

VOLUME 01

2026

CONTAINER

MAGAZINE

**COURSE
COMPANION
GUIDE**

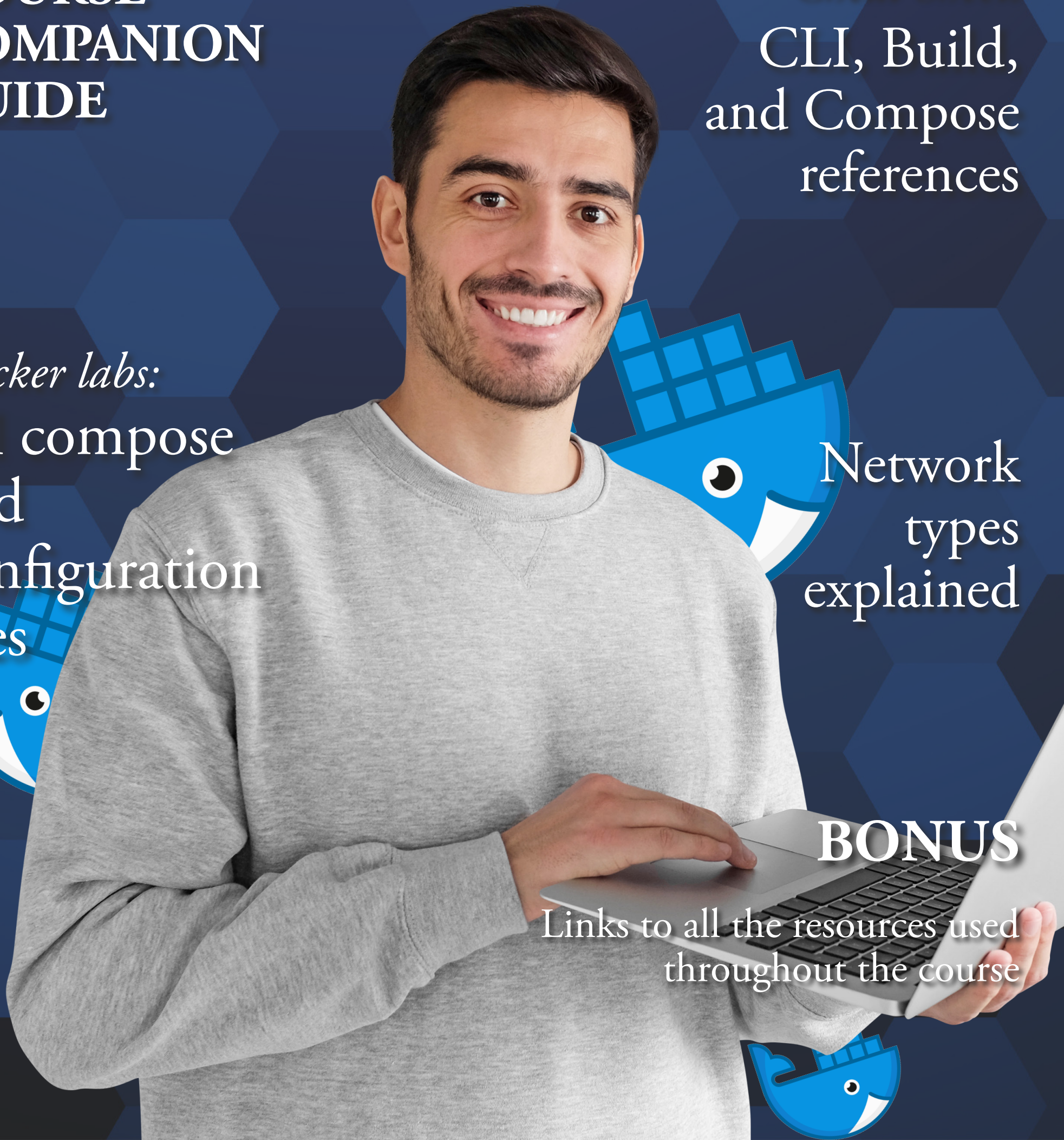
Cheat Sheets
CLI, Build,
and Compose
references

Docker labs:
All compose
and
configuration
files

Network
types
explained

BONUS

Links to all the resources used
throughout the course



A container is a process running on a system, but with isolation and a fully prepared environment.

02 **CLI CHEAT SHEET**

06 **NETWORKS**

15 **DOCKER COMPOSE**

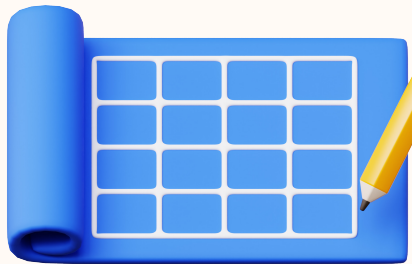
28 **DOCKERFILES**

34 **LAB FILES**

63 **LINKS**

Container States

CREATED



A container has been created

It is not yet in the running state

It does not consume CPU or memory

RUNNING

The container has been started

Processes are running in the isolated container environment

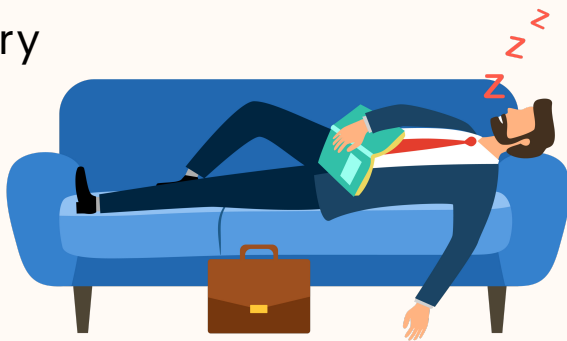


PAUSED

Processes in the container are suspended

Continues to consume memory

Does not consume cpu



RESTARTING



In the process of restarting

EXITED

The container has stopped

No memory and CPU are used

may be a natural exit, or an error

Container can be restarted

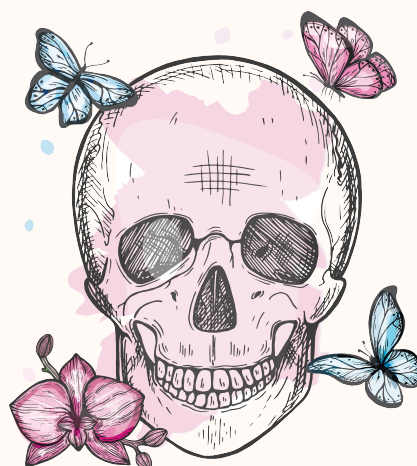


DEAD

A non-functioning container

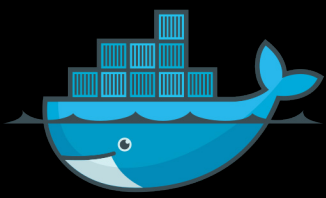
Container could not be removed

Some resources are locked by an external process



Docker Cheat Sheet

Docker
Command
Line



docker

COMMAND CHEAT SHEET



RUNNING CONTAINERS

`docker run <image-name>`
Start a container from an image

`docker run --name <name>`
Give a container a friendly name

`docker run -p <outside-port>:<inside-port>`
Publish a container's port on the host

`docker run --rm`
Automatically remove a container when it stops

`docker run -d`
Run a container in *detached* mode

`docker run -p <image> <command>`
Run a command within the container when the container starts

`docker run --entrypoint <entrypoint>`
Start a container with a custom entry point

`docker run -e <ENV-VARIABLE>=<value>`
Pass an environment variable to the container at runtime

`docker run --labels <label>`
Add a label to a container at runtime



IMAGE MANAGEMENT

`docker image list`
Show a list of images available locally

`docker image inspect <image>`
Show additional information about an image

`docker pull <repository>:<tag>`
Pull an image from Docker Hub

`docker tag <image> <username>/<image>:<tag>`
Add tags to an existing image

`docker login -u <username>`
Login to an image registry, like Docker Hub

`docker push`
Push an image to a registry

`docker logout`
Logout from a registry

`docker system df`
Show disk space used by images



CONTAINER MANAGEMENT

`docker create <image>`
Create a container, but don't start it

`docker start <container>`
Start a container that's in the *created* state

`docker stop <id>`
Stop a running container by its ID

`docker stop <name>`
Stop a running container by its name

`docker pause <container>`
Pause a *running* container

`docker unpause <container>`
Unpause, or resume, a *paused* container

`docker ps`
Show all *running* containers

`docker ps -a`
Show all containers, in any state

`docker rm <id>`
Remove a stopped container, by its ID

`docker rm <name>`
Remove a stopped container, by its name

`docker inspect <container>`
Get information about a running container



INTERACTIVE COMMANDS

`docker run -it`
Start in interactive mode, with a *pseudo-tty*

`docker exec <container> <command>`
Send a command to a container

`docker exec -it`
Connect to the container interactively



All commands start with the word 'docker'

Add '--help' to any major command to get a summary of options



STORAGE

```
docker volume create <name>
```

Create a named volume

```
docker volume list
```

Show a list of volumes on the host

```
docker volume inspect <volume>
```

Get more information about a volume

```
docker volume rm <id>
```

Remove a volume, by its ID

```
docker volume rm <name>
```

Remove a volume, by its name

```
docker run --volume <volume>:<mount-point>
```

Mount a named volume at runtime

```
docker run --mount  
  type=volume,src=<name>,dst=<mount>
```

Alternate method to mounting a volume

```
docker run --volume <mount-point>
```

Add an unnamed volume at container runtime

```
docker run --volume <host-directory>:<mount-point>
```

Create a bind mount at runtime



COMPOSE

```
docker compose up
```

Start an application from a compose file

```
docker compose up -d
```

Start an application in detached mode

```
docker compose up --build
```

Rebuild any container with a build section

```
docker compose up --force-recreate
```

