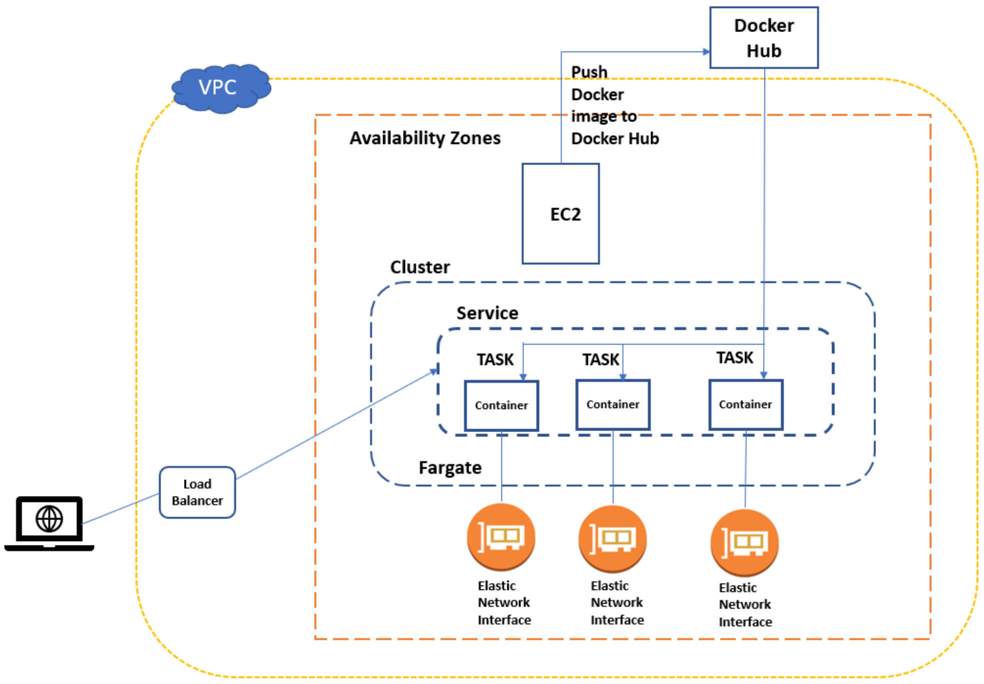
Deploying a Web Application to ECS

Deploying a Java web application on AWS Elastic Container Service(ECS). The web application will be bundled as a Docker image running on Apache Tomcat.

FINAL OUTCOME

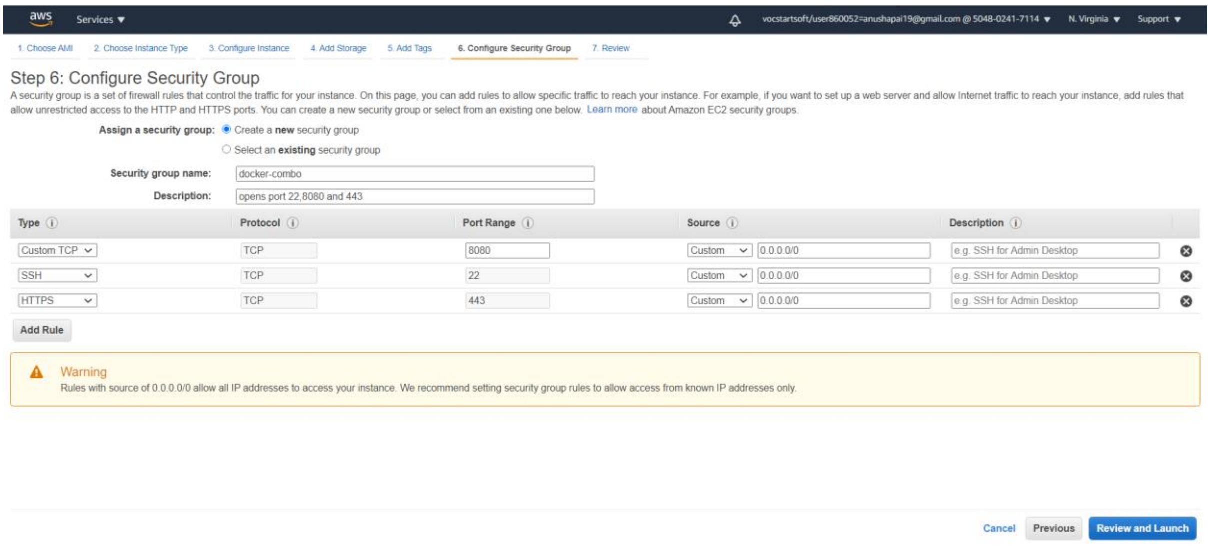


SOLUTION

1) Create an EC2 instance(SG: 22,80,443 and 8080), Install Docker and download the WAR file from https://storage.googleapis.com/skl-training/aws-codelabs/aws-intro/HelloWorld.war to /opt/helloworld
2) Create a Docker Image using DockerFile with Tomcat:JRE8 version Base image
3) Once the image is created, we will run a container using this image and validate the Java web application using EC2 instance Public_IP and Public_IP/HelloWorld
4) As ECR service is unavailable in AWS Educate account, we will sign up to Docker Hub and create a public repository to push the image
5) Using this docker image and ECS Fargate we will create a task, cluster and services

STEPS:

1)Create an Ubuntu 18.04 instance



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.

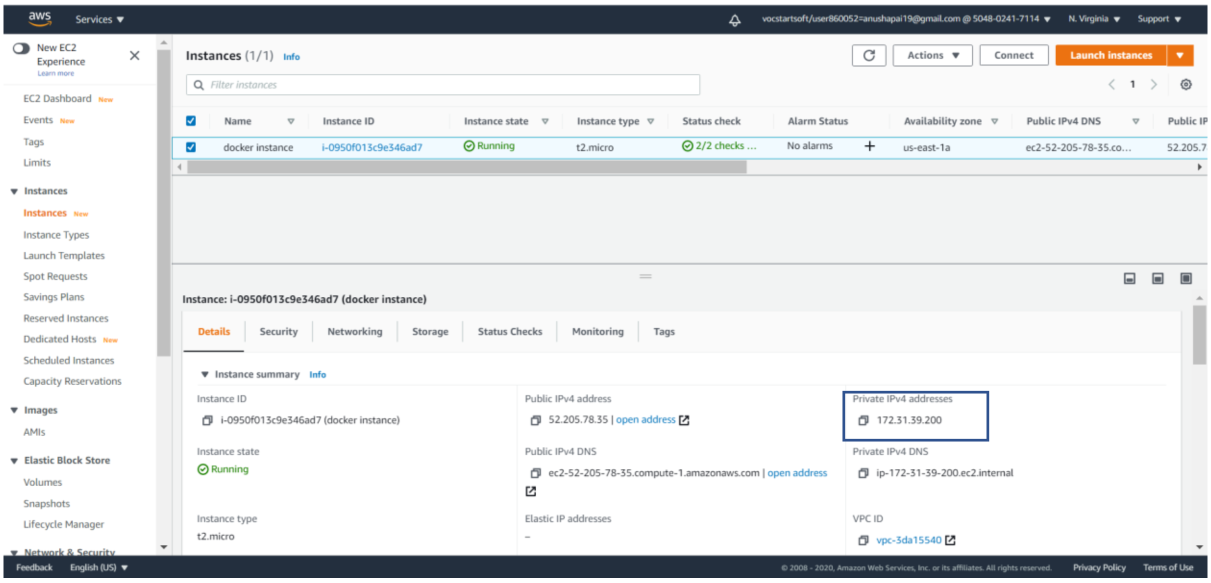
Assign a security group: [✓ Create a **new** security group] Select an **existing** security group

Security group name: docker-combo

Description: opens port 22,8080 and 443

Type	Protocol	Port Range	Source		Description
Custom TCP	TCP	8080	Custom	0.0.0.0/0	e.g. SSH for Admin Desktop
SSH	TCP	22	Custom	0.0.0.0/0	e.g. SSH for Admin Desktop
HTTPS	TCP	443	Custom	0.0.0.0/0	e.g. SSH for Admin Desktop

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.



