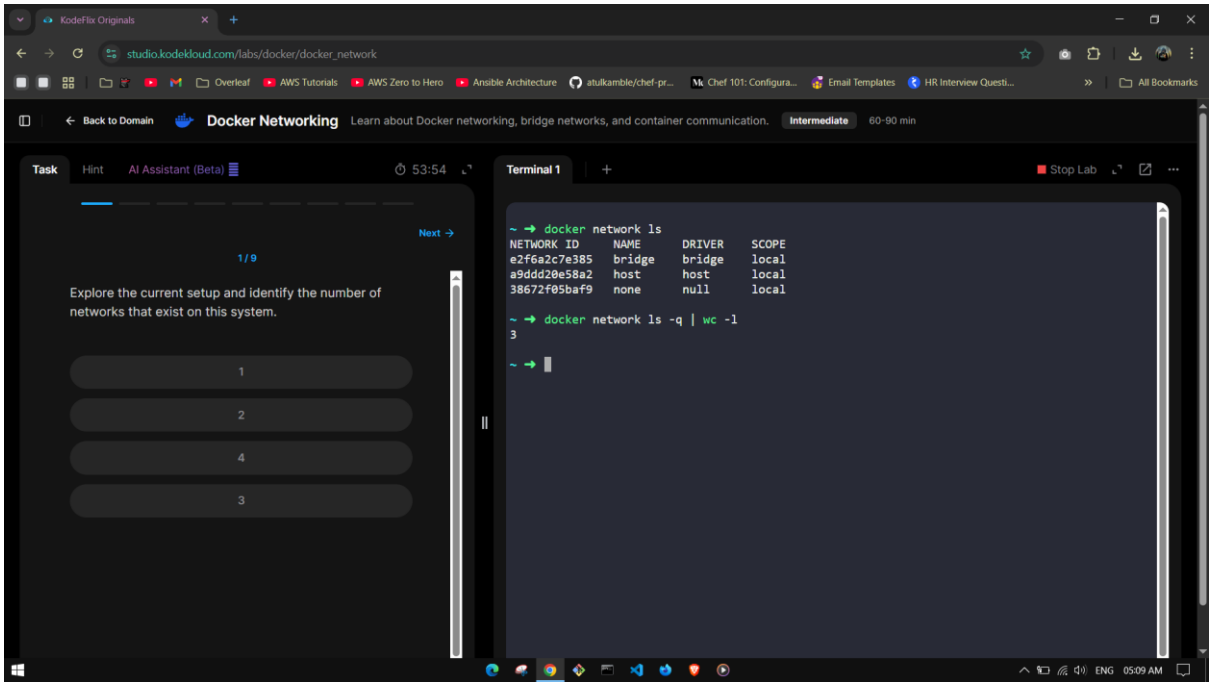
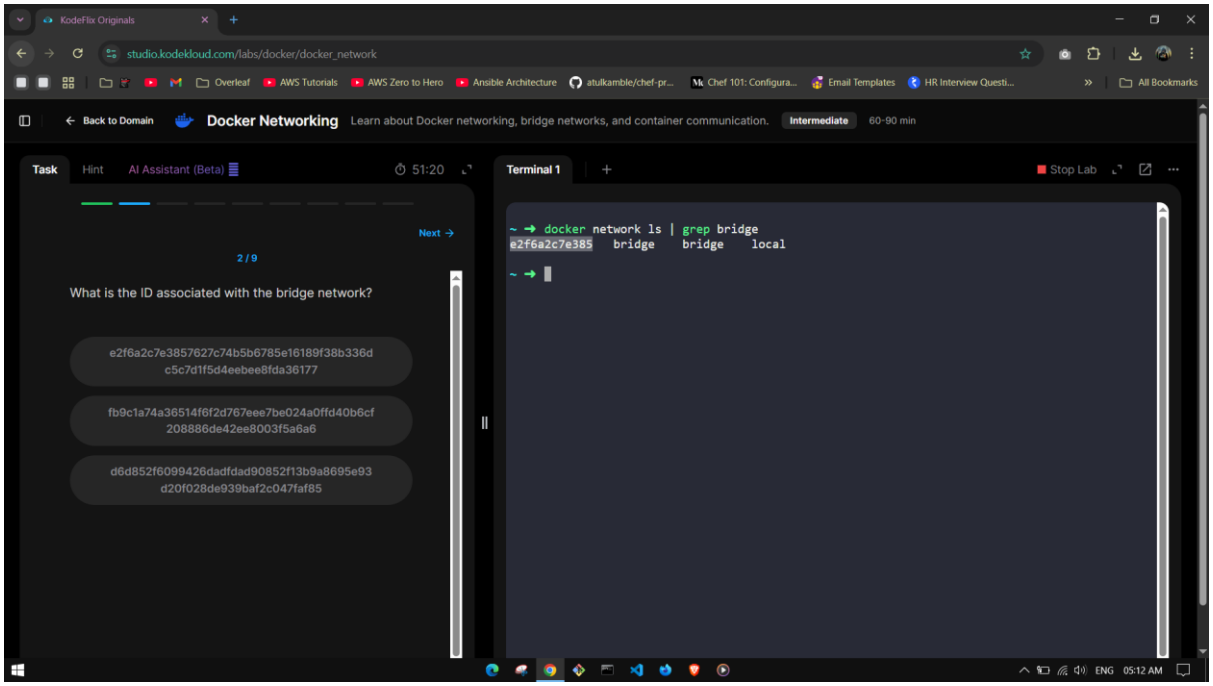
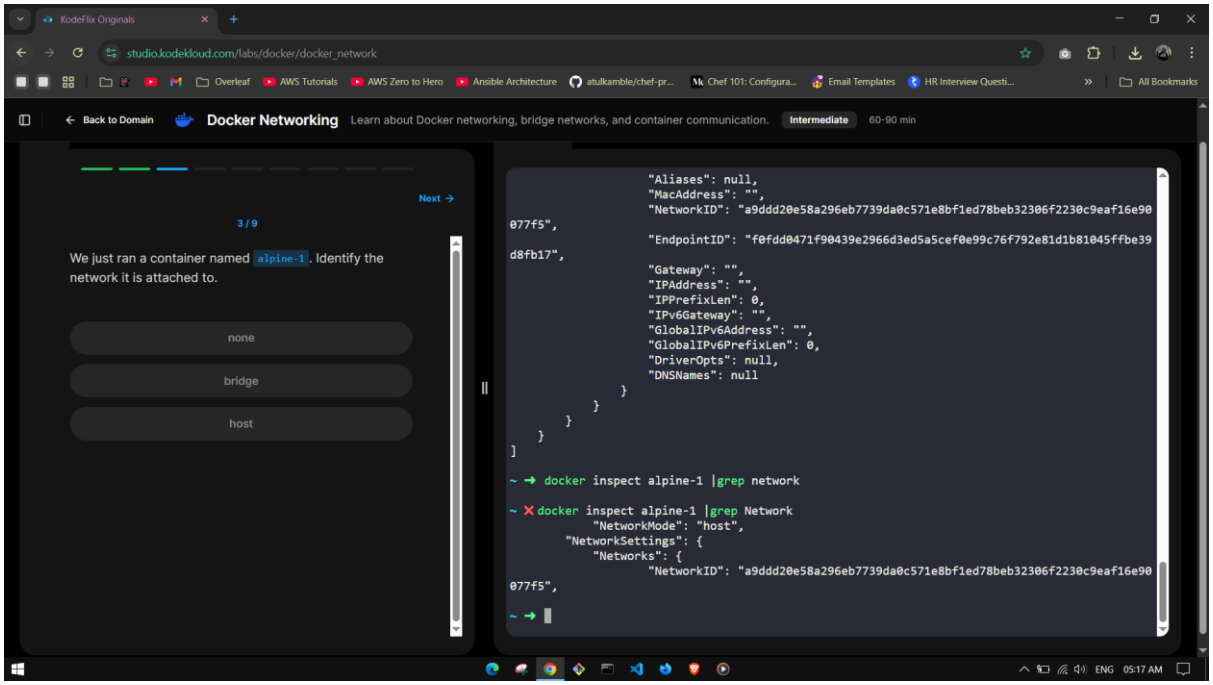
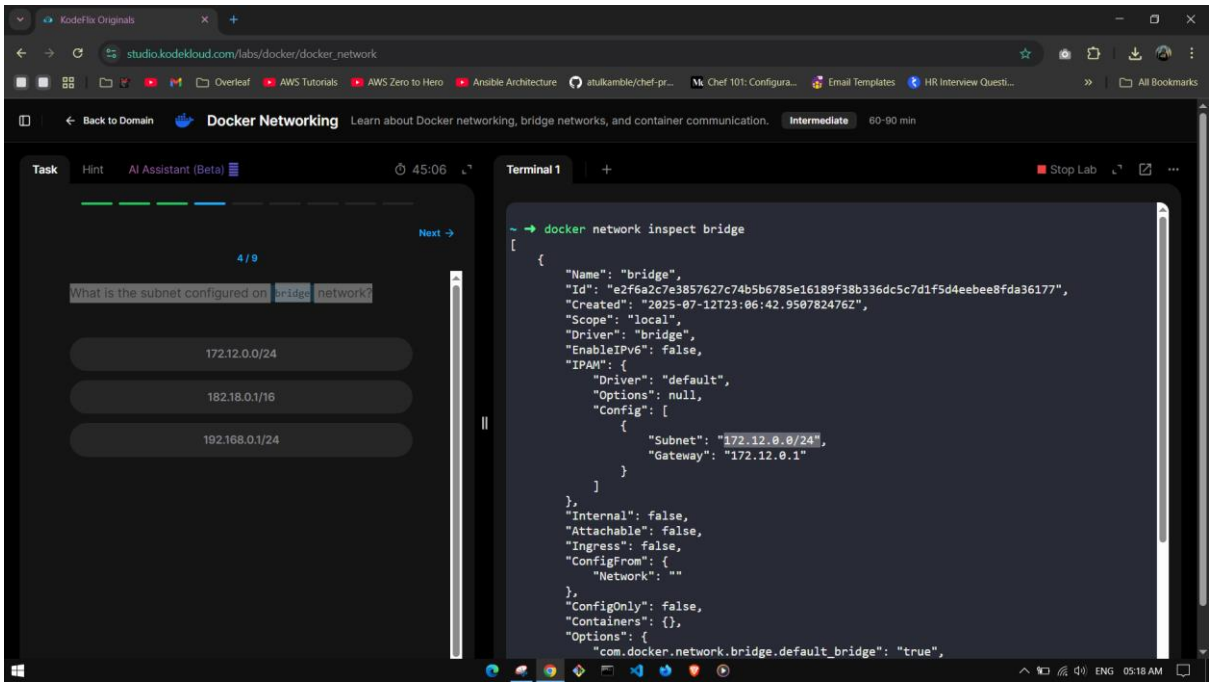
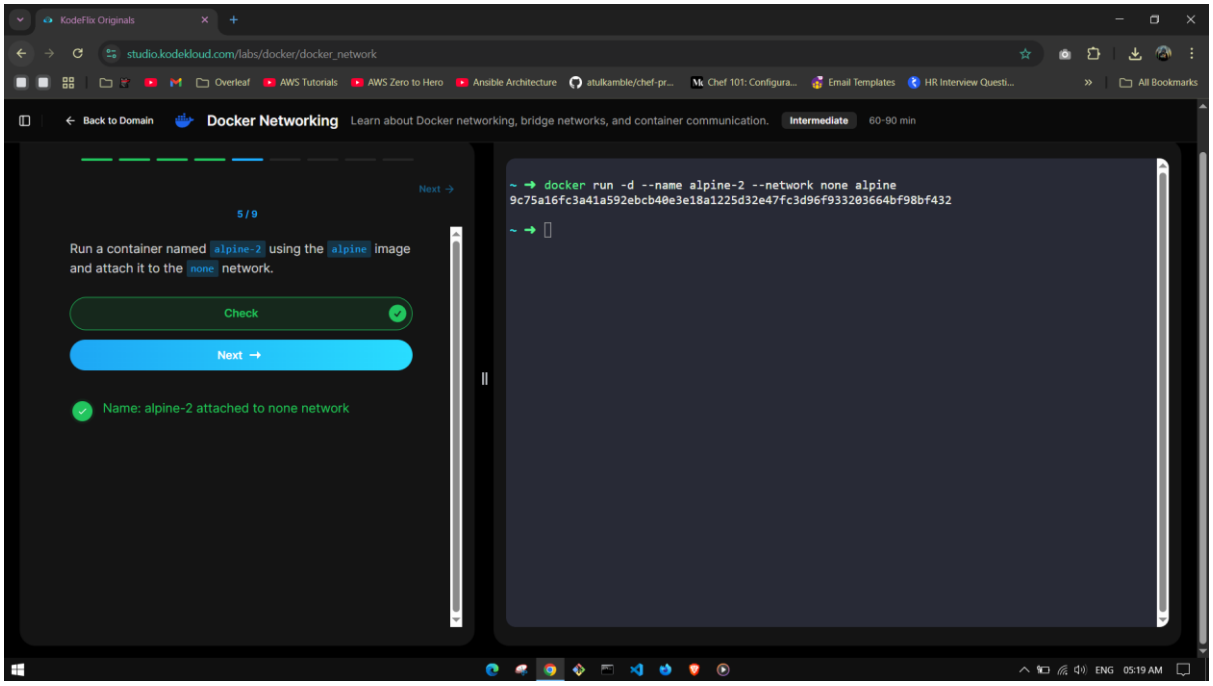
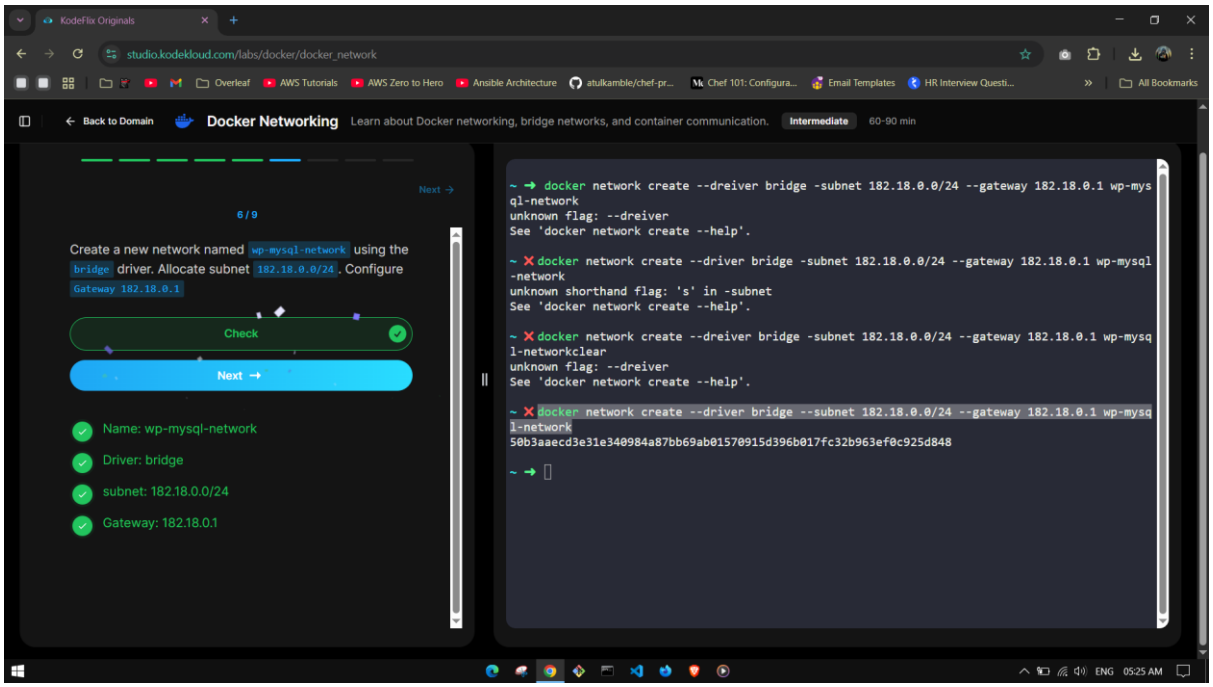

KodeFlix Originals
studio.kodekloud.com/labs/docker/docker_network
Back to Domain