Restart the containers in the application

```
docker compose ls
```

See compose apps that are running

```
docker compose down
```

Stop a running compose application

```
docker compose version
```

Get the version of docker compose

```
docker compose logs
```

Show logs from a compose application

```
docker compose logs [-f]
```

Show compose logs in real time



NETWORK

```
docker network create <name>
```

Create a bridge network

```
docker network create -d macvlan  
  -o parent=<host-NIC> <name>
```

Create a MACVLAN network

```
docker network create -d ipvlan -o ipvlan_mode=12  
  -o parent=<host-NIC> <name>
```

Create an IPVLAN network in L2 mode

```
docker network create -d ipvlan -o ipvlan_mode=12  
  -o parent=<host-NIC>.<vlan> <name>
```

Create an IPVLAN network on a specific VLAN

```
docker network create -d ipvlan -o ipvlan_mode=13  
  -o parent=<host-NIC> <name>
```

Create an IPVLAN network in L3 mode

```
docker network create ... --subnet=<CIDR>
```

Assign a subnet to a network

```
docker network create ... --gateway=<ip>
```

Assign a default gateway to a network

```
docker network ls
```

Show a list of all networks on the host

```
docker network inspect <name>
```

Get more information about a network

```
docker run --network <name>
```

Assign a container to a network at runtime

```
docker run --ip <ip>
```

Assign an IP address to a container at runtime



BUILD

```
docker build . -t <image>:<tag>
```

Build an image from a dockerfile in the working directory

```
docker build -f <dockerfile>
```

Build an image using a specific dockerfile

```
docker history <image>
```

See the layers of an image



New syntax for container management uses the format 'docker container ...'

'docker ps' is 'docker container ls'

'docker run' is 'docker container run'



RESOURCES AND LIMITS

```
docker stats <container>
```

Check container resource usage

```
docker run --memory=<value>m
```

Set the memory limit on a container

```
docker run --memory=<value> --memory-swap=<value>
```

Set physical and swap memory limits

```
docker run --device-read-bps <disk>:<limit>
```

Limit disk reads by bytes per second

```
docker run --device-write-bps <disk>:<limit>
```

Limit disk writes by bytes per second

```
docker run --device-read-iops <disk>:<limit>
```

Limit disk read by IOPS

```
docker run --device-write-iops <disk>:<limit>
```

Limit disk write by IOPS

```
docker run --cpus="<value>"
```

Hard limit to CPU cores

```
docker run --cpu-shares=<value>
```

Soft CPU limit based on shares

```
docker run --cpu-period=<value> --cpu-quota=<value>
```

Hard CPU limit by time slices

```
docker run --pids-limit <value>
```

Set the limit of processes that can start

```
docker run --ulimit nfile=<soft>:<hard>
```

Soft and hard limits on the number of file descriptors

```
docker run --ulimit nproc=<soft>:<hard>
```

Soft and hard limits on the number of processes



LOGGING

```
docker logs <container>
```

Get logs from a container

```
docker logs --follow <container>
```

Get logs as they are generated in real-time

```
docker logs --since <second>s <container>
```

Show logs within a timeframe

```
docker logs --timestamps <container>
```

Add timestamps to logs



SECURITY

```
docker run ... --read-only
```

Set the filesystem to read only

```
docker run ... --cap-drop <capability>
```

Drop a Linux capability

```
docker run ... --cap-drop ALL
```

Drop all capabilities

```
docker run ... --cap-add <capability>
```

Add a Linux capability

```
docker run ... --privileged
```

Runs a container with virtually no isolation

```
docker run <image> id
```

Start a container and get the user ID

```
docker run -u $(id -u):$(id -g)
```

Run a container with a specific UID

```
docker run --pid <mode>
```

Set the process ID namespace for the container

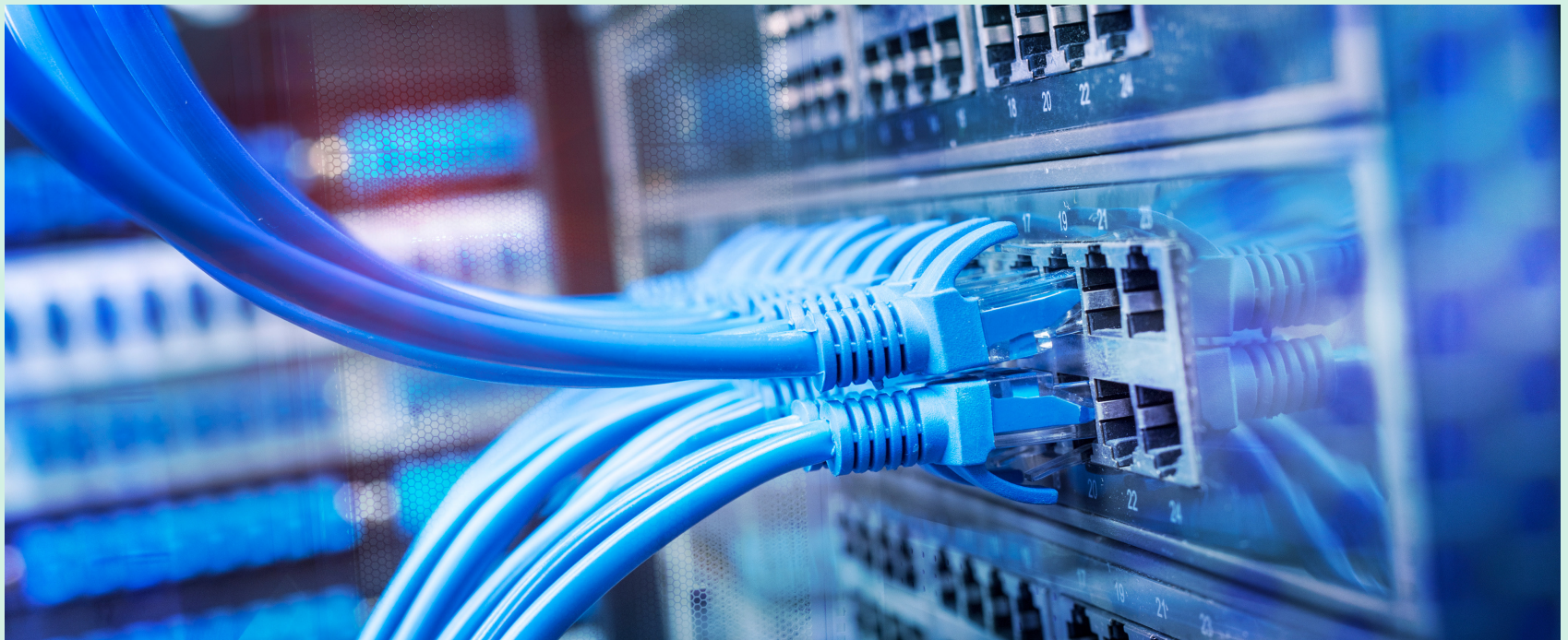
Docker's official cheat sheet:

https://docs.docker.com/get-started/docker_cheatsheet.pdf



Command line documentation:

<https://docs.docker.com/reference/cli/docker/>



DOCKER



NETWORKS



NETWORK TYPES

Bridge

An internal network. Allows containers to communicate with each other.

Host

Share the host's network. Ports in a container are mapped to the host.

MACVLAN

A Layer 2 network. Each container has its own MAC and IP address.

IPVLAN

Layer 2 or 3. Containers appear to be devices on the external network.

None

Complete Isolation. No network is available to these containers.

Exposing Ports

A container has some of its TCP or UDP ports mapped directly to the host.

This is like configuring NAT or PAT on a router.

```
docker run -p 80:80 ...
```

Bridge Network

Bridge networks are used within Docker to allow containers to communicate with each other.

Ports do not need to be exposed to allow this type of communication.

Docker has a default bridge network, but creating custom networks is preferred.

Separate bridge networks provides isolation between containers.

```
docker network create <name>
```

```
docker run --network <name> ...
```


Host

The container shares the host's network stack directly.
There is no need to manually expose ports.

This allows the use of non-TCP/UDP traffic.

This does not need a network to be created first.

```
docker run --network host ...
```

MACVLAN

A layer-2 network, where each container gets its own unique MAC and IP address.

The container appears to be a device on the external network.

Allows layer-2 traffic, such as ARP.

```
docker network create \  
  -d macvlan \  
  --subnet=<subnet>/<mask> \  
  --gateway=<ip> \  
  -o parent=<int> \  
  <name>
```


IPVLAN (Layer-2)

Each container has its own IP address, but they all share the MAC of the host.

This makes each container appear to be a device on the external network.

This also allows the use of layer-2 traffic such as ARP and DHCP.

This network needs to be associated with a subnet that already exists on one of the host's interfaces.

This also needs to use an external default gateway.

```
docker network create \  
  -d ipvlan \  
  --subnet=<subnet>/<mask> \  
  --gateway=<ip> \  
  -o ipvlan_mode=12 \  
  -o parent=<int> \  
  <name>
```

```
docker run --network=<name> \  
  --ip=<ip> ...
```

IPVLAN (Layer-3)

A layer-3 network, which uses routing rather than bridging.

Each container has its own IP address, and appear to be devices on the external network.

This network has a subnet which is managed within docker. The docker host is the default gateway.

External routers need to be configured to forward traffic to the docker host. This can be done with static routes, or a dynamic routing protocol such as OSPF or BGP.

