Ishita Hardasmalani

C14-2103058

EXPERIMENT NO.9

Aim: To study and implement containerization using Docker

Theory:

Docker and it's architecture:

Docker is a platform designed to make it easier to create, deploy, and run applications using containers. Containers allow developers to package an application with all of its dependencies into a standardized unit for software development. Docker's architecture is based on a client-server model and consists of the following components:

Docker Daemon: The Docker daemon, dockerd, is a persistent process that manages Docker containers and handles container lifecycle operations. It listens for Docker API requests and manages Docker objects such as images, containers, networks, and volumes.

Docker Client: The Docker client, docker, is a command-line interface (CLI) tool that allows users to interact with the Docker daemon through Docker commands. Users can use the Docker client to build, manage, and monitor Docker containers and services.

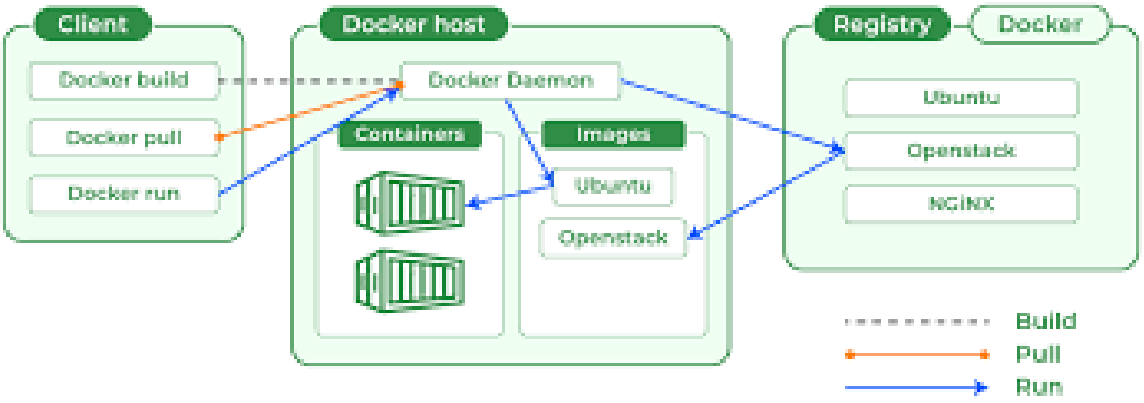
Docker Registries: Docker registries are repositories for Docker images. Docker Hub is the default public registry where users can find and share Docker images. Organizations can also set up private registries to store and distribute custom Docker images internally.

Docker Images: Docker images are read-only templates used to create Docker containers. Images contain everything needed to run an application, including the code, runtime, libraries, and dependencies. Docker images are built using Dockerfiles and can be stored and versioned in Docker registries.

Docker Containers: Docker containers are lightweight, portable, and isolated runtime environments that run applications based on Docker images. Each container encapsulates an application and its dependencies, making it easy to deploy and scale applications across different environments.

Docker Networking: Docker provides networking capabilities to connect containers to each other and to external networks. Docker networking allows containers to communicate with each other and with other services running on the same host or on remote hosts.

Docker Volumes: Docker volumes are persistent storage mechanisms used to store data generated by Docker containers. Volumes enable data sharing and data persistence between containers and can be managed and backed up independently of containers.



Benefits of Containerization:

1. Portability: Containers encapsulate applications and their dependencies, making them highly portable across different environments, such as development, testing, and production.
2. Isolation: Containers provide lightweight, isolated runtime environments, ensuring that applications and their dependencies do not interfere with each other.
3. Resource Efficiency: Containers share the host operating system kernel, which reduces overhead and resource utilization compared to virtual machines.
4. Consistency: Containers ensure consistency between development, testing, and production environments by packaging applications and dependencies into standardized units.
5. Scalability: Containers make it easy to scale applications horizontally by replicating containers across multiple hosts or vertically by increasing resource allocations for individual containers.

6. Flexibility: Containers support microservices architectures and enable developers to modularize and deploy applications as smaller, independent components.

Container: A container is a lightweight, portable, and isolated runtime environment that encapsulates an application and its dependencies. Containers share the host operating system kernel but are isolated from each other, providing consistency and resource efficiency.

Images: Docker images are read-only templates used to create Docker containers. Images contain everything needed to run an application, including the code, runtime, libraries, and dependencies. Images are built using Dockerfiles and can be stored and versioned in Docker registries.

Dockerfile: A Dockerfile is a text file that contains instructions for building a Docker image. Dockerfiles define the steps required to assemble an image layer by layer, including specifying the base image, installing dependencies, copying files, and configuring the runtime environment. Dockerfiles enable developers to automate the image-building process and ensure consistency across environments.

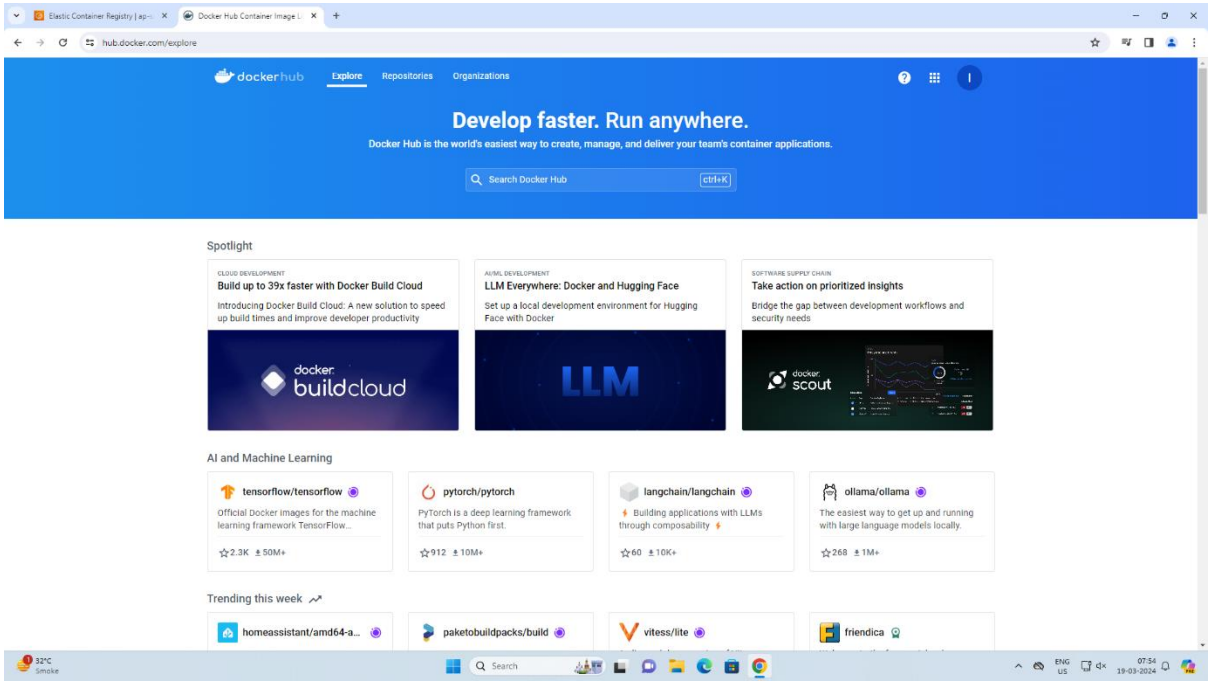
Container and Virtual Machine

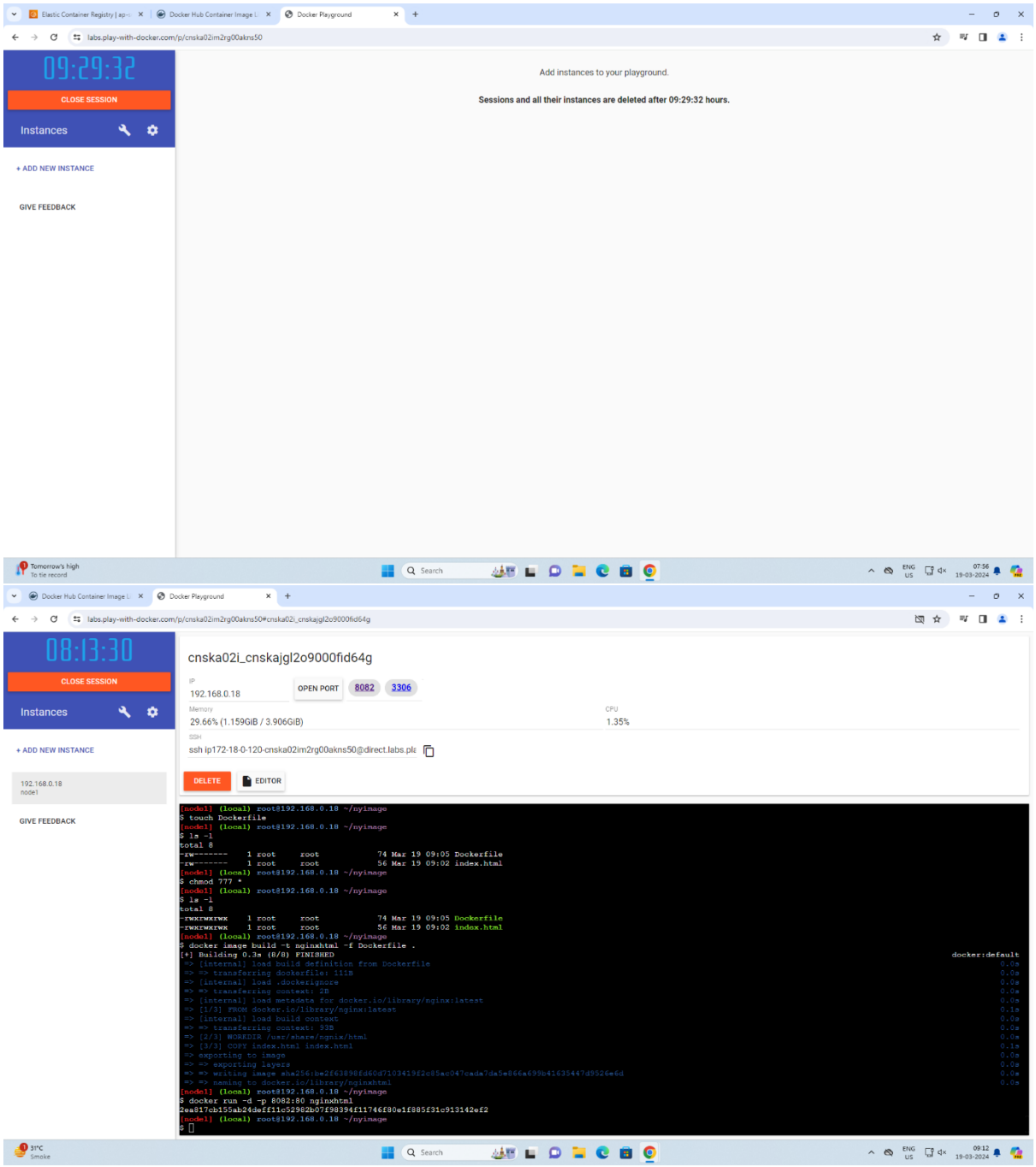
Aspect	Container	Virtual Machine
Technology	Containers use OS-level virtualization	Virtual machines use hardware-level virtualization
Isolation	Containers share the host OS kernel	Virtual machines have their own guest OS running on a hypervisor
Resource Usage	Lightweight, minimal overhead	Heavier, greater resource overhead
Startup Time	Almost instantaneous	Longer startup time due to booting a guest OS
Performance	Better performance due to shared kernel	Performance may be lower due to overhead of virtualization
Scalability	Easily scalable horizontally and vertically	Scalable but with more resource overhead
Deployment	Rapid deployment and scaling	Longer deployment time due to booting guest OS
Portability	Highly portable across different environments	Less portable due to dependencies on guest OS
Maintenance	Easier maintenance and updates	More complex maintenance and updates due to guest OS
Use Cases	Ideal for microservices architectures	Suitable for running multiple different OSes on the same hardware

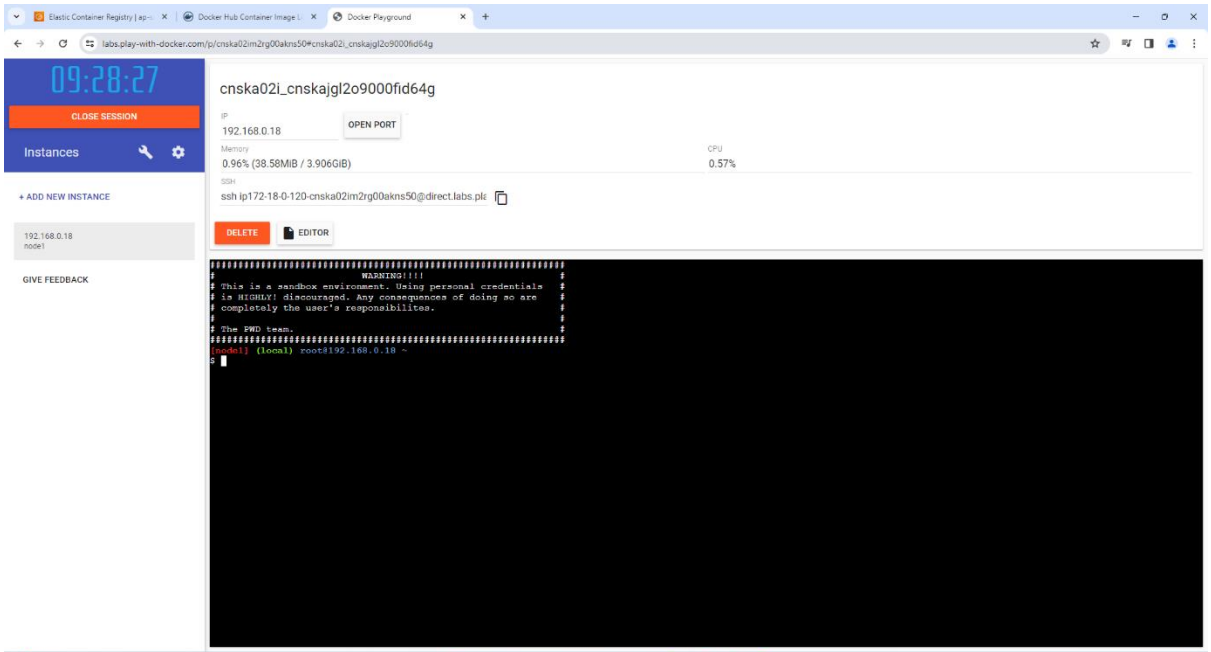
Docker Image and Docker Container

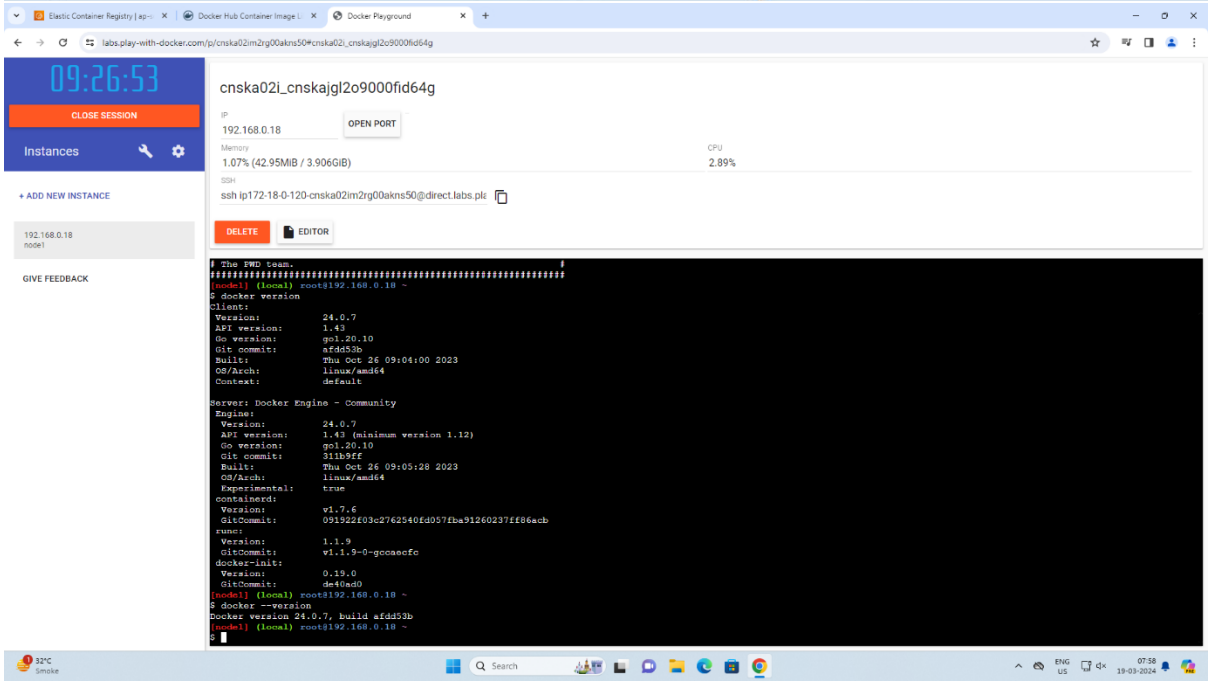
Aspect	Docker Image	Docker Container
Definition	Read-only template for creating containers.	Runnable instance of a Docker image.
Content	Contains everything needed to run an app.	Encapsulates app and its runtime.
Mutability	Immutable; changes require building a new image.	Mutable; state can be modified.
Storage	Stored in registries or locally on disk.	Temporary; exists while container is running.
Lifecycle	Created, updated, and versioned.	Started, paused, stopped, and removed.
Relationship	Basis for container creation.	Instance of an image running as a process.
Usage	Built using Dockerfiles or pulled from registries.	Created from images for application execution.
Size	Generally larger due to containing dependencies.	Smaller as it includes only runtime components.
Persistence	Immutable; can be versioned and shared.	Temporary; any changes are lost when container stops.
Example	Dockerfile builds an image.	Container created from an image via `docker run`.

