Containers and Kubernetes

Large Scale Computing: Lab5 assignment report

Piotr Urbańczyk

November 6, 2024

Contents

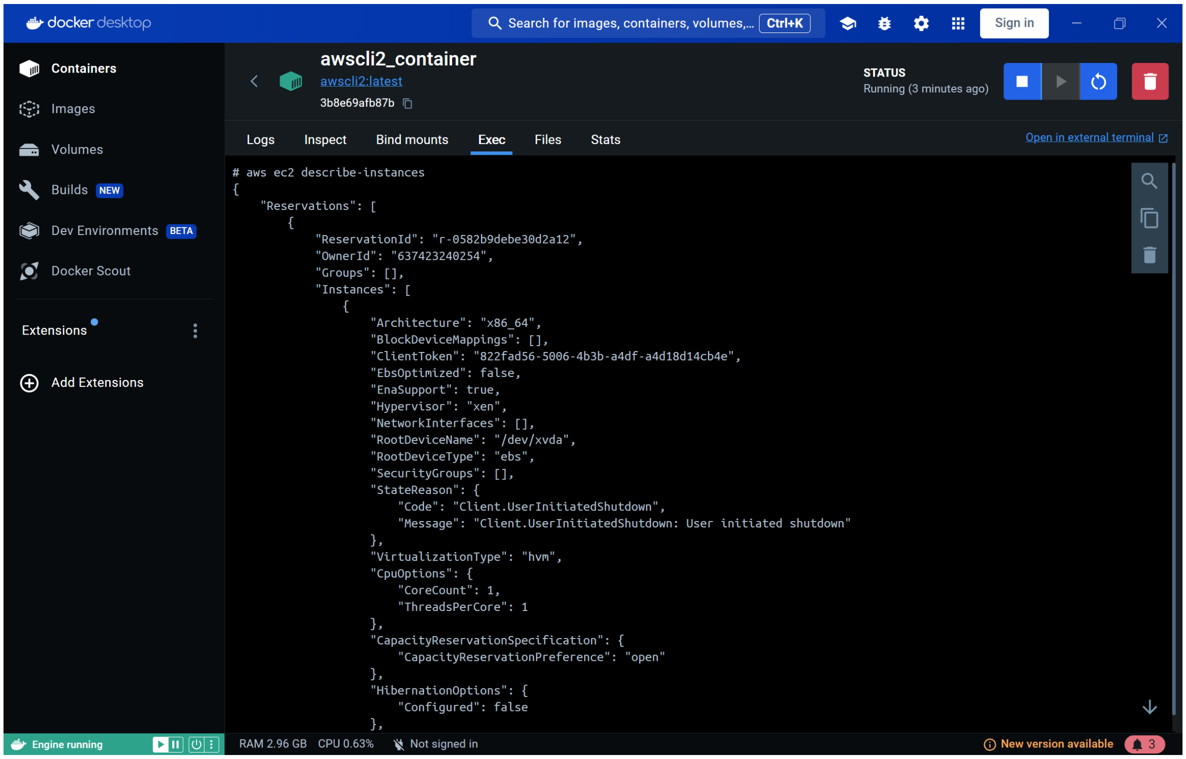
1 Dockerized AWS-CLI

Dockerfile

```
FROM public.ecr.aws/amazonlinux/amazonlinux:2 as awscli_builder
RUN yum update -y \
  && yum install -y unzip curl \
  && curl "https://awscli.amazonaws.com/awscli-exe-linux-x86_64.zip" -o "
     ↪ awscli-exe-linux-x86_64.zip" \
  && unzip awscli-exe-linux-x86_64.zip \
  && ./aws/install --bin-dir /aws-cli-bin/ \
