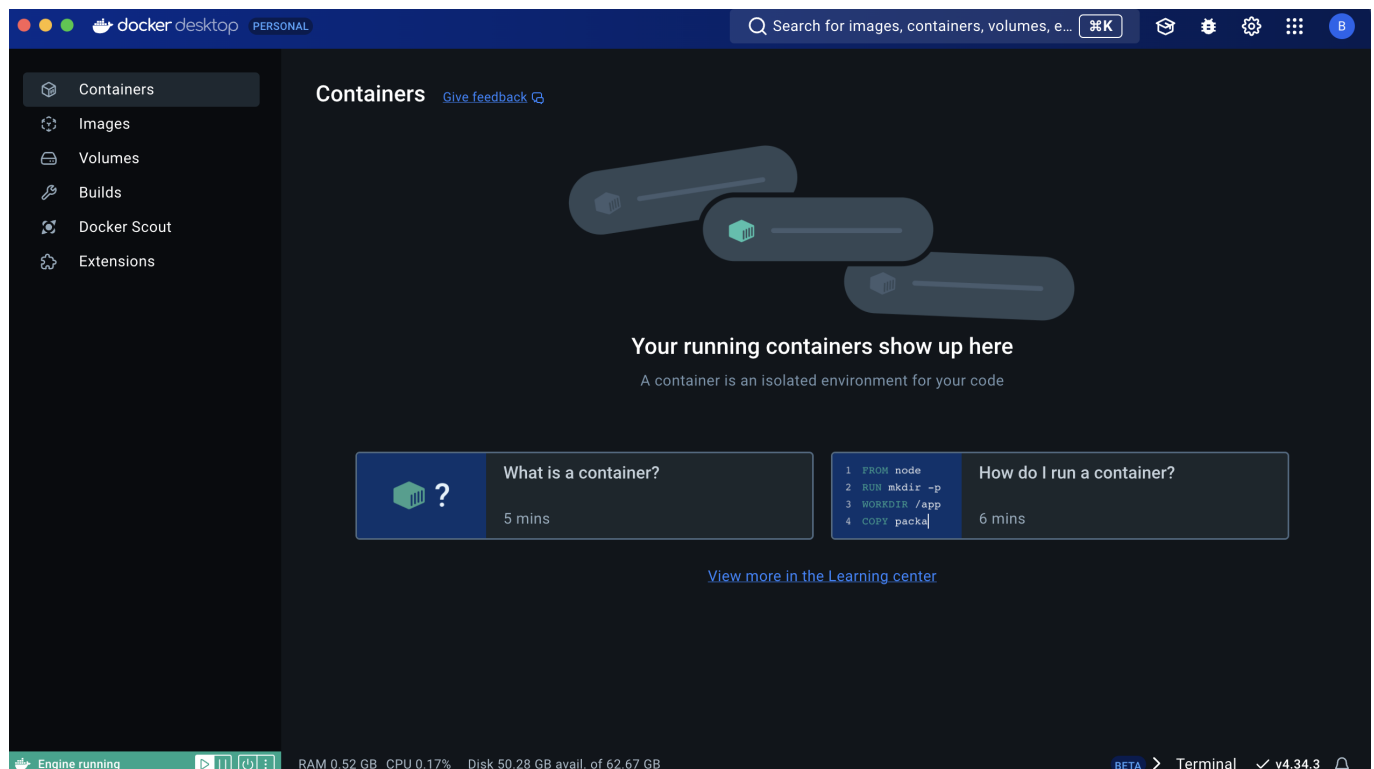


Author: Bartłomiej Kruczek

Date: 2024-11-04

Prerequisites

Docker installation



Docker version

```
[bk@MacBook-Pro-BartomiejK ~ % docker --version
Docker version 27.2.0, build 3ab4256
bk@MacBook-Pro-BartomiejK ~ %
```

