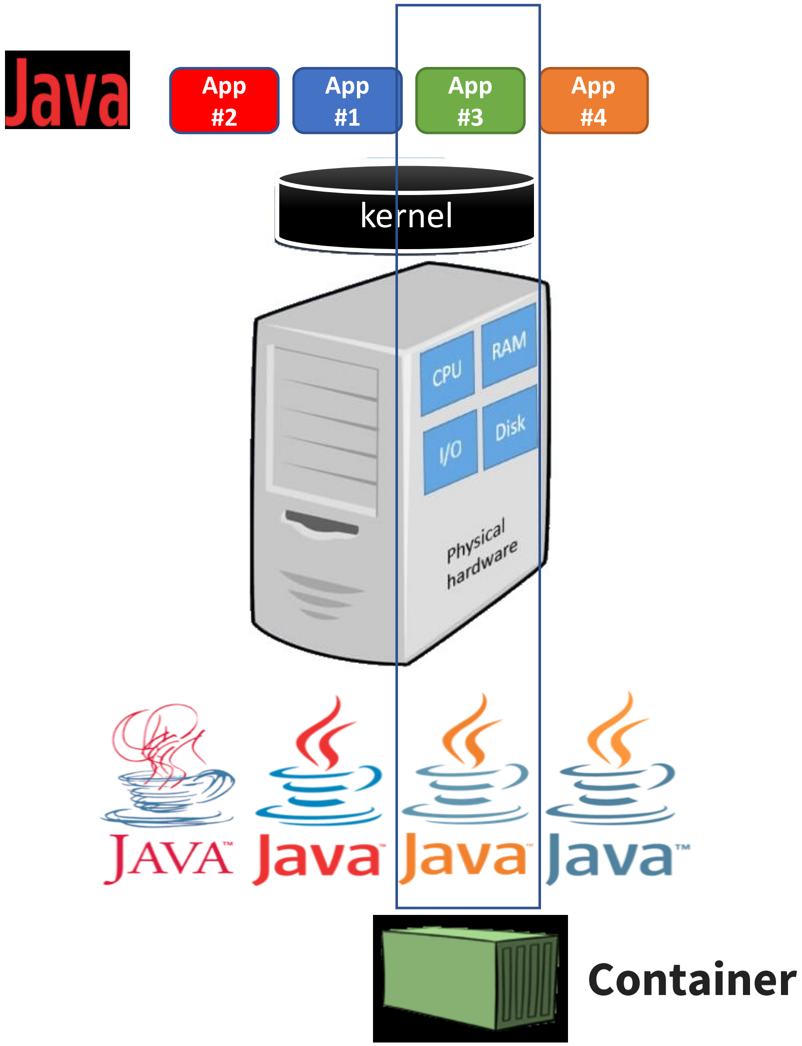
Cloud-native is the **Destination.**

Ziv Dai Cloud expert

zivd@il.ibm.com

What are containers and why do you need them?



OS level virtualization

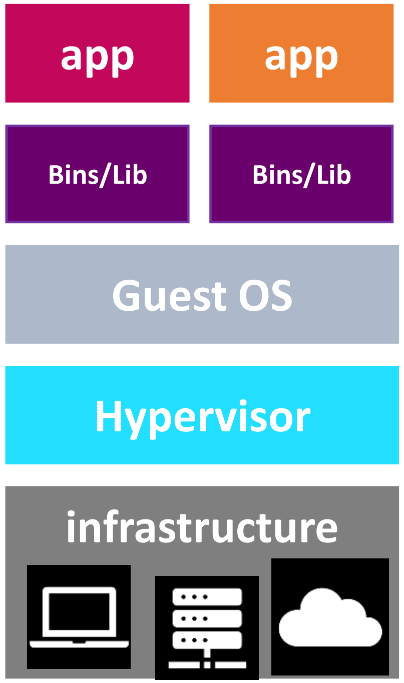
- Isolate user space instances on a single host machine
- Gives isolated view of processes, user space and file system for user owner
- Share host Linux kernel
- A self-contained sealed unit of software
- Contains everything required to run the code

Build **Ship** **Run**

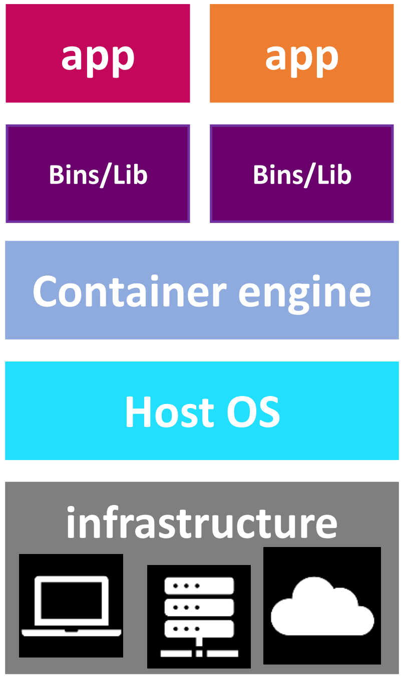
Virtual Machines vs Containerization

Container is computer software used for Virtualization in order to have multiple Operating systems running on the same host.

Container Virtualization is performed on system-level is Docker containers.

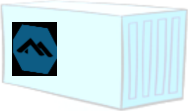


Virtual Machine

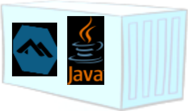


Container

Container Architecture



Container Image The "source code" for our containers; we use them to build containers.