Docker Networking
Learn about Docker networking, bridge networks, and container communication.
Intermediate
60-90 min
Task
Hint
AI Assistant (Beta)
53:54
Next
1 / 9
Explore the current setup and identify the number of networks that exist on this system.
1
2
4
3
Terminal 1
Stop Lab
~ ➜ docker network ls
NETWORK ID NAME DRIVER SCOPE
e2f6a2c7e385 bridge bridge local
a9ddd20e58a2 host host local
38672f05baf9 none null local
~ ➜ docker network ls -q | wc -l
3
~ ➜
ENG 05:09 AM

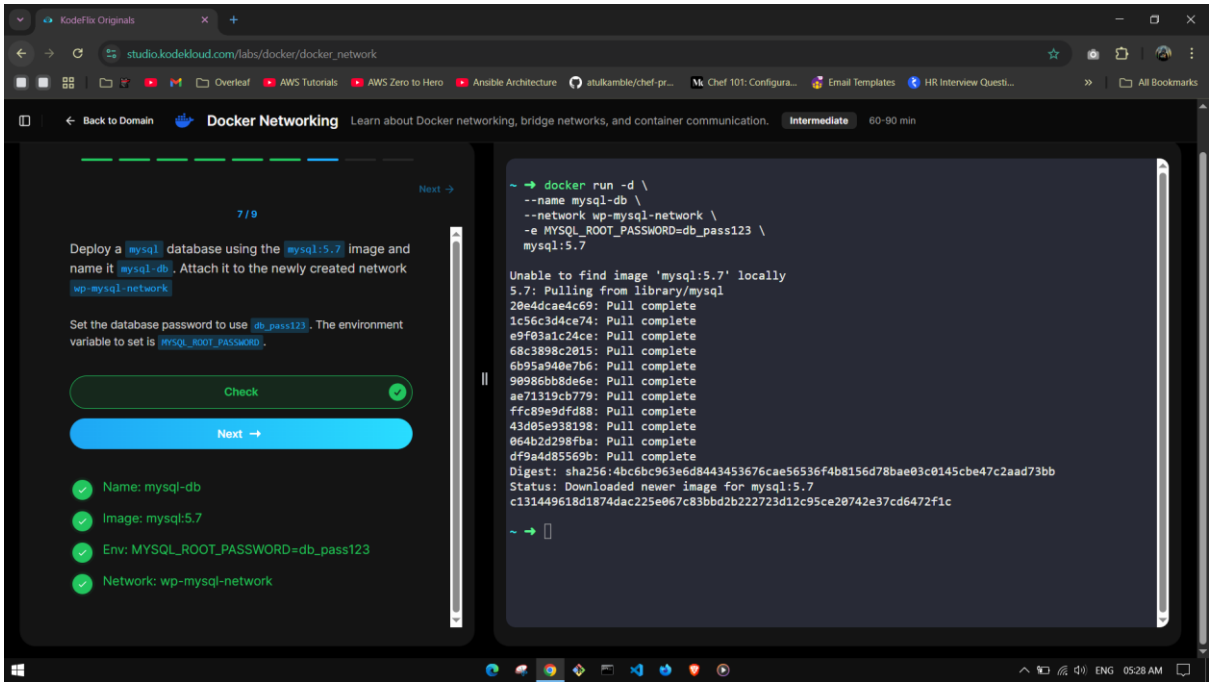

KodeFlix Originals
studio.kodekloud.com/labs/docker/docker_network
Back to Domain
Docker Networking
Learn about Docker networking, bridge networks, and container communication.
