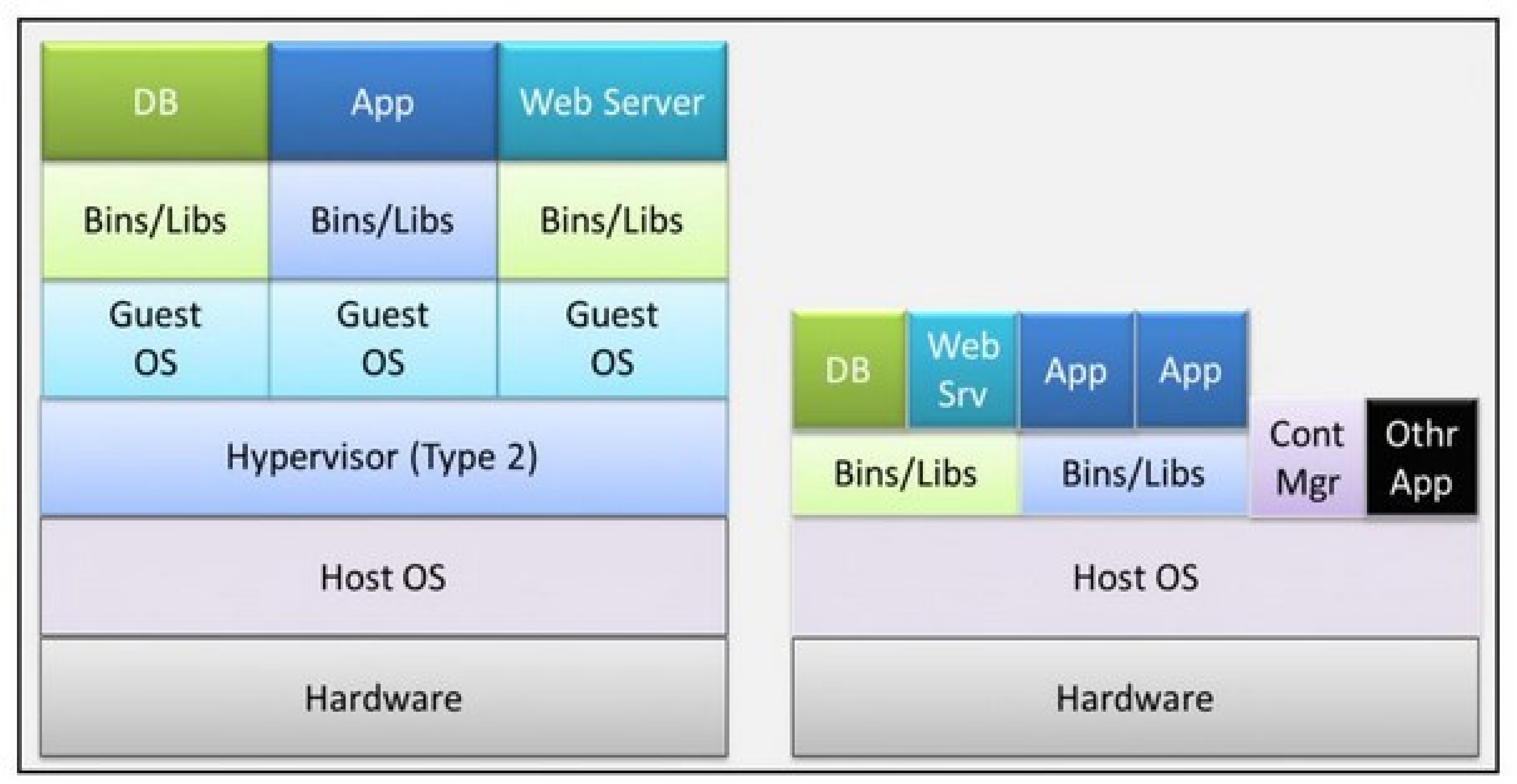
Amrita Vishwa Vidyapeetham

Amritapuri Campus

22AIE305: CLOUD COMPUTING

Virtualisation vs Containerisation



Run Containers in background

Containers can run in the background without being attached to any input or output. They can also be interactive.

Docker run –detach \

name xxx sftname:latest

This will bring the latest version of sftname from Dockerhub

Both - - detach and – d are the same

Docker run - - interactive - - tty (same as -- i and -- t)

This will allow multiple images to be executed simultaneously

The npm command is used to setup dependencies

Each container must be given a label to distinguish it from others.

The label can be any string including your email address.

LABEL **container_name** student@amrita.edu

We can rename the container docker rename oldname newname

It is convenient to keep all container related files in a separate directory. This is useful when multiple applications are deployed.

This is done using the **WORKDIR** command

WORKDIR myhomedir

The last WORKDIR command decides which directory you are in when you start your container from your built image.

RUN git clone -q https://github.com/docker-in-practice/todo.git use EXPOSE command to assign an extended PORT number

```
EXPOSE  8080
```

The cmd command is used to specify which command will run when the application is started

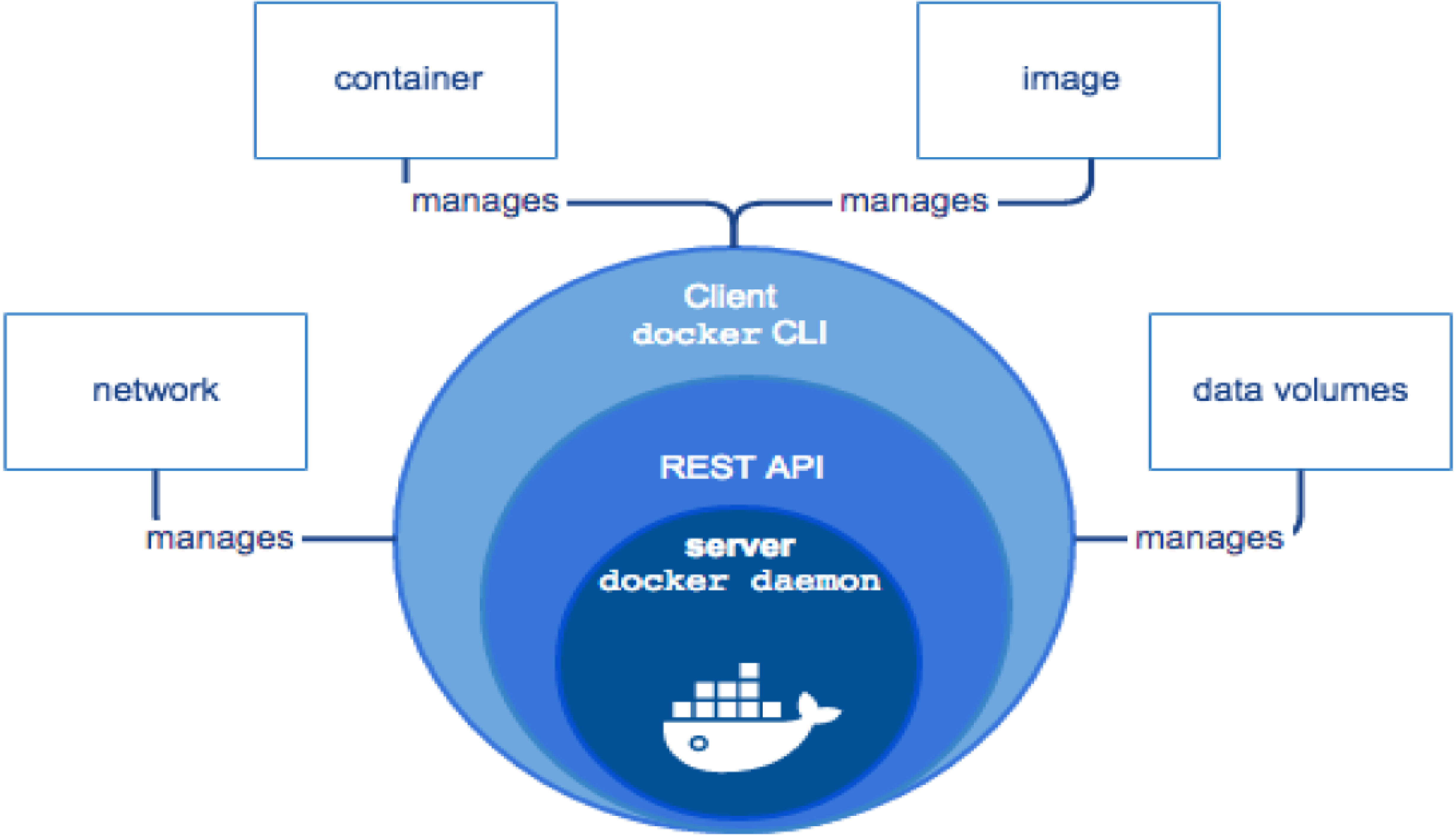
```
cmd [“npm”, “start”]
```

Following command builds the application using dockerfile

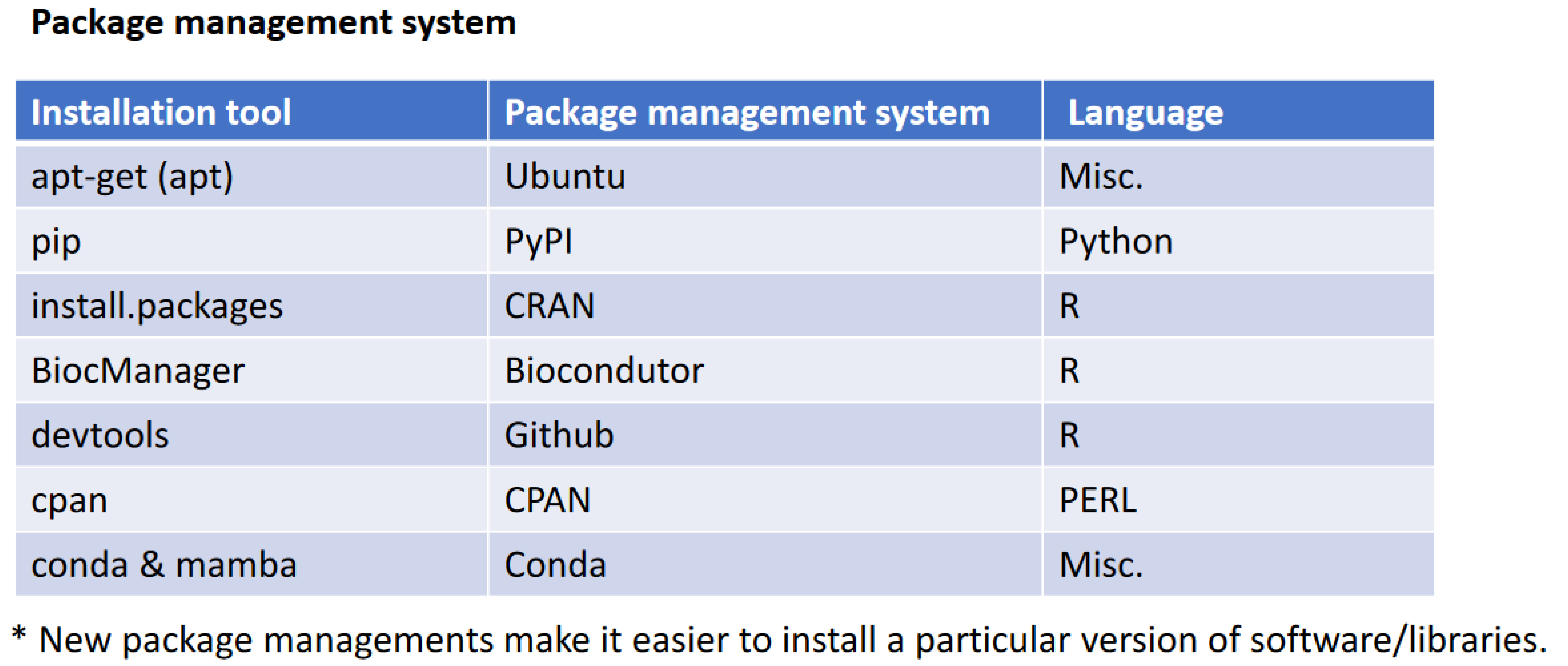
```
docker build .
```

```
docker run --rm hello-world
```

```
docker --version
```


container
image
manages
manages
Client
docker CLI
network
data volumes
REST API
manages
manages
server
docker daemon