The host's Kernel must be configured to allow IP forwarding for this to work.

```
docker network create -d ipvlan \  
  --subnet=<subnet>/<mask> \  
  -o parent=<int> \  
  -o ipvlan_mode=13 \  
  <name>
```

```
docker run --network=<name> \  
  --ip=<ip> ...
```


None

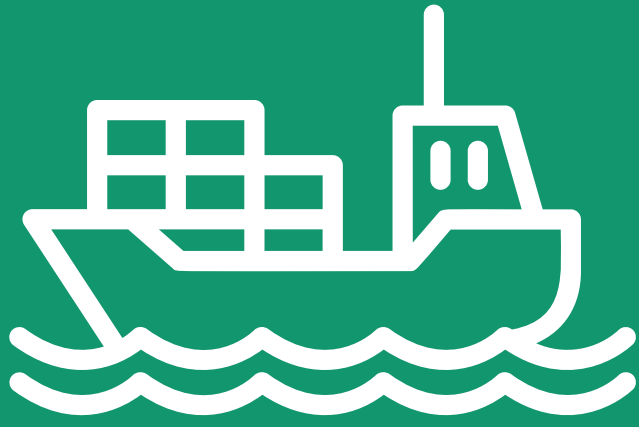
The simplest network type in docker.

This completely isolates a container from any network, both internal and external.

The container has a loopback interface, but no other interfaces.

This is a built in network. There is no need to manually create it.

```
docker run --network none ...
```



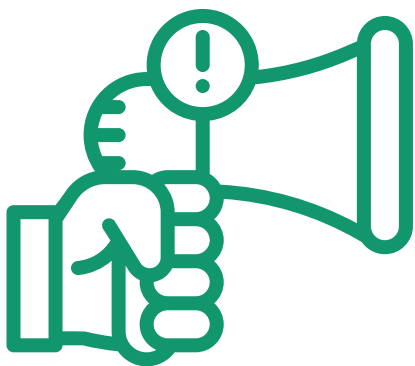
INTERACTIVE CONTAINERS



START A CONTAINER INTERACTIVELY

Use '-it' to enable interactive mode, as well as creating a pseudo-tty for the container.

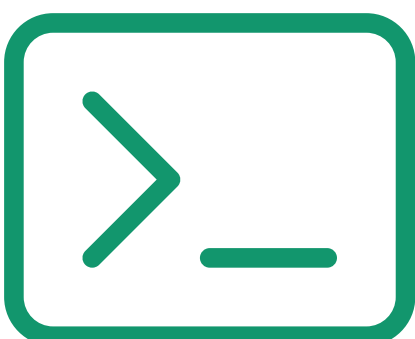
```
docker run -d -it ubuntu /bin/bash
```



RUN COMMANDS ON A CONTAINER

Send commands to a container that is already running.

```
docker exec <container> <command>
```



CONNECT TO THE CONTAINER'S SHELL

Connect to a container's shell interactively.

'sh' refers to the container's default shell, which may vary from container to container.

```
docker exec -it <container> sh
```


STORAGE



Persistent vs Non-persistent

- Persistent storage survives the container's lifecycle
- Non-persistent storage is lost when the container is stopped
- The root filesystem in a container is non-persistent

NAMED VOLUMES

Persistent Storage
Attached to a container at runtime

ANONYMOUS VOLUMES

Non-Persistent Storage
Created at container runtime

LOCATION

`/var/lib/docker/volumes`

Bind Mounts

- Persistent storage
- Map a real location on the host's filesystem to a location in the container
- Can be mounted as read only or read write

Read-Only Filesystem

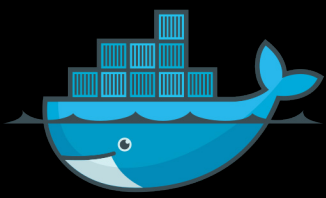
- Mount a container's filesystem as read only
- Prevents any modifications to the internal filesystem
- Security best practice

Temporary Filesystem

- A virtual filesystem created in memory
- Non-persistent
- High-performance for temporary data

Docker Cheat Sheet

Docker
Compose
Reference



docker

Compose Reference



COMPOSE FILES

Describe an application with one or more containers

Written in YAML

compose.yaml or compose.yml by default

MAIN STRUCTURE

name:	Optional friendly name for the project
services:	A group of services (containers). The name of each service is user-defined.
container_name:	Optional friendly name for a container within a service
image:	The image used to start the container. Optionally includes a project name, repository, and tag.
restart:	The restart policy. Options are no, always, on-failure, unless-stopped
command:	Override the container's default command. Used to customise the container's start up behaviour
extra_hosts:	Provides a static hostname to IP mapping

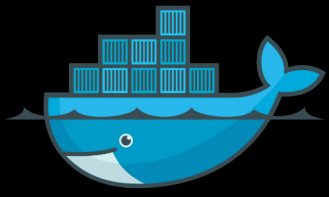


Compose Reference

MAIN STRUCTURE - EXAMPLE

```
name: nginx-webserver

services:
  nginx:
    container_name: nginx-web
    image: nginx:latest
    command: [nginx, "-g", "daemon off;"]
    extra_hosts:
      - "cloudflare-dns:1.1.1.1"
    restart: unless-stopped
```

Compose Reference

NETWORKING

networks:	The top level area where a network is defined
driver:	Set the driver type for a network
external:	When true, this indicates that the network has already been created, separate to this compose file.

networks:	Under a service, this is the area to add a list of networks.
ipv4_address:	Assign a static IPv4 address
ports:	Expose, or 'publish' container ports on the host

NETWORKING - EXAMPLE

```
services:
  web-app:
    networks:
      - internal
    ports:
      - 80:80

networks:
  internal:
    driver: bridge
```



Compose Reference

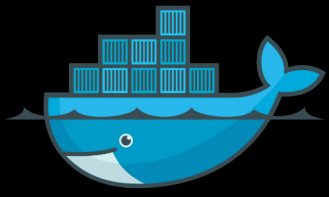
STORAGE

volumes:	The top level area where named volumes are defined.
<name>:	A named volume. No additional parameters are needed.
volumes:	Under a service, this is the area to add a list of volumes.
read-only:	When true, the root filesystem of the container is read-only.
tmpfs:	Create a virtual filesystem in memory.

STORAGE - EXAMPLE

```
services:
  my-service:
    volumes:
      - data:/data          # Named volume
      - /cache              # Anon volume
      - /local:/config:ro   # Bind mount, read only
    read_only: true         # Read only FS
    tmpfs:
      - /tmp

volumes:
  data:
```

Compose Reference

ENVIRONMENT VARIABLES

environment: Area within the service to place environment variables.

<ENV>= Create an environment variable and assign a value

\${VAR} Reference an environment variable

\${VAR:-default}
Set a default value, or use an environment variable if it exists

ENVIRONMENT VARIABLES - EXAMPLE

```
services:
```

```
  special-app:
```

```
    environment:
```

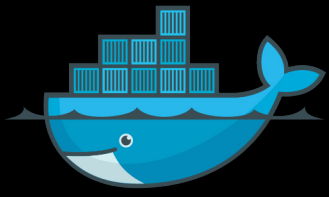
```
      URL=networkdirection.net
```

```
      TZ=Australia/Sydney
```

```
      TAG=v1
```

```
    # Default to 'latest' if 'TAG' not set
```

```
    image: myImage:${TAG:-latest}
```



Compose Reference

HEALTH CHECKS

healthcheck: Area within the service to configure health checks.

test: The command to use to determine health.

interval: Time between checks. Default is 30s.

timeout: Max time to wait before marking the test as failed.
Default is 30s.

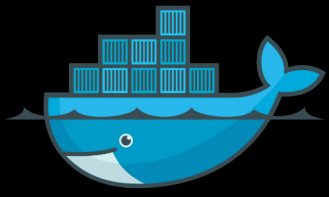
retries: Number of consecutive failures before marking the container as unhealthy. Defaults to 3.

start_period: Grace period between container start and health check start.

depends_on: Configure a service to rely on another service.
May optionally set health conditions.

HEALTH CHECK - EXAMPLE

```
services:
  healthy-container:
    healthcheck:
      test: ["ping", "127.0.0.1"]
      interval: 10s
      timeout: 4s
      retries: 5
      start_period: 10s
```

Compose Reference

SECURITY

<code>cap_drop:</code>	Drop some or all Linux capabilities.
<code>cap_add:</code>	Add Linux capabilities to a container.
<code>sysctls:</code>	Linux system controls. Configure kernel modules.

SECURITY - EXAMPLE

```
services:
  secure-container:
    cap_drop:
      - ALL
    cap_add:
      - NET_ADMIN
    sysctls:
      net.ipv4.conf.all.arp_ignore: 1
```



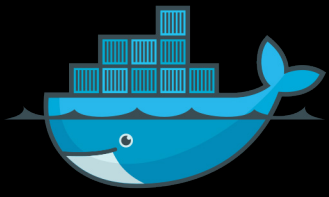
Compose Reference

BUILD

- build:** The area in the service that contains build information.
- context:** The location of the files used in the build.
- dockerfile:** The location of the dockerfile.

SECURITY - EXAMPLE

```
services:
  custom-container:
    build:
      context: .
      dockerfile: Dockerfile
```



Compose Reference

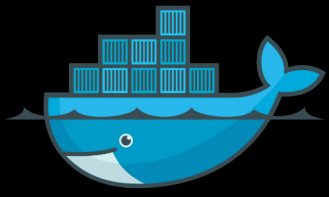
RESOURCE LIMITS

<code>deploy:</code>	The new location for some limits.
<code>memory:</code>	The newer way to limit memory usage; Under 'deploy'.
<code>reservations:</code>	Reserve memory; Under 'deploy'.
<code>mem_limit:</code>	The older way to set a memory limit; Under 'services'.

<code>ulimits:</code>	Limit open files and processes.
<code>nofile:</code>	Set the max number of open file descriptors.
<code>nproc:</code>	Set the max number of open processes.

