

Large Scale Computing

Lab 5 Kubernetes

Michał Świstowski

1.

```
FROM python:3.9-alpine AS builder

RUN apk add --no-cache \
    gcc \
    musl-dev \
    libffi-dev \
    openssl-dev \
    make \
    bash \
    curl

WORKDIR /usr/src/app

RUN curl -o awscli.tar.gz https://awscli.amazonaws.com/awscli.tar.gz && \
    \
    tar -xzf awscli.tar.gz && \
    cd awscli-2.19.4 && \
    ./configure --with-download-deps --prefix=/usr/local && \
    make && \
    make DESTDIR=/ install

# FROM alpine:3.16
FROM python:3.9-alpine

RUN apk add --no-cache \
    bash

COPY --from=builder /usr/local/bin/aws /usr/local/bin/
COPY --from=builder /usr/local/lib/aws-cli/ /usr/local/lib/aws-cli/

WORKDIR /aws
ENTRYPOINT ["/usr/local/bin/aws"]
```

```

* michal@fedora ~/D/a/s/l/lab5 [1]> docker build -t aws-cli .
[+] Building 353.4s (13/13) FINISHED
=> [internal] load build definition from Dockerfile
=> => transferring dockerfile: 924B
=> [internal] load metadata for docker.io/library/python:3.9-alpine
=> [auth] library/python:pull token for registry-1.docker.io
=> [internal] load .dockerignore
=> => transferring context: 2B
=> [builder 1/4] FROM docker.io/library/python:3.9-alpine@sha256:2ae855d07a137e4e39f9da8995f2fcd938c5bcdde466f9a8ac437b2de68e1b1
=> CACHED [stage-1 2/5] RUN apk add --no-cache bash

```

```

michal@fedora ~/D/a/s/l/lab5 [254]> docker run --rm \
    -v ~/.aws:/root/.aws \
    aws-cli s3 ls

```

2024-11-03 17:45:26 lab4-bucket-ms

2.

```

michal@fedora ~/D/a/s/l/lab5> minikube start
🌐 minikube v1.33.1 on Fedora 40
🔧 Automatically selected the docker driver. Other choices: qemu2, virtualbox, ssh
🔧 Using Docker driver with root privileges
👉 Starting "minikube" primary control-plane node in "minikube" cluster
📡 Pulling base image v0.0.44 ...
🏠 Creating docker container (CPUs=2, Memory=7800MB) ...
🔧 Preparing Kubernetes v1.30.0 on Docker 26.1.1 ...
  ▪ Generating certificates and keys ...
  ▪ Booting up control plane ...
  ▪ Configuring RBAC rules ...
🔑 Configuring bridge CNI (Container Networking Interface) ...
🔧 Verifying Kubernetes components...
  ▪ Using image gcr.io/k8s-minikube/storage-provisioner:v5
🌟 Enabled addons: storage-provisioner, default-storageclass
🎉 Done! kubectl is now configured to use "minikube" cluster and "default" namespace by default

```

```

michal@fedora ~/D/a/s/l/lab5 [1]> helm install nfs-server nfs-ganesha-server-and-external-provisioner/nfs-server-provisioner --set storageClass.name=nfs-storage
NAME: nfs-server
LAST DEPLOYED: Sun Nov 10 19:33:07 2024
NAMESPACE: default
STATUS: deployed
REVISION: 1
TEST SUITE: None
NOTES:
The NFS Provisioner service has now been installed.

A storage class named 'nfs-storage' has now been created
and is available to provision dynamic volumes.

You can use this storageclass by creating a 'PersistentVolumeClaim' with the
correct storageClassName attribute. For example:

---
kind: PersistentVolumeClaim
apiVersion: v1
metadata:
  name: test-dynamic-volume-claim
spec:
  storageClassName: "nfs-storage"
  accessModes:
    - ReadWriteOnce
  resources:
    requests:
      storage: 100Mi

```

pvc.yaml:

```

apiVersion: v1
kind: PersistentVolumeClaim
metadata:
  name: nfs-pvc
spec:
  accessModes:

