

Jayden Lewis

Pritesh Patel

COMP 3104 - 15783

101484621

05/11/2025

Lab 08 - Docker Introduction & Kubernetes

Lab Contents

VM Created & Deployed Successfully.....	1
Connecting To VM.....	2
Successfully Installed JDK & Exited VM.....	3
\$ minikube version, status, & start.....	4
\$ minikube dashboard.....	4
Kubernetes Dashboard.....	5
Kubectl Commands.....	5
\$ docker version.....	6
\$ docker run hello-world.....	6
\$ docker images.....	7
\$ docker ps.....	7

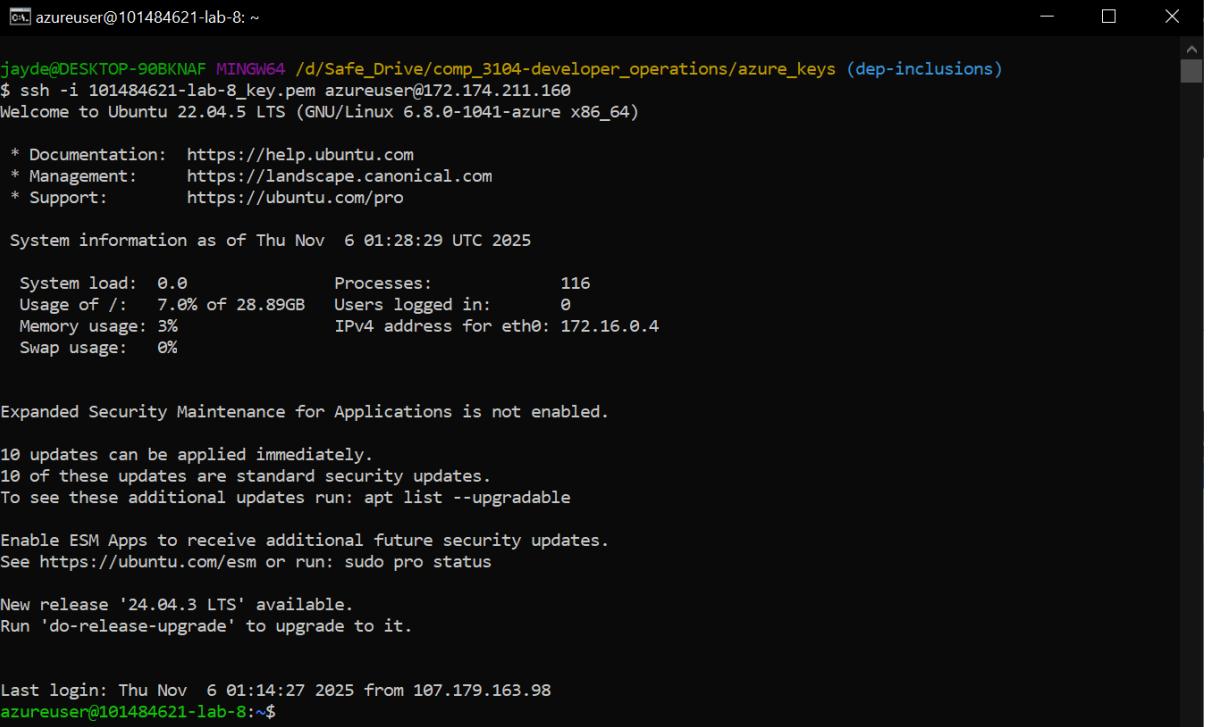
VM Created & Deployed Successfully

The screenshot shows the Microsoft Azure portal's deployment overview for a completed deployment named "CreateVm-canonical.0001-com-ubuntu-server-jammy-2-20251105200132". The deployment summary indicates it was completed on 2025-11-05 at 8:06:37 p.m. with a Correlation ID of e57a93b5-9729-4114-aac3-a749045270b4. The deployment details table lists five resources:

Resource	Type	Status	Operation details
101484621-lab-8	Virtual machine	OK	Operation details
101484621-lab-8530_z1	Microsoft.Network/networkInterface	OK	Operation details
101484621-lab-8-ip	Public IP address	OK	Operation details
101484621-lab-8-nsg	Network security group	OK	Operation details
network-interface-associated-virtual-network-20251105200611	Deployment	OK	Operation details

Under the "Next steps" section, there are three recommended actions: "Set up auto-shutdown", "Monitor VM health, performance, and network dependencies", and "Run a script inside the virtual machine". At the bottom, there are buttons for "Go to resource" and "Create another VM".