Intermediate
60-90 min
Task
Hint
AI Assistant (Beta)
51:20
Next
2 / 9
What is the ID associated with the bridge network?
e2f6a2c7e3857627c74b5b6785e16189f38b336d c5c7d1f5d4eebee8fda36177
fb9c1a74a36514f6f2d767eee7be024a0ffd40b6cf 208886de42ee8003f5a6a6
d6d852f6099426dadfdad90852f13b9a8695e93 d20f028de939baf2c047faf85
Terminal 1
Stop Lab
~ ➜ docker network ls | grep bridge
e2f6a2c7e385 bridge bridge local
~ ➜
ENG 05:12 AM

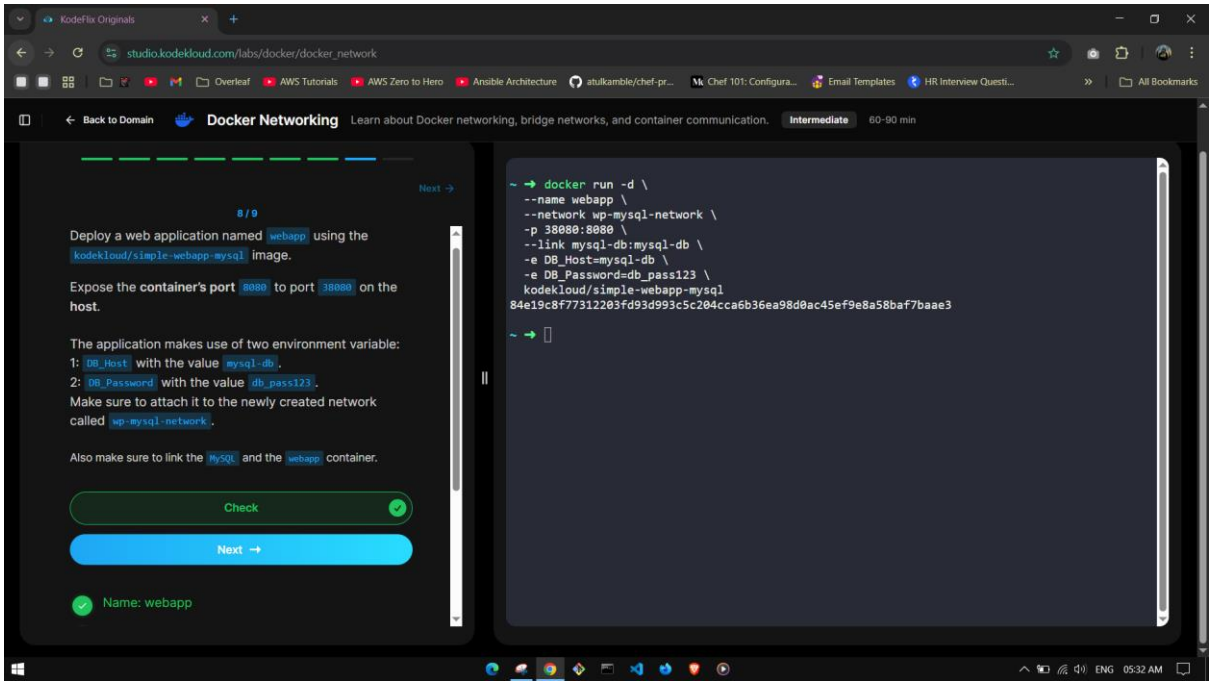

KodeFlix Originals
studio.kodekloud.com/labs/docker/docker_network
Back to Domain
Docker Networking
Learn about Docker networking, bridge networks, and container communication.