Instances (1/1) Info

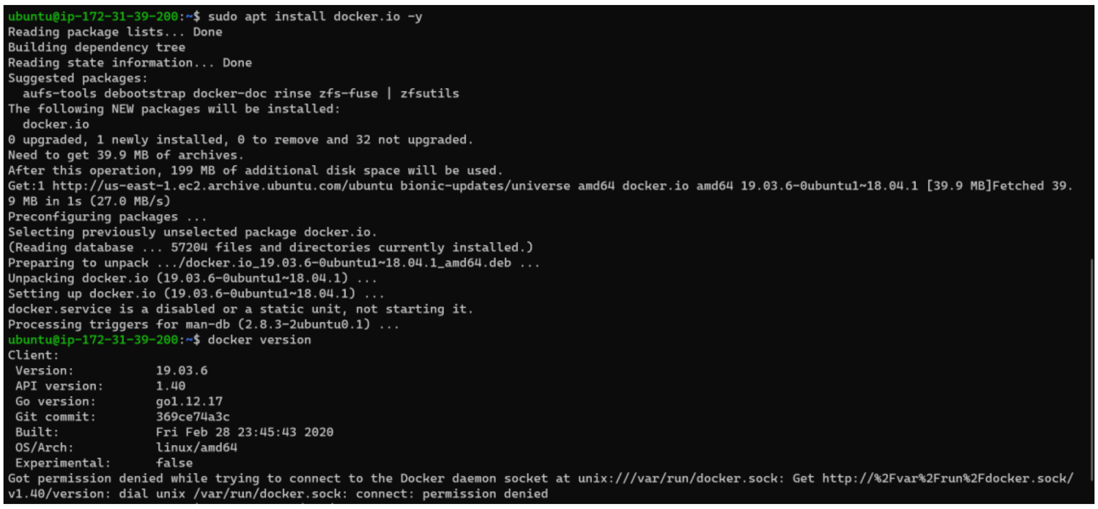
Name	Instance ID	Instance state	Instance type	Status check	Alarm Status	Availability zone	Public IPv4 DNS	Public IP
docker instance	i-0950f013c9e346ad7	Running	t2.micro	2/2 checks ...	No alarms	us-east-1a	ec2-52-205-78-35.co...	52.205.7

Instance: i-0950f013c9e346ad7 (docker instance)

Instance summary Info

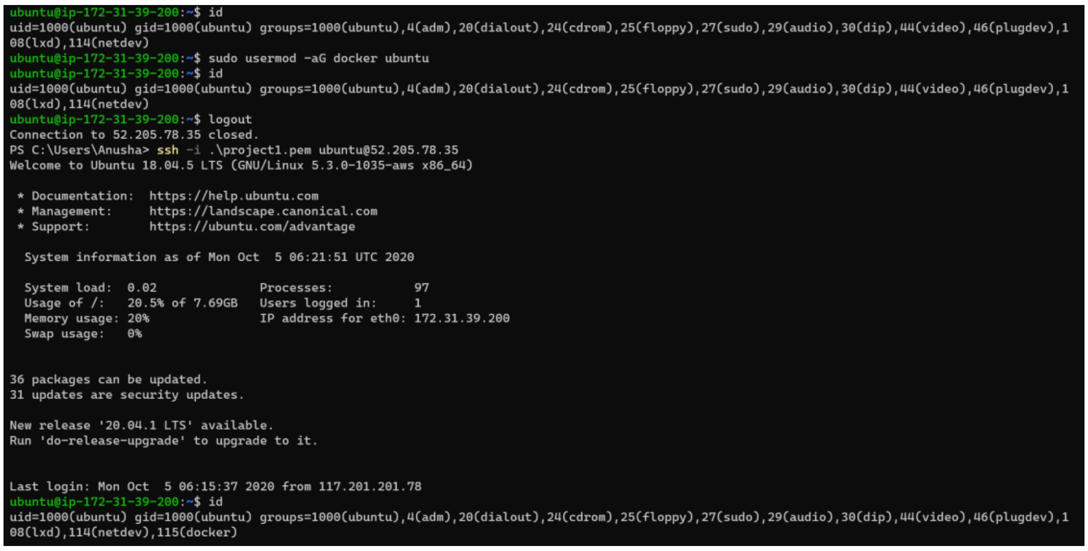
Instance ID i-0950f013c9e346ad7 (docker instance)	Public IPv4 address 52.205.78.35 | open address	Private IPv4 addresses 172.31.39.200
Instance state Running	Public IPv4 DNS ec2-52-205-78-35.compute-1.amazonaws.com | open address	Private IPv4 DNS ip-172-31-39-200.ec2.internal
Instance type t2.micro	Elastic IP addresses –	VPC ID vpc-3da15540

2)Log in to the EC2 instance and install Docker Client



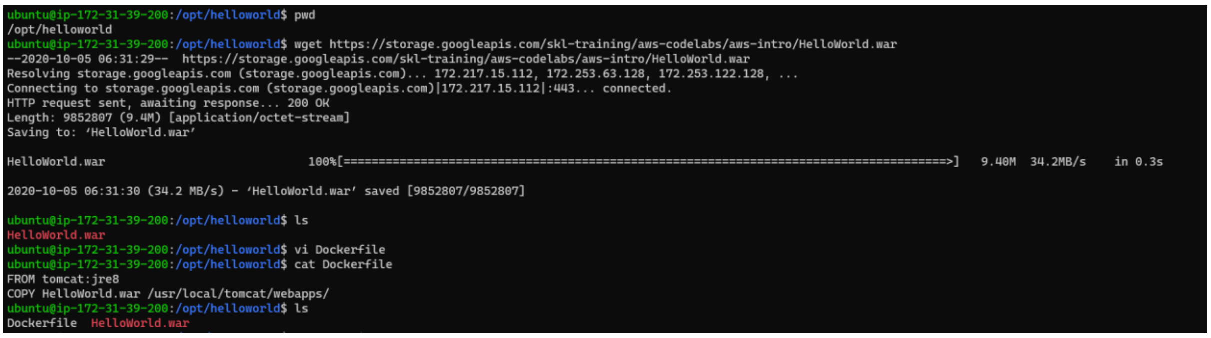
```
ubuntu@ip-172-31-39-200:~$ sudo apt install docker.io -y
Reading package lists... Done
Building dependency tree
Reading state information... Done
Suggested packages:
  aufs-tools debootstrap docker-doc rinse zfs-fuse | zfsutils
The following NEW packages will be installed:
  docker.io
0 upgraded, 1 newly installed, 0 to remove and 32 not upgraded.
Need to get 39.9 MB of archives.
After this operation, 199 MB of additional disk space will be used.
Get:1 http://us-east-1.ec2.archive.ubuntu.com/ubuntu bionic-updates/universe amd64 docker.io amd64 19.03.6-0ubuntu1~18.04.1 [39.9 MB]Fetched 39.
9 MB in 1s (27.0 MB/s)
Preconfiguring packages ...
Selecting previously unselected package docker.io.
(Reading database ... 57204 files and directories currently installed.)
Preparing to unpack .../docker.io_19.03.6-0ubuntu1~18.04.1_amd64.deb ...
Unpacking docker.io (19.03.6-0ubuntu1~18.04.1) ...
Setting up docker.io (19.03.6-0ubuntu1~18.04.1) ...
docker.service is a disabled or a static unit, not starting it.
Processing triggers for man-db (2.8.3-2ubuntu0.1) ...
ubuntu@ip-172-31-39-200:~$ docker version
Client:
 Version:           19.03.6
 API version:       1.40
 Go version:        go1.12.17
 Git commit:        369ce74a3c
 Built:             Fri Feb 28 23:45:43 2020
 OS/Arch:           linux/amd64
 Experimental:      false
Got permission denied while trying to connect to the Docker daemon socket at unix:///var/run/docker.sock: Get http://%2Fvar%2Frun%2Fdocker.sock/
v1.40/version: dial unix /var/run/docker.sock: connect: permission denied
```