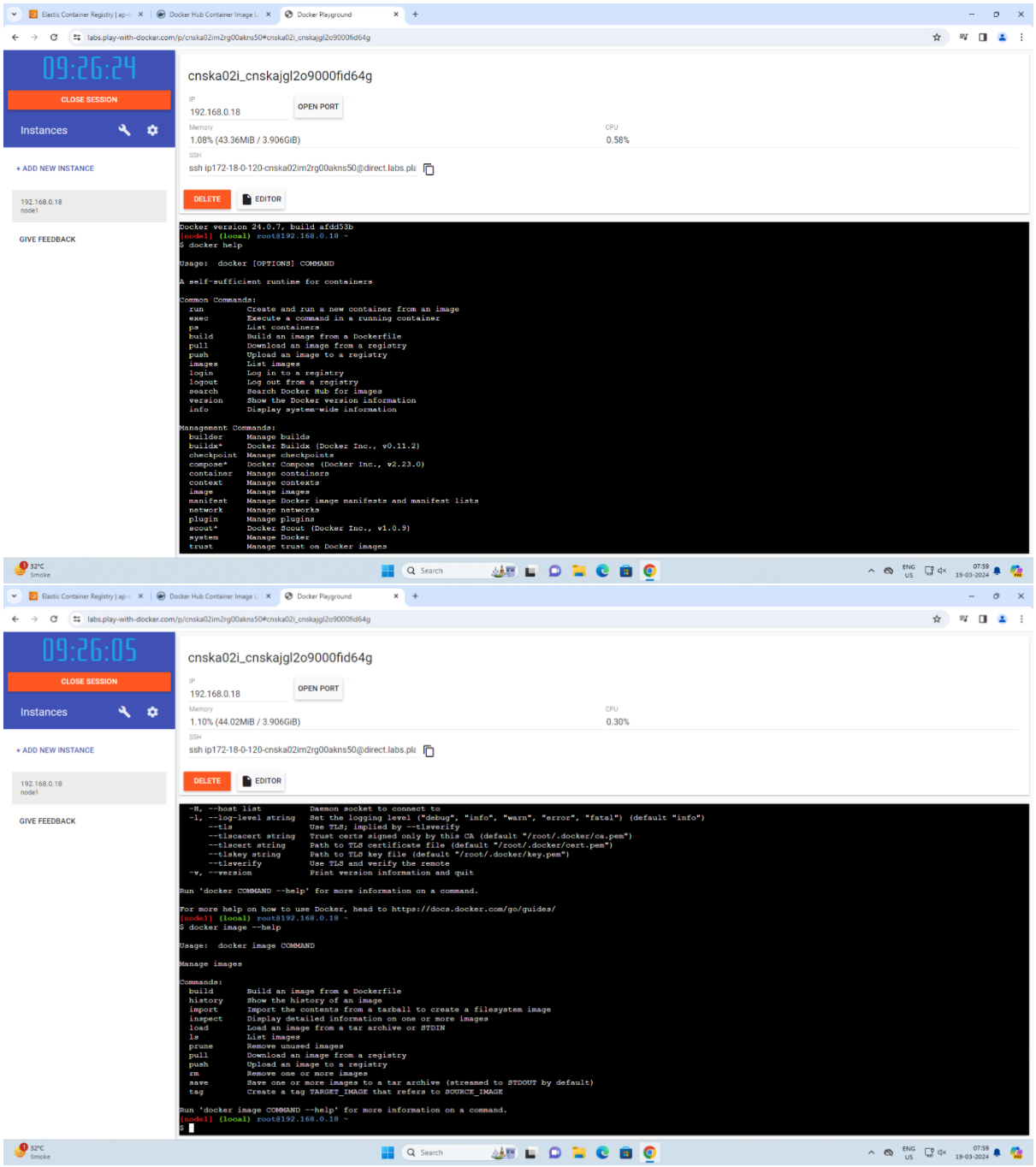
Implementation:



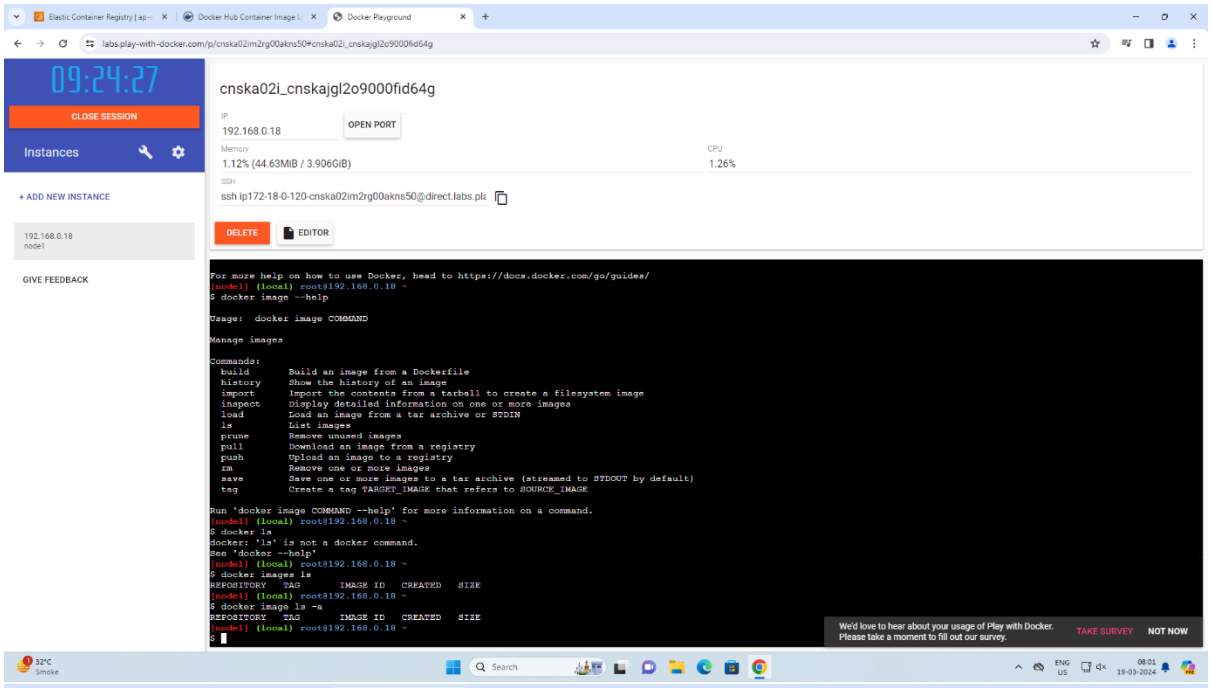

09:29:32
CLOSE SESSION
Instances
+ ADD NEW INSTANCE
GIVE FEEDBACK
Add instances to your playground.
Sessions and all their instances are deleted after 09:29:32 hours.
08:13:30
CLOSE SESSION
Instances
+ ADD NEW INSTANCE
192.168.0.18
node1
GIVE FEEDBACK
cnska02i_cnskajgl2o9000fid64g
IP
192.168.0.18
OPEN PORT
8082
3306
Memory
29.66% (1.159GiB / 3.906GiB)
CPU
1.35%
SSH
ssh ip172-18-0-120-cnska02im2rg00akns50@direct.labs.pla
DELETE
EDITOR
[node1] (local) root@192.168.0.18 ~/nyimage
$ touch Dockerfile
[node1] (local) root@192.168.0.18 ~/nyimage
$ ls -l
total 8
-rw------- 1 root root 74 Mar 19 09:05 Dockerfile
-rw------- 1 root root 56 Mar 19 09:02 index.html
[node1] (local) root@192.168.0.18 ~/nyimage
$ chmod 777 *
[node1] (local) root@192.168.0.18 ~/nyimage
$ ls -l
total 8
-rwxrwxrwx 1 root root 74 Mar 19 09:05 Dockerfile
-rwxrwxrwx 1 root root 56 Mar 19 09:02 index.html
[node1] (local) root@192.168.0.18 ~/nyimage
$ docker image build -t nginxhtml -f Dockerfile .
[+] Building 0.3s (0/0) FINISHED docker:default
=> [internal] load build definition from Dockerfile 0.0s
=> => transferring dockerfile: 111B 0.0s
=> [internal] load .dockerignore 0.0s
=> => transferring context: 2B 0.0s
=> [internal] load metadata for docker.io/library/nginx:latest 0.0s
=> [1/3] FROM docker.io/library/nginx:latest 0.1s
=> [internal] load build context 0.0s
=> => transferring context: 93B 0.0s
=> [2/3] WORKDIR /usr/share/nginx/html 0.0s
=> [3/3] COPY index.html index.html 0.1s
=> exporting to image 0.0s
=> => exporting layers 0.0s
=> => writing image sha256:be2f63898fd60d7103419f2c85ac047cada7da5e866a699b41635447d9526e6d 0.0s
=> => naming to docker.io/library/nginxhtml 0.0s
[node1] (local) root@192.168.0.18 ~/nyimage
$ docker run -d -p 8082:80 nginxhtml
2ea817cb155ab24deff11c52982b07f98394f11746f80e1f885f31c913142ef2
[node1] (local) root@192.168.0.18 ~/nyimage
$

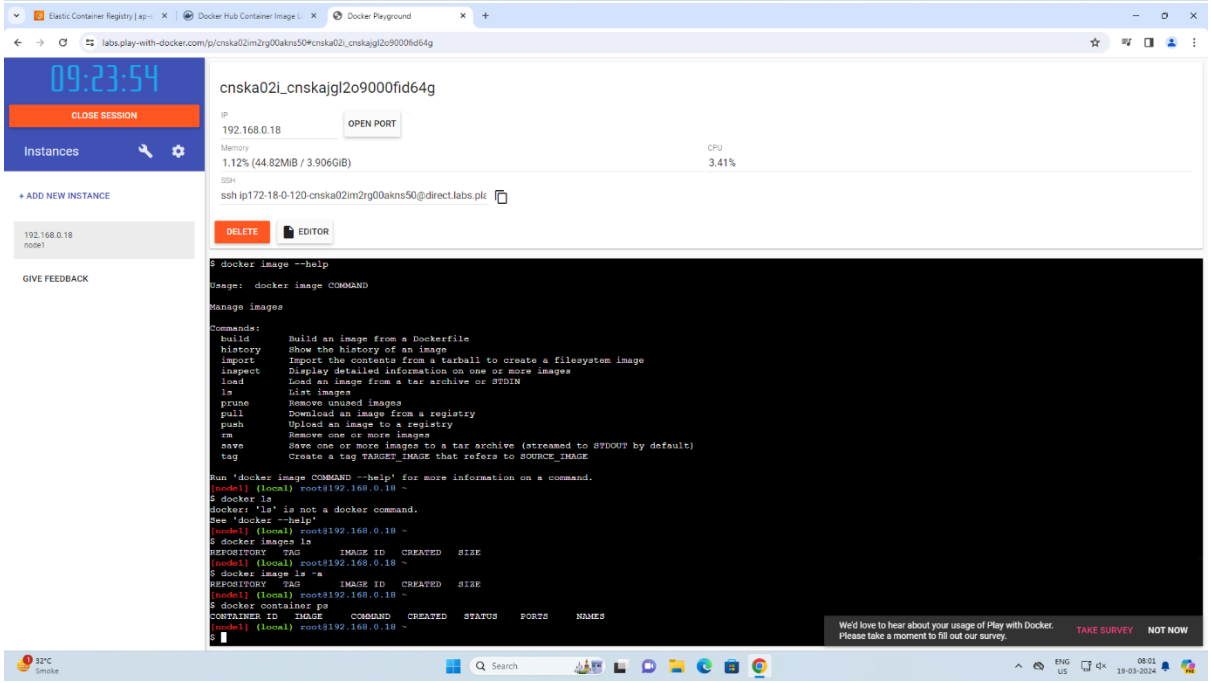

09:28:27
CLOSE SESSION
Instances
+ ADD NEW INSTANCE
192.168.0.18
node1
GIVE FEEDBACK
cnska02i_cnskajgl2o9000fid64g
IP
192.168.0.18
OPEN PORT
Memory
0.96% (38.58MiB / 3.906GiB)
CPU
0.57%
SSH
ssh ip172-18-0-120-cnska02im2rg00akns50@direct.labs.pl
DELETE
EDITOR
WARNING!!!
This is a sandbox environment. Using personal credentials
is HIGHLY! discouraged. Any consequences of doing so are
completely the user's responsibilites.
The PWD team.
[node1] (local) root@192.168.0.18 ~
$