  && rm -rf aws awscli-exe-linux-x86_64.zip \
  && yum clean all \
  && rm -rf /var/cache/yum

FROM debian:bookworm-slim
RUN apt-get update \
  && apt-get install -y --no-install-recommends \
    less \
    groff \
    ca-certificates \
  && apt-get clean \
  && rm -rf /var/lib/apt/lists/*

COPY --from=awscli_builder /usr/local/aws-cli/ /usr/local/aws-cli/
COPY --from=awscli_builder /aws-cli-bin/ /usr/local/bin/

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

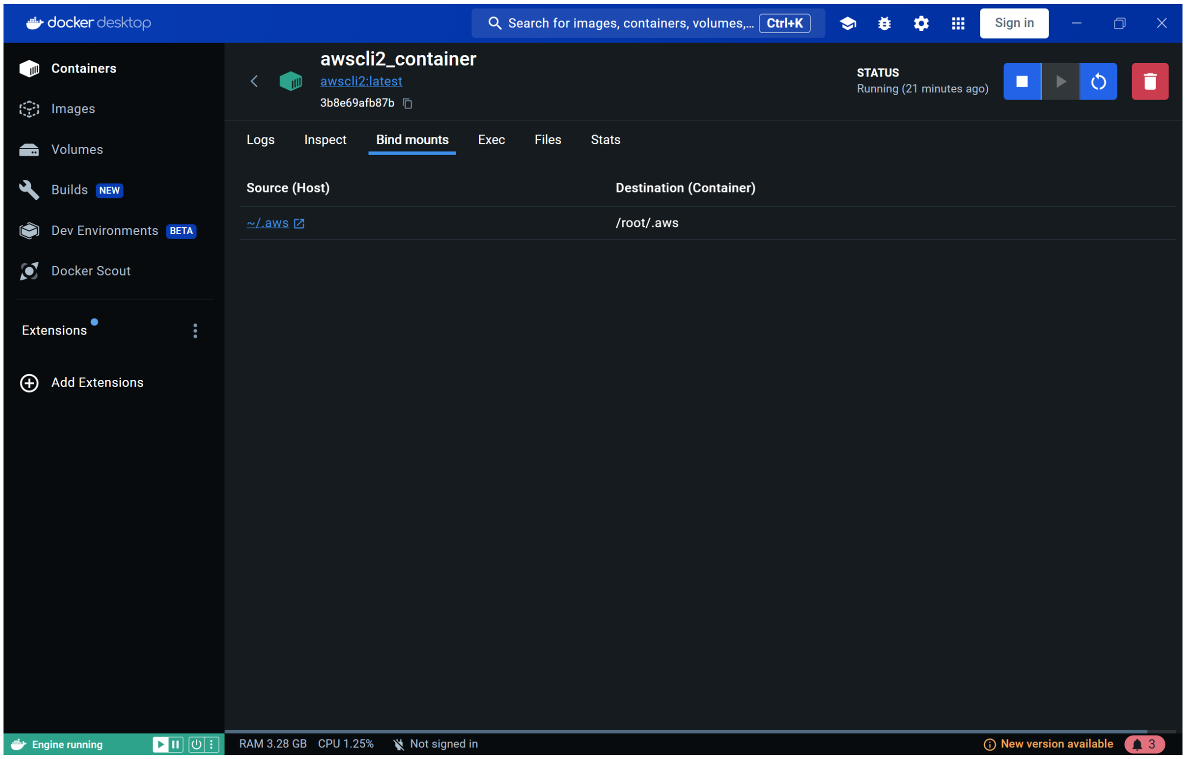
2 Kubernetes deployment

