



Адміністрування комп'ютерних систем

Модуль 2. Адміністрування ОС Linux Server

Лекція 6. Основи Docker

Кудін Олексій Володимирович
avk256@gmail.com



План

- 1) Основні поняття технологій контейнеризації
- 2) Структура Docker
- 3) Інсталяція Docker та Docker compose

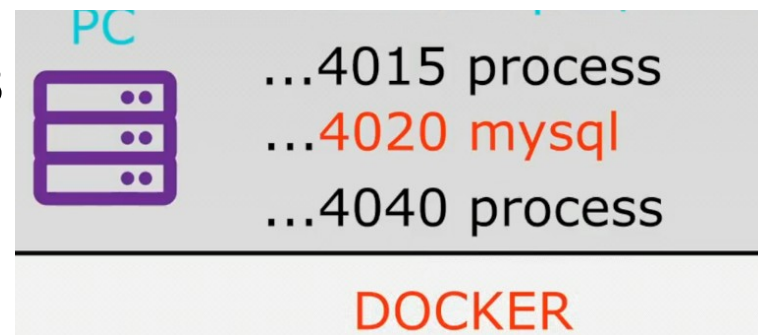
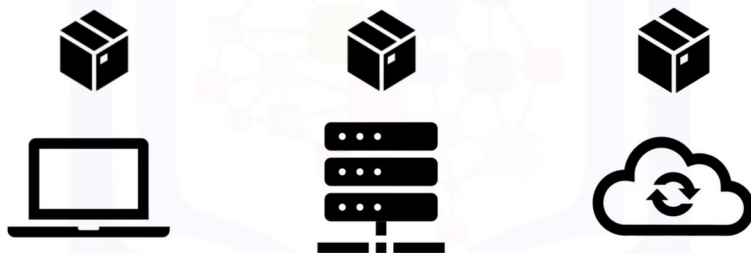
Що таке контейнер?

1) Модуль програмного забезпечення

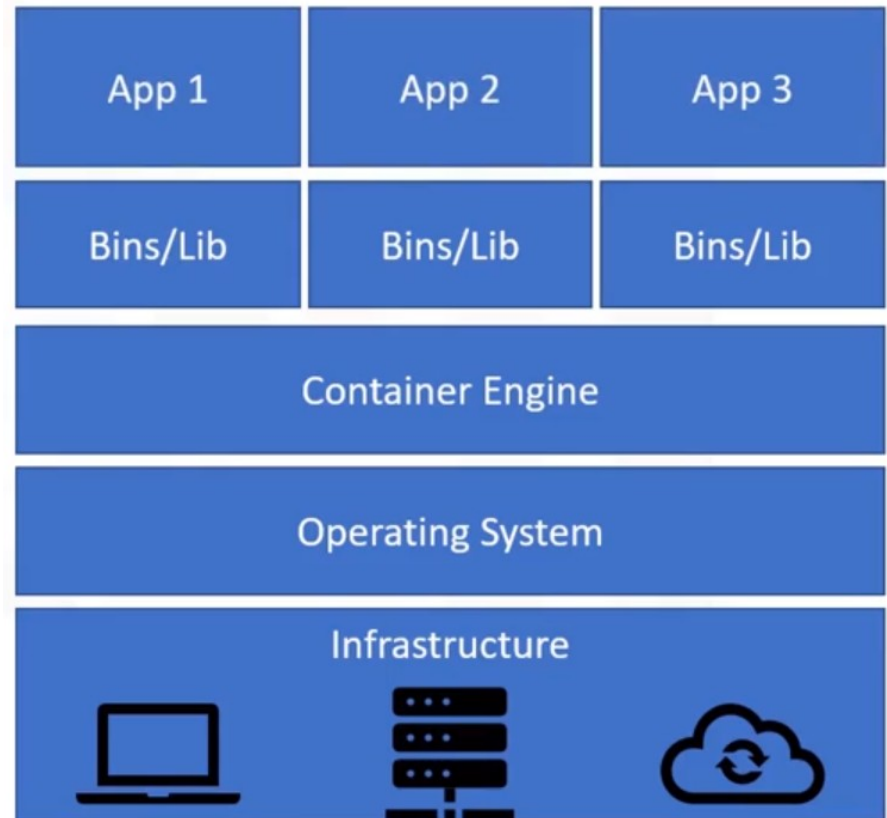
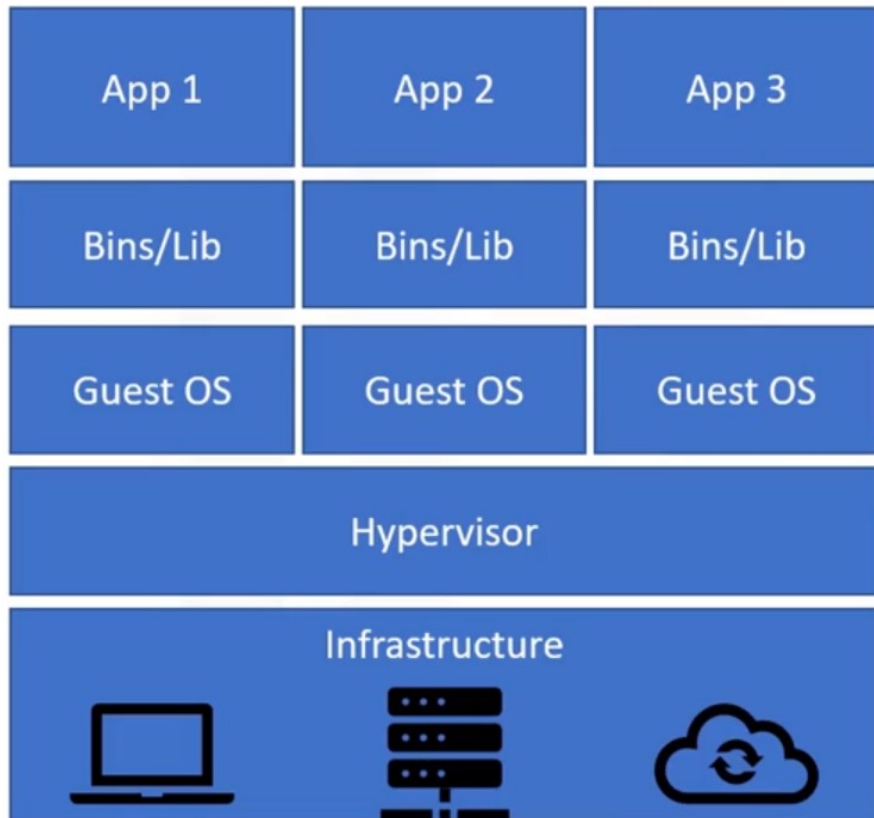
- Включає все необхідне
- Працює в довільному оточенні