Connecting To VM



The screenshot shows a terminal window with the following content:

```
azureuser@101484621-lab-8: ~
jayde@DESKTOP-90BKN4F MINGW64 /d/Safe_Drive/comp_3104-developer_operations/azure_keys (dep-inclusions)
$ ssh -i 101484621-lab-8_key.pem azureuser@172.174.211.160
Welcome to Ubuntu 22.04.5 LTS (GNU/Linux 6.8.0-1041-azure x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:     https://landscape.canonical.com
 * Support:        https://ubuntu.com/pro

System information as of Thu Nov  6 01:28:29 UTC 2025

System load:  0.0          Processes:           116
Usage of /:   7.0% of 28.89GB  Users logged in:      0
Memory usage: 3%          IPv4 address for eth0: 172.16.0.4
Swap usage:   0%

Expanded Security Maintenance for Applications is not enabled.

10 updates can be applied immediately.
10 of these updates are standard security updates.
To see these additional updates run: apt list --upgradable

Enable ESM Apps to receive additional future security updates.
See https://ubuntu.com/esm or run: sudo pro status

New release '24.04.3 LTS' available.
Run 'do-release-upgrade' to upgrade to it.

Last login: Thu Nov  6 01:14:27 2025 from 107.179.163.98
azureuser@101484621-lab-8:~$
```

Successfully Installed JDK & Exited VM

```
[ca] MINGW64:/d/Safe_Drive/comp_3104-developer_operations/azure_keys
Adding debian:USERTrust_RSA_Certification_Authority.pem
Adding debian:Microsoft_ECC_Root_Certificate_Authority_2017.pem
Adding debian:HiPKI_Root_CA_-_G1.pem
Adding debian:Certigna_Root_CA.pem
Adding debian:Sectigo_Public_Server_Authentication_Root_R46.pem
Adding debian:Buypass_Class_2_Root_CA.pem
Adding debian:QuoVadis_Root_CA_2_G3.pem
Adding debian:emSign_Root_CA_-_G1.pem
Adding debian:TUBITAK_Kamu_SM_SSL_Kok_Sertifikasi_-_Surum_1.pem
Adding debian:TWCA_Global_Root_CA.pem
Adding debian:CommScope_Public_Trust_RSA_Root-02.pem
Adding debian:vTrus_Root_CA.pem
Adding debian:SwissSign_Gold_CA_-_G2.pem
Adding debian:emSign_ECC_Root_CA_-_C3.pem
Adding debian:SSL.com_Root_Certification_Authority_RSA.pem
Adding debian:GTS_Root_R2.pem
Adding debian:QuoVadis_Root_CA_3_G3.pem
Adding debian:T-TeleSec_GlobalRoot_Class_3.pem
done.

Processing triggers for libc-bin (2.35-0ubuntu3.11) ...
Processing triggers for man-db (2.10.2-1) ...
Processing triggers for ca-certificates (20240203~22.04.1) ...
Updating certificates in /etc/ssl/certs...
0 added, 0 removed; done.
Running hooks in /etc/ca-certificates/update.d...

done.
done.
Setting up openjdk-21-jre-headless:amd64 (21.0.8+9~us1-0ubuntu1~22.04.1) ...
update-alternatives: using /usr/lib/jvm/java-21-openjdk-amd64/bin/java to provide /usr/bin/java (java) in auto mode
update-alternatives: using /usr/lib/jvm/java-21-openjdk-amd64/bin/jpackage to provide /usr/bin/jpackage (jpackage) in auto mode
update-alternatives: using /usr/lib/jvm/java-21-openjdk-amd64/bin/keytool to provide /usr/bin/keytool (keytool) in auto mode
update-alternatives: using /usr/lib/jvm/java-21-openjdk-amd64/bin/rmiregistry to provide /usr/bin/rmiregistry (rmiregistry) in auto mode
update-alternatives: using /usr/lib/jvm/java-21-openjdk-amd64/lib/jexec to provide /usr/bin/jexec (jexec) in auto mode
Scanning processes...
Scanning linux images...

Running kernel seems to be up-to-date.

No services need to be restarted.

No containers need to be restarted.

No user sessions are running outdated binaries.

No VM guests are running outdated hypervisor (qemu) binaries on this host.
azureuser@101484621-lab-8:~$ exit
logout
Connection to 172.174.211.160 closed.

jayde@DESKTOP-90BKN4F MINGW64 /d/Safe_Drive/comp_3104-developer_operations/azure_keys (dep-inclusions)
$
```

