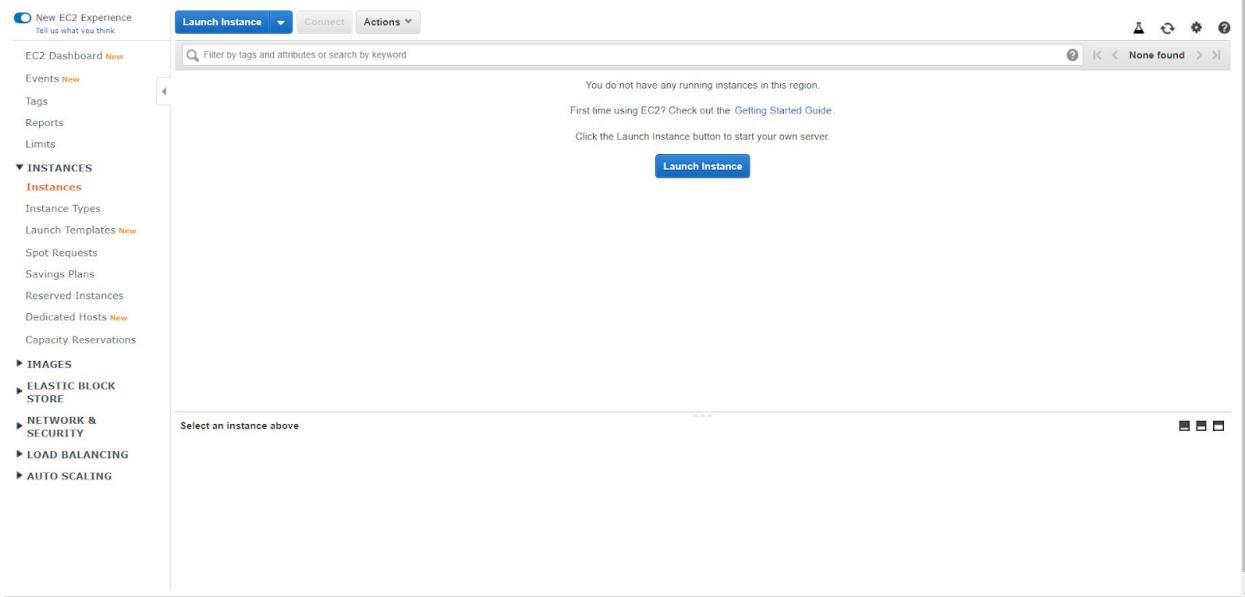
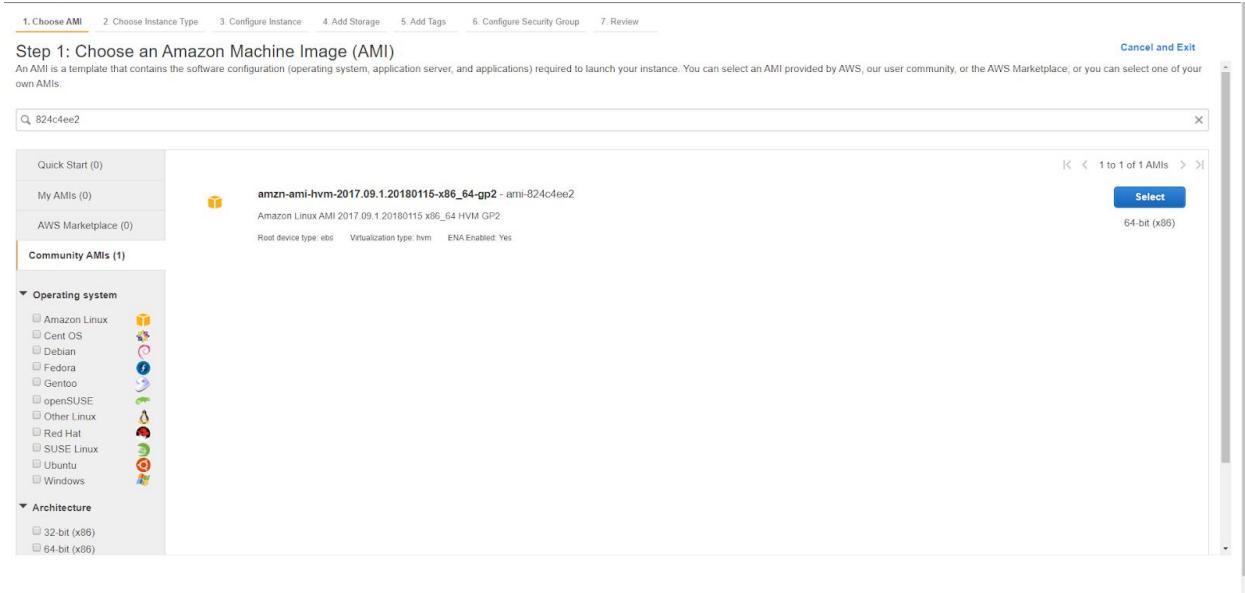


Module 4 Docker & ECS Lab: Create Docker Image & Deploy with ECS



Starting an EC2 instance in N. California



AMI 824c4ee2 is selected

Step 2: Choose an Instance Type

Amazon EC2 provides a wide selection of instance types optimized to fit different use cases. Instances are virtual servers that can run applications. They have varying combinations of CPU, memory, storage, and networking capacity, and give you the flexibility to choose the appropriate mix of resources for your application. Learn more about instance types and how they can meet your computing needs.

Currently selected: t2.micro (Variable ECUs, 1 vCPUs, 2.5 GHz, Intel Xeon Family, 1 GiB memory, EBS only)								
Family	Type	vCPUs	Memory (GiB)	Instance Storage (GB)	EBS-Optimized Available	Network Performance	IPv6 Support	
General purpose	t2.nano	1	0.5	EBS only	-	Low to Moderate	Yes	
General purpose	t2.micro <small>Froze tier eligible</small>	1	1	EBS only	-	Low to Moderate	Yes	
General purpose	t2.small <small>Free tier eligible</small>	1	2	EBS only	-	Low to Moderate	Yes	
General purpose	t2.medium <small>Free tier eligible</small>	2	4	EBS only	-	Low to Moderate	Yes	
General purpose	t2.large <small>Free tier eligible</small>	2	8	EBS only	-	Low to Moderate	Yes	
General purpose	t2.xlarge	4	16	EBS only	-	Moderate	Yes	
General purpose	t2.2xlarge	8	32	EBS only	-	Moderate	Yes	
General purpose	t3a.nano	2	0.5	EBS only	Yes	Up to 5 Gigabit	Yes	
General purpose	t3a.micro	2	1	EBS only	Yes	Up to 5 Gigabit	Yes	
General purpose	t3a.small	2	2	EBS only	Yes	Up to 5 Gigabit	Yes	
General purpose	t3a.medium	2	4	EBS only	Yes	Up to 5 Gigabit	Yes	
General purpose	t3a.large	2	8	EBS only	Yes	Up to 5 Gigabit	Yes	

[Cancel](#) [Previous](#) [Review and Launch](#) [Next: Configure Instance Details](#)

t2.micro Instance Type

Step 3: Configure Instance Details

Configure the instance to suit your requirements. You can launch multiple instances from the same AMI, request Spot instances to take advantage of the lower pricing, assign an access management role to the instance, and more.

Number of instances	<input type="text" value="1"/>	Launch into Auto Scaling Group
Purchasing option	<input type="checkbox"/> Request Spot instances	
Network	<input type="text" value="vpc-56524731 (default)"/>	Create new VPC
Subnet	<input type="text" value="No preference (default subnet in any Availability Zone)"/>	Create new subnet
Auto-assign Public IP	<input type="checkbox"/> Use subnet setting (Enable)	
Placement group	<input type="checkbox"/> Add instance to placement group	
Capacity Reservation	<input type="text" value="Open"/>	Create new Capacity Reservation
IAM role	<input type="text" value="None"/>	
Shutdown behavior	<input type="text" value="Stop"/>	Create new IAM role
Enable termination protection	<input type="checkbox"/> Protect against accidental termination	
Monitoring	<input type="checkbox"/> Enable CloudWatch detailed monitoring Additional charges apply	
Tenancy	<input type="text" value="Shared - Run a shared hardware instance"/> Additional charges will apply for dedicated tenancy.	
T2/T3 Unlimited	<input type="checkbox"/> Enable Additional charges may apply	
File systems	Add file system Create new file system	

[Cancel](#) [Previous](#) [Review and Launch](#) [Next: Add Storage](#)

All defaults. Note there are advanced features in the lab pdf

1. Choose AMI 2. Choose Instance Type 3. Configure Instance 4. Add Storage 5. Add Tags 6. Configure Security Group 7. Review

Step 4: Add Storage

Your instance will be launched with the following storage device settings. You can attach additional EBS volumes and instance store volumes to your instance, or edit the settings of the root volume. You can also attach additional EBS volumes after launching an instance, but not instance store volumes. Learn more about storage options in Amazon EC2.

Volume Type	Device	Snapshot	Size (GiB)	Volume Type	IOPS	Throughput (MB/s)	Delete on Termination	Encryption
Root	/dev/xvda	snap-078b40dfe5a28c530	8	General Purpose SSD (gp2)	100 / 3000	N/A	<input checked="" type="checkbox"/>	Not Encrypted

[Add New Volume](#)

Free tier eligible customers can get up to 30 GB of EBS General Purpose (SSD) or Magnetic storage. Learn more about free usage tier eligibility and usage restrictions.

Cancel Previous **Review and Launch** Next: Add Tags

Storage at 8GiB is enough for this Hello World example

1. Choose AMI 2. Choose Instance Type 3. Configure Instance 4. Add Storage 5. Add Tags 6. Configure Security Group 7. Review

Step 5: Add Tags

A tag consists of a case-sensitive key-value pair. For example, you could define a tag with key = Name and value = Webserver. A copy of a tag can be applied to volumes, instances or both. Tags will be applied to all instances and volumes. Learn more about tagging your Amazon EC2 resources.

Key	(128 characters maximum)	Value	(256 characters maximum)	Instances	Volumes
useCase		EC2 Dockers Container		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>

[Add another tag](#) (Up to 50 tags maximum)

Cancel Previous **Review and Launch** Next: Configure Security Group

Tags for later reference

1. Choose AMI 2. Choose Instance Type 3. Configure Instance 4. Add Storage 5. Add Tags 6. Configure Security Group 7. Review

Step 6: Configure Security Group

A security group is a set of firewall rules that control the traffic for your instance. On this page, you can add rules to allow specific traffic to reach your instance. For example, if you want to set up a web server and allow Internet traffic to reach your instance, add rules that allow unrestricted access to the HTTP and HTTPS ports. You can create a new security group or select from an existing one below. [Learn more about Amazon EC2 security groups](#).

Assign a security group:

- Create a new security group
- Select an existing security group

Security group name: launch-wizard-2

Description: launch-wizard-2 created 2020-03-05T10:41:35.297-08:00

Type	Protocol	Port Range	Source	Description
SSH	TCP	22	Custom 0.0.0.0/0	e.g. SSH for Admin Desktop
HTTP	TCP	80	Custom 0.0.0.0/0	e.g. SSH for Admin Desktop

[Add Rule](#)

Warning
Rules with source of 0.0.0.0/0 allow all IP addresses to access your instance. We recommend setting security group rules to allow access from known IP addresses only.

[Cancel](#) [Previous](#) [Review and Launch](#)

Source is set to 0.0.0.0/0 allowing access from any IP.

1. Choose AMI 2. Choose Instance Type 3. Configure Instance 4. Add Storage 5. Add Tags 6. Configure Security Group 7. Review

Step 7: Review Instance Launch

Please review your instance launch details. You can go back to edit changes for each section. Click **Launch** to assign a key pair to your instance and complete the launch process.

AMI Details

amzn-ami-hvm-2017.09.1.20180115-x86_64-gp2 - ami-824c4ee2
 Amazon Linux AMI 2017.09.1.20180115 x86_64 HVM GP2
Root Device Type: ebs Virtualization type: hvm

[Edit AMI](#)

Instance Type

Instance Type	ECUs	vCPUs	Memory (GiB)	Instance Storage (GB)	EBS-Optimized Available	Network Performance
t2.micro	Variable	1	1	EBS only	-	Low to Moderate

[Edit Instance type](#)

Security Groups

Security group name: launch-wizard-2
Description: launch-wizard-2 created 2020-03-05T10:41:35.297-08:00

Type	Protocol	Port Range	Source	Description
SSH	TCP	22	0.0.0.0/0	
HTTP	TCP	80	0.0.0.0/0	
HTTP	TCP	80	::/0	

[Edit security groups](#)

[Cancel](#) [Previous](#) [Launch](#)

1. Choose AMI 2. Choose Instance Type 3. Configure Instance 4. Add Storage 5. Add Tags 6. Configure Security Group 7. Review

Step 7: Review Instance Launch

Network: vpc-56524731
 Subnet: No preference (default subnet in any Availability Zone)
 EBS-optimized: No
 Monitoring: No
 Termination protection: No
 Shutdown behavior: Stop
 Stop - Hibernate behavior: Disabled
 Capacity Reservation:
 - IAM role: None
 - Tenancy: default
 T2/T3 Unlimited: Disabled
 Host ID: Off
 Affinity: Off
 Kernel ID: Use default
 RAM disk ID: Use default
 Metadata accessible: Enabled
 Metadata token response hop limit: 1
 User data:
 - Assign Public IP: Use subnet setting (Enable)
 - Assign IPv6 IP: Use subnet setting (Enable)

Storage

Volume Type	Device	Snapshot	Size (GiB)	Volume Type	IOPS	Throughput (MB/s)	Delete on Termination	Encrypted
Root	/dev/xvda	snap-078b40dfe5a29c530	8	gp2	100 / 3000	N/A	Yes	Not Encrypted

Tags

Key	Value	Instances	Volumes
useCase	EC2 Docker Container	✓	✓

Buttons: Cancel | Previous | **Launch**

Overview of EC2 Instance

1. Choose AMI 2. Choose Instance Type 3. Configure Instance 4. Add Storage 5. Add Tags 6. Configure Security Group 7. Review

Step 7: Review Instance Launch

Please review your instance launch details. You can go back to edit changes for each section. Click **Launch** to assign a key pair to your instance and complete the launch process.