Add the docker group as the supplementary group for Ubuntu user



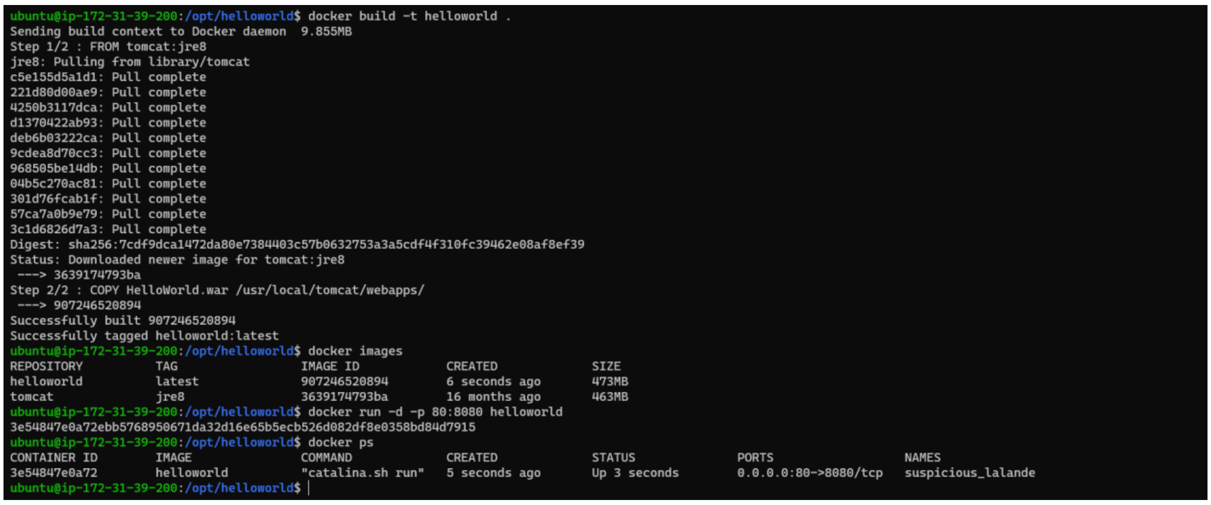
```
ubuntu@ip-172-31-39-200:~$ id
uid=1000(ubuntu) gid=1000(ubuntu) groups=1000(ubuntu),4(adm),20(dialout),24(cdrom),25(floppy),27(sudo),29(audio),30(dip),44(video),46(plugdev),1
08(lxd),114(netdev)
ubuntu@ip-172-31-39-200:~$ sudo usermod -aG docker ubuntu
ubuntu@ip-172-31-39-200:~$ id
uid=1000(ubuntu) gid=1000(ubuntu) groups=1000(ubuntu),4(adm),20(dialout),24(cdrom),25(floppy),27(sudo),29(audio),30(dip),44(video),46(plugdev),1
08(lxd),114(netdev)
ubuntu@ip-172-31-39-200:~$ logout
Connection to 52.205.78.35 closed.
PS C:\Users\Anusha> ssh -i .\project1.pem ubuntu@52.205.78.35
Welcome to Ubuntu 18.04.5 LTS (GNU/Linux 5.3.0-1035-aws x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:     https://landscape.canonical.com
 * Support:        https://ubuntu.com/advantage

  System information as of Mon Oct  5 06:21:51 UTC 2020

  System load:  0.02              Processes:           97
  Usage of /:   20.5% of 7.69GB   Users logged in:     1
  Memory usage: 20%               IP address for eth0: 172.31.39.200
  Swap usage:   0%


36 packages can be updated.
31 updates are security updates.

New release '20.04.1 LTS' available.
Run 'do-release-upgrade' to upgrade to it.


Last login: Mon Oct  5 06:15:37 2020 from 117.201.201.78
ubuntu@ip-172-31-39-200:~$ id
uid=1000(ubuntu) gid=1000(ubuntu) groups=1000(ubuntu),4(adm),20(dialout),24(cdrom),25(floppy),27(sudo),29(audio),30(dip),44(video),46(plugdev),1
08(lxd),114(netdev),115(docker)
```

Download the WAR file and create a Dockerfile to create a custom docker image to server our java web application



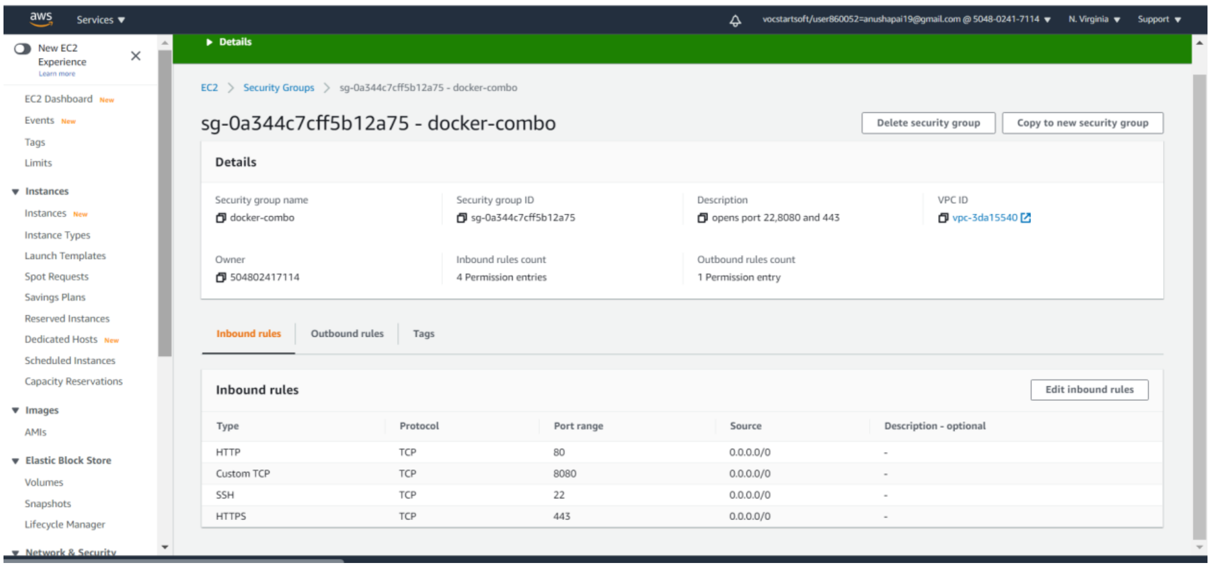
```
ubuntu@ip-172-31-39-200:/opt/helloworld$ pwd
/opt/helloworld
ubuntu@ip-172-31-39-200:/opt/helloworld$ wget https://storage.googleapis.com/skl-training/aws-codelabs/aws-intro/HelloWorld.war
--2020-10-05 06:31:29--  https://storage.googleapis.com/skl-training/aws-codelabs/aws-intro/HelloWorld.war
Resolving storage.googleapis.com (storage.googleapis.com)... 172.217.15.112, 172.253.63.128, 172.253.122.128, ...
Connecting to storage.googleapis.com (storage.googleapis.com)|172.217.15.112|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 9852807 (9.4M) [application/octet-stream]
Saving to: 'HelloWorld.war'

HelloWorld.war                    100%[===================================================================================>]   9.40M  34.2MB/s    in 0.3s

2020-10-05 06:31:30 (34.2 MB/s) - 'HelloWorld.war' saved [9852807/9852807]

ubuntu@ip-172-31-39-200:/opt/helloworld$ ls
HelloWorld.war
ubuntu@ip-172-31-39-200:/opt/helloworld$ vi Dockerfile
ubuntu@ip-172-31-39-200:/opt/helloworld$ cat Dockerfile
FROM tomcat:jre8
COPY HelloWorld.war /usr/local/tomcat/webapps/
ubuntu@ip-172-31-39-200:/opt/helloworld$ ls
Dockerfile  HelloWorld.war
```