Kubernetes installation

```

[bk@MacBook-Pro-BartomiejK ~ % brew install kubectl
==> Auto-updating Homebrew...
Adjust how often this is run with HOMEBREW_AUTO_UPDATE_SECS or disable with
HOMEBREW_NO_AUTO_UPDATE. Hide these hints with HOMEBREW_NO_ENV_HINTS (see `man brew`).
==> Auto-updated Homebrew!
Updated 2 taps (homebrew/core and homebrew/cask).
==> New Formulae
action-docs      distill-cli      kubetail         markdown-oxide   sequoia-sq       vipsdisp
ansible-builder  fofit            langgraph-cli    minijinja-cli    sf               wasi-libc
boring           foot             lbfgspp          node@22          sleek            wasi-runtimes
carapace         hawkeye          libmsquic         python-freethreading  surfer           wasm-component-ld
clickhouse-sql-parser  icu4c@74        libspelling@0.2  rip2             termscp          zizmor
dipc             icu4c@76         localai           rust-script      tex-fmt          zsh-system-clipboard

==> New Casks
adlock           follow@alpha     font-mynauri-icons  langgraph-studio  singlebox        yellowdot
claude           follow@nightly   font-zpiz           lets              sketchup
cocoapackettanalyzer  font-doto        github-copilot-for-xcode  mailbird
dockside         font-faculty-glyphic  huggingchat        mailsteward
follow          font-greybeard     huly               morisawa-desktop-manager  webkinz

==> Deleted Installed Formulae
icu4c ✖

You have 47 outdated formulae installed.

==> Downloading https://ghcr.io/v2/homebrew/core/kubernetes-cli/manifests/1.31.2
##### 100.0%
==> Fetching kubernetes-cli
==> Downloading https://ghcr.io/v2/homebrew/core/kubernetes-cli/blobs/sha256:96b8128ae14250531c3263f9af97b94e77f812b3966ba7ac7de83c4ce19c866c
##### 100.0%
==> Pouring kubernetes-cli--1.31.2.arm64_sequoia.bottle.tar.gz
==> Caveats
zsh completions have been installed to:
/opt/homebrew/share/zsh/site-functions
==> Summary
📦 /opt/homebrew/Cellar/kubernetes-cli/1.31.2: 237 files, 60.2MB
==> Running `brew cleanup kubernetes-cli`...
Disable this behaviour by setting HOMEBREW_NO_INSTALL_CLEANUP.
Hide these hints with HOMEBREW_NO_ENV_HINTS (see `man brew`).
[bk@MacBook-Pro-BartomiejK ~ %

```

Kubernetes version

```

[bk@MacBook-Pro-BartomiejK ~ % kubectl version --client
Client Version: v1.30.2
Kustomize Version: v5.0.4-0.20230601165947-6ce0bf390ce3
[bk@MacBook-Pro-BartomiejK ~ %

```

Minikube

```
bk@MacBook-Pro-BartomiejK docker_files_section % minikube start

🐳 minikube v1.34.0 na Darwin 15.1 (arm64)
🌟 Automatycznie wybrano sterownik docker
👍 Using Docker Desktop driver with root privileges
🚀 Starting "minikube" primary control-plane node in "minikube" cluster
📦 Pulling base image v0.0.45 ...
📦 Downloading Kubernetes v1.31.0 preload ...
> preloaded-images-k8s-v18-v1...: 307.61 MiB / 307.61 MiB 100.00% 8.25 Mi
> gcr.io/k8s-minikube/kicbase...: 441.45 MiB / 441.45 MiB 100.00% 5.25 Mi
🔥 Creating docker container (CPUs=2, Memory=7790MB) ...
🔥 Przygotowywanie Kubernetesa v1.31.0 na Docker 27.2.0...
  ▪ Generating certificates and keys ...
  ▪ Uruchamianie płaszczyzny kontrolnej ...
  ▪ Konfigurowanie zasad RBAC ...
🔗 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
bk@MacBook-Pro-BartomiejK docker_files_section %
```

```
bk@MacBook-Pro-BartomiejK docker_files_section % minikube kubectl -- get po -A

> kubectl.sha256: 64 B / 64 B [-----] 100.00% ? p/s 0s
> kubectl: 53.94 MiB / 53.94 MiB [-----] 100.00% 9.09 MiB p/s 6.1s

NAMESPACE      NAME                                                    READY   STATUS    RESTARTS   AGE
kube-system    coredns-6f6b679f8f-grm2m                             1/1     Running   0           78s
kube-system    etcd-minikube                                           1/1     Running   0           83s
kube-system    kube-apiserver-minikube                               1/1     Running   0           84s
kube-system    kube-controller-manager-minikube                     1/1     Running   0           83s
kube-system    kube-proxy-8tnm9                                       1/1     Running   0           78s
kube-system    kube-scheduler-minikube                               1/1     Running   0           84s
kube-system    storage-provisioner                                    1/1     Running   0           82s
bk@MacBook-Pro-BartomiejK docker_files_section %
```