Intermediate
60-90 min
Next →
3 / 9
We just ran a container named alpine-1. Identify the network it is attached to.
none
bridge
host
"Aliases": null,
"MacAddress": "",
"NetworkID": "a9ddd20e58a296eb7739da0c571e8bf1ed78beb32306f2230c9eaf16e90077f5",
"EndpointID": "f0fdd0471f90439e2966d3ed5a5cef0e99c76f792e81d1b81045ffbe39d8fb17",
"Gateway": "",
"IPAddress": "",
"IPPrefixLen": 0,
"IPv6Gateway": "",
"GlobalIPv6Address": "",
"GlobalIPv6PrefixLen": 0,
"DriverOpts": null,
"DNSNames": null
~ ➜ docker inspect alpine-1 |grep network
~ ✗ docker inspect alpine-1 |grep Network
"NetworkMode": "host",
"NetworkSettings": {
"Networks": {
"NetworkID": "a9ddd20e58a296eb7739da0c571e8bf1ed78beb32306f2230c9eaf16e90077f5",
~ ➜
ENG 05:17 AM

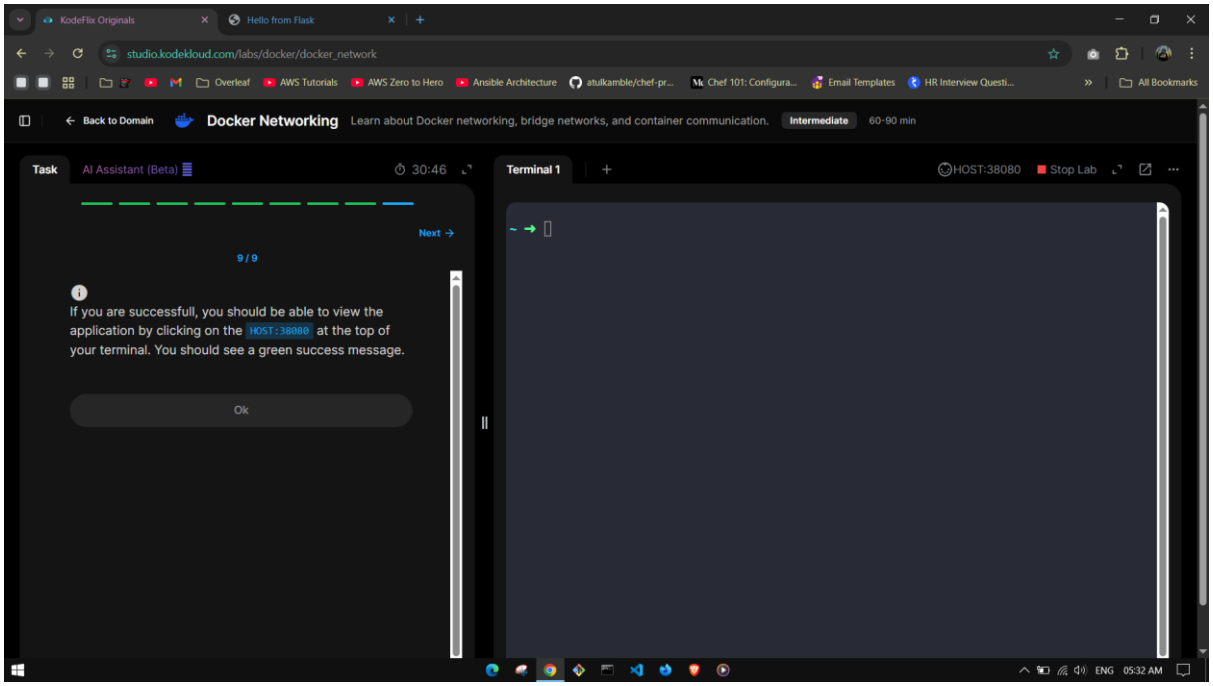

KodeFlix Originals
studio.kodekloud.com/labs/docker/docker_network
Back to Domain
Docker Networking
Learn about Docker networking, bridge networks, and container communication.
Intermediate
60-90 min
Task
Hint
AI Assistant (Beta)
45:06
Terminal 1
Stop Lab
Next →
4 / 9
What is the subnet configured on bridge network?