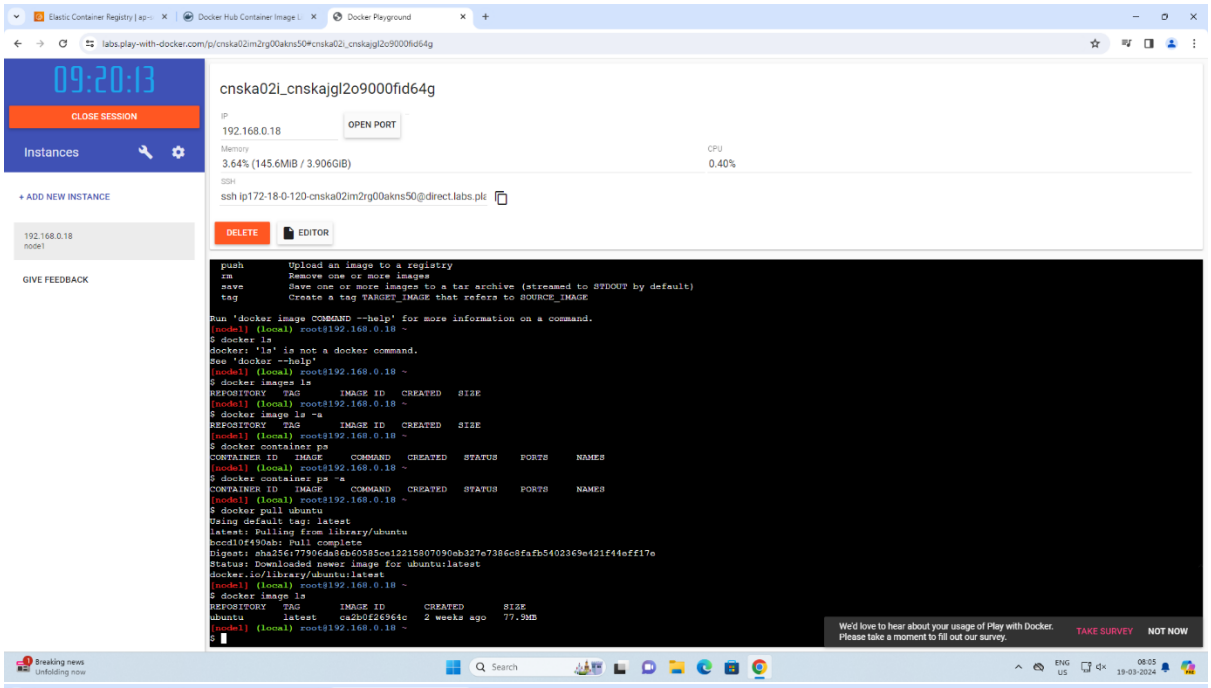

09:26:53
CLOSE SESSION
Instances
+ ADD NEW INSTANCE
192.168.0.18
node1
GIVE FEEDBACK
cnska02i_cnskajgl2o9000fid64g
IP
192.168.0.18
OPEN PORT
Memory
1.07% (42.95MiB / 3.906GiB)
CPU
2.89%
SSH
ssh ip172-18-0-120-cnska02im2rg00akns50@direct.labs.pl
DELETE
EDITOR
The PWD team.
[node1] (local) root@192.168.0.18 ~
$ docker version
Client:
Version: 24.0.7
API version: 1.43
Go version: go1.20.10
Git commit: afdd53b
Built: Thu Oct 26 09:04:00 2023
OS/Arch: linux/amd64
Context: default
Server: Docker Engine - Community
Engine:
Version: 24.0.7
API version: 1.43 (minimum version 1.12)
Go version: go1.20.10
Git commit: 311b9ff
Built: Thu Oct 26 09:05:28 2023
OS/Arch: linux/amd64
Experimental: true
containerd:
Version: v1.7.6
GitCommit: 091922f03c2762540fd057fba91260237ff86acb
runc:
Version: 1.1.9
GitCommit: v1.1.9-0-gccaecfc
docker-init:
Version: 0.19.0
GitCommit: de40ad0
[node1] (local) root@192.168.0.18 ~
$ docker --version
Docker version 24.0.7, build afdd53b
[node1] (local) root@192.168.0.18 ~
$

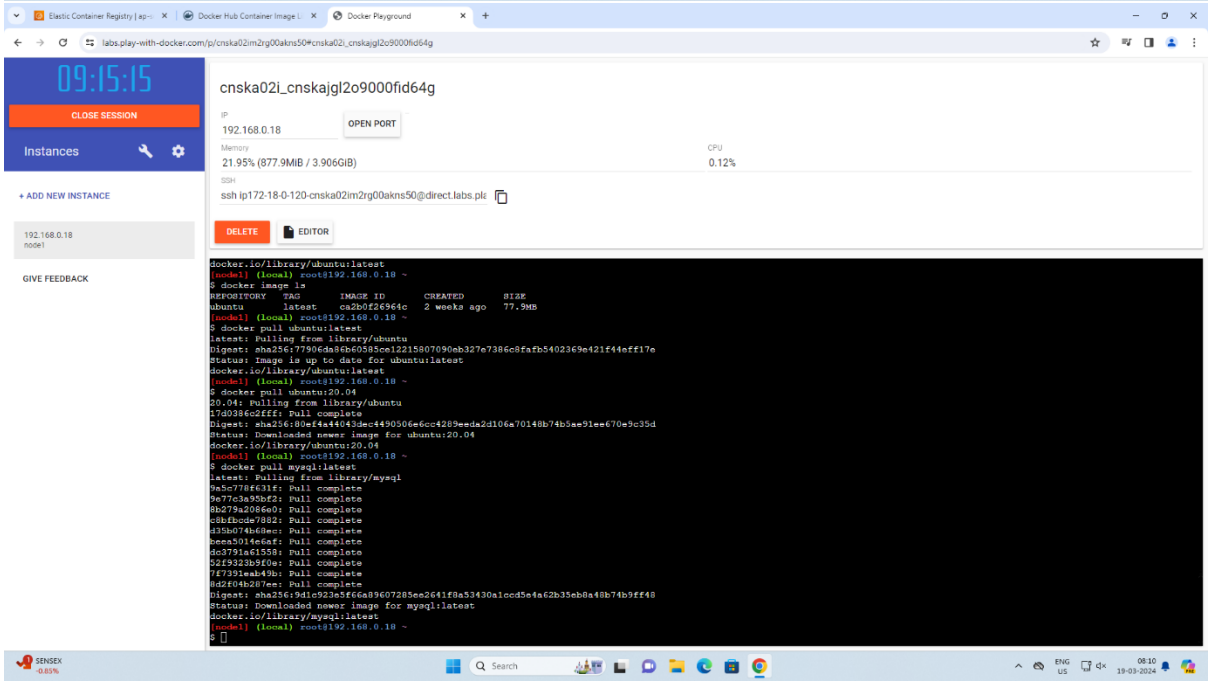

09:26:24
CLOSE SESSION
Instances
+ ADD NEW INSTANCE
192.168.0.18
node1
GIVE FEEDBACK
cnska02i_cnskajgl2o9000fid64g
IP
192.168.0.18
OPEN PORT
Memory
1.08% (43.36MiB / 3.906GiB)
CPU
0.58%
SSH
ssh ip172-18-0-120-cnska02im2rg00akns50@direct.labs.pla
DELETE
EDITOR
Docker version 24.0.7, build afdd53b
[node1] (local) root@192.168.0.18 ~
$ docker help
Usage: docker [OPTIONS] COMMAND
A self-sufficient runtime for containers
Common Commands:
run Create and run a new container from an image
exec Execute a command in a running container
ps List containers
build Build an image from a Dockerfile
pull Download an image from a registry
push Upload an image to a registry
images List images
login Log in to a registry
logout Log out from a registry
search Search Docker Hub for images
version Show the Docker version information
info Display system-wide information
Management Commands:
builder Manage builds
buildx* Docker Buildx (Docker Inc., v0.11.2)
checkpoint Manage checkpoints
compose* Docker Compose (Docker Inc., v2.23.0)
container Manage containers
context Manage contexts
image Manage images
manifest Manage Docker image manifests and manifest lists
network Manage networks
plugin Manage plugins
scout* Docker Scout (Docker Inc., v1.0.9)
system Manage Docker
trust Manage trust on Docker images

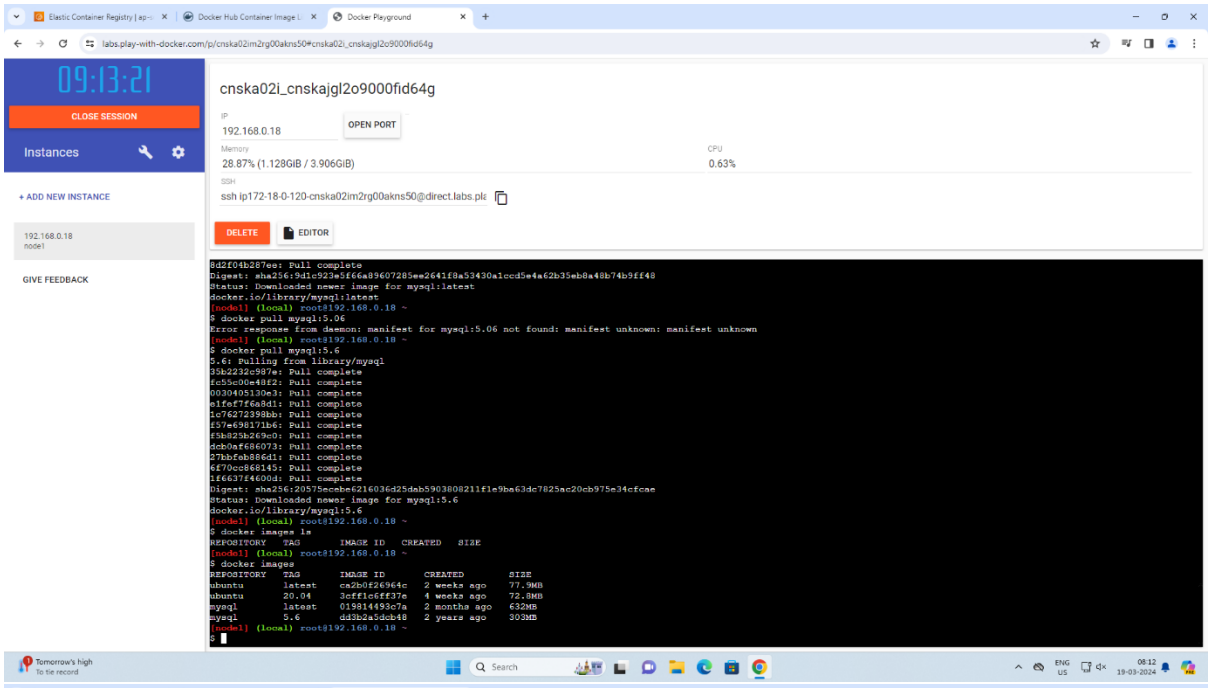
09:26:05
CLOSE SESSION
Instances
+ ADD NEW INSTANCE
192.168.0.18
node1
GIVE FEEDBACK
cnska02i_cnskajgl2o9000fid64g
IP
192.168.0.18
OPEN PORT
Memory
1.10% (44.02MiB / 3.906GiB)
CPU
0.30%
SSH
ssh ip172-18-0-120-cnska02im2rg00akns50@direct.labs.pla
DELETE
EDITOR
-H, --host list Daemon socket to connect to
-l, --log-level string Set the logging level ("debug", "info", "warn", "error", "fatal") (default "info")
--tls Use TLS; implied by --tlsverify
--tlscacert string Trust certs signed only by this CA (default "/root/.docker/ca.pem")
--tlscert string Path to TLS certificate file (default "/root/.docker/cert.pem")
--tlskey string Path to TLS key file (default "/root/.docker/key.pem")
--tlsverify Use TLS and verify the remote
-v, --version Print version information and quit
Run 'docker COMMAND --help' for more information on a command.
For more help on how to use Docker, head to https://docs.docker.com/go/guides/
[node1] (local) root@192.168.0.18 ~
$ docker image --help
Usage: docker image COMMAND
Manage images
Commands:
build Build an image from a Dockerfile
history Show the history of an image
import Import the contents from a tarball to create a filesystem image
inspect Display detailed information on one or more images
load Load an image from a tar archive or STDIN
ls List images
prune Remove unused images
pull Download an image from a registry
push Upload an image to a registry
rm Remove one or more images
save Save one or more images to a tar archive (streamed to STDOUT by default)
tag Create a tag TARGET_IMAGE that refers to SOURCE_IMAGE
Run 'docker image COMMAND --help' for more information on a command.
[node1] (local) root@192.168.0.18 ~
$

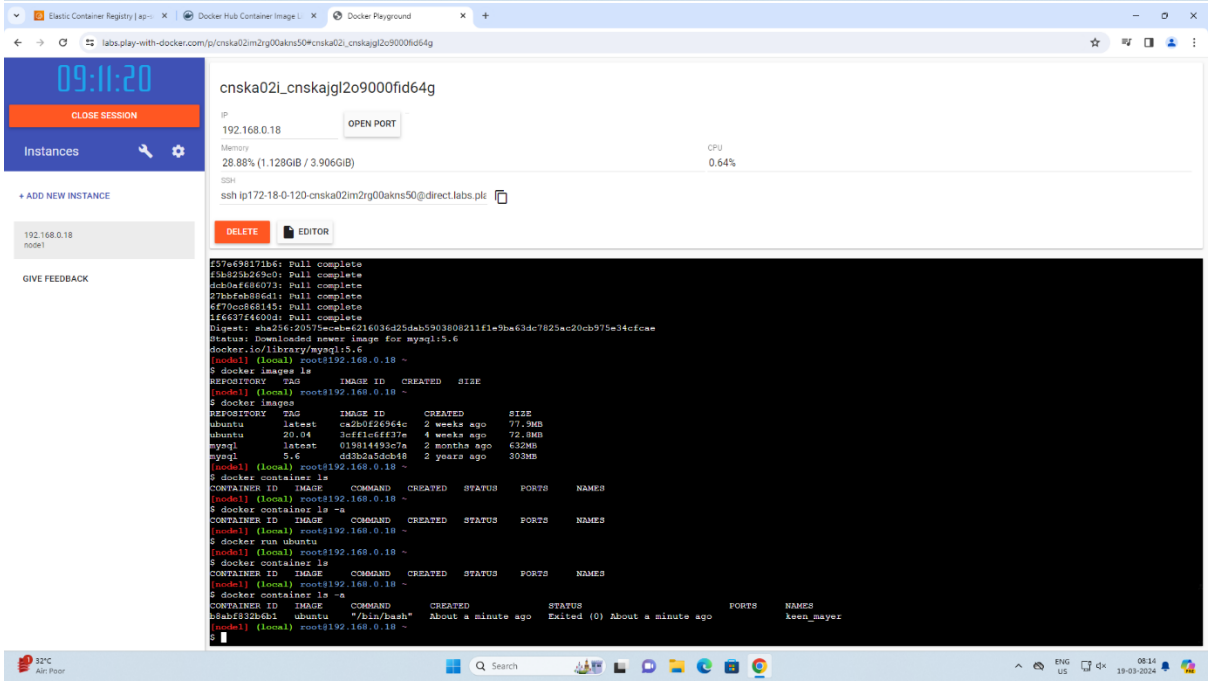

09:24:27
CLOSE SESSION
Instances
+ ADD NEW INSTANCE
192.168.0.18
node1
GIVE FEEDBACK
cnska02i_cnskajgl2o9000fid64g
IP
192.168.0.18
OPEN PORT
Memory
1.12% (44.63MiB / 3.906GiB)
CPU
1.26%
SSH
ssh ip172-18-0-120-cnska02im2rg00akns50@direct.labs.pl
DELETE
EDITOR
For more help on how to use Docker, head to https://docs.docker.com/go/guides/
[node1] (local) root@192.168.0.18 ~
$ docker image --help
Usage: docker image COMMAND
Manage images
Commands:
build Build an image from a Dockerfile
history Show the history of an image
import Import the contents from a tarball to create a filesystem image
inspect Display detailed information on one or more images
load Load an image from a tar archive or STDIN
ls List images
prune Remove unused images
pull Download an image from a registry
push Upload an image to a registry
rm Remove one or more images
save Save one or more images to a tar archive (streamed to STDOUT by default)
tag Create a tag TARGET_IMAGE that refers to SOURCE_IMAGE
Run 'docker image COMMAND --help' for more information on a command.
[node1] (local) root@192.168.0.18 ~
$ docker ls
docker: 'ls' is not a docker command.
See 'docker --help'
[node1] (local) root@192.168.0.18 ~
$ docker images ls
REPOSITORY TAG IMAGE ID CREATED SIZE
[node1] (local) root@192.168.0.18 ~
$ docker image ls -a
REPOSITORY TAG IMAGE ID CREATED SIZE
[node1] (local) root@192.168.0.18 ~
$
We'd love to hear about your usage of Play with Docker.
Please take a moment to fill out our survey.
TAKE SURVEY
NOT NOW