2) Віртулізація на рівні ОС

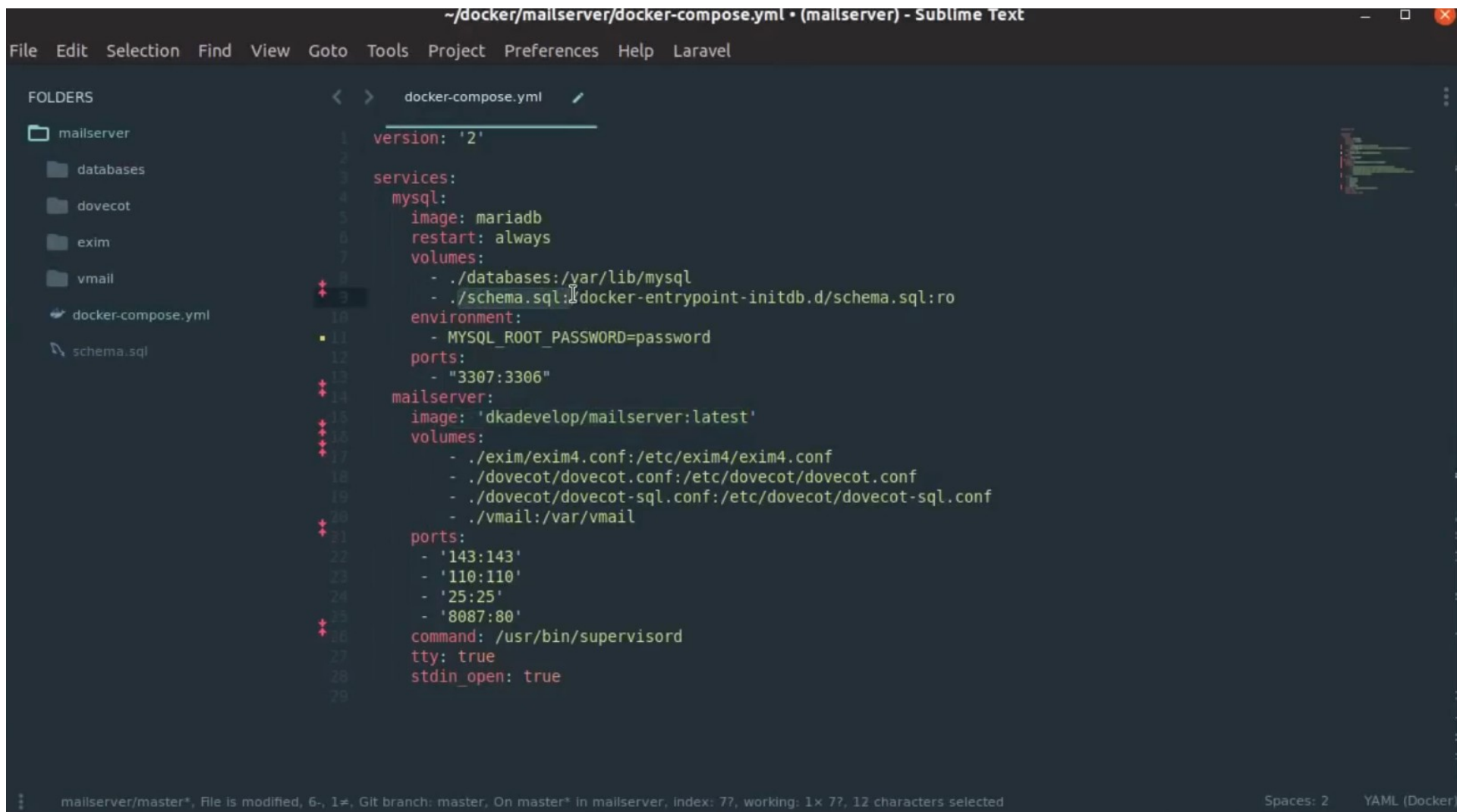
- Ізольовані процеси
- Контролює ресурси процесів