\$ minikube version, status, & start

```
jayde@DESKTOP-90BKN4F MINGW64 /d/Safe_Drive/comp_3104-developer_operations/azure_keys
$ minikube version
minikube version: v1.37.0
commit: 65318f4cfffc12cc87ec9eb8f4cdd57b25047f3

jayde@DESKTOP-90BKN4F MINGW64 /d/Safe_Drive/comp_3104-developer_operations/azure_keys (dep-inclusions)
$ minikube status
[?] Profile "minikube" not found. Run "minikube profile list" to view all profiles.
[?] To start a cluster, run: "minikube start"

jayde@DESKTOP-90BKN4F MINGW64 /d/Safe_Drive/comp_3104-developer_operations/azure_keys (dep-inclusions)
$ minikube start
[?] minikube v1.37.0 on Microsoft Windows 10 Pro 10.0.19045.6456 Build 19045.6456
[?] Automatically selected the docker driver. Other choices: virtualbox, ssh
[?] Using Docker Desktop driver with root privileges
[?] Starting "minikube" primary control-plane node in "minikube" cluster
[?] Pulling base image v0.0.48 ...
[?] Downloading Kubernetes v1.34.0 preload ...
  > gcr.io/k8s-minikube/kicbase...: 488.52 MiB / 488.52 MiB 100.00% 13.39 M
  > preloaded-images-k8s-v18-v1...: 337.07 MiB / 337.07 MiB 100.00% 6.68 Mi
[?] Creating docker container (CPUs=2, Memory=4000MB) ...
[?] Failing to connect to https://registry.k8s.io/ from inside the minikube container
[?] To pull new external images, you may need to configure a proxy: https://minikube.sigs.k8s.io/docs/reference/networking/proxy/
[?] Preparing Kubernetes v1.34.0 on Docker 28.4.0 ...
[?] Configuring bridge CNI (Container Networking Interface) ...
[?] Verifying Kubernetes components...
  • Using image gcr.io/k8s-minikube/storage-provisioner:v5
[?] Enabled addons: storage-provisioner, default-storageclass

[?] C:\Program Files\Docker\Dockers\resources\bin\kubectl.exe is version 1.32.2, which may have incompatibilities with Kubernetes 1.34.0.
  • Want kubectl v1.34.0? Try 'minikube kubectl -- get pods -A'
[?] Done! kubectl is now configured to use "minikube" cluster and "default" namespace by default
```

\$ minikube dashboard

```
jayde@DESKTOP-90BKN4F MINGW64 /d/Safe_Drive/comp_3104-developer_operations/azure_keys
$ minikube dashboard
[?] Enabling dashboard ...
  • Using image docker.io/kubernetesui/metrics-scraper:v1.0.8
  • Using image docker.io/kubernetesui/dashboard:v2.7.0
[?] Some dashboard features require the metrics-server addon. To enable all features please run:

  minikube addons enable metrics-server

[?] Verifying dashboard health ...
[?] Launching proxy ...
[?] Verifying proxy health ...
[?] Opening http://127.0.0.1:64594/api/v1/namespaces/kubernetes-dashboard/services/http:kubernetes-dashboard:/proxy/ in your default browser...
```

Kubernetes Dashboard

The screenshot shows the Kubernetes Dashboard interface. At the top, there's a navigation bar with tabs for Compute infrastructure, minikube start | minikube, and Kubernetes Dashboard. Below the navigation bar is a header with the Kubernetes logo and a dropdown menu set to 'default'. A search bar is also present. The main content area has a blue header bar with the title 'Workloads'. Underneath, a sidebar lists various Kubernetes resources: Cron Jobs, Daemon Sets, Deployments, Jobs, Pods, Replica Sets, Replication Controllers, Stateful Sets, Service, Ingresses, Ingress Classes, Services, Config and Storage, Config Maps, Persistent Volume Claims, Secrets, and Storage Classes. The main panel displays a message: 'There is nothing to display here' and a note: 'You can deploy a containerized app, select other namespace or take the Dashboard Tour to learn more.'