AMI Details

amzn-amazonlinux-2017.09.1.20180115-x86_64-gp2 - ami-824
 Amazon Linux AMI 2017.09.1.20180115.x86_64 HVM GP2
 Root Device Type: ebs Virtualization type: hvm

Instance Type

Instance Type	ECUs	vCPUs	Memory (GiB)
t2.micro	Variable	1	1

Security Groups

Security group name	Description
launch-wizard-2	launch-wizard-2 created 2020-03-05T10:41

Type	Protocol
SSH	TCP
HTTP	TCP
HTTP	TCP

Select an existing key pair or create a new key pair

A key pair consists of a **public key** that AWS stores, and a **private key file** that you store. Together, they allow you to connect to your instance securely. For Windows AMIs, the private key file is required to obtain the password used to log into your instance. For Linux AMIs, the private key allows you to securely SSH into your instance.

Note: The selected key pair will be added to the set of keys authorized for this instance. Learn more about removing existing key pairs from a public AMI.

Create a new key pair
 Key pair name: dockerLabKey
 Download Key Pair

You have to download the **private key file** (*.pem file) before you can continue. Store it in a secure and accessible location. You will not be able to download the file again after it's created.

Buttons: Cancel | **Launch Instances**

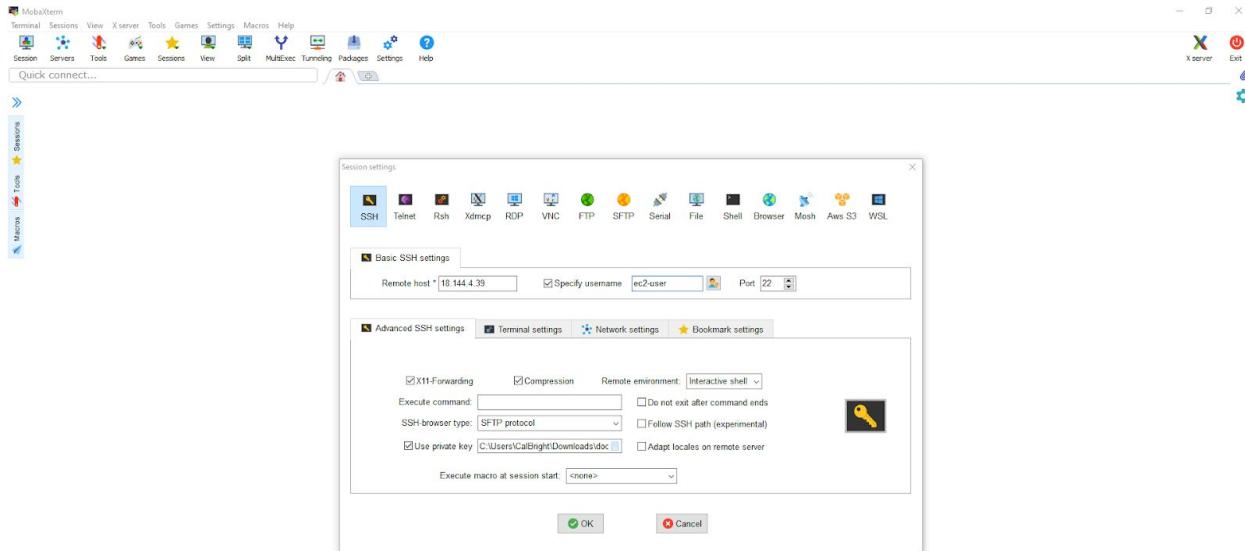
creation and local storage of key pair

The screenshot shows the AWS EC2 Dashboard. On the left, there's a sidebar with navigation links like 'New EC2 Experience', 'Events New', 'Tags', 'Reports', 'Limits', 'INSTANCES', 'Instances', 'Instance Types', 'Launch Templates New', 'Spot Requests', 'Savings Plans', 'Reserved Instances', 'Dedicated Hosts New', 'Capacity Reservations', 'IMAGES', 'AMIs', 'Bundle Tasks', 'ELASTIC BLOCK STORE', 'Volumes', 'Snapshots', 'Lifecycle Manager', 'NETWORK & SECURITY', and 'Security Groups'. The main panel displays an instance named 'i-0d6718f9ca62f35dc' with a Public DNS of 'ec2-18-144-4-39.us-west-1.compute.amazonaws.com'. The 'Status Checks' tab is selected, showing detailed information about the instance's state, network interfaces, and security settings. The IPv4 Public IP is highlighted in blue as '18.144.4.39'.

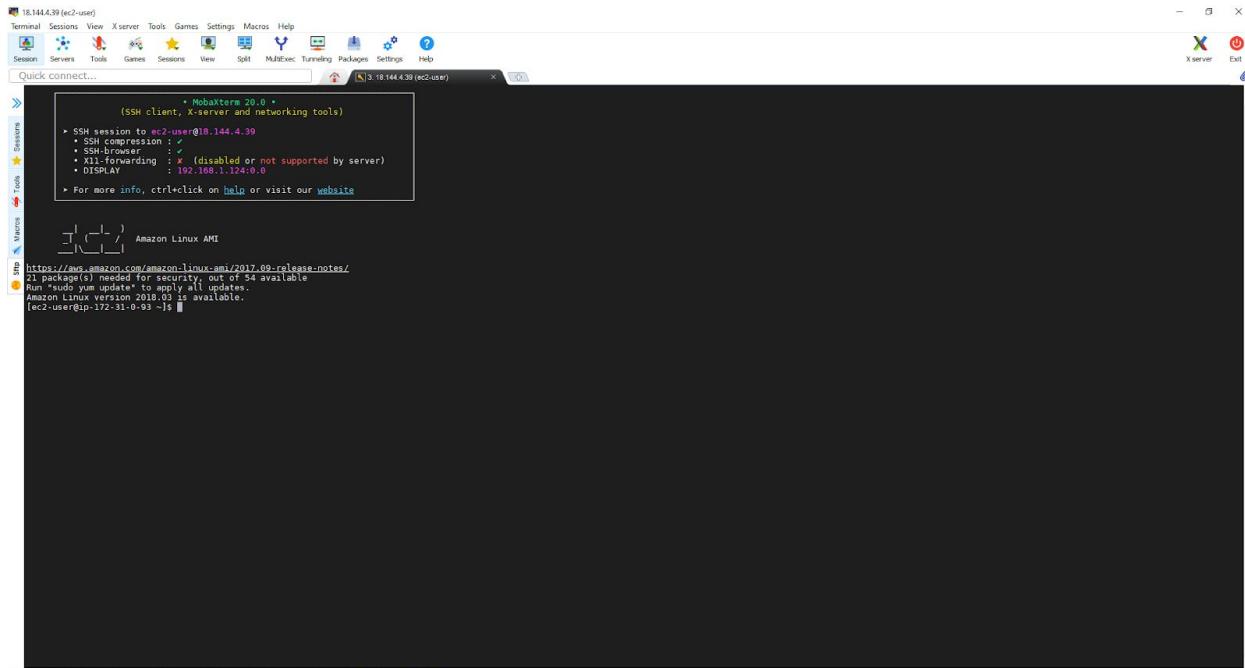
note the IPv4 Public IP

The screenshot shows the MoboTerm application window. The top menu bar includes 'Terminal', 'Sessions', 'View', 'Xserver', 'Tools', 'Games', 'Settings', 'Macros', 'Help', 'Session', 'Servers', 'Tools', 'Games', 'Sessions', 'View', 'Split', 'MultiTerm', 'Tunneling', 'Packages', 'Settings', and 'Help'. Below the menu is a toolbar with icons for 'Quick connect...', 'Session', 'Sessions', 'Tools', 'Macros', and 'Help'. A central window titled 'Session settings' displays a grid of icons representing different session types: SSH, Telnet, Rsh, Xdmcp, RDP, VNC, FTP, SFTP, Serial, File, Shell, Browser, Mosh, Aws S3, and WSL. A message 'Choose a session type...' is centered below the icons. At the bottom of the window are 'OK' and 'Cancel' buttons.

Choosing Session -> SSH



Note the remote host, ec2-user, and private key downloaded previously



Successful SSH into EC2

These steps have occurred thus far: (1) Launch an instance with the Amazon Linux AMI and (2) Connect to your instance.

```
18.144.4.39 [ec2-user]
Terminal Sessions View Xserver Tools Games Settings Macros Help
Session Servers Tools Games Sessions View Split Multicard Tunneling Packages Settings Help
Quick connect... 3.18.144.4.39 [ec2-user] x
Authenticating with public key "Imported-OpenSSH-Key: C:\Users\Calbright\Downloads\dockerLabKey.pem"
  * Mobacterm 20.0
  * (SSH client, X-server and networking tools)

* SSH session to ec2-user@18.144.4.39
* SSH compression: ✓
* SSH-browser: ✓
* X11-forwarding: ✘ (disabled or not supported by server)
* DISPLAY: 192.168.1.124:0.0

> For more info, ctrl+click on help or visit our website

[ ] Amazon Linux AMI
[ ] Step 1: Resolving Dependencies
https://aws.amazon.com/amazon-linux-ami/2017.98-release-notes/
21 package(s) needed for security, out of 54 available
Run 'sudo yum update' to apply all updates.
Resolving Dependencies
--> Running transaction check
--> Package amazon-ssm-agent.x86_64 0:2.2.120.0-1.amzn1 will be updated
--> Package amazon-ssm-agent.x86_64 0:2.3.714.0-1.amzn1 will be an update
--> Package aws-cfn-bootstrap.noarch 0:1.4-27.19.amzn1 will be updated
--> Package aws-cfn-bootstrap.noarch 0:1.4-27.19.amzn1 will be an update
--> Package aws-cfn-bootstrap.noarch 0:1.4-27.19.amzn1 will be an update
--> Package aws-cfn-bootstrap.noarch 0:1.14.9-1.48.amzn1 will be updated
--> Package aws-cfn-bootstrap.noarch 0:1.16.102-1.50.amzn1 will be an update
--> Package ca-certificates.noarch 0:2015.2.6-65.0.1.16.amzn1 will be updated
--> Package cloud-init.noarch 0:0.7-6.2.15.amzn1 will be an update
--> Package cloud-init.noarch 0:0.7-6.2.17.amzn1 will be an update
--> Package dbus.x86_64 1:1.6.12-14.29.amzn1 will be updated
--> Package dbus.x86_64 1:1.6.12-14.28.amzn1 will be updated
--> Package dbus.x86_64 1:1.6.12-14.29.amzn1 will be an update
--> Package dracut.noarch 0:004-409.31.amzn1 will be updated
--> Package dracut.noarch 0:004-409.32.amzn1 will be an update
amazon-main/latest/testfilelists.db
amazon-main/latest/testfilelists.db
--> Package efsprogs.x86_64 0:1.42.12-4.40.amzn1 will be updated
--> Package efsprogs.x86_64 0:1.43-2.43.amzn1 will be an update
--> Processing Dependency: libfuse.so.2(FUSE 2.3)(64bit) for package: efsprogs-1.43.5-2.43.amzn1.x86_64
--> Processing Dependency: libfuse.so.2(1)(64bit) for package: efsprogs-1.43.5-2.43.amzn1.x86_64
--> Processing Dependency: libfuse.so.2(1)(64bit) for package: efsprogs-1.43.5-2.43.amzn1.x86_64
--> Package e2fsprogs.x86_64 0:1.43.5-2.43.amzn1 will be an update
--> Package e2-net-utils.noarch 0:0.5-1.34.amzn1 will be updated
--> Package e2-net-utils.noarch 0:0.5-1.34.amzn1 will be an update
--> Package e2-utils.noarch 0:0.5-1.34.amzn1 will be updated
--> Package e2-utils.noarch 0:0.6-1.1.amzn1 will be an update
| 2.1 kB 00:00:00
| 2.5 kB 00:00:00
| 5.7 MB 00:00:00
| 13 MB 00:00:01
```

```
sudo yum update -y
```