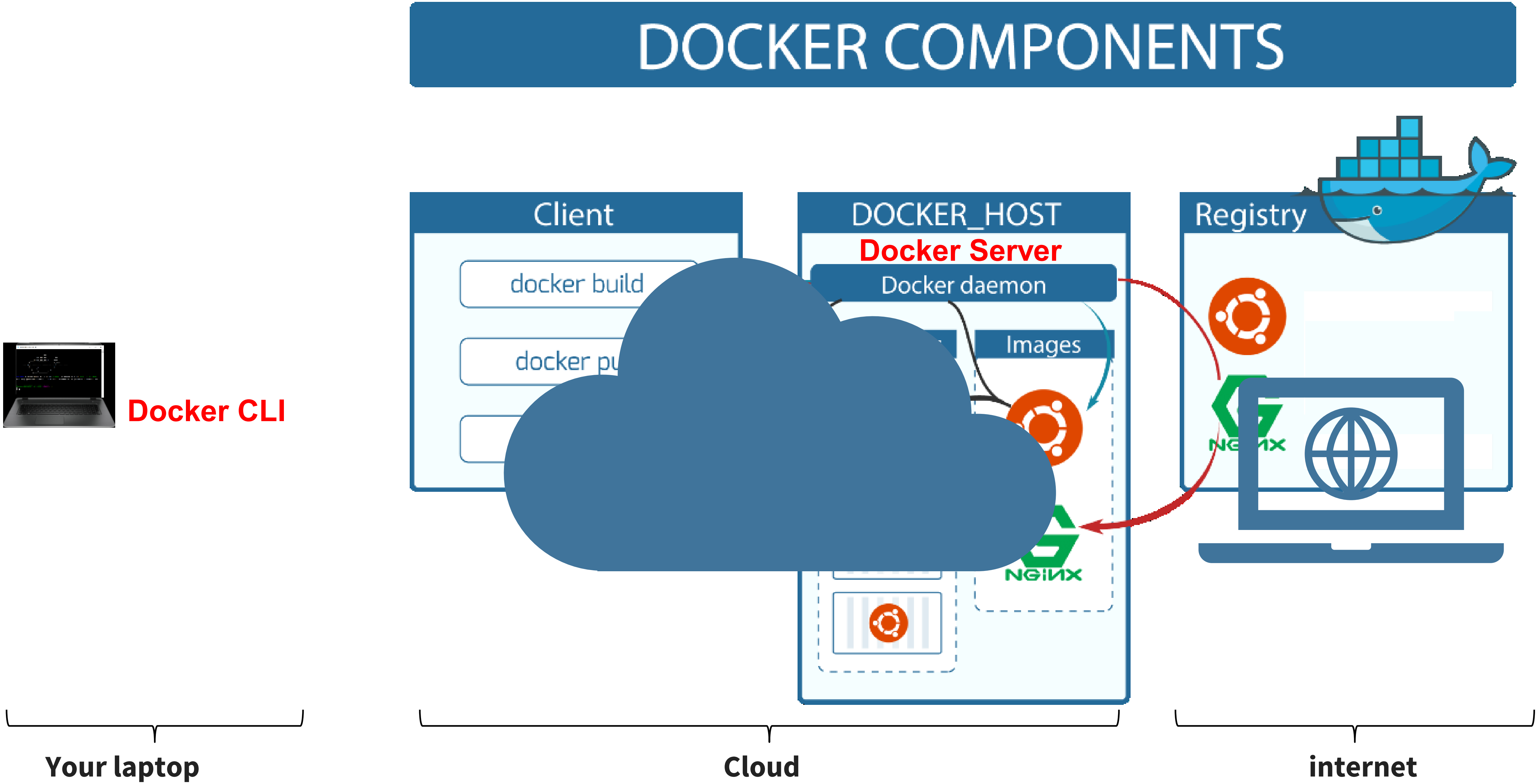
Containers Containers can be ship, modify, manage, create or destroy

In simple terms, an image is a template, and a container is a copy of that template. You can have multiple containers (copies) of the same image.

Container Registry Container images are stored in registries. There are public and private registries.

Container Engine Docker is the client-server type of application. The dockerd is the Docker engine which represents the server. The docker daemon and the clients can be run on the same or remote host, and they communicate through command line

Lab containers architecture



DOCKER COMPONENTS

Docker CLI

Client

docker build

docker pu

DOCKER_HOST

Docker Server

Docker daemon

Images

Registry

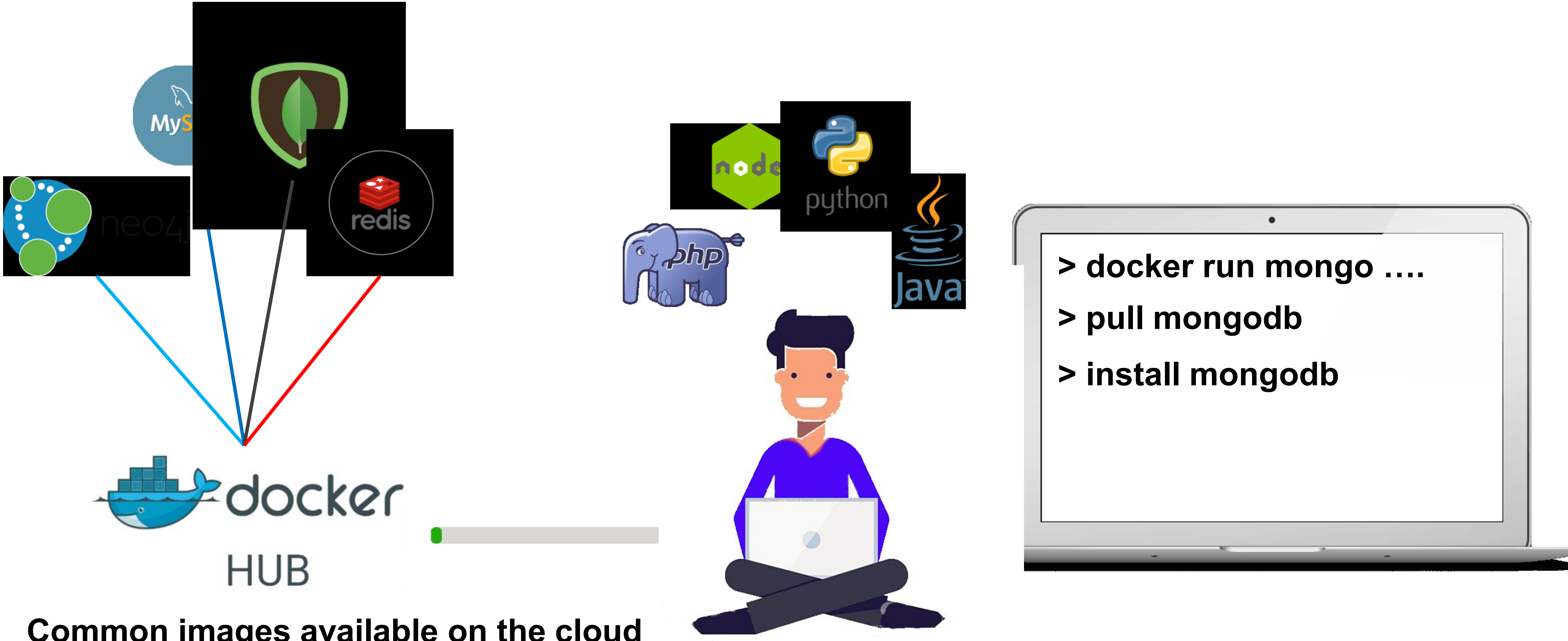
Your laptop

Cloud

internet

Container :