3) Create a custom Docker Image using our Dockerfile and tag it as "helloworld" , validate the image has been created and run a container using this image in the detached mode serving the java web application on Host port 80



```
ubuntu@ip-172-31-39-200:/opt/helloworld$ docker build -t helloworld .
Sending build context to Docker daemon  9.855MB
Step 1/2 : FROM tomcat:jre8
jre8: Pulling from library/tomcat
c5e155d5a1d1: Pull complete
221d80d00ae9: Pull complete
4250b3117dca: Pull complete
d1370422ab93: Pull complete
deb6b03222ca: Pull complete
9cdea8d70cc3: Pull complete
968505be14db: Pull complete
04b5c270ac81: Pull complete
301d76fcab1f: Pull complete
57ca7a0b9e79: Pull complete
3c1d6826d7a3: Pull complete
Digest: sha256:7cdf9dca1472da80e7384403c57b0632753a3a5cdf4f310fc39462e08af8ef39
Status: Downloaded newer image for tomcat:jre8
 ---> 3639174793ba
Step 2/2 : COPY HelloWorld.war /usr/local/tomcat/webapps/
 ---> 907246520894
Successfully built 907246520894
Successfully tagged helloworld:latest
ubuntu@ip-172-31-39-200:/opt/helloworld$ docker images
REPOSITORY          TAG                 IMAGE ID            CREATED             SIZE
helloworld          latest              907246520894        6 seconds ago       473MB
tomcat              jre8                3639174793ba        16 months ago       463MB
ubuntu@ip-172-31-39-200:/opt/helloworld$ docker run -d -p 80:8080 helloworld
3e54847e0a72ebb5768950671da32d16e65b5ecb526d082df8e0358bd84d7915
ubuntu@ip-172-31-39-200:/opt/helloworld$ docker ps
CONTAINER ID        IMAGE               COMMAND             CREATED             STATUS              PORTS                  NAMES
3e54847e0a72        helloworld          "catalina.sh run"   5 seconds ago       Up 3 seconds        0.0.0.0:80->8080/tcp   suspicious_lalande
ubuntu@ip-172-31-39-200:/opt/helloworld$ |
```

Allow port 80 in the Security Group attached to the EC2 instance as this port is used as Host port for accessing the web application outside the container



sg-0a344c7cff5b12a75 - docker-combo

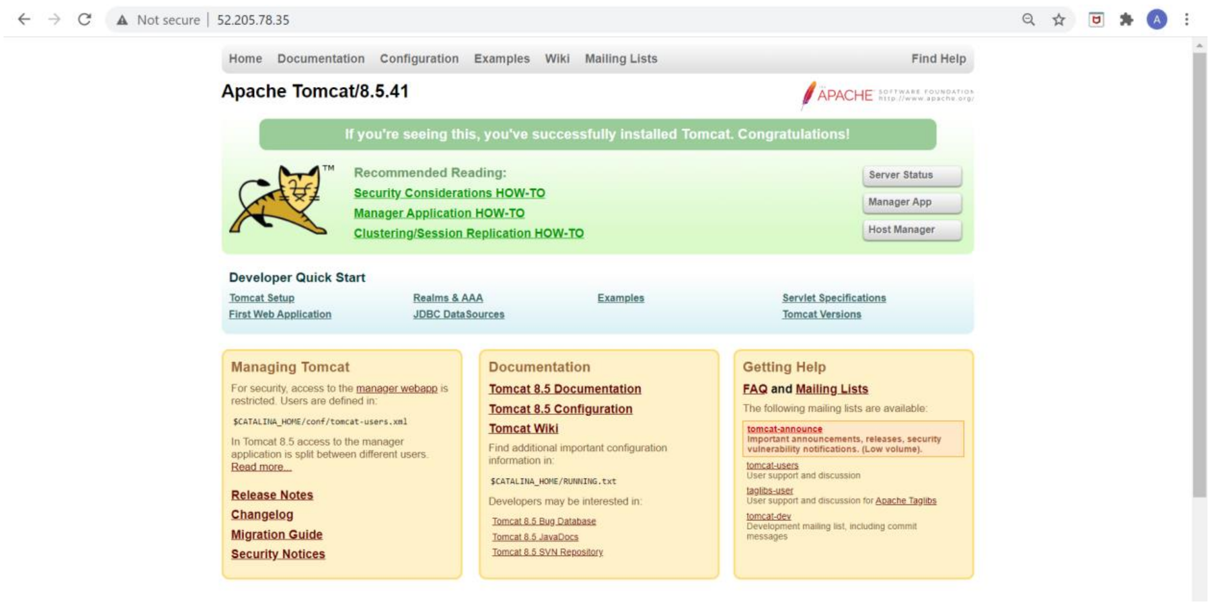
Details

Security group name	Security group ID	Description	VPC ID
docker-combo	sg-0a344c7cff5b12a75	opens port 22,8080 and 443	vpc-3da15540
Owner	Inbound rules count	Outbound rules count	
504802417114	4 Permission entries	1 Permission entry	

Inbound rules

Type	Protocol	Port range	Source	Description - optional
HTTP	TCP	80	0.0.0.0/0	-
Custom TCP	TCP	8080	0.0.0.0/0	-
SSH	TCP	22	0.0.0.0/0	-
HTTPS	TCP	443	0.0.0.0/0	-

Hit the EC2 public IP and validate if the default tomcat web page is served



Apache Tomcat/8.5.41

If you're seeing this, you've successfully installed Tomcat. Congratulations!

Recommended Reading:
Security Considerations HOW-TO
Manager Application HOW-TO
Clustering/Session Replication HOW-TO

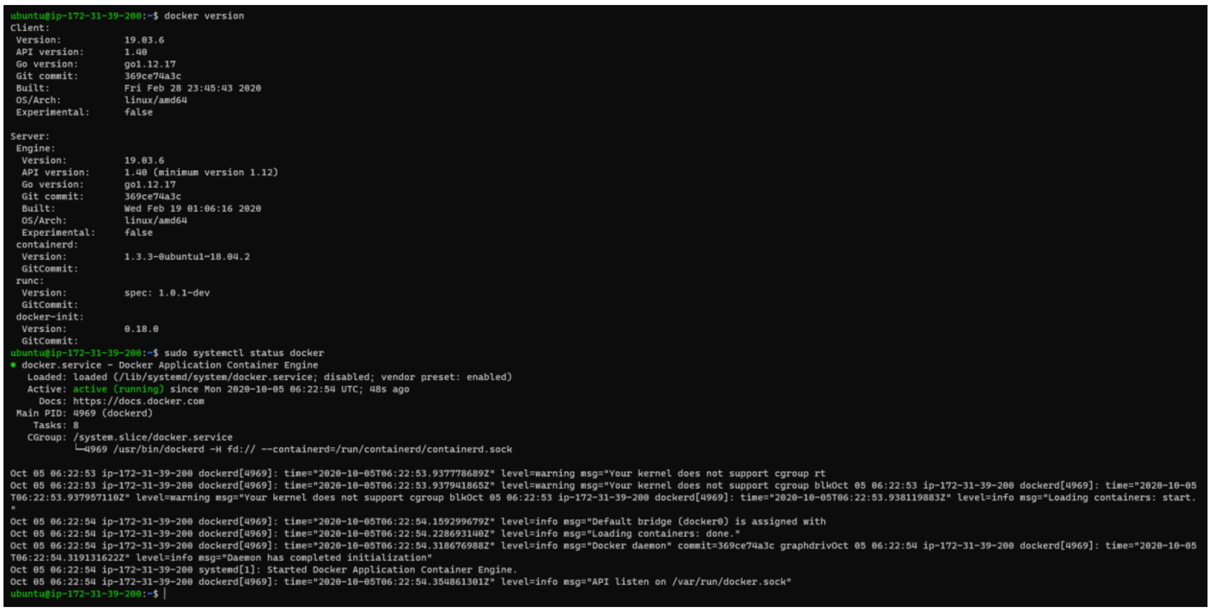
Developer Quick Start
Tomcat Setup
First Web Application
Realms & AAA
JDBC DataSources
Examples
Servlet Specifications
Tomcat Versions

Managing Tomcat
For security, access to the manager webapp is restricted. Users are defined in:
```
$CATALINA_HOME/conf/tomcat-users.xml
```
In Tomcat 8.5 access to the manager application is split between different users. Read more...