Software installation methods



Package management system

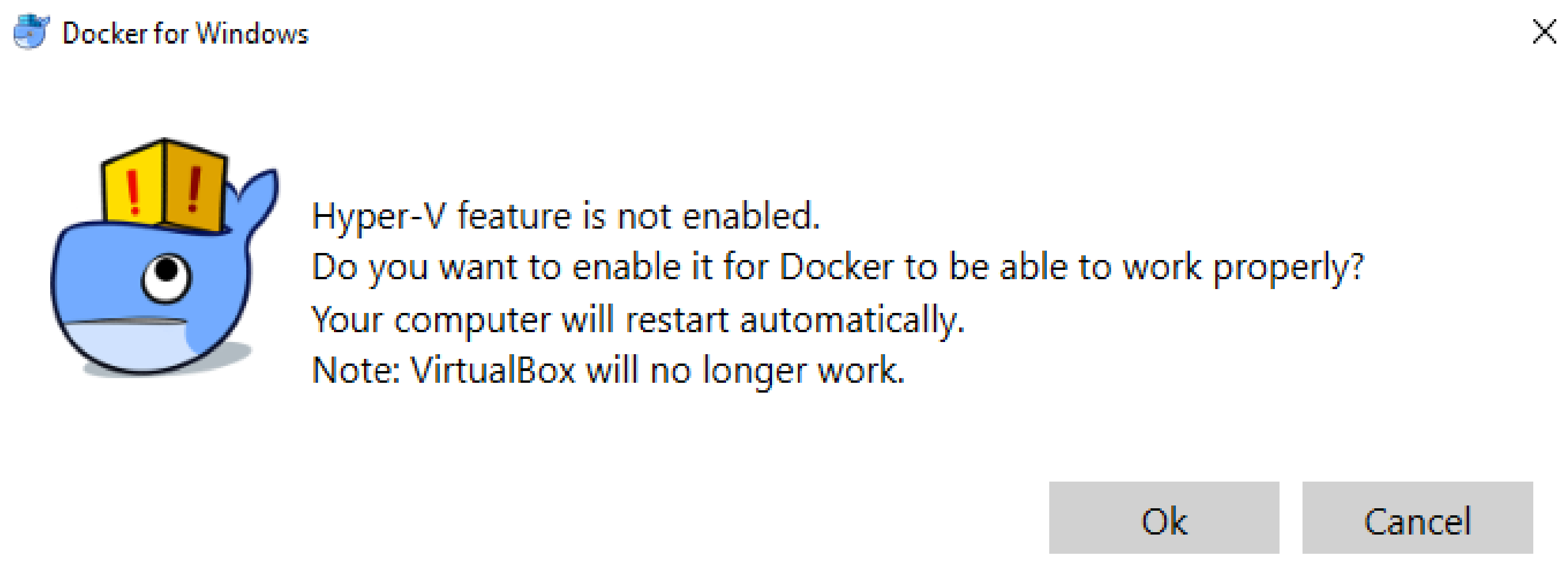
Installation tool	Package management system	Language
apt-get (apt)	Ubuntu	Misc.
pip	PyPI	Python
install.packages	CRAN	R
BiocManager	Biocondutor	R
devtools	Github	R
cpan	CPAN	PERL
conda & mamba	Conda	Misc.

* New package managements make it easier to install a particular version of software/libraries.

Enable Hyper-V on Windows

The Hyper-V must be enabled for docker to run in Windows.

If it is not already done, you may get following prompt:



Docker for Windows

Hyper-V feature is not enabled.
Do you want to enable it for Docker to be able to work properly?
Your computer will restart automatically.
Note: VirtualBox will no longer work.

How to turn on hardware virtualization

To enable virtualization in Windows 10, you must take several steps. First, you must know how to turn on hardware virtualization in BIOS settings. From there, you can use PowerShell, DISM, or the Settings app to turn on Hyper-V.

How to enable virtualization in BIOS settings

The exact steps for how to enable virtualization in BIOS can vary depending on your computer's motherboard manufacturer and BIOS version, but here's a general guide:

Enter the BIOS setup:

- Restart your computer.
- During the boot process, press a key to enter the BIOS setup. This key varies but is often one of the following: Del, F2, F10, Esc or F12. The correct key should be displayed during boot (e.g., "Press [Key] to enter setup").

Navigate to the Virtualization Settings:

- Once in the BIOS setup, look for a section related to CPU Configuration, Advanced or Security settings. The exact name can vary.
- Within this section, search for options related to virtualization. Common names include:
  - Virtualization Technology (VT-x)
  - Intel Virtualization Technology
  - AMD-V (for AMD processors)
  - SVM Mode (for AMD processors)

Enable virtualization:

- Select the virtualization option and change the setting to Enabled.
- If you see options like VT-d or IOMMU, enabling them can also improve performance for virtual machines but is not strictly necessary for Hyper-V.

Save and exit:

- Follow the instructions within your BIOS to Save and Exit. This is often done by pressing the F10 key, but ensure to follow the prompts on your screen.
- Your computer will reboot with the changes applied.

Enable Hyper-V with DISM

DISM is a command-line tool that can be used to service and prepare Windows images, including those used for Windows PE, Windows Recovery Environment (Windows RE) and Windows Setup. It can also be used to enable Windows features while the operating system is running.

Here's how to enable Hyper-V using DISM:

Open Command Prompt as Administrator:

Right-click the Start button or press Windows + X, then select "Command Prompt (Admin)" or "Windows Terminal (Admin)" if Command Prompt isn't directly listed. On Windows 11, you're more likely to find "Windows Terminal (Admin)".

If prompted by the User Account Control (UAC) dialog, click "Yes" to allow the app to make changes to your PC.

Enable Hyper-V using DISM:

Type the following command into the Command Prompt and press Enter:

```
DISM /Online /Enable-Feature /All /FeatureName:Microsoft-Hyper-V
```

Restart your computer:

After the command completes, you'll likely need to restart your computer for the changes to take effect. The command prompt might prompt you to do this or you can manually restart your computer.

Enable Hyper-V through Settings

If you prefer a more user-friendly approach, you can enable virtualization on Windows 10 through the Settings app. This method provides a graphical user interface (GUI) for users who prefer not to use the command line. To do this, follow these steps:

Open the Settings app:

Press Windows + I to open the Settings app, or click on the Start menu and then click on the gear/settings icon.

Access the Windows Features:

For Windows 10: Navigate to “Apps” > “Optional Features” > “More Windows features” at bottom of page.

For Windows 11: Navigate to “Apps” > “Optional Features” > “More Windows features” at the bottom of the “Related settings” section.

Enable Hyper-V:

In the “Windows Features” window that opens, scroll down to find “Hyper-V” and check the box next to it.

Make sure both “Hyper-V Management Tools” and “Hyper-V Platform” are selected. The management tools include the Hyper-V Manager, which is a GUI tool for managing a Hyper-V virtual machine on Windows 10, and the platform provides the necessary services and management for virtual machines.

Click “OK” to begin the installation process. Windows will apply the changes, which may take a few minutes.

Restart your computer:

After the installation is complete, you will likely be prompted to restart your computer. Make sure to save any open work and then restart your computer to complete the installation process.

WINDOWS VERSION

Windows 10 with 64 bit

Windows 11 with 64 bit

2004 or higher version of Pro

1909 version or higher for Enterprise or Education Version

4GB RAM or Higher

BIOS settings should have hardware virtualization support enabled

Hyper V feature, WSL 2 feature and Container feature should be enabled in windows

SET UP DOCKER ON YOUR COMPUTER

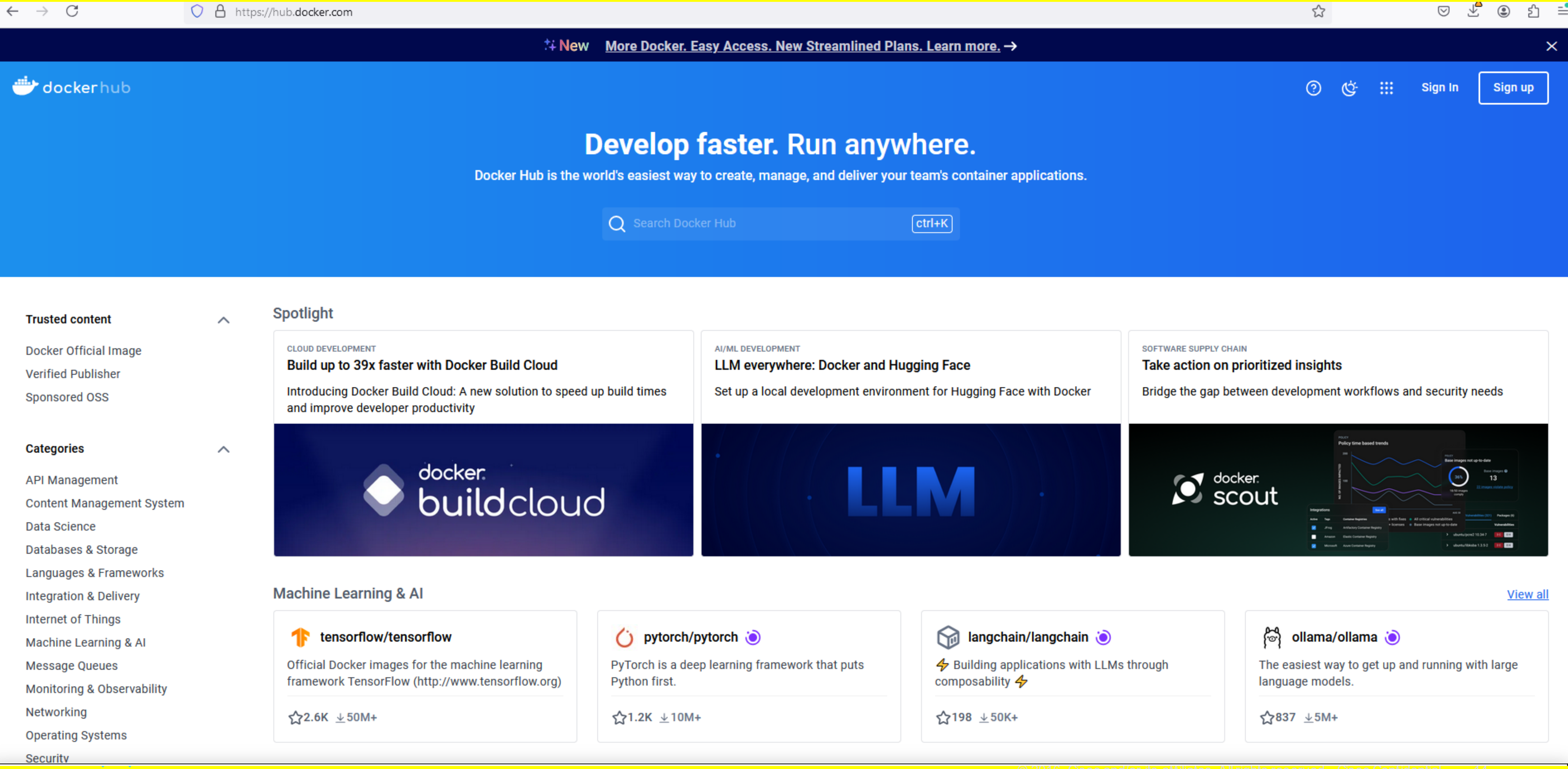
If you haven't already, create a DockerHub account and install Docker on your computer.

You'll want to look for the Docker Community Edition for your operating system.

It takes some time for Docker to start, especially the first time.

Once Docker starts, it won't open a window; you'll just see a little whale and container icon in one of your computers toolbars.

In order to actually use Docker, you'll need to open a command line program (like Terminal, or Command Prompt) and run commands there.



https://hub.docker.com

✨ New More Docker. Easy Access. New Streamlined Plans. Learn more. →

docker hub

Sign In Sign up

Develop faster. Run anywhere.

Docker Hub is the world's easiest way to create, manage, and deliver your team's container applications.

Search Docker Hub ctrl+K

Trusted content

- Docker Official Image
- Verified Publisher
- Sponsored OSS

Categories

- API Management
- Content Management System
- Data Science
- Databases & Storage
- Languages & Frameworks
- Integration & Delivery
- Internet of Things
- Machine Learning & AI
- Message Queues
- Monitoring & Observability
- Networking
- Operating Systems
- Security

Spotlight

CLOUD DEVELOPMENT

Build up to 39x faster with Docker Build Cloud

Introducing Docker Build Cloud: A new solution to speed up build times and improve developer productivity

docker buildcloud

AI/ML DEVELOPMENT

LLM everywhere: Docker and Hugging Face

Set up a local development environment for Hugging Face with Docker

LLM

SOFTWARE SUPPLY CHAIN

Take action on prioritized insights

Bridge the gap between development workflows and security needs

docker scout

Machine Learning & AI

View all

tensorflow/tensorflow

Official Docker images for the machine learning framework TensorFlow (http://www.tensorflow.org)

☆2.6K ↓50M+

pytorch/pytorch

PyTorch is a deep learning framework that puts Python first.

☆1.2K ↓10M+

langchain/langchain

⚡ Building applications with LLMs through composability ⚡

☆198 ↓50K+

ollama/ollama

The easiest way to get up and running with large language models.