```
PS C:\Users\URBANCZYKP\Documents\studiaAGH\lsc\lab5> aws eks create-cluster `
>>   --name nfs-eks-cluster `
>>   --region us-east-1 `
>>   --role-arn arn:aws:iam::637423240254:role/LabRole `
>>   --resources-vpc-config subnetIds=subnet-006e480015beef182,subnet-0
    ↪ ec6276a49d82f7c9,subnet-0bc509828a0b2926f,subnet-0f5c172da77b1b4a9,
    ↪ subnet-04eaf52627defc331,securityGroupIds=sg-06615c37f24f4e07e
```



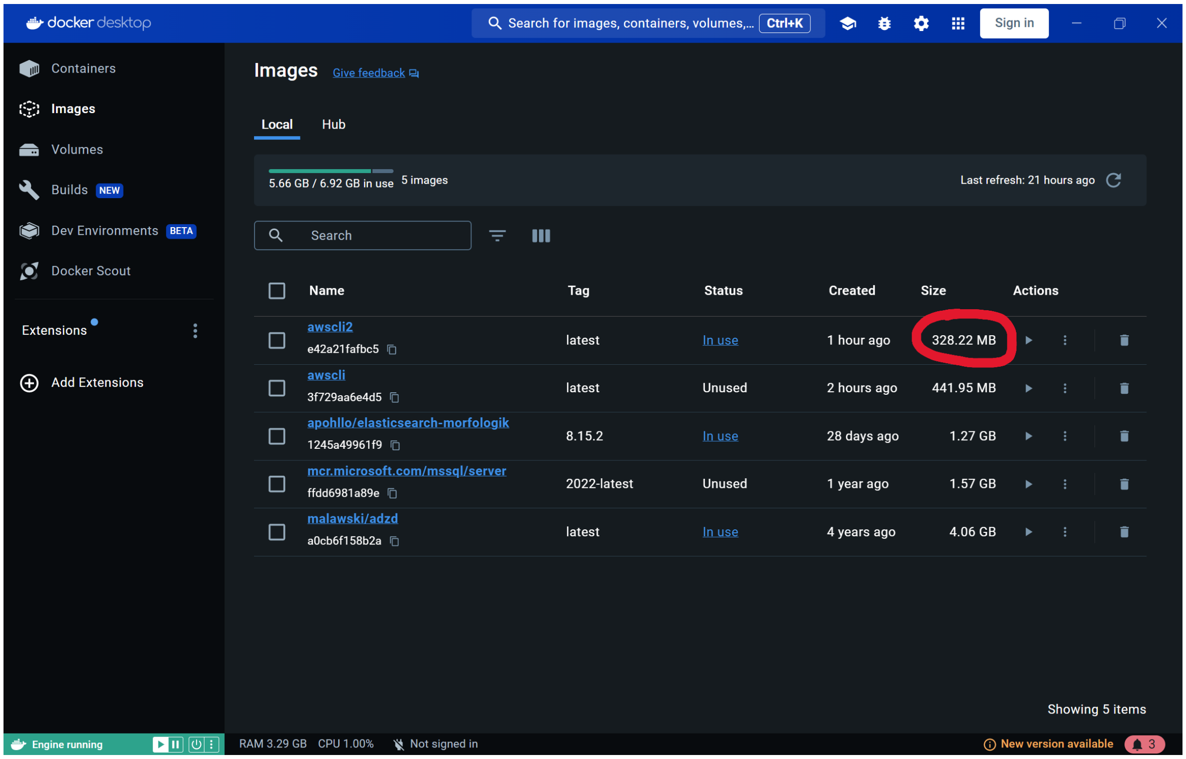
```
# aws ec2 describe-instances
{
    "Reservations": [
        {
            "ReservationId": "r-0582b9debe30d2a12",
            "OwnerId": "637423240254",
            "Groups": [],
            "Instances": [
                {
                    "Architecture": "x86_64",
                    "BlockDeviceMappings": [],
                    "ClientToken": "822fad56-5006-4b3b-a4df-a4d18d14cb4e",
                    "EbsOptimized": false,
                    "EnaSupport": true,
                    "Hypervisor": "xen",
                    "NetworkInterfaces": [],
                    "RootDeviceName": "/dev/xvda",
                    "RootDeviceType": "ebs",
                    "SecurityGroups": [],
                    "StateReason": {
                        "Code": "Client.UserInitiatedShutdown",
                        "Message": "Client.UserInitiatedShutdown: User initiated shutdown"
                    },
                    "VirtualizationType": "hvm",
                    "CpuOptions": {
                        "CoreCount": 1,
                        "ThreadsPerCore": 1
                    },
                    "CapacityReservationSpecification": {
                        "CapacityReservationPreference": "open"
                    },
                    "HibernationOptions": {
                        "Configured": false
                    },
```

Figure 1: Screenshot showing using the container 1



Source (Host)	Destination (Container)
~/.aws	/root/.aws

Figure 2: Screenshot showing using the container 2



5.66 GB / 6.92 GB in use 5 images

Last refresh: 21 hours ago

Name	Tag	Status	Created	Size	Actions
awscli2 e42a21fafbc5	latest	In use	1 hour ago	328.22 MB	
awscli 3f729aa6e4d5	latest	Unused	2 hours ago	441.95 MB	
apohllo/elasticsearch-morfologik 1245a49961f9	8.15.2	In use	28 days ago	1.27 GB	
mcr.microsoft.com/mssql/server ffdd6981a89e	2022-latest	Unused	1 year ago	1.57 GB	
malawski/adzd a0cb6f158b2a	latest	In use	4 years ago	4.06 GB	

Showing 5 items

RAM 3.29 GB CPU 1.00%

Figure 3: Screenshot showing using the container 3

