

## Assignmet no. 2

### AWS EC2 instance

AWS

Services

Resource Groups

Ashish

singapore

Support

1. Choose AMI

2. Choose Instance Type

3. Configure Instance

4. Add Storage

5. Add Tags

6. Configure Security Group

7. Review

Step 1: Choose an Amazon Machine Image (AMI)

Cancel and Exit

AMI is a template that contains the software configuration (operating system, application server, and applications) required to launch your instance. You can select an AMI provided by AWS, our user community, or the AWS Marketplace, or you can select one of your own AMIs.

Q Search for an AMI by entering a search term e.g. "Windows"

X

Quick Start

My AMIs

AWS Marketplace

Community AMIs

Free tier only

Amazon Linux

Free tier eligible

Amazon Linux 2 AMI (HVM), SSD Volume Type

ami-0b419c3a4b01d1850

Amazon Linux 2 comes with five years support. It provides Linux kernel 4.14 tuned for optimal performance on Amazon EC2, systemd 219, GCC 7.3, Glibc 2.26, Binutils 2.29.1, and the latest software packages through extras.

Root device type: ebs

Virtualization type: hvm

Select

64-bit (x86)

Amazon Linux

Free tier eligible

Amazon Linux AMI 2018.03.0 (HVM), SSD Volume Type

ami-05b3bc7f311194b3

The Amazon Linux AMI is an EBS-backed, AWS-supported image. The default image includes AWS command line tools, Python, Ruby, Perl, and Java. The repositories include Docker, PHP, MySQL, PostgreSQL, and other packages.

Root device type: ebs

Virtualization type: hvm

Select

64-bit (x86)

Red Hat

Free tier eligible

Red Hat Enterprise Linux 7.5 (HVM), SSD Volume Type

ami-76144b0a

Red Hat Enterprise Linux version 7.5 (HVM), EBS General Purpose (SSD) Volume Type

Root device type: ebs

Virtualization type: hvm

Select

64-bit (x86)

SUSE Linux

Free tier eligible

SUSE Linux Enterprise Server 15 (HVM), SSD Volume Type

ami-0920c364049458e86

SUSE Linux Enterprise Server 15 (HVM), EBS General Purpose (SSD) Volume Type. Public Cloud, Advanced Systems Management, Web and Scripting, and Legacy modules enabled.

Root device type: ebs

Virtualization type: hvm

Select

64-bit (x86)

1. Choose AMI

2. Choose Instance Type

3. Configure Instance

4. Add Storage

5. Add Tags

6. Configure Security Group

7. Review

Step 3: Configure Instance Details

Purchasing option

Request Spot instances

Network

vpc-ab70edcc | default VPC (default)

Create new VPC

Subnet

No preference (default subnet in any Availability Zone)

Create new subnet

Auto-assign Public IP

Use subnet setting (Enable)

Placement group

Add instance to placement group

Capacity Reservation

Open

Create new Capacity Reservation

IAM role

None

Create new IAM role

Shutdown behavior

Stop

Enable termination protection

Protect against accidental termination

Monitoring

Enable CloudWatch detailed monitoring

Additional charges apply.

Tenancy

Shared - Run a shared hardware instance

Additional charges will apply for dedicated tenancy.

T2/T3 Unlimited

Enable

Additional charges may apply

Advanced Details

Knm

## docker install

login as: ec2-user  
Authenticating with public key "imported-openssh-key"

```
  _ | _ | _ |  
  _ | ( _ | /  
  _ | \ _ | _ |  
  _ | _ | _ |  
Amazon Linux 2 AMI
```

<https://aws.amazon.com/amazon-linux-2/>  
1 package(s) needed for security, out of 3 available  
Run "sudo yum update" to apply all updates.  
[ec2-user@ip-172-31-23-50 ~]\$

```
> Running transaction check
-> Package docker.x86_64 0:18.06.1ce-8.amzn2 will be installed
-> Processing Dependency: pigz for package: docker-18.06.1ce-8.amzn2.x86_64
-> Processing Dependency: libcgroup for package: docker-18.06.1ce-8.amzn2.x86_64
-> Processing Dependency: libltdl.so.7()(64bit) for package: docker-18.06.1ce-8.amzn2.x86_64
-> Running transaction check
-> Package libcgroup.x86_64 0:0.41-15.amzn2 will be installed
-> Package libtool-ltdl.x86_64 0:2.4.2-22.2.amzn2.0.2 will be installed
-> Package pigz.x86_64 0:2.3.4-1.amzn2.0.1 will be installed
-> Finished Dependency Resolution
```

