f0203c30bbf8f532d73d4f8
Status: Image is up to date for fqym/bimbosss:latest
docker.io/fqym/bimbosss:latest
```

запускаем контейнер



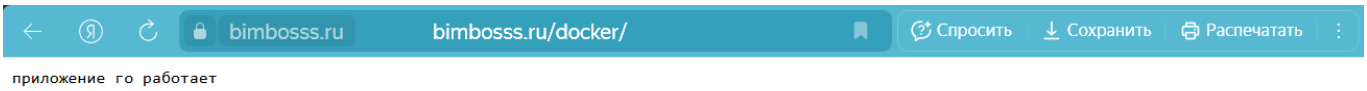
```
root@bimbo:~# docker run -d -p 8089:8089 --name bimbosss-echo fqym/bimbosss
737d9802194cbd847d55642fd29d4c701ab3ed01a6c45f020bfb582c0e57e73a
root@bimbo:~# ping localhost:8089
ping: localhost:8089: Name or service not known
root@bimbo:~# docker ps
CONTAINER ID   IMAGE           COMMAND   CREATED          STATUS          PORTS                                       NAMES
737d9802194c   fqym/bimbosss   "./app"   51 seconds ago   Up 50 seconds   0.0.0.0:8089->8089/tcp, [::]:8089->8089/tcp   bimbosss-echo
```

Настройка Ngnix конфига



```
server {
    server_name bimbosss.ru www.bimbosss.ru;
    root /var/www/bimbosss.ru/html;
    index index.html index.htm;

    # Главная (статическая часть)
    location / {
        try_files $uri $uri/ =404;
    }

    # Docker-приложение (порт 8089) по пути /docker/
    location /docker/ {
        proxy_pass          http://127.0.0.1:8089/;
        proxy_http_version 1.1;
        proxy_set_header   Upgrade $http_upgrade;
        proxy_set_header   Connection keep-alive;
        proxy_set_header   Host $host;
        proxy_set_header   X-Real-IP $remote_addr;
        proxy_set_header   X-Forwarded-For $proxy_add_x_forwarded_for;
        proxy_set_header   X-Forwarded-Proto $scheme;
        proxy_redirect    http://127.0.0.1:8089/   /docker/;
    }

    listen [::]:443 ssl ipv6only=on; # managed by Certbot
    listen 443 ssl; # managed by Certbot
    ssl_certificate /etc/letsencrypt/live/bimbosss.ru/fullchain.pem; # managed by Certbot
    ssl_certificate_key /etc/letsencrypt/live/bimbosss.ru/privkey.pem; # managed by Certbot
    include /etc/letsencrypt/options-ssl-nginx.conf; # managed by Certbot
    ssl_dhparam /etc/letsencrypt/ssl-dhparams.pem; # managed by Certbot
}

server {
    if ($host = www.bimbosss.ru) {
        return 301 https://$host$request_uri;
    } # managed by Certbot

    if ($host = bimbosss.ru) {
        return 301 https://$host$request_uri;
    } # managed by Certbot

    listen 80;
    listen [::]:80;
    server_name bimbosss.ru www.bimbosss.ru;
    return 404; # managed by Certbot
}
```

Проверяем работу



приложение го работает

Ссылка на Docker Hub

Репа