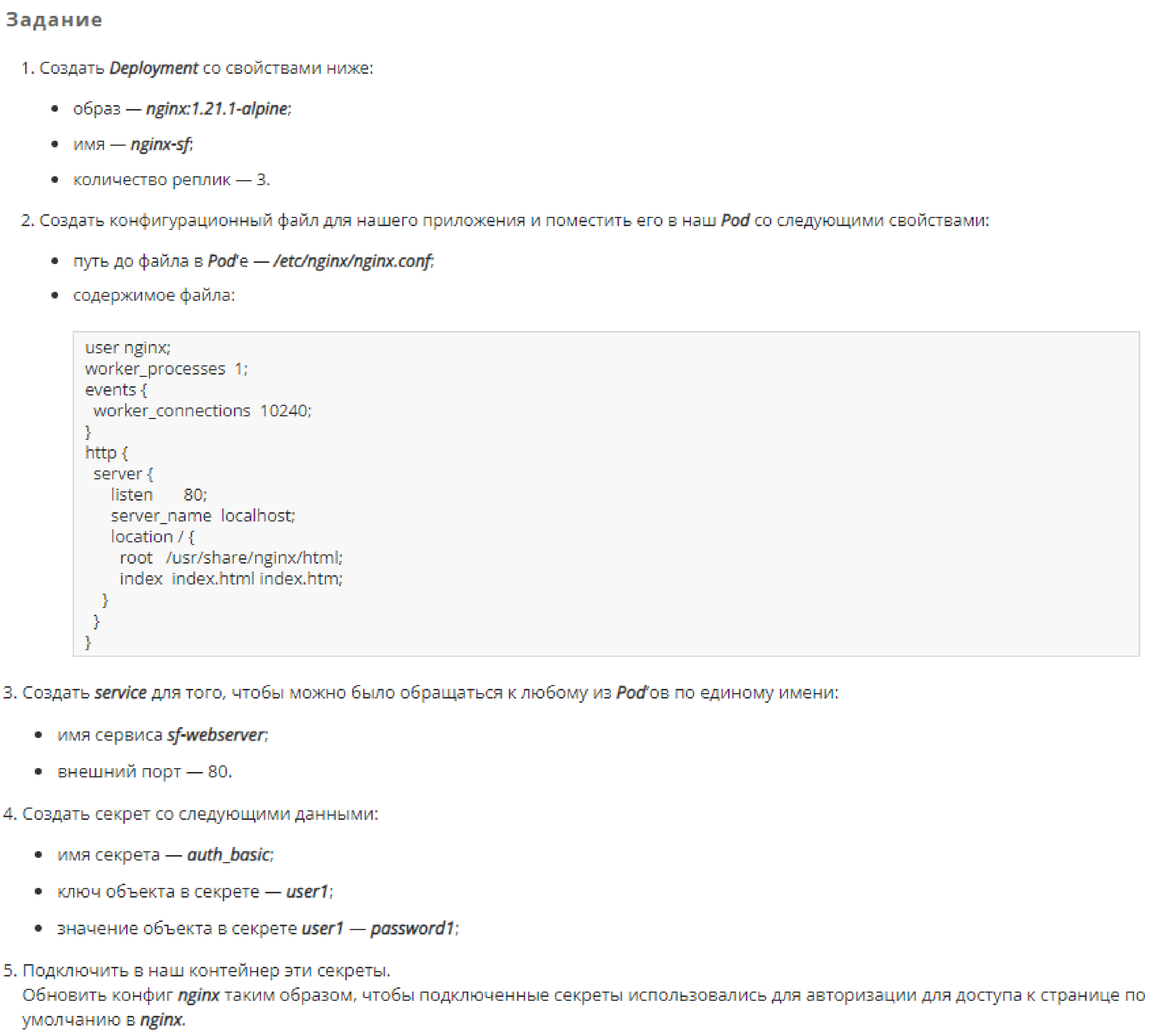
Module_D2_K8S_Report



Задание

1. Создать *Deployment* со свойствами ниже:
   - образ — *nginx:1.21.1-alpine*;
   - имя — *nginx-sf*;
   - количество реплик — 3.
2. Создать конфигурационный файл для нашего приложения и поместить его в наш *Pod* со следующими свойствами:
   - путь до файла в *Pod*'е — */etc/nginx/nginx.conf*;
   - содержимое файла:

```
user nginx;
worker_processes  1;
events {
 worker_connections  10240;
}
http {
 server {
     listen       80;
     server_name  localhost;
     location / {
       root   /usr/share/nginx/html;
       index  index.html index.htm;
   }
 }
}
```

3. Создать *service* для того, чтобы можно было обращаться к любому из *Pod*'ов по единому имени:
   - имя сервиса *sf-webserver*;
   - внешний порт — 80.
4. Создать секрет со следующими данными:
   - имя секрета — *auth_basic*;
   - ключ объекта в секрете — *user1*;
   - значение объекта в секрете *user1* — *password1*;
5. Подключить в наш контейнер эти секреты.
   Обновить конфиг *nginx* таким образом, чтобы подключенные секреты использовались для авторизации для доступа к странице по умолчанию в *nginx*.