☆837 ↓5M+

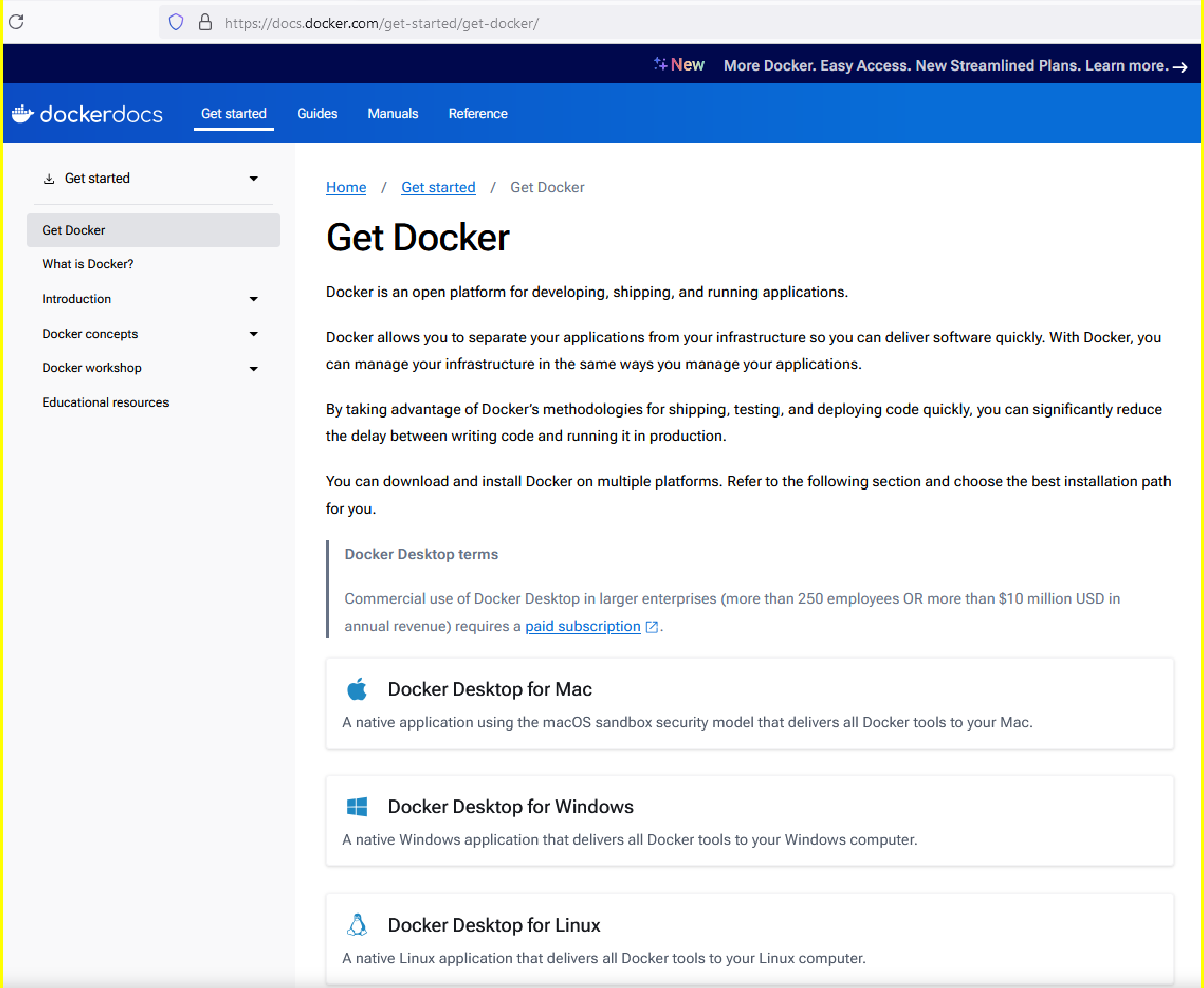
Set Up Docker on Your Computer

Start a Docker Hub account at https://hub.docker.com/signup

Install Docker Desktop (OSX or Win 10 Pro) or Docker on a Linux Distribution

Use an official image and choose the smallest image that meets your needs.

Debian is easy because it's one of the smaller containers that already contains the libraries R and Python rely on.


https://docs.docker.com/get-started/get-docker/
New More Docker. Easy Access. New Streamlined Plans. Learn more. →
dockerdocs
Get started
Guides
Manuals
Reference
Get started
Get Docker
What is Docker?
Introduction
Docker concepts
Docker workshop
Educational resources
Home / Get started / Get Docker

Get Docker

Docker is an open platform for developing, shipping, and running applications.

Docker allows 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.

> **Docker Desktop terms**
>
> Commercial use of Docker Desktop in larger enterprises (more than 250 employees OR more than $10 million USD in annual revenue) requires a paid subscription.

Docker Desktop for Mac

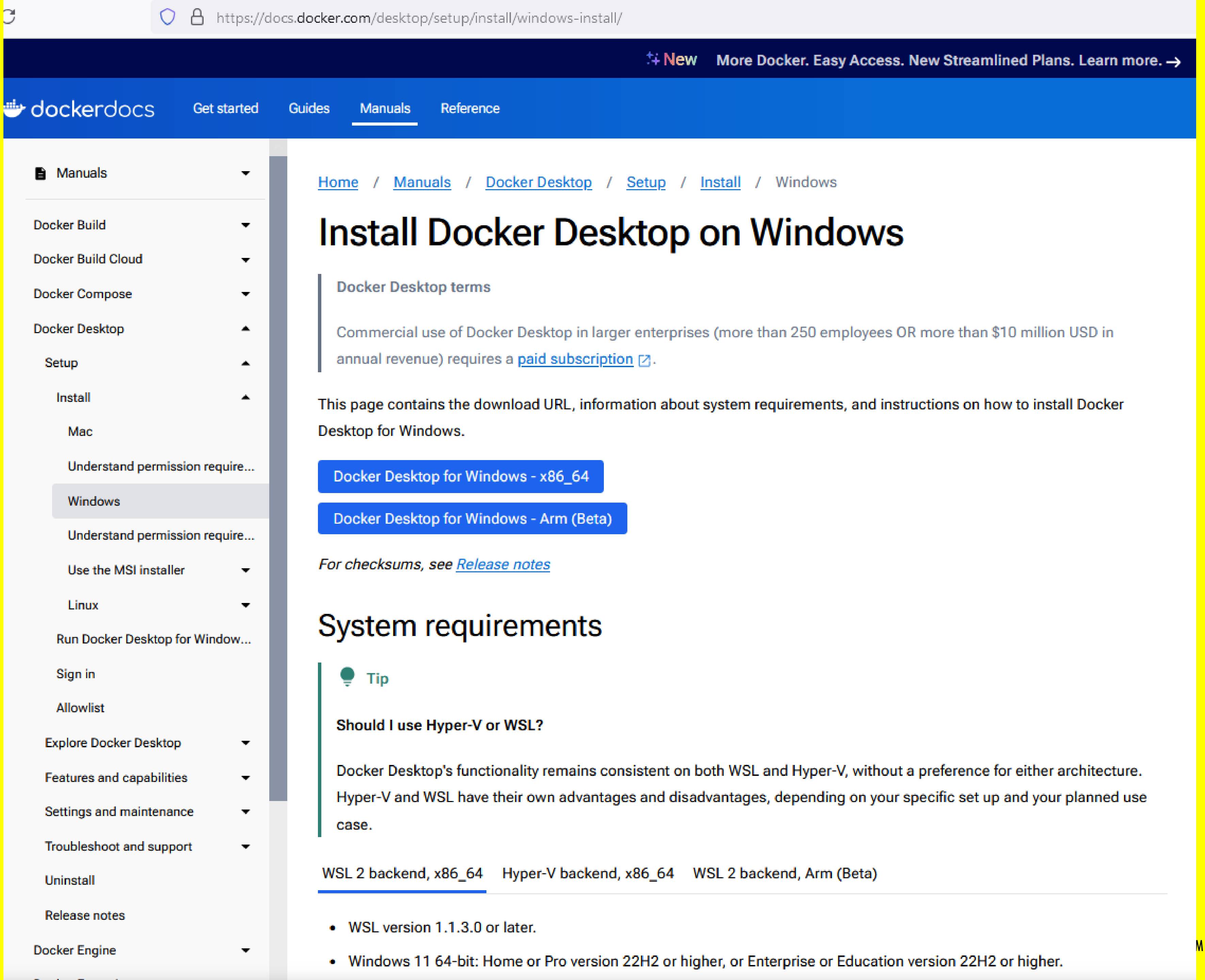
A native application using the macOS sandbox security model that delivers all Docker tools to your Mac.

Docker Desktop for Windows

A native Windows application that delivers all Docker tools to your Windows computer.

Docker Desktop for Linux

A native Linux application that delivers all Docker tools to your Linux computer.


https://docs.docker.com/desktop/setup/install/windows-install/
New More Docker. Easy Access. New Streamlined Plans. Learn more. →
dockerdocs
Get started
Guides
Manuals
Reference
Manuals
Docker Build
Docker Build Cloud
Docker Compose
Docker Desktop
Setup
Install
Mac
Understand permission require...
Windows
Understand permission require...
Use the MSI installer
Linux
Run Docker Desktop for Window...
Sign in
Allowlist
Explore Docker Desktop
Features and capabilities
Settings and maintenance
Troubleshoot and support
Uninstall
Release notes
Docker Engine
Home / Manuals / Docker Desktop / Setup / Install / Windows

Install Docker Desktop on Windows

> **Docker Desktop terms**
>
> Commercial use of Docker Desktop in larger enterprises (more than 250 employees OR more than $10 million USD in annual revenue) requires a paid subscription.

This page contains the download URL, information about system requirements, and instructions on how to install Docker Desktop for Windows.

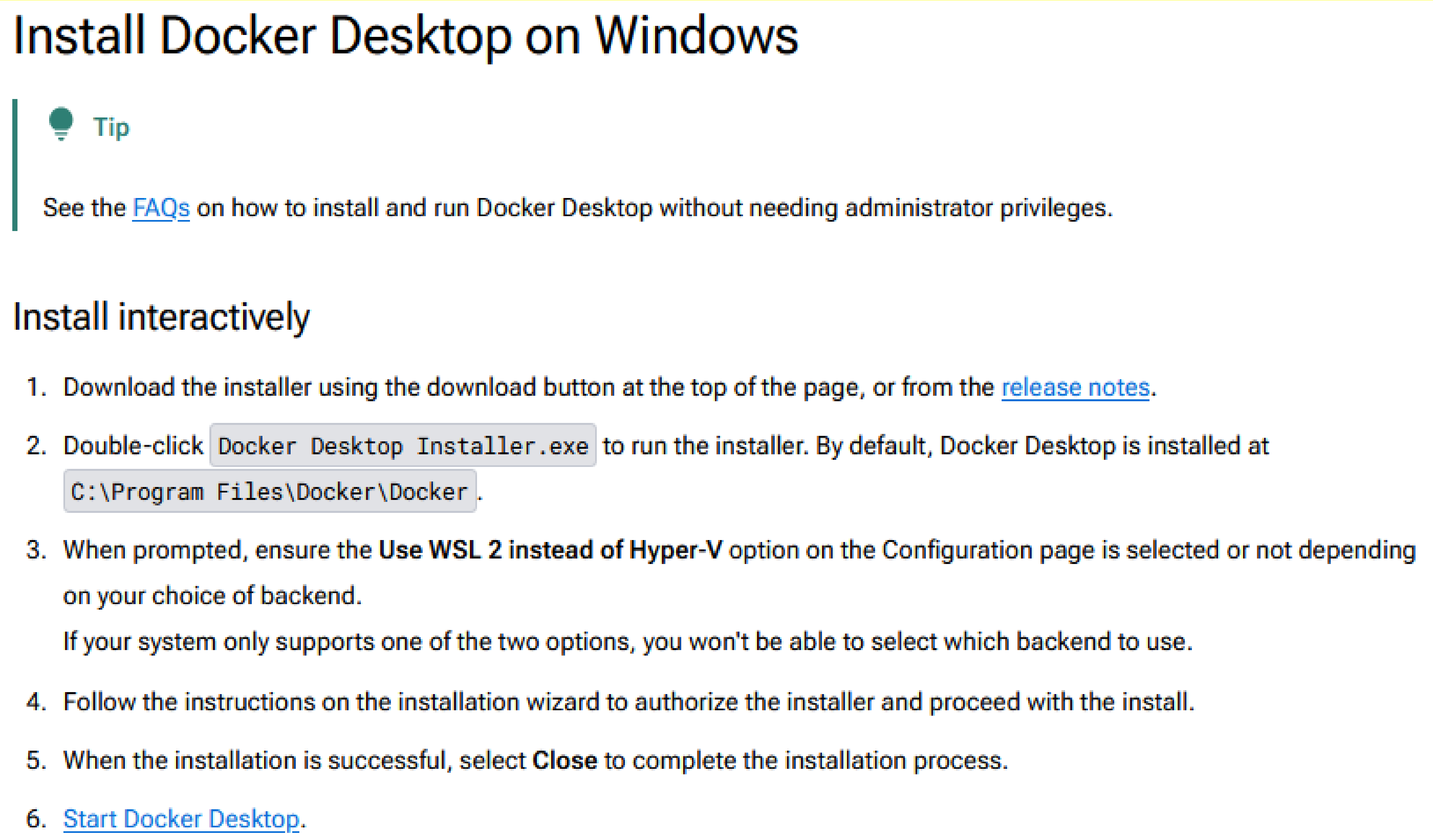
Docker Desktop for Windows - x86_64
Docker Desktop for Windows - Arm (Beta)

For checksums, see Release notes

System requirements

> Tip
>
> **Should I use Hyper-V or WSL?**
>
> Docker Desktop's functionality remains consistent on both WSL and Hyper-V, without a preference for either architecture. Hyper-V and WSL have their own advantages and disadvantages, depending on your specific set up and your planned use case.