System updated

```
sudo yum install -y docker
```

```
Terminal Sessions View Xserver Tools Games Settings Macros Help
Session Servers Tools Games Sessions View Split Multitab Tunneling Packages Settings Help
Quick connect... 3: 19.144.4.39 (ec2-user) Xserver Be

---- Package pigz.x86_64 0:2.3.3-1.6.amzn1 will be installed
----> Package runc.x86_64 0:1.0.0-0.1.20190510.git2b18fe1.0.amzn1 will be installed
----> Package xfsprogs.x86_64 0:4.5.0-18.23.amzn1 will be installed
--> Finished Dependency Resolution
Dependencies Resolved



| Package                      | Arch   | Version                               | Repository   | Size  |
|------------------------------|--------|---------------------------------------|--------------|-------|
| Installing:                  |        |                                       |              |       |
| docker                       | x86_64 | 18.09.9ce-2.52.amzn1                  | amzn-updates | 35 M  |
| Installing for dependencies: |        |                                       |              |       |
| libseccomp                   | x86_64 | 2.3.1-2.4.amzn1                       | amzn-updates | 26 M  |
| libseccomp                   | x86_64 | 2.3.1-2.4.amzn1                       | amzn-main    | 79 M  |
| pigz                         | x86_64 | 2.3.3-1.6.amzn1                       | amzn-main    | 71 k  |
| runc                         | x86_64 | 1.0.0-0.1.20190510.git2b18fe1.0.amzn1 | amzn-updates | 2.3 M |
| xfsprogs                     | x86_64 | 4.5.0-18.23.amzn1                     | amzn-updates | 1.7 M |


Transaction Summary

Install 1 Package (+5 Dependent packages)

Total download size: 65 M
Installed size: 223 M
Downloading packages:
(1/6): libseccomp.x86_64-2.3.1-2.4.amzn1.x86_64.rpm 79 KB 00:00:00
(2/6): runc.x86_64-1.0.0-0.1.20190510.git2b18fe1.0.amzn1.x86_64.rpm 71 MB 00:00:00
(3/6): libseccomp.x86_64-2.3.1-2.4.amzn1.x86_64.rpm 2.3 MB 00:00:00
(4/6): xfsprogs.x86_64-4.5.0-18.23.amzn1.x86_64.rpm 1.7 MB 00:00:00
(5/6): libseccomp.x86_64-2.3.1-2.4.amzn1.x86_64.rpm 0.0 kB 00:00:00
(6/6): docker.x86_64-18.09.9ce-2.52.amzn1.x86_64.rpm 35 MB 00:00:05

Total
Running transaction check
Running transaction test
Transaction test succeeded
Running transaction
  Installing : libseccomp-2.3.1-2.4.amzn1.x86_64 1/6
  Installing : runc-1.0.0-0.1.20190510.git2b18fe1.0.amzn1.x86_64 2/6
  Installing : libseccomp-2.3.1-2.4.amzn1.x86_64 3/6
  Installing : pigz-2.3.3-1.6.amzn1.x86_64 4/6
  Installing : xfsprogs-4.5.0-18.23.amzn1.x86_64 5/6
  Installing : libseccomp-2.3.1-2.4.amzn1.x86_64 6/6
  Verifying : docker-18.09.9ce-2.52.amzn1.x86_64 1/6
  Verifying : xfsprogs-4.5.0-18.23.amzn1.x86_64 2/6
  Verifying : libseccomp-2.3.1-2.4.amzn1.x86_64 3/6
  Verifying : libseccomp-2.3.1-2.4.amzn1.x86_64 4/6
  Verifying : runc-1.0.0-0.1.20190510.git2b18fe1.0.amzn1.x86_64 5/6
  Verifying : pigz-2.3.3-1.6.amzn1.x86_64 6/6

Installed:
  docker.x86_64 0:18.09.9ce-2.52.amzn1

Dependency Installed:
  containedr.x86_64 0:1.2.6-1.2.amzn1      libseccomp.x86_64 0:2.3.1-2.4.amzn1      pigz.x86_64 0:2.3.3-1.6.amzn1      runc.x86_64 0:1.0.0-0.1.20190510.git2b18fe1.0.amzn1      xfsprogs.x86_64 0:4.5.0-18.23.amzn1

Completes!
[ec2-user@ip-172-31-0-93 ~]
```

Docker updated and installed

```

18.144.4.39 (ec2-user)
Terminal Sessions View Xserver Tools Games Settings Macros Help
Session Servers Tools Games Sessions View Split Multiview Tunneling Packages Settings Help
Quick connect... [ 3. 18.144.4.39 (ec2-user) ] X server Exit
--> Finished Dependency Resolution
Dependencies Resolved
=====
Package           Arch      Version            Repository          Size
Installing:
docker           x86_64    18.09.9ce-2.52.amzn1
Installing for dependencies:
containerd        x86_64    1.2.6-1.2.amzn1
libseccomp        x86_64    2.3.1-2.4.amzn1
ipig              x86_64    2.3.3-1.6.amzn1
runc              x86_64    1.0.0-0.1.20190510.git2b18fe1.0.amzn1
xfsprogs         x86_64    4.5.0-18.23.amzn1
=====
Transaction Summary
Install 1 Package (+5 Dependent packages)
Total download size: 65 M
Installed size: 223 M
Downloading packages:
(1/6): libseccomp-2.3.1-2.4.amzn1.x86_64.rpm   79 kB  00:00:00
(2/6): ipig-2.3.3-1.6.amzn1.x86_64.rpm       71 kB  00:00:00
(3/6): runc-1.0.0-0.1.20190510.git2b18fe1.0.amzn1.x86_64.rpm  2.3 MB  00:00:00
(4/6): xfsprogs-4.5.0-18.23.amzn1.x86_64.rpm   1.7 MB  00:00:00
(5/6): containerd-1.2.6-1.2.amzn1.x86_64.rpm   26 kB  00:00:04
(6/6): docker-18.09.9ce-2.52.amzn1.x86_64.rpm  35 kB  00:00:05
=====
Total
Running transaction check
Running transaction test
Transaction test succeeded
Running transaction
  Installing : libseccomp-2.3.1-2.4.amzn1.x86_64
  Installing : containerd-1.2.6-1.2.amzn1.x86_64
  Installing : ipig-2.3.3-1.6.amzn1.x86_64
  Installing : runc-1.0.0-0.1.20190510.git2b18fe1.0.amzn1.x86_64
  Verifying  : docker-18.09.9ce-2.52.amzn1.x86_64
  Verifying  : xfsprogs-4.5.0-18.23.amzn1.x86_64
  Verifying  : containerd-1.2.6-1.2.amzn1.x86_64
  Verifying  : runc-1.0.0-0.1.20190510.git2b18fe1.0.amzn1.x86_64
  Verifying  : ipig-2.3.3-1.6.amzn1.x86_64
  Verifying  : docker-18.09.9ce-2.52.amzn1.x86_64
  Verifying  : containerd-1.2.6-1.2.amzn1.x86_64
  Verifying  : runc-1.0.0-0.1.20190510.git2b18fe1.0.amzn1.x86_64
  Verifying  : ipig-2.3.3-1.6.amzn1.x86_64
  Installed:
    docker.x86_64 0:18.09.9ce-2.52.amzn1
  Dependency Installed:
    containerd.x86_64 0:1.2.6-1.2.amzn1    libseccomp.x86_64 0:2.3.1-2.4.amzn1    ipig.x86_64 0:2.3.3-1.6.amzn1    runc.x86_64 0:1.0.0-0.1.20190510.git2b18fe1.0.amzn1    xfsprogs.x86_64 0:4.5.0-18.23.amzn1
  Complete!
[ec2-user@ip-172-31-0-93 ~]$ sudo service docker start
Starting docker:                                         [ OK ]
[ec2-user@ip-172-31-0-93 ~]$ 

```

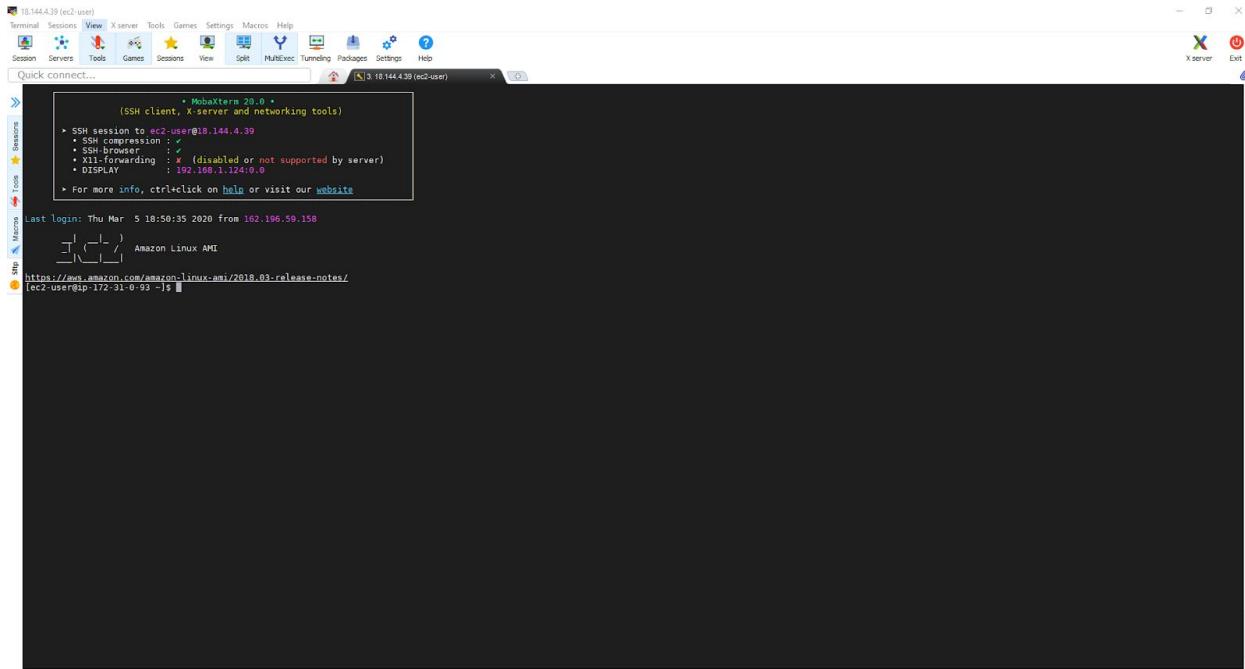
sudo service docker start

```

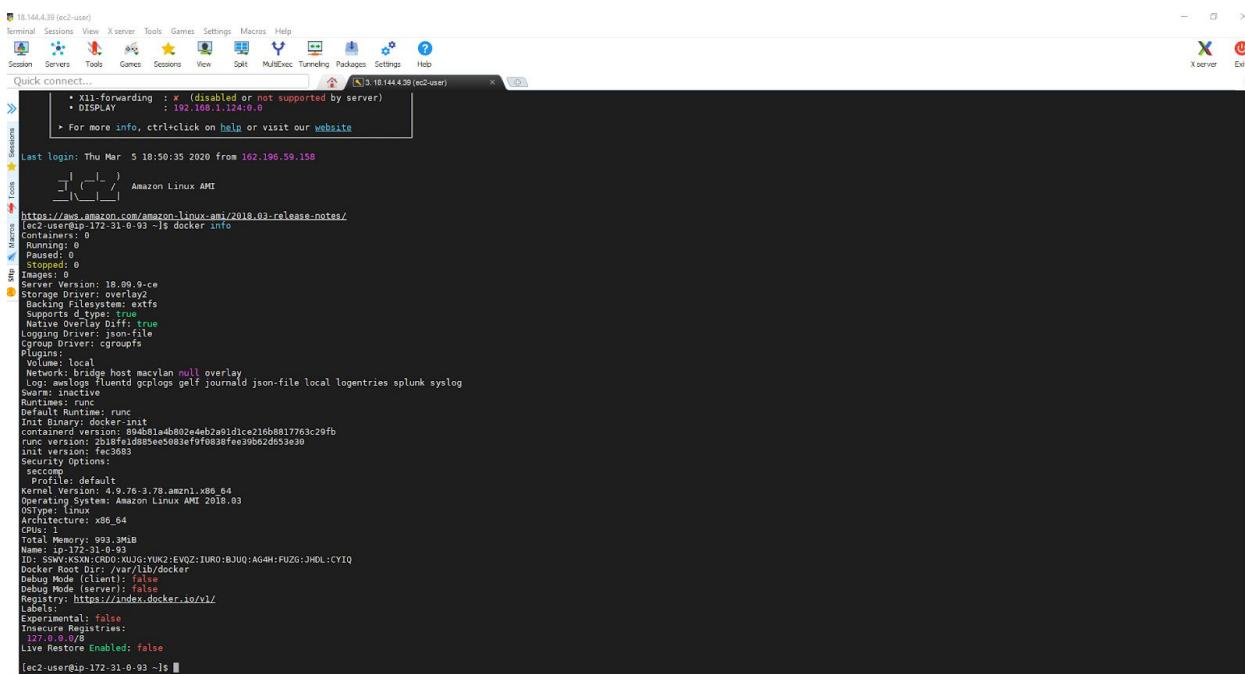
18.144.4.39 (ec2-user)
Terminal Sessions View Xserver Tools Games Settings Macros Help
Session Servers Tools Games Sessions View Split Multiview Tunneling Packages Settings Help
Quick connect... [ 3. 18.144.4.39 (ec2-user) ] X server Exit
--> Dependencies Resolved
Dependencies Resolved
=====
Package           Arch      Version            Repository          Size
Installing:
docker           x86_64    18.09.9ce-2.52.amzn1
Installing for dependencies:
containerd        x86_64    1.2.6-1.2.amzn1
libseccomp        x86_64    2.3.1-2.4.amzn1
ipig              x86_64    2.3.3-1.6.amzn1
runc              x86_64    1.0.0-0.1.20190510.git2b18fe1.0.amzn1
xfsprogs         x86_64    4.5.0-18.23.amzn1
=====
Transaction Summary
Install 1 Package (+5 Dependent packages)
Total download size: 65 M
Installed size: 223 M
Downloading packages:
(1/6): libseccomp-2.3.1-2.4.amzn1.x86_64.rpm   79 kB  00:00:00
(2/6): ipig-2.3.3-1.6.amzn1.x86_64.rpm       71 kB  00:00:00
(3/6): runc-1.0.0-0.1.20190510.git2b18fe1.0.amzn1.x86_64.rpm  2.3 MB  00:00:00
(4/6): xfsprogs-4.5.0-18.23.amzn1.x86_64.rpm   1.7 MB  00:00:00
(5/6): containerd-1.2.6-1.2.amzn1.x86_64.rpm   26 kB  00:00:04
(6/6): docker-18.09.9ce-2.52.amzn1.x86_64.rpm  35 kB  00:00:05
=====
Total
Running transaction check
Running transaction test
Transaction test succeeded
Running transaction
  Installing : libseccomp-2.3.1-2.4.amzn1.x86_64
  Installing : containerd-1.2.6-1.2.amzn1.x86_64
  Installing : ipig-2.3.3-1.6.amzn1.x86_64
  Installing : runc-1.0.0-0.1.20190510.git2b18fe1.0.amzn1.x86_64
  Verifying  : docker-18.09.9ce-2.52.amzn1.x86_64
  Verifying  : xfsprogs-4.5.0-18.23.amzn1.x86_64
  Verifying  : containerd-1.2.6-1.2.amzn1.x86_64
  Verifying  : runc-1.0.0-0.1.20190510.git2b18fe1.0.amzn1.x86_64
  Verifying  : ipig-2.3.3-1.6.amzn1.x86_64
  Installed:
    docker.x86_64 0:18.09.9ce-2.52.amzn1
  Dependency Installed:
    containerd.x86_64 0:1.2.6-1.2.amzn1    libseccomp.x86_64 0:2.3.1-2.4.amzn1    ipig.x86_64 0:2.3.3-1.6.amzn1    runc.x86_64 0:1.0.0-0.1.20190510.git2b18fe1.0.amzn1    xfsprogs.x86_64 0:4.5.0-18.23.amzn1
  Complete!
[ec2-user@ip-172-31-0-93 ~]$ sudo service docker start
Starting docker:                                         [ OK ]
[ec2-user@ip-172-31-0-93 ~]$ sudo usermod -a -G docker ec2-user
[ec2-user@ip-172-31-0-93 ~]$ 