```
PS C:\Users\URBANCZYKP\Documents\studiaAGH\lsc\lab5> aws eks describe-cluster
    ↪ ‘
>>   --name nfs-eks-cluster ‘
>>   --region us-east-1 ‘
>>   --query "cluster.status"
"ACTIVE"
```

```
PS C:\Users\URBANCZYKP\Documents\studiaAGH\lsc\lab5> aws eks describe-cluster
    ↪ ‘
>>   --name nfs-eks-cluster ‘
>>   --region us-east-1 ‘
>>   --query "cluster.status"
"ACTIVE"
```

```
PS C:\Users\URBANCZYKP\Documents\studiaAGH\lsc\lab5> aws eks create-nodegroup
    ↪ ‘
>>   --cluster-name nfs-eks-cluster ‘
>>   --nodegroup-name nfs-nodes ‘
>>   --node-role arn:aws:iam::637423240254:role/LabRole ‘
>>   --subnets subnet-006e480015beef182 subnet-0ec6276a49d82f7c9 subnet-0
    ↪ bc509828a0b2926f subnet-0f5c172da77b1b4a9 subnet-04eaf52627defc331 ‘
>>   --scaling-config minSize=1,maxSize=3,desiredSize=2 ‘
>>   --instance-types t3.medium ‘
>>   --region us-east-1
```

```
PS C:\Users\URBANCZYKP\Documents\studiaAGH\lsc\lab5> aws eks --region us-east
    ↪ -1 update-kubeconfig --name nfs-eks-cluster
Added new context arn:aws:eks:us-east-1:637423240254:cluster/nfs-eks-cluster
    ↪ to C:\Users\URBANCZYKP\.kube\config
```

```
PS C:\Users\URBANCZYKP\Documents\studiaAGH\lsc\lab5> helm install nfs-server
   ↪ nfs-provisioner/nfs-subdir-external-provisioner `
>>   --set storageClass.name=nfs-storage `
>>   --set storageClass.defaultClass=true `
>>   --set nfs.server=fs-08403e3f44c4a90a7.efs.us-east-1.amazonaws.com `
>>   --set nfs.path=/
```

pvc.yaml

```
apiVersion: v1
kind: PersistentVolumeClaim
metadata:
  name: nfs-pvc
spec:
  storageClassName: nfs-storage
  accessModes:
    - ReadWriteMany
  resources:
    requests:
      storage: 1Gi
```

```
PS C:\Users\URBANCZYKP\Documents\studiaAGH\lsc\lab5> kubectl apply -f pvc.yaml
persistentvolumeclaim/nfs-pvc created
```

nginx-deployment.yaml

```
apiVersion: apps/v1
kind: Deployment
metadata:
  name: web-server
spec:
  replicas: 1
  selector:
    matchLabels:
      app: web-server
  template:
    metadata:
      labels:
        app: web-server
    spec:
      containers:
      - name: nginx
        image: nginx:latest
        ports:
        - containerPort: 80
        volumeMounts:
        - mountPath: /usr/share/nginx/html
          name: web-content
      volumes:
      - name: web-content
        persistentVolumeClaim:
          claimName: nfs-pvc
```