- WSL version 1.1.3.0 or later.
- Windows 11 64-bit: Home or Pro version 22H2 or higher, or Enterprise or Education version 22H2 or higher.



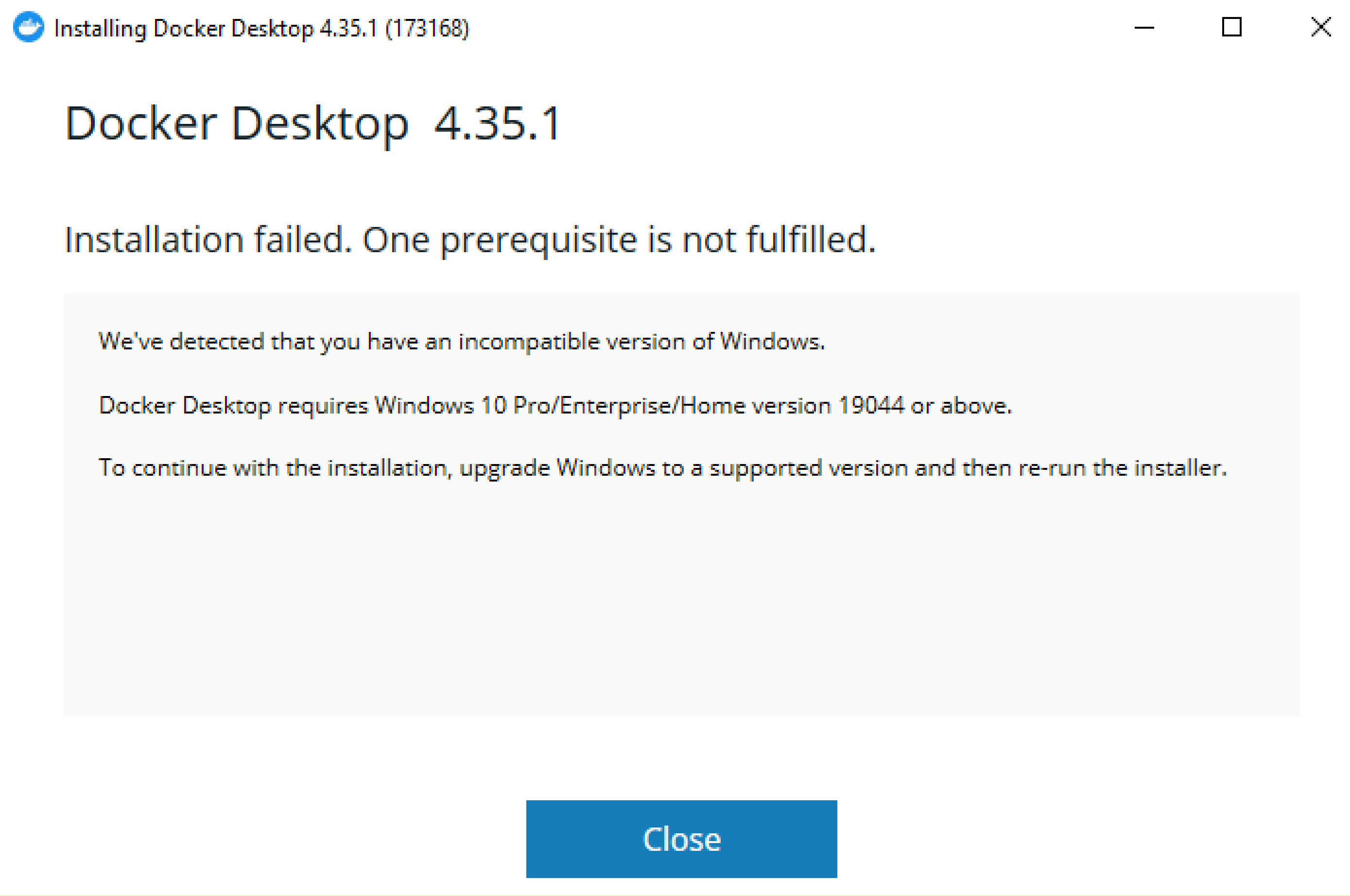
Install Docker Desktop on Windows

> Tip
>
> See the FAQs on how to install and run Docker Desktop without needing administrator privileges.

Install interactively

1. Download the installer using the download button at the top of the page, or from the release notes.
2. Double-click `Docker Desktop Installer.exe` to run the installer. By default, Docker Desktop is installed at `C:\Program Files\Docker\Docker`.
3. When prompted, ensure the **Use WSL 2 instead of Hyper-V** option on the Configuration page is selected or not depending on your choice of backend.

   If your system only supports one of the two options, you won't be able to select which backend to use.
4. Follow the instructions on the installation wizard to authorize the installer and proceed with the install.
5. When the installation is successful, select **Close** to complete the installation process.
6. Start Docker Desktop.



Installing Docker Desktop 4.35.1 (173168)

Docker Desktop 4.35.1

Installation failed. One prerequisite is not fulfilled.

We've detected that you have an incompatible version of Windows.

Docker Desktop requires Windows 10 Pro/Enterprise/Home version 19044 or above.

To continue with the installation, upgrade Windows to a supported version and then re-run the installer.

Set Up Docker on Your Computer

Debian has an official image in Docker Hub

Create the Dockerfile (pure text file)

FROM Debian:stretch-slim:

FROM is Docker’s way of specifying that this is the base image layer. We just have containerized Linux distribution.

apt is a package manager.

apt-get is used to install software and dependencies (just like pip in windows)

Set Up Docker on Your Computer

To run Windows containers, you need Windows 10 or Windows 11 Professional or Enterprise edition.

Windows Home or Education editions only allow you to run Linux containers.

If you have a prior version, you must update the Windows.

Our CTS team in room 301A can help you update your windows OS.

Docker run

The docker run command creates and starts a new container, also pulling the image if it is needed to start the container.

It is used when we want to create a container for the first time.

```
docker run -d  --name <container-name>
```

-d: run the container in the background and print the new container ID

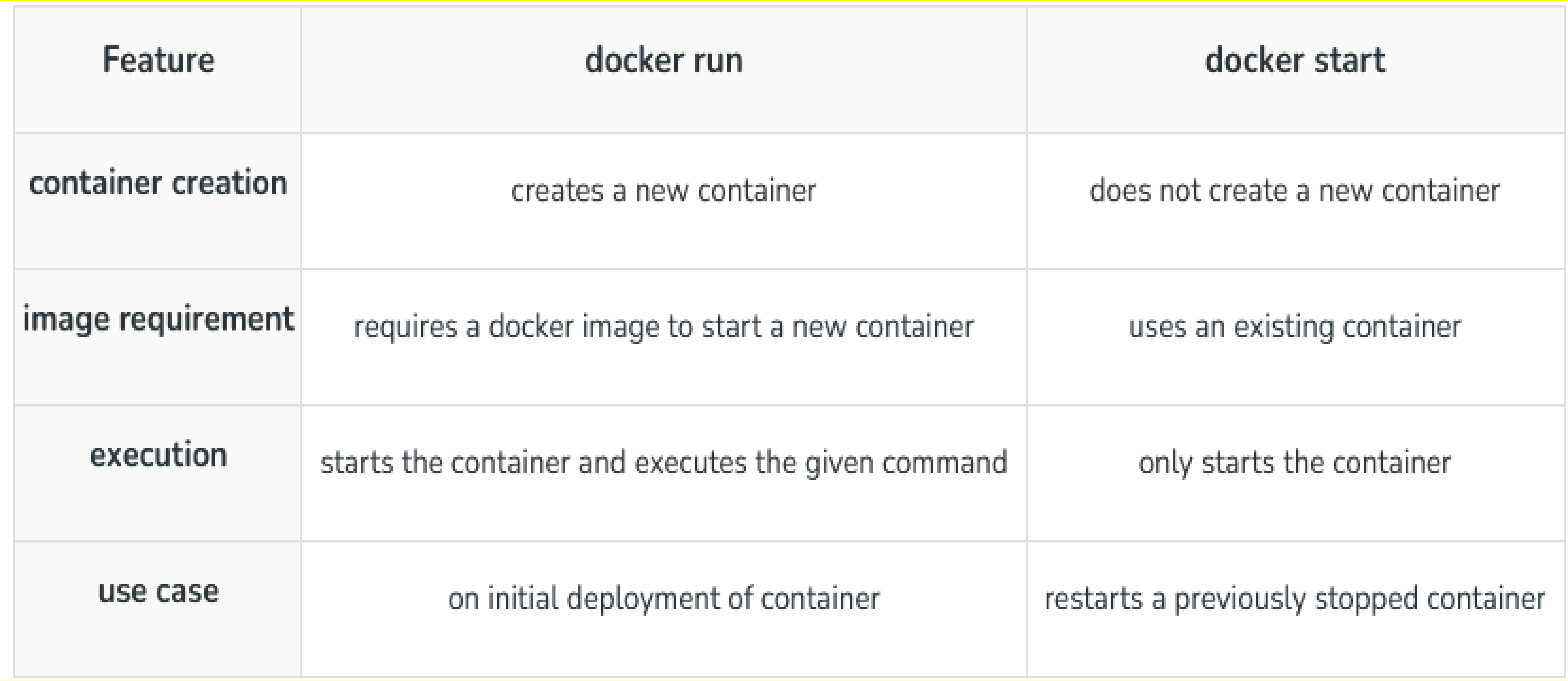
--name: assign the name container-name to the container

```
docker pull dockerinaction/hello_world
```

Step-by-step procedure for Docker run

1. Pull the image: If the specified image is not available, docker pulls it from the registry.
2. Create container: a new container is created from the specified image.
3. Assign resources: Compute resources like CPU, memory, volumes, and ports are assigned to the container.
4. Start the container: The container is started, and the given command is executed.

Docker run vs Docker start



Feature	docker run	docker start
container creation	creates a new container	does not create a new container
image requirement	requires a docker image to start a new container	uses an existing container
execution	starts the container and executes the given command	only starts the container
use case	on initial deployment of container	restarts a previously stopped container

DOCKERFILE

A Dockerfile is a plain text file with keywords that add elements to a Docker image. There are many keywords that can be used in a Dockerfile (Dockerfile keywords), but we will use a subset of these keywords following this basic outline:

Starting point: Which Docker image do you want to start with?

Additions: What needs to be added? Folders? Data? Other software?

Environment: What variables (if any) are set as part of the software installation?

If you are planning on making multiple images for different parts of your workflow, you should create a separate folder for each new image with a Dockerfile inside each of them.

DOCKERFILE

Dockerfile is essentially a file that outlines each step in creating your image. The common commands used in a Dockerfile are:

FROM - Dictates what the base image you're building off of.

LABEL - A simple label attached to your image as metadata.

A common label is a description of the image.

RUN - Runs the command you specify in the image. For example, if the base image is Ubuntu, then you can run any Ubuntu commands here. Common things to run would be apt-get install <package> to install an Ubuntu package into your container.

CMD - The command that should run when the container is started. This tends to be the major software that is being packaged.

Install all dependent software

We need to install R, Python, and Git.

We can do this all using the apt package manager with the command apt-get.

Pip package manager will work for windows:

The following command installs numpy and scipy

```
FROM python:3.8
RUN pip3 install numpy scipy
```

Set up the environment with ENV

Your software might rely on certain environment variables being set correctly.

If you’re installing a program to a custom location (like a home directory), you may need to add that directory to the image’s system PATH.

For example, if you installed some scripts to /home/software/bin, you could use

```
ENV PATH="/home/software/bin:${PATH}"
```

to add them to your PATH.

Install all dependent software

```
RUN apt-get update \
&& apt-get install -y --no-install-recommends r-base r-base-dev git python3 \
&& apt-get clean
```

RUN tells Docker to RUN the commands that follow.

apt-get update tells “apt-get” to get the most recent sources for packages you’ll install.

apt-get install is the command that proceeds the list of applications you wish to install.

Docker Platform

The core component of the Docker platform is called Docker Engine, which is composed of three parts:

- A server, which is the Docker daemon process (dockerd) that manages the different types of Docker objects: containers, images, etc.
- A REST (Representational state transfer) API (Application programming interface) that offers an intermediate layer to interact with the Docker server and control all its features.
- A command line interface which allows a user to communicate and interact with the server through the REST API.

Docker image

A software file, including OS, libraries and executables.

(typical size 0.1-2.0 gb)

Deposited in Dockerhub

Docker container

A running instance of the image.

Running on your computer. Removed when power down.