Контейнери та віртуальні машини



Приклад роботи: конфігурація контейнера



The screenshot displays the Sublime Text editor with a Docker Compose configuration file named `docker-compose.yml` open. The file is located at `~/docker/mailserver/docker-compose.yml`. The editor's sidebar on the left shows the project structure, including folders for `databases`, `dovecot`, `exim`, and `vmail`, along with files `docker-compose.yml` and `schema.sql`.

The configuration file content is as follows:

```
1 version: '2'
2
3 services:
4   mysql:
5     image: mariadb
6     restart: always
7     volumes:
8       - ../databases:/var/lib/mysql
9       - ../schema.sql:/docker-entrypoint-initdb.d/schema.sql:ro
10    environment:
11      - MYSQL_ROOT_PASSWORD=password
12    ports:
13      - "3307:3306"
14  mailserver:
15    image: 'dkadevelop/mailserver:latest'
16    volumes:
17      - ../exim/exim4.conf:/etc/exim4/exim4.conf
18      - ../dovecot/dovecot.conf:/etc/dovecot/dovecot.conf
19      - ../dovecot/dovecot-sql.conf:/etc/dovecot/dovecot-sql.conf
20      - ../vmail:/var/vmail
21    ports:
22      - '143:143'
23      - '110:110'
24      - '25:25'
25      - '8087:80'
26    command: /usr/bin/supervisord
27    tty: true
28    stdin_open: true
29
```

The status bar at the bottom indicates the current file is `mailserver/master*`, it has been modified 6 times, and the Git branch is `master`. It also shows the current index is 77, with 12 characters selected. The editor is set to 2 spaces and is editing a YAML (Docker) file.

Приклад роботи: docker compose

```
$ docker
Usage: docker [OPTIONS] COMMAND [ARG...]
       docker [ --help | -v | --version ]

A self-sufficient runtime for containers.

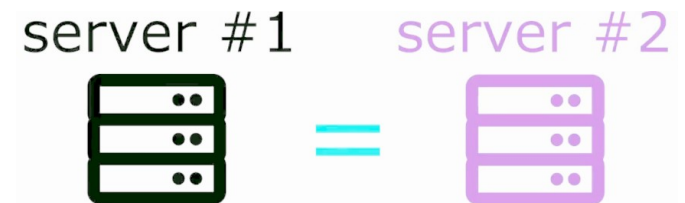
Options:
  --config string      Location of client config files (default "/root/.docker")
  -c, --context string  Name of the context to use to connect to the daemon (overrides DOCKER_HOST env var
  -D, --debug           Enable debug mode
  --help               Print usage
  -H, --host value      Daemon socket(s) to connect to (default [])
  -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

Commands:
  attach      Attach to a running container
  # [...]
  # [...]
  # [...]
```

```
$ docker-compose up
WARNING: Found orphan containers (mailserver_adminer_1) for this project. If you removed or renamed this service in your compose
se file, you can run this command with the --remove-orphans flag to clean it up.
Starting mailserver_mailserver_1 ...
Starting mailserver_mysql_1 ...
Starting mailserver_mailserver_1
Starting mailserver_mysql_1
```