Assignments

Dockerize AWS-CLI

1. Create a Dockerfile with aws-cli, built from source files (using tar.gz file / make)

```
[+] Building 72.0s (9/9) FINISHED                                docker:desktop-linux
=> [internal] load build definition from Dockerfile              0.0s
=> => transferring dockerfile: 627B                             0.0s
=> [internal] load metadata for docker.io/library/python:3.8-alpine 2.4s
=> [auth] library/python:pull token for registry-1.docker.io    0.0s
=> [internal] load .dockerignore                                 0.0s
=> => transferring context: 2B                                     0.0s
=> [builder 1/2] FROM docker.io/library/python:3.8-alpine@sha256:3d93b1f77efce339aa77db726656872517b0d67837989aa7c4b35bd5ae7e81ba 2.3s
=> => resolve docker.io/library/python:3.8-alpine@sha256:3d93b1f77efce339aa77db726656872517b0d67837989aa7c4b35bd5ae7e81ba 0.0s
=> => sha256:6cfd366dbc53a2b83e7757be346b01d35ccd730de9c2f0552b5c1e51f779d9d0 249B / 249B 0.3s
=> => sha256:3d93b1f77efce339aa77db726656872517b0d67837989aa7c4b35bd5ae7e81ba 10.29kB / 10.29kB 0.0s
=> => sha256:b67ef8bb357c456c34b2cafb1127a4ddcf596069daf43c85cdee34e19c46fc8c 1.74kB / 1.74kB 0.0s
=> => sha256:83001c624401aa11580c97e56c54d556f580e847c90e6ff5efbc3da54cb2eb77 4.95kB / 4.95kB 0.0s
=> => sha256:f0d3d23155cbbb3aca6949baedd1935e7ad9f792b943689e594e2efc8a828dec 457.46kB / 457.46kB 0.5s
=> => sha256:fcfb6bf9f4c3fd6e39f9d7615cf202f9cc6e5d9a8f555120dde9e3b459b68f2 14.31MB / 14.31MB 1.8s
=> => extracting sha256:f0d3d23155cbbb3aca6949baedd1935e7ad9f792b943689e594e2efc8a828dec 0.1s
=> => extracting sha256:fcfb6bf9f4c3fd6e39f9d7615cf202f9cc6e5d9a8f555120dde9e3b459b68f2 0.4s
=> => extracting sha256:6cfd366dbc53a2b83e7757be346b01d35ccd730de9c2f0552b5c1e51f779d9d0 0.0s
=> [builder 2/2] RUN apk add --no-cache curl make cmake gcc g++ libc-dev libffi-dev openssl-dev && curl https://aws 65.2s
=> [stage-1 2/3] RUN apk --no-cache add groff 1.7s
=> [stage-1 3/3] COPY --from=builder /opt/aws-cli/ /opt/aws-cli/ 0.8s
=> exporting to image 0.8s
=> exporting layers 0.8s
=> writing image sha256:22a60b49d679fdebef2adbaeefc3036b670bb2954a40b3fd538f266847c9359 0.0s
=> naming to docker.io/library/awscli-alpine 0.0s
```

```
bk@MacBook-Pro-BartomiejK docker_files_section % docker run --rm -it awscli-alpine --version
aws-cli/2.10.1 Python/3.8.20 Linux/6.10.4-linuxkit source-sandbox/aarch64.alpine.3 prompt/off
bk@MacBook-Pro-BartomiejK docker_files_section %
```

2. Build an image based on your Dockerfile and test it enabling -v option, buckets creation and listing

```
bk@MacBook-Pro-BartomiejK docker_files_section % docker run --rm -v ~/.aws:/root/.aws aws-cli s3 ls

2024-11-03 15:45:53 bartekkruczekbucket
2024-10-28 08:16:10 bk-lab-bucket-20231015
bk@MacBook-Pro-BartomiejK docker_files_section %
```

Kubernetes deployment

1. Create a cluster -> kind used

```
bk@MacBook-Pro-BartomiejK Lab_5 (04.11.2024) % kind create cluster --name=lscdocker
Creating cluster "lscdocker" ...
✓ Ensuring node image (kindest/node:v1.31.2)
✓ Preparing nodes
✓ Writing configuration
✓ Starting control-plane
✓ Installing CNI
✓ Installing StorageClass
Set kubectl context to "kind-lscdocker"
You can now use your cluster with:

kubectl cluster-info --context kind-lscdocker

Have a nice day!
```

```
bk@MacBook-Pro-BartomiejK Lab_5 (04.11.2024) % helm install nfs-server-provisioner nfs-ganesha-server-and-external-provisioner/nfs-server-provisioner --set=storageClass.name=lscdocker
NAME: nfs-server-provisioner
LAST DEPLOYED: Tue Nov 12 14:51:49 2024
NAMESPACE: default
STATUS: deployed
REVISION: 1
TEST SUITE: None
NOTES:
The NFS Provisioner service has now been installed.

A storage class named 'lscdocker' 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: "lscdocker"
  accessModes:
    - ReadWriteOnce
  resources:
    requests:
      storage: 100Mi
```

2. Using helm install an NFS server and provisioner

```
bk@MacBook-Pro-BartomiejK docker_files_section % helm repo add nfs-ganesha-server-and-external-provisioner https://kubernetes-sigs.github.io/nfs-ganesha-server-and-external-provisioner/
"nfs-ganesha-server-and-external-provisioner" has been added to your repositories
bk@MacBook-Pro-BartomiejK docker_files_section % helm install my-release nfs-ganesha-server-and-external-provisioner/nfs-server-provisioner
NAME: my-release
LAST DEPLOYED: Sun Nov 10 19:51:02 2024
NAMESPACE: default
STATUS: deployed
REVISION: 1
TEST SUITE: None
NOTES:
The NFS Provisioner service has now been installed.

A storage class named 'nfs' 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"
  accessModes:
    - ReadWriteOnce
  resources:
    requests:
      storage: 100Mi
```

3. PVC creation

```
bk@MacBook-Pro-BartomiejK Lab_5 (04.11.2024) % kubectl apply --context kind-lscdocker --filename=lscpvc.yaml
persistentvolumeclaim/lsc-lab5-pvc created
bk@MacBook-Pro-BartomiejK Lab_5 (04.11.2024) % kubectl get pvc
NAME          STATUS  VOLUME                                     CAPACITY  ACCESS MODES  STORAGECLASS  VOLUMEATTRIBUTESCLASS  AGE
lsc-lab5-pvc  Bound  pvc-3840657d-5213-4251-8448-52b8e706d824  1Gi       RWX           lscdocker     <unset>                 8s
bk@MacBook-Pro-BartomiejK Lab_5 (04.11.2024) %
```


4. Deployment with HTTP server

```

bk@MacBook-Pro-BartomiejK Lab_5 (04.11.2024) % kubectl apply --context kind-lscdocker --filename=lscdeployment.yaml
deployment.apps/nginx-deployment created
bk@MacBook-Pro-BartomiejK Lab_5 (04.11.2024) % kubectl get deployment
NAME          READY   UP-TO-DATE   AVAILABLE   AGE
nginx-deployment 1/2     2            1           7s
bk@MacBook-Pro-BartomiejK Lab_5 (04.11.2024) % kubectl get deployment
NAME          READY   UP-TO-DATE   AVAILABLE   AGE
nginx-deployment 2/2     2            2           18s
bk@MacBook-Pro-BartomiejK Lab_5 (04.11.2024) % kubectl get deployment
NAME          READY   UP-TO-DATE   AVAILABLE   AGE
nginx-deployment 2/2     2            2           20s
bk@MacBook-Pro-BartomiejK Lab_5 (04.11.2024) %

```

5. Service creation

```

bk@MacBook-Pro-BartomiejK Lab_5 (04.11.2024) % kubectl apply --context kind-lscdocker --filename=lscservice.yaml
service/nginx-service created
bk@MacBook-Pro-BartomiejK Lab_5 (04.11.2024) % kubectl get service
NAME          TYPE          CLUSTER-IP      EXTERNAL-IP      PORT(S)
kubernetes    ClusterIP     10.96.0.1        <none>            443/TCP
nfs-server-provisioner ClusterIP     10.96.77.193     <none>            2049/TCP,2049/UDP,32803/TCP,32803/UDP,20048/TCP,20048/UDP,875/TCP,875/UDP,111/TCP,111/UDP,662/TCP,662/UDP
nginx-service  NodePort      10.96.89.193     <none>            80:30000/TCP

```

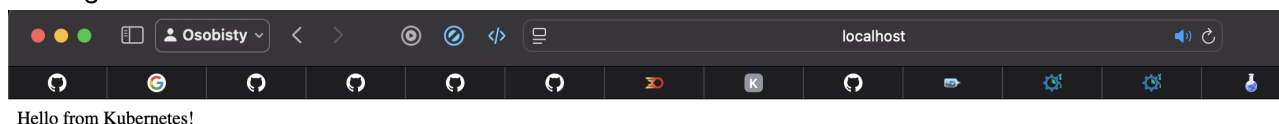
6. Copy job creation

```

bk@MacBook-Pro-BartomiejK Lab_5 (04.11.2024) % kubectl apply --context kind-lscdocker --filename=lscjob.yaml
job.batch/copy-content-job created
bk@MacBook-Pro-BartomiejK Lab_5 (04.11.2024) % kubectl get job
NAME          STATUS    COMPLETIONS   DURATION   AGE
copy-content-job Complete   1/1            6s         8s
bk@MacBook-Pro-BartomiejK Lab_5 (04.11.2024) %

```

7. Testing HTTP server



The screenshot shows a web browser window with the address bar set to 'localhost'. The page content displays 'Hello from Kubernetes!'. The browser's address bar and tabs are visible at the top, and the page content is centered below the browser interface.

GitHub

Link to the repository: [link](#)