```
PS C:\Users\URBANCZYKP\Documents\studiaAGH\lsc\lab5> kubectl apply -f nginx-
   ↪ deployment.yaml --validate=false
deployment.apps/web-server created
```

nginx-service.yaml

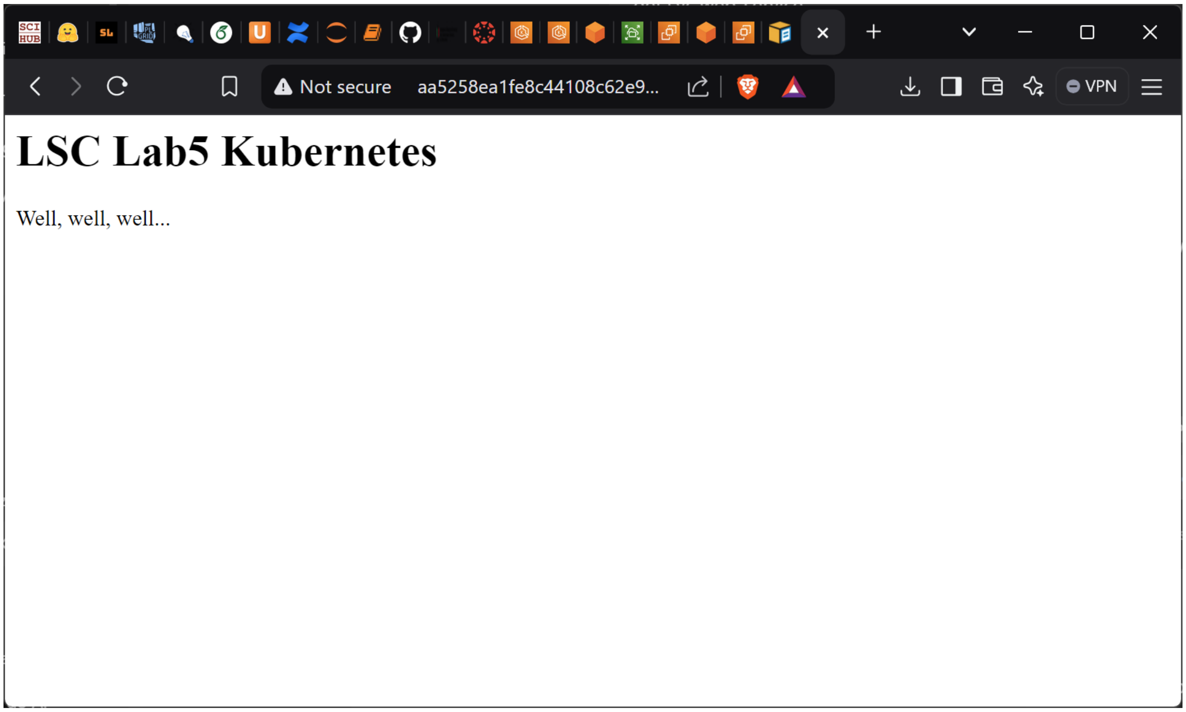
```
apiVersion: v1
kind: Service
metadata:
  name: web-service
spec:
  selector:
    app: web-server
  ports:
    - protocol: TCP
      port: 80
      targetPort: 80
  type: LoadBalancer
```

```
PS C:\Users\URBANCZYKP\Documents\studiaAGH\lsc\lab5> kubectl apply -f nginx-
    ↪ service.yaml
service/web-service created
```

nginx-job.yaml

```
apiVersion: batch/v1
kind: Job
metadata:
  name: content-populator
spec:
  template:
    spec:
      containers:
      - name: busybox
        image: busybox
        command: [’sh’, ’-c’, "echo ’<h1>LSC Lab5 Kubernetes</h1><p>Well, well
            ↪ , well...</p>’ > /mnt/index.html"]
        volumeMounts:
        - mountPath: /mnt
          name: web-content
      restartPolicy: OnFailure
      volumes:
      - name: web-content
        persistentVolumeClaim:
          claimName: nfs-pvc
```

```
PS C:\Users\URBANCZYKP\Documents\studiaAGH\lsc\lab5> kubectl apply -f nginx-
    ↪ job.yaml
job.batch/content-populator created
```



LSC Lab5 Kubernetes

Well, well, well...

Figure 4: Nginx server running