Kubectl Commands

```
jayde@DESKTOP-90BKNFAF MINGW64 /d/Safe_Drive/comp_3104-developer_operations/azure_keys
$ kubectl version
Client Version: v1.32.2
Kustomize Version: v5.5.0
Server Version: v1.34.0
WARNING: version difference between client (1.32) and server (1.34) exceeds the supported minor version skew of +/-1

jayde@DESKTOP-90BKNFAF MINGW64 /d/Safe_Drive/comp_3104-developer_operations/azure_keys (dep-inclusions)
$ kubectl cluster-info
Kubernetes control plane is running at https://127.0.0.1:64476
CoreDNS is running at https://127.0.0.1:64476/api/v1/namespaces/kube-system/services/kube-dns:dns/proxy

To further debug and diagnose cluster problems, use 'kubectl cluster-info dump'.

jayde@DESKTOP-90BKNFAF MINGW64 /d/Safe_Drive/comp_3104-developer_operations/azure_keys (dep-inclusions)
$ kubectl get namespaces
NAME          STATUS   AGE
default       Active   2m49s
kube-node-lease Active   2m49s
kube-public   Active   2m49s
kube-system   Active   2m49s
kubernetes-dashboard Active  2m12s

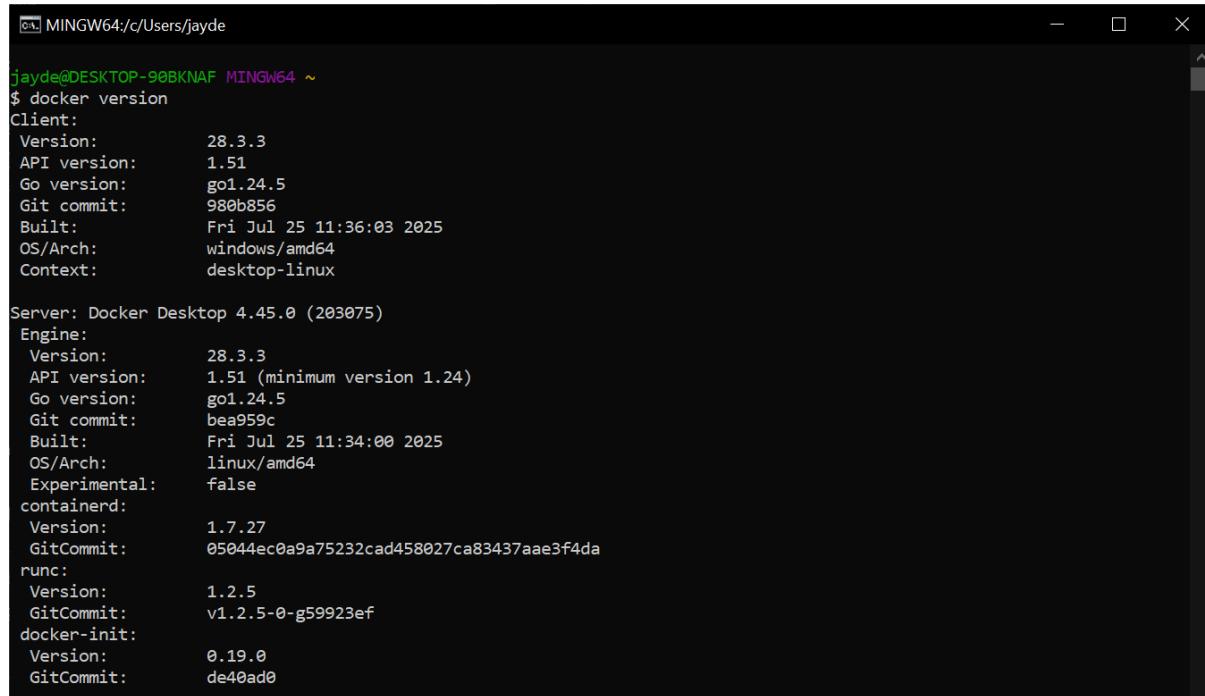
jayde@DESKTOP-90BKNFAF MINGW64 /d/Safe_Drive/comp_3104-developer_operations/azure_keys (dep-inclusions)
$ kubectl get pods
No resources found in default namespace.

jayde@DESKTOP-90BKNFAF MINGW64 /d/Safe_Drive/comp_3104-developer_operations/azure_keys (dep-inclusions)
$ kubectl get svc
NAME      TYPE      CLUSTER-IP    EXTERNAL-IP    PORT(S)    AGE
kubernetes  ClusterIP  10.96.0.1    <none>        443/TCP   3m4s

jayde@DESKTOP-90BKNFAF MINGW64 /d/Safe_Drive/comp_3104-developer_operations/azure_keys (dep-inclusions)
$ kubectl get deploy
No resources found in default namespace.

jayde@DESKTOP-90BKNFAF MINGW64 /d/Safe_Drive/comp_3104-developer_operations/azure_keys (dep-inclusions)
$ kubectl get nodes
NAME      STATUS    ROLES     AGE      VERSION
minikube  Ready     control-plane  3m19s   v1.34.0
```