<code>blkio_config:</code>	Set block IO device limits (hard drives)
<code>device_read_iops</code>	Limit read IOPS
<code>device_write_iops:</code>	Limit write IOPS
<code>device_read_bps</code>	Limit read BPS
<code>device_write_bps:</code>	Limit write BPS

<code>mem_swap_limit:</code>	Set a limit on swap memory
------------------------------	----------------------------



Compose Reference

RESOURCE LIMITS - EXAMPLE

```
services:
  limited-container:
    deploy:
      resources:
        limits:
          memory: 512M
        reservations:
          memory: 256M
    ulimits:
      nofile:
        soft: 1024
        hard: 2048
      nproc: 128
    blkio_config:
      device_read_iops:
        /dev/sda:100
      device_write_iops:
        /dev/sda:100
      device_read_bps:
        /dev/sda:5mb
      device_write_bps:
        /dev/sda:5mb
    memswap_limit: 512M
```

KERNEL SECURITY

Namespaces



Provides container isolation.

These provide a partitioned view of the system.
The container thinks it can see the entire system,
while it really has limited access

Containers running in different namespaces
cannot see each other

Control Groups



Manages access to resources

This limits the resources each container has access to, so
one container can't overtake the entire system.

Sets soft or hard limits on resources a container can use

Capabilities



Defines granular permissions over the system.

Capabilities can be dropped or added according to
need.

Most capabilities are dropped by default

Containers have a standard set of capabilities



`/var/run/docker.sock`

A special file called a Unix Domain Socket (UDS).

This file gives a container full access to the Docker Daemon on the local host.

It allows control over containers and other docker items on the server.

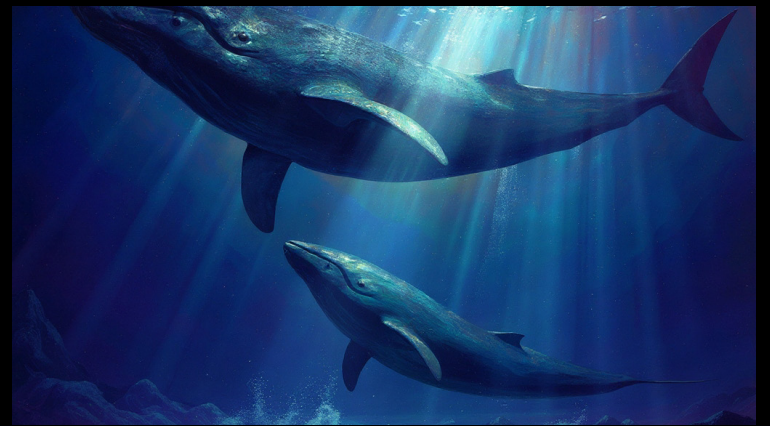
Docker Cheat Sheet

Docker
Build
Reference



docker

Dockerfile Reference



DOCKER FILES

Contain instructions, or directives

Explain how to build an image

Named 'Dockerfile' by default

Custom names are supported

BASIC DIRECTIVES

FROM

Select the base image to start from

RUN

Run a command during the build

Commonly used to install packages, add users, create directories, set file permissions

COPY

Copy files from the source system into the image

ENTRYPOINTS

CMD

A command to run when the container first starts.

Intended to be a default command, which is easily overridden.

May be written in shell form or exec form.

Very similar to CMD

ENTRYPOINT

Not intended to be as easily overridden

Can be combined with CMD



Dockerfile Reference

METADATA

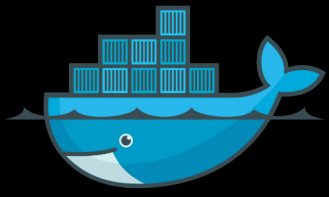
EXPOSE	Provides information about which network ports are used in the container
LABEL	An annotation that describes an image A key-value pair

VARIABLES

ENV	Define environment variables within the container Key-value pair
ARG	A variable used within the docker file Used for the build process only Does not become a variable in the final image

ENVIRONMENT

WORKDIR	Sets the working directory within the container The default location for commands to run from Like a home directory when you use regular linux
USER	Set the active user for the container Commands before this statement are run as root Build commands run after this statement are run as this user



Dockerfile Reference

MULTISTAGE BUILDS

Multistage builds break the Docker file into separate stages. Each performs a separate build.

One stage can handle initial tasks, such as compiling code, processing data, and so on.

A second or final stage can select the relevant parts of the first stage to keep, while the rest is discarded.

The result is an optimised image with fewer packages and files.

IGNORE FILES

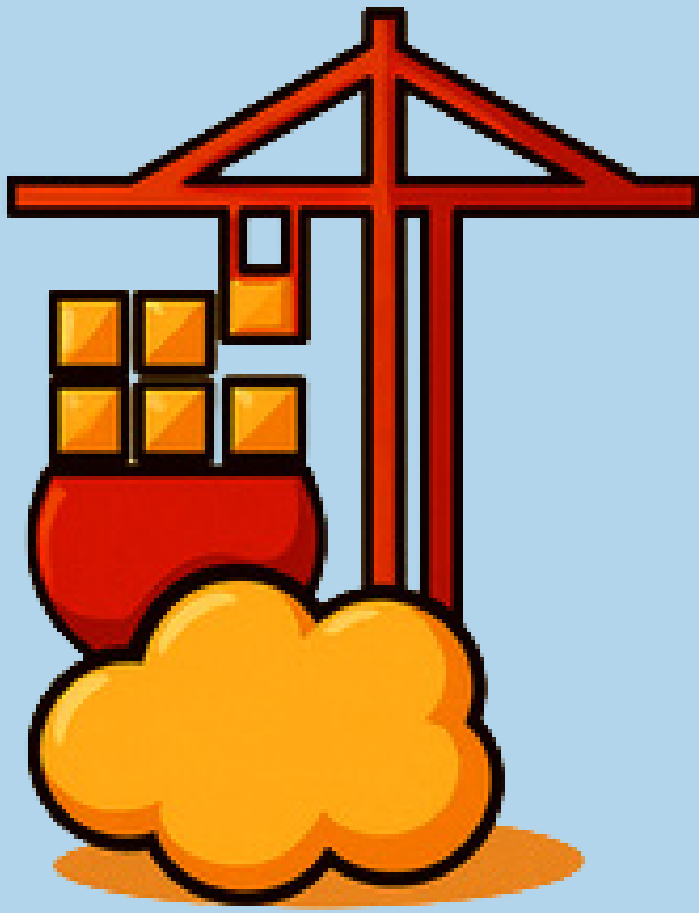
An ignore file can be used to define a list of files or file types that should not be copied into the final image.

These files may include environment variables, secret keys, dependancy files, and others. These may be a security risk, or may just increase the size of the final image.

This is a regular text file named `.dockerignore`

ALWAYS use an ignore file to prevent mistakes.

PORTAINER



```
docker volume create portainer_data
```

```
docker run -d \  
  -p -p 9443:9443 \  
  --name portainer \  
  --restart=always \  
  -v /var/run/docker.sock:/var/run/docker.sock \  
  -v portainer_data:/data \  
  portainer/portainer-ce:latest
```


STUB RESOLVER

Systemd-resolved is the default name resolution service in Ubuntu and other distributions.

This service provides caching, and handles DNS queries for local applications.

To run a different DNS service, the caching stub resolver needs to be disabled.



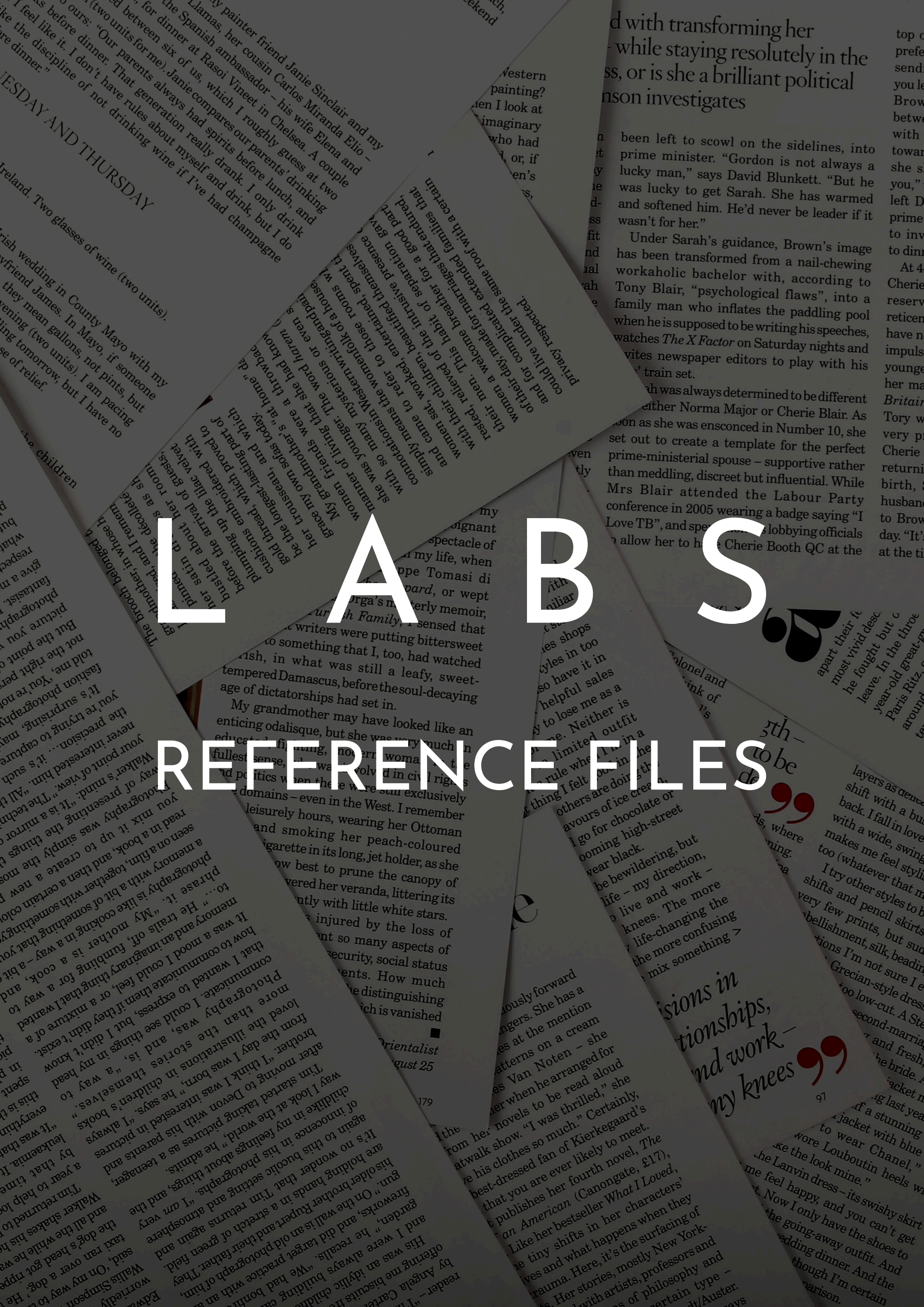
```
sudo sh -c 'mkdir -p /etc/systemd/resolved.conf.d
```

```
printf "[Resolve]\nDNSStubListener=no\n" | tee  
/etc/systemd/resolved.conf.d/no-stub.conf'
```

```
sudo sh -c 'rm -f /etc/resolv.conf && ln -s  
/run/systemd/resolve/resolv.conf  
/etc/resolv.conf'
```