0_env.sh

```
#!/bin/sh
# ---------------------------------------------------------------------------------
#   0_env.sh / 2022_05_17 / ANa
# ---------------------------------------------------------------------------------

sudo service docker start
sudo chmod 777 /var/run/docker.sock
docker login
```

1_run.sh

```
#!/bin/sh
# ---------------------------------------------------------------------------------
#   Kubernates / 2022_05_17 / ANa
# ---------------------------------------------------------------------------------

echo -e "\n"
echo ----------------------------------------------------- Create temp folders kubernates_root
mkdir ./kubernates_root
chmod 777 ./kubernates_root
cd kubernates_root



echo -e "\n"
echo ----------------------------------------------------- Create index.html
cat << EOF > ./index.html
<!-- ---------------------------- index.html START -------------------- -->
<!DOCTYPE html>
<html>
<body>
<h2>Page created by Andrey</h2>
</body>
</html>
<!-- ---------------------------- index.html END -------------------- -->
EOF
cat ./index.html



echo -e "\n"
echo ----------------------------------------------------- Create nginx.conf
cat << EOF > ./nginx.conf
# ---------------------------- nginx.conf START --------------------
user nginx;
worker_processes  1;
events {
  worker_connections  10240;
}
http {
  server {
      listen       80;
      server_name  localhost;
      location / {
        auth_basic          "Module_D3 Private Zone";
        auth_basic_user_file /etc/nginx/.htpasswd;
        root   /usr/share/nginx/html;
        index  index.html index.htm;
    }
  }
}
# ---------------------------- nginx.conf END  --------------------
EOF
cat ./nginx.conf



echo -e "\n"
echo ----------------------------------------------------- Create Dockerfile
cat << EOF > ./Dockerfile
# ---------------------------- Dockerfile START --------------------
FROM nginx:1.21.1-alpine

RUN apk --update --no-cache --virtual build-dependencies add apache2-utils

COPY ./nginx.conf /etc/nginx/nginx.conf
COPY ./index.html /usr/share/nginx/html/index.html
COPY ./and5_generate_nginx_credentials.sh /etc/nginx/and5_generate_nginx_credentials.sh

EXPOSE 80

# ---------------------------- Dockerfile END  --------------------
EOF
cat ./Dockerfile



echo -e "\n"
echo ---------------------------------------------------- Create and5_generate_nginx_credentials.sh
cat << EOF > ./and5_generate_nginx_credentials.sh
#!/bin/sh
# ---------------------------- and5_generate_nginx_credentials.sh START --------------------
# -----------------------------------------------------------------------------------
#   Kubernates / 2022_05_17 / ANa
# -----------------------------------------------------------------------------------

if [ -z "\$NGINX_USERNAME" ]
then
     export NGINX_USERNAME=andrey
fi

if [ -z "\$NGINX_PASSWORD" ]
then
     export NGINX_PASSWORD=andrey
fi

htpasswd -b -c /etc/nginx/.htpasswd \$NGINX_USERNAME \$NGINX_PASSWORD

# ---------------------------- and5_generate_nginx_credentials.sh END  --------------------
EOF
cat ./and5_generate_nginx_credentials.sh


echo -e "\n"
echo ---------------------------------------------------- Publishing Image into Docker HUB
docker rm `docker ps -q -l`
docker image rm `docker image ls -q` -f
docker build -t andreyk8s .
docker tag `docker images -q andreyk8s` silverstandart/andreyk8s:latest
docker push silverstandart/andreyk8s:latest
```