Структура Docker складається:

- 1) Образи
- 2) Реєстри
- 3) Контейнери

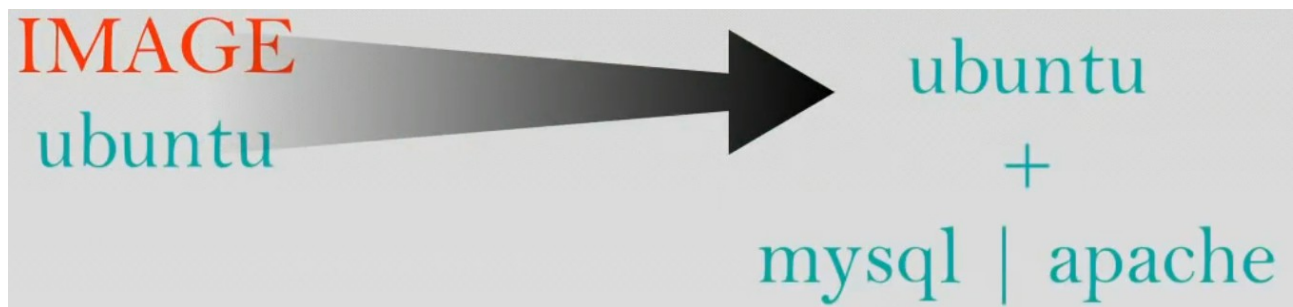


Docker образ

Read only шаблон

Наприклад, може містити:

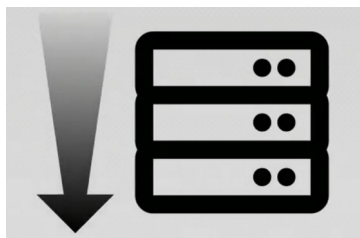
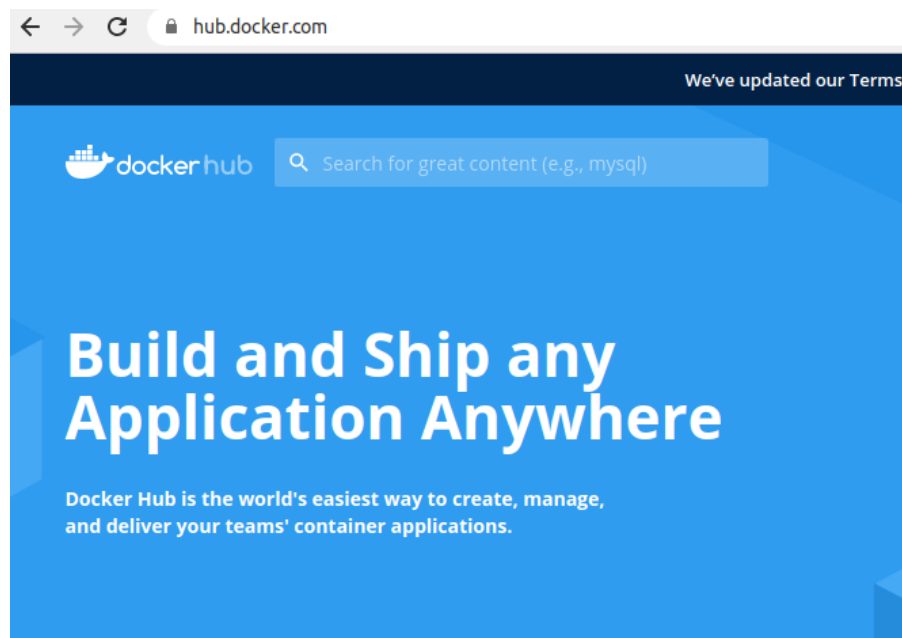
- Ubuntu
- Apache
- MySql
- App



Docker реєстр

Місце, де зберігаються образи

Наприклад, Docker-hub:



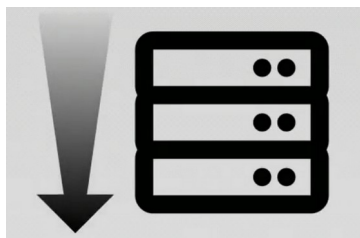
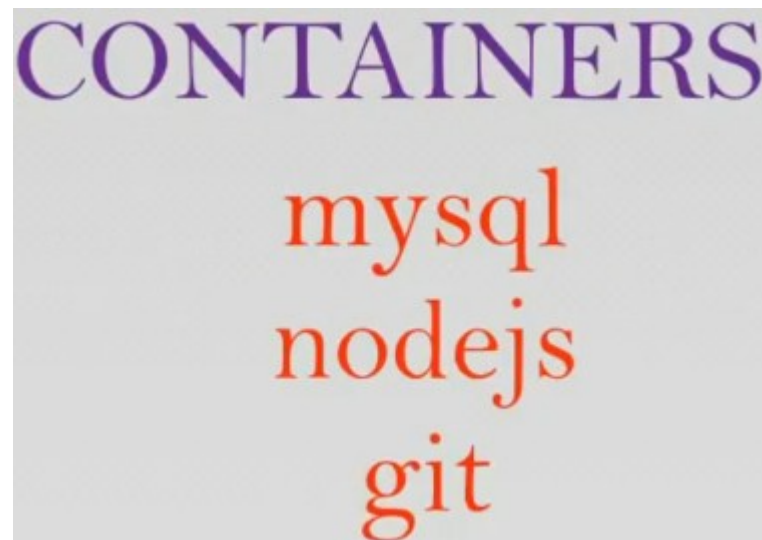
- hub.docker.com



Docker контейнери

Містять необхідне програмне середовище

Наприклад, Docker-hub:



• hub.docker.com



Інсталяція Docker



The screenshot shows the Docker documentation website at `docs.docker.com/get-docker/`. The page has a blue header with the Docker logo, a search bar, and navigation links: Home, Guides, Product manuals, Reference, and Samples. A left sidebar contains a list of links: Docker overview, Get Docker (highlighted), Get started, Develop with Docker, Deploy your app to the cloud, Run your app in production, Open source at Docker, and Documentation archive. The main content area is titled 'Get Docker' and contains the following text:

Docker is an open platform for developing, shipping, and running applications. Docker enables you to separate your applications from your infrastructure so you can deliver software quickly. With Docker, you can manage your infrastructure in the same ways you manage your applications. By taking advantage of Docker's methodologies for shipping, testing, and deploying code quickly, you can significantly reduce the delay between writing code and running it in production.

You can download and install Docker on multiple platforms. Refer to the following section and choose the best installation path for you.

Below this text are three installation options, each with an icon, a title, and a description:

- Docker Desktop for Mac**: A native application using the macOS sandbox security model which delivers all Docker tools to your Mac.
- Docker Desktop for Windows**: A native Windows application which delivers all Docker tools to your Windows computer.
- Docker for Linux**: Install Docker on a computer which already has a Linux distribution installed.

Інсталяція Docker

```
Hello from Docker!
```

```
This message shows that your installation appears to be working correctly.
```

```
To generate this message, Docker took the following steps:
```

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

```
To try something more ambitious, you can run an Ubuntu container with:
```

```
$ docker run -it ubuntu bash
```