Release Notes
Changelog
Migration Guide
Security Notices

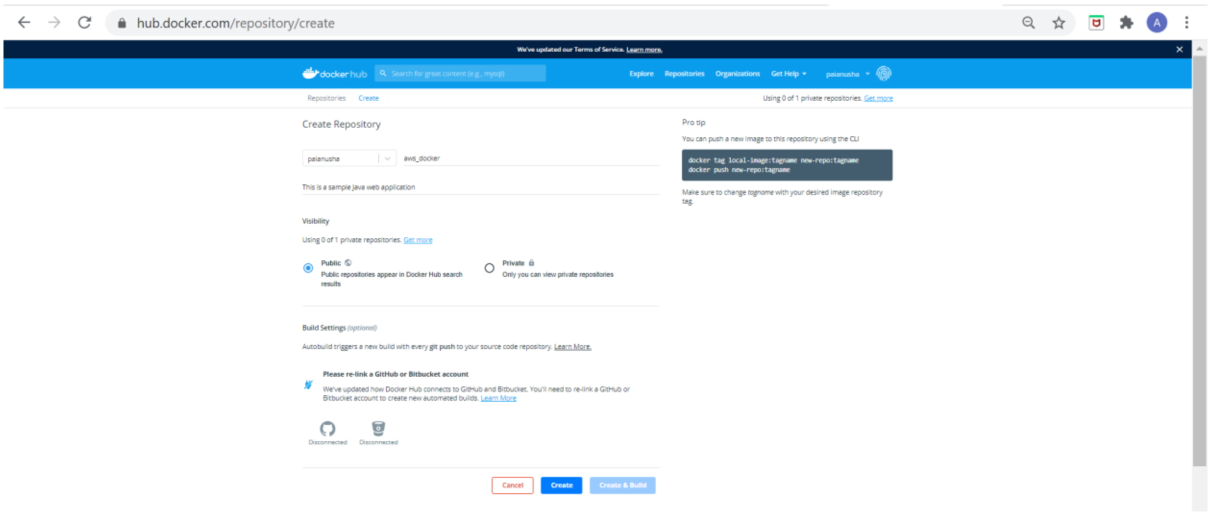
Documentation
Tomcat 8.5 Documentation
Tomcat 8.5 Configuration
Tomcat Wiki
Find additional important configuration information in:
```
$CATALINA_HOME/RUNNING.txt
```
Developers may be interested in:
Tomcat 8.5 Bug Database
Tomcat 8.5 JavaDocs
Tomcat 8.5 SVN Repository

Getting Help
FAQ and Mailing Lists
The following mailing lists are available:
tomcat-announce
Important announcements, releases, security vulnerability notifications. (Low volume).
tomcat-users
User support and discussion
taglibs-user
User support and discussion for Apache Taglibs
tomcat-dev
Development mailing list, including commit messages



```
ubuntu@ip-172-31-39-200:~$ docker version
Client:
 Version:           19.03.6
 API version:       1.40
 Go version:        go1.12.17
 Git commit:        369ce74a3c
 Built:             Fri Feb 28 23:45:43 2020
 OS/Arch:           linux/amd64
 Experimental:      false

Server:
 Engine:
  Version:          19.03.6
  API version:      1.40 (minimum version 1.12)
  Go version:       go1.12.17
  Git commit:       369ce74a3c
  Built:            Wed Feb 19 01:06:16 2020
  OS/Arch:          linux/amd64
  Experimental:     false
 containerd:
  Version:          1.3.3-0ubuntu1~18.04.2
  GitCommit:
 runc:
  Version:          spec: 1.0.1-dev
  GitCommit:
 docker-init:
  Version:          0.18.0
  GitCommit:
ubuntu@ip-172-31-39-200:~$ sudo systemctl status docker
● docker.service - Docker Application Container Engine
   Loaded: loaded (/lib/systemd/system/docker.service; disabled; vendor preset: enabled)
   Active: active (running) since Mon 2020-10-05 06:22:54 UTC; 48s ago
     Docs: https://docs.docker.com
 Main PID: 4969 (dockerd)
    Tasks: 8
   CGroup: /system.slice/docker.service
           └─4969 /usr/bin/dockerd -H fd:// --containerd=/run/containerd/containerd.sock

Oct 05 06:22:53 ip-172-31-39-200 dockerd[4969]: time="2020-10-05T06:22:53.937778689Z" level=warning msg="Your kernel does not support cgroup rt
Oct 05 06:22:53 ip-172-31-39-200 dockerd[4969]: time="2020-10-05T06:22:53.937941865Z" level=warning msg="Your kernel does not support cgroup blkOct 05 06:22:53 ip-172-31-39-200 dockerd[4969]: time="2020-10-05
T06:22:53.937957110Z" level=warning msg="Your kernel does not support cgroup blkOct 05 06:22:53 ip-172-31-39-200 dockerd[4969]: time="2020-10-05T06:22:53.938119883Z" level=info msg="Loading containers: start.
"
Oct 05 06:22:54 ip-172-31-39-200 dockerd[4969]: time="2020-10-05T06:22:54.159299679Z" level=info msg="Default bridge (docker0) is assigned with
Oct 05 06:22:54 ip-172-31-39-200 dockerd[4969]: time="2020-10-05T06:22:54.228693140Z" level=info msg="Loading containers: done."
Oct 05 06:22:54 ip-172-31-39-200 dockerd[4969]: time="2020-10-05T06:22:54.318676988Z" level=info msg="Docker daemon" commit=369ce74a3c graphdrivOct 05 06:22:54 ip-172-31-39-200 dockerd[4969]: time="2020-10-05
T06:22:54.319131622Z" level=info msg="Daemon has completed initialization"
Oct 05 06:22:54 ip-172-31-39-200 systemd[1]: Started Docker Application Container Engine.
Oct 05 06:22:54 ip-172-31-39-200 dockerd[4969]: time="2020-10-05T06:22:54.354861301Z" level=info msg="API listen on /var/run/docker.sock"
ubuntu@ip-172-31-39-200:~$ |
```

4) Sign in to Docker Hub and create a repo to host our docker image



Using 0 of 1 private repositories. Get more

Create Repository

paianusha aws_docker

This is a sample Java web application

Visibility

Using 0 of 1 private repositories. Get more

[✓ Public] Public repositories appear in Docker Hub search results

Private Only you can view private repositories

Build Settings (optional)

Autobuild triggers a new build with every git push to your source code repository. Learn More.

Please re-link a GitHub or Bitbucket account

We've updated how Docker Hub connects to GitHub and Bitbucket. You'll need to re-link a GitHub or Bitbucket account to create new automated builds. Learn More

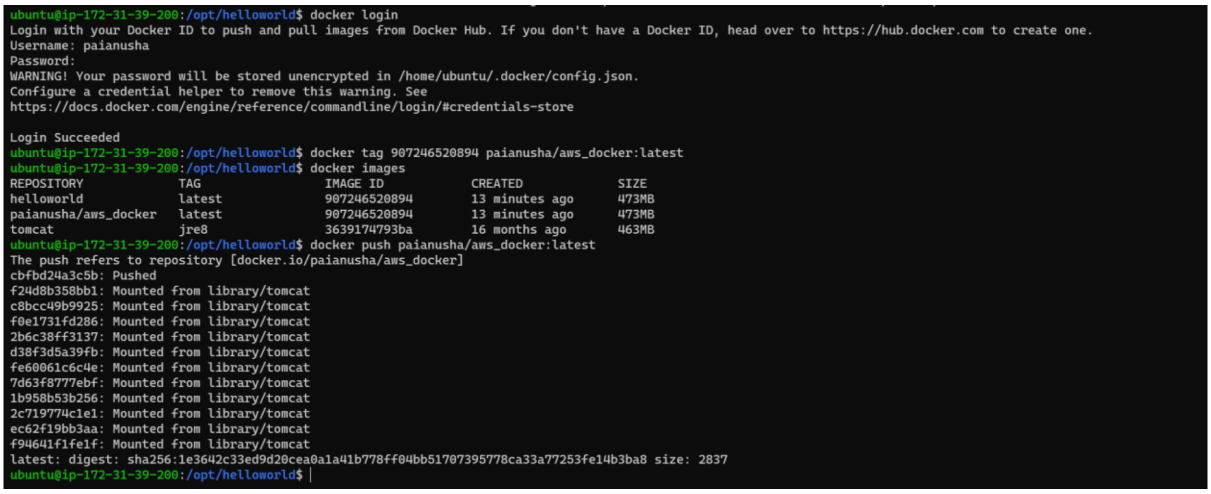
Pro tip

You can push a new image to this repository using the CLI

```
docker tag local-image:tagname new-repo:tagname
docker push new-repo:tagname
```