2_k8s.sh

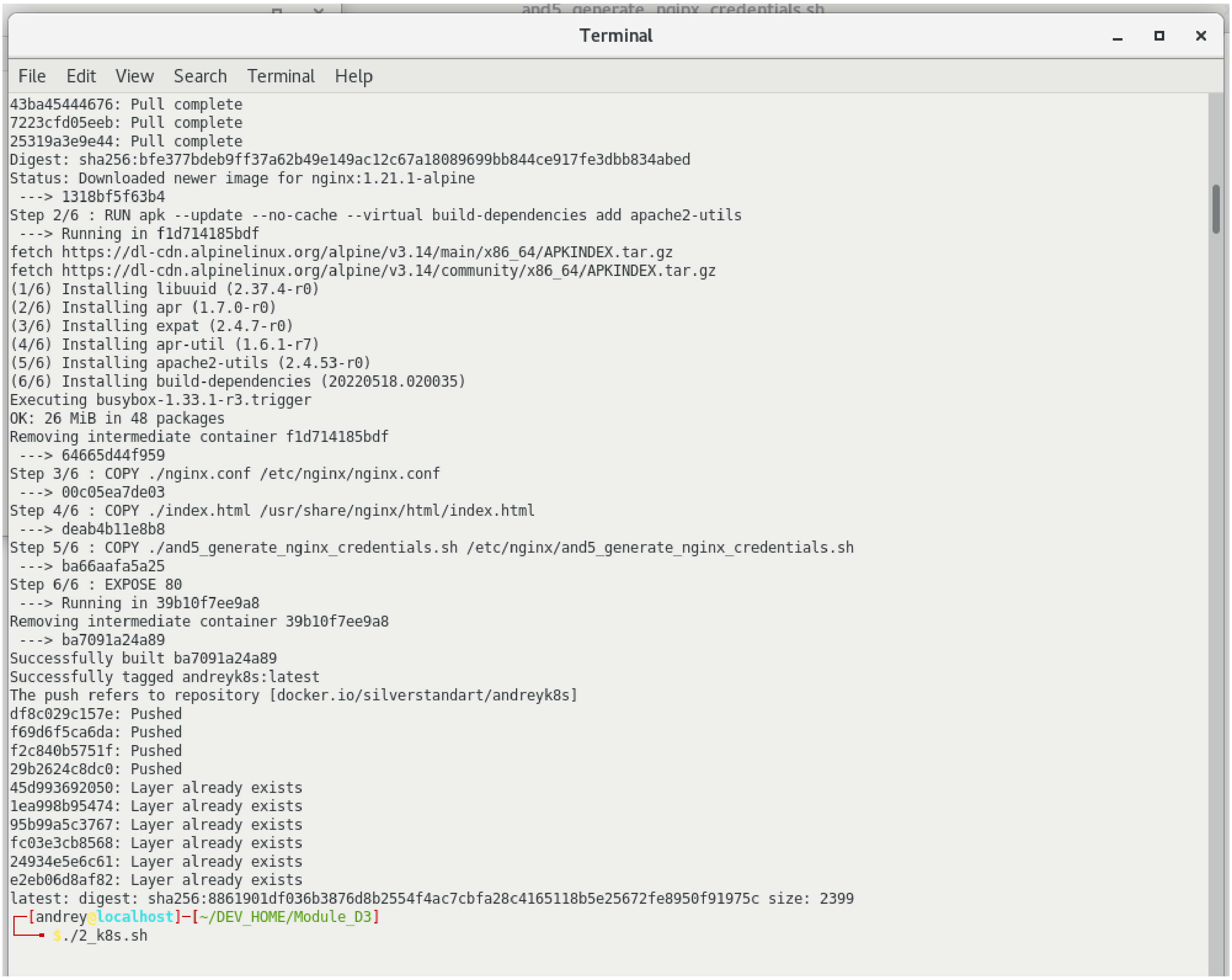
```
#!/bin/sh
# -----------------------------------------------------------------------------------
#   Kubernates / 2022_05_17 / ANa
# -----------------------------------------------------------------------------------

alias k=kubectl
alias m=minikube


echo -e "\n"
echo ---------------------------------------------------- Create temp folders kubernates_root
mkdir ./kubernates_root
```

```
chmod 777 ./kubernates_root
cd kubernates_root



echo -e "\n"
echo ----------------------------------------------------- Create and5-secret.yml
cat << EOF > ./and5-secret.yml
# ---------------------------- and5-secret.yml START --------------------
apiVersion: v1
kind: Secret
metadata:
  name: auth-basic
type: kubernetes.io/basic-auth
stringData:
  username: user1
  password: password1
# ---------------------------- and5-secret.yml END  --------------------
EOF
cat ./and5-secret.yml



echo -e "\n"
echo ----------------------------------------------------- Create and5-dl.yml
cat << EOF > ./and5-dl.yml
# ---------------------------- and5-dl.yml START --------------------
apiVersion : apps/v1
kind: Deployment
metadata:
  name: nginx-sf
  labels:
    tier: 2tiers
    owner: andrey
spec:
  replicas: 3
  selector:
    matchLabels:
      tier: 2tiers
  template:
    metadata:
      labels:
        tier: 2tiers
    spec:
      containers:
        - name: and5-container-replica
          image: silverstandart/andreyk8s:latest
          ports:
            - containerPort: 80
          env:
            - name: NGINX_USERNAME
              valueFrom:
                secretKeyRef:
                  name: auth-basic
                  key: username
            - name: NGINX_PASSWORD
              valueFrom:
                secretKeyRef:
                  name: auth-basic
                  key: password
        command: ["/bin/sh"]
        args: ["-c", "chmod 777 /etc/nginx/and5_generate_nginx_credentials.sh;
/etc/nginx/and5_generate_nginx_credentials.sh; /usr/sbin/nginx -g 'daemon off;'"]


---
apiVersion: v1
kind: Service
metadata:
  name: sf-webserver
spec:
  type: NodePort
  selector:
    tier: 2tiers
  ports:
  - name: http
    protocol: TCP
    port: 80
    targetPort: 80
# ---------------------------- and5-dl.yml END  --------------------
EOF
cat ./and5-dl.yml



echo -e "\n"
echo ---------------------------------------------------- Create Kubernates Cluster
m -p and5-main start --cpus=2 --memory=8gb --namespace and5-ns
k create namespace and5-ns

k delete -f and5-secret.yml
k apply  -f and5-secret.yml
k delete -f and5-dl.yml
k apply  -f and5-dl.yml

sleep 10

k get pods --selector=tier=2-4tiers -o jsonpath='{.items[*].status.podIP}'
echo -e "\n"

echo -e "\n"
echo ---------------------------------------------------- Pods Info
k get pods --show-labels

echo -e "\n"
echo ---------------------------------------------------- Services Info
k get services --show-labels

echo -e "\n"
k describe services sf-webserver

echo -e "\n"
k get pods -o wide -o yaml | grep podIP

echo -e "\n"
kubectl get nodes -o jsonpath='{.items[*].status.addresses[?(@.type=="ExternalIP")].address}'

echo -e "\n"
echo ----------------------------------------------------
echo -- NOTE to see tunnel use command in new console -----
echo m service list -p and5-main
echo ------------------------------------------------------

echo -e "\n"
m tunnel -p and5-main
```

