

EC2 Dashboard

Events

Tags

Limits

Instances

Instances New

Instance Types

Launch Templates

Spot Requests

Savings Plans

Reserved Instances New

Dedicated Hosts

Capacity Reservations

Instances (2) Info



Connect

Instance state

Actions

Launch instances

Filter instances

< 1 >

<input type="checkbox"/>	Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone
<input type="checkbox"/>	-	i-0aec3391b09328c19	Terminated	t2.micro	-	No alarms	us-east-1a
<input type="checkbox"/>	Docker	i-0b1b67df4eb59a1be	Running	t2.micro	2/2 checks passed	No alarms	us-east-1a

← → ↻ <https://us-east-2.console.aws.amazon.com/ec2/v2/connect/ec2-user/i-0b1b67df4eb59a>

 Amazon Linux 2 AMI

<https://aws.amazon.com/amazon-linux-2/>

1 package(s) needed for security, out of 16 available

Run "sudo yum update" to apply all updates.

[ec2-user@ip-172-31-33-170 ~]\$ sudo yum update

Loaded plugins: extras_suggestions, langpacks, priorities, update-motd

Resolving Dependencies

--> Running transaction check

--> Package curl.x86_64 0:7.61.1-12.amzn2.0.4 will be updated

--> Package curl.x86_64 0:7.76.1-4.amzn2.0.1 will be an update

← → ↻ <https://us-east-2.console.aws.amazon.com/ec2/v2/connect/ec2-user/i-0b1b67df4eb59a1be>



```
Verifying : ec2-utils-1.2-44.amzn2.noarch
Verifying : 1:grub2-tools-2.06-2.amzn2.0.1.x86_64
Verifying : 1:grub2-2.06-2.amzn2.0.1.x86_64
Verifying : systemtap-runtime-4.4-1.amzn2.0.1.x86_64
Verifying : 1:grub2-tools-minimal-2.06-2.amzn2.0.1.x86_64
Verifying : curl-7.61.1-12.amzn2.0.4.x86_64
Verifying : 1:grub2-pc-2.06-2.amzn2.0.1.x86_64
Verifying : kernel-tools-4.14.238-182.422.amzn2.x86_64
Verifying : libcurl-7.61.1-12.amzn2.0.4.x86_64
Verifying : 1:grub2-pc-modules-2.06-2.amzn2.0.1.noarch
Verifying : grubby-8.28-23.amzn2.0.1.x86_64
```

Installed:

```
grub2.x86_64 1:2.06-2.amzn2.0.3          grub2-pc.x86_64 1:2.06-2.amzn2.0.3      grub2-tools.x86_64 1:2.06-2.amzn2.0.3
grub2-tools-efi.x86_64 1:2.06-2.amzn2.0.3  grub2-tools-extra.x86_64 1:2.06-2.amzn2.0.3  grub2-tools-minimal.x86_64 1:2.06-2.amzn2.0.3
kernel.x86_64 0:4.14.241-184.433.amzn2
```

Updated:

```
curl.x86_64 0:7.76.1-4.amzn2.0.1      ec2-utils.noarch 0:1.2-45.amzn2          grub2-common.noarch 1:2.06-2.amzn2.0.1
grub2-efi-x64-ec2.x86_64 1:2.06-2.amzn2.0.3  grub2-pc-modules.noarch 1:2.06-2.amzn2.0.3  grubby.x86_64 0:8.28-23.amzn2.0.1
kernel-tools.x86_64 0:4.14.241-184.433.amzn2  libcurl.x86_64 0:7.76.1-4.amzn2.0.1      systemtap-runtime.x86_64 1:4.4-1.amzn2.0.1
```

Replaced:

```
grub2.x86_64 1:2.06-2.amzn2.0.1      grub2-tools.x86_64 1:2.06-2.amzn2.0.1
```