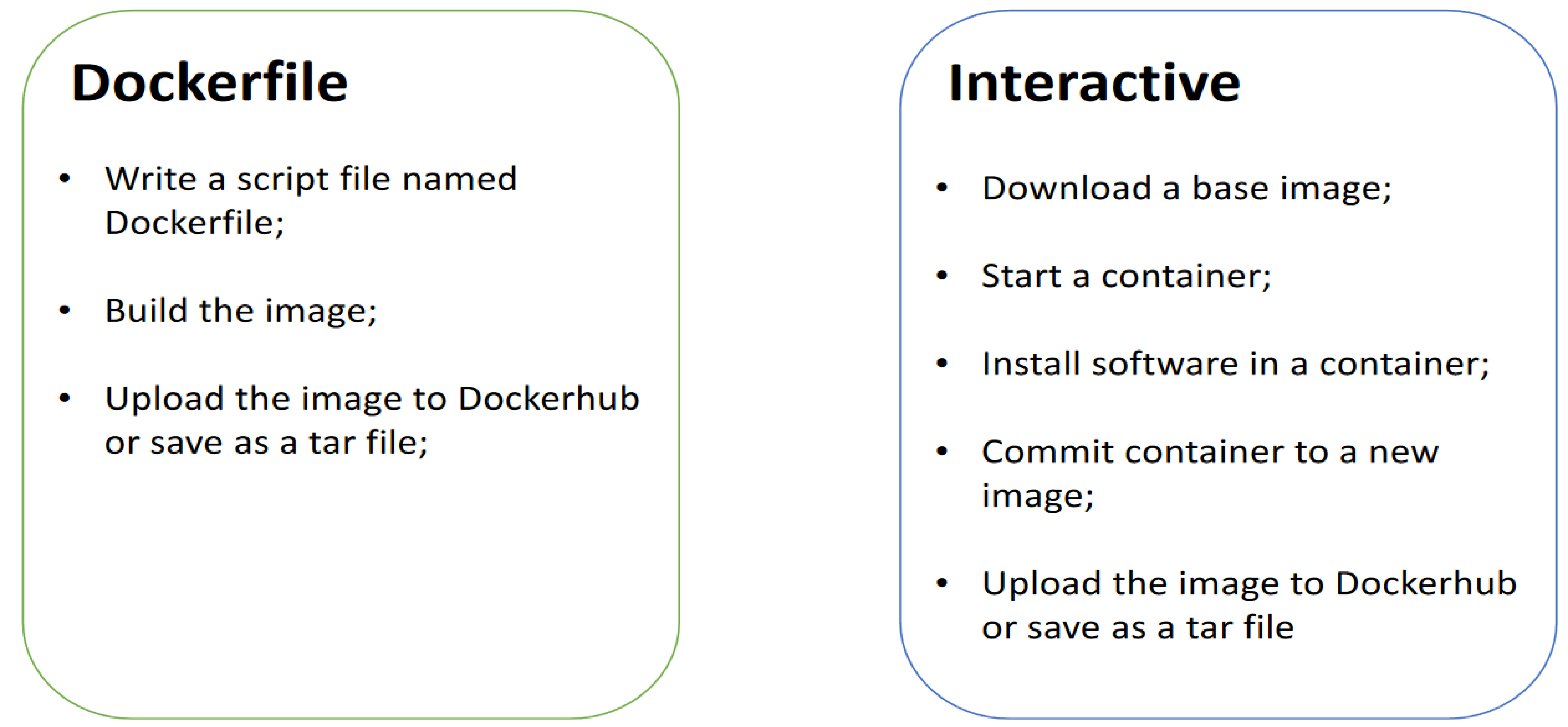
Dockerfile

A script text file with instructions how a image is build.

(typical size <10 kb)

Deposited in Github

Many ways to build a Docker Image



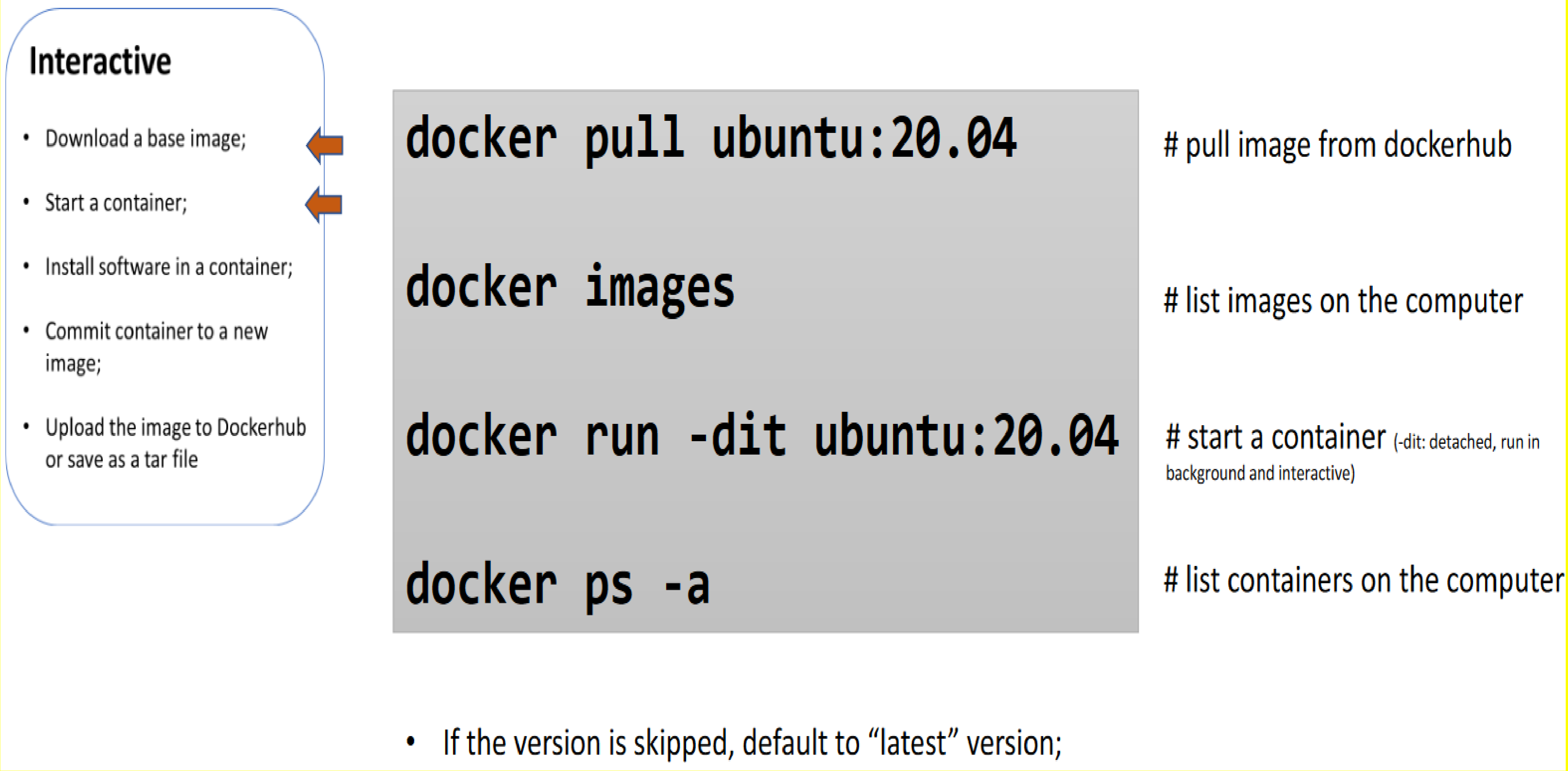
Dockerfile

- Write a script file named Dockerfile;
- Build the image;
- Upload the image to Dockerhub or save as a tar file;

Interactive

- Download a base image;
- Start a container;
- Install software in a container;
- Commit container to a new image;
- Upload the image to Dockerhub or save as a tar file

build a Docker Image Interactively



Interactive

- Download a base image;
- Start a container;
- Install software in a container;
- Commit container to a new image;
- Upload the image to Dockerhub or save as a tar file

```
docker pull ubuntu:20.04           # pull image from dockerhub

docker images                      # list images on the computer

docker run -dit ubuntu:20.04       # start a container (-dit: detached, run in background and interactive)

docker ps -a                       # list containers on the computer
```

- If the version is skipped, default to “latest” version;

BASE IMAGE

Usually you don’t want to start building your image from scratch. Instead you’ll want to choose a “base” image to add things to. You can find a base image by searching DockerHub.

If you’re using a scripting language like Python, R or perl, you could start with the “official” image from these languages.

If you’re not sure what to start with, using a basic Linux image (Debian, Ubuntu and CentOS are common examples) is often a good place to start.

Once you’ve decided on a base image and version, add it as the first line of your Dockerfile

```
FROM repository/image:tag
```

BASE IMAGE

Some images are maintained by DockerHub itself (these are the “official” images mentioned above), and do not have a repository. For example, to start with Centos 7, you could use

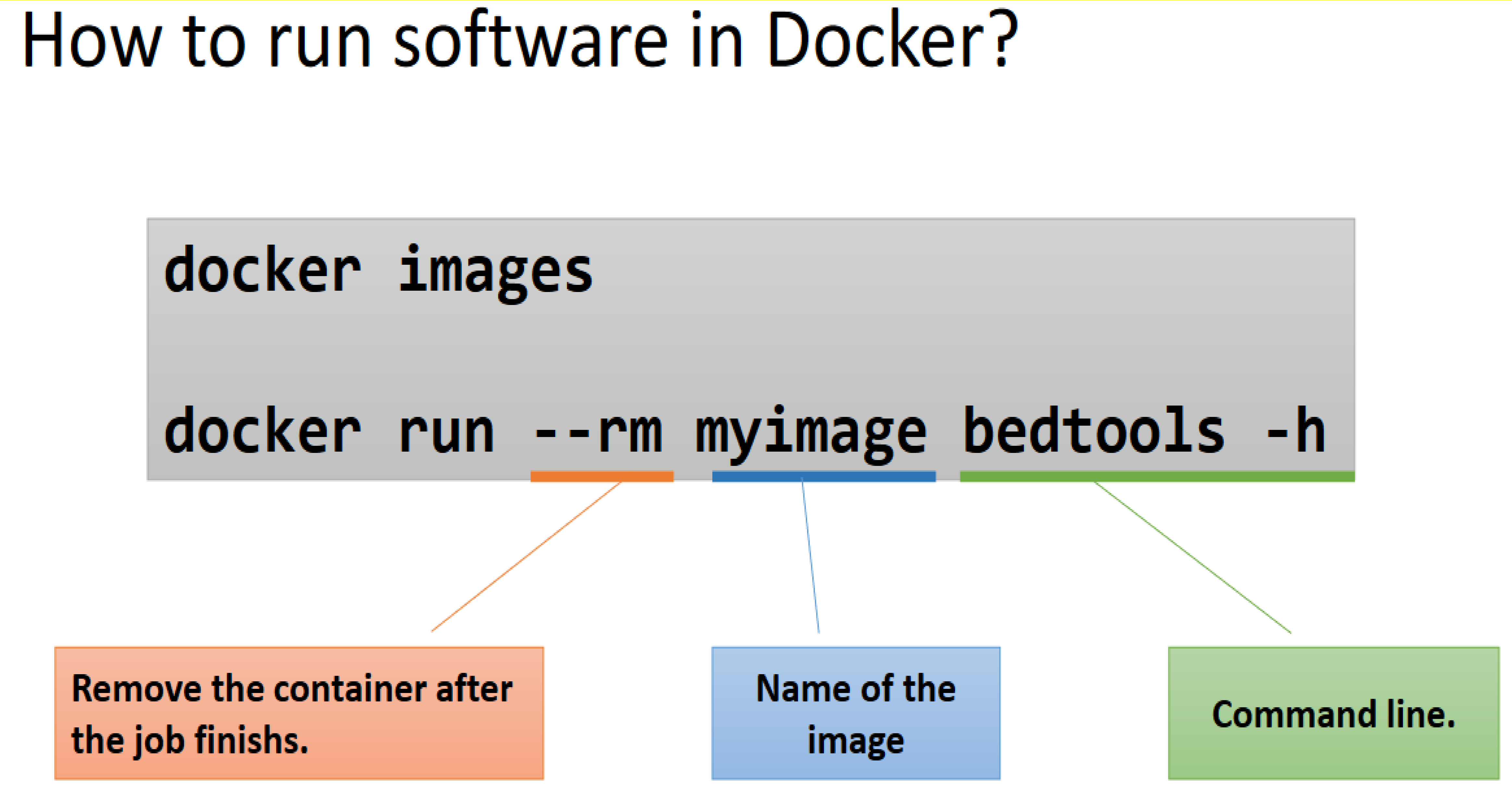
```
FROM centos:7
```

while starting from one of HTCondor’s HTC Jupyter notebook images might look like

```
FROM htcondor/htc-minimal-notebook:2019-12-02
```

When possible, you should use a specific tag (not the automatic latest tag) in FROM statements. Here are some base images you might find useful to build off of:

- Centos
- Ubuntu
- Python / Anaconda / Miniconda
- R / Tidyverse
- Tensorflow
- PyTorch



How to run software in Docker?

```
docker images

docker run --rm myimage bedtools -h
```

Remove the container after the job finishs.

Name of the image

Command line.