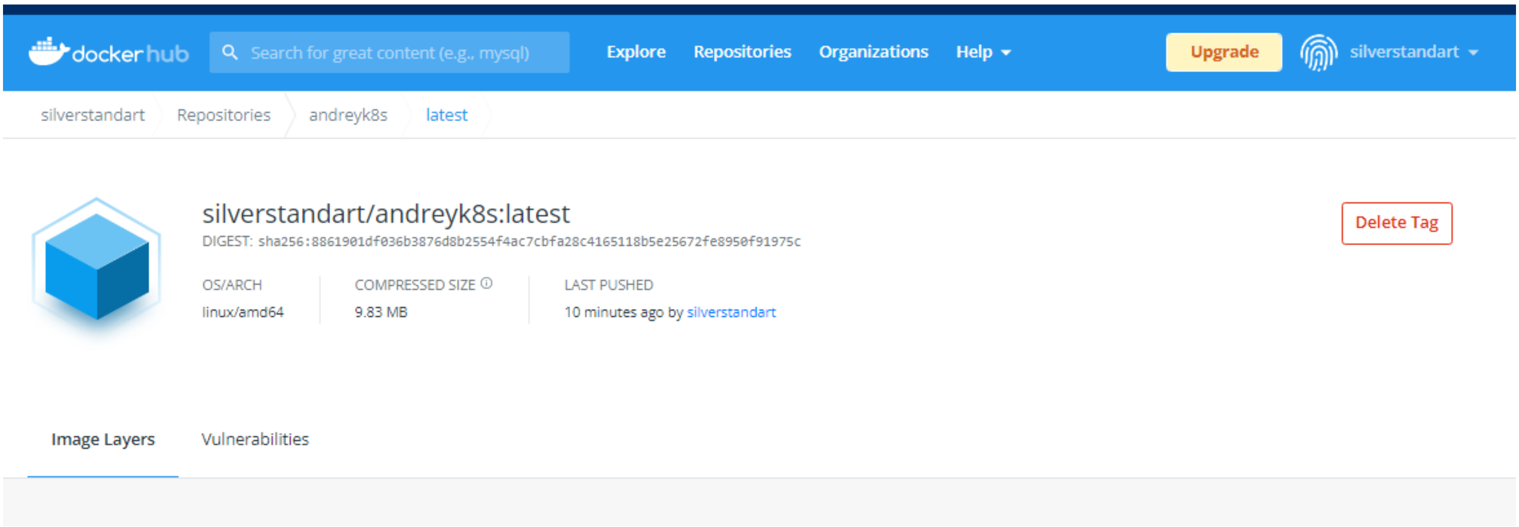
4_delete_garbage.sh

```
#!/bin/sh
# ---------------------------------------------------------------------------------------
#   Kubernates / 2022_05_17 / ANa
# ---------------------------------------------------------------------------------------

alias k=kubectl
alias m=minikube

m -p and5-main delete
k config view

echo -e "\n"

docker rm `docker ps -q -l`
docker image rm `docker image ls -q` -f
```