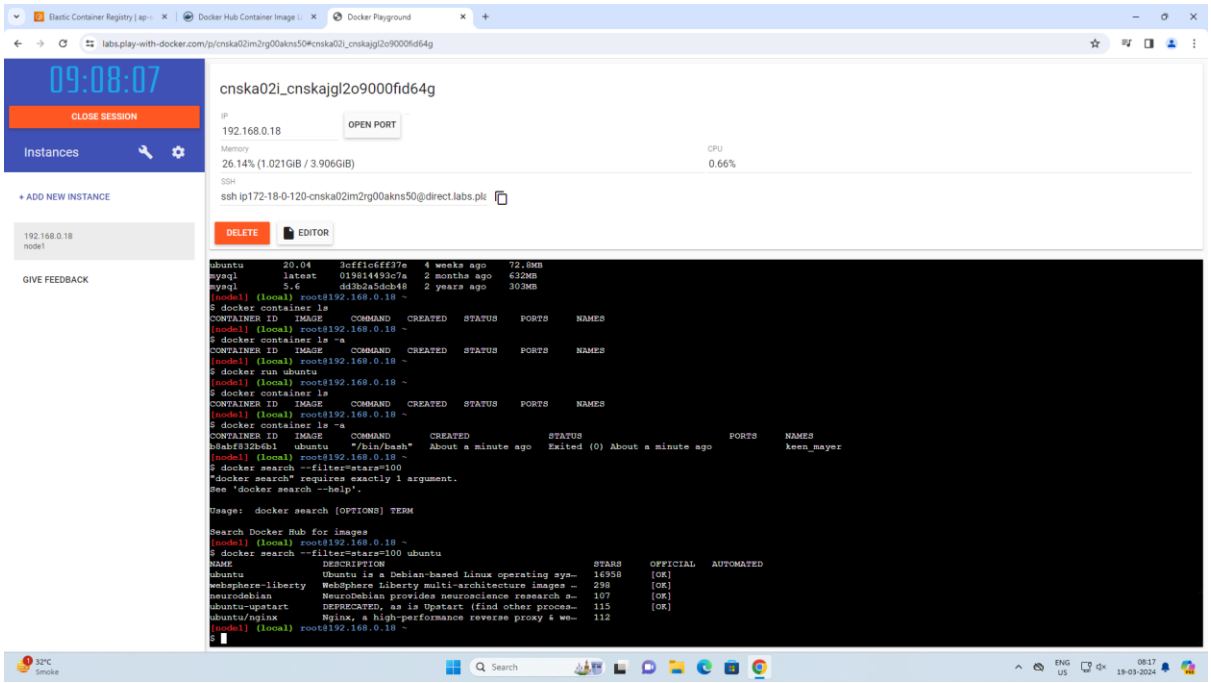

09:23:54
CLOSE SESSION
Instances
+ ADD NEW INSTANCE
192.168.0.18
node1
GIVE FEEDBACK
cnska02i_cnskajgl2o9000fid64g
IP
192.168.0.18
OPEN PORT
Memory
1.12% (44.82MiB / 3.906GiB)
CPU
3.41%
SSH
ssh ip172-18-0-120-cnska02im2rg00akns50@direct.labs.pl
DELETE
EDITOR
$ docker image --help
Usage: docker image COMMAND
Manage images
Commands:
build Build an image from a Dockerfile
history Show the history of an image
import Import the contents from a tarball to create a filesystem image
inspect Display detailed information on one or more images
load Load an image from a tar archive or STDIN
ls List images
prune Remove unused images
pull Download an image from a registry
push Upload an image to a registry
rm Remove one or more images
save Save one or more images to a tar archive (streamed to STDOUT by default)
tag Create a tag TARGET_IMAGE that refers to SOURCE_IMAGE
Run 'docker image COMMAND --help' for more information on a command.
[node1] (local) root@192.168.0.18 ~
$ docker ls
docker: 'ls' is not a docker command.
See 'docker --help'
[node1] (local) root@192.168.0.18 ~
$ docker images ls
REPOSITORY TAG IMAGE ID CREATED SIZE
[node1] (local) root@192.168.0.18 ~
$ docker image ls -a
REPOSITORY TAG IMAGE ID CREATED SIZE
[node1] (local) root@192.168.0.18 ~
$ docker container ps
CONTAINER ID IMAGE COMMAND CREATED STATUS PORTS NAMES
[node1] (local) root@192.168.0.18 ~
$
We'd love to hear about your usage of Play with Docker.
Please take a moment to fill out our survey.
TAKE SURVEY
NOT NOW

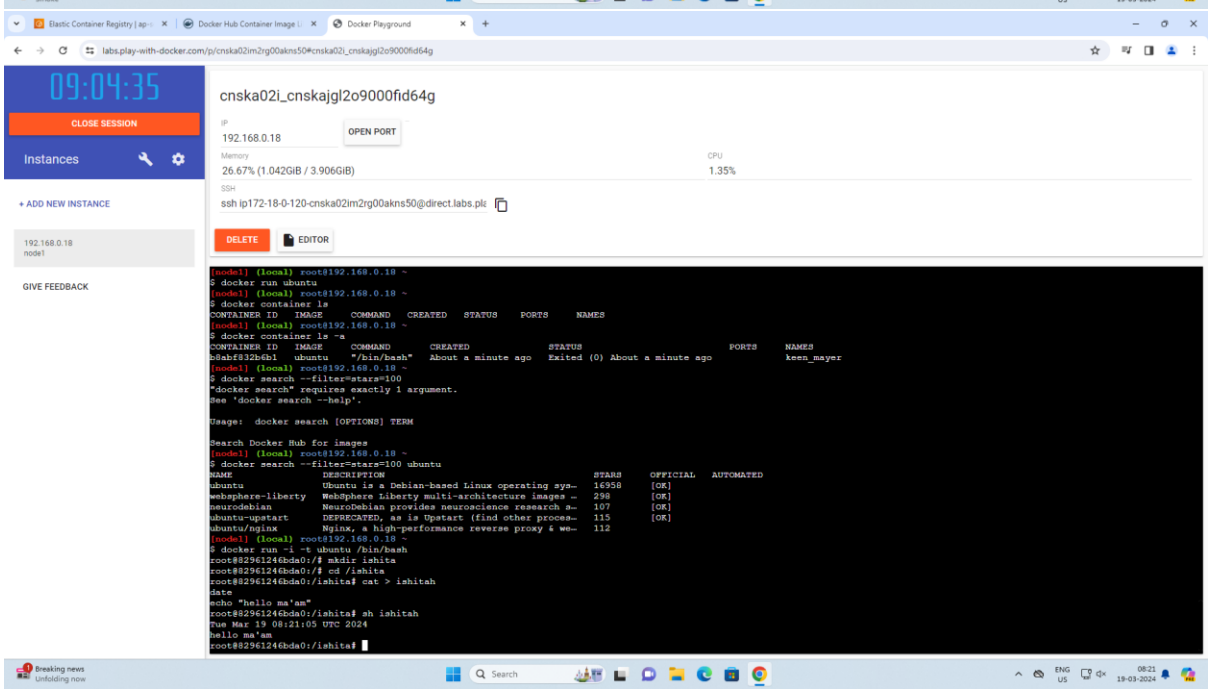

09:20:13
CLOSE SESSION
Instances
+ ADD NEW INSTANCE
192.168.0.18
node1
GIVE FEEDBACK
cnska02i_cnskajgl2o9000fid64g
IP
192.168.0.18
OPEN PORT
Memory
3.64% (145.6MiB / 3.906GiB)
CPU
0.40%
SSH
ssh ip172-18-0-120-cnska02im2rg00akns50@direct.labs.pl
DELETE
EDITOR
push Upload an image to a registry
rm Remove one or more images
save Save one or more images to a tar archive (streamed to STDOUT by default)
tag Create a tag TARGET_IMAGE that refers to SOURCE_IMAGE
Run 'docker image COMMAND --help' for more information on a command.
$ docker ls
docker: 'ls' is not a docker command.
See 'docker --help'
$ docker images ls
REPOSITORY TAG IMAGE ID CREATED SIZE
$ docker image ls -a
REPOSITORY TAG IMAGE ID CREATED SIZE
$ docker container ps
CONTAINER ID IMAGE COMMAND CREATED STATUS PORTS NAMES
$ docker container ps -a
CONTAINER ID IMAGE COMMAND CREATED STATUS PORTS NAMES
$ docker pull ubuntu
Using default tag: latest
latest: Pulling from library/ubuntu
bccd10f490ab: Pull complete
Digest: sha256:77906da86b60585ce12215807090eb327e7386c8fafb5402369e421f44eff17e
Status: Downloaded newer image for ubuntu:latest
docker.io/library/ubuntu:latest
$ docker image ls
REPOSITORY TAG IMAGE ID CREATED SIZE
ubuntu latest ca2b0f26964c 2 weeks ago 77.9MB
We'd love to hear about your usage of Play with Docker. Please take a moment to fill out our survey.
TAKE SURVEY
NOT NOW
Breaking news
Unfolding now
Search
ENG
US
08:05
19-03-2024

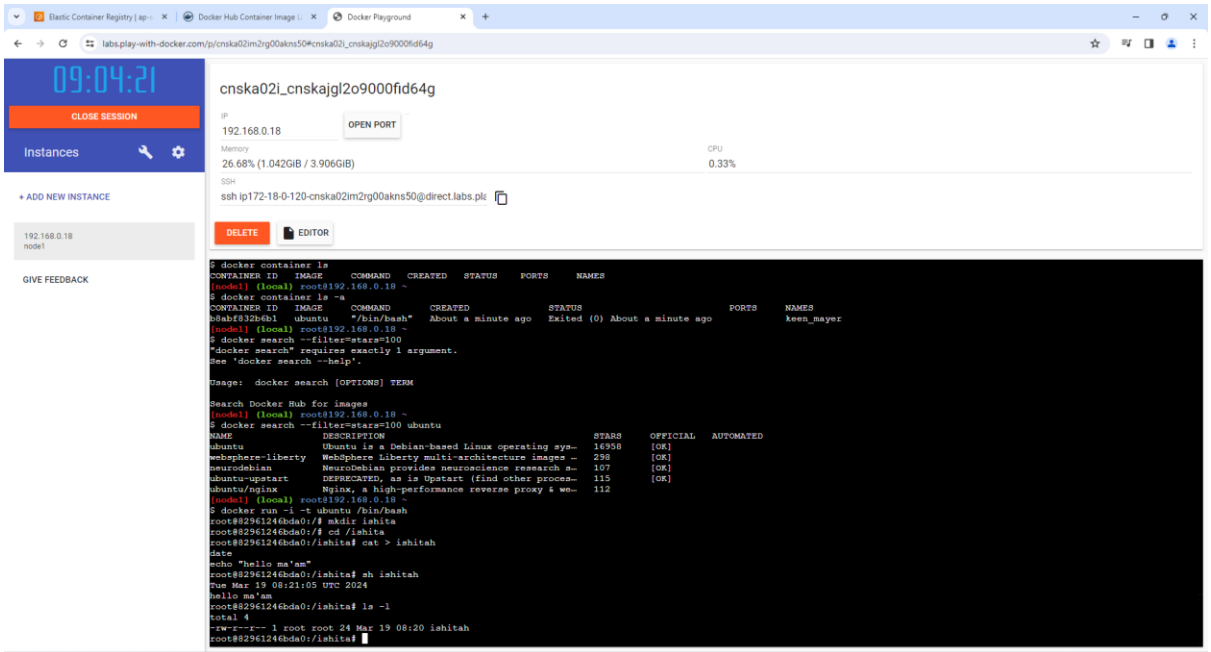

09:15:15
CLOSE SESSION
Instances
+ ADD NEW INSTANCE
192.168.0.18
node1
GIVE FEEDBACK
cnska02i_cnskajgl2o9000fid64g
IP
192.168.0.18
OPEN PORT
Memory
21.95% (877.9MiB / 3.906GiB)
CPU
0.12%
SSH
ssh ip172-18-0-120-cnska02im2rg00akns50@direct.labs.pl
DELETE
EDITOR
docker.io/library/ubuntu:latest
$ docker image ls
REPOSITORY TAG IMAGE ID CREATED SIZE
ubuntu latest ca2b0f26964c 2 weeks ago 77.9MB
$ docker pull ubuntu:latest
latest: Pulling from library/ubuntu
Digest: sha256:77906da86b60585ce12215807090eb327e7386c8fafb5402369e421f44eff17e
Status: Image is up to date for ubuntu:latest
docker.io/library/ubuntu:latest
$ docker pull ubuntu:20.04
20.04: Pulling from library/ubuntu
17d0386c2fff: Pull complete
Digest: sha256:80ef4a44043dec4490506e6cc4289eeda2d106a70148b74b5ae91ee670e9c35d
Status: Downloaded newer image for ubuntu:20.04
docker.io/library/ubuntu:20.04
$ docker pull mysql:latest
latest: Pulling from library/mysql
9e5c778f631f: Pull complete
9e77c3a95bf2: Pull complete
8b279a2086e0: Pull complete
c8bfbcde7882: Pull complete
d35b074b68ec: Pull complete
beea5014e6af: Pull complete
dc3791a61558: Pull complete
52f9323b9f0e: Pull complete
7f7391eab49b: Pull complete
8d2f04b287ee: Pull complete
Digest: sha256:9d1c923e5f66a89607285ee2641f8a53430a1ccd5e4a62b35eb0a48b74b9ff48
Status: Downloaded newer image for mysql:latest
docker.io/library/mysql:latest
SENSEX
-0.65%
Search
ENG
US
08:10
19-03-2024

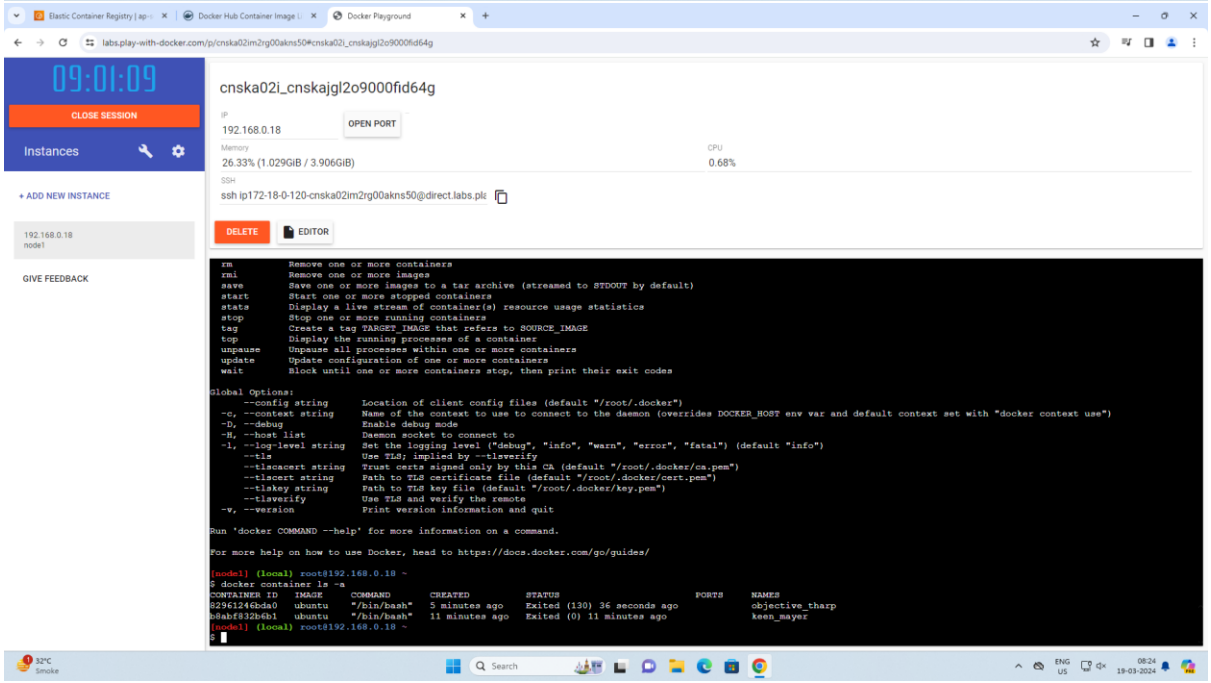

09:13:21
CLOSE SESSION
Instances
+ ADD NEW INSTANCE
192.168.0.18
node1
GIVE FEEDBACK
cnska02i_cnskajgl2o9000fid64g
IP
192.168.0.18
OPEN PORT
Memory
28.87% (1.128GiB / 3.906GiB)
CPU
0.63%
SSH
ssh ip172-18-0-120-cnska02im2rg00akns50@direct.labs.pla
DELETE
EDITOR
8d2f04b287ee: Pull complete
Digest: sha256:9d1c923e5f66a88607285ee2641f8a53430a1ccd5e4a62b35eb8a48b74b9ff48
Status: Downloaded newer image for mysql:latest
docker.io/library/mysql:latest
[node1] (local) root@192.168.0.18 ~
$ docker pull mysql:5.06
Error response from daemon: manifest for mysql:5.06 not found: manifest unknown: manifest unknown
[node1] (local) root@192.168.0.18 ~
$ docker pull mysql:5.6
5.6: Pulling from library/mysql
33b2232c997e: Pull complete
fc55c00e48f2: Pull complete
0030405130e3: Pull complete
e1fef7f6a8d1: Pull complete
1c76272398bb: Pull complete
f57e698171b6: Pull complete
f5b825b269c0: Pull complete
dcb0af686073: Pull complete
27bbfbb886d1: Pull complete
6f70cc868145: Pull complete
1f6637f4600d: Pull complete
Digest: sha256:20575ecebe6216036d25dab5903808211f1e9ba63dc7825ac20cb975e34cfcae
Status: Downloaded newer image for mysql:5.6
docker.io/library/mysql:5.6
[node1] (local) root@192.168.0.18 ~
$ docker images ls
REPOSITORY   TAG   IMAGE ID   CREATED   SIZE
[node1] (local) root@192.168.0.18 ~
$ docker images
REPOSITORY   TAG   IMAGE ID   CREATED   SIZE
ubuntu   latest   ca2b0f26964c   2 weeks ago   77.9MB
ubuntu   20.04   3cff1c6ff37e   4 weeks ago   72.8MB
mysql   latest   01981493c7a   2 months ago   632MB
mysql   5.6   dd3b2a5dcb48   2 years ago   303MB
[node1] (local) root@192.168.0.18 ~
$