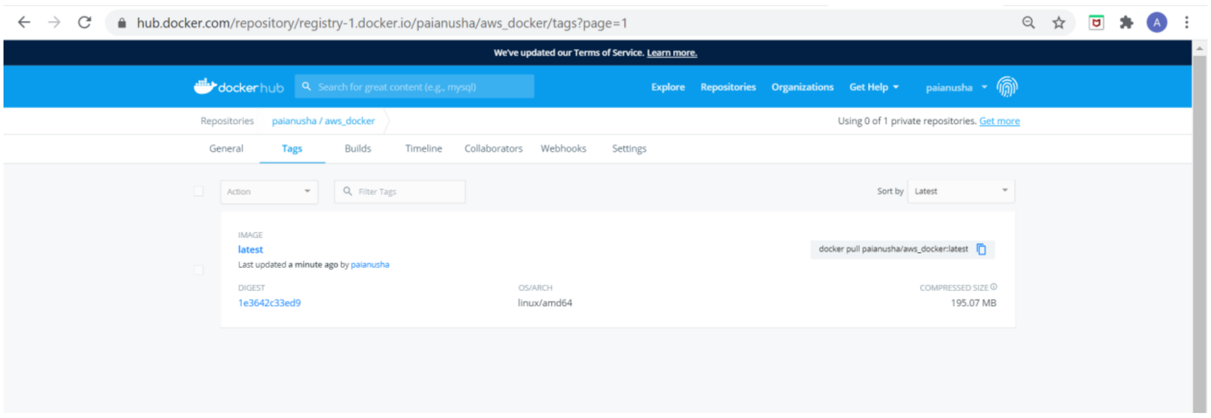
Make sure to change *tagname* with your desired image repository tag.

Login to Docker Hub in our EC2 instance and Push the local docker image to Docker hub



```
ubuntu@ip-172-31-39-200:/opt/helloworld$ docker login
Login with your Docker ID to push and pull images from Docker Hub. If you don't have a Docker ID, head over to https://hub.docker.com to create one.
Username: paianusha
Password:
WARNING! Your password will be stored unencrypted in /home/ubuntu/.docker/config.json.
Configure a credential helper to remove this warning. See
https://docs.docker.com/engine/reference/commandline/login/#credentials-store

Login Succeeded
ubuntu@ip-172-31-39-200:/opt/helloworld$ docker tag 907246520894 paianusha/aws_docker:latest
ubuntu@ip-172-31-39-200:/opt/helloworld$ docker images
REPOSITORY             TAG                 IMAGE ID            CREATED             SIZE
helloworld             latest              907246520894        13 minutes ago      473MB
paianusha/aws_docker   latest              907246520894        13 minutes ago      473MB
tomcat                 jre8                3639174793ba        16 months ago       463MB
ubuntu@ip-172-31-39-200:/opt/helloworld$ docker push paianusha/aws_docker:latest
The push refers to repository [docker.io/paianusha/aws_docker]
cbfbd24a3c5b: Pushed
f24d8b358bb1: Mounted from library/tomcat
c8bcc49b9925: Mounted from library/tomcat
f0e1731fd286: Mounted from library/tomcat
2b6c38ff3137: Mounted from library/tomcat
d38f3d5a39fb: Mounted from library/tomcat
fe60061c6c4e: Mounted from library/tomcat
7d63f8777ebf: Mounted from library/tomcat
1b958b53b256: Mounted from library/tomcat
2c719774c1e1: Mounted from library/tomcat
ec62f19bb3aa: Mounted from library/tomcat
f94641f1fe1f: Mounted from library/tomcat
latest: digest: sha256:1e3642c33ed9d20cea0a1a41b778ff04bb51707395778ca33a77253fe14b3ba8 size: 2837
ubuntu@ip-172-31-39-200:/opt/helloworld$ |
```

Validate the pushed image is available in our repository


hub.docker.com/repository/registry-1.docker.io/paianusha/aws_docker/tags?page=1

We've updated our Terms of Service. Learn more.

Repositories / paianusha / aws_docker — Using 0 of 1 private repositories. Get more

Tabs: General, Tags, Builds, Timeline, Collaborators, Webhooks, Settings

Sort by: Latest

IMAGE
latest
Last updated a minute ago by paianusha

```
docker pull paianusha/aws_docker:latest
```

DIGEST	OS/ARCH	COMPRESSED SIZE
1e3642c33ed9	linux/amd64	195.07 MB

5)We will spin up a ECS cluster using Fargate service

Provide Container Definition

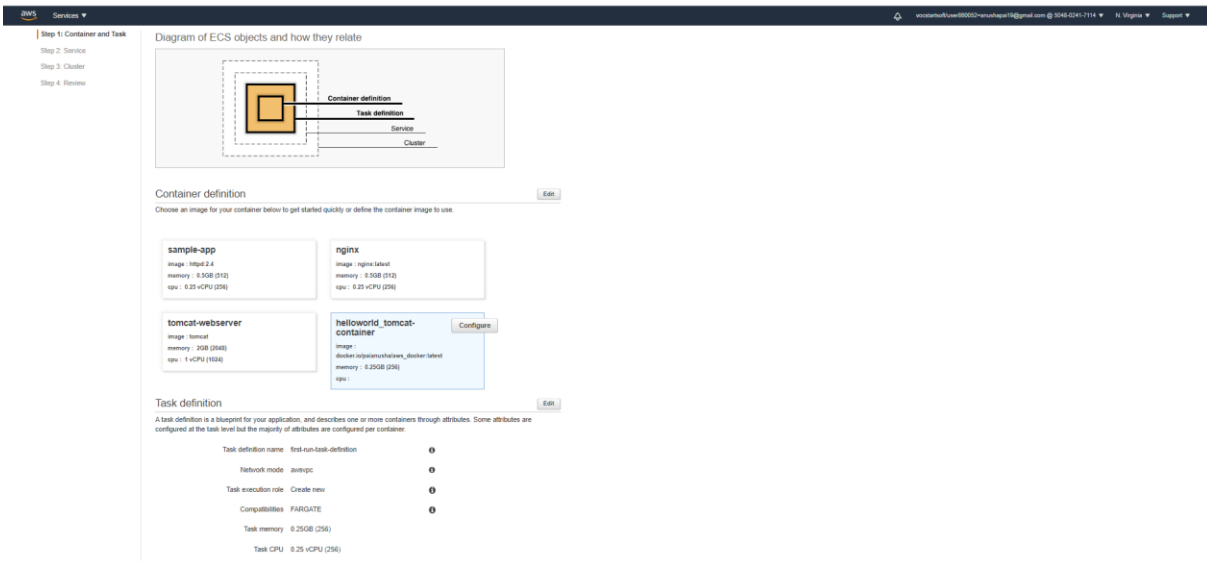


Diagram of ECS objects and how they relate

Container definition

Choose an image for your container below to get started quickly or define the container image to use.

sample-app
image : httpd:2.4
memory : 0.5GB (512)
cpu : 0.25 vCPU (256)

nginx
image : nginx:latest
memory : 0.5GB (512)
cpu : 0.25 vCPU (256)