Dependencies Resolved

Package	Arch	Version	Repository
Installing:			
docker	x86_64	18.06.1ce-8.amzn2	amzn2extra-docker
Installing for dependencies:			
libcgroup	x86_64	0.41-15.amzn2	amzn2-core
libtool-ltdl	x86_64	2.4.2-22.2.amzn2.0.2	amzn2-core
pigz	x86_64	2.3.4-1.amzn2.0.1	amzn2-core

Transaction Summary

Install 1 Package (+3 Dependent packages)

Total download size: 37 M

Installed size: 151 M

Is this ok [y/d/N]:

```

Redirecting to /bin/systemctl start docker.service
root@ip-172-31-23-50 ec2-user]#
root@ip-172-31-23-50 ec2-user]#
root@ip-172-31-23-50 ec2-user]# service docker status
Redirecting to /bin/systemctl status docker.service
● docker.service - Docker Application Container Engine
   Loaded: loaded (/usr/lib/systemd/system/docker.service; disabled; vendor preset: disabled)
   Active: active (running) since Fri 2019-03-29 14:21:18 UTC; 10s ago
     Docs: https://docs.docker.com
  Process: 3572 ExecStartPre=/usr/libexec/docker/docker-setup-runtimes.sh (code=exited, status=0/SUCCESS)
 Main PID: 3584 (dockerd)
    Tasks: 18
   Memory: 36.7M
   CGroup: /system.slice/docker.service
           └─3584 /usr/bin/dockerd --default-ulimit nfile=1024:4096
             └─3594 docker-containerd --config /var/run/docker/containerd/containerd.toml

lar 29 14:21:17 ip-172-31-23-50.ap-southeast-1.compute.internal dockerd[3584]: time="2019-03-29T14:21:17.123456789Z" level=info msg="Starting Docker Application Container Engine"
lar 29 14:21:17 ip-172-31-23-50.ap-southeast-1.compute.internal dockerd[3584]: time="2019-03-29T14:21:17.123456789Z" level=info msg="Starting Docker Application Container Engine"
lar 29 14:21:17 ip-172-31-23-50.ap-southeast-1.compute.internal dockerd[3584]: time="2019-03-29T14:21:17.123456789Z" level=info msg="Starting Docker Application Container Engine"
lar 29 14:21:17 ip-172-31-23-50.ap-southeast-1.compute.internal dockerd[3584]: time="2019-03-29T14:21:17.123456789Z" level=info msg="Starting Docker Application Container Engine"
lar 29 14:21:17 ip-172-31-23-50.ap-southeast-1.compute.internal dockerd[3584]: time="2019-03-29T14:21:17.123456789Z" level=info msg="Starting Docker Application Container Engine"
lar 29 14:21:18 ip-172-31-23-50.ap-southeast-1.compute.internal dockerd[3584]: time="2019-03-29T14:21:18.123456789Z" level=info msg="Starting Docker Application Container Engine"
lar 29 14:21:18 ip-172-31-23-50.ap-southeast-1.compute.internal dockerd[3584]: time="2019-03-29T14:21:18.123456789Z" level=info msg="Starting Docker Application Container Engine"
lar 29 14:21:18 ip-172-31-23-50.ap-southeast-1.compute.internal dockerd[3584]: time="2019-03-29T14:21:18.123456789Z" level=info msg="Starting Docker Application Container Engine"
lar 29 14:21:18 ip-172-31-23-50.ap-southeast-1.compute.internal dockerd[3584]: time="2019-03-29T14:21:18.123456789Z" level=info msg="Starting Docker Application Container Engine"
lar 29 14:21:18 ip-172-31-23-50.ap-southeast-1.compute.internal dockerd[3584]: time="2019-03-29T14:21:18.123456789Z" level=info msg="Starting Docker Application Container Engine"
lar 29 14:21:18 ip-172-31-23-50.ap-southeast-1.compute.internal systemd[1]: Started Docker Application Container Engine
lar 29 14:21:18 ip-172-31-23-50.ap-southeast-1.compute.internal dockerd[3584]: time="2019-03-29T14:21:18.123456789Z" level=info msg="Starting Docker Application Container Engine"
hint: Some lines were ellipsized, use -l to show in full.

```

```

C:\Users\pranav>docker

Usage: docker [OPTIONS] COMMAND

A self-sufficient runtime for containers

Options:
  --config string      Location of client config files (default
                        "C:\Users\pranav\.docker")
  -c, --context string  Name of the context to use to connect to the
                        daemon (overrides DOCKER_HOST env var and
                        default context set with "docker context use")
  -D, --debug           Enable debug mode
  -H, --host list       Daemon socket(s) to connect to
  -l, --log-level string Set the logging level
                        ("debug"|"info"|"warn"|"error"|"fatal")
                        (default "info")
  --tls                Use TLS; implied by --tlsverify
  --tlscacert string    Trust certs signed only by this CA (default
                        "C:\Users\pranav\.docker\ca.pem")
  --tlscert string      Path to TLS certificate file (default
                        "C:\Users\pranav\.docker\cert.pem")
  --tlskey string       Path to TLS key file (default
                        "C:\Users\pranav\.docker\key.pem")
  --tlsverify           Use TLS and verify the remote
  -v, --version         Print version information and quit

```

Docker --version

```

stats      Display a live stream of container(s) resource usage statistics
stop       Stop one or more running containers
tag        Create a tag TARGET_IMAGE that refers to SOURCE_IMAGE
top        Display the running processes of a container
unpause    Unpause all processes within one or more containers
update     Update configuration of one or more containers
version    Show the Docker version information
wait       Block until one or more containers stop, then print their exit codes

```

Run 'docker COMMAND --help' for more information on a command.

To get more help with docker, check out our guides at <https://docs.docker.com/go/guides/>

```

C:\Users\pranav> docker --version
Docker version 20.10.7, build f0df350

```

```

C:\Users\pranav>

```

## Docker start:

```

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/>

```

C:\Users\pranav>docker ps
CONTAINER ID   IMAGE      COMMAND                  CREATED    STATUS    PORTS    NAMES
C:\Users\pranav>docker ps -a
CONTAINER ID   IMAGE      COMMAND                  CREATED    STATUS    PORTS    NAMES
372ff665691a   hello-world  "/hello"                About a minute ago   Exited (0) About a minute ago   nruuffled_gauss
4846e3fd9bf8   ubuntu:latest  "bash"                  25 hours ago        Exited (255) 3 hours ago
ad-tom
d36492550db8   ubuntu     "bash"                  25 hours ago        Exited (0) 25 hours ago
doring_satoshi
9bbbce87c248   ubuntu:latest  "bash"                  26 hours ago        Exited (255) 3 hours ago
ervent_jones
28059bacc2c    hello-world  "/hello"                26 hours ago        Exited (0) 26 hours ago
olly_cohen
37a8667f4ebb   alpine/git    "git clone https://g..." 3 weeks ago         Exited (0) 3 weeks ago
epo

```

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

C:\Users\pranav>docker ps
CONTAINER ID   IMAGE      COMMAND                  CREATED        STATUS        PORTS        NAMES
C:\Users\pranav>docker ps -a
CONTAINER ID   IMAGE      COMMAND                  CREATED        STATUS        PORTS        NAMES
72ff665691a    hello-world  "/hello"                About a minute ago    Exited (0)    About a minute ago
ruffled_gauss  846e3fd9bf8  ubuntu:latest "bash"          25 hours ago        Exited (255)  3 hours ago
sad-tom        36492550db8  ubuntu      "bash"          25 hours ago        Exited (0)    25 hours ago
doring_satoshi 9bbce87c248  ubuntu:latest "bash"          26 hours ago        Exited (255)  3 hours ago
ervent_jones    8059bacc2c    hello-world  "/hello"        26 hours ago        Exited (0)    26 hours ago
olly_cohen      37a8667f4ebb  alpine/git    "git clone https://g..." 3 weeks ago        Exited (0)    3 weeks ago
epo

C:\Users\pranav>docker start sad-tom
sad-tom

C:\Users\pranav>
```

## Docker stop:

```
https://docs.docker.com/get-started/

C:\Users\pranav>docker ps
CONTAINER ID   IMAGE      COMMAND                  CREATED        STATUS        PORTS        NAMES
C:\Users\pranav>docker ps -a
CONTAINER ID   IMAGE      COMMAND                  CREATED        STATUS        PORTS        NAMES
72ff665691a    hello-world  "/hello"                About a minute ago    Exited (0)    About a minute ago
ruffled_gauss  846e3fd9bf8  ubuntu:latest "bash"          25 hours ago        Exited (255)  3 hours ago
sad-tom        36492550db8  ubuntu      "bash"          25 hours ago        Exited (0)    25 hours ago
doring_satoshi 9bbce87c248  ubuntu:latest "bash"          26 hours ago        Exited (255)  3 hours ago
ervent_jones    8059bacc2c    hello-world  "/hello"        26 hours ago        Exited (0)    26 hours ago
olly_cohen      37a8667f4ebb  alpine/git    "git clone https://g..." 3 weeks ago        Exited (0)    3 weeks ago
epo

C:\Users\pranav>docker start sad-tom
sad-tom

C:\Users\pranav>docker stop sad-tom
sad-tom

C:\Users\pranav>
```

Docker stats:

CONTAINER ID	NAME	CPU %	MEM USAGE / LIMIT	MEM %	NET I/O	BLOCK I/O	PIDS
4846e3fd9bf8	sad-tom	0.00%	0B / 0B	0.00%	0B / 0B	0B / 0B	0

Docker run hello-world

```
C:\Users\pranav>\docker -ps
'\docker' is not recognized as an internal or external command,
operable program or batch file.

C:\Users\pranav>docker run hello-world

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/

C:\Users\pranav>
```



## Docker inspect

```
b877a23b86c8 ubuntu "bash" 4 minutes ago Exited (127) 15 seconds ago affectio
nate_antonelli
872ff665691a hello-world "/hello" 8 minutes ago Exited (0) 8 minutes ago unruffle
d_gauss
4846e3fd9bf8 ubuntu:latest "bash" 25 hours ago Exited (0) 5 minutes ago sad-tom
436492550db8 ubuntu "bash" 25 hours ago Exited (0) 25 hours ago adoring_
satoshi
9bbbce87c248 ubuntu:latest "bash" 26 hours ago Exited (255) 3 hours ago fervent_
jones
28059baccc2c hello-world "/hello" 26 hours ago Exited (0) 26 hours ago jolly_co
nen
87a8667f4ebb alpine/git "git clone https://g--" 3 weeks ago Exited (0) 3 weeks ago repo

C:\Users\pranav>docker volume create
f22a235587c2649a85e15035d31939e18f6b68b1451bfe60c598a13f8f151e4b

C:\Users\pranav>docker volume inspect f22a235587c2649a85e15035d31939e18f6b68b1451bfe60c598a13f8f151e4b
{
  "CreatedAt": "2021-08-18T14:17:25Z",
  "Driver": "local",
  "Labels": {},
  "Mountpoint": "/var/lib/docker/volumes/f22a235587c2649a85e15035d31939e18f6b68b1451bfe60c598a13f8f151e4b/_data",
  "Name": "f22a235587c2649a85e15035d31939e18f6b68b1451bfe60c598a13f8f151e4b",
  "Options": {},
  "Scope": "local"
}

C:\Users\pranav>
```

## Docker rm

```
C:\Users\pranav>docker volume rm f22a235587c2649a85e15035d31939e18f6b68b1451bfe60c598a13f8f151e4b
f22a235587c2649a85e15035d31939e18f6b68b1451bfe60c598a13f8f151e4b

C:\Users\pranav>
```

## Docker run -it ubuntu bash

```
CONTAINER ID   NAME      CPU %     MEM USAGE / LIMIT   MEM %     NET I/O   BLOCK I/O   PIDS
4846e3fd9bf8   sad-tom   0.00%     0B / 0B              0.00%     0B / 0B   0B / 0B     0

C:\Users\pranav>d
'd' is not recognized as an internal or external command,
operable program or batch file.

C:\Users\pranav>docker run -it ubuntu bash
root@b877a23b86c8:/#
```

## Working with volumes part:

```
C:\Users\pranav>docker volume ls
DRIVER      VOLUME NAME
local       4c66d8ff8bf0e17224e14f367c50a0ba26f3c484dcedcbb9d055d6ad7acee3a0
```

```
C:\Users\pranav>
```

```
C:\Users\pranav>docker ps -a
```

CONTAINER ID	IMAGE	COMMAND	CREATED	STATUS	PORTS	NAMES
b877a23b86c8	ubuntu	"bash"	4 minutes ago	Exited (127) 15 seconds ago		affection
nate_antonelli	hello-world	"/hello"	8 minutes ago	Exited (0) 8 minutes ago		unruffle
d_gauss	ubuntu:latest	"bash"	25 hours ago	Exited (0) 5 minutes ago		sad-tom
d36492550db8	ubuntu	"bash"	25 hours ago	Exited (0) 25 hours ago		adoring_
satoshi	ubuntu:latest	"bash"	26 hours ago	Exited (255) 3 hours ago		fervent_
9bbbce87c248	hello-world	"/hello"	26 hours ago	Exited (0) 26 hours ago		jolly_co
hen	alpine/git	"git clone https://g--"	3 weeks ago	Exited (0) 3 weeks ago		repo

```
C:\Users\pranav>docker volume create
f22a235587c2649a85e15035d31939e18f6b68b1451bfe60c598a13f8f151e4b
```

```
C:\Users\pranav>
```

operable program or batch file.

```
C:\Users\pranav>docker stop
"docker stop" requires at least 1 argument.
See 'docker stop --help'.
```

```
Usage: docker stop [OPTIONS] CONTAINER [CONTAINER...]
```

Stop one or more running containers

```
C:\Users\pranav>
```