LABS

REFERENCE FILES



LAB NETWORKS

IPVLAN Network

```
docker network create \
  -d ipvlan \
  --subnet=192.168.200.0/24 \
  --gateway=192.168.200.254 \
  -o ipvlan_mode=12 -o parent=eno0 \
  lan-services
```

Monitoring network

```
docker network create -d bridge
monitoring
```

Reverse Proxy network

```
docker network create -d bridge
traefik
```



```
services:
  pihole:
    container_name: pihole
    image: pihole/pihole:latest
    networks:
      - pihole_network
    ports:
      - "53:53/tcp"
      - "53:53/udp"
      - "443:443/tcp"
    environment:
      TZ: 'Australia/Sydney'
      FTLCONF_dns_listeningMode: 'ALL'
      FTLCONF_dns_upstreams: '1.1.1.1;8.8.8.8'
      FTLCONF_dns_domain_name: 'your-domain.com'
      FTLCONF_webserver_api_password: 'Change-this!!!'
    volumes:
      - pihole-data:/etc/pihole
      - pihole-dnsmasq:/etc/dnsmasq.d
    cap_add:
      - CAP_NET_BIND_SERVICE
      - CAP_SYS_NICE
      - CAP_CHOWN
    restart: unless-stopped
```

```
networks:
  pihole_network:
    driver: bridge
```

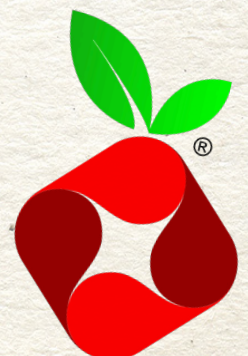
```
volumes:
  pihole-data:
  pihole-dnsmasq:
```




```
services:
  pihole:
    container_name: pihole
    image: pihole/pihole:latest
    networks:
      lan-services:
        ipv4_address: 192.168.200.201
    environment:
      TZ: 'Australia/Sydney'
      FTLCONF_dns_listeningMode: 'ALL'
      FTLCONF_dns_upstreams: '1.1.1.1;8.8.8.8'
      FTLCONF_dns_domain_name: 'your-domain.com'
      FTLCONF_webserver_api_password: 'Change-this!!!'
      FTLCONF_dhcp_active: true
      FTLCONF_dhcp_start: '192.168.200.10'
      FTLCONF_dhcp_end: '192.168.200.100'
      FTLCONF_dhcp_router: '192.168.200.254'
      FTLCONF_dhcp_netmask: '255.255.255.0'
    volumes:
      - pihole-data:/etc/pihole
      - pihole-dnsmasq:/etc/dnsmasq.d
    cap_add:
      - CAP_NET_BIND_SERVICE
      - CAP_SYS_NICE
      - CAP_CHOWN
      - CAP_NET_RAW
      - CAP_NET_ADMIN
    restart: unless-stopped

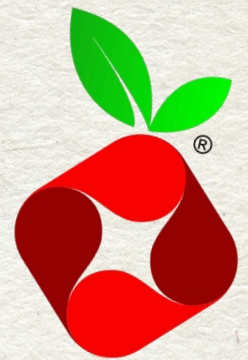
networks:
  lan-services:
    external: true

volumes:
  pihole-data:
  pihole-dnsmasq:
```



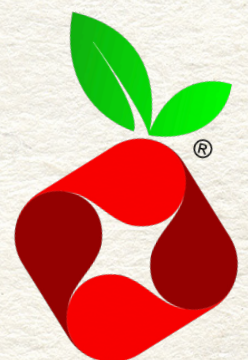

```
services:
  pihole:
    networks:
      monitoring:

networks:
  monitoring:
    external: true
```



Pi-Hole Exporter, Video 14-4

```
services:
  pihole-exporter:
    container_name: pihole-exporter
    image: ekofr/pihole-exporter:latest
    networks:
      - monitoring
    ports:
      - "9617:9617"
    environment:
      PIHOLE_HOSTNAME: 'pihole'
      PIHOLE_PASSWORD: 'API_KEY'
      PORT: 9617
    depends_on: pihole
    restart: unless-stopped
```



Uptime Kuma

Videos 12-3, 14-4

```
services:
  uptime-kuma:
    image: louislam/uptime-kuma:2
    container_name: uptime-kuma
    networks:
      - monitoring
    volumes:
      - uptime-kuma:/app/data
      - /var/run/docker.sock:/var/run/docker.sock:ro
    ports:
      - "3001:3001"
    restart: always
```

```
volumes:
  uptime-kuma:
```

```
networks:
  monitoring:
    external: true
```



Uptime Kuma

Video 15-3
Add Traefik

```
services:
  uptime-kuma:
    image: louislam/uptime-kuma:2
    container_name: uptime-kuma
    networks:
      - monitoring
    volumes:
      - uptime-kuma:/app/data
      - /var/run/docker.sock:/var/run/docker.sock:ro
    ports:
      - "3001:3001"
    labels:
      - "traefik.enable=true"
      - "traefik.http.routers.uptime.rule=Host(`uptime.home.networkdi
rektion.net`)"
      - "traefik.http.routers.uptime.entrypoints=websecure"
      - "traefik.http.routers.uptime.tls=true"
      - "traefik.docker.network=traefik"
      - "traefik.http.routers.uptime.middlewares=secure-headers@file"
      - "traefik.http.services.uptime.loadbalancer.server.port=3001"
    restart: always

volumes:
  uptime-kuma:

networks:
  monitoring:
    external: true
```



Heimdall

Video 13-1

First, create ~/heimdall/config on the Docker server.

```
services:
  heimdall:
    image: lscr.io/linuxserver/heimdall:latest
    container_name: heimdall
    networks:
      - heimdall
      - monitoring
    volumes:
      - /home/dockadmin/heimdall/config:/config
    ports:
      - 80:80
      - 443:443
    restart: unless-stopped

networks:
  heimdall:
    driver: bridge
  monitoring
  external: true
```



Heimdall

Video 15-3
Add Traefik

```
services:
  heimdall:
    image: lscr.io/linuxserver/heimdall:latest
    container_name: heimdall
    networks:
      - traefik
      - monitoring
    volumes:
      - /home/dockadmin/heimdall/config:/config
    labels:
      - "traefik.enable=true"
      -
      "traefik.http.routers.heimdall.rule=Host(`heimdall.home.network
direction.net`)"
      - "traefik.http.routers.heimdall.entrypoints=websecure"
      - "traefik.http.routers.heimdall.tls=true"
      -
      "traefik.http.routers.heimdall.middlewares=secure-headers@file"
    restart: unless-stopped

networks:
  traefik:
    external: true
  monitoring:
    external: true
```




```
services:
  netalertx:
    container_name: netalert
    image: ghcr.io/netalertx/netalertx:latest
    networks:
      lan-services:
        ipv4_address: 192.168.200.202
    monitoring:
    volumes:
      - netalertx_data:/data
      - /etc/localtime:/etc/localtime:ro
    environment:
      LISTEN_ADDR: ${LISTEN_ADDR:-0.0.0.0}
      PORT: ${PORT:-20211}
      GRAPHQL_PORT: ${GRAPHQL_PORT:-20212}
    cap_drop:
      - ALL
    cap_add:
      - NET_ADMIN
      - NET_RAW
      - NET_BIND_SERVICE
      - CHOWN
      - SETUID
      - SETGID
    sysctls:
      net.ipv4.conf.all.arp_ignore: 1
      net.ipv4.conf.all.arp_announce: 2
    read_only: true
    tmpfs:
      -
      /tmp:uid=${NETALERTX_UID:-20211},gid=${NETALERTX_GID:-20211},mode=1700,rw,noexec,nosuid,nodev,async,noatime,nodiratime
    restart: unless-stopped

networks:
  lan-services:
    external: true
  monitoring:
    external: true

volumes:
  netalertx_data:
```