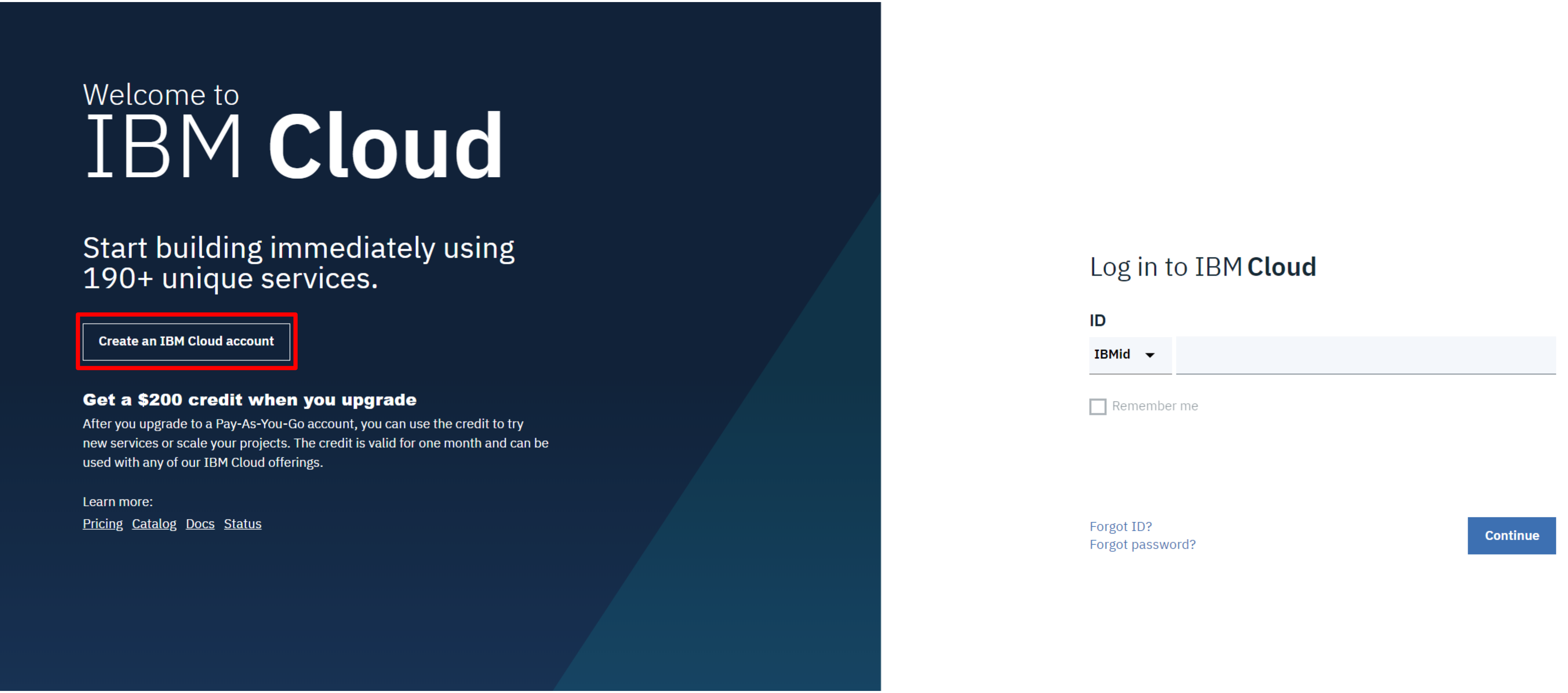
Docker makes it really easy to install and run software without worrying about setup or dependencies



```
> docker run mongo ….
> pull mongodb
> install mongodb
```

Common images available on the cloud

Create IBM ID



Welcome to

IBM **Cloud**

Start building immediately using 190+ unique services.

Create an IBM Cloud account

Get a $200 credit when you upgrade

After you upgrade to a Pay-As-You-Go account, you can use the credit to try new services or scale your projects. The credit is valid for one month and can be used with any of our IBM Cloud offerings.

Learn more:

Pricing Catalog Docs Status

Log in to IBM **Cloud**

ID

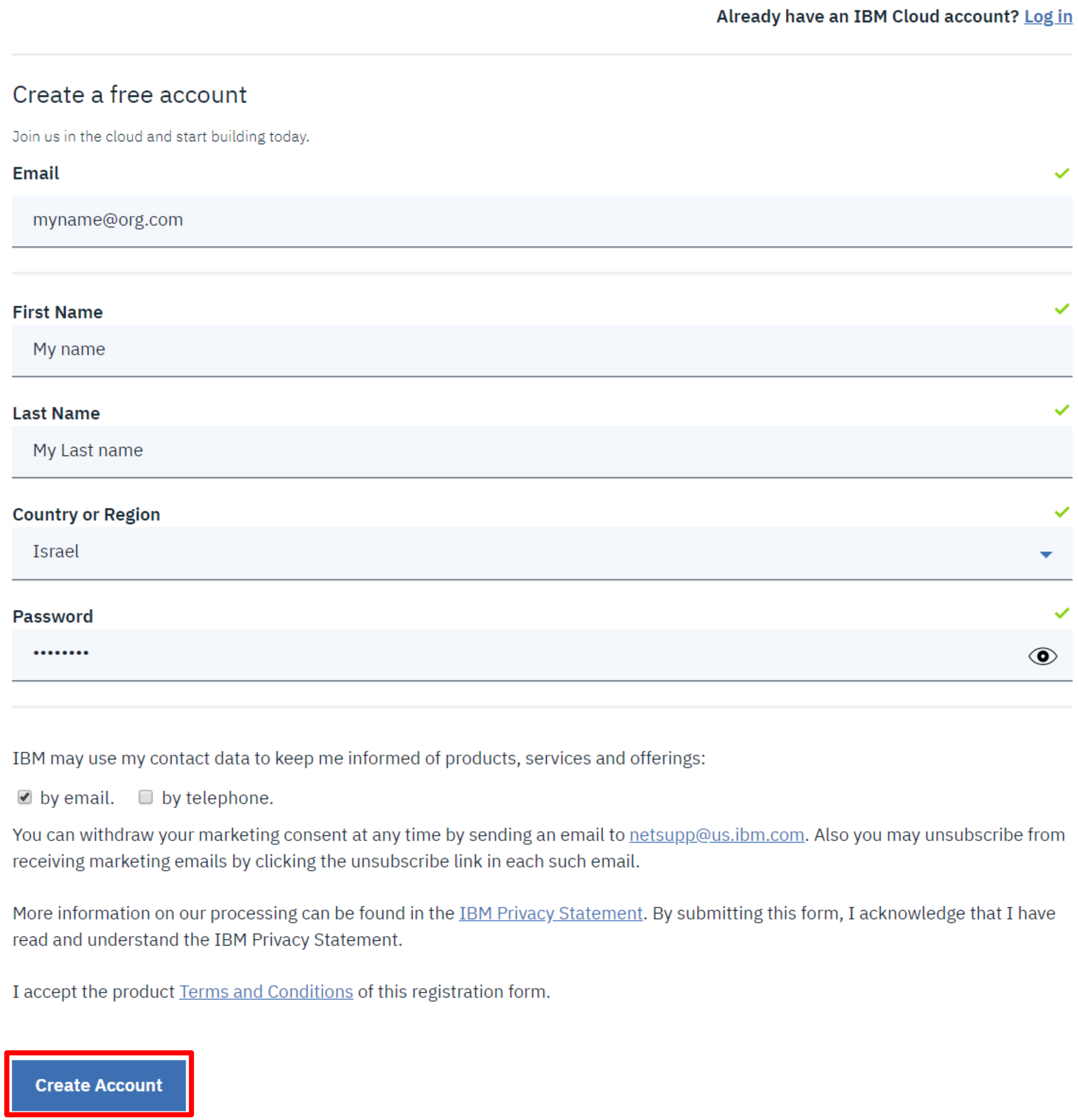
IBMid ▾

□ Remember me

Forgot ID?
Forgot password?