Complete!

```
[ec2-user@ip-172-31-33-170 ~]$
```

← → ↻ <https://us-east-2.console.aws.amazon.com/ec2/v2/connect/ec2-user/i-0b1b67df4eb59a1be>

kernel.x86_64 0:4.14.241-184.433.amzn2

Updated:

curl.x86_64 0:7.76.1-4.amzn2.0.1	ec2-utils.noarch 0:1.2-45.amzn2	grub2-common.noarch 1:
grub2-efi-x64-ec2.x86_64 1:2.06-2.amzn2.0.3	grub2-pc-modules.noarch 1:2.06-2.amzn2.0.3	grubby.x86_64 0:8.28-2
kernel-tools.x86_64 0:4.14.241-184.433.amzn2	libcurl.x86_64 0:7.76.1-4.amzn2.0.1	systemtap-runtime.x86_64 1:1.15.1-1.amzn2.0.1

Replaced:

grub2.x86_64 1:2.06-2.amzn2.0.1	grub2-tools.x86_64 1:2.06-2.amzn2.0.1
---------------------------------	---------------------------------------

Complete!

[ec2-user@ip-172-31-33-170 ~]\$ sudo yum install docker

Loaded plugins: extras_suggestions, langpacks, priorities, update-motd

Resolving Dependencies

--> Running transaction check

--> Package docker.x86_64 0:20.10.4-1.amzn2 will be installed

--> Processing Dependency: runc >= 1.0.0 for package: docker-20.10.4-1.amzn2.x86_64

--> Processing Dependency: libcgrouper >= 0.40.rc1-5.15 for package: docker-20.10.4-1.amzn2.x86_64

--> Processing Dependency: containerd >= 1.3.2 for package: docker-20.10.4-1.amzn2.x86_64

--> Processing Dependency: pigz for package: docker-20.10.4-1.amzn2.x86_64

--> Running transaction check

--> Package containerd.x86_64 0:1.4.6-2.amzn2 will be installed

--> Package libcgrouper.x86_64 0:0.41-21.amzn2 will be installed

--> Package pigz.x86_64 0:2.3.4-1.amzn2.0.1 will be installed

--> Package runc.x86_64 0:1.0.0-1.amzn2 will be installed

```
Total 64 MB/s |
Running transaction check
Running transaction test
Transaction test succeeded
Running transaction
Installing : runc-1.0.0-1.amzn2.x86_64
Installing : containerd-1.4.6-2.amzn2.x86_64
Installing : libcgrou-0.41-21.amzn2.x86_64
Installing : pigz-2.3.4-1.amzn2.0.1.x86_64
Installing : docker-20.10.4-1.amzn2.x86_64
Verifying  : docker-20.10.4-1.amzn2.x86_64
Verifying  : pigz-2.3.4-1.amzn2.0.1.x86_64
Verifying  : libcgrou-0.41-21.amzn2.x86_64
Verifying  : containerd-1.4.6-2.amzn2.x86_64
Verifying  : runc-1.0.0-1.amzn2.x86_64

Installed:
  docker.x86_64 0:20.10.4-1.amzn2

Dependency Installed:
  containerd.x86_64 0:1.4.6-2.amzn2  libcgrou.x86_64 0:0.41-21.amzn2  pigz.x86_64 0:2.3.4-1.amzn2.0.1  runc.x86_64 0:1.0.0-1.amzn2.x86_64

Complete!
[ec2-user@ip-172-31-33-170 ~]$
```

Management Commands:

builder	Manage builds
config	Manage Docker configs
container	Manage containers
context	Manage contexts
image	Manage images
manifest	Manage Docker image manifests and manifest lists
network	Manage networks
node	Manage Swarm nodes
plugin	Manage plugins
secret	Manage Docker secrets
service	Manage services
stack	Manage Docker stacks
swarm	Manage Swarm
system	Manage Docker
trust	Manage trust on Docker images
volume	Manage volumes

Commands:

attach	Attach local standard input, output, and error streams to a running container
build	Build an image from a Dockerfile
commit	Create a new image from a container's changes
cp	Copy files/folders between a container and the local filesystem
create	Create a new container
diff	Inspect changes to files or directories on a container's filesystem
events	Get real time events from the server

restart	Restart one or more containers
rm	Remove one or more containers
rmi	Remove one or more images
run	Run a command in a new container
save	Save one or more images to a tar archive (streamed to STDOUT by default)
search	Search the Docker Hub for images
start	Start one or more stopped containers
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/>

```
[ec2-user@ip-172-31-33-170 ~]$ docker --version
```

```
Docker version 20.10.4, build d73b82c
```

See system logs and `systemctl status docker.service` for details.

```
[ec2-user@ip-172-31-33-170 ~]$ sudo service docker start
```

```
Redirecting to /bin/systemctl start docker.service
```

```
[ec2-user@ip-172-31-33-170 ~]$ sudo 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 Sun 2021-08-15 14:12:54 UTC; 16s ago
```

```
Docs: https://docs.docker.com
```

```
Process: 7963 ExecStartPre=/usr/libexec/docker/docker-setup-runtimes.sh (code=exited, status=0/SUCCESS)
```

```
Process: 7953 ExecStartPre=/bin/mkdir -p /run/docker (code=exited, status=0/SUCCESS)
```

```
Main PID: 7971 (dockerd)
```

```
Tasks: 7
```

```
Memory: 37.6M
```

```
CGroup: /system.slice/docker.service
```

```
└─7971 /usr/bin/dockerd -H fd:// --containerd=/run/containerd/containerd.sock --default-ulimit nofile=1024:
```



```
[ec2-user@ip-172-31-33-170 ~]$ sudo su
[root@ip-172-31-33-170 ec2-user]# docker run hello-world
Unable to find image 'hello-world:latest' locally
latest: Pulling from library/hello-world
b8dfde127a29: Pull complete
Digest: sha256:0fe98d7debd9049c50b597ef1f85b7c1e8cc81f59c8d623fcb2250e8bec85b38
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

```
[root@ip-172-31-33-170 ec2-user]# ^C
[root@ip-172-31-33-170 ec2-user]# docker run -it ubuntu bash
Unable to find image 'ubuntu:latest' locally
latest: Pulling from library/ubuntu
16ec32c2132b: Pull complete
Digest: sha256:82becede498899ec668628e7cb0ad87b6e1c371cb8a1e597d83a47fac21d6af3
Status: Downloaded newer image for ubuntu:latest
oot@52aab9d31a4c: /root@52aab9d31a4c:/# █
```

```
[root@ip-172-31-33-170 ec2-user]# docker info
```

```
Client:
```

```
Context:    default
```

```
Debug Mode: false
```

```
Server:
```

```
Containers: 2
```

```
Running: 0
```

```
Paused: 0
```

```
Stopped: 2
```

```
Images: 2
```

```
Server Version: 20.10.4
```

```
Storage Driver: overlay2
```

```
Backing Filesystem: xfs
```

```
Supports d_type: true
```

```
Native Overlay Diff: true
```

```
Logging Driver: json-file
```

```
Cgroup Driver: cgroupfs
```

```
Cgroup Version: 1
```

```
Plugins:
```

```
Volume: local
```

```
Network: bridge host ipvlan macvlan null overlay
```

```
Log: awslogs fluentd gcplogs gelf journald json-file local logentr
```

```
Swarm: inactive
```

```
Runtimes: io.containerd.runc.v2 io.containerd.runtime.v1.linux runc
```

```
Default Runtime: runc
```

```
Kernel Version: 4.14.238-102.422.amzn2.x86_64
Operating System: Amazon Linux 2
OSType: linux
Architecture: x86_64
CPUs: 1
Total Memory: 983.3MiB
Name: ip-172-31-33-170.us-east-2.compute.internal
ID: L6KD:4WUY:EOVB:QTDU:JATK:EHHH:RYRH:VOMY:MJJK:3ZLW:QCRV:CYT5
Docker Root Dir: /var/lib/docker
Debug Mode: false
Registry: https://index.docker.io/v1/
Labels:
Experimental: false
Insecure Registries:
  127.0.0.0/8
Live Restore Enabled: false
```

```
[root@ip-172-31-33-170 ec2-user]# service docker stop
Redirecting to /bin/systemctl stop docker.service
Warning: Stopping docker.service, but it can still be activated by:
  docker.socket
[root@ip-172-31-33-170 ec2-user]#
```

```
Redirecting to /bin/systemctl stop docker.service
Warning: Stopping docker.service, but it can still be activated by:
  docker.socket
[root@ip-172-31-33-170 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: inactive (dead) since Sun 2021-08-15 14:20:08 UTC; 20s ago
     Docs: https://docs.docker.com
   Process: 7971 ExecStart=/usr/bin/dockerd -H fd:// --containerd=/run/containerd/containerd.sock $OPTIONS $DOCKER_STORAGE_OPTIONS $DOCKER_ADD_RUNTIMES (code=exited, status=0/SUCCESS)
   Process: 7963 ExecStartPre=/usr/libexec/docker/docker-setup-runtimes.sh (code=exited, status=0/SUCCESS)
   Process: 7953 ExecStartPre=/bin/mkdir -p /run/docker (code=exited, status=0/SUCCESS)
   Main PID: 7971 (code=exited, status=0/SUCCESS)

Aug 15 14:12:54 ip-172-31-33-170.us-east-2.compute.internal dockerd[7971]: time="2021-08-15T14:12:54.431740669Z" level=info msg="...10
Aug 15 14:12:54 ip-172-31-33-170.us-east-2.compute.internal dockerd[7971]: time="2021-08-15T14:12:54.432184261Z" level=info msg="...io
Aug 15 14:12:54 ip-172-31-33-170.us-east-2.compute.internal systemd[1]: Started Docker Application Container Engine.
Aug 15 14:12:54 ip-172-31-33-170.us-east-2.compute.internal dockerd[7971]: time="2021-08-15T14:12:54.456217820Z" level=info msg="...ocl
Aug 15 14:15:01 ip-172-31-33-170.us-east-2.compute.internal dockerd[7971]: time="2021-08-15T14:15:01.007254221Z" level=info msg="...et
Aug 15 14:18:38 ip-172-31-33-170.us-east-2.compute.internal dockerd[7971]: time="2021-08-15T14:18:38.104271222Z" level=info msg="...et
Aug 15 14:20:08 ip-172-31-33-170.us-east-2.compute.internal systemd[1]: Stopping Docker Application Container Engine...
Aug 15 14:20:08 ip-172-31-33-170.us-east-2.compute.internal dockerd[7971]: time="2021-08-15T14:20:08.720775722Z" level=info msg="...ed
Aug 15 14:20:08 ip-172-31-33-170.us-east-2.compute.internal dockerd[7971]: time="2021-08-15T14:20:08.722722532Z" level=info msg="...et
Aug 15 14:20:08 ip-172-31-33-170.us-east-2.compute.internal systemd[1]: Stopped Docker Application Container Engine.
Hint: Some lines were ellipsized, use -l to show in full.
[root@ip-172-31-33-170 ec2-user]#
```

```
https://aws.amazon.com/amazon-linux-2/
[ec2-user@ip-172-31-33-170 ~]$ docker --version
Docker version 20.10.4, build d3cb89e
[ec2-user@ip-172-31-33-170 ~]$ docker volume create
Got permission denied while trying to connect to the Docker daemon socket at unix:///var/run/docker.sock: Post "http://%2Fvar%2Frun%2Fdocker.sock/v1.24/volumes/create": dial unix /var/run/docker.sock: connect: permission denied
[ec2-user@ip-172-31-33-170 ~]$ sudo su
[root@ip-172-31-33-170 ec2-user]# docker volume create
ab366ceba4d8a5050fa3a9342adb13898f1c6d64e9b9d73a66f0fd5ec0738f15
[root@ip-172-31-33-170 ec2-user]# docker volume inspect ab366ceba4d8a5050fa3a9342adb13898f1c6d64e9b9d73a66f0fd5ec0738f15
[
  {
    "CreatedAt": "2021-08-15T14:35:58Z",
    "Driver": "local",
    "Labels": {},
    "Mountpoint": "/var/lib/docker/volumes/ab366ceba4d8a5050fa3a9342adb13898f1c6d64e9b9d73a66f0fd5ec0738f15/_data",
    "Name": "ab366ceba4d8a5050fa3a9342adb13898f1c6d64e9b9d73a66f0fd5ec0738f15",
    "Options": {},
    "Scope": "local"
  }
]
[root@ip-172-31-33-170 ec2-user]#
```

```

[ec2-user@ip-172-31-33-170 ~]$ docker --version
Docker version 20.10.4, build d3cb89e
[ec2-user@ip-172-31-33-170 ~]$ docker volume create
Got permission denied while trying to connect to the Docker daemon socket at unix:///var/run/docker.sock: Post "http://%2Fvar%2Fdocker.sock/v1.24/volumes/create": dial unix /var/run/docker.sock: connect: permission denied
[ec2-user@ip-172-31-33-170 ~]$ sudo su
[root@ip-172-31-33-170 ec2-user]# docker volume create
ab366ceba4d8a5050fa3a9342adb13898f1c6d64e9b9d73a66f0fd5ec0738f15
[root@ip-172-31-33-170 ec2-user]# docker volume inspect ab366ceba4d8a5050fa3a9342adb13898f1c6d64e9b9d73a66f0fd5ec0738f15
[
  {
    "CreatedAt": "2021-08-15T14:35:58Z",
    "Driver": "local",
    "Labels": {},
    "Mountpoint": "/var/lib/docker/volumes/ab366ceba4d8a5050fa3a9342adb13898f1c6d64e9b9d73a66f0fd5ec0738f15/_data",
    "Name": "ab366ceba4d8a5050fa3a9342adb13898f1c6d64e9b9d73a66f0fd5ec0738f15",
    "Options": {},
    "Scope": "local"
  }
]
[root@ip-172-31-33-170 ec2-user]# docker volume create
2bfed0d6c23c34ebc25c3371669c885cf2f1102a887e8c4921ee644c10b83168
[root@ip-172-31-33-170 ec2-user]# docker volume ls
DRIVER      VOLUME NAME
local       2bfed0d6c23c34ebc25c3371669c885cf2f1102a887e8c4921ee644c10b83168
local       ab366ceba4d8a5050fa3a9342adb13898f1c6d64e9b9d73a66f0fd5ec0738f15
[root@ip-172-31-33-170 ec2-user]# █

```

```

[ec2-user@ip-172-31-33-170 ~]$ sudo su
[root@ip-172-31-33-170 ec2-user]# docker volume create
ab366ceba4d8a5050fa3a9342adb13898f1c6d64e9b9d73a66f0fd5ec0738f15
[root@ip-172-31-33-170 ec2-user]# docker volume inspect ab366ceba4d8a5050fa3a9342adb13898f1c6d64e9b9d73a66f0fd5ec0738f15
[
  {
    "CreatedAt": "2021-08-15T14:35:58Z",
    "Driver": "local",
    "Labels": {},
    "Mountpoint": "/var/lib/docker/volumes/ab366ceba4d8a5050fa3a9342adb13898f1c6d64e9b9d73a66f0fd5ec0738f15/_data",
    "Name": "ab366ceba4d8a5050fa3a9342adb13898f1c6d64e9b9d73a66f0fd5ec0738f15",
    "Options": {},
    "Scope": "local"
  }
]
[root@ip-172-31-33-170 ec2-user]# docker volume create
2bfed0d6c23c34ebc25c3371669c885cf2f1102a887e8c4921ee644c10b83168
[root@ip-172-31-33-170 ec2-user]# docker volume ls
DRIVER      VOLUME NAME
local       2bfed0d6c23c34ebc25c3371669c885cf2f1102a887e8c4921ee644c10b83168
local       ab366ceba4d8a5050fa3a9342adb13898f1c6d64e9b9d73a66f0fd5ec0738f15
[root@ip-172-31-33-170 ec2-user]# docker volume rm 2bfed0d6c23c34ebc25c3371669c885cf2f1102a887e8c4921ee644c10b83168
2bfed0d6c23c34ebc25c3371669c885cf2f1102a887e8c4921ee644c10b83168

```



```
[root@ip-172-31-33-170 ec2-user]# docker volume ls
DRIVER      VOLUME NAME
local       ab366ceba4d8a5050fa3a9342adb13898f1c6d64e9b9d73a66f0fd5ec0738f15
[root@ip-172-31-33-170 ec2-user]#
```