Running



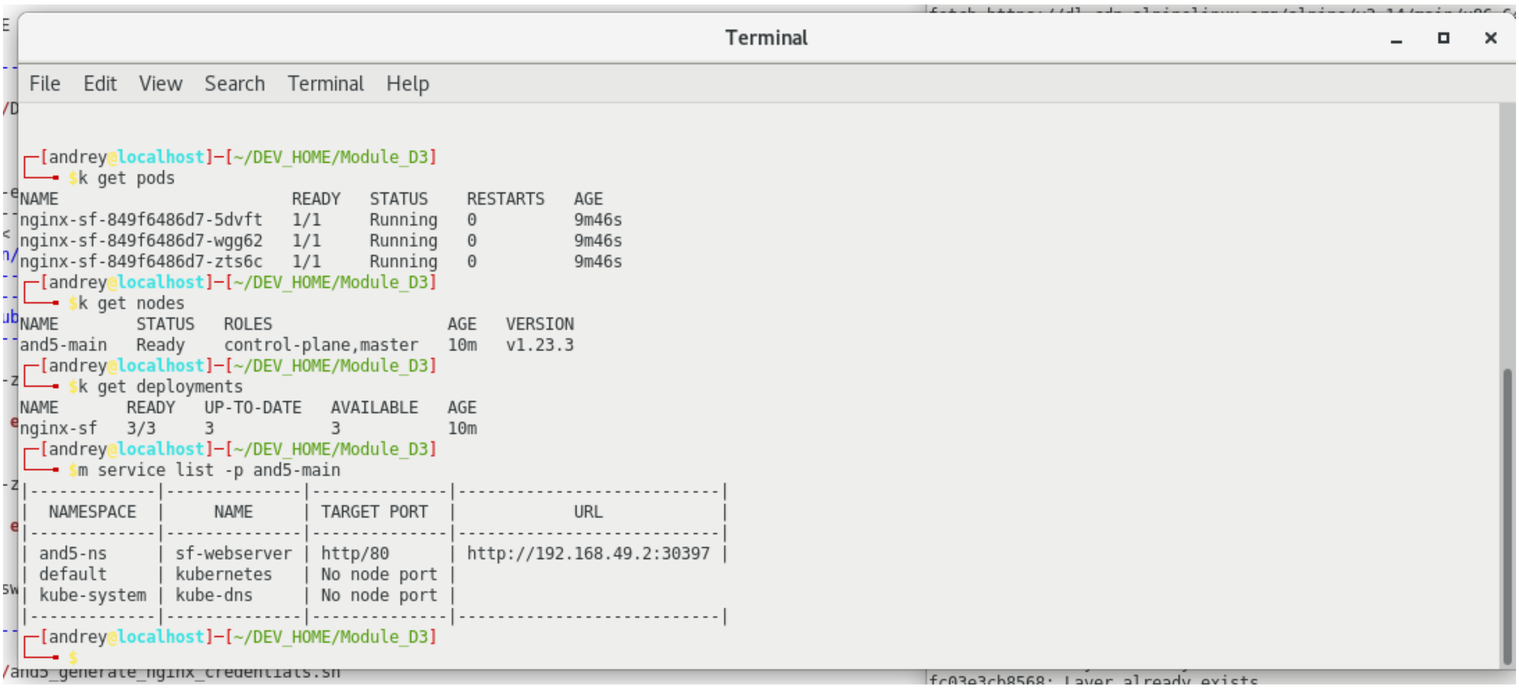
```
43ba45444676: Pull complete
7223cfd05eeb: Pull complete
25319a3e9e44: Pull complete
Digest: sha256:bfe377bdeb9ff37a62b49e149ac12c67a18089699bb844ce917fe3dbb834abed
Status: Downloaded newer image for nginx:1.21.1-alpine
 ---> 1318bf5f63b4
Step 2/6 : RUN apk --update --no-cache --virtual build-dependencies add apache2-utils
 ---> Running in f1d714185bdf
fetch https://dl-cdn.alpinelinux.org/alpine/v3.14/main/x86_64/APKINDEX.tar.gz
fetch https://dl-cdn.alpinelinux.org/alpine/v3.14/community/x86_64/APKINDEX.tar.gz
(1/6) Installing libuuid (2.37.4-r0)
(2/6) Installing apr (1.7.0-r0)
(3/6) Installing expat (2.4.7-r0)
(4/6) Installing apr-util (1.6.1-r7)
(5/6) Installing apache2-utils (2.4.53-r0)
(6/6) Installing build-dependencies (20220518.020035)
Executing busybox-1.33.1-r3.trigger
OK: 26 MiB in 48 packages
Removing intermediate container f1d714185bdf
 ---> 64665d44f959
Step 3/6 : COPY ./nginx.conf /etc/nginx/nginx.conf
 ---> 00c05ea7de03
Step 4/6 : COPY ./index.html /usr/share/nginx/html/index.html
 ---> deab4b11e8b8
Step 5/6 : COPY ./and5_generate_nginx_credentials.sh /etc/nginx/and5_generate_nginx_credentials.sh
 ---> ba66aafa5a25
Step 6/6 : EXPOSE 80
 ---> Running in 39b10f7ee9a8
Removing intermediate container 39b10f7ee9a8
 ---> ba7091a24a89
Successfully built ba7091a24a89
Successfully tagged andreyk8s:latest
The push refers to repository [docker.io/silverstandart/andreyk8s]
df8c029c157e: Pushed
f69d6f5ca6da: Pushed
f2c840b5751f: Pushed
29b2624c8dc0: Pushed
45d993692050: Layer already exists
1ea998b95474: Layer already exists
95b99a5c3767: Layer already exists
fc03e3cb8568: Layer already exists
24934e5e6c61: Layer already exists
e2eb06d8af82: Layer already exists
latest: digest: sha256:8861901df036b3876d8b2554f4ac7cbfa28c4165118b5e25672fe8950f91975c size: 2399
┌─[andrey@localhost]─[~/DEV_HOME/Module_D3]
└──╼ $./2_k8s.sh
```