172.12.0.0/24
182.18.0.1/16
192.168.0.1/24
~ ➜ docker network inspect bridge
"Name": "bridge",
"Id": "e2f6a2c7e3857627c74b5b6785e16189f38b336dc5c7d1f5d4eebee8fda36177",
"Created": "2025-07-12T23:06:42.950782476Z",
"Scope": "local",
"Driver": "bridge",
"EnableIPv6": false,
"IPAM": {
"Driver": "default",
"Options": null,
"Config": [
{
"Subnet": "172.12.0.0/24",
"Gateway": "172.12.0.1"
"Internal": false,
"Attachable": false,
"Ingress": false,
"ConfigFrom": {
"Network": ""
"ConfigOnly": false,
"Containers": {},
"Options": {
"com.docker.network.bridge.default_bridge": "true",
ENG 05:18 AM

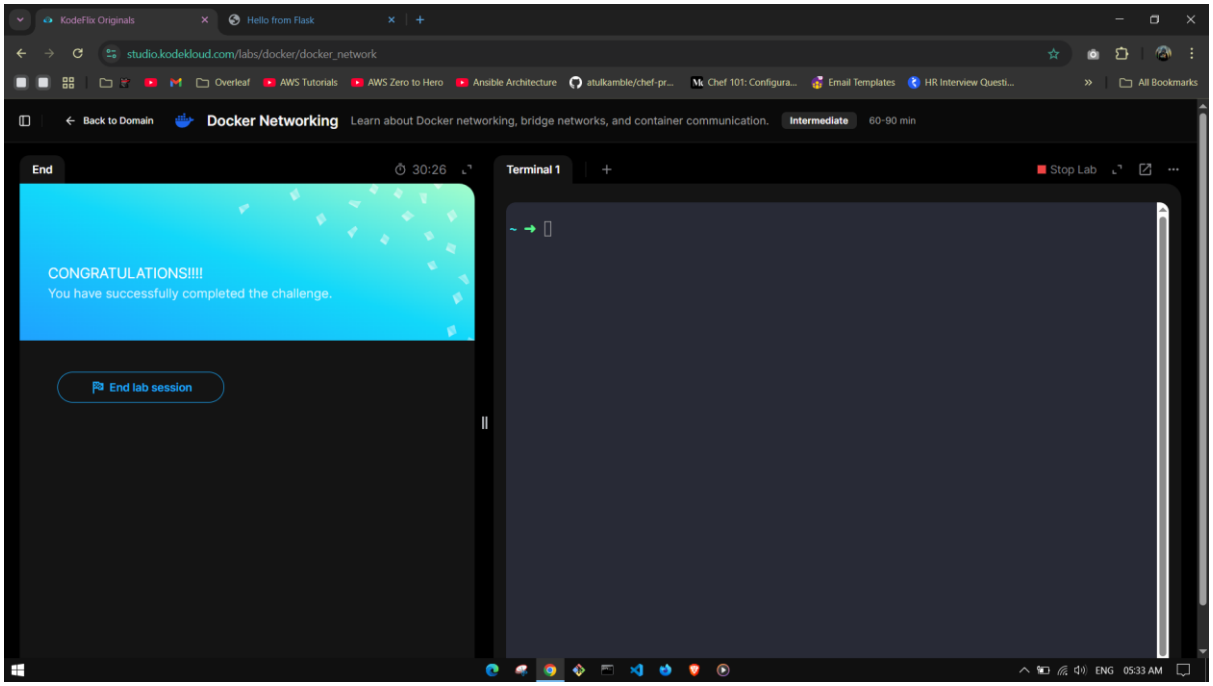

KodeFlix Originals
studio.kodekloud.com/labs/docker/docker_network
Back to Domain
Docker Networking
Learn about Docker networking, bridge networks, and container communication.
Intermediate
60-90 min
Next →
5 / 9
Run a container named alpine-2 using the alpine image and attach it to the none network.
Check
Next →
Name: alpine-2 attached to none network
~ ➜ docker run -d --name alpine-2 --network none alpine
9c75a16fc3a41a592ebcb40e3e18a1225d32e47fc3d96f933203664bf98bf432
~ ➜
ENG 05:19 AM


KodeFlix Originals
studio.kodekloud.com/labs/docker/docker_network
Back to Domain
Docker Networking
Learn about Docker networking, bridge networks, and container communication.
Intermediate
60-90 min
Next →
6 / 9
Create a new network named wp-mysql-network using the bridge driver. Allocate subnet 182.18.0.0/24. Configure Gateway 182.18.0.1
Check
Next →
Name: wp-mysql-network
Driver: bridge
subnet: 182.18.0.0/24
Gateway: 182.18.0.1
~ ➜ docker network create --dreiver bridge -subnet 182.18.0.0/24 --gateway 182.18.0.1 wp-mysql-network
unknown flag: --dreiver
See 'docker network create --help'.