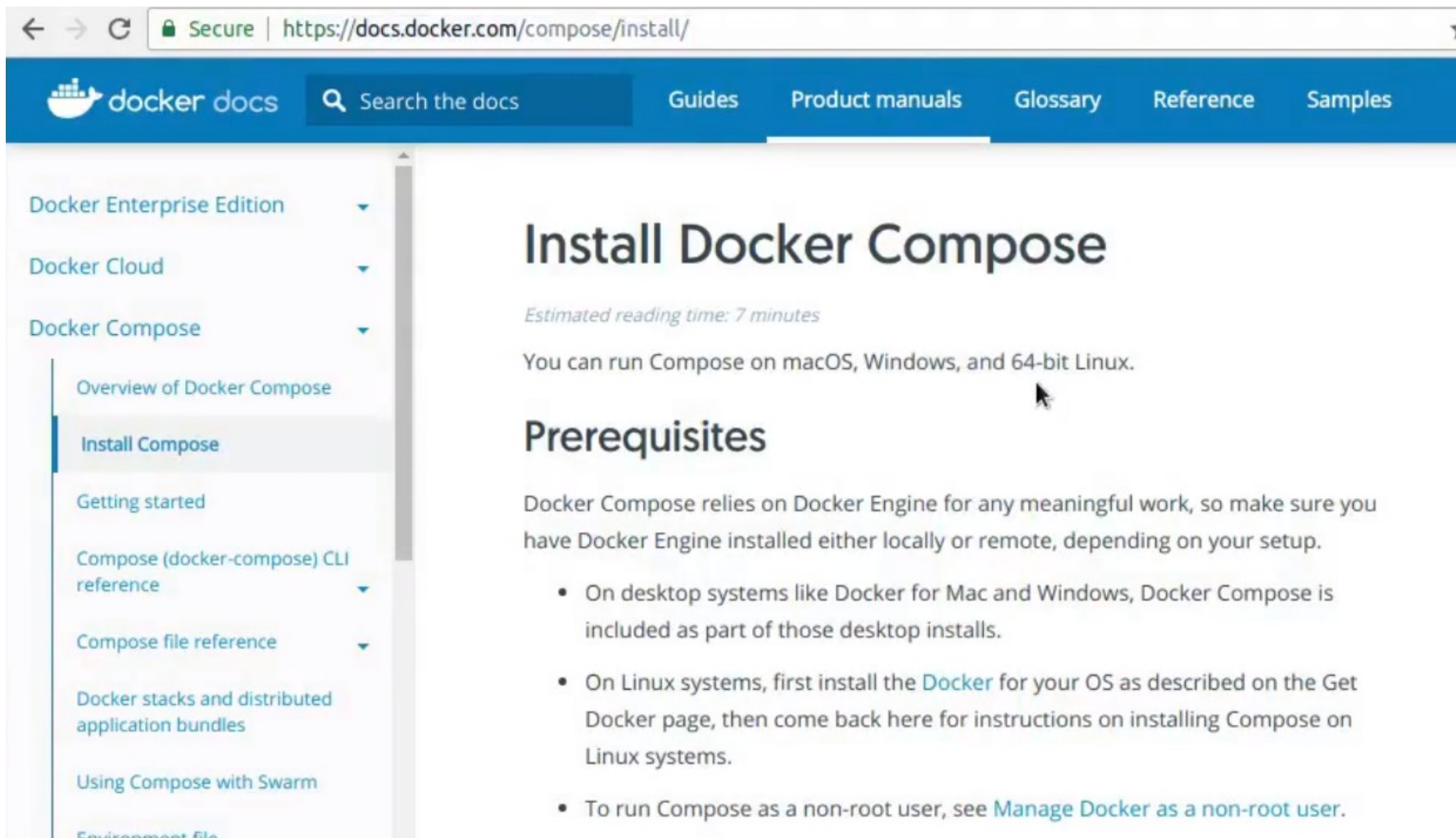
```
Share images, automate workflows, and more with a free Docker ID:
```

```
https://cloud.docker.com/
```

```
For more examples and ideas, visit:
```

```
https://docs.docker.com/engine/userguide/
```

Інсталяція Docker



The screenshot shows a web browser window with the URL <https://docs.docker.com/compose/install/>. The page title is "Install Docker Compose" with an estimated reading time of 7 minutes. The left sidebar shows the navigation menu with "Docker Compose" expanded, and "Install Compose" selected. The main content area includes a paragraph stating that Compose can run on macOS, Windows, and 64-bit Linux, followed by a "Prerequisites" section. This section explains that Docker Compose relies on Docker Engine and provides a list of requirements for different operating systems.

← → ↻ Secure | <https://docs.docker.com/compose/install/>

 docker docs [Guides](#) [Product manuals](#) [Glossary](#) [Reference](#) [Samples](#)

Docker Enterprise Edition ▾

Docker Cloud ▾

Docker Compose ▾

- Overview of Docker Compose
- Install Compose**
- Getting started
- Compose (docker-compose) CLI reference ▾
- Compose file reference ▾
- Docker stacks and distributed application bundles
- Using Compose with Swarm
- Environment file

Install Docker Compose

Estimated reading time: 7 minutes

You can run Compose on macOS, Windows, and 64-bit Linux.

Prerequisites

Docker Compose relies on Docker Engine for any meaningful work, so make sure you have Docker Engine installed either locally or remote, depending on your setup.

- On desktop systems like Docker for Mac and Windows, Docker Compose is included as part of those desktop installs.
- On Linux systems, first install the [Docker](#) for your OS as described on the Get Docker page, then come back here for instructions on installing Compose on Linux systems.
- To run Compose as a non-root user, see [Manage Docker as a non-root user](#).