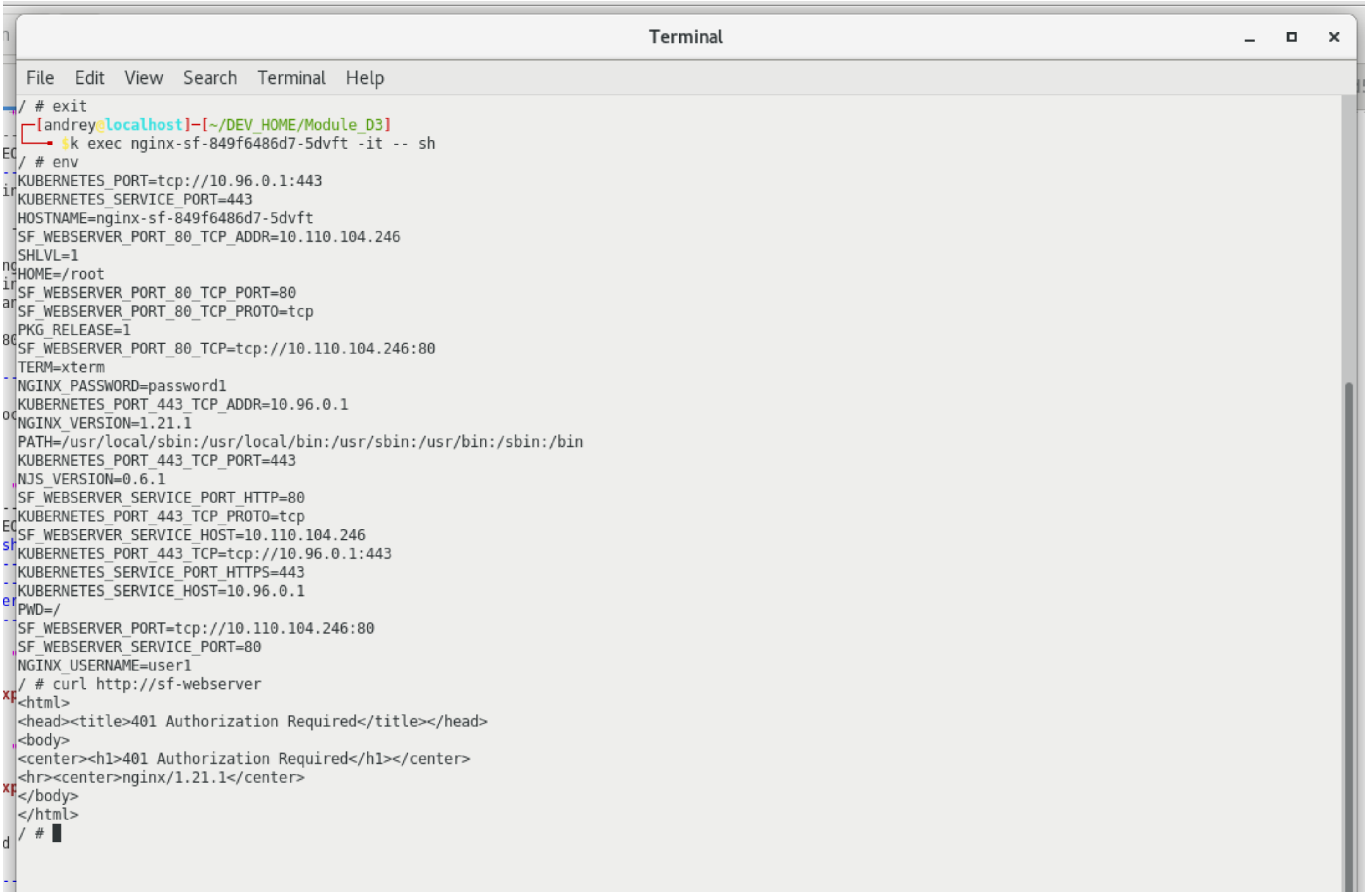
silverstandart/andreyk8s:latest

DIGEST: sha256:8861901df036b3876d8b2554f4ac7cbfa28c4165118b5e25672fe8950f91975c

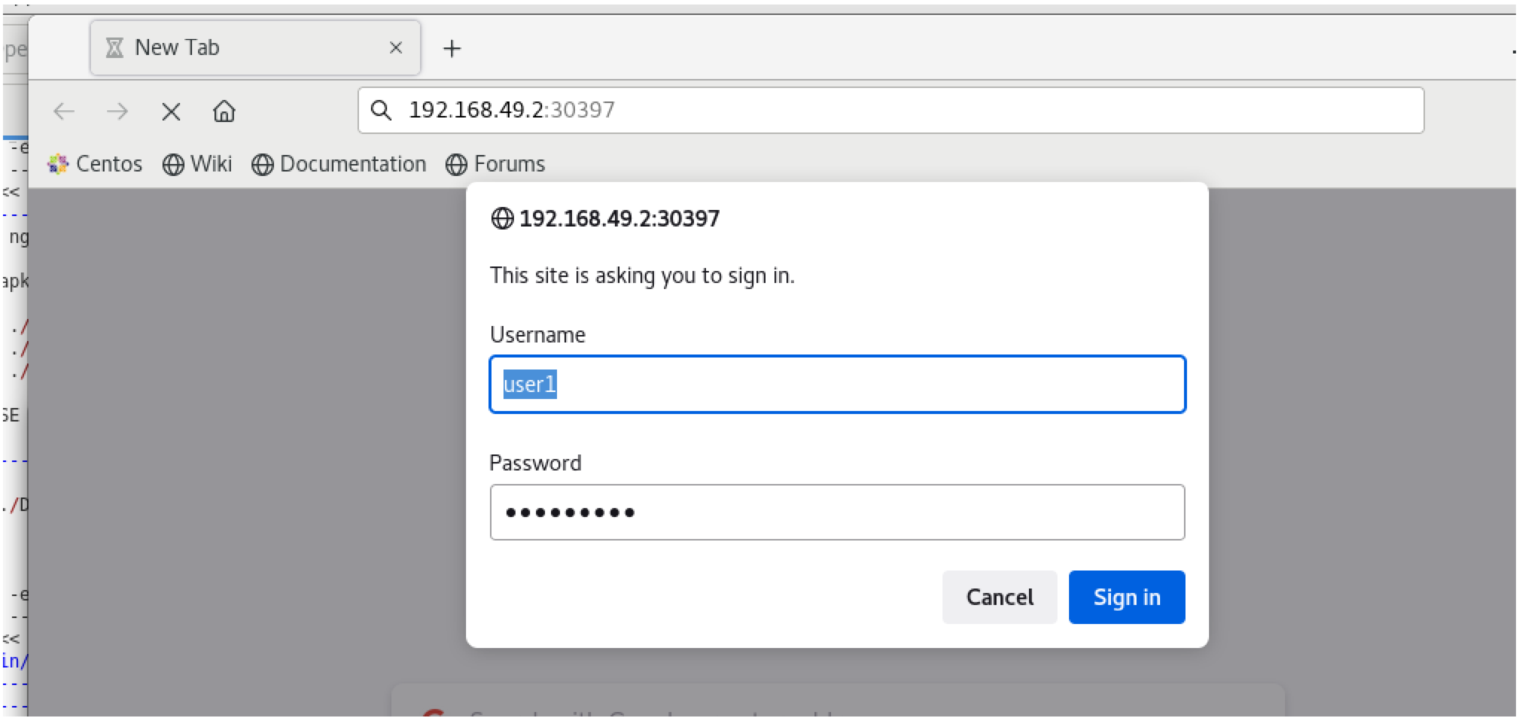
OS/ARCH	COMPRESSED SIZE	LAST PUSHED
linux/amd64	9.83 MB	10 minutes ago by silverstandart



```
┌─[andrey@localhost]─[~/DEV_HOME/Module_D3]
└──╼ $k get pods
NAME                        READY   STATUS    RESTARTS   AGE
nginx-sf-849f6486d7-5dvft   1/1     Running   0          9m46s
nginx-sf-849f6486d7-wgg62   1/1     Running   0          9m46s
nginx-sf-849f6486d7-zts6c   1/1     Running   0          9m46s
┌─[andrey@localhost]─[~/DEV_HOME/Module_D3]
└──╼ $k get nodes
NAME        STATUS   ROLES                  AGE   VERSION
and5-main   Ready    control-plane,master   10m   v1.23.3
┌─[andrey@localhost]─[~/DEV_HOME/Module_D3]
└──╼ $k get deployments
NAME       READY   UP-TO-DATE   AVAILABLE   AGE
nginx-sf   3/3     3            3           10m
┌─[andrey@localhost]─[~/DEV_HOME/Module_D3]
└──╼ $m service list -p and5-main
|-------------|--------------|-------------|---------------------------|
|  NAMESPACE  |     NAME     | TARGET PORT |            URL            |
|-------------|--------------|-------------|---------------------------|
| and5-ns     | sf-webserver | http/80     | http://192.168.49.2:30397 |
| default     | kubernetes   | No node port |