Continue

https://ibm.biz/BdqBhs



Already have an IBM Cloud account? Log in

Create a free account

Join us in the cloud and start building today.

Email ✓

myname@org.com

First Name ✓

My name

Last Name ✓

My Last name

Country or Region ✓

Israel

Password ✓

••••••••

IBM may use my contact data to keep me informed of products, services and offerings:

☑ by email. □ by telephone.

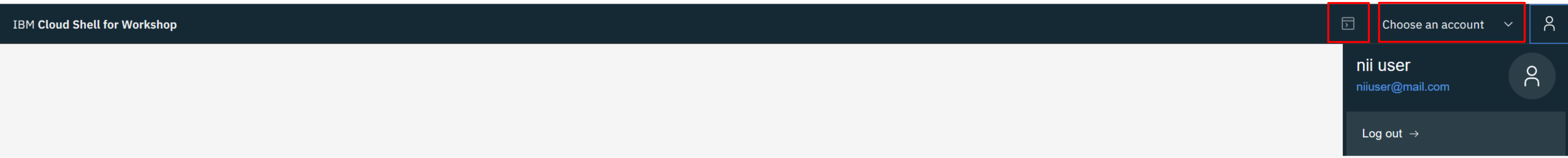
You can withdraw your marketing consent at any time by sending an email to netsupp@us.ibm.com. Also you may unsubscribe from receiving marketing emails by clicking the unsubscribe link in each such email.

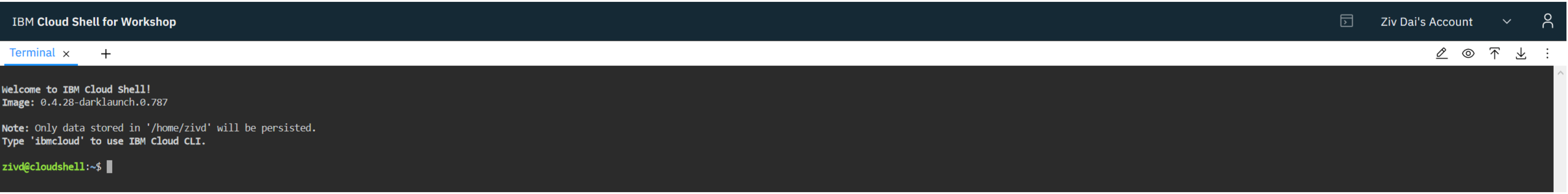
More information on our processing can be found in the IBM Privacy Statement. By submitting this form, I acknowledge that I have read and understand the IBM Privacy Statement.

I accept the product Terms and Conditions of this registration form.

https://workshop.shell.cloud.ibm.com

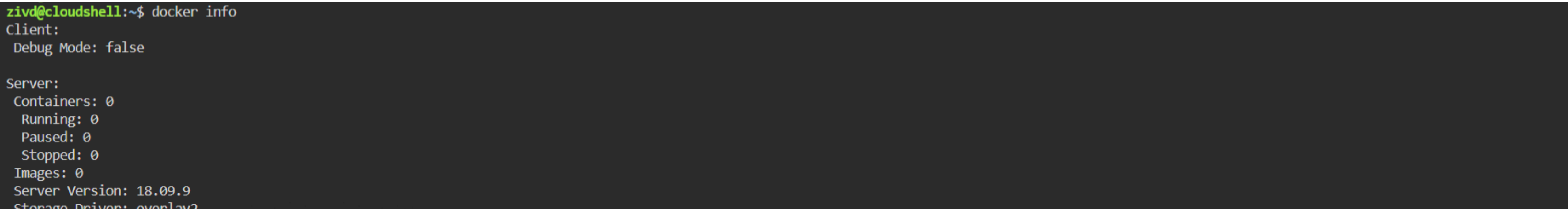
Password : ikslab





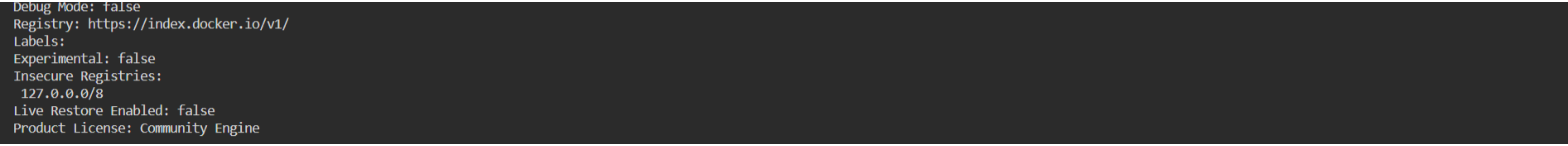
```
Welcome to IBM Cloud Shell!
Image: 0.4.28-darklaunch.0.787

Note: Only data stored in '/home/zivd' will be persisted.
Type 'ibmcloud' to use IBM Cloud CLI.

zivd@cloudshell:~$
```