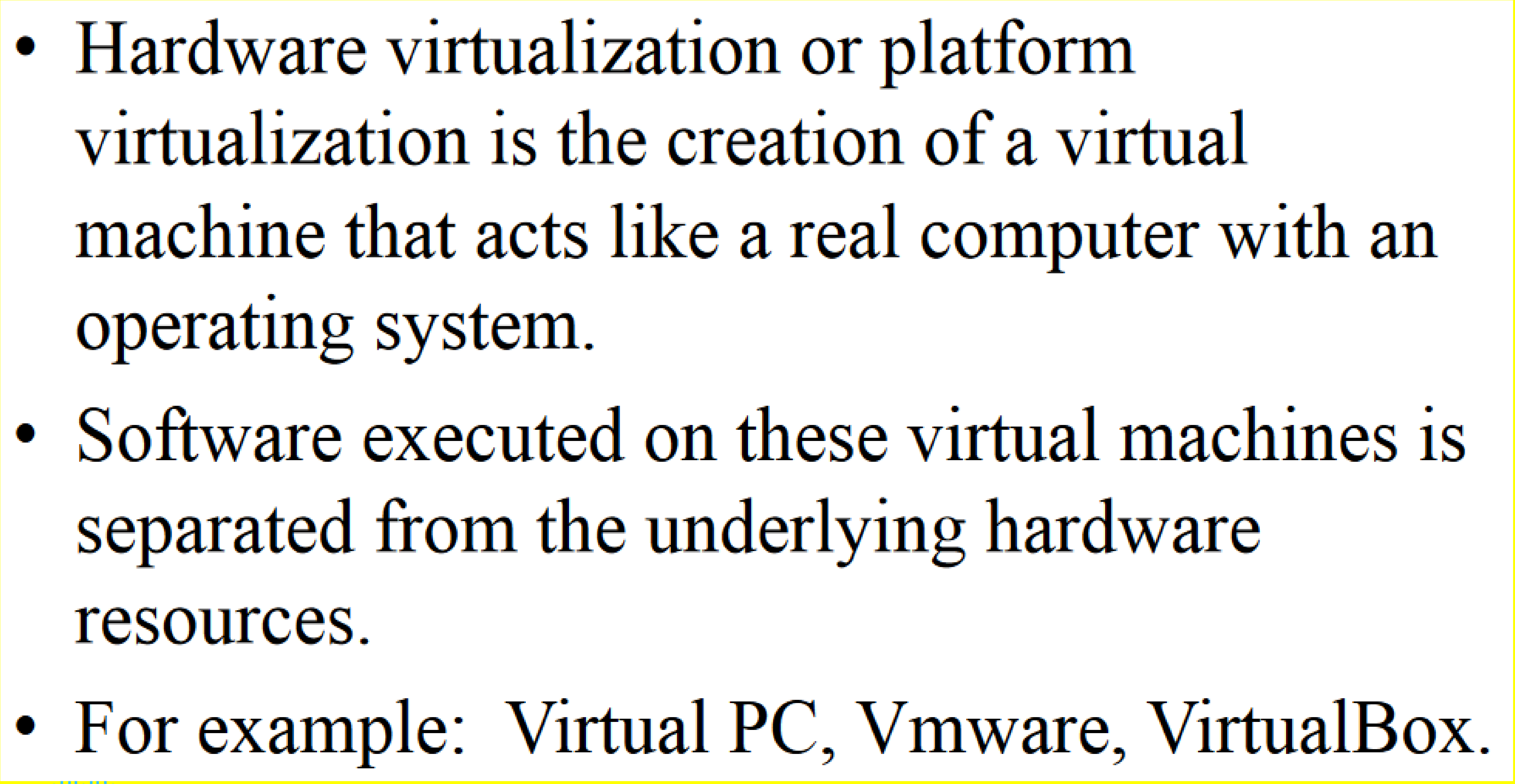
DOCKER IS DEPENDENT ON LINUX KERNEL

The Docker server/daemon, part of the Docker Engine, can only run under an environment that can have access to Linux kernel specific features.

It is possible to run Docker Engine in other OS as long as they are capable of virtualizing Linux.

Docker has a set of tools called Docker Toolbox for users to run Docker in non-Linux environments.

Hyper-V on Windows machines is enough to run Docker in Windows.



- Hardware virtualization or platform virtualization is the creation of a virtual machine that acts like a real computer with an operating system.
- Software executed on these virtual machines is separated from the underlying hardware resources.
- For example: Virtual PC, Vmware, VirtualBox.

AMAZON MACHINE IMAGE (AMI)

- It is a special type of virtual machine.
- The main component of an AMI is a read-only file system image which includes an operating system (e.g., Windows, Linux, UBUNTU, etc) and any additional software required to deliver a service or run an application.

From the AWS services drop-down list, search for “IAM”, which stands for Identity Access Management.

This is under the “Security” group of service

AMAZON EC2

To use Amazon EC2, you need to:

– Select the virtual hardware.

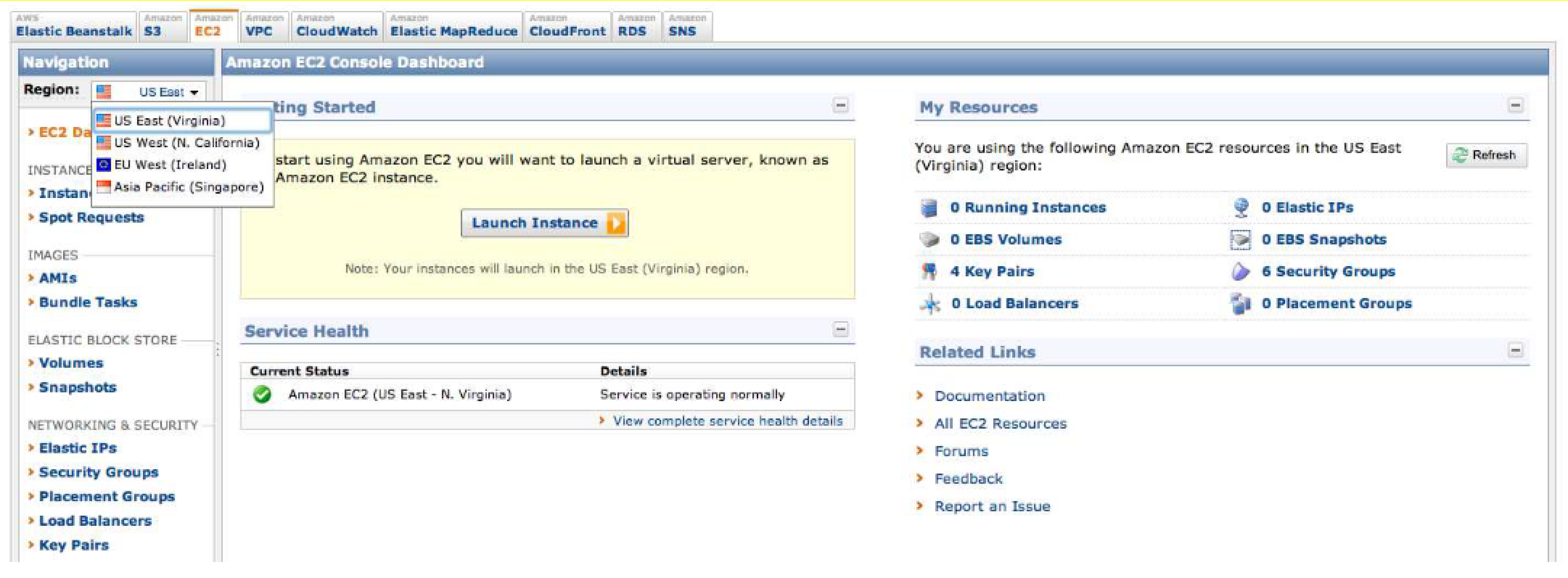
– Pick an AMI to run on the virtual hardware.

– Associate security credentials in order to login to the virtual machine.

Once you have access to AWS, create AWS account credentials to work with virtual

machines on EC2.

Credentials are required to access virtual machines by remote shell (SSH), and also to use the AWS command line interface, and programming APIs



Amazon EC2 Console Dashboard

ting Started

start using Amazon EC2 you will want to launch a virtual server, known as Amazon EC2 instance.

Note: Your instances will launch in the US East (Virginia) region.

Service Health

Current Status	Details
✓ Amazon EC2 (US East - N. Virginia)	Service is operating normally

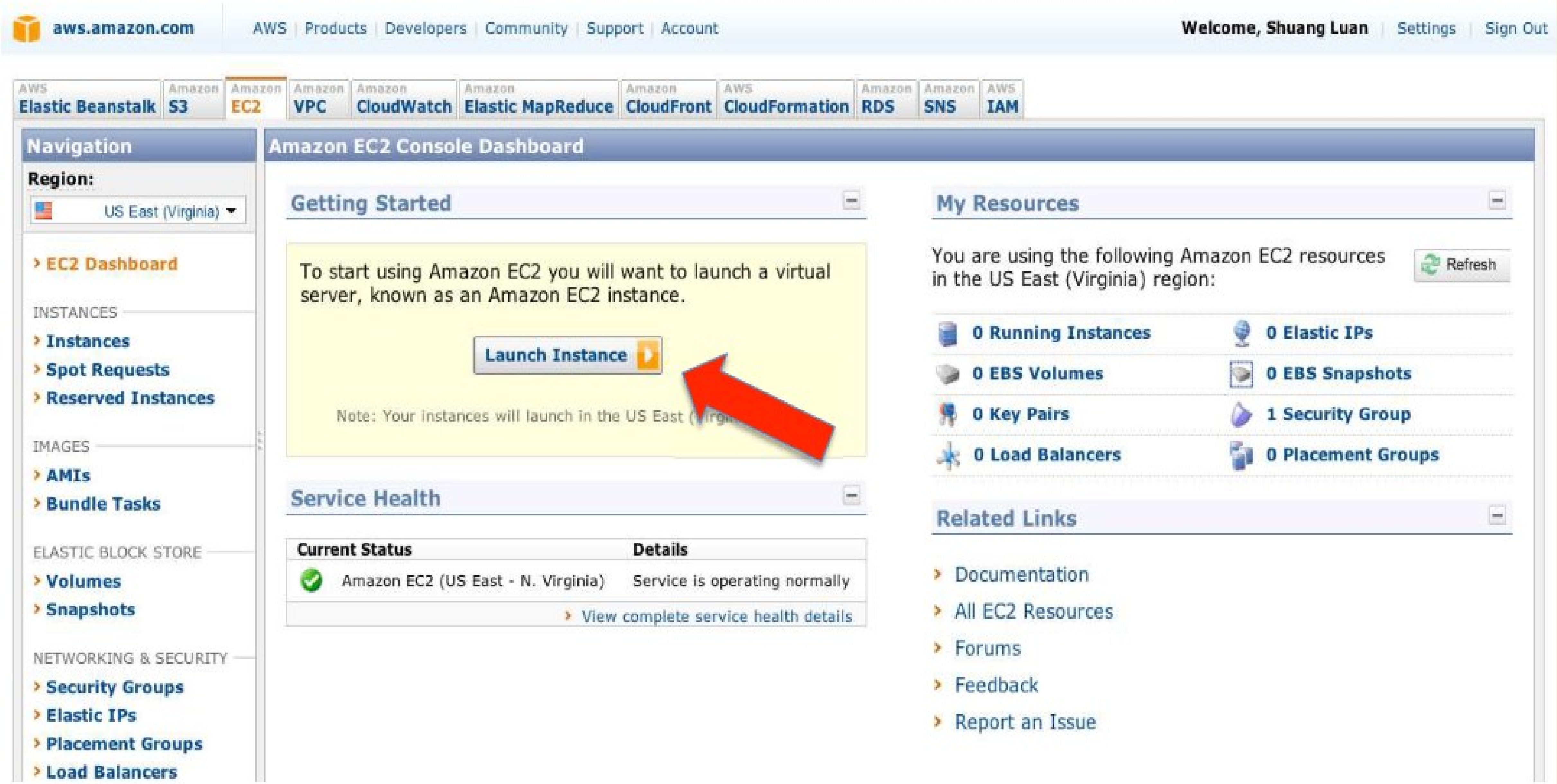
View complete service health details

My Resources

You are using the following Amazon EC2 resources in the US East (Virginia) region:

0 Running Instances	0 Elastic IPs
0 EBS Volumes	0 EBS Snapshots
4 Key Pairs	6 Security Groups
0 Load Balancers	0 Placement Groups

Related Links



Amazon EC2 Console Dashboard

Getting Started

To start using Amazon EC2 you will want to launch a virtual server, known as an Amazon EC2 instance.