Add the labels shown here to the compose file.

```
services:
  netaalertx:
    labels:
      - "traefik.enable=true"
      -
      "traefik.http.routers.netaalertx.rule=Host(`netaalertx.home.networkdirection.net`)"
      - "traefik.http.routers.netaalertx.entrypoints=websecure"
      - "traefik.http.routers.netaalertx.tls=true"
      - "traefik.docker.network=traefik"
      -
      "traefik.http.routers.netaalertx.middlewares=secure-headers@file,lan-only@file"
      -
      "traefik.http.services.netaalertx.loadbalancer.server.port=20211"
```




```
services:
  db:
    image: mariadb:10.6
    container_name: nextcloud_db
    command: --transaction-isolation=READ-COMMITTED
    --log-bin=binlog --binlog-format=ROW
    networks:
      - nextcloud_internal
    volumes:
      - db:/var/lib/mysql
    environment:
      - MYSQL_ROOT_PASSWORD=<PASSWORD>
      - MYSQL_USER=nextcloud
      - MYSQL_PASSWORD=<PASSWORD>
      - MYSQL_DATABASE=nextcloud
    restart: unless-stopped
```

```
app:
  image: nextcloud
  container_name: nextcloud_app
  networks:
    - nextcloud_internal
  ports:
    - 8080:80
  volumes:
    - nextcloud:/var/www/html
  environment:
    - MYSQL_HOST=db
    - MYSQL_DATABASE=nextcloud
    - MYSQL_USER=nextcloud
    - MYSQL_PASSWORD=<PASSWORD>
    - NEXTCLOUD_TRUSTED_DOMAINS=<DNS DOMAIN>
  depends_on:
    - db
  restart: unless-stopped
```

```
networks:
  nextcloud_internal:
    driver: bridge
```

```
volumes:
  nextcloud:
  db:
```




```
services:
  app:
    networks:
      - traefik
    labels:
      - "traefik.enable=true"
      - "traefik.http.routers.nextcloud.rule=Host(`nextcloud.home.networkdirection.net`)"
      - "traefik.http.routers.nextcloud.entrypoints=websecure"
      - "traefik.http.routers.nextcloud.tls=true"
      - "traefik.docker.network=traefik"
      - "traefik.http.routers.nextcloud.middlewares=secure-headers@file"
      - "traefik.http.services.nextcloud.loadbalancer.server.port=80"

networks:
  traefik:
    external: true
```



After adding Traefik configuration, the allowed domains needs to be updated.

```
docker exec -u www-data \
  nextcloud-app-1 php occ \
  config:system:set trusted_domains 1 \
  --value="nextcloud.home.networkdirection.net"
```


NetBox

Video 14-1

Clone the repository from GitHub

```
git clone -b release  
https://github.com/netbox-community/netbox-docker.git  
cd netbox-docker
```

Copy or rename the example override file

```
cp docker-compose.override.yml.example  
docker-compose.override.yml
```

Edit the override file, and make any needed changes

Start the service

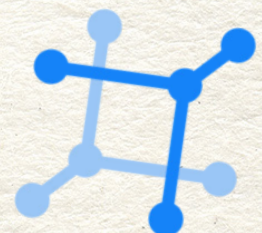
```
docker compose up -d
```

Wait a few minutes for the application to configure itself

Create a super user account

```
docker compose exec netbox  
/opt/netbox/netbox/manage.py createsuperuser
```

```
services:  
  netbox:  
    ports:  
      - "8000:8080"  
    restart: unless-stopped
```



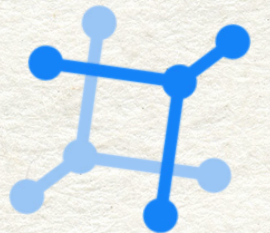
NetBox

Video 15-3
Add Traefik

Update the override file as shown

```
services:
  netbox:
    networks:
      - default
      - traefik
    labels:
      - "traefik.enable=true"
      -
"traefik.http.routers.netbox.rule=Host(`netbox.home.networkdirection.net`)"
  - "traefik.http.routers.netbox.entrypoints=websecure"
  - "traefik.http.routers.netbox.tls=true"
  - "traefik.docker.network=traefik"
  -
"traefik.http.routers.netbox.middlewares=secure-headers@file,lan-only@file"
  - "traefik.http.services.netbox.loadbalancer.server.port=8080"
  restart: unless-stopped

networks:
  traefik:
    external: true
```



Monitoring Stack

Video 14-2

```
services:
  prometheus:
    image: prom/prometheus:latest
    container_name: prometheus
    networks:
      - monitoring
    ports:
      - "9090:9090"
    volumes:
      -
/home/dockadmin/prometheus/prometheus.yml:/etc/prometh
eus/prometheus.yml
    restart: unless-stopped

grafana:
  image: grafana/grafana:latest
  container_name: grafana
  networks:
    - monitoring
  ports:
    - "3000:3000"
  volumes:
    - grafana-storage:/var/lib/grafana
  environment:
    -
GF_SECURITY_ADMIN_PASSWORD=SuperSecretPassword1234$
    depends_on: prometheus
    restart: unless-stopped

networks:
  monitoring:
    external: true

volumes:
  grafana-storage:
```



Monitoring Stack

Video 14-3
Add cAdvisor

```
services:
  prometheus:
    extra_hosts:
      - "host.docker.internal:host-gateway"

  cAdvisor:
    image: gcr.io/cadvisor/cadvisor:latest
    container_name: cadvisor
    networks:
      - monitoring
    ports:
      - "8081:8080"
    volumes:
      - /:/rootfs:ro
      - /var/run:/var/run:rw
      - /sys:/sys:ro
      - /var/lib/docker/containers:/var/lib/docker:ro
      - /dev/disk/":"/dev/disk:ro
    devices:
      - /dev/kmsg
    restart: unless-stopped
```



Edit `/etc/docker/daemon.json` and enable metrics.

```
{
  "metrics-addr": "0.0.0.0:9323"
}
```


Monitoring Stack

Video 15-3
Add Traefik

```
services:
  prometheus:
    networks:
      - traefik
    labels:
      - "traefik.enable=true"
      -
"traefik.http.routers.prometheus.rule=Host(`prometheus.home.networkdirection.net`)"
  - "traefik.http.routers.prometheus.entrypoints=websecure"
  - "traefik.http.routers.prometheus.tls=true"
  - "traefik.docker.network=traefik"
  -
"traefik.http.routers.prometheus.middlewares=secure-headers@file,basic-auth@file,lan-only@file"
  -
"traefik.http.services.prometheus.loadbalancer.server.port=9090"

grafana:
  networks:
    - traefik
  labels:
    - "traefik.enable=true"
    -
"traefik.http.routers.grafana.rule=Host(`grafana.home.networkdirection.net`)"
  - "traefik.http.routers.grafana.entrypoints=websecure"
  - "traefik.http.routers.grafana.tls=true"
  - "traefik.docker.network=traefik"
  -
"traefik.http.routers.grafana.middlewares=secure-headers@file,lan-only@file"
  -
"traefik.http.services.grafana.loadbalancer.server.port=3000"

networks:
  traefik:
    external: true
```



```
global:
  scrape_interval: 60s

scrape_configs:
- job_name: 'prometheus'
  static_configs:
    - targets: ['localhost:9090']

- job_name: 'docker'
  static_configs:
    targets: ['host.docker.internal:9323']

- job_name: 'cadvisor'
  static_configs:
    targets: ['cadvisor:8081']

- job_name: 'uptime'
  static_configs:
    - targets: ['uptime-kuma:3001']
  basic_auth:
    password: 'API_KEY'

- job_name: 'netalertx'
  static_configs:
    - targets: ['netalert:20212']
  authorization:
    type: Bearer
    credentials: 'API_KEY'

- job_name: 'pihole'
  static_configs:
    - targets: ['pihole-exporter:9617']

- job_name: 'traefik'
  static_configs:
    - targets: ['traefik:8080']
```



Create a directory for certificates

```
mkdir -p traefik/certs
```

Install OpenSSL

```
sudo apt install openssl
```

Generate a self-signed certificate

```
openssl req -x509 -nodes -days 365 -newkey rsa:2048  
-keyout certs/local.key -out certs/local.crt -subj  
"/CN=*.<domain-name>" -addext  
"subjectAltName=DNS:<domain-name>,DNS:*.<domain-name>"
```

Create a config file for TLS

```
mkdir -p traefik/config  
sudo nano traefik/config/tls.yaml
```

Create a config for middleware

```
sudo nano traefik/config/middlewares.yaml
```

Generate a hashed password

```
sudo apt install apache2-utils  
htpasswd -nb admin "<Your-Password>"
```

Generate a hashed password (Compose format)

```
htpasswd -nb admin "SuperSecretPassword1234$" | sed  
-e 's/\\$/\\$\\$/g'
```


traefik/config/tls.yaml

```
tls:
  certificates:
    - certFile: /certs/local.crt
      keyFile:  /certs/local.key

  options:
    default:
      minVersion: VersionTLS12
      curvePreferences:
        - X25519MLKEM768
        - X25519
        - CurveP256
        - CurveP384
      cipherSuites:
        - TLS_ECDHE_ECDSA_WITH_AES_128_GCM_SHA256
        - TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256
        - TLS_ECDHE_ECDSA_WITH_AES_256_GCM_SHA384
        - TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384
        - TLS_ECDHE_ECDSA_WITH_CHACHA20_POLY1305_SHA256
        - TLS_ECDHE_RSA_WITH_CHACHA20_POLY1305_SHA256
      sniStrict: true
```



traefik/config/middlewares.yaml

```
http:
  middlewares:
    secure-headers:
      headers:
        frameDeny: true
        sslRedirect: true
        contentTypeNosniff: true
        stsIncludeSubdomains: true
        stsPreload: true
        stsSeconds: 31536000

    basic-auth:
      basicAuth:
        users:
          - "admin:<HASH>"

    lan-only:
      ipAllowList:
        sourceRange:
          - 192.168.200.0/24
          - 10.8.0.0/24
```