```

```
- ReadWriteMany
storageClassName: nfs-storage
resources:
  requests:
    storage: 1Gi
```

nginx-deployment.yaml:

```
apiVersion: apps/v1
kind: Deployment
metadata:
  name: nginx-deployment
spec:
  replicas: 1
  selector:
    matchLabels:
      app: nginx
  template:
    metadata:
      labels:
        app: nginx
    spec:
      containers:
        - name: nginx
          image: nginx:latest
          volumeMounts:
            - mountPath: /usr/share/nginx/html
              name: nfs-storage
      volumes:
        - name: nfs-storage
          persistentVolumeClaim:
            claimName: nfs-pvc
```

nginx-service.yaml

```
apiVersion: v1
kind: Service
metadata:
  name: nginx-service
spec:
  selector:
    app: nginx
  ports:
    - protocol: TCP
      port: 80
      targetPort: 80
```

```
type: LoadBalancer
```

```

[1]
persistentvolumeclaim/nfs-pvc created
deployment.apps/nginx-deployment created
service/nginx-service created

```

```

NAME                                READY   STATUS    RESTARTS   AGE
coredns-7db6d8ff4d-4dht2            1/1     Running   0           7m9s
etcd-minikube                        1/1     Running   0           7m23s
kube-apiserver-minikube              1/1     Running   0           7m23s
kube-controller-manager-minikube     1/1     Running   0           7m23s
kube-proxy-vnbt9                     1/1     Running   0           114s
kube-scheduler-minikube              1/1     Running   0           7m23s

```

```

nfs-server-nfs-server-provisioner-0

```

```

NAME                                READY   STATUS    RESTARTS   AGE
nfs-server-nfs-server-provisioner-0 1/1     Running   0           38s

```

copy-content-job.yaml:

```

apiVersion: batch/v1
kind: Job
metadata:
  name: copy-content-job
spec:
  template:
    spec:
      containers:
        - name: content-copier
          image: busybox
          command: ["/bin/sh", "-c"]
          args: ["echo 'Hello from NFS!' > /mnt/share/index.html"]
          volumeMounts:
            - name: nfs-storage
              mountPath: /mnt/share
      restartPolicy: Never
      volumes:
        - name: nfs-storage
          persistentVolumeClaim:
            claimName: nfs-pvc

```

```

job.batch/copy-content-job created
NAME                                STATUS    COMPLETIONS   DURATION   AGE
copy-content-job                    Running   0/1            9s         9s
NAME                                STATUS    COMPLETIONS   DURATION   AGE
copy-content-job                    Complete  1/1            9s         20s




```



```








michal@fedora ~/D/a/s/1/lab5> kubectl get pvc
NAME          STATUS    VOLUME                                     CAPACITY   ACCESS MODES   STORAGECLASS   VOLUMEATTRIBUTESCLASS   AGE
nfs-pvc       Bound     pvc-1791d37e-44c1-40f2-a72d-323a1941404f   1Gi        RWX            nfs-storage    <unset>                 3m13s
michal@fedora ~/D/a/s/1/lab5> minikube service nginx-service

|-----|-----|-----|-----|
| NAMESPACE |     NAME     | TARGET PORT |          URL           |
|-----|-----|-----|-----|
| default   | nginx-service |          80  | http://192.168.49.2:31878 |
|-----|-----|-----|-----|
🔗 Opening service default/nginx-service in default browser...
kf.iconthemes: Icon theme "" not found.
kf.coreaddons: Checking for plugins in QList("/usr/bin/kf6/kio", "/usr/lib64/qt6/plugins/kf6/kio")
kf.kio.core: "/usr/lib64/qt6/plugins/kf6/kio/activities.so" supports protocols QList("activities")
kf.kio.core: "/usr/lib64/qt6/plugins/kf6/kio/afc.so" supports protocols QList("afc")
kf.kio.core: "/usr/lib64/qt6/plugins/kf6/kio/archive.so" supports protocols QList("ar", "sevenz", "tar", "zip")
kf.kio.core: "/usr/lib64/qt6/plugins/kf6/kio/filter.so" supports protocols QList("bzip", "bzip2", "gzip", "lzma", "xz", "zstd")
kf.kio.core: "/usr/lib64/qt6/plugins/kf6/kio/fish.so" supports protocols QList("fish")
kf.kio.core: "/usr/lib64/qt6/plugins/kf6/kio/kio_filenamesearch.so" supports protocols QList("filenamesearch")

```



192.168.49.2:31878

 Most Visited
 Fedora Docs
 Fedora Magazine
 Fedora Project
 User Communities
 Red Hat
 Free Content

Hello from NFS!