Note: Your instances will launch in the US East (Virgi

Service Health

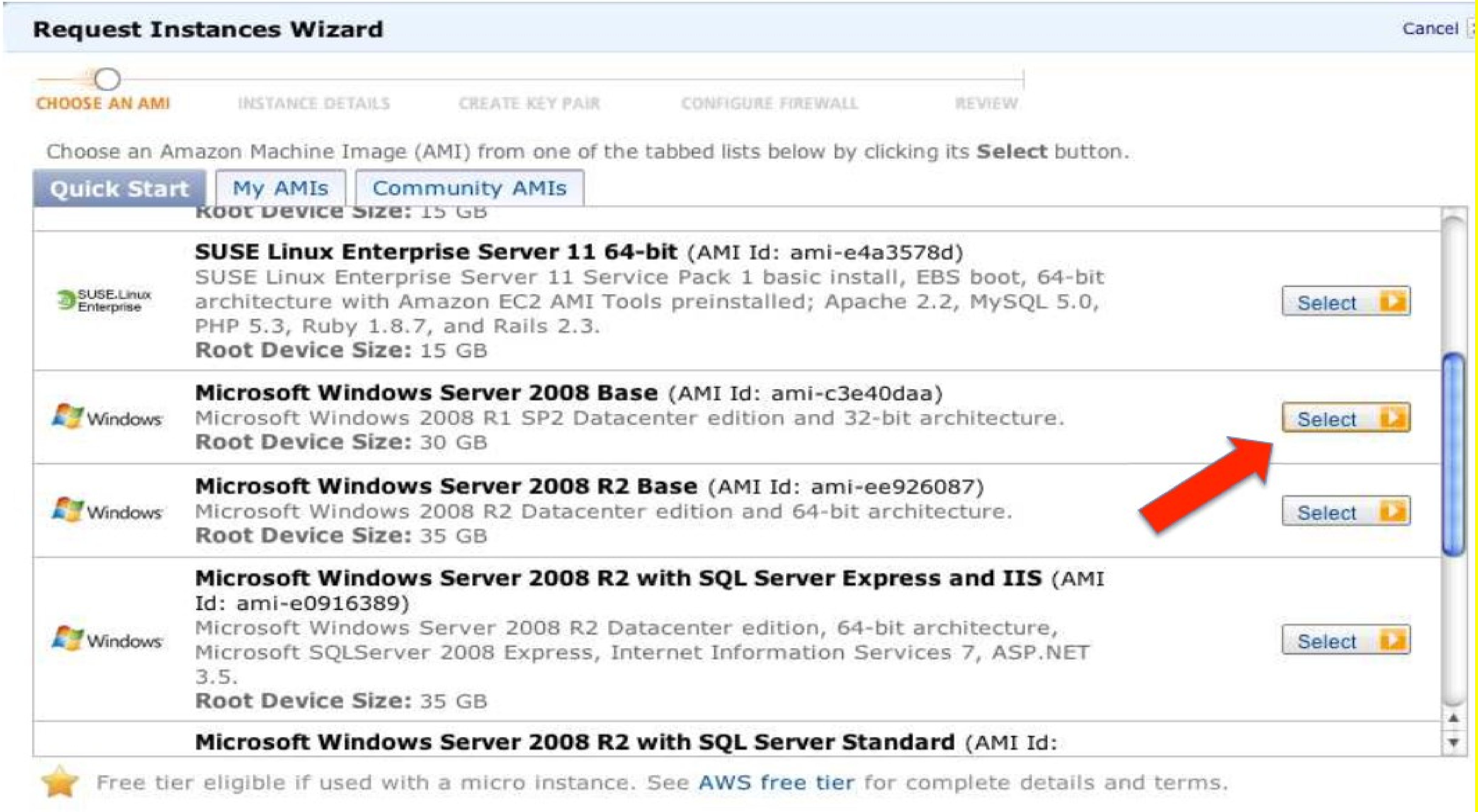
Current Status	Details
✓ Amazon EC2 (US East - N. Virginia)	Service is operating normally

My Resources

You are using the following Amazon EC2 resources in the US East (Virginia) region:

0 Running Instances	0 Elastic IPs
0 EBS Volumes	0 EBS Snapshots
0 Key Pairs	1 Security Group
0 Load Balancers	0 Placement Groups

Related Links



Choose an Amazon Machine Image (AMI) from one of the tabbed lists below by clicking its **Select** button.

Root Device Size: 15 GB

SUSE Linux Enterprise Server 11 64-bit (AMI Id: ami-e4a3578d)
SUSE Linux Enterprise Server 11 Service Pack 1 basic install, EBS boot, 64-bit architecture with Amazon EC2 AMI Tools preinstalled; Apache 2.2, MySQL 5.0, PHP 5.3, Ruby 1.8.7, and Rails 2.3.
Root Device Size: 15 GB

Microsoft Windows Server 2008 Base (AMI Id: ami-c3e40daa)
Microsoft Windows 2008 R1 SP2 Datacenter edition and 32-bit architecture.
Root Device Size: 30 GB

Microsoft Windows Server 2008 R2 Base (AMI Id: ami-ee926087)
Microsoft Windows 2008 R2 Datacenter edition and 64-bit architecture.
Root Device Size: 35 GB

Microsoft Windows Server 2008 R2 with SQL Server Express and IIS (AMI Id: ami-e0916389)
Microsoft Windows Server 2008 R2 Datacenter edition, 64-bit architecture, Microsoft SQLServer 2008 Express, Internet Information Services 7, ASP.NET 3.5.
Root Device Size: 35 GB

Microsoft Windows Server 2008 R2 with SQL Server Standard (AMI Id:

Free tier eligible if used with a micro instance. See AWS free tier for complete details and terms.

INTERACTING WITH AWS FROM WINDOWS

AWS only provides Windows Server AMIs, although Windows Server 2016 is the server equivalent platform to Windows 10.

AWS also provides licensed copies of Windows 10 through WorkSpaces

AWS CloudShell

The fifth highlighted item is the AWS CloudShell icon. By selecting this icon, you will launch a browser-based shell environment that is pre-authenticated with your console credentials. Use this to execute AWS CLI commands or scripts using the AWS CDK from your browser. If you add any files to CloudShell (up to the 1GB limit), it will persist the files between sessions.

INSTALL AND UPDATE THE AWS CLI VERSION 1 USING THE MSI INSTALLER

Check the Releases page on GitHub to see when the latest version was released. When updates are released, you must repeat the installation process to get latest version of AWS CLI version 1.

Download the appropriate MSI installer:

AWS CLI MSI installer for Windows (64-bit): https://s3.amazonaws.com/aws-cli/AWSCLI64PY3.msi

AWS CLI MSI installer for Windows (32-bit): https://s3.amazonaws.com/aws-cli/AWSCLI32PY3.msi

AWS CLI combined setup file for Windows: https://s3.amazonaws.com/aws-cli/AWSCLISetup.exe

(includes both the 32-bit and 64-bit MSI installers, and automatically installs the correct version)

Run the downloaded MSI installer or the setup file.

DOCKER IN VIRTUALBOX

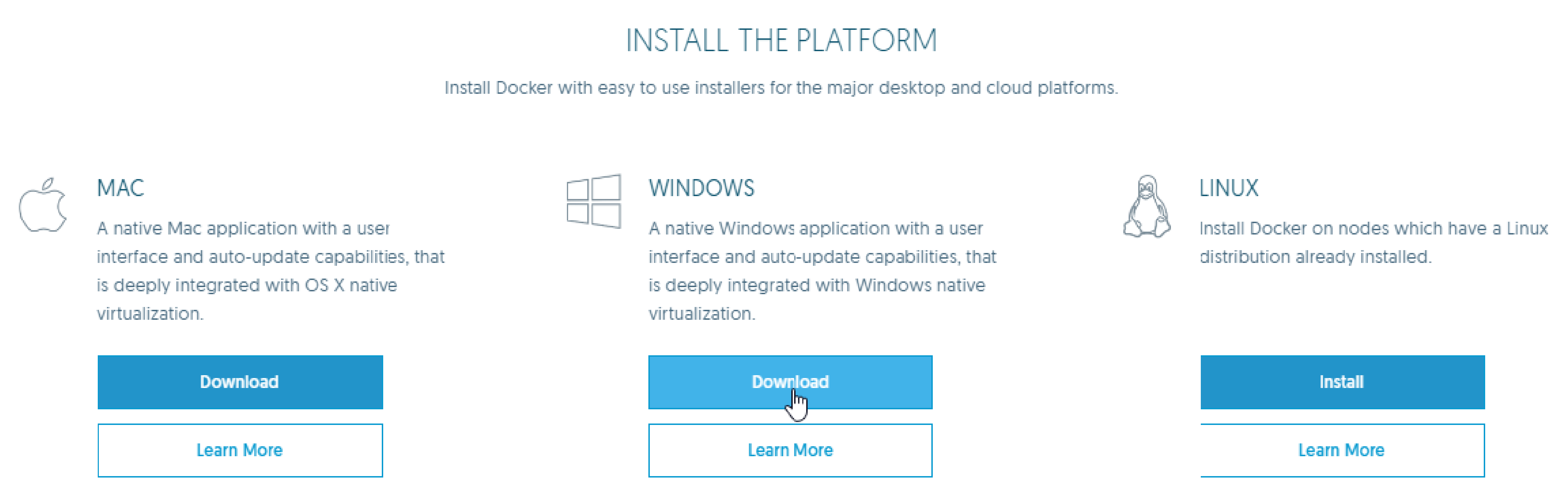
This solution installs Oracle VirtualBox, a VM hypervisor, that spins up a VirtualBox VM called boot2docker, containing a running Linux OS with a Docker daemon that is later used with the client in the host machine.

This is a functional solution, but requires deeper knowledge about how the connection with the daemon is being made, more dependencies (such as an hypervisor) and potential issues (boot2docker VM needed to be updated manually for each new daemon version, VM not operating correctly after unsuspending, worse performance, etc.).

DOCKER CAN RUN UNDER WINDOWS 10

Downloads the MSI (Microsoft Installer) file from http://www.docker.com/products/overview.

The download link can be found in the “Install the platform” section as shown below



INSTALL THE PLATFORM

Install Docker with easy to use installers for the major desktop and cloud platforms.

MAC

A native Mac application with a user interface and auto-update capabilities, that is deeply integrated with OS X native virtualization.

WINDOWS

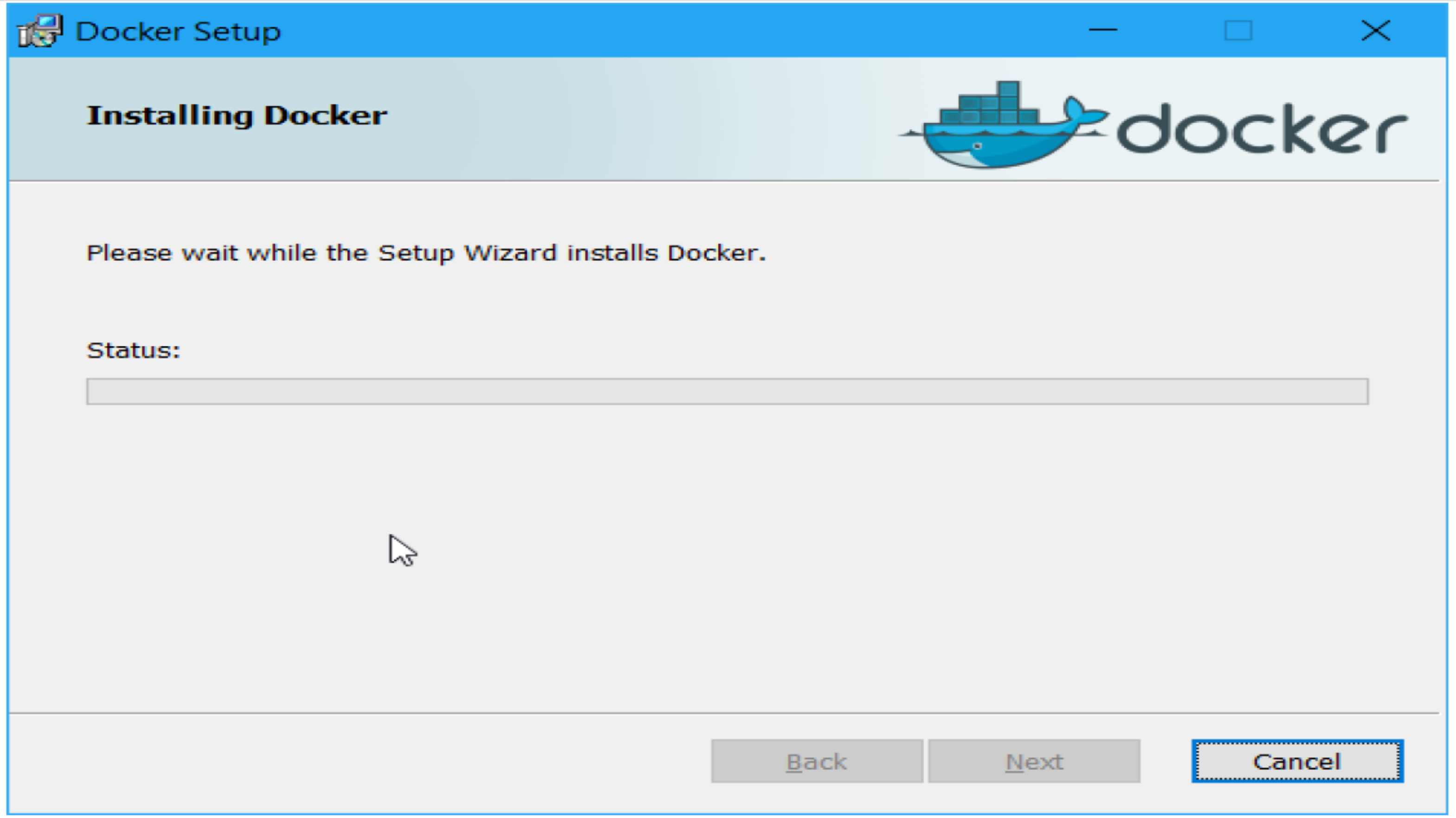
A native Windows application with a user interface and auto-update capabilities, that is deeply integrated with Windows native virtualization.

LINUX

Install Docker on nodes which have a Linux distribution already installed.