```

Elevated EC2 user permissions for docker services



Log out and log back in



Confirmation that docker is set up and running

```

18.144.4.39 (ec2-user)
Terminal Sessions View Xserver Tools Games Settings Macros Help
Session Servers Tools Games Sessions View Split MultiExec Tunneling Packages Settings Help
Quick connect... [3] 18.144.4.39 (ec2-user) X server Exit
DISPLAY : 102.168.1.124:0.0
For more info, ctrl+click on help or visit our website

Last login: Thu Mar 5 18:50:35 2020 from 162.106.59.158
[ec2-user@ip-172-31-0-93 ~]$ docker info
Containers: 0
Running: 0
Paused: 0
Stopped: 0
Images: 0
Server Version: 18.09.9-ce
Storage Driver: overlay2
  Backing Filesystem: ext4
  Supports d_type: true
  Native Overlay Diff: true
Logging Driver: json-file
Log Level: info
Cgroup Driver: cgroups
Plugins:
  Volume: local
  Network: bridge host macvlan null overlay
Log: awslogs fluentd gcplogs gelf journal json-file local logentries splunk syslog
Swarm: inactive
Networks: none
Default Runtime: runc
Init Binary: docker-init
containerd version: 774c483a4b802e4eb2a93dice21eb8817763c29fb
runc version: 2b18fe1d855ea5983ef9f0838fee39b62d6553e30
init version: fec3683
Security Options:
  seccomp
Profile: default
Kernel Version: 4.19.76-3.78.ami.1-x86_64
Operating System: Amazon Linux AMI 2018.03
OSType: linux
Architecture: x86_64
CPUs: 4
Total Memory: 993.3MiB
Name: ip-172-31-0-93
ID: EYK2:EVQZ:IURO:BJUQ:AG4H:FUZG:JHDL:CYIQ
Docker Root Dir: /var/lib/docker
Debug Mode (client): false
Debug Mode (server): false
Registry: https://index.docker.io/v1/
Labels:
Experimental: false
Insecure Registries:
  127.0.0.0/8
Live Restore Enabled: false
[ec2-user@ip-172-31-0-93 ~]$ touch Dockerfile
[ec2-user@ip-172-31-0-93 ~]$ 

```

“touch Dockerfile”

```

18.144.4.39 (ec2-user)
Terminal Sessions View Xserver Tools Games Settings Macros Help
Session Servers Tools Games Sessions View Split MultiExec Tunneling Packages Settings Help
Quick connect... [3] 18.144.4.39 (ec2-user) X server Exit
File: Dockerfile

```

“nano Dockerfile”

```

18.144.4.39 (ec2-user)
Terminal Sessions View Xserver Tools Games Settings Macros Help
Session Servers Tools Games Sessions View Split MultiExec Tunneling Packages Settings ? 
Quick connect... File: Dockerfile Modified
FROM ubuntu:12.04
# Install dependencies
RUN apt-get update -y
RUN apt-get install -y apache2
# Write hello world message
RUN echo "Hello World!" > /var/www/index.html
# Configure apache
RUN a2enmod rewrite
RUN chmod 755 /var/www-data /var/www
ENV APACHE_RUN_USER www-data
ENV APACHE_RUN_GROUP www-data
ENV APACHE_LOG_DIR /var/log/apache2
EXPOSE 80
CMD ["/usr/sbin/apache2", "-D", "foreground"]

```

Container file within Nano (code from Lab pdf)

```

18.144.4.39 (ec2-user)
Terminal Sessions View Xserver Tools Games Settings Macros Help
Session Servers Tools Games Sessions View Split MultiExec Tunneling Packages Settings ? 
Quick connect... File: Dockerfile Modified
Stopped: 0
Images: 0
Filesystem Size: 18.09.9-ce
Storage Driver: overlay2
Backing Filesystems: extfs
Driver-Config: 2 types
Migrate Over Diff: true
Logging Driver: json-file
Group Driver: cgroups
Pids Driver: default
Volume: local
Network: bridge host macvlan null overlay
Swarm: inactive
Runtime: runc
Default Runtime: runc
Init Binary: docker-init
Containerd version: 894d81a4b802e4ebza9idice21b8837763c29fb
Init version: fec3983
Security Options:
  seccomp
    Profile: default
Kernel Version: 4.16-3.78.amzn1.x86_64
Operating System: Amazon Linux AMI 2018.03
OSType: linux
Architecture: x86_64
CPUs: 1
Total Memory: 993.3MiB
Name: ip-172-31-0-93
ID: SSWX:KSNX:CHD0:XUJG:YUK2:EVOZ:IURO:BJU0:AG4H:FUZG:JHDL:CYIQ
Docker Root Dir: /var/lib/docker
Debug Mode (server): false
Debug Mode (client): false
Registry: https://index.docker.io/v1/
Labels:
Experimental: false
Insecure Registries:
  127.0.0.1:5000
Live Restore Enabled: false
[ec2-user@ip-172-31-0-93 ~]$ touch Dockerfile
[ec2-user@ip-172-31-0-93 ~]$ nano Dockerfile
[ec2-user@ip-172-31-0-93 ~]$ cat Dockerfile
# Install dependencies
RUN apt-get update -y
RUN apt-get install -y apache2
# Write hello world message
RUN echo "Hello World!" > /var/www/index.html
# Configure apache
RUN chmod 755 /var/www-data /var/www
ENV APACHE_RUN_USER www-data
ENV APACHE_RUN_GROUP www-data
ENV APACHE_LOG_DIR /var/log/apache2
EXPOSE 80
CMD ["/usr/sbin/apache2", "-D", "foreground"]
[ec2-user@ip-172-31-0-93 ~]$

```

Concatenated Dockerfile within CLI

```

18.144.4.39 (ec2-user)
Terminal Sessions View Xserver Tools Games Settings Macros Help
Session Servers Tools Games Sessions View Split MultiExec Tunneling Packages Settings ? X server Exit
Quick connect...
[3 18.144.4.39 (ec2-user)] 

Enabling module dir.
Enabling module env.
Enabling module mime.
Enabling module negotiation.
Enabling module rewrite.
Enabling module stevenciftci.
Enabling module status.
Enabling module auth_basic.
Enabling module auth_digest.
Enabling module authz_default.
Enabling module authz_user.
Enabling module authn_file.
Enabling module authn_host.
Enabling module authn_socache.
Setting up apache2-worker (2.2.22-lubuntu.11) ...
invoke-rc.d: policy-rc.d denied execution of start.
Setting up apache2 (2.2.22-lubuntu.11) ...
Processing triggers for libapache2-mod-wsgi-py2 (1.0-4) ...
ldconfig deferred processing now taking place
Removing intermediate container 75ea79d5d124c
Step 4/11 : RUN echo "Hello World!" > /var/www/index.html
--> Running in 56917b58c2e0
--> b668ac974c38
Step 5/11 : RUN a2enmod rewrite
--> Running in 8a01ab1bb800
Enabling module rewrite
To activate the new configuration, you need to run:
service apache2 restart
--> Running in 8a01ab1bb800
--> ff1fc2af1c20
Step 6/11 : RUN chown -R www-data:www-data /var/www
--> Running in 56917b58c2e0
--> d218d15d4425
--> d910f2cbde
Step 7/11 : ENV APACHE_RUN_USER www-data
--> Running in e047e37c18c
--> Removing intermediate container e947e327c18c
--> 5f90c29f6e6a
Step 8/11 : EXPOSE 80
--> Running in ab577329cf77
--> Removing intermediate container a0577329cf77
Step 9/11 : ENV APACHE_LOG_DIR /var/log/apache2
--> Running in b29c802911f5
--> Removing intermediate container b29c802911f5
--> a02d78baeaea
Step 10/11 : EXPOSE 80
--> Running in ab897bdc4c69
--> Removing intermediate container ab897bdc4c69
--> ac542dd5d5be
Step 11/11 : CMD ["/usr/sbin/apache2", "-D", "FOREGROUND"]
--> Running in b668ac974c38
--> b668ac974c38
Successfullly tagged hello-world:latest
[ec2-user@ip-172-31-0-93 ~]$ 

```

“docker build -t hello-world .”

```

18.144.4.39 (ec2-user)
Terminal Sessions View Xserver Tools Games Settings Macros Help
Session Servers Tools Games Sessions View Split MultiExec Tunneling Packages Settings ? X server Exit
Quick connect...
[3 18.144.4.39 (ec2-user)] 

Enabling module stevenciftci.
Enabling module status.
Enabling module auth_basic.
Enabling module auth_digest.
Enabling module authz_default.
Enabling module authz_user.
Enabling module authn_file.
Enabling module authn_host.
Enabling module authn_socache.
Setting up apache2-worker (2.2.22-lubuntu.11) ...
invoke-rc.d: policy-rc.d denied execution of start.
Setting up apache2 (2.2.22-lubuntu.11) ...
Processing triggers for libapache2-mod-wsgi-py2 (1.0-4) ...
ldconfig deferred processing now taking place
Removing intermediate container 75ea79d5d124c
Step 4/11 : RUN echo "Hello World!" > /var/www/index.html
--> Running in 56917b58c2e0
--> b668ac974c38
Step 5/11 : RUN a2enmod rewrite
--> Running in 8a01ab1bb800
Enabling module rewrite
To activate the new configuration, you need to run:
service apache2 restart
--> Running in 8a01ab1bb800
--> ff1fc2af1c20
Step 6/11 : RUN chown -R www-data:www-data /var/www
--> Running in 56917b58c2e0
--> d218d15d4425
--> d910f2cbde
Step 7/11 : ENV APACHE_RUN_USER www-data
--> Running in e047e37c18c
--> Removing intermediate container e947e327c18c
--> 5f90c29f6e6a
Step 8/11 : EXPOSE 80
--> Running in ab577329cf77
--> Removing intermediate container a0577329cf77
Step 9/11 : ENV APACHE_LOG_DIR /var/log/apache2
--> Running in b29c802911f5
--> Removing intermediate container b29c802911f5
--> a02d78baeaea
Step 10/11 : EXPOSE 80
--> Running in ab897bdc4c69
--> Removing intermediate container ab897bdc4c69
--> ac542dd5d5be
Step 11/11 : CMD ["/usr/sbin/apache2", "-D", "FOREGROUND"]
--> Running in 838288ce7a88
--> b668ac974c38
Successfullly tagged hello-world:latest
[ec2-user@ip-172-31-0-93 ~]$ docker images
REPOSITORY          TAG      IMAGE ID      CREATED             SIZE
hello-world         latest   b668ac974c38   38 seconds ago   179MB
ubuntu              12.04    5b117edd076   2 years ago       104MB
[ec2-user@ip-172-31-0-93 ~]$ 