Basic traefik service

```
services:
  traefik:
    image: traefik:latest
    container_name: traefik
    networks:
      - traefik
    ports:
      - "80:80"
      - "443:443"
    volumes:
      - /var/run/docker.sock:/var/run/docker.sock:ro
      - /home/dockadmin/traefik/certs:/certs:ro
      - /home/dockadmin/traefik/config:/dynamic:ro
    security_opt:
      - no-new-privileges:true
    restart: unless-stopped
```

```
networks:
  traefik:
    external: true
```



Entrypoint configuration

```
services:
  traefik:
    command:
      # Entrypoints
      - "--entrypoints.web.address=:80"
      - "--entrypoints.websecure.address=:443"
      - "--entrypoints.websecure.http.tls=true"

      # HTTPS Redirection
      -
      "--entrypoints.web.http.redirections.entrypoint.to=websecure"
      -
      "--entrypoints.web.http.redirections.entrypoint.scheme=https"
      -
      "--entrypoints.web.http.redirections.entrypoint.permanent=true"

      # Configure 'providers'
      - "--providers.docker=true"
      - "--providers.docker.exposedbydefault=false"
      - "--providers.docker.network=traefik"

      # Add dynamic config files
      - "--providers.file.directory=/dynamic"
      - "--providers.file.watch=true"

      # Enable the API
      - "--api.dashboard=true"
      - "--api.insecure=false"

      # Logging and metrics
      - "--log.level=INFO"
      - "--accesslog=true"
      - "--metrics.prometheus=true"
```



Labels

```
services:
  traefik:
    labels:
      # Enable traefik on this web service (dashboard)
      - "traefik.enable=true"

      # Routing rules
      -
"traefik.http.routers.dashboard.rule=Host(`dashboard.home.net
workdirection.net`)"
      -
"traefik.http.routers.dashboard.entrypoints=websecure"
      - "traefik.http.routers.dashboard.tls=true"
      - "traefik.http.routers.dashboard.service=api@internal"

      # Authentication middleware
      -
"traefik.http.routers.dashboard.middlewares=basic-auth@file"
```



Whoami service

```
services:
  whoami:
    image: traefik/whoami
    container_name: whoami
    networks:
      - traefik
    labels:
      - "traefik.enable=true"
      - "traefik.http.routers.whoami.rule=Host(`whoami.home.networkdi
rektion.net`)"
      - "traefik.http.routers.whoami.entrypoints=websecure"
      - "traefik.http.routers.whoami.tls=true"
      - "traefik.http.routers.whoami.middlewares=secure-headers@file"
```




```
services:
  vaultwarden:
    image: vaultwarden/server:latest
    container_name: vaultwarden
    networks:
      - traefik
    volumes:
      - vaultwarden-data:/data
    environment:
      - DOMAIN=https://vault.home.networkdirection.net
      - SIGNUPS_ALLOWED=true

    labels:
      - "traefik.enable=true"
      -
      "traefik.http.routers.vaultwarden.rule=Host(`vault.home.networkdirection.net`)"
      -
      "traefik.http.routers.vaultwarden.entrypoints=websecure"
      - "traefik.http.routers.vaultwarden.tls=true"
      - "traefik.docker.network=traefik"
      -
      "traefik.http.routers.vaultwarden.middlewares=secure-headers@file"

    restart: unless-stopped

networks:
  traefik:
    external: true

volumes:
  vaultwarden-data:
```



WireGuard

Video 15-5

```
services:
  wg-easy:
    image: ghcr.io/wg-easy/wg-easy:15
    container_name: wg-easy
    networks:
      - traefik
      - monitoring
    ports:
      - "51820:51820/udp"
    volumes:
      - etc_wireguard:/etc/wireguard
      - /lib/modules:/lib/modules:ro
    cap_add:
      - NET_ADMIN
      - SYS_MODULE
    sysctls:
      - net.ipv4.ip_forward=1
      - net.ipv4.conf.all.src_valid_mark=1
    environment:
      - TZ=Australia/Sydney
      - WG_HOST=vpn.networkdirection.net
    labels:
      - "traefik.enable=true"
      - "traefik.http.routers.vpn.rule=Host(`vpn.home.networkdirection.net`)"
      - "traefik.http.routers.vpn.entrypoints=websecure"
      - "traefik.http.routers.vpn.tls=true"
      - "traefik.docker.network=traefik"
      - "traefik.http.routers.vpn.middlewares=secure-headers@file,lan-only@file"
      - "traefik.http.services.vpn.loadbalancer.server.port=51821"
    restart: unless-stopped

networks:
  traefik:
    external: true
  monitoring:
    external: true

volumes:
  etc_wireguard:
```



Grafana

▶ > > >			
+	Docker Containers		21743
	Docker Server		15798
	NetAlertX		N/A
	+	Pi-Hole	14283
		Traefik	17346
	Uptime Kuma		18667
	WireGuard		21733

Dashboards

LINKS



...my painter friend Janie Sinclair and my ...
...the Spanish ambassador - his wife Elena and ...
...for dinner at Rasoi Vineet in Chelsea. A couple ...
...between six of us, which I roughly guess at two ...
...Janie compares our parents' drinking ...
...That generation always had spirits before lunch, and ...
...I don't have rules about myself and drink, but I do ...
...of not drinking wine if I've had champagne ...
...TUESDAY AND THURSDAY ...
...Two glasses of wine (two units). ...
...wedding in County Mayo with my ...
...In Mayo, if someone ...
...mean gallons, not pints, but ...
...I am pacing ...
...but I have no ...
...children ...
...my ...
...spectacle of ...
...my life, when ...
...Tomasi di ...
...the Leopard, or wept ...
...Orga's masterly memoir, ...
...sensed that ...
...writers were putting ...
...something that I, too, ...
...had ...
...in what was still a ...
...swe ...
...tempered Damascus, before the ...
...decaying ...
...age of dictatorships had set in. ...
...My grandmother may have looked like an ...
...enticing odalisque, but she was very much an ...
...educated, fighting, modern woman in the ...
...fullest sense, who was involved in civil rights ...
...and politics when these were still exclusively ...
...domains - even in the West. I remember ...
...leisurely hours, wearing her Ottoman ...
...and smoking her peach-coloured ...
...cigarette in its long, jet holder, as she ...
...now best to prune the canopy of ...
...ver her veranda, littering its ...
...with little white stars. ...
...injured by the loss of ...
...so many aspects of ...
...security, social status ...
...elements. How much ...
...the distinguishing ...
...which is vanished ...
...Orientalist ...
...August 25 ...
...179 ...
...the ...
...from her ...
...atwalk show. "I was thrilled," she ...
...his clothes so much. "Certainly, ...
...best-dressed fan of Kierkegaard's ...
...that you are ever likely to meet. ...
...publishes her fourth novel, *The ...
...an American* (Canongate, £17). ...
...Like her bestseller *What I Loved*, ...
...the tiny shifts in her characters' ...
...ives and what happens when they ...
...trauma. Here, it's the surfacing of ...
...with artists, professors and ...
...of philosophy and ...
...certain type - ...
...at/Auster. ...
...says ...
...top of ...
...prefere ...
...sendi ...
...you le ...
...Brow ...
...betwe ...
...with ...
...toward ...
...she s ...
...you," ...
...left D ...
...prim ...
...to in ...
...to dim ...
...At 4 ...
...Cherie ...
...reserv ...
...reticen ...
...have n ...
...impuls ...
...younge ...
...her ma ...
...*Britain* ...
...Tory w ...
...very p ...
...Cherie ...
...returni ...
...birth, ...
...husban ...
...to Brow ...
...day. "It' ...
...at the ti ...
...d with transforming her ...
...while staying resolutely in the ...
...ss, or is she a brilliant political ...
...nson investigates ...
...been left to scowl on the sidelines, into ...
...prime minister. "Gordon is not always a ...
...lucky man," says David Blunkett. "But he ...
...was lucky to get Sarah. She has warmed ...
...and softened him. He'd never be leader if it ...
...wasn't for her." ...
...Under Sarah's guidance, Brown's image ...
...has been transformed from a nail-chewing ...
...workaholic bachelor with, according to ...
...Tony Blair, "psychological flaws", into a ...
...family man who inflates the paddling pool ...
...when he is supposed to be writing his speeches, ...
...watches *The X Factor* on Saturday nights and ...
...vites newspaper editors to play with his ...
...train set. ...
...ah was always determined to be different ...
...either Norma Major or Cherie Blair. As ...
...oon as she was ensconced in Number 10, she ...
...set out to create a template for the perfect ...
...prime-ministerial spouse - supportive rather ...
...than meddling, discreet but influential. While ...
...Mrs Blair attended the Labour Party ...
...conference in 2005 wearing a badge saying "I ...
...Love TB", and spent months lobbying officials ...
...to allow her to have Cherie Booth QC at the ...
...3th - ...
...to be ...
...de ...
...layers as ...
...shift with a bu ...
...back. I fall in love ...
...with a wide, swing ...
...makes me feel styli ...
...too (whatever that m ...
...I try other styles to b ...
...shifts and pencil skirts ...
...very few prints, but sud ...
...bellishment, silk, beadin ...
...tions I'm not sure I e ...
...Grecian-style dress ...
...too low-cut. A Ste ...
...second-marria ...
...and fresh, ...
...the bride. A ...
...sacket n ...
...ag last year ...
...off a stunning ...
...to wear Chanel," ...
...wore Louboutin heels w ...
...ke the look mine." ...
...he Lanvin dress - its swishy skirt ...
...me feel happy, and you can't get ...
...the going-away outfit. And the ...
...wedding dinner. And the ...
...though I'm certain ...
...parison. ...