DOCKER CAN RUN UNDER WINDOWS 10

The user executes the MSI file and follows the steps while making sure the latest checkbox is ticked so Docker is launched after the setup is finished.



Docker Setup

Installing Docker

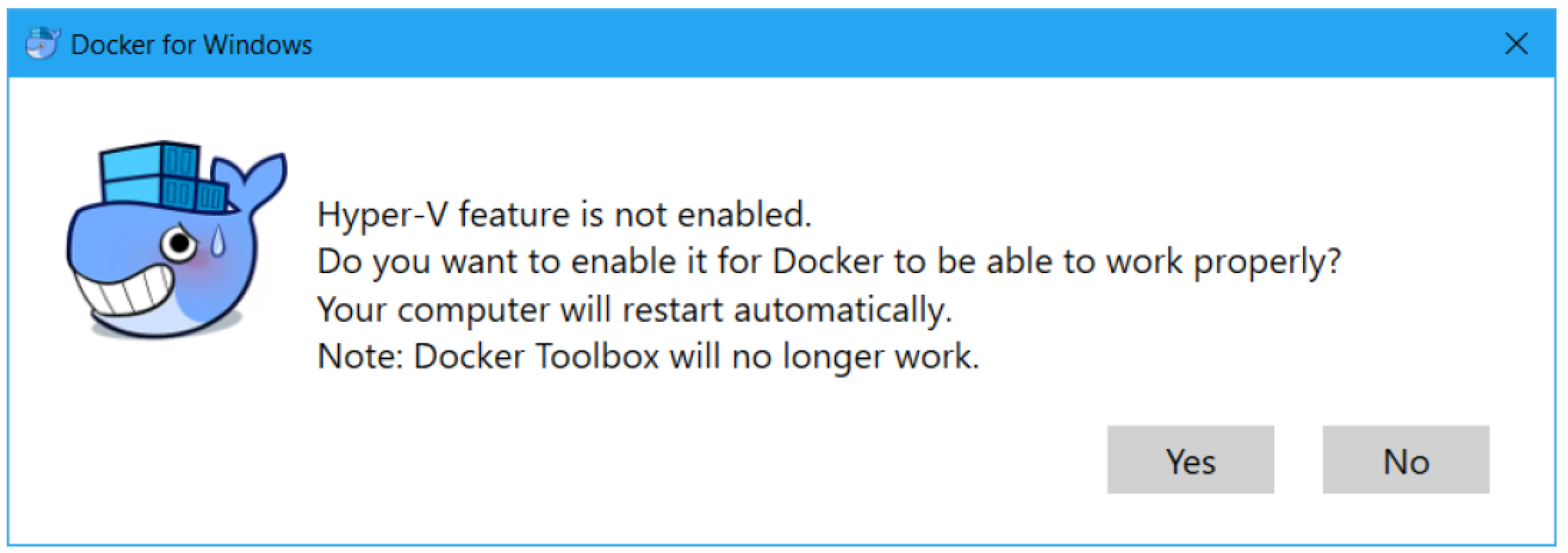
Please wait while the Setup Wizard installs Docker.

Status:

DOCKER CAN RUN UNDER WINDOWS 10

Once Docker is launched for the first time, it detects if Hyper-V, the native hypervisor, is enabled.

If it is not, it requests the user to enable it

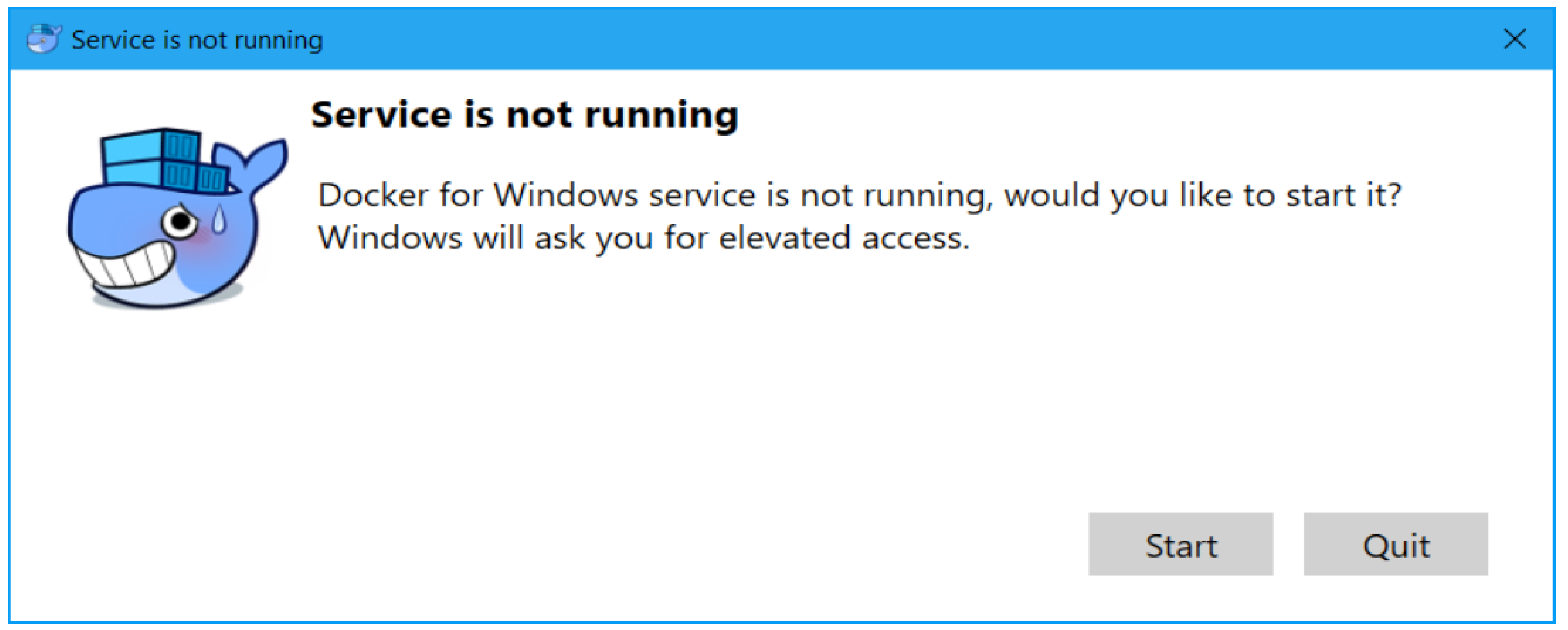


Docker for Windows

Hyper-V feature is not enabled.
Do you want to enable it for Docker to be able to work properly?
Your computer will restart automatically.
Note: Docker Toolbox will no longer work.

DOCKER CAN RUN UNDER WINDOWS 10

After reboot, Windows automatically launches Docker. The first time that this happens, it prompts the user to enable the Docker service.



Service is not running

Service is not running

Docker for Windows service is not running, would you like to start it?
Windows will ask you for elevated access.

DOCKER CAN RUN UNDER WINDOWS 10

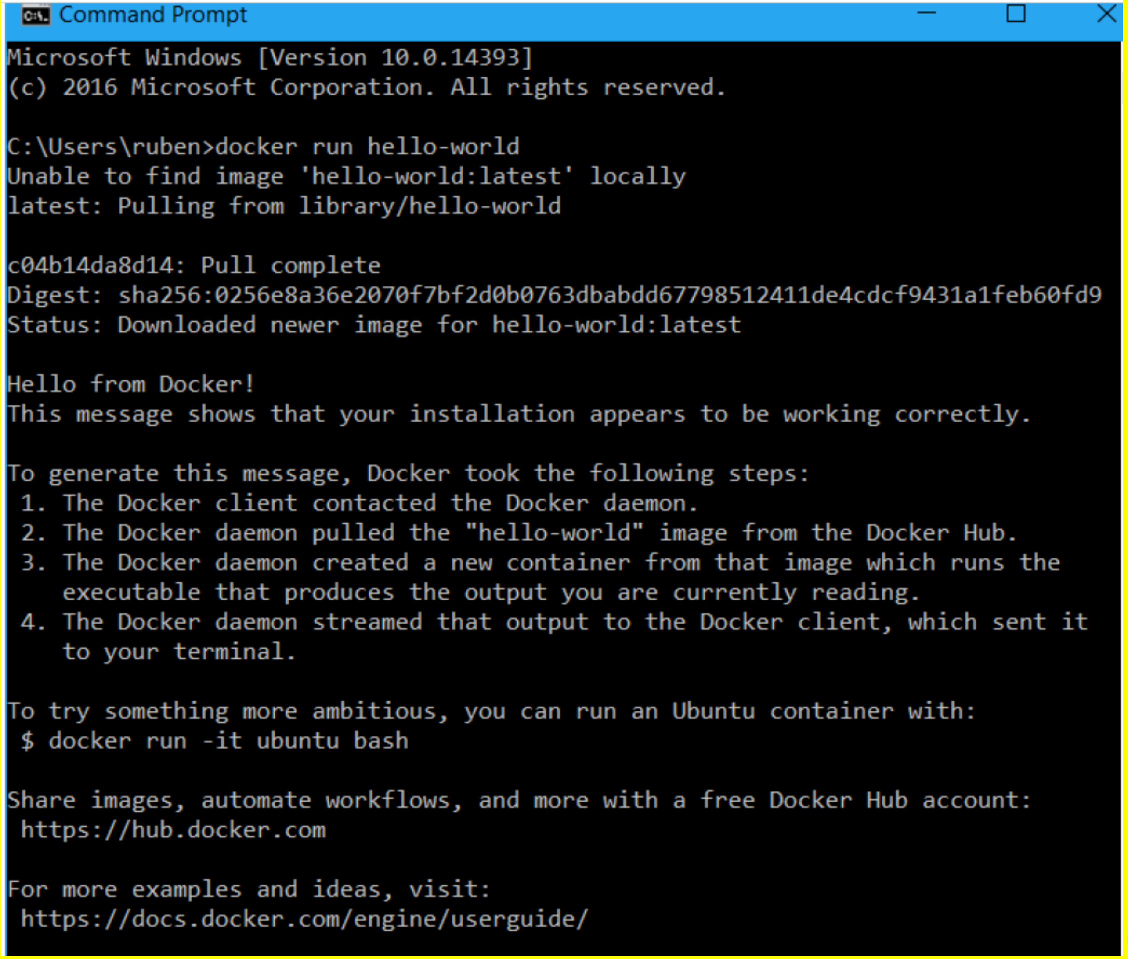
You can verify that Docker is working by running its first container through the Windows Command Prompt command "docker run hello-world".

The command calls the Docker command line interface tool to reach the Docker daemon and tries to find the container “hello-world” locally, which can not be found, so it pulls it from the registry (assuming there is internet connectivity).

After all the layers have been pulled successfully, the container runs and shows an output.

Check if the application is running in a browser by entering http://localhost:8080

Issuing “**docker run hello-world**” at Windows command prompt invokes the Docker CLI and pass the arguments to it.



```
Microsoft Windows [Version 10.0.14393]
(c) 2016 Microsoft Corporation. All rights reserved.

C:\Users\ruben>docker run hello-world
Unable to find image 'hello-world:latest' locally
latest: Pulling from library/hello-world

c04b14da8d14: Pull complete
Digest: sha256:0256e8a36e2070f7bf2d0b0763dbabdd67798512411de4cdcf9431a1feb60fd9
Status: Downloaded newer image for hello-world:latest

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

To generate this message, Docker took the following steps:
 1. The Docker client contacted the Docker daemon.
 2. The Docker daemon pulled the "hello-world" image from the Docker Hub.
 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 Hub account:
 https://hub.docker.com

For more examples and ideas, visit:
 https://docs.docker.com/engine/userguide/
```

DOCKER BUILD

use Docker COPY for code that's on the same machine as where you're building your image.

COPY hello.py /

Docker build has many options but it's often faster to just build and tag an image at the same time although this can be done separately if you prefer. The syntax to build and tag an image is is as follows:

docker build -t your_dockerhub_username/image_name:tag .

The . indicates that you want to build in the current working directory, -t flag to indicate that you want to tag the image at the same time you do a build. Get help:

Docker help

Docker help cp

DOCKER COMMANDS

To login to a Container Registry Enterprise Edition instance

```
docker login container_label
```

It will prompt for username and password to login. Login password is set in the "Access Credential page".

Use docker pull to download a copy of container instance, docker images to view downloaded images

```
docker pull container_label
```

```
docker images
```

Push images to a Container Registry Enterprise Edition instance using

```
docker push container_label
```

to check the status of running docker jobs

```
docker ps -l
```

DOCKER COMMANDS CONT'D

stats returns a data stream of resource utilization by container instance

```
docker stats
```

To monitor resource usage continuously:

```
docker stats --no-stream=true
```

Inspect command outputs info in JSON format, and contains detailed description of the container and its properties.

```
docker inspect container_label
```

To display the list of processes and the information about them that are running inside the container

```
docker vote container_label
```

to print the logs of the application inside the container

```
docker logs container_label
```

DOCKER COMMANDS CONT'D

Can copy files and directories from host to container and vice-versa

```
docker cp filename container_name:/subdir
cd subdir
ls
```

From container to Host

```
docker cp container_name:/subdir/filename  .
```