| kube-system | kube-dns     | No node port |
|-------------|--------------|-------------|---------------------------|
┌─[andrey@localhost]─[~/DEV_HOME/Module_D3]
└──╼ $
```



```
/ # exit
┌─[andrey@localhost]─[~/DEV_HOME/Module_D3]
└──╼ $k exec nginx-sf-849f6486d7-5dvft -it -- sh
/ # env
KUBERNETES_PORT=tcp://10.96.0.1:443
KUBERNETES_SERVICE_PORT=443
HOSTNAME=nginx-sf-849f6486d7-5dvft
SF_WEBSERVER_PORT_80_TCP_ADDR=10.110.104.246
SHLVL=1
HOME=/root
SF_WEBSERVER_PORT_80_TCP_PORT=80
SF_WEBSERVER_PORT_80_TCP_PROTO=tcp
PKG_RELEASE=1
SF_WEBSERVER_PORT_80_TCP=tcp://10.110.104.246:80
TERM=xterm
NGINX_PASSWORD=password1
KUBERNETES_PORT_443_TCP_ADDR=10.96.0.1
NGINX_VERSION=1.21.1
PATH=/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin
KUBERNETES_PORT_443_TCP_PORT=443
NJS_VERSION=0.6.1
SF_WEBSERVER_SERVICE_PORT_HTTP=80
KUBERNETES_PORT_443_TCP_PROTO=tcp
SF_WEBSERVER_SERVICE_HOST=10.110.104.246
KUBERNETES_PORT_443_TCP=tcp://10.96.0.1:443
KUBERNETES_SERVICE_PORT_HTTPS=443
KUBERNETES_SERVICE_HOST=10.96.0.1
PWD=/
SF_WEBSERVER_PORT=tcp://10.110.104.246:80
SF_WEBSERVER_SERVICE_PORT=80
NGINX_USERNAME=user1
/ # curl http://sf-webserver
<html>
<head><title>401 Authorization Required</title></head>
<body>
<center><h1>401 Authorization Required</h1></center>
<hr><center>nginx/1.21.1</center>
</body>
</html>
/ #
```