docker info

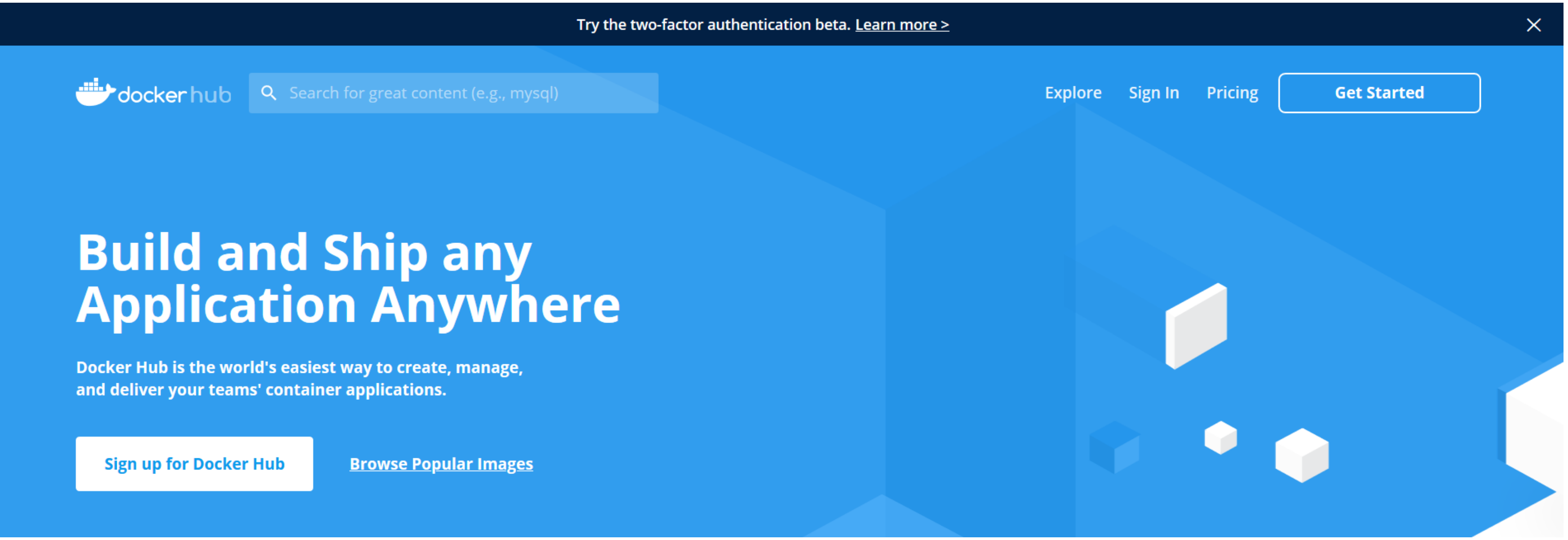


```
zivd@cloudshell:~$ docker info
Client:
 Debug Mode: false

Server:
 Containers: 0
  Running: 0
  Paused: 0
  Stopped: 0
 Images: 0
 Server Version: 18.09.9
 Storage Driver: overlay2
```

Container registry : https://index.docker.io



```
 Debug Mode: false
 Registry: https://index.docker.io/v1/
 Labels:
 Experimental: false
 Insecure Registries:
  127.0.0.0/8
 Live Restore Enabled: false
 Product License: Community Engine
```



Try the two-factor authentication beta. Learn more >

Build and Ship any Application Anywhere

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

Essential Docker Commands

docker images - list images

docker ps - lists running containers

docker ps -a - lists running and history containers

docker pull - pulls (copies) container image from repository (ie: docker hub)

docker run - run a docker container

docker logs - display logs of a container

docker rm - remove one or more containers

docker rmi - remove one or more images

docker stop - stops one or more containers

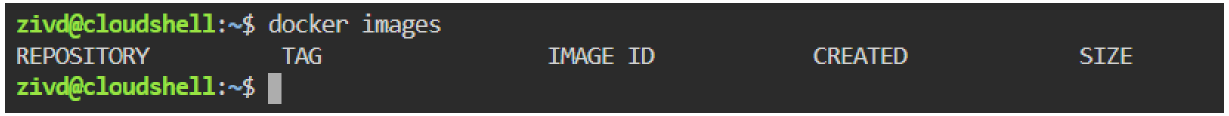
USAGE

This command will download the hello-world Docker image from the Docker hub, if not present already, and run it
The docker run command first creates a writeable container layer over the specified image, and then starts it.

```
$ docker run [OPTIONS] IMAGE [COMMAND] [ARG...]
```

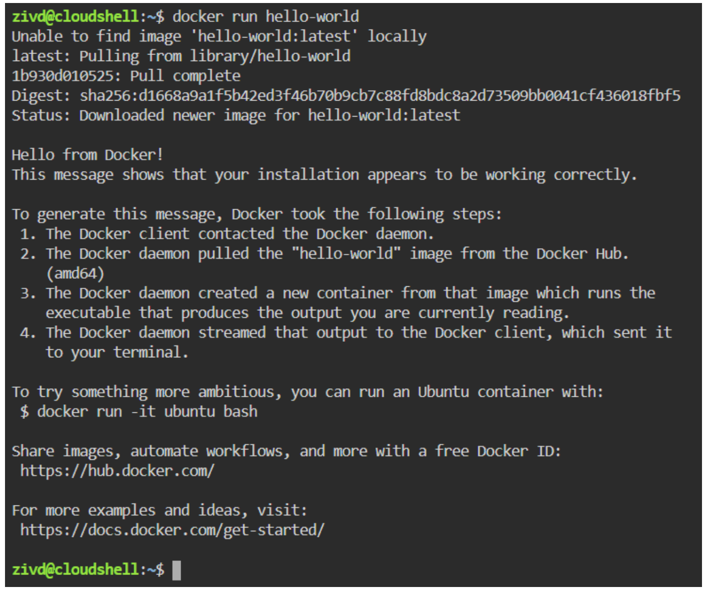
1. Run hello world

```
docker images
```

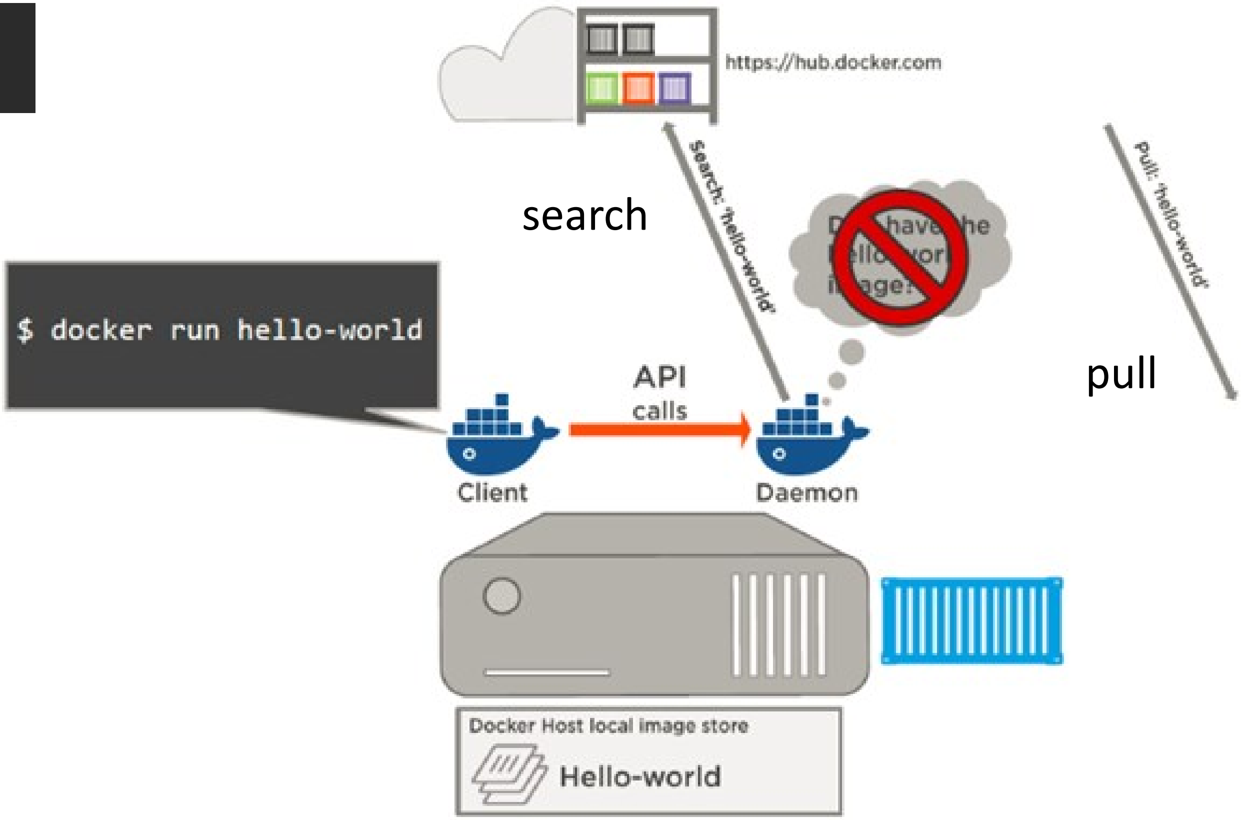


```
zivd@cloudshell:~$ docker images
REPOSITORY          TAG                 IMAGE ID            CREATED             SIZE
zivd@cloudshell:~$ 
```

```
docker run hello-world
```



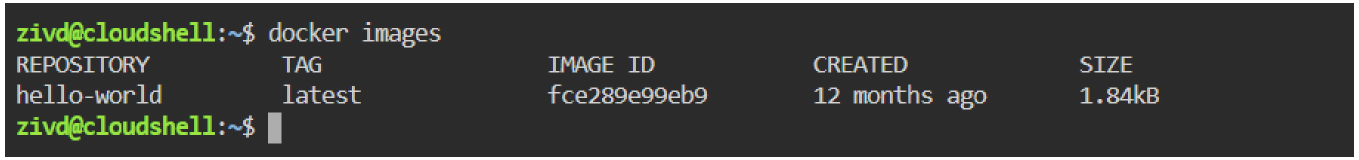
```
zivd@cloudshell:~$ docker run hello-world
Unable to find image 'hello-world:latest' locally
latest: Pulling from library/hello-world
1b930d010525: Pull complete
Digest: sha256:d1668a9a1f5b42ed3f46b70b9cb7c88fd8bdc8a2d73509bb0041cf436018fbf5
Status: Downloaded newer image for hello-world:latest

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.
    (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://hub.docker.com/

For more examples and ideas, visit:
 https://docs.docker.com/get-started/

zivd@cloudshell:~$ 
```



The default docker images will show all top level images, their repository and tags, and their size.

```
$ docker images [OPTIONS] [REPOSITORY[:TAG]]
```

docker images

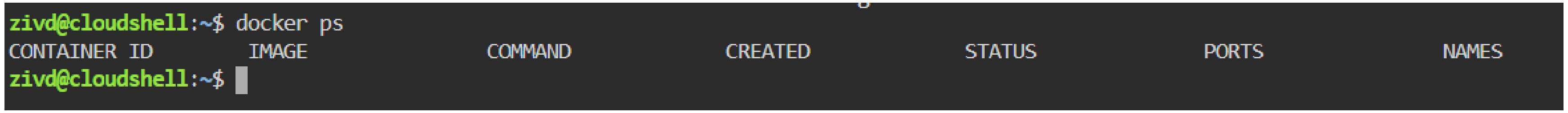


```
zivd@cloudshell:~$ docker images
REPOSITORY          TAG                 IMAGE ID            CREATED             SIZE
hello-world         latest              fce289e99eb9        12 months ago       1.84kB
zivd@cloudshell:~$
```

The docker ps command only shows running containers by default. To see all containers.

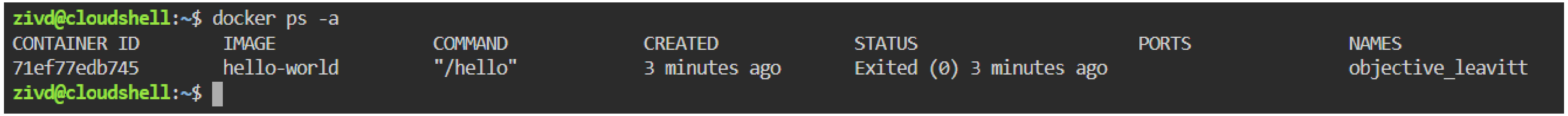
```
$ docker ps [OPTIONS]
```

docker ps



```
zivd@cloudshell:~$ docker ps
CONTAINER ID        IMAGE               COMMAND             CREATED             STATUS              PORTS               NAMES
zivd@cloudshell:~$
```

docker ps -a

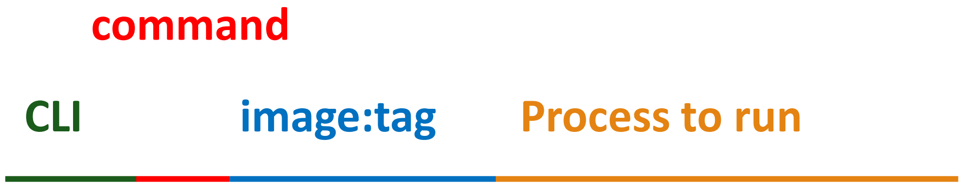


```
zivd@cloudshell:~$ docker ps -a
CONTAINER ID        IMAGE               COMMAND             CREATED             STATUS                      PORTS               NAMES
71ef77edb745        hello-world         "/hello"            3 minutes ago       Exited (0) 3 minutes ago                        objective_leavitt
zivd@cloudshell:~$
```

2. Create hello world image

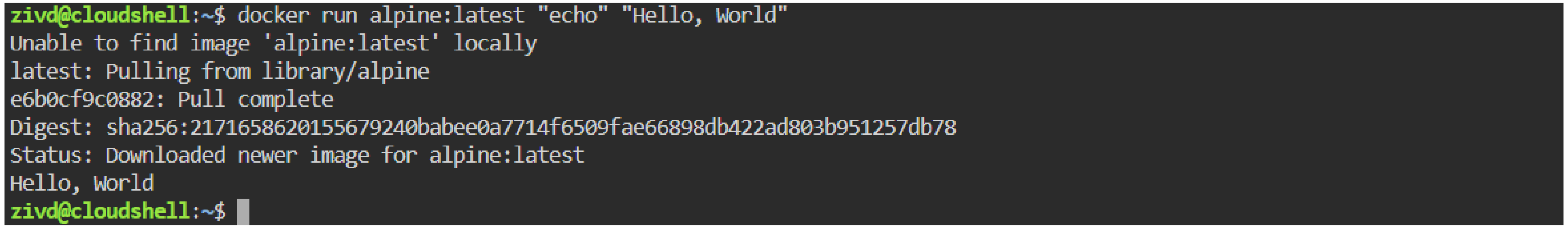
Get a “Hello, world” Printed from Another Basic Docker Image

This command downloads the Alpine base image the first time and creates a Docker **container**. It then runs the container and **executes** the **echo command**. The echo command echoes the “Hello, World” string.



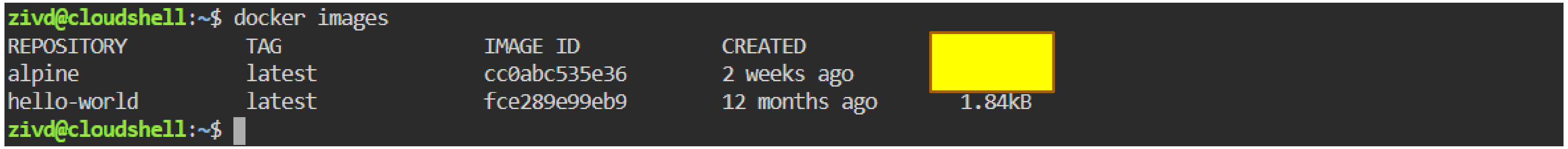
```
docker run alpine:latest "echo" "Hello, World"
```

All of that happened pretty quickly. Imagine booting up a container, running a command and then killing it.



```
zivd@cloudshell:~$ docker run alpine:latest "echo" "Hello, World"
Unable to find image 'alpine:latest' locally
latest: Pulling from library/alpine
e6b0cf9c0882: Pull complete
Digest: sha256:2171658620155679240babee0a7714f6509fae66898db422ad803b951257db78
Status: Downloaded newer image for alpine:latest
Hello, World
zivd@cloudshell:~$
```

docker images

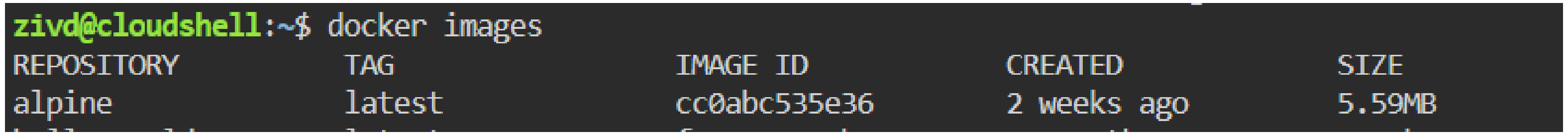


```
zivd@cloudshell:~$ docker images
REPOSITORY          TAG                 IMAGE ID            CREATED             [redacted]
alpine              latest              cc0abc535e36        2 weeks ago
hello-world         latest              fce289e99eb9        12 months ago         1.84kB
zivd@cloudshell:~$
```

only 5 MB size!

http://gliderlabs.viewdocs.io/docker-alpine/

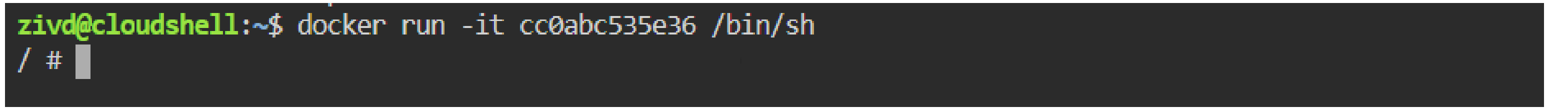
docker images



```
zivd@cloudshell:~$ docker images
REPOSITORY          TAG                 IMAGE ID            CREATED             SIZE
alpine              latest              cc0abc535e36        2 weeks ago         5.59MB
```

Run image as container using alpine image id, enter container bash to Linux commands

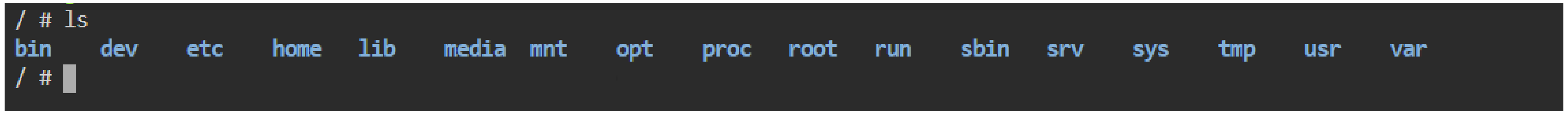
docker run -it 196d12cf6ab1 /bin/sh



```
zivd@cloudshell:~$ docker run -it cc0abc535e36 /bin/sh
/ #
```

listing the contents of container directory

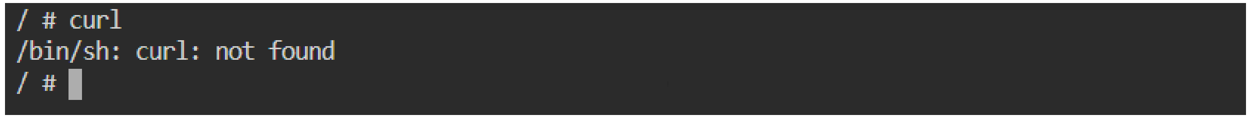
ls



```
/ # ls
bin    dev    etc    home   lib    media  mnt    opt    proc   root   run    sbin   srv    sys    tmp    usr    var
/ #
```

Try to run ‘curl’ command, alpine Linux required curl installation

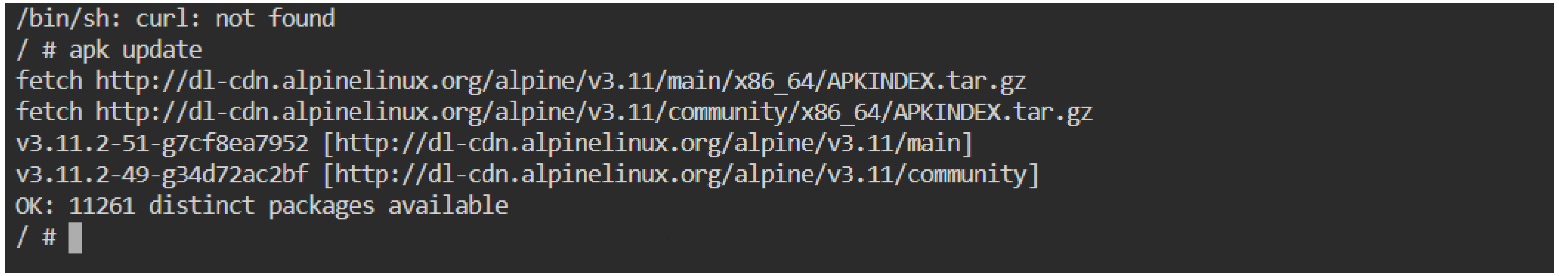
curl



```
/ # curl
/bin/sh: curl: not found
/ #
```

Update alpine software on a running system.

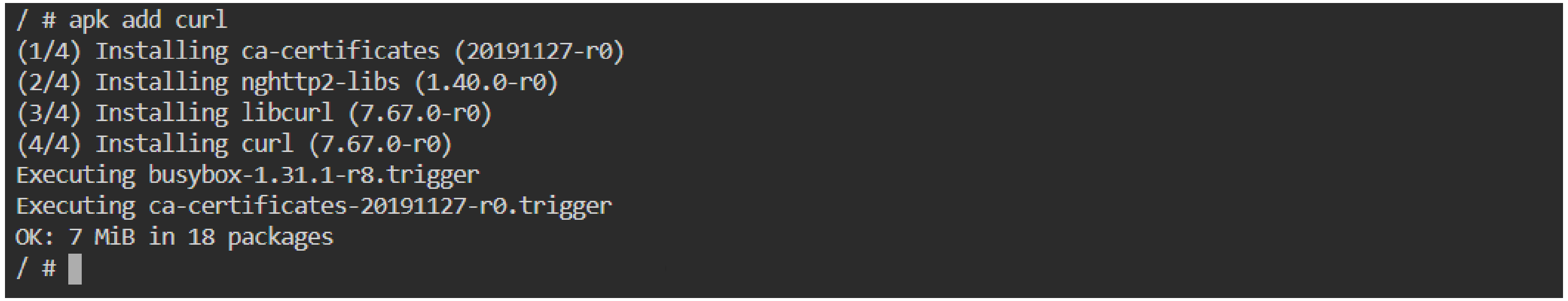
apk update



```
/bin/sh: curl: not found
/ # apk update
fetch http://dl-cdn.alpinelinux.org/alpine/v3.11/main/x86_64/APKINDEX.tar.gz
fetch http://dl-cdn.alpinelinux.org/alpine/v3.11/community/x86_64/APKINDEX.tar.gz
v3.11.2-51-g7cf8ea7952 [http://dl-cdn.alpinelinux.org/alpine/v3.11/main]
v3.11.2-49-g34d72ac2bf [http://dl-cdn.alpinelinux.org/alpine/v3.11/community]
OK: 11261 distinct packages available
/ #
```

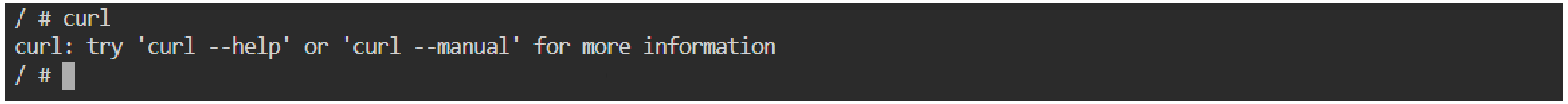
Install curl on alpine Linux

apk add curl



```
/ # apk add curl
(1/4) Installing ca-certificates (20191127-r0)
(2/4) Installing nghttp2-libs (1.40.0-r0)
(3/4) Installing libcurl (7.67.0-r0)
(4/4) Installing curl (7.67.0-r0)
Executing busybox-1.31.1-r8.trigger
Executing ca-certificates-20191127-r0.trigger
OK: 7 MiB in 18 packages
/ #
```

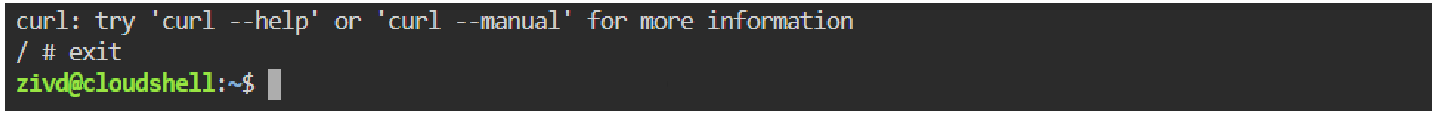
run curl



```
/ # curl
curl: try 'curl --help' or 'curl --manual' for more information
/ #
```

Exist alpine bash

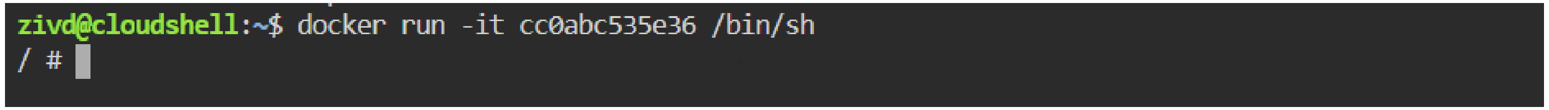
```
exit
```



```
curl: try 'curl --help' or 'curl --manual' for more information
/ # exit
zivd@cloudshell:~$ 
```

Run image as container using alpine image id, enter container bash to Linux commands

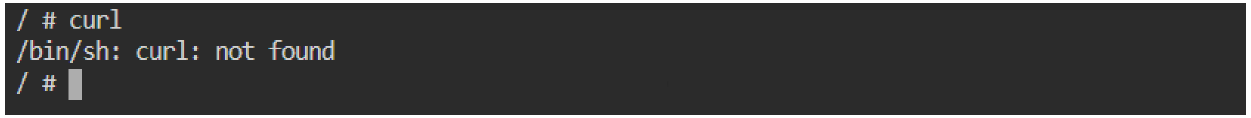
```
docker run -it 196d12cf6ab1 /bin/sh
```



```
zivd@cloudshell:~$ docker run -it cc0abc535e36 /bin/sh
/ # 
```

Try again to run ‘curl’ command, alpine Linux required curl installation !!!

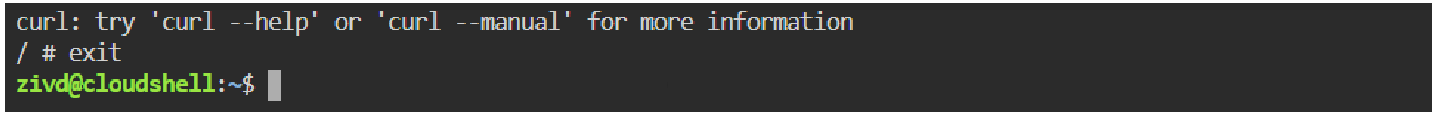
```
curl
```



```
/ # curl
/bin/sh: curl: not found
/ # 
```

Exist alpine bash

```
exit
```



```
curl: try 'curl --help' or 'curl --manual' for more information
/ # exit
zivd@cloudshell:~$ 
```

Container changes do not saved , this is why we are about to use Dockerfile to add and save a RW layer to the container