DOCKER DOCUMENTATION

Docker Desktop

<https://docs.docker.com/get-started/introduction/get-docker-desktop/>

Docker Engine

<https://docs.docker.com/engine/install/>

Post Install Guide

<https://docs.docker.com/engine/install/linux-postinstall/>

Docker Hardened Images

<https://www.docker.com/products/hardened-images/>

User Namespaces

<https://docs.docker.com/engine/security/userns-remap/>

Rootless Mode

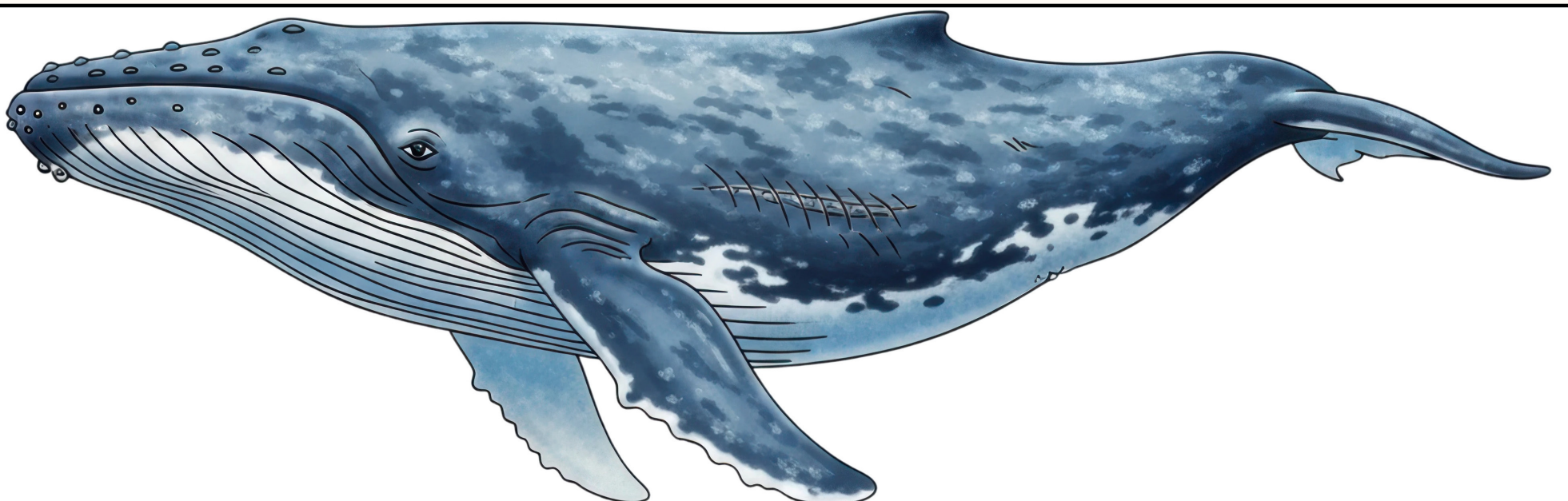
<https://docs.docker.com/engine/security/rootless/>

Secure Transport

<https://docs.docker.com/engine/security/protect-access/>

Docker Content Trust

<https://www.docker.com/blog/retiring-docker-content-trust/>



UBUNTU GUIDES

Ubuntu Server (Download)

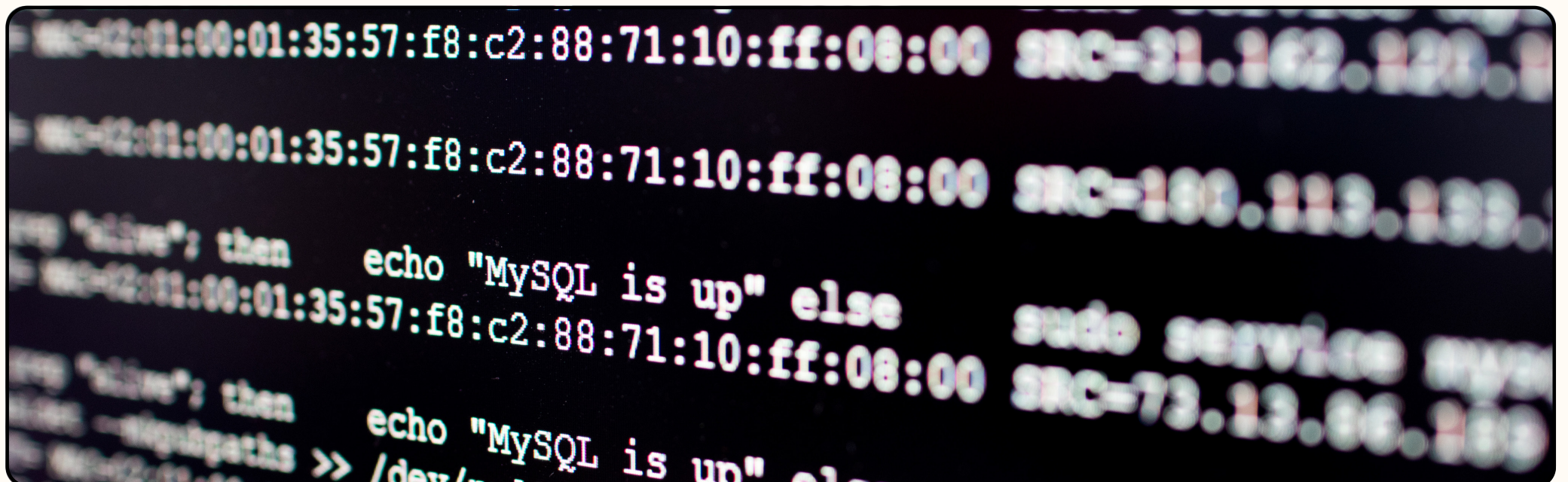
<https://ubuntu.com/download/server>

Ubuntu Install Guide

<https://ubuntu.com/tutorials/install-ubuntu-server#3-boot-from-install-media>

Docker Engine on Ubuntu

<https://docs.docker.com/engine/install/ubuntu/>



GENERAL DOCUMENTATION

OpenContainers Annotations

<https://docs.docker.com/get-started/introduction/get-docker-desktop/>

Linux Capabilities

<https://docs.docker.com/engine/install/>

CIS Docker Benchmark

<https://docs.docker.com/engine/install/linux-postinstall/>

Ciphers

<https://www.docker.com/products/hardened-images/>

Mozilla Cipher List

<https://docs.docker.com/engine/security/usersns-remap/>



PIHOLE GUIDES

Pi-Hole on docker

<https://docs.pi-hole.net/docker/>

Configuration

<https://docs.pi-hole.net/docker/configuration/>

DHCP

<https://docs.pi-hole.net/docker/DHCP/>

Tips and Tricks

<https://docs.pi-hole.net/docker/tips-and-tricks/>

Pi-Hole Exporter

<https://github.com/eko/pihole-exporter>



PROMETHEUS AND GRAFANA

Install on Docker

<https://signoz.io/guides/how-to-install-prometheus-and-grafana-on-docker/>

Configure Docker Metrics

<https://docs.docker.com/engine/daemon/prometheus/>

Grafana Dashboards

<https://grafana.com/grafana/dashboards/>



APPLICATIONS

Portainer Features

<https://www.portainer.io/features>

Traefik, Docker install

<https://doc.traefik.io/traefik/setup/docker/>

Traefik (Docker Hub)

<https://hub.docker.com/u/traefik>

Uptime Kuma

<https://github.com/louislam/uptime-kuma>

Heimdall

<https://hub.docker.com/r/linuxserver/heimdall>

NetAlertX

https://docs.netalertx.com/DOCKER_INSTALLATION/

NetAlertX Grafana Dashboard

https://docs.netalertx.com/API_METRICS/#grafana-dashboard-template



APPLICATIONS

Snyk

<https://snyk.io/partners/docker/>

NextCloud

<https://nextcloud.com/>

NetBox

<https://netboxlabs.com/>

NetBox Installation

<https://netboxlabs.com/docs/netbox/installation/>

NetBox on Docker

<https://github.com/netbox-community/netbox-docker>

VaultWarden

<https://github.com/dani-garcia/vaultwarden>

WireGuard, WG-Easy

<https://github.com/wg-easy/wg-easy>



USEFUL TOOLS

Hardened Images Catalog

<https://hub.docker.com/hardened-images/catalog>

Docker Bench

<https://github.com/docker/docker-bench-security>

Docker Bench (Docker Hub)

<https://hub.docker.com/r/docker/docker-bench-security>

Fail2Ban

<https://github.com/fail2ban/fail2ban>



DOCKER COMMANDS

General commands

<https://docs.docker.com/reference/cli/docker/>

Containers

<https://docs.docker.com/reference/cli/docker/container/>

Exec

<https://docs.docker.com/reference/cli/docker/container/exec/>

Images

<https://docs.docker.com/reference/cli/docker/image/>

Inspect

<https://docs.docker.com/reference/cli/docker/inspect/>

Network

<https://docs.docker.com/reference/cli/docker/network/>

Scout

<https://docs.docker.com/reference/cli/docker/scout/>

Volumes

<https://docs.docker.com/reference/cli/docker/volume/>



BUILD AND COMPOSE

Build Commands

<https://docs.docker.com/reference/cli/docker/buildx/build/>

Docker files

<https://docs.docker.com/reference/dockerfile/>

YAML Validation

<https://www.yamllint.com/>

Awesome Compose

<https://github.com/docker/awesome-compose>

Compose commands

<https://docs.docker.com/reference/cli/docker/compose/>

Services

<https://docs.docker.com/reference/compose-file/services>

Networks

<https://docs.docker.com/reference/compose-file/networks/>

Volumes

<https://docs.docker.com/reference/compose-file/volumes/>