```

“docker images” Docker created successfully

```

18.144.4.39 (ec2-user)
Terminal Sessions View X server Tools Games Settings Macros Help
Sessions Servers Tools Games Sessions View Split MultiExec Tunneling Packages Settings ? X server Exit
Quick connect... 3.18.144.4.39 (ec2-user)

Enabling module deflate.
Enabling module authz_default.
Enabling module authz_groupfile.
Enabling module authz_file.
Enabling module negotiation.
Enabling module rewrite.
Setting up apache2-mpm-worker (2.2.22-1ubuntu1.11) ...
invoke-rc.d: policy-rc.d denied execution of start.
Processing triggers for libc-bin ...
ldconfig deferred processing now taking place
Removing intermediate container 75ea70d5124c
Reusing existing container 7cd6227fcea
Step 4/11 : RUN echo "Hello World!" > /var/www/index.html
Removing intermediate container 56017658c2e0
--> 088d750c4fc8
Step 5/11 : RUN a2enmod rewrite
Enabling module rewrite.
To activate the new configuration, you need to run:
  service apache2 reload
Removing intermediate container 0a01a1b1b880
--> ff0fca2f1c20
Step 6/11 : ENV WWW_HOME -v www-data:www-data /var/www
--> Running in d218dd154425
Removing intermediate container d218dd154425
--> e947e37c1b0c
Step 7/11 : ENV APACHE_RUN_USER www-data
--> Running in e947e37c1b0c
Removing intermediate container e947e327c18c
--> a577329cf778
Step 8/11 : ENV APACHE_RUN_GROUP www-data
--> Running in a0577329cf77
Removing intermediate container a0577329cf77
--> b2dd20d889f2
Step 9/11 : ENV APACHE_LOG_DIR /var/log/apache2
--> b22c8020211f5
Removing intermediate container b22c8020211f5
--> a62d70a6ed5e
Step 10/11 : EXPOSE 80
--> Running in ab897bdc4c69
Removing intermediate container ab897bdc4c69
--> ac526d5d5b9e
Step 11/11 : CMD ["/usr/sbin/apache2", "-D", "FOREGROUND"]
--> Running in 838288ce7a88
Removing intermediate container 838288ce7a88
Successfully built b668ac974c38
Successfully tagged hello-world:latest
18.144.4.39:~$ docker images
REPOSITORY          TAG      IMAGE ID      CREATED       SIZE
hello-world         latest   b668ac974c38   38 seconds ago  179MB
ubuntu              14.04    5b1840707070   2 years ago   104MB
18.144.4.39:~$ docker run -p 80:80 hello-world
apache2: Could not reliably determine the server's fully qualified domain name, using 172.17.0.2 for ServerName

```

“docker run -p 80:80 hello-world” Container needs to know which port in EC2 Instance



Apache server running from container inside EC2 instance

```

18.144.4.39 (ec2-user)
Terminal Sessions View Xserver Tools Games Settings Macros Help
Session Servers Tools Games Sessions View Split MultiExec Tunneling Packages Settings Help
Quick connect: 3. 18.144.4.39 (ec2-user)
Enabling module authn_file.
Enabling module authz_host.
Enabling module rewrite.
Setting up apache2.conf-worker (2.2.22-lubuntu1.11) ...
[warn] [Fri Jul 21 17:47:00.000 2017] [mpm-worker] [pid 1] mod_wsgi: denied execution of start.
Setting up apache2 (2.2.22-lubuntu1.11) ...
Processing triggers for libc-bin ...
Removing intermediate container 75ea79d5d124
--> 7cd50227f7ea
Step 5/11 : RUN ["Hello World!"] > /var/www/index.html
--> Running in 56917b58c2e9
Removing intermediate container 56917b58c2e9
Step 6/11 : RUN a2enmod rewrite
--> Running in 0a01ab1bb080
Enabling module rewrite ...
To activate the new configuration, you need to run:
service apache2 restart
--> Removing intermediate container 0a01ab1bb080
Step 7/11 : ENV APACHE_MODE_SET=ER www-data
--> Running in 5196529fe67b
Removing intermediate container 5196529fe67b
Step 8/11 : ENV APACHE_LOG_DIR /var/log/apache2
--> Running in d218dd154425
Removing intermediate container d218dd154425
--> d9916f2cbdee
Step 9/11 : ENV APACHE_PID_FILE /var/run/apache2.pid
--> Running in e947e327c18c
Removing intermediate container e947e327c18c
--> 5196529fe67b
Step 10/11 : EXPOSE 80
--> Running in ab897bdc4c69
Removing intermediate container ab897bdc4c69
--> ac542dd5b6
Step 11/11 : CMD ["/usr/sbin/apache2", "-D", "FOREGROUND"]
--> b668ac974c38
Successfully tagged hello-world:latest
[ec2-user@ip-172-31-0-93 ~]$ docker images
REPOSITORY          TAG      IMAGE ID      CREATED       SIZE
hello-world         latest   b668ac974c38   38 seconds ago  179MB
ubuntu              12.04    5b17addd0b76   2 years ago   104MB
[ec2-user@ip-172-31-0-93 ~]$ docker run -it b668ac974c38 hello-world
[ec2-user@ip-172-31-0-93 ~]$ aws configure
[ec2-user@ip-172-31-0-93 ~]$ ls aws configure
Access Key ID [None]:
AWS Secret Access Key [None]:
Default region name [None]: us-west-1
Default output format [None]:
[ec2-user@ip-172-31-0-93 ~]$ 

```

Defining Access Key ID and Secret Access Key

Amazon ECS makes it easy to deploy, manage, and scale Docker containers running applications, services, and batch processes. Amazon ECS places containers across your cluster based on your resource needs and is integrated with familiar features like Elastic Load Balancing, EC2 security groups, EBS volumes and IAM roles.

[Get started](#)

Learn more about Amazon ECS



“Amazon ECR Repositories” from AWS web console

ECR > Repositories > Create repository

Create repository

Repository configuration

Repository name
758287676861.dkr.ecr.us-east-1.amazonaws.com/

A namespace can be included with your repository name (e.g. namespace/repo-name).

Tag immutability
Enable tag immutability to prevent image tags from being overwritten by subsequent image pushes using the same tag. Disable tag immutability to allow image tags to be overwritten.
 Disabled

Scan on push
Enable scan on push to have each image automatically scanned after being pushed to a repository. If disabled, each image scan must be manually started to get scan results.
 Disabled

[Cancel](#) [Create repository](#)

[Feedback](#) [English \(US\)](#)

© 2008 - 2020, Amazon Web Services, Inc. or its affiliates. All rights reserved. [Privacy Policy](#) [Terms of Use](#)

Note the Repository name

Amazon Container Services X

Successfully created repository X [View push commands](#)

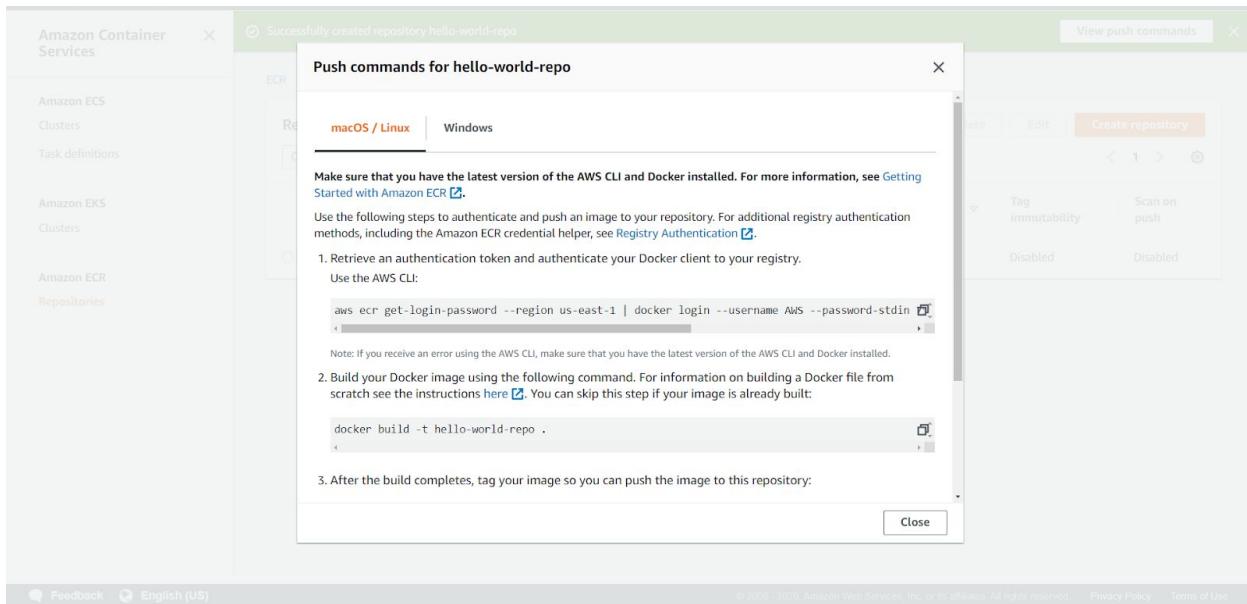
Repositories (1) X [View push commands](#) [Delete](#) [Edit](#) [Create repository](#)

Repository name	URI	Created at	Tag immutability	Scan on push
hello-world-repo	758287676861.dkr.ecr.us-east-1.amazonaws.com/hello-world-repo	03/05/20, 12:04:44 PM	Disabled	Disabled

[Feedback](#) [English \(US\)](#)

© 2008 - 2020, Amazon Web Services, Inc. or its affiliates. All rights reserved. [Privacy Policy](#) [Terms of Use](#)

ECR repository created within AWS web console



Note the first Push commands for a SSH session

```
[ec2-user@ip-172-31-0-93 ~]$ aws ecr get-login-password --region us-east-1 | docker login --username AWS --password-stdin 758287676861.dkr.ecr.us-east-1.amazonaws.com/hello-world-repo
? see help text, you can run:
aws help
aws <command> help
aws: error: argument operation: Invalid choice, valid choices are:
batch-check-layer-availability
batch-get-image
cancel-layer-upload
create-layer
delete-layer
delete-repository
describe-images
describe-lifecycle-policy
get-authorization-token
get-lifecycle-policy
get-repository-policy
list-images
put-image
set-repository-policy
upload-layer-part
help
Error: cannot perform an interactive login from a non TTY device
[ec2-user@ip-172-31-0-93 ~]$ aws ecr get-login --no-include-email --region us-east-1
[ec2-user@ip-172-31-0-93 ~]$
```

Copied and pasted the first set of Push commands then used the lecture command. Login successfully.
key not shown.

```

WARNING! Using --password via the CLI is insecure. Use --password-stdin.
WARNING! Your password will be stored unencrypted in /home/ec2-user/.docker/config.json.
Configure a credential helper to remove this warning. See
https://docs.docker.com/engine/reference/commandline/login/#credentials-store

Login Succeeded
[ec2-user@ip-172-31-0-93 ~]$ docker build -t hello-world-repo .
Sending build context to Docker daemon 14.34kB
Step 1/11 : FROM ubuntu:12.04
--> 5b117edd0b76
Step 2/11 : RUN apt-get update -y
--> Using cache
--> c446d78319ec
Step 3/11 : RUN apt-get install -y apache2
--> Using cache
--> 7cd9d227fcea
Step 4/11 : RUN echo "Hello World!" > /var/www/index.html
--> 088d750c4fc8
Step 5/11 : RUN a2enmod rewrite
--> Using cache
--> f0f1ca2fc120
Step 6/11 : RUN chown -R www-data:www-data /var/www
--> Using cache
--> d9910f2cbdee
Step 7/11 : ENV APACHE_RUN_USER www-data
--> Using cache
--> 519b629ffeb7
Step 8/11 : ENV APACHE_RUN_GROUP www-data
--> Using cache
--> b688ac974c38
Successfully tagged hello-world-repo:latest
[ec2-user@ip-172-31-0-93 ~]$ 
[ec2-user@ip-172-31-0-93 ~]$ 

```

“docker build -t hello-world-repo .”

```

WARNING! Using --password via the CLI is insecure. Use --password-stdin.
WARNING! Your password will be stored unencrypted in /home/ec2-user/.docker/config.json.
Configure a credential helper to remove this warning. See
https://docs.docker.com/engine/reference/commandline/login/#credentials-store

Login Succeeded
[ec2-user@ip-172-31-0-93 ~]$ docker build -t hello-world-repo .
Sending build context to Docker daemon 14.34kB
Step 1/11 : FROM ubuntu:12.04
--> 5b117edd0b76
Step 2/11 : RUN apt-get update -y
--> Using cache
--> c446d78319ec
Step 3/11 : RUN apt-get install -y apache2
--> Using cache
--> 7cd9d227fcea
Step 4/11 : RUN echo "Hello World!" > /var/www/index.html
--> 088d750c4fc8
Step 5/11 : RUN a2enmod rewrite
--> Using cache
--> f0f1ca2fc120
Step 6/11 : RUN chown -R www-data:www-data /var/www
--> Using cache
--> d9910f2cbdee
Step 7/11 : ENV APACHE_RUN_USER www-data
--> Using cache
--> 519b629ffeb7
Step 8/11 : ENV APACHE_RUN_GROUP www-data
--> Using cache
--> b688ac974c38
Step 9/11 : ENV APACHE_LOG_DIR /var/log/apache2
--> Using cache
--> 598d26089f72
Step 10/11 : EXPOSE 80
--> Using cache
--> 598d26089f72
Step 11/11 : CMD ["/usr/sbin/apache2", "-D", "FOREGROUND"]
--> Using cache
--> b688ac974c38
Successfully tagged hello-world-repo:latest
[ec2-user@ip-172-31-0-93 ~]$ docker images
REPOSITORY          TAG           IMAGE ID            CREATED             SIZE
hello-world-repo   latest        b688ac974c38   34 minutes ago   179MB
ubuntu              12.04         5b117edd0b76   2 years ago      104MB
[ec2-user@ip-172-31-0-93 ~]$ 

```

Note “hello-world-repo” successfully created and tagged “latest” and the same image Id

```
18.144.43.49 (ec2-user) Terminal Sessions View Xserver Tools Games Settings Macros Help Session Servers Tools Games Sessions View Split MultiExec Tunneling Packages Settings Help Quick connect: https://18.144.43.49:80/ec2-user/ [WARNING] Using -password via the CLI is insecure. Use -passwordstdin. [INFO] The password will be stored unencrypted in /root/.ec2-user/docker/config.json. Configure a credential helper to remove this warning. See https://docs.docker.com/engine/reference/commandline/login/#credentials-store

Log: Successed
[ec2-user@ip-172-31-0-93 ~]$ docker build -t hello-world-repo .
Sending build context to Docker daemon 14.34kB
Step 1/11 : FROM alpine:3.12.0
5b117eddb076
Step 2/11 : RUN apt-get update -y
5c44d78319e
Step 3/11 : RUN apt-get install -y apache2
5c44d78319e
Step 4/11 : RUN echo "Hello World!" > /var/www/index.html
5c44d78319e
Step 5/11 : RUN a2enmod rewrite
5c44d78319e
Step 6/11 : RUN a2enconf www -R www-data:www-data /var/www
5c44d78319e
Step 7/11 : ENV APACHE_RUN_USER www-data
5c44d78319e
Step 8/11 : ENV APACHE_RUN_GROUP www-data
5c44d78319e
Step 9/11 : EXPOSE 80
5c44d78319e
Step 10/11 : EXPOSE 80
5c44d78319e
Step 11/11 : CMD ["usr/sbin/apache2", "-D", "+foreground"]
5c44d78319e
Successfully built b668ac07438
Successfully tagged hello-world-repo:latest
[ec2-user@ip-172-31-0-93 ~]$ docker images
REPOSITORY TAG IMAGE ID CREATED SIZE
hello-world-repo latest b668ac07438 38 minutes ago 170MB
hello-world-repo latest b668ac974c38 34 minutes ago 170MB
ubuntu 12.04 5b117eddb076 2 years ago 104MB
[ec2-user@ip-172-31-0-93 ~]$ docker push 758287678061.dkr.ecr.us-east-1.amazonaws.com/hello-world-repo:latest
The push refers to repository [758287678061.dkr.ecr.us-east-1.amazonaws.com/hello-world-repo]
An image does not exist locally with the tag: 758287678061.dkr.ecr.us-east-1.amazonaws.com/hello-world-repo
[ec2-user@ip-172-31-0-93 ~]$ docker images
REPOSITORY TAG IMAGE ID CREATED SIZE
hello-world-repo latest b668ac074c38 38 minutes ago 170MB
hello-world-repo latest b668ac974c38 34 minutes ago 170MB
ubuntu 12.04 5b117eddb076 2 years ago 104MB
[ec2-user@ip-172-31-0-93 ~]$ docker tag hello-world-repo:latest 758287678061.dkr.ecr.us-east-1.amazonaws.com/hello-world-repo:latest
[ec2-user@ip-172-31-0-93 ~]$ 
[ec2-user@ip-172-31-0-93 ~]$ 
```

“docker tag hello-world-repo:latest

758287676861.dkr.ecr.us-east-1.amazonaws.com/hello-world-repo:latest" then checked with "docker images"

“docker push 758287676861.dkr.ecr.us-east-1.amazonaws.com/hello-world-repo:latest” Which will push the docker image (container) to the ECR repository

Amazon ECS

- Clusters
- Task Definitions**
- Account Settings

Amazon EKS

- Clusters
- Amazon ECR
- Repositories
- AWS Marketplace
- Discover software
- Subscriptions

Task Definitions

Task definitions specify the container information for your application, such as how many containers are part of your task, what resources they will use, how they are linked together, and which host ports they will use. [Learn more](#)

[Create new Task Definition](#) [Create new revision](#) [Actions](#)

Last updated on March 5, 2020 12:27:39 PM (0m ago)

Status: ACTIVE INACTIVE
Filter in this page
<input type="checkbox"/> Task Definition Latest revision status
No results

[Feedback](#) [English \(US\)](#)

© 2008 - 2020, Amazon Web Services, Inc. or its affiliates. All rights reserved. [Privacy Policy](#) [Terms of Use](#)

A Task will dictate how resources will be utilized

Create new Task Definition

Step 1: Select launch type compatibility

Step 2: Configure task and container definitions

Select launch type compatibility

Select which launch type you want your task definition to be compatible with based on where you want to launch your task.

FARGATE



Price based on task size

Requires network mode awsvpc

AWS-managed infrastructure, no Amazon EC2 instances to manage

EC2



Price based on resource usage

Multiple network modes available

Self-managed infrastructure using Amazon EC2 instances

*Required

[Cancel](#)

[Next step](#)

[Feedback](#) [English \(US\)](#)

© 2008 - 2020, Amazon Web Services, Inc. or its affiliates. All rights reserved. [Privacy Policy](#) [Terms of Use](#)

Fargate will aid in reducing the managing of resources

Create new Task Definition

Step 1: Select launch type compatibility

Step 2: Configure task and container definitions

Configure task and container definitions

A task definition specifies which containers are included in your task and how they interact with each other. You can also specify data volumes for your containers to use. [Learn more](#)

Task Definition Name* hello-world-task

Requires Compatibilities* FARGATE

Task Role None

Optional IAM role that tasks can use to make API requests to authorized AWS services. Create an Amazon Elastic Container Service Task Role in the IAM Console [Create a role](#)

Network Mode awsvpc

If you choose <default>, ECS will start your container using Docker's default networking mode, which is Bridge on Linux and NAT on Windows. <default> is the only supported mode on Windows.

Task execution IAM role

This role is required by tasks to pull container images and publish container logs to Amazon CloudWatch on your behalf. If you do not have the ecsTaskExecutionRole already, we can create one for you.

Task execution role You are giving permission to Elastic Container Service to create and use ecsTaskExecutionRole.

Task size

The task size allows you to specify a fixed size for your task. Task size is required for tasks using the Fargate launch type and is optional for the EC2 launch type. Container level memory settings are optional when task size is set. Task size is not supported for Windows containers.

Task memory (GB) 0.5GB

The valid memory range for 0.25 vCPU is: 0.5GB - 2GB.

Task CPU (vCPU) 0.25 vCPU

The valid CPU for 0.5 GB memory is: 0.25 vCPU

Task memory maximum allocation for container memory reservation



Task CPU maximum allocation for containers

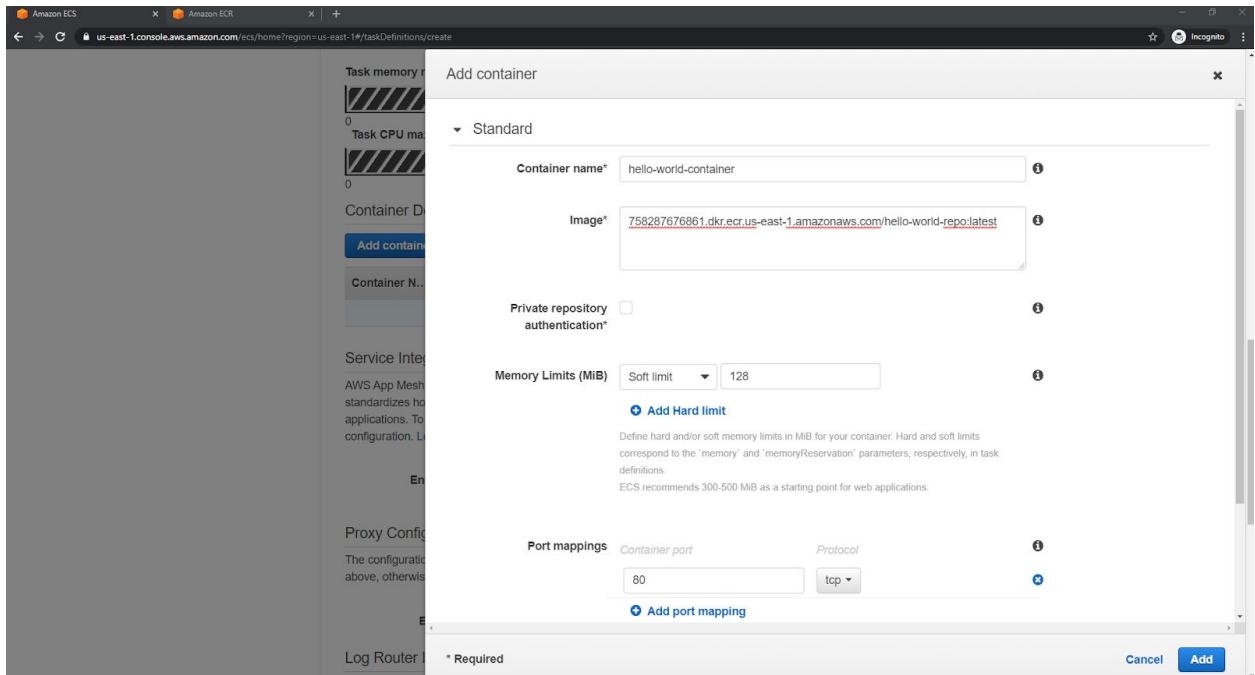


Container Definitions

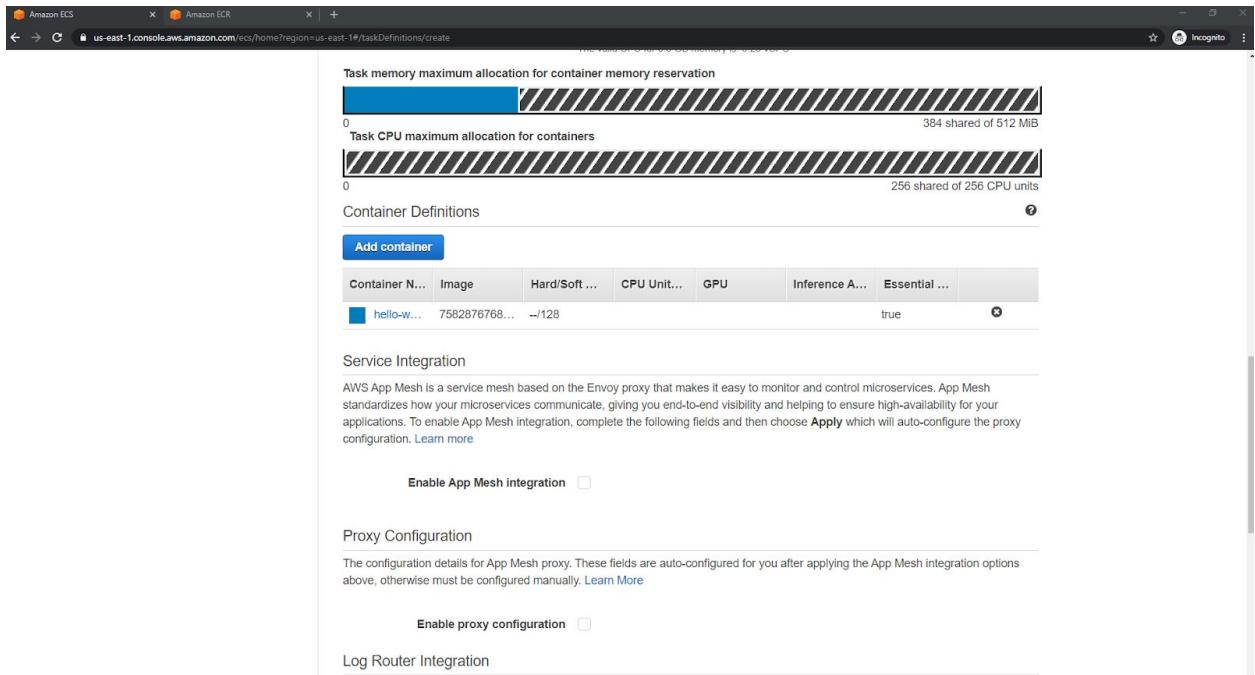
Add container

Container N...	Image	Hard/Soft ...	CPU Unit...	GPU	Inference A...	Essential ...
No results						

Minimum size of memory and CPU



Adding Container details. Note the tag for image found at “docker images” from SSH



Note the Task memory allocation has changed

The configuration details for App Mesh proxy. These fields are auto-configured for you after applying the App Mesh integration options above, otherwise must be configured manually. [Learn More](#)

Enable proxy configuration

Log Router Integration

FireLens for Amazon ECS helps you route logs to an AWS service or AWS Partner Network (APN) destination for log storage and analysis. FireLens works with Fluentd and Fluent Bit. To auto-configure a log router container, complete the following fields and then choose **Apply**. [Learn more](#)

Enable FireLens integration

Volumes

Use a volume configuration to add volumes for use by the containers within a task. To add a volume, choose **Add volume**, complete the fields, and then choose **Add**. [Learn more](#)

Add volume

Configure via JSON

Tags

Key	Value
Add key	Add value

*Required Cancel Previous Create

Leaving the rest at default and create

Launch Status

Task definition status - 3 of 3 completed

Create Execution Role

Execution Role AmazonECSTaskExecutionRole created

Create Task Definition: hello-world-task

hello-world-task succeeded

Create CloudWatch Log Group

CloudWatch Log Group created
CloudWatch Log Group /ecs/hello-world-task

Back View task definition

Feedback English (US)

© 2008 - 2020, Amazon Web Services, Inc. or its affiliates. All rights reserved. [Privacy Policy](#) [Terms of Use](#)

Launch status

Amazon ECS
Clusters
Task Definitions
Account Settings
Amazon EKS
Clusters
Amazon ECR
Repositories
AWS Marketplace
Discover software
Subscriptions ↗

Task Definition: hello-world-task:1

View detailed information for your task definition. To modify the task definition, you need to create a new revision and then make the required changes to the task definition

Create new revision Actions ▾

Builder JSON Tags

Task Definition Name

Task Role None
Optional IAM role that tasks can use to make API requests to authorized AWS services. Create an Amazon Elastic Container Service Task Role in the IAM Console ↗

Network Mode awsvpc
If you choose <default>, ECS will start your container using Docker's default networking mode, which is Bridge on Linux and NAT on Windows. <default> is the only supported mode on Windows.

Compatibilities EC2, FARGATE

Requires compatibilities FARGATE

Task definition: hello-world-task:1

Task execution IAM role

This role is required by tasks to pull container images and publish container logs to Amazon CloudWatch on your behalf. If you do not have the `ecsTaskExecutionRole` already, we can create one for you.

Task execution role `ecsTaskExecutionRole`

Task size

The task size allows you to specify a fixed size for your task. Task size is required for tasks using the Fargate launch type and is optional for the EC2 launch type. Container level memory settings are optional when task size is set. Task size is not supported for Windows containers.

Task memory (MiB)

Task CPU (unit)

Task memory maximum allocation for container memory reservation


Task CPU maximum allocation for containers


Task Placement

Constraint No constraints

Container Definitions

Container Name ...	Image	CPU Units	GPU	Inference Accelerator	Hard/Soft memory limits (MiB)	Essential
hello-world-c...	758287676861.dkr...	0			~128	true

Volumes

Task definition: hello-world-task:1 cont.

Task CPU maximum allocation for containers

Task Placement

Constraint No constraints

Container Definitions

Container Name ...	Image	CPU Units	GPU	Inference Accele...	Hard/Soft memory limits (MiB)	Essential
hello-world-c...	758287676861.dkr...	0			-/-128	true

Volumes

Requires attributes

Name	Value
com.amazonaws.ecs.capability.logging-driver.awslogs	
ecs.capability.execution-role-awslogs	
com.amazonaws.ecs.capability.ecr-auth	
com.amazonaws.ecs.capability.docker-remote-api.1.19	
com.amazonaws.ecs.capability.docker-remote-api.1.21	
ecs.capability.execution-role-ecr-pull	
com.amazonaws.ecs.capability.docker-remote-api.1.18	
ecs.capability.task-eni	

Feedback English (US) © 2008 - 2020, Amazon Web Services, Inc. or its affiliates. All rights reserved. Privacy Policy Terms of Use

Task definition: hello-world-task:1 cont.

Clusters

An Amazon ECS cluster is a regional grouping of one or more container instances on which you can run task requests. Each account receives a default cluster the first time you use the Amazon ECS service. Clusters may contain more than one Amazon EC2 instance type.

For more information, see the [ECS documentation](#).

Create Cluster Get Started

View list card view all refresh

No clusters found

Get Started

Feedback English (US) © 2008 - 2020, Amazon Web Services, Inc. or its affiliates. All rights reserved. Privacy Policy Terms of Use

Clusters will group resources containers together as well as instances

Create Cluster

Step 1: Select cluster template

Step 2: Configure cluster

Select cluster template

The following cluster templates are available to simplify cluster creation. Additional configuration and integrations can be added later.

Networking only

Resources to be created:
Cluster
VPC (optional)
Subnets (optional)

Powered by AWS Fargate

EC2 Linux + Networking

Resources to be created:
Cluster
VPC
Subnets
Auto Scaling group with Linux AMI

EC2 Windows + Networking

Resources to be created:
Cluster
VPC
Subnets
Auto Scaling group with Windows AMI

Because we are using Fargate Networking only is selected

The screenshot shows the 'Create Cluster' wizard on the AWS ECS console. The first step, 'Step 1: Select cluster template', is active. It displays three options:

- Networking only**: This option is selected. It creates a cluster, a VPC (optional), and subnets (optional). It is powered by AWS Fargate.
- EC2 Linux + Networking**: This option creates a cluster, a VPC, subnets, and an Auto Scaling group with a Linux AMI.
- EC2 Windows + Networking**: This option creates a cluster, a VPC, subnets, and an Auto Scaling group with a Windows AMI.

The second step, 'Step 2: Configure cluster', is shown below. It includes fields for the cluster name (set to 'hello-world-cluster'), networking (with an option to create a new VPC), tags, CloudWatch Container Insights (disabled), and a summary of required fields.

Common to run many containers in one cluster.

Launch status

Your container instances are launching, and it may take a few minutes until they are in the running state and ready to access. Usage hours on your new container instances start immediately and continue to accrue until you stop or terminate them.

[Back](#) [View Cluster](#)

ECS status - 1 of 1 complete **hello-world-cluster**



ECS cluster

ECS Cluster hello-world-cluster successfully created

[Feedback](#) [English \(US\)](#)

© 2008 - 2020, Amazon Web Services, Inc. or its affiliates. All rights reserved. [Privacy Policy](#) [Terms of Use](#)

This worked the second time. The first time a role error was shown.

Amazon ECS
Clusters
Task Definitions
Account Settings
Amazon EKS
Clusters
Amazon ECR
Repositories
AWS Marketplace
Discover software
Subscriptions ↗

[Clusters](#) > [hello-world-cluster](#)

Cluster : hello-world-cluster

[Update Cluster](#)

[Delete Cluster](#)

Get a detailed view of the resources on your cluster.

Status **ACTIVE**

Registered container instances 0

Pending tasks count 0 Fargate, 0 EC2

Running tasks count 0 Fargate, 0 EC2

Active service count 0 Fargate, 0 EC2

Draining service count 0 Fargate, 0 EC2

[Services](#) [Tasks](#) [ECS Instances](#) [Metrics](#) [Scheduled Tasks](#) [Tags](#) [Capacity Providers](#)

[Create](#)

[Update](#)

[Delete](#)

[Actions ▾](#)

Last updated on March 5, 2020 1:44:58 PM (0m ago)



[Filter in this page](#)

Launch type ALL

Service type ALL

Service Name

Status

Service type...

Task Definiti...

Desired task...

Running tas...

Launch type... Platform ver...

No results

[Feedback](#) [English \(US\)](#)

© 2008 - 2020, Amazon Web Services, Inc. or its affiliates. All rights reserved. [Privacy Policy](#) [Terms of Use](#)

Creating a service inside cluster hello-world-cluster

Create Service

Step 1: Configure service

Step 2: Configure network
Step 3: Set Auto Scaling (optional)
Step 4: Review

Configure service

A service lets you specify how many copies of your task definition to run and maintain in a cluster. You can optionally use an Elastic Load Balancing load balancer to distribute incoming traffic to containers in your service. Amazon ECS maintains that number of tasks and coordinates task scheduling with the load balancer. You can also optionally use Service Auto Scaling to adjust the number of tasks in your service.

Launch type FARGATE EC2

Task Definition Family

Revision

Platform version

Cluster

Service name

Service type* REPLICA

Number of tasks

Note the number of tasks

us-east-1.console.aws.amazon.com/ecs/home?region=us-east-1#/clusters/hello-world-cluster/createService

Number of tasks

Minimum healthy percent

Maximum percent

Deployments

Choose a deployment option for the service.

Deployment type* Rolling update Blue/green deployment (powered by AWS CodeDeploy)

This sets AWS CodeDeploy as the deployment controller for the service. A CodeDeploy application and deployment group are created automatically with default settings for the service. To change to the rolling update deployment type after the service has been created, you must re-create the service and select the "rolling update" deployment type.

Tagging requires that you opt in to the new ARN and resource ID format.
The IAM user/role has not opted in to the new ARN format. Opt-in to the new format to use this feature. Manage your opt-in settings.

*Required

Feedback English (US) © 2008 - 2020, Amazon Web Services, Inc. or its affiliates. All rights reserved. Privacy Policy Terms of Use

Create Service

Step 1: Configure service
Step 2: Configure network
 Step 3: Set Auto Scaling (optional)
 Step 4: Review

Configure network

VPC and security groups

VPC and security groups are configurable when your task definition uses the awsvpc network mode.

Cluster VPC* vpc-c64509bc (172.31.0.0/16)

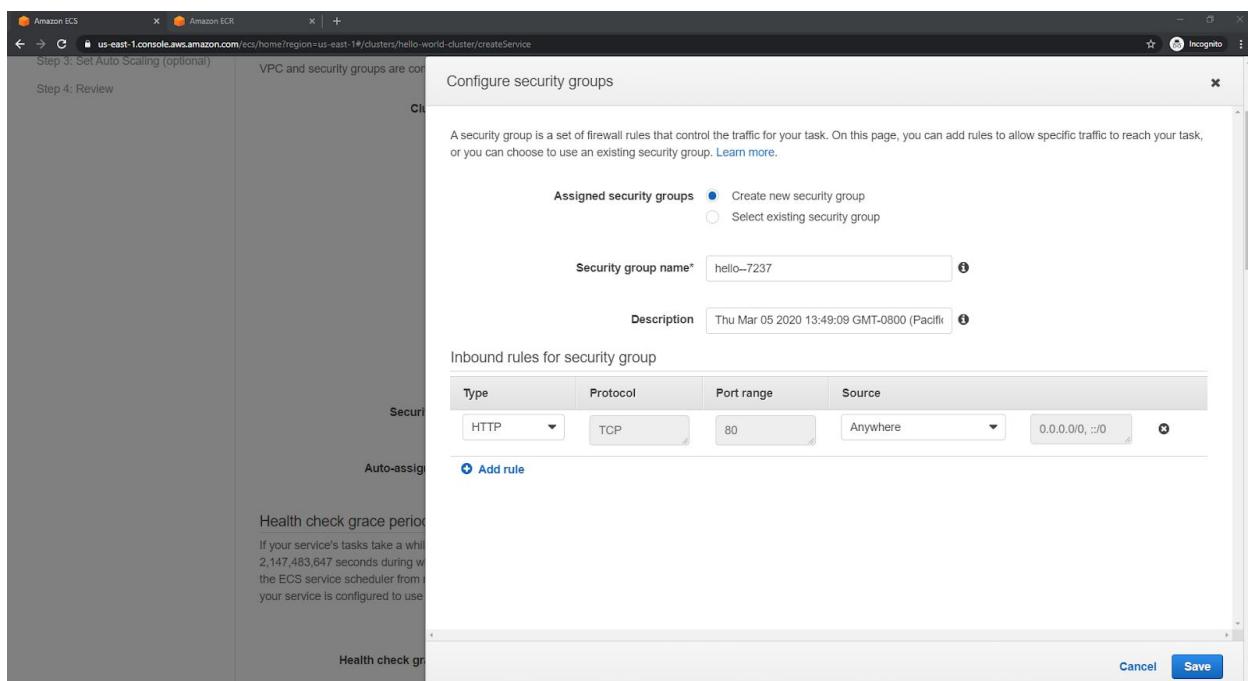
Subnets*

- subnet-d33bbfb9e (172.31.16.0/20) - us-east-1d assign ipv6 on creation: Disabled
- subnet-fbe1a5c5 (172.31.48.0/20) - us-east-1e assign ipv6 on creation: Disabled
- subnet-590fe557 (172.31.64.0/20) - us-east-1f assign ipv6 on creation: Disabled

Security groups* hello-7237 Edit

Auto-assign public IP ENABLED

Use default settings and subnets (Note there are 3 unused subnets available)



Check that port 80 is open

Health check grace period

If your service's tasks take a while to start and respond to ELB health checks, you can specify a health check grace period of up to 2,147,483,647 seconds during which the ECS service scheduler will ignore ELB health check status. This grace period can prevent the ECS service scheduler from marking tasks as unhealthy and stopping them before they have time to come up. This is only valid if your service is configured to use a load balancer.

Health check grace period requires a load balancer. ?

Load balancing

An Elastic Load Balancing load balancer distributes incoming traffic across the tasks running in your service. Choose an existing load balancer, or create a new one in the Amazon EC2 console.

Load balancer type*

None
Your service will not use a load balancer.

Application Load Balancer
Allows containers to use dynamic host port mapping (multiple tasks allowed per container instance). Multiple services can use the same listener port on a single load balancer with rule-based routing and paths.

Network Load Balancer
A Network Load Balancer functions at the fourth layer of the Open Systems Interconnection (OSI) model. After the load balancer receives a request, it selects a target from the target group for the default rule using a flow hash routing algorithm.

Classic Load Balancer
Requires static host port mappings (only one task allowed per container instance); rule-based routing and paths are not supported.

Service IAM role Task definitions that use the awsvpc network mode use the AWSServiceRoleForECS service-linked role, which is created for you automatically. [Learn more](#).

Load balancing is set to None

App Mesh

To use your service with App Mesh, you must

- Ensure your task definition is configured properly. Edit your task definition if you haven't done this.
- Set up your service to use Service Discovery.

Service discovery (optional)

Service discovery uses Amazon Route 53 to create a namespace for your service, which allows it to be discoverable via DNS.

Enable service discovery integration

Namespace* ?

Namespace name* local ?