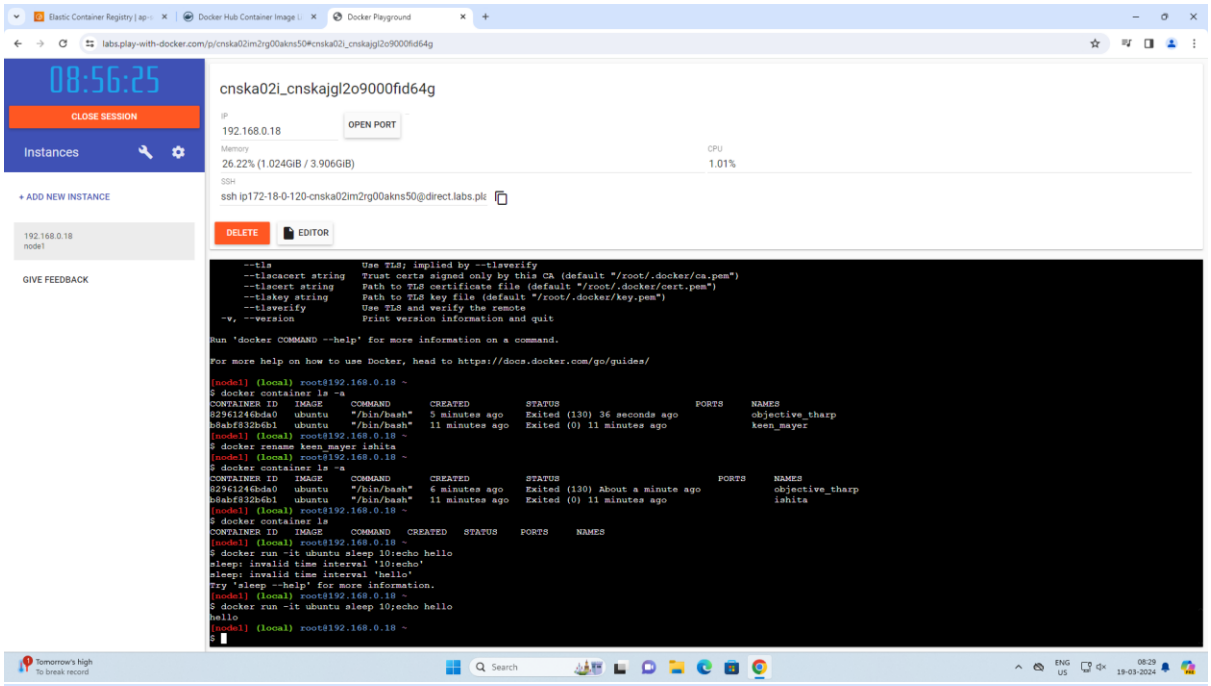

09:11:20
CLOSE SESSION
Instances
+ ADD NEW INSTANCE
192.168.0.18
node1
GIVE FEEDBACK
cnska02i_cnskajgl2o9000fid64g
IP
192.168.0.18
OPEN PORT
Memory
28.88% (1.128GiB / 3.906GiB)
CPU
0.64%
SSH
ssh ip172-18-0-120-cnska02im2rg00akns50@direct.labs.pla
DELETE
EDITOR
f57e698171b6: Pull complete
f5b825b269c0: Pull complete
dcb0af686073: Pull complete
27bbfbb886d1: Pull complete
6f70cc868145: Pull complete
1f6637f4600d: Pull complete
Digest: sha256:20575ecebe6216036d25dab5903808211f1e9ba63dc7825ac20cb975e34cfcae
Status: Downloaded newer image for mysql:5.6
docker.io/library/mysql:5.6
[node1] (local) root@192.168.0.18 ~
$ docker images ls
REPOSITORY   TAG   IMAGE ID   CREATED   SIZE
[node1] (local) root@192.168.0.18 ~
$ docker images
REPOSITORY   TAG   IMAGE ID   CREATED   SIZE
ubuntu   latest   ca2b0f26964c   2 weeks ago   77.9MB
ubuntu   20.04   3cff1c6ff37e   4 weeks ago   72.8MB
mysql   latest   01981493c7a   2 months ago   632MB
mysql   5.6   dd3b2a5dcb48   2 years ago   303MB
[node1] (local) root@192.168.0.18 ~
$ docker container ls
CONTAINER ID   IMAGE   COMMAND   CREATED   STATUS   PORTS   NAMES
[node1] (local) root@192.168.0.18 ~
$ docker container ls -a
CONTAINER ID   IMAGE   COMMAND   CREATED   STATUS   PORTS   NAMES
[node1] (local) root@192.168.0.18 ~
$ docker run ubuntu
[node1] (local) root@192.168.0.18 ~
$ docker container ls
CONTAINER ID   IMAGE   COMMAND   CREATED   STATUS   PORTS   NAMES
[node1] (local) root@192.168.0.18 ~
$ docker container ls -a
CONTAINER ID   IMAGE   COMMAND   CREATED   STATUS   PORTS   NAMES
b8abf832b6b1   ubuntu   "/bin/bash"   About a minute ago   Exited (0) About a minute ago   keen_mayer
[node1] (local) root@192.168.0.18 ~
$

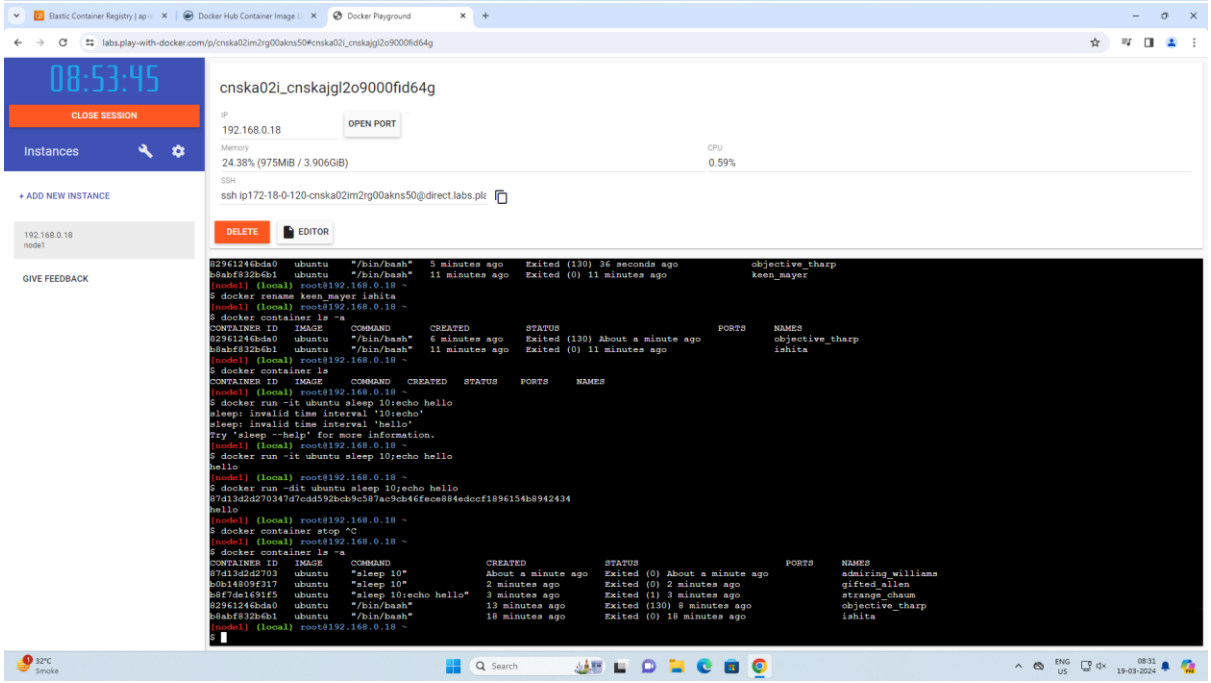

09:08:07
CLOSE SESSION
Instances
+ ADD NEW INSTANCE
192.168.0.18
node1
GIVE FEEDBACK
cnska02i_cnskajgl2o9000fid64g
IP
192.168.0.18
OPEN PORT
Memory
26.14% (1.021GiB / 3.906GiB)
CPU
0.66%
SSH
ssh ip172-18-0-120-cnska02im2rg00akns50@direct.labs.pla
DELETE
EDITOR
ubuntu 20.04 3cff1c6ff37e 4 weeks ago 72.8MB
mysql latest 019814493c7a 2 months ago 632MB
mysql 5.6 dd3b2a5dcb48 2 years ago 303MB
[node1] (local) root@192.168.0.18 ~
$ docker container ls
CONTAINER ID IMAGE COMMAND CREATED STATUS PORTS NAMES
[node1] (local) root@192.168.0.18 ~
$ docker container ls -a
CONTAINER ID IMAGE COMMAND CREATED STATUS PORTS NAMES
[node1] (local) root@192.168.0.18 ~
$ docker run ubuntu
[node1] (local) root@192.168.0.18 ~
$ docker container ls
CONTAINER ID IMAGE COMMAND CREATED STATUS PORTS NAMES
[node1] (local) root@192.168.0.18 ~
$ docker container ls -a
CONTAINER ID IMAGE COMMAND CREATED STATUS PORTS NAMES
b8abf832b6b1 ubuntu "/bin/bash" About a minute ago Exited (0) About a minute ago keen_mayer
[node1] (local) root@192.168.0.18 ~
$ docker search --filter=stars=100
"docker search" requires exactly 1 argument.
See 'docker search --help'.
Usage: docker search [OPTIONS] TERM
Search Docker Hub for images
[node1] (local) root@192.168.0.18 ~
$ docker search --filter=stars=100 ubuntu
NAME DESCRIPTION STARS OFFICIAL AUTOMATED
ubuntu Ubuntu is a Debian-based Linux operating sys… 16958 [OK]
websphere-liberty WebSphere Liberty multi-architecture images … 298 [OK]
neurodebian NeuroDebian provides neuroscience research s… 107 [OK]
ubuntu-upstart DEPRECATED, as is Upstart (find other proces… 115 [OK]
ubuntu/nginx Nginx, a high-performance reverse proxy & we… 112
[node1] (local) root@192.168.0.18 ~
$

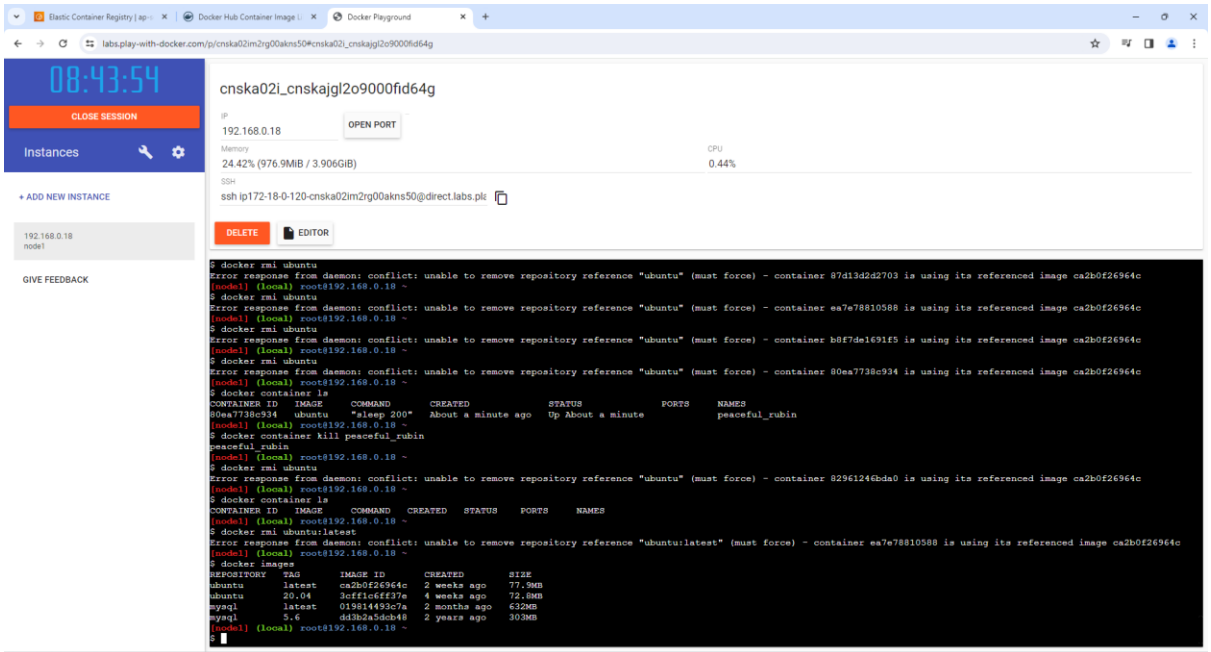

09:04:35
CLOSE SESSION
Instances
+ ADD NEW INSTANCE
192.168.0.18
node1
GIVE FEEDBACK
cnska02i_cnskajgl2o9000fid64g
IP
192.168.0.18
OPEN PORT
Memory
26.67% (1.042GiB / 3.906GiB)
CPU
1.35%
SSH
ssh ip172-18-0-120-cnska02im2rg00akns50@direct.labs.pla
DELETE
EDITOR
[node1] (local) root@192.168.0.18 ~
$ docker run ubuntu
[node1] (local) root@192.168.0.18 ~
$ docker container ls
CONTAINER ID IMAGE COMMAND CREATED STATUS PORTS NAMES
[node1] (local) root@192.168.0.18 ~
$ docker container ls -a
CONTAINER ID IMAGE COMMAND CREATED STATUS PORTS NAMES
b8abf832b6b1 ubuntu "/bin/bash" About a minute ago Exited (0) About a minute ago keen_mayer
[node1] (local) root@192.168.0.18 ~
$ docker search --filter=stars=100
"docker search" requires exactly 1 argument.
See 'docker search --help'.
Usage: docker search [OPTIONS] TERM
Search Docker Hub for images
[node1] (local) root@192.168.0.18 ~
$ docker search --filter=stars=100 ubuntu
NAME DESCRIPTION STARS OFFICIAL AUTOMATED
ubuntu Ubuntu is a Debian-based Linux operating sys… 16958 [OK]
websphere-liberty WebSphere Liberty multi-architecture images … 298 [OK]
neurodebian NeuroDebian provides neuroscience research s… 107 [OK]
ubuntu-upstart DEPRECATED, as is Upstart (find other proces… 115 [OK]
ubuntu/nginx Nginx, a high-performance reverse proxy & we… 112
[node1] (local) root@192.168.0.18 ~
$ docker run -i -t ubuntu /bin/bash
root@8296124bda0:/# mkdir ishita
root@8296124bda0:/# cd /ishita
root@8296124bda0:/ishita# cat > ishitah
date
echo "hello ma'am"
root@8296124bda0:/ishita# sh ishitah
Tue Mar 19 08:21:05 UTC 2024
hello ma'am
root@8296124bda0:/ishita#