~ ✗ docker network create --driver bridge -subnet 182.18.0.0/24 --gateway 182.18.0.1 wp-mysql-network
unknown shorthand flag: 's' in -subnet
See 'docker network create --help'.
~ ✗ docker network create --dreiver bridge -subnet 182.18.0.0/24 --gateway 182.18.0.1 wp-mysql-networkclear
unknown flag: --dreiver
See 'docker network create --help'.
~ ✗ docker network create --driver bridge --subnet 182.18.0.0/24 --gateway 182.18.0.1 wp-mysql-network
50b3aaecd3e31e340984a87bb69ab01570915d396b017fc32b963ef0c925d848
~ ➜
ENG 05:25 AM


KodeFlix Originals
studio.kodekloud.com/labs/docker/docker_network
Back to Domain
Docker Networking
Learn about Docker networking, bridge networks, and container communication.
Intermediate
60-90 min
Next →
7 / 9
Deploy a mysql database using the mysql:5.7 image and name it mysql-db. Attach it to the newly created network wp-mysql-network
Set the database password to use db_pass123. The environment variable to set is MYSQL_ROOT_PASSWORD.
Check
Next →
Name: mysql-db
Image: mysql:5.7
Env: MYSQL_ROOT_PASSWORD=db_pass123
Network: wp-mysql-network
~ ➜ docker run -d \
--name mysql-db \
--network wp-mysql-network \
-e MYSQL_ROOT_PASSWORD=db_pass123 \
mysql:5.7
Unable to find image 'mysql:5.7' locally
5.7: Pulling from library/mysql
20e4dcae4c69: Pull complete
1c56c3d4ce74: Pull complete
e9f03a1c24ce: Pull complete
68c3898c2015: Pull complete
6b95a940e7b6: Pull complete
90986bb8de6e: Pull complete
ae71319cb779: Pull complete
ffc89e9dfd88: Pull complete
43d05e938198: Pull complete
064b2d298fba: Pull complete
df9a4d85569b: Pull complete
Digest: sha256:4bc6bc963e6d8443453676cae56536f4b8156d78bae03c0145cbe47c2aad73bb
Status: Downloaded newer image for mysql:5.7
c131449618d1874dac225e067c83bbd2b222723d12c95ce20742e37cd6472f1c
~ ➜
ENG 05:28 AM


KodeFlix Originals
studio.kodekloud.com/labs/docker/docker_network
Back to Domain
Docker Networking
Learn about Docker networking, bridge networks, and container communication.
Intermediate
60-90 min
Next →
8 / 9
Deploy a web application named webapp using the kodekloud/simple-webapp-mysql image.
Expose the container's port 8080 to port 38080 on the host.
The application makes use of two environment variable:
1: DB_Host with the value mysql-db.
2: DB_Password with the value db_pass123.
Make sure to attach it to the newly created network called wp-mysql-network.
Also make sure to link the MySQL and the webapp container.
Check
Next →
Name: webapp
~ ➜ docker run -d \
--name webapp \
--network wp-mysql-network \
-p 38080:8080 \
--link mysql-db:mysql-db \
-e DB_Host=mysql-db \
-e DB_Password=db_pass123 \
kodekloud/simple-webapp-mysql
84e19c8f77312203fd93d993c5c204cca6b36ea98d0ac45ef9e8a58baf7baae3
~ ➜
ENG 05:32 AM


KodeFlix Originals
Hello from Flask
studio.kodekloud.com/labs/docker/docker_network
Overleaf
AWS Tutorials
AWS Zero to Hero
Ansible Architecture
atulkamble/chef-pr...
Chef 101: Configura...
Email Templates
HR Interview Questi...
All Bookmarks
Back to Domain
Docker Networking
Learn about Docker networking, bridge networks, and container communication.
Intermediate
60-90 min
Task
AI Assistant (Beta)
30:46
Next
9 / 9
If you are successfull, you should be able to view the application by clicking on the HOST:38080 at the top of your terminal. You should see a green success message.
Ok
Terminal 1
HOST:38080
Stop Lab
ENG
05:32 AM


KodeFlix Originals
Hello from Flask
studio.kodekloud.com/labs/docker/docker_network
Overleaf
AWS Tutorials
AWS Zero to Hero
Ansible Architecture
atulkamble/chef-pr...
Chef 101: Configura...
Email Templates
HR Interview Questi...
All Bookmarks
Back to Domain
Docker Networking
Learn about Docker networking, bridge networks, and container communication.
Intermediate
60-90 min
End
30:26
CONGRATULATIONS!!!!
You have successfully completed the challenge.
End lab session
Terminal 1
Stop Lab
ENG
05:33 AM