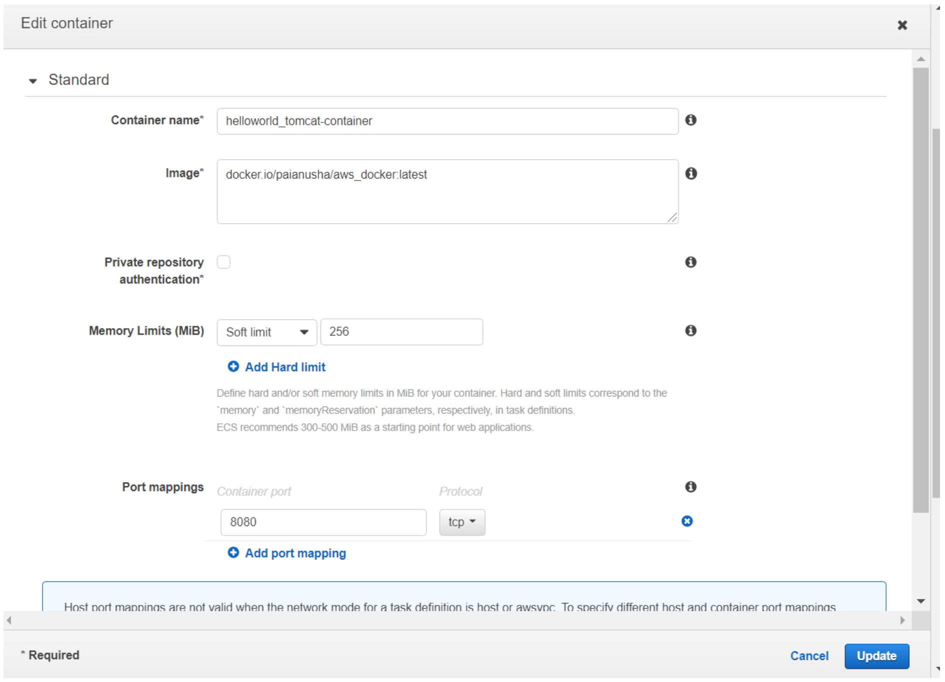
tomcat-webserver
image : tomcat
memory : 2GB (2048)
cpu : 1 vCPU (1024)

helloworld_tomcat-container
image : docker.io/paianusha/aws_docker:latest
memory : 0.25GB (256)
cpu :

Task definition

A task definition is a blueprint for your application, and describes one or more containers through attributes. Some attributes are configured at the task level but the majority of attributes are configured per container.

Task definition name	first-run-task-definition
Network mode	awsvpc
Task execution role	Create new
Compatibilities	FARGATE
Task memory	0.25GB (256)
Task CPU	0.25 vCPU (256)



Container name*	helloworld_tomcat-container
Image*	docker.io/paianusha/aws_docker:latest
Private repository authentication*	☐
Memory Limits (MiB)	Soft limit 256

Define hard and/or soft memory limits in MiB for your container. Hard and soft limits correspond to the `memory` and `memoryReservation` parameters, respectively, in task definitions.
ECS recommends 300-500 MiB as a starting point for web applications.

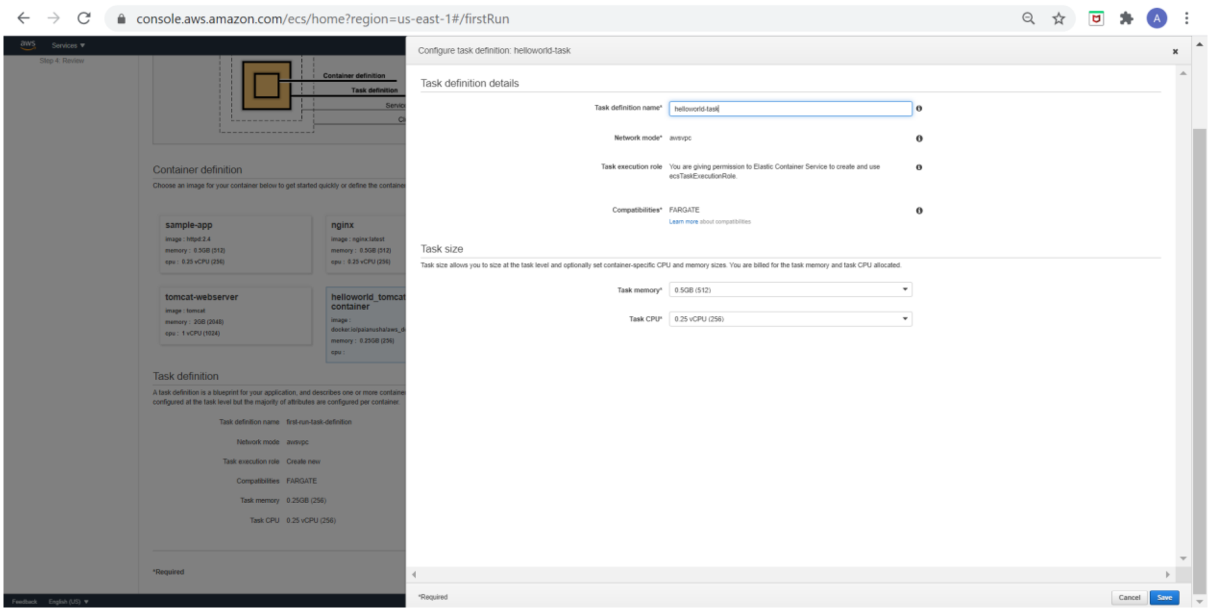
Port mappings

Container port	Protocol
8080	tcp

Host port mappings are not valid when the network mode for a task definition is host or awsvpc. To specify different host and container port mappings

* Required

Provide Task Definition



console.aws.amazon.com/ecs/home?region=us-east-1#/firstRun

Configure task definition: helloworld-task

Task definition details

Task definition name*: helloworld-task

Network mode*: awsvpc

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

Compatibilities*: FARGATE

Learn more about compatibilities

Task size

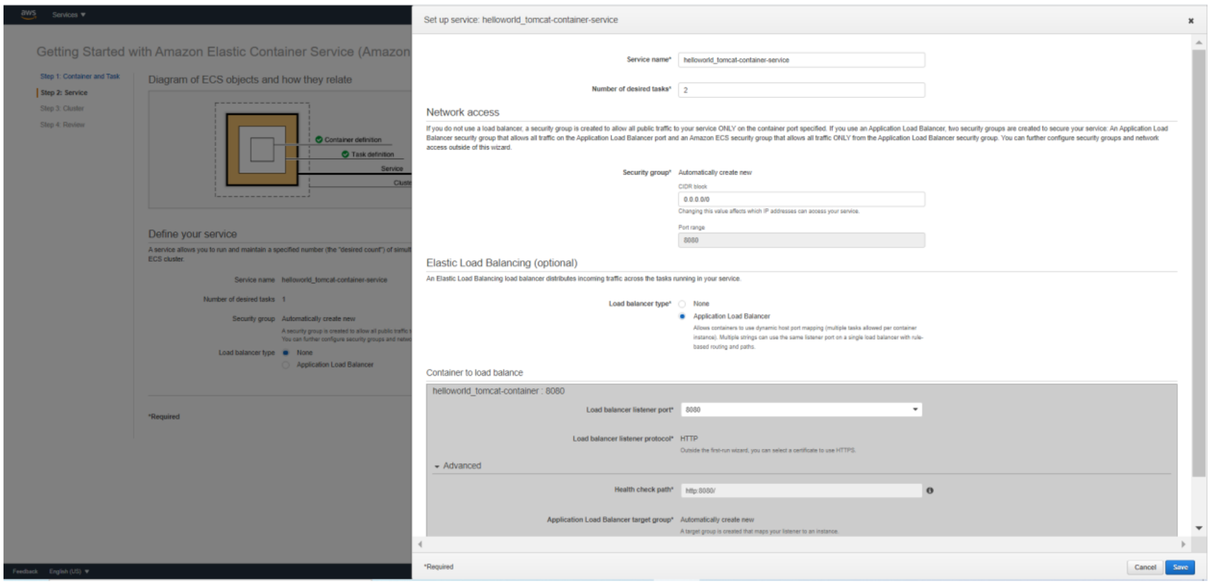
Task size allows you to size at the task level and optionally set container-specific CPU and memory sizes. You are billed for the task memory and task CPU allocated.