Configure service discovery service

Create new service discovery service

Select existing service discovery service

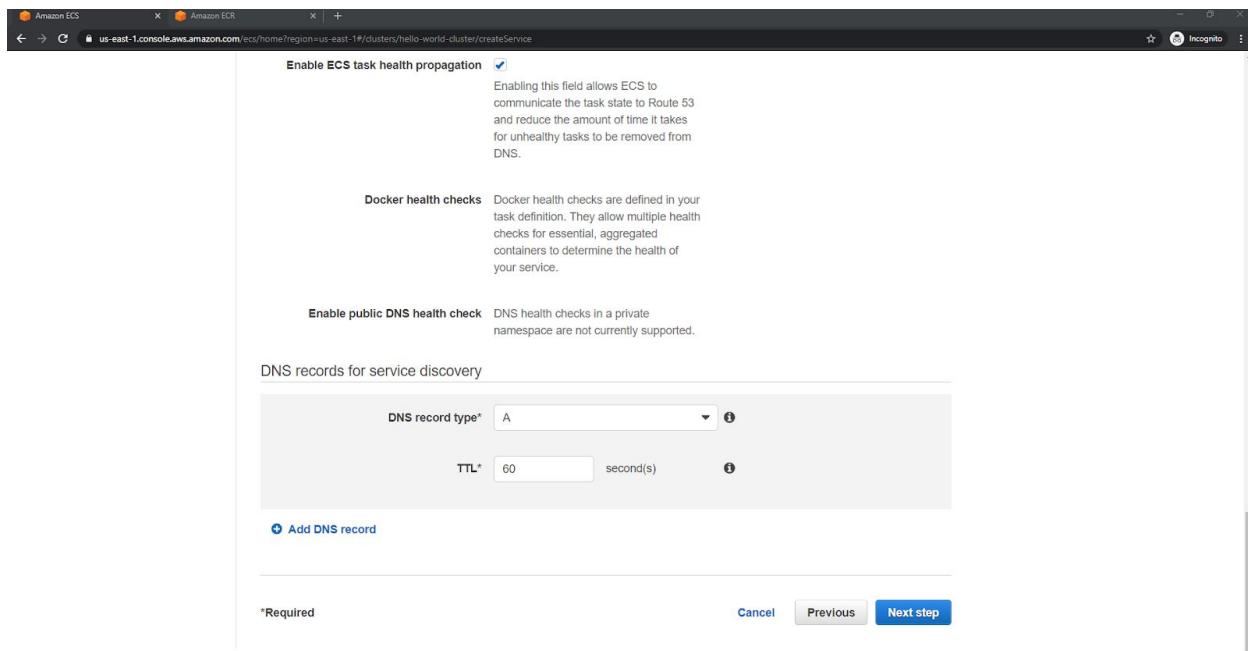
Service discovery name* hello-world-service ?

Alphanumeric and underscores strings, with periods in the middle are valid. For more information, see [Route 53 Auto Naming documentation](#).

Enable ECS task health propagation

Enabling this field allows ECS to

Defaults



Defaults the entire page

Create Service

- Step 1: Configure service
- Step 2: Configure network
- Step 3: Set Auto Scaling (optional)**
- Step 4: Review

Set Auto Scaling (optional)

Automatically adjust your service's desired count up and down within a specified range in response to CloudWatch alarms. You can modify your Service Auto Scaling configuration at any time to meet the needs of your application.

- Service Auto Scaling Do not adjust the service's desired count
 Configure Service Auto Scaling to adjust your service's desired count

*Required

Cancel Previous Next step



Containers based on load

Create Service

Step 1: Configure service
Step 2: Configure network
Step 3: Set Auto Scaling (optional)

Step 4: Review

Review

Edit

Cluster hello-world-cluster
Launch type FARGATE
Task Definition hello-world-task:1
Service name hello-world-service
Service type REPLICA
Number of tasks 1
Minimum healthy percent 100
Maximum percent 200

Configure network

Edit

VPC Id vpc-c64509bc
Subnets subnet-d33fb9e, subnet-fbe1a5c5, subnet-590fe557
Create new security group hello-7237
Auto assign IP ENABLED

Amazon ECS Amazon ECR us-east-1.console.aws.amazon.com/ecs/home?region=us-east-1#clusters/hello-world-cluster/createService

Configure network

Edit

VPC Id vpc-c64509bc
Subnets subnet-d33fb9e, subnet-fbe1a5c5, subnet-590fe557
Create new security group hello-7237
Auto assign IP ENABLED

Configure service discovery

Edit

Namespace local
Service discovery name hello-world-service
Enable ECS task health propagation true
DNS record type and TTL A 60

Set Auto Scaling (optional)

Edit

not configured

Cancel Previous Create Service

Feedback English (US)

© 2008 - 2020, Amazon Web Services, Inc. or its affiliates. All rights reserved. Privacy Policy Terms of Use

Settings before creating service

Launch Status

ECS Service status - 5 of 5 completed

Configure Task Networking

Create security group

- Create security group
hello--7237 succeeded sg-030730c36fc448980

Set inbound rules

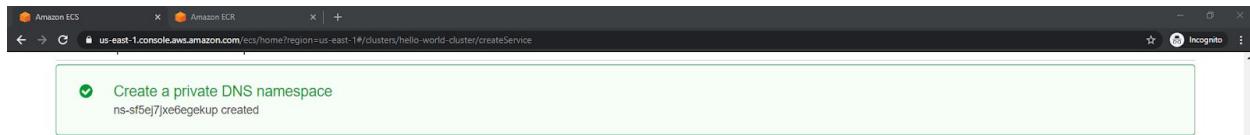
- Set inbound rules
succeeded sg-030730c36fc448980

Configure Service Discovery

Create a private DNS namespace

- Create a private DNS namespace
ns-sf5ej7jxe6gekup created

Create service discovery service



Create service discovery service

- Create service discovery service
arn:aws:servicediscovery:us-east-1:758287676861:service/srv-sqpzofkbi7sbjath created

Create Service

Create service: hello-world-service

- Service created
Service created. Tasks will start momentarily. View: hello-world-service

Additional integrations you can connect to your ECS service

Code Pipeline

Setup a CI/CD process from your service. You can build from source or have an ECR repository as the source for your deployment.

[Create a pipeline](#)

[Back](#) [View Service](#)



Amazon ECS

Clusters

- Task Definitions
- Account Settings

Amazon EKS

Clusters

Amazon ECR

Repositories

AWS Marketplace

Discover software

Subscriptions

Clusters

Clusters > hello-world-cluster > Task: 339b3a33-b92c-4fee-bd5e-2041e240392c

Task : 339b3a33-b92c-4fee-bd5e-2041e240392c

Run more like this Stop

Details Tags Logs

Cluster hello-world-cluster

Launch type FARGATE

Platform version 1.3.0

Task definition hello-world-task:1

Group service:hello-world-service

Task role None

Last status RUNNING

Desired status RUNNING

Created at 2020-03-05 14:01:44 -0800

Started at 2020-03-05 14:02:46 -0800

Network

Network mode awsvpc

ENI Id eni-0e53fcbfd5edac3b8

Subnet Id subnet-d33fb9e

Private IP 172.31.19.82

Public IP 18.212.230.71

Mac address 0a:2e:d2:6e:6e:53

Containers

Service Running



Container active

Amazon ECS

- Clusters**
- Task Definitions
- Account Settings

Amazon EKS

- Clusters
- Amazon ECR
- Repositories
- AWS Marketplace
- Discover software
- Subscriptions ↗

Clusters

An Amazon ECS cluster is a regional grouping of one or more container instances on which you can run task requests. Each account receives a default cluster the first time you use the Amazon ECS service. Clusters may contain more than one Amazon EC2 instance type.

For more information, see the [ECS documentation](#).

[Create Cluster](#) [Get Started](#)

View [list](#) [card](#)

[hello-world-cluster >](#) CloudWatch monitoring
Default Monitoring

FARGATE

1	1	0
Services	Running tasks	Pending tasks
EC2		
0	0	0
CPUUtilization	No data	MemoryUtilization
Container instances	0	

Feedback English (US) © 2008 - 2020, Amazon Web Services, Inc. or its affiliates. All rights reserved. Privacy Policy Terms of Use

Active Cluster

Amazon ECS

- Clusters**
- Task Definitions
- Account Settings

Amazon EKS

- Clusters
- Amazon ECR
- Repositories
- AWS Marketplace
- Discover software
- Subscriptions ↗

Clusters > hello-world-cluster

Cluster : hello-world-cluster

Status **ACTIVE**

Get a detailed view of the resources on your cluster.

[Update Cluster](#) [Delete Cluster](#)

Registered container instances 0

Pending tasks count 0 Fargate, 0 EC2

Running tasks count 1 Fargate, 0 EC2

Active service count 1 Fargate, 0 EC2

Draining service count 0 Fargate, 0 EC2

Services Tasks ECS Instances Metrics Scheduled Tasks Tags Capacity Providers

Create Update Delete Actions ▾ Last updated on March 5, 2020 2:10:07 PM (0m ago)

Filter in this page	Launch type	ALL	Service type	ALL
<input type="checkbox"/> Service Name	Status	Service type...	Task Definiti...	Desired task...
<input type="checkbox"/> hello-world-service	ACTIVE	REPLICA	hello-world-ta...	1
			Running tas...	Running tas...
			Launch type...	Platform ver...
			FARGATE	LATEST(1.3.0)

Feedback English (US) © 2008 - 2020, Amazon Web Services, Inc. or its affiliates. All rights reserved. Privacy Policy Terms of Use

hello-world-cluster services

Amazon ECS

- Clusters**
- Task Definitions
- Account Settings
- Amazon EKS
- Clusters
- Amazon ECR
- Repositories
- AWS Marketplace
- Discover software
- Subscriptions ↗

Clusters > hello-world-cluster > Service: hello-world-service

Service : hello-world-service

Cluster hello-world-cluster **Status** ACTIVE **Task definition** hello-world-task:1 **Service type** REPLICA **Launch type** FARGATE **Platform version** LATEST(1.3.0) **Service role** AWSServiceRoleForECS

Update **Delete**

Details **Tasks** **Events** **Auto Scaling** **Deployments** **Metrics** **Tags** **Logs**

Load Balancing

Load Balancer Name	Container Name	Container Port
No load balancers		

Network Access

Allowed VPC vpc-c64509bc
Allowed subnets subnet-d33fb9e, subnet-fbe1a5c5, subnet-590fe557
Security groups* sg-030730c36fc448980
Auto-assign public IP ENABLED

Service discovery

Service: hello-world-service

Amazon ECS

- Clusters**
- Task Definitions
- Account Settings
- Amazon EKS
- Clusters
- Amazon ECR
- Repositories
- AWS Marketplace
- Discover software
- Subscriptions ↗

Clusters > hello-world-cluster

Cluster : hello-world-cluster

Get a detailed view of the resources on your cluster.

Status ACTIVE

Registered container instances 0
Pending tasks count 0 Fargate, 0 EC2
Running tasks count 1 Fargate, 0 EC2
Active service count 1 Fargate, 0 EC2
Draining service count 0 Fargate, 0 EC2

Services **Tasks** **ECS Instances** **Metrics** **Scheduled Tasks** **Tags** **Capacity Providers**

Run new Task **Stop** **Stop All** **Actions** Last updated on March 5, 2020 2:11:06 PM (0m ago)

Desired task status: **Running** Stopped

Filter in this page		Launch type	ALL	< 1-1 > Page size 50				
Task	Task definition...	Container Inst...	Last status	Desired status...	Started By	Group	Launch type	Platform versl...
339b3a33-b92c-...	hello-world-task:1	--	RUNNING	RUNNING	ecs-svc/914407...	service:hello-wo...	FARGATE	1.3.0

Feedback English (US) © 2008 - 2020, Amazon Web Services, Inc. or its affiliates. All rights reserved. Privacy Policy Terms of Use

Cluster: hello-world-cluster Tasks

Amazon ECS

- Clusters**
- Task Definitions
- Account Settings
- Amazon EKS
- Clusters
- Amazon ECR
- Repositories
- AWS Marketplace
- Discover software
- Subscriptions

Clusters > hello-world-cluster > Task: 339b3a33-b92c-4fee-bd5e-2041e240392c

Task : 339b3a33-b92c-4fee-bd5e-2041e240392c

Run more like this Stop

Details Tags Logs

Cluster hello-world-cluster
Launch type FARGATE
Platform version 1.3.0
Task definition hello-world-task:1
Group service:hello-world-service
Task role None
Last status RUNNING
Desired status RUNNING
Created at 2020-03-05 14:01:44 -0800
Started at 2020-03-05 14:02:46 -0800

Network

Network mode awsvpc
ENI Id eni-0e53fcfd5edac3b8
Subnet Id subnet-d33fb9e
Private IP 172.31.19.82
Public IP 18.212.230.71
Mac address 0a:2e:d2:6e:6e:53

Containers

Amazon ECS us-east-1.console.aws.amazon.com ecs/home?region=us-east-1#clusters/hello-world-cluster/tasks/339b3a33-b92c-4fee-bd5e-2041e240392c/details Cluster: hello-world-cluster

Clusters Launch type FARGATE Platform version 1.3.0 Task definition hello-world-task:1 Group service:hello-world-service Task role None Last status RUNNING Desired status RUNNING Created at 2020-03-05 14:01:44 -0800 Started at 2020-03-05 14:02:46 -0800

Network

Network mode awsvpc
ENI Id eni-0e53fcfd5edac3b8
Subnet Id subnet-d33fb9e
Private IP 172.31.19.82
Public IP 18.212.230.71
Mac address 0a:2e:d2:6e:6e:53

Containers

Last updated on March 5, 2020 2:13:49 PM (0m ago)

Name	Container Runtime I...	Status ...	Image	Image Digest	CPU U...	Hard/S...	Essenti...	Resour...
hello-world...	1ed74d92c59ba14435...	RUNNING	758287676861.dkr.ecr.us-east-1.a...	sha256:2fafd33b10d5c57a5134bd...	0	~128	true	4e19f95...

Feedback English (US) © 2008 - 2020, Amazon Web Services, Inc. or its affiliates. All rights reserved. Privacy Policy Terms of Use

Task running inside hello-world-cluster