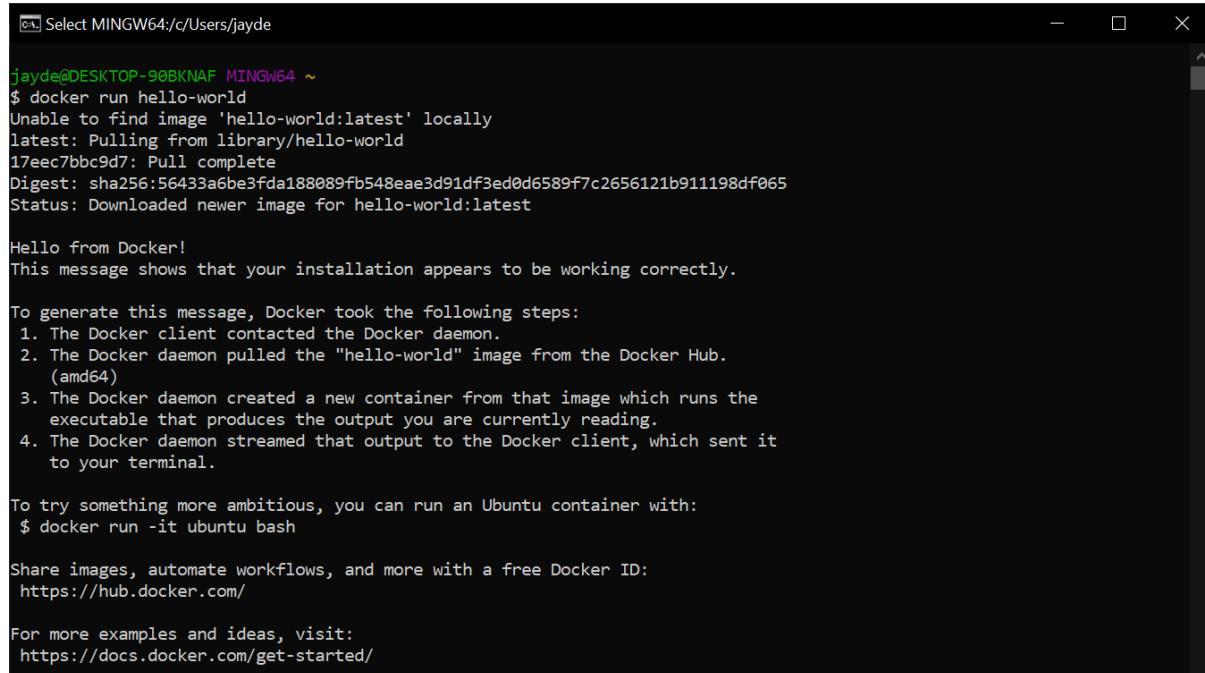
\$ docker version



```
jayde@DESKTOP-90BKN4F MINGW64 ~
$ docker version
Client:
Version:          28.3.3
API version:      1.51
Go version:       go1.24.5
Git commit:       980b856
Built:            Fri Jul 25 11:36:03 2025
OS/Arch:          windows/amd64
Context:          desktop-linux

Server: Docker Desktop 4.45.0 (203075)
Engine:
Version:          28.3.3
API version:      1.51 (minimum version 1.24)
Go version:       go1.24.5
Git commit:       bea959c
Built:            Fri Jul 25 11:34:00 2025
OS/Arch:          linux/amd64
Experimental:     false
containerd:
Version:          1.7.27
GitCommit:        05044ec0a9a75232cad458027ca83437aae3f4da
runc:
Version:          1.2.5
GitCommit:        v1.2.5-0-g59923ef
docker-init:
Version:          0.19.0
GitCommit:        de40ad0
```

\$ docker run hello-world



```
jayde@DESKTOP-90BKN4F MINGW64 ~
$ docker run hello-world
Unable to find image 'hello-world:latest' locally
latest: Pulling from library/hello-world
17eec7bbc9d7: Pull complete
Digest: sha256:56433a6be3fd188089fb548ae3d91df3ed0d6589f7c2656121b911198df065
Status: Downloaded newer image for hello-world:latest

Hello from Docker!
This message shows that your installation appears to be working correctly.

To generate this message, Docker took the following steps:
 1. The Docker client contacted the Docker daemon.
 2. The Docker daemon pulled the "hello-world" image from the Docker Hub.
    (amd64)
 3. The Docker daemon created a new container from that image which runs the
    executable that produces the output you are currently reading.
 4. The Docker daemon streamed that output to the Docker client, which sent it
    to your terminal.

To try something more ambitious, you can run an Ubuntu container with:
$ docker run -it ubuntu bash

Share images, automate workflows, and more with a free Docker ID:
https://hub.docker.com/

For more examples and ideas, visit:
https://docs.docker.com/get-started/
```

\$ docker images

```
[jayde@DESKTOP-90BKN4F MINGW64 ~]$ docker images
REPOSITORY          TAG      IMAGE ID      CREATED       SIZE
postgres-local-wellness-service    latest   609c5b60477d  43 minutes ago  795MB
comp3095-integrated-wellness-service    latest   1e778241f024  43 minutes ago  795MB
product-services      latest   fc03fe659c2b  25 hours ago   733MB
inventory-service     latest   b1cd84abb0ef  26 hours ago   781MB
order-service        latest   062c30fb64ea  27 hours ago   921MB
postgres             16      21f6013073bc  45 hours ago   635MB
postgres             latest   6f3e42ad37de  45 hours ago   643MB
mongo                latest   86835e8da0f9  8 days ago    1.24GB
postgres             15-alpine 64583b3cb4f2  3 weeks ago    390MB
hello-world          latest   56433a6be3fd  2 months ago   20.3kB
eclipse-temurin      21-jdk   7d1d666ddafa  3 months ago   675MB
testcontainers/ryuk   0.12.0   dd3f023a6ed7  5 months ago   29MB
redis                7.4.3    236e397c1d5a  6 months ago   173MB
dpage/pgadmin4        9.2      52cb72a9e3da  7 months ago   774MB
mongo-express        latest   1b23d7976f02  20 months ago  286MB
redislabs/redisinsight 1.14.0   b03ab1426d0d  2 years ago    1.82GB
```

\$ docker ps

```
[jayde@DESKTOP-90BKN4F MINGW64 ~]$ docker ps
CONTAINER ID   IMAGE           COMMAND                  CREATED          STATUS          PORTS          NAMES
dsc87ce14d45   dpage/pgadmin4:9.2   "/entrypoint.sh"   27 minutes ago   Up 27 minutes   0.0.0.0:8888->80/tcp, [::]:8888->80/tcp   wellness-service-pgadmin
aec9ed84ac84   postgres-local-wellness-service   "java -jar /app/well..."  27 minutes ago   Up 27 minutes   0.0.0.0:8081->8081/tcp, [::]:8081->8081/tcp   wellness-service
feec90f61842   redislabs/redisinsight:1.14.0   "bash ./docker-entry..."  27 minutes ago   Up 27 minutes   0.0.0.0:8001->8001/tcp, [::]:8001->8001/tcp   redisinsight
27e2446beb36   postgres:16            "docker-entrypoint.s..."  27 minutes ago   Up 27 minutes   0.0.0.0:5435->5432/tcp, [::]:5435->5432/tcp   wellness-service-postgres
853a063ae13c   redis:7.4.3         "docker-entrypoint.s..."  27 minutes ago   Up 27 minutes   0.0.0.0:6379->6379/tcp, [::]:6379->6379/tcp   redis
```