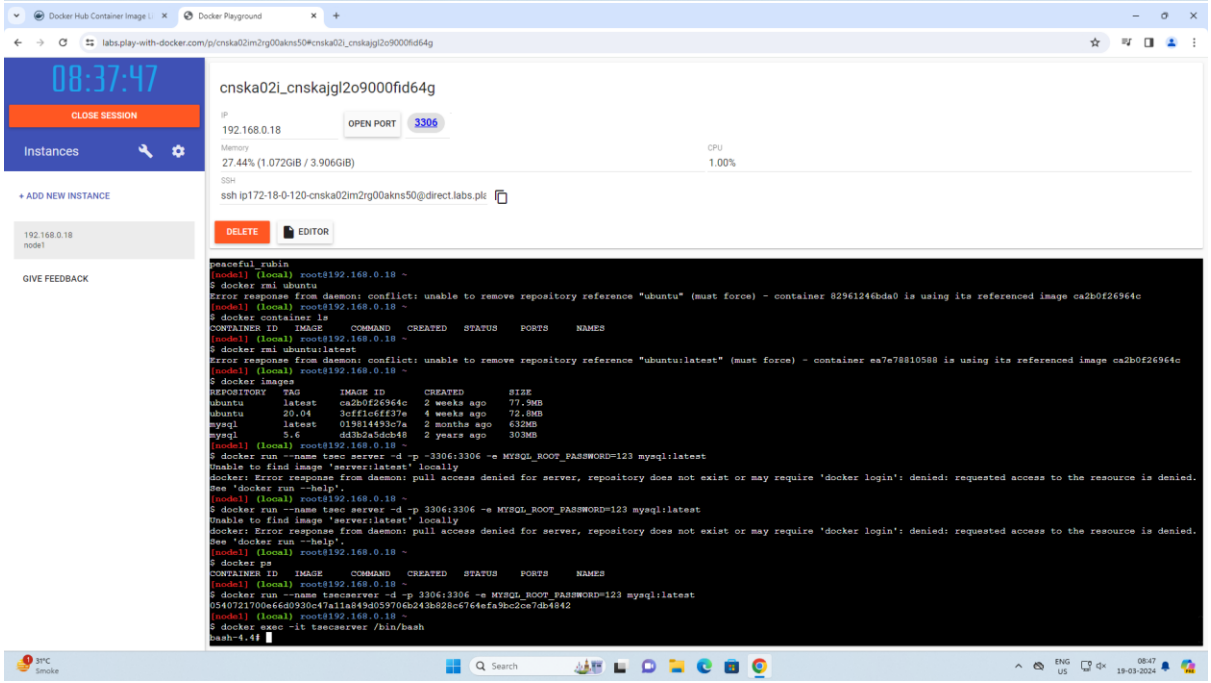

09:04:21
CLOSE SESSION
Instances
+ ADD NEW INSTANCE
192.168.0.18
node1
GIVE FEEDBACK
cnska02i_cnskajgl2o9000fid64g
IP
192.168.0.18
OPEN PORT
Memory
26.68% (1.042GiB / 3.906GiB)
CPU
0.33%
SSH
ssh ip172-18-0-120-cnska02im2rg00akns50@direct.labs.pla
DELETE
EDITOR
$ docker container ls
CONTAINER ID IMAGE COMMAND CREATED STATUS PORTS NAMES
[node1] (local) root@192.168.0.18 ~
$ docker container ls -a
CONTAINER ID IMAGE COMMAND CREATED STATUS PORTS NAMES
b8abf832b6b1 ubuntu "/bin/bash" About a minute ago Exited (0) About a minute ago keen_mayer
[node1] (local) root@192.168.0.18 ~
$ docker search --filter=stars=100
"docker search" requires exactly 1 argument.
See 'docker search --help'.
Usage: docker search [OPTIONS] TERM
Search Docker Hub for images
[node1] (local) root@192.168.0.18 ~
$ docker search --filter=stars=100 ubuntu
NAME DESCRIPTION STARS OFFICIAL AUTOMATED
ubuntu Ubuntu is a Debian-based Linux operating sys… 16958 [OK]
websphere-liberty WebSphere Liberty multi-architecture images … 298 [OK]
neurodebian NeuroDebian provides neuroscience research s… 107 [OK]
ubuntu-upstart DEPRECATED, as is Upstart (find other proces… 115 [OK]
ubuntu/nginx Nginx, a high-performance reverse proxy & we… 112
[node1] (local) root@192.168.0.18 ~
$ docker run -i -t ubuntu /bin/bash
root@8296124 6bda0:/# mkdir ishita
root@8296124 6bda0:/# cd /ishita
root@8296124 6bda0:/ishita# cat > ishitah
date
echo "hello ma'am"
root@8296124 6bda0:/ishita# sh ishitah
Tue Mar 19 08:21:05 UTC 2024
hello ma'am
root@8296124 6bda0:/ishita# ls -l
total 4
-rw-r--r-- 1 root root 24 Mar 19 08:20 ishitah
root@8296124 6bda0:/ishita#

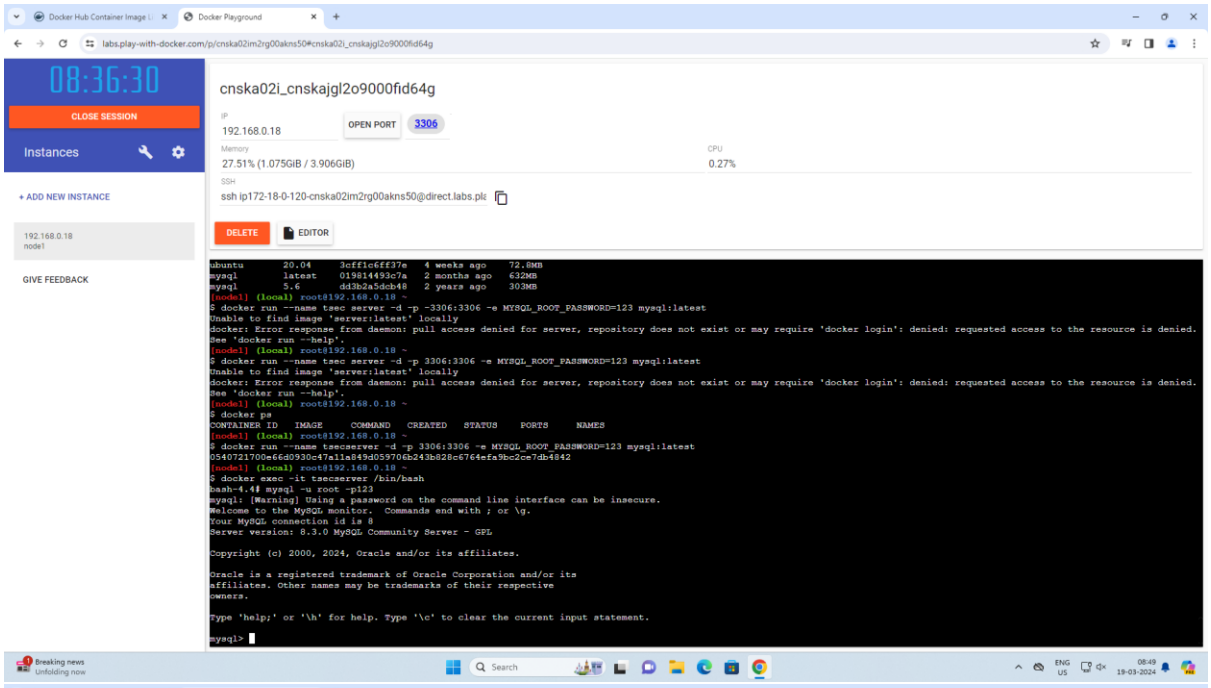

09:01:09
CLOSE SESSION
Instances
+ ADD NEW INSTANCE
192.168.0.18
node1
GIVE FEEDBACK
cnska02i_cnskajgl2o9000fid64g
IP
192.168.0.18
OPEN PORT
Memory
26.33% (1.029GiB / 3.906GiB)
CPU
0.68%
SSH
ssh ip172-18-0-120-cnska02im2rg00akns50@direct.labs.pla
DELETE
EDITOR
rm Remove one or more containers
rmi Remove one or more images
save Save one or more images to a tar archive (streamed to STDOUT by default)
start Start one or more stopped containers
stats Display a live stream of container(s) resource usage statistics
stop Stop one or more running containers
tag Create a tag TARGET_IMAGE that refers to SOURCE_IMAGE
top Display the running processes of a container
unpause Unpause all processes within one or more containers
update Update configuration of one or more containers
wait Block until one or more containers stop, then print their exit codes
Global Options:
--config string Location of client config files (default "/root/.docker")
-c, --context string Name of the context to use to connect to the daemon (overrides DOCKER_HOST env var and default context set with "docker context use")
-D, --debug Enable debug mode
-H, --host list Daemon socket to connect to
-l, --log-level string Set the logging level ("debug", "info", "warn", "error", "fatal") (default "info")
--tls Use TLS; implied by --tlsverify
--tlscacert string Trust certs signed only by this CA (default "/root/.docker/ca.pem")
--tlscert string Path to TLS certificate file (default "/root/.docker/cert.pem")
--tlskey string Path to TLS key file (default "/root/.docker/key.pem")
--tlsverify Use TLS and verify the remote
-v, --version Print version information and quit
Run 'docker COMMAND --help' for more information on a command.
For more help on how to use Docker, head to https://docs.docker.com/go/guides/
[node1] (local) root@192.168.0.18 ~
$ docker container ls -a
CONTAINER ID IMAGE COMMAND CREATED STATUS PORTS NAMES
8296124 6bda0 ubuntu "/bin/bash" 5 minutes ago Exited (130) 36 seconds ago objective_tharp
b8abf832b6b1 ubuntu "/bin/bash" 11 minutes ago Exited (0) 11 minutes ago keen_mayer
[node1] (local) root@192.168.0.18 ~
$

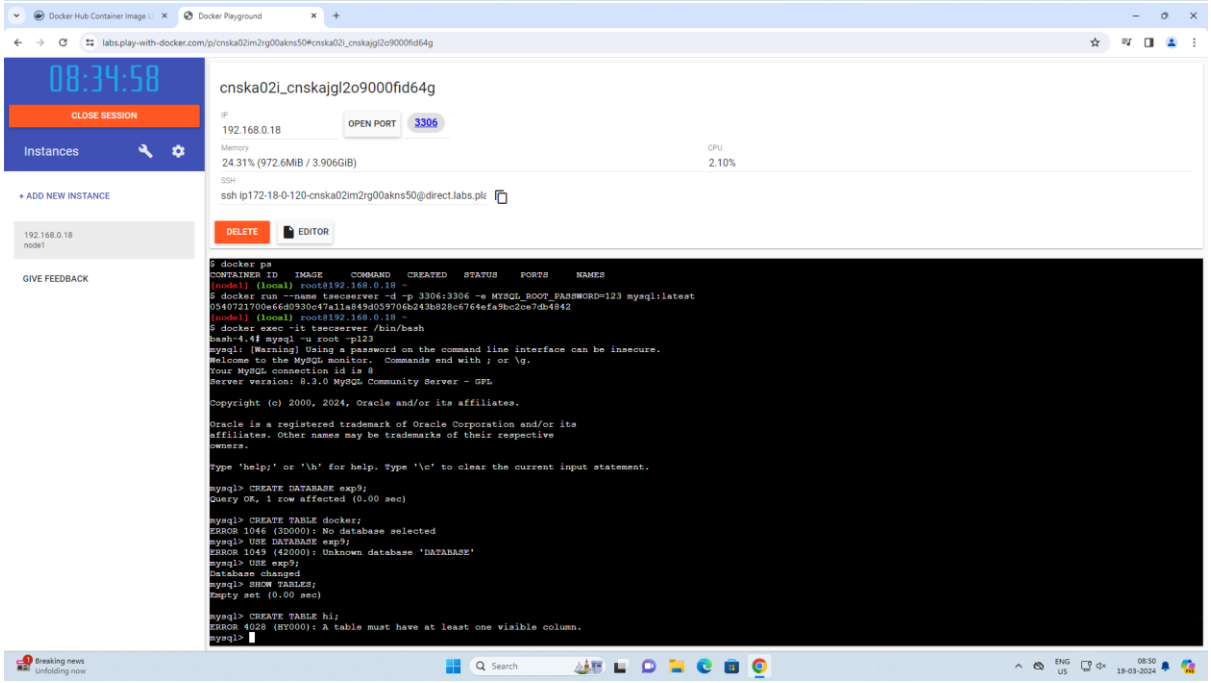

08:56:25
CLOSE SESSION
Instances
+ ADD NEW INSTANCE
192.168.0.18
node1
GIVE FEEDBACK
cnska02i_cnskajgl2o9000fid64g
IP
192.168.0.18
OPEN PORT
Memory
26.22% (1.024GiB / 3.906GiB)
CPU
1.01%
SSH
ssh ip172-18-0-120-cnska02im2rg00akns50@direct.labs.pla
DELETE
EDITOR
--tls Use TLS; implied by --tlsverify
--tlscacert string Trust certs signed only by this CA (default "/root/.docker/ca.pem")
--tlscert string Path to TLS certificate file (default "/root/.docker/cert.pem")
--tlskey string Path to TLS key file (default "/root/.docker/key.pem")
--tlsverify Use TLS and verify the remote
-v, --version Print version information and quit
Run 'docker COMMAND --help' for more information on a command.
For more help on how to use Docker, head to https://docs.docker.com/go/guides/
[node1] (local) root@192.168.0.18 ~
$ docker container ls -a
CONTAINER ID IMAGE COMMAND CREATED STATUS PORTS NAMES
82961246bda0 ubuntu "/bin/bash" 5 minutes ago Exited (130) 36 seconds ago objective_tharp
b8abf832b6b1 ubuntu "/bin/bash" 11 minutes ago Exited (0) 11 minutes ago keen_mayer
[node1] (local) root@192.168.0.18 ~
$ docker rename keen_mayer ishita
[node1] (local) root@192.168.0.18 ~
$ docker container ls -a
CONTAINER ID IMAGE COMMAND CREATED STATUS PORTS NAMES
82961246bda0 ubuntu "/bin/bash" 6 minutes ago Exited (130) About a minute ago objective_tharp
b8abf832b6b1 ubuntu "/bin/bash" 11 minutes ago Exited (0) 11 minutes ago ishita
[node1] (local) root@192.168.0.18 ~
$ docker container ls
CONTAINER ID IMAGE COMMAND CREATED STATUS PORTS NAMES
[node1] (local) root@192.168.0.18 ~
$ docker run -it ubuntu sleep 10;echo hello
sleep: invalid time interval '10:echo'
sleep: invalid time interval 'hello'
Try 'sleep --help' for more information.
[node1] (local) root@192.168.0.18 ~
$ docker run -it ubuntu sleep 10;echo hello
hello
[node1] (local) root@192.168.0.18 ~
$

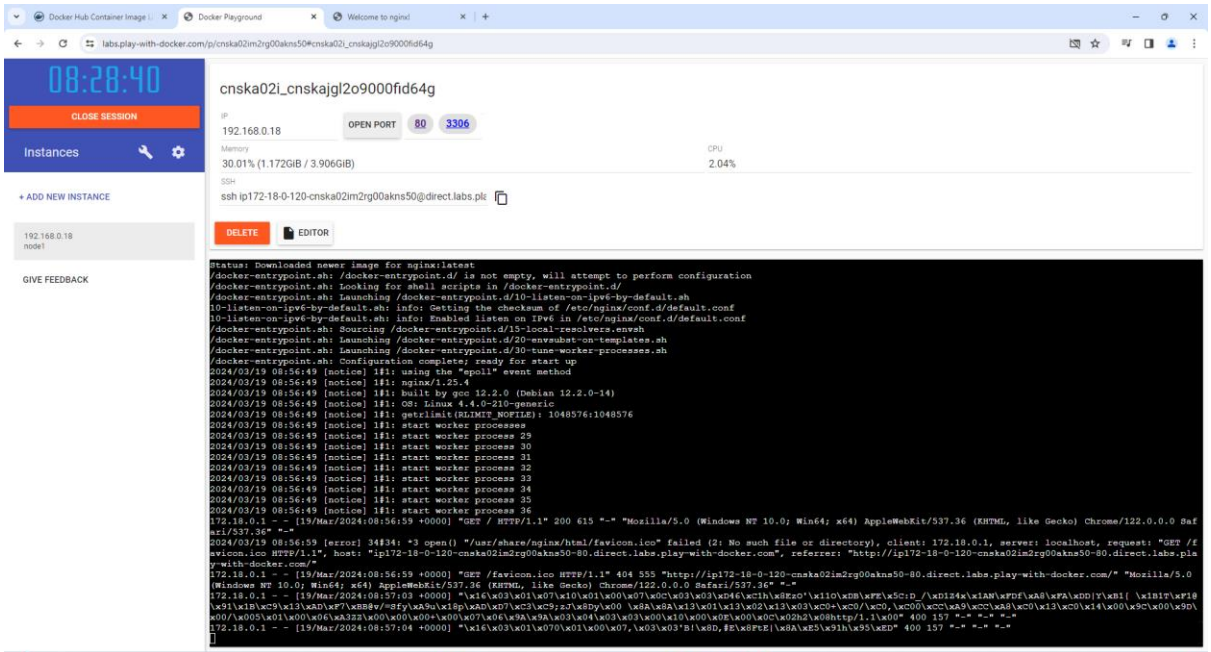

08:53:45
CLOSE SESSION
Instances
+ ADD NEW INSTANCE
192.168.0.18
node1
GIVE FEEDBACK
cnska02i_cnskajgl2o9000fid64g
IP
192.168.0.18
OPEN PORT
Memory
24.38% (975MiB / 3.906GiB)
CPU
0.59%
SSH
ssh ip172-18-0-120-cnska02im2rg00akns50@direct.labs.pla
DELETE
EDITOR
82961246bda0 ubuntu "/bin/bash" 5 minutes ago Exited (130) 36 seconds ago objective_tharp
b8abf832b6b1 ubuntu "/bin/bash" 11 minutes ago Exited (0) 11 minutes ago keen_mayer
[node1] (local) root@192.168.0.18 ~
$ docker rename keen_mayer ishita
[node1] (local) root@192.168.0.18 ~
$ docker container ls -a
CONTAINER ID IMAGE COMMAND CREATED STATUS PORTS NAMES
82961246bda0 ubuntu "/bin/bash" 6 minutes ago Exited (130) About a minute ago objective_tharp
b8abf832b6b1 ubuntu "/bin/bash" 11 minutes ago Exited (0) 11 minutes ago ishita
[node1] (local) root@192.168.0.18 ~
$ docker container ls
CONTAINER ID IMAGE COMMAND CREATED STATUS PORTS NAMES
[node1] (local) root@192.168.0.18 ~
$ docker run -it ubuntu sleep 10;echo hello
sleep: invalid time interval '10:echo'
sleep: invalid time interval 'hello'
Try 'sleep --help' for more information.
[node1] (local) root@192.168.0.18 ~
$ docker run -it ubuntu sleep 10;echo hello
hello
[node1] (local) root@192.168.0.18 ~
$ docker run -dit ubuntu sleep 10;echo hello
07d13d2d270347d7cdd592bcb9c587ac9cb46fece884edccf1896154b8942434
hello
[node1] (local) root@192.168.0.18 ~
$ docker container stop ^C
[node1] (local) root@192.168.0.18 ~
$ docker container ls -a
CONTAINER ID IMAGE COMMAND CREATED STATUS PORTS NAMES
07d13d2d2703 ubuntu "sleep 10" About a minute ago Exited (0) About a minute ago admiring_williams
2c0b14809f317 ubuntu "sleep 10" 2 minutes ago Exited (0) 2 minutes ago gifted_allen
b8f7de1691f5 ubuntu "sleep 10;echo hello" 3 minutes ago Exited (1) 3 minutes ago strange_chaum
82961246bda0 ubuntu "/bin/bash" 13 minutes ago Exited (130) 8 minutes ago objective_tharp
b8abf832b6b1 ubuntu "/bin/bash" 18 minutes ago Exited (0) 18 minutes ago ishita
[node1] (local) root@192.168.0.18 ~
$

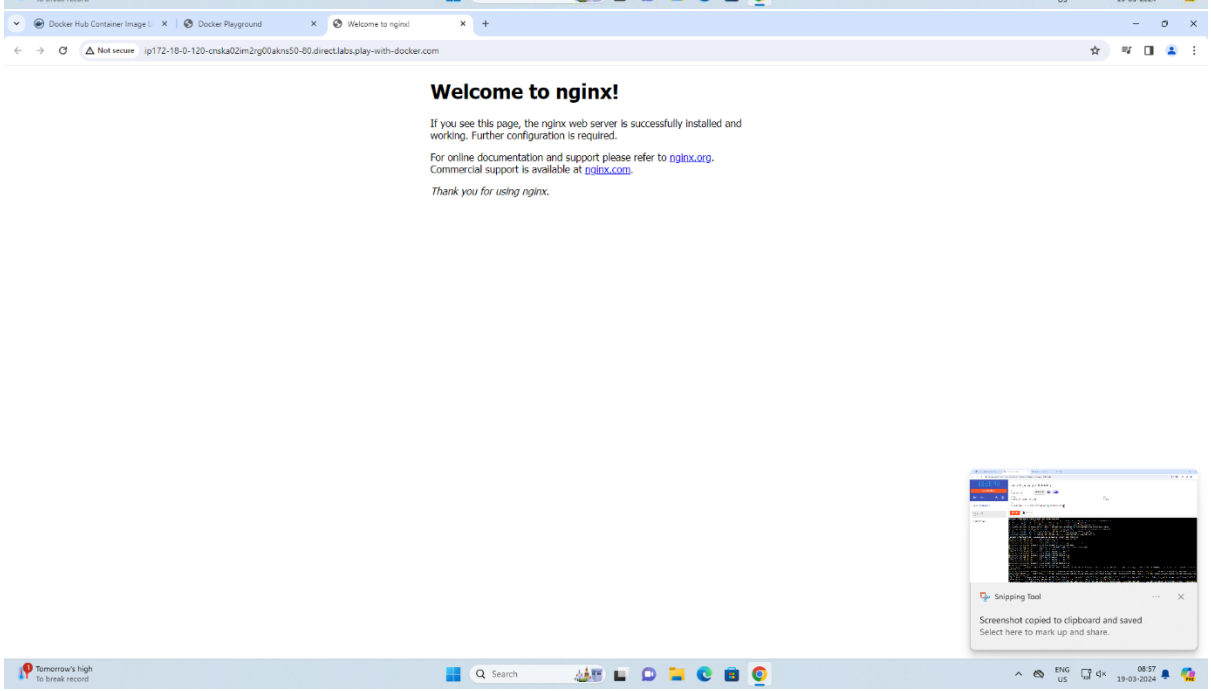

08:43:54
CLOSE SESSION
Instances
+ ADD NEW INSTANCE
192.168.0.18
node1
GIVE FEEDBACK
cnska02i_cnskajgl2o9000fid64g
IP
192.168.0.18
OPEN PORT
Memory
24.42% (976.9MiB / 3.906GiB)
CPU
0.44%
SSH
ssh ip172-18-0-120-cnska02im2rg00akns50@direct.labs.pla
DELETE
EDITOR
$ docker rmi ubuntu
Error response from daemon: conflict: unable to remove repository reference "ubuntu" (must force) - container 87d13d2d2703 is using its referenced image ca2b0f26964c
[node1] (local) root@192.168.0.18 ~
$ docker rmi ubuntu
Error response from daemon: conflict: unable to remove repository reference "ubuntu" (must force) - container ea7e78810588 is using its referenced image ca2b0f26964c
[node1] (local) root@192.168.0.18 ~
$ docker rmi ubuntu
Error response from daemon: conflict: unable to remove repository reference "ubuntu" (must force) - container b8f7de1691f5 is using its referenced image ca2b0f26964c
[node1] (local) root@192.168.0.18 ~
$ docker rmi ubuntu
Error response from daemon: conflict: unable to remove repository reference "ubuntu" (must force) - container 80ea7738c934 is using its referenced image ca2b0f26964c
[node1] (local) root@192.168.0.18 ~
$ docker container ls
CONTAINER ID IMAGE COMMAND CREATED STATUS PORTS NAMES
80ea7738c934 ubuntu "sleep 200" About a minute ago Up About a minute peaceful_rubin
[node1] (local) root@192.168.0.18 ~
$ docker container kill peaceful_rubin
peaceful_rubin
[node1] (local) root@192.168.0.18 ~
$ docker rmi ubuntu
Error response from daemon: conflict: unable to remove repository reference "ubuntu" (must force) - container 82961246bda0 is using its referenced image ca2b0f26964c
[node1] (local) root@192.168.0.18 ~
$ docker container ls
CONTAINER ID IMAGE COMMAND CREATED STATUS PORTS NAMES
[node1] (local) root@192.168.0.18 ~
$ docker rmi ubuntu:latest
Error response from daemon: conflict: unable to remove repository reference "ubuntu:latest" (must force) - container ea7e78810588 is using its referenced image ca2b0f26964c
[node1] (local) root@192.168.0.18 ~
$ docker images
REPOSITORY TAG IMAGE ID CREATED SIZE
ubuntu latest ca2b0f26964c 2 weeks ago 77.9MB
ubuntu 20.04 3cff1c6ff37e 4 weeks ago 72.8MB
mysql latest 019814493c7a 2 months ago 632MB
mysql 5.6 dd3b2a5dcb48 2 years ago 303MB
[node1] (local) root@192.168.0.18 ~
$

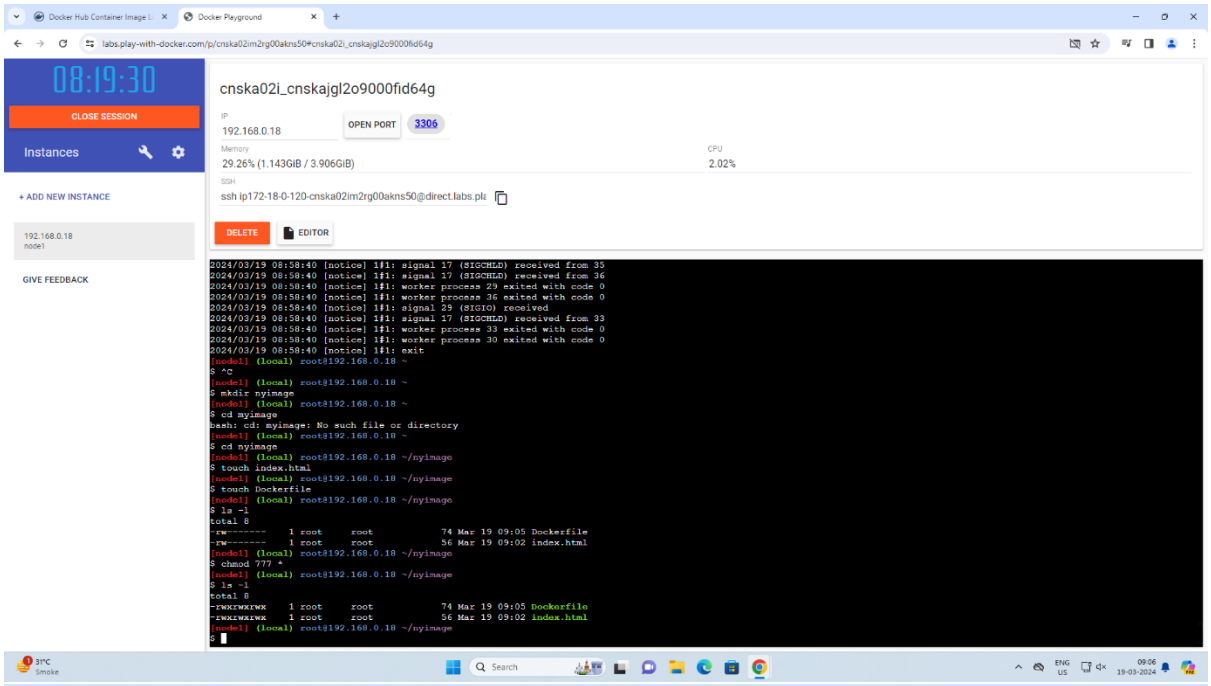

08:37:47
CLOSE SESSION
Instances
+ ADD NEW INSTANCE
192.168.0.18
node1
GIVE FEEDBACK
cnska02i_cnskajgl2o9000fid64g
IP
192.168.0.18
OPEN PORT
3306
Memory
27.44% (1.072GiB / 3.906GiB)
CPU
1.00%
SSH
ssh ip172-18-0-120-cnska02im2rg00akns50@direct.labs.pla
DELETE
EDITOR
peaceful_rubin
[node1] (local) root@192.168.0.18 ~
$ docker rmi ubuntu
Error response from daemon: conflict: unable to remove repository reference "ubuntu" (must force) - container 82961246bda0 is using its referenced image ca2b0f26964c
[node1] (local) root@192.168.0.18 ~
$ docker container ls
CONTAINER ID IMAGE COMMAND CREATED STATUS PORTS NAMES
[node1] (local) root@192.168.0.18 ~
$ docker rmi ubuntu:latest
Error response from daemon: conflict: unable to remove repository reference "ubuntu:latest" (must force) - container ea7e78810588 is using its referenced image ca2b0f26964c
[node1] (local) root@192.168.0.18 ~
$ docker images
REPOSITORY TAG IMAGE ID CREATED SIZE
ubuntu latest ca2b0f26964c 2 weeks ago 77.9MB
ubuntu 20.04 3cff1c6ff37e 4 weeks ago 72.8MB
mysql latest 019814493c7a 2 months ago 632MB
mysql 5.6 dd3b2a5dcb48 2 years ago 303MB
[node1] (local) root@192.168.0.18 ~
$ docker run --name tsec server -d -p -3306:3306 -e MYSQL_ROOT_PASSWORD=123 mysql:latest
Unable to find image 'server:latest' locally
docker: Error response from daemon: pull access denied for server, repository does not exist or may require 'docker login': denied: requested access to the resource is denied.
See 'docker run --help'.
[node1] (local) root@192.168.0.18 ~
$ docker run --name tsec server -d -p 3306:3306 -e MYSQL_ROOT_PASSWORD=123 mysql:latest
Unable to find image 'server:latest' locally
docker: Error response from daemon: pull access denied for server, repository does not exist or may require 'docker login': denied: requested access to the resource is denied.
See 'docker run --help'.
[node1] (local) root@192.168.0.18 ~
$ docker ps
CONTAINER ID IMAGE COMMAND CREATED STATUS PORTS NAMES
[node1] (local) root@192.168.0.18 ~
$ docker run --name tsecserver -d -p 3306:3306 -e MYSQL_ROOT_PASSWORD=123 mysql:latest
0540721700e66d0930c47a11a849d059706b243b828c6764efa9bc2ce7db4842
[node1] (local) root@192.168.0.18 ~
$ docker exec -it tsecserver /bin/bash
bash-4.4#

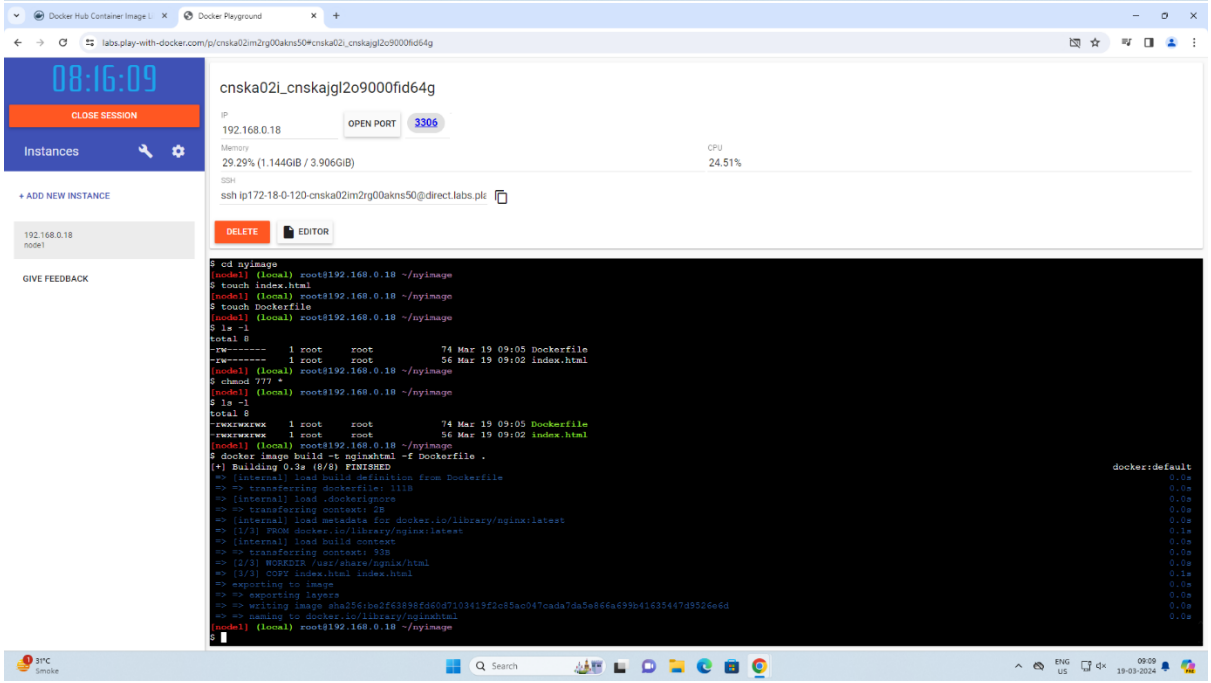

08:36:30
CLOSE SESSION
Instances
+ ADD NEW INSTANCE
192.168.0.18
node1
GIVE FEEDBACK
cnska02i_cnskajgl2o9000fid64g
IP
192.168.0.18
OPEN PORT
3306
Memory
27.51% (1.075GiB / 3.906GiB)
CPU
0.27%
SSH
ssh ip172-18-0-120-cnska02im2rg00akns50@direct.labs.pla
DELETE
EDITOR
ubuntu 20.04 3cff1c6ff37e 4 weeks ago 72.8MB
mysql latest 019814493c7a 2 months ago 632MB
mysql 5.6 dd3b2a5dcb48 2 years ago 303MB
[node1] (local) root@192.168.0.18 ~
$ docker run --name tsec server -d -p -3306:3306 -e MYSQL_ROOT_PASSWORD=123 mysql:latest
Unable to find image 'server:latest' locally
docker: Error response from daemon: pull access denied for server, repository does not exist or may require 'docker login': denied: requested access to the resource is denied.
See 'docker run --help'.
[node1] (local) root@192.168.0.18 ~
$ docker run --name tsec server -d -p 3306:3306 -e MYSQL_ROOT_PASSWORD=123 mysql:latest
Unable to find image 'server:latest' locally
docker: Error response from daemon: pull access denied for server, repository does not exist or may require 'docker login': denied: requested access to the resource is denied.
See 'docker run --help'.
[node1] (local) root@192.168.0.18 ~
$ docker ps
CONTAINER ID IMAGE COMMAND CREATED STATUS PORTS NAMES
[node1] (local) root@192.168.0.18 ~
$ docker run --name tsecserver -d -p 3306:3306 -e MYSQL_ROOT_PASSWORD=123 mysql:latest
0540721700e66d0930c47a11a849d059706b243b828c6764efa9bc2ce7db4842
[node1] (local) root@192.168.0.18 ~
$ docker exec -it tsecserver /bin/bash
bash-4.4# mysql -u root -p123
mysql: [Warning] Using a password on the command line interface can be insecure.
Welcome to the MySQL monitor. Commands end with ; or \g.
Your MySQL connection id is 8
Server version: 8.3.0 MySQL Community Server - GPL
Copyright (c) 2000, 2024, Oracle and/or its affiliates.
Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.
Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.
mysql>


08:34:58
CLOSE SESSION
Instances
+ ADD NEW INSTANCE
192.168.0.18
node1
GIVE FEEDBACK
cnska02i_cnskajgl2o9000fid64g
IP
192.168.0.18
OPEN PORT
3306
Memory
24.31% (972.6MiB / 3.906GiB)
CPU
2.10%
SSH
ssh ip172-18-0-120-cnska02im2rg00akns50@direct.labs.pla
DELETE
EDITOR
$ docker ps
CONTAINER ID IMAGE COMMAND CREATED STATUS PORTS NAMES
[node1] (local) root@192.168.0.18 ~
$ docker run --name tsecserver -d -p 3306:3306 -e MYSQL_ROOT_PASSWORD=123 mysql:latest
0540721700e66d0930c47a11a849d059706b243b828c6764efa9bc2ce7db4842
[node1] (local) root@192.168.0.18 ~
$ docker exec -it tsecserver /bin/bash
bash-4.4# mysql -u root -p123
mysql: [Warning] Using a password on the command line interface can be insecure.
Welcome to the MySQL monitor. Commands end with ; or \g.
Your MySQL connection id is 8
Server version: 8.3.0 MySQL Community Server - GPL
Copyright (c) 2000, 2024, Oracle and/or its affiliates.
Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.
Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.
mysql> CREATE DATABASE exp9;
Query OK, 1 row affected (0.00 sec)
mysql> CREATE TABLE docker;
ERROR 1046 (3D000): No database selected
mysql> USE DATABASE exp9;
ERROR 1049 (42000): Unknown database 'DATABASE'
mysql> USE exp9;
Database changed
mysql> SHOW TABLES;
Empty set (0.00 sec)
mysql> CREATE TABLE hi;
ERROR 4028 (HY000): A table must have at least one visible column.
mysql>


08:28:40
CLOSE SESSION
Instances
+ ADD NEW INSTANCE
192.168.0.18
node1
GIVE FEEDBACK
cnska02i_cnskajgl2o9000fid64g
IP
192.168.0.18
OPEN PORT
80
3306
Memory
30.01% (1.172GiB / 3.906GiB)
CPU
2.04%
SSH
ssh ip172-18-0-120-cnska02im2rg00akns50@direct.labs.pla
DELETE
EDITOR
Status: Downloaded newer image for nginx:latest
/docker-entrypoint.sh: /docker-entrypoint.d/ is not empty, will attempt to perform configuration
/docker-entrypoint.sh: Looking for shell scripts in /docker-entrypoint.d/
/docker-entrypoint.sh: Launching /docker-entrypoint.d/10-listen-on-ipv6-by-default.sh
10-listen-on-ipv6-by-default.sh: info: Getting the checksum of /etc/nginx/conf.d/default.conf
10-listen-on-ipv6-by-default.sh: info: Enabled listen on IPv6 in /etc/nginx/conf.d/default.conf
/docker-entrypoint.sh: Sourcing /docker-entrypoint.d/15-local-resolvers.envsh
/docker-entrypoint.sh: Launching /docker-entrypoint.d/20-envsubst-on-templates.sh
/docker-entrypoint.sh: Launching /docker-entrypoint.d/30-tune-worker-processes.sh
/docker-entrypoint.sh: Configuration complete; ready for start up
2024/03/19 08:56:49 [notice] 1#1: using the "epoll" event method
2024/03/19 08:56:49 [notice] 1#1: nginx/1.25.4
2024/03/19 08:56:49 [notice] 1#1: built by gcc 12.2.0 (Debian 12.2.0-14)
2024/03/19 08:56:49 [notice] 1#1: OS: Linux 4.4.0-210-generic
2024/03/19 08:56:49 [notice] 1#1: getrlimit(RLIMIT_NOFILE): 1048576:1048576
2024/03/19 08:56:49 [notice] 1#1: start worker processes
2024/03/19 08:56:49 [notice] 1#1: start worker process 29
2024/03/19 08:56:49 [notice] 1#1: start worker process 30
2024/03/19 08:56:49 [notice] 1#1: start worker process 31
2024/03/19 08:56:49 [notice] 1#1: start worker process 32
2024/03/19 08:56:49 [notice] 1#1: start worker process 33
2024/03/19 08:56:49 [notice] 1#1: start worker process 34
2024/03/19 08:56:49 [notice] 1#1: start worker process 35
2024/03/19 08:56:49 [notice] 1#1: start worker process 36
172.18.0.1 - - [19/Mar/2024:08:56:59 +0000] "GET / HTTP/1.1" 200 615 "-" "Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/122.0.0.0 Safari/537.36" "-"
2024/03/19 08:56:59 [error] 34#34: *3 open() "/usr/share/nginx/html/favicon.ico" failed (2: No such file or directory), client: 172.18.0.1, server: localhost, request: "GET /favicon.ico HTTP/1.1", host: "ip172-18-0-120-cnska02im2rg00akns50-80.direct.labs.play-with-docker.com", referrer: "http://ip172-18-0-120-cnska02im2rg00akns50-80.direct.labs.play-with-docker.com/"
172.18.0.1 - - [19/Mar/2024:08:56:59 +0000] "GET /favicon.ico HTTP/1.1" 404 555 "http://ip172-18-0-120-cnska02im2rg00akns50-80.direct.labs.play-with-docker.com/" "Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/122.0.0.0 Safari/537.36" "-"
172.18.0.1 - - [19/Mar/2024:08:57:03 +0000] "\x16\x03\x01\x07\x10\x01\x00\x07\x00\x03\x03\x04$\xc1h\x8Ez0'\x11Q\xDB\xFE\x5C:D_/\xD1I4x\x1AN\xFDf\xA8\xFA\xDD|Y\xB1[\x1B1T\xF10\x91\x18\xC9\x13\xAD\xF7\xBB0r/~Sfy\xA9u\x18p\xAD\xD7\xC3\xC5;z7\x8Dy\x00 \x8A\x8A\x12\x01\x13\x02\x13\x03\xC0+\xC0/\xC0,\xC00\xCC\xA9\xCC\xA8\xC0\x13\xC0\x14\x00\x9C\x00\x9D\x00/\x005\x01\x00\x06\xA323\x00\x00\x00+\x00\x07\x06\x9A\x9A\x03\x04\x03\x03\x00\x10\x00\x0E\x00\x0C\x02h2\x08http/1.1\x00" 400 157 "-" "-" "-"
172.18.0.1 - - [19/Mar/2024:08:57:04 +0000] "\x16\x03\x01\x070\x01\x00\x07,\x03\x03'B!\x8D,$E\x8FtE|\x8A\xE5\x91h\x95\xED" 400 157 "-" "-" "-"
Tomorrow's high
To break record
Search
ENG
US
08:57
19-03-2024


Not secure
ip172-18-0-120-cnska02im2rg00akns50-80.direct.labs.play-with-docker.com
Welcome to nginx!
If you see this page, the nginx web server is successfully installed and working. Further configuration is required.
For online documentation and support please refer to nginx.org.
Commercial support is available at nginx.com.
Thank you for using nginx.
Snipping Tool
Screenshot copied to clipboard and saved
Select here to mark up and share.
Tomorrow's high
To break record
Search
ENG
US
08:57
19-03-2024


08:19:30
CLOSE SESSION
Instances
+ ADD NEW INSTANCE
192.168.0.18
node1
GIVE FEEDBACK
cnska02i_cnskajgl2o9000fid64g
IP
192.168.0.18
OPEN PORT
3306
Memory
29.26% (1.143GiB / 3.906GiB)
CPU
2.02%
SSH
ssh ip172-18-0-120-cnska02im2rg00akns50@direct.labs.pl
DELETE
EDITOR
2024/03/19 08:58:40 [notice] 1#1: signal 17 (SIGCHLD) received from 35
2024/03/19 08:58:40 [notice] 1#1: signal 17 (SIGCHLD) received from 36
2024/03/19 08:58:40 [notice] 1#1: worker process 29 exited with code 0
2024/03/19 08:58:40 [notice] 1#1: worker process 36 exited with code 0
2024/03/19 08:58:40 [notice] 1#1: signal 29 (SIGIO) received
2024/03/19 08:58:40 [notice] 1#1: signal 17 (SIGCHLD) received from 33
2024/03/19 08:58:40 [notice] 1#1: worker process 33 exited with code 0
2024/03/19 08:58:40 [notice] 1#1: worker process 30 exited with code 0
2024/03/19 08:58:40 [notice] 1#1: exit
[node1] (local) root@192.168.0.18 ~
$ ^C
[node1] (local) root@192.168.0.18 ~
$ mkdir nyimage
[node1] (local) root@192.168.0.18 ~
$ cd myimage
bash: cd: myimage: No such file or directory
[node1] (local) root@192.168.0.18 ~
$ cd nyimage
[node1] (local) root@192.168.0.18 ~/nyimage
$ touch index.html
[node1] (local) root@192.168.0.18 ~/nyimage
$ touch Dockerfile
[node1] (local) root@192.168.0.18 ~/nyimage
$ ls -l
total 8
-rw------- 1 root root 74 Mar 19 09:05 Dockerfile
-rw------- 1 root root 56 Mar 19 09:02 index.html
[node1] (local) root@192.168.0.18 ~/nyimage
$ chmod 777 *
[node1] (local) root@192.168.0.18 ~/nyimage
$ ls -l
total 8
-rwxrwxrwx 1 root root 74 Mar 19 09:05 Dockerfile
-rwxrwxrwx 1 root root 56 Mar 19 09:02 index.html
[node1] (local) root@192.168.0.18 ~/nyimage
$


08:16:09
CLOSE SESSION
Instances
+ ADD NEW INSTANCE
192.168.0.18
node1
GIVE FEEDBACK
cnska02i_cnskajgl2o9000fid64g
IP
192.168.0.18
OPEN PORT
3306
Memory
29.29% (1.144GiB / 3.906GiB)
CPU
24.51%
SSH
ssh ip172-18-0-120-cnska02im2rg00akns50@direct.labs.pl
DELETE
EDITOR
$ cd nyimage
[node1] (local) root@192.168.0.18 ~/nyimage
$ touch index.html
[node1] (local) root@192.168.0.18 ~/nyimage
$ touch Dockerfile
[node1] (local) root@192.168.0.18 ~/nyimage
$ ls -l
total 8
-rw------- 1 root root 74 Mar 19 09:05 Dockerfile
-rw------- 1 root root 56 Mar 19 09:02 index.html
[node1] (local) root@192.168.0.18 ~/nyimage
$ chmod 777 *
[node1] (local) root@192.168.0.18 ~/nyimage
$ ls -l
total 8
-rwxrwxrwx 1 root root 74 Mar 19 09:05 Dockerfile
-rwxrwxrwx 1 root root 56 Mar 19 09:02 index.html
[node1] (local) root@192.168.0.18 ~/nyimage
$ docker image build -t nginxhtml -f Dockerfile .
[+] Building 0.3s (8/8) FINISHED
docker:default
=> [internal] load build definition from Dockerfile
=> => transferring dockerfile: 111B
=> [internal] load .dockerignore
=> => transferring context: 2B
=> [internal] load metadata for docker.io/library/nginx:latest
=> [1/3] FROM docker.io/library/nginx:latest
=> [internal] load build context
=> => transferring context: 93B
=> [2/3] WORKDIR /usr/share/nginx/html
=> [3/3] COPY index.html index.html
=> exporting to image
=> => exporting layers
=> => writing image sha256:be2f63898fd60d7103419f2c85ac047cada7da5e866a699b41635447d9526e6d
=> => naming to docker.io/library/nginxhtml
[node1] (local) root@192.168.0.18 ~/nyimage
$