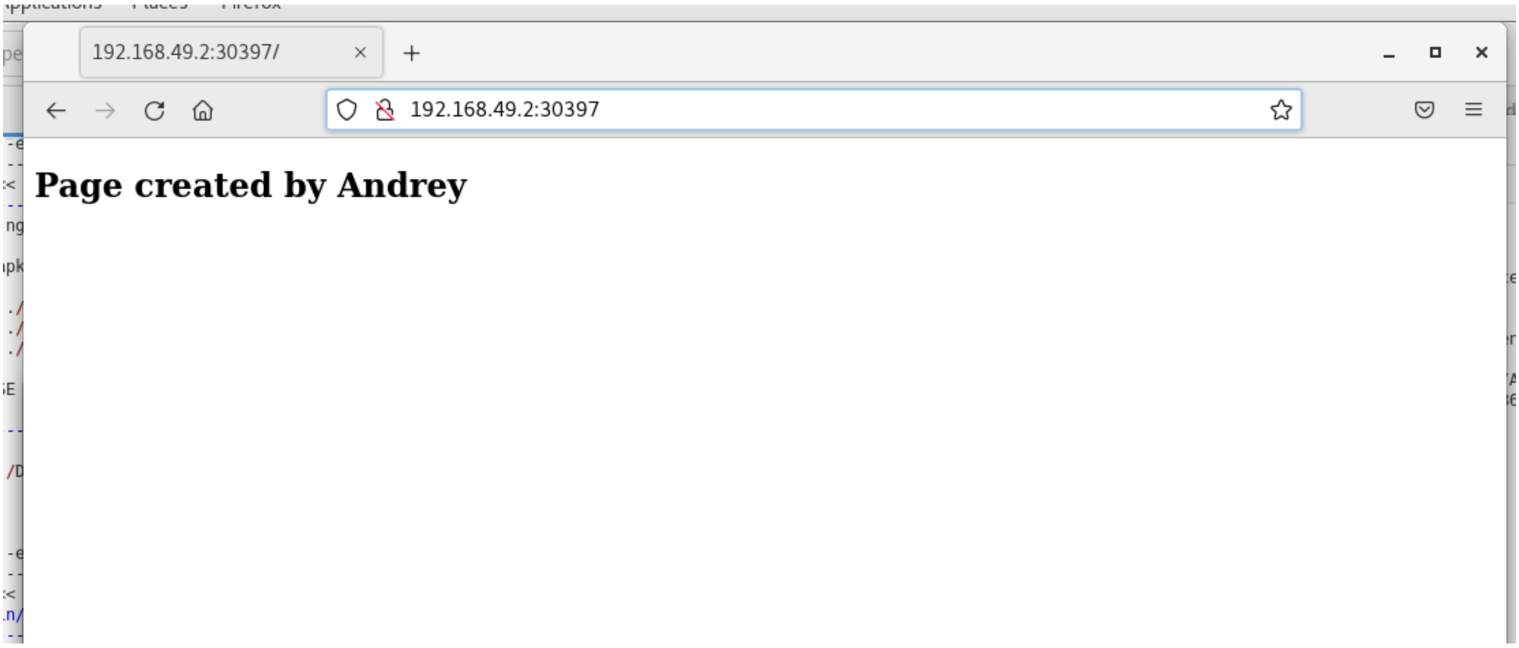


192.168.49.2:30397

This site is asking you to sign in.

Username: user1

Password: ••••••••••



Page created by Andrey