Task memory*: 0.5GB (512)

Task CPU*: 0.25 vCPU (256)

*Required

Provide Service Definition



Set up service: helloworld_tomcat-container-service

Service name*: helloworld_tomcat-container-service

Number of desired tasks*: 2

Network access

If you do not use a load balancer, a security group is created to allow all public traffic to your service ONLY on the container port specified. If you use an Application Load Balancer, two security groups are created to secure your service: An Application Load Balancer security group that allows all traffic on the Application Load Balancer port and an Amazon ECS security group that allows all traffic ONLY from the Application Load Balancer security group. You can further configure security groups and network access outside of this wizard.

Security group*: Automatically create new

CIDR block: 0.0.0.0/0

Changing this value affects which IP addresses can access your service.

Port range: 8080

Elastic Load Balancing (optional)

An Elastic Load Balancing load balancer distributes incoming traffic across the tasks running in your service.

Load balancer type*: None [✓ Application Load Balancer]

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

Container to load balance

helloworld_tomcat-container : 8080

Load balancer listener port*: 8080

Load balancer listener protocol*: HTTP

Outside the first-run wizard, you can select a certificate to use HTTPS.

Advanced

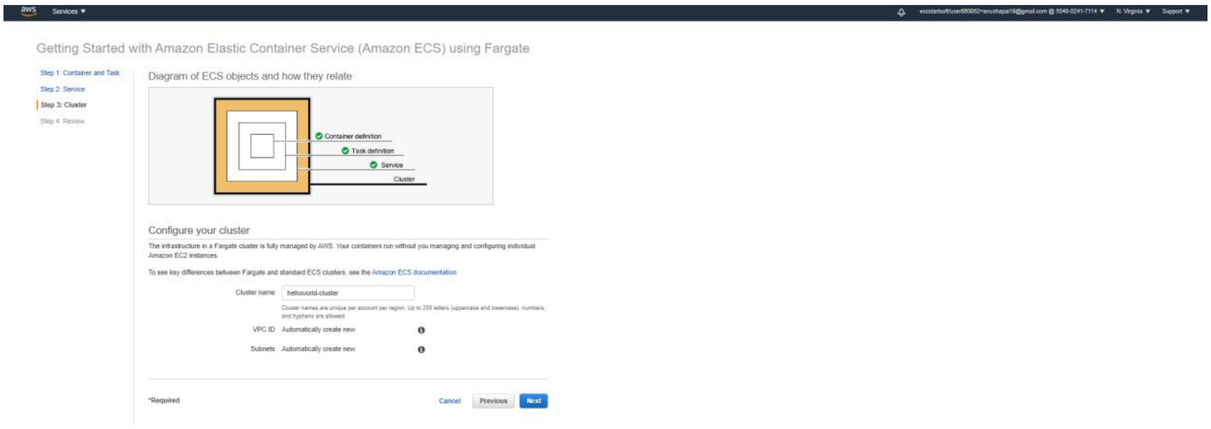
Health check path*: http:8080/

Application Load Balancer target group*: Automatically create new

A target group is created that maps your listener to an instance.

*Required

Provide Cluster Definition



Getting Started with Amazon Elastic Container Service (Amazon ECS) using Fargate

Step 1: Container and Task
Step 2: Service
Step 3: Cluster
Step 4: Review

Diagram of ECS objects and how they relate

Configure your cluster

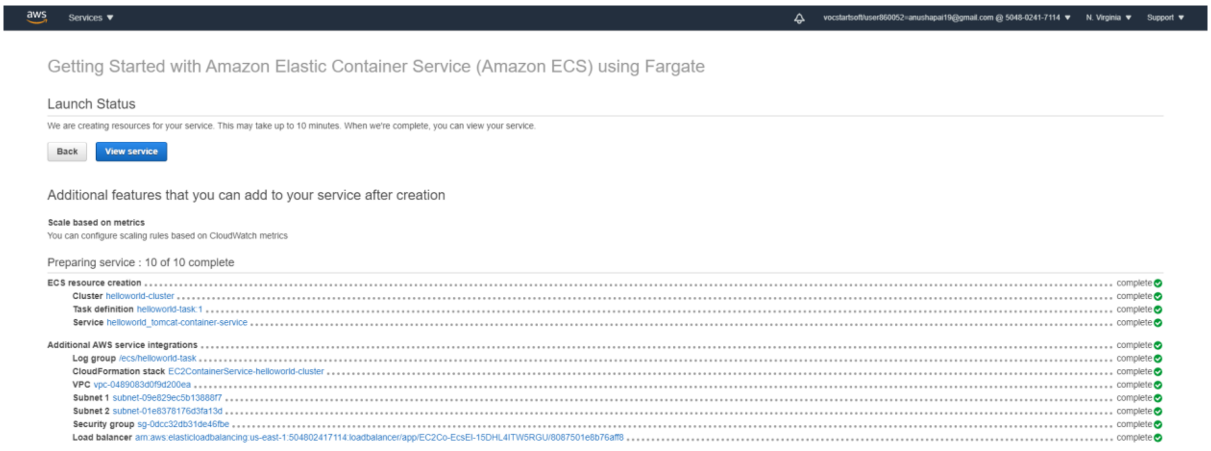
The infrastructure in a Fargate cluster is fully managed by AWS. Your containers run without you managing and configuring individual Amazon EC2 instances.

To see key differences between Fargate and standard ECS clusters, see the Amazon ECS documentation.

Cluster name	helloworld-cluster
	Cluster names are unique per account per region. Up to 255 letters (uppercase and lowercase), numbers, and hyphens are allowed.
VPC ID	Automatically create new
Subnets	Automatically create new

*Required

Validate the status of the cluster creation



Getting Started with Amazon Elastic Container Service (Amazon ECS) using Fargate

Launch Status

We are creating resources for your service. This may take up to 10 minutes. When we're complete, you can view your service.

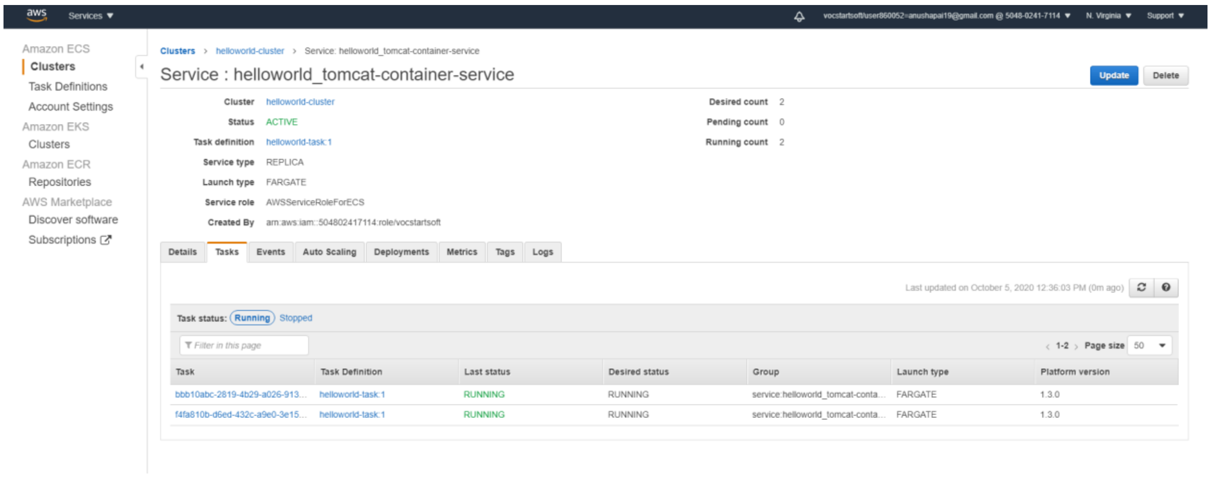
Additional features that you can add to your service after creation

Scale based on metrics
You can configure scaling rules based on CloudWatch metrics

Preparing service : 10 of 10 complete

```
ECS resource creation ............................................................ complete
    Cluster helloworld-cluster ................................................... complete
    Task definition helloworld-task:1 ............................................ complete
    Service helloworld_tomcat-container-service .................................. complete

Additional AWS service integrations .............................................. complete
    Log group /ecs/helloworld-task ............................................... complete
    CloudFormation stack EC2ContainerService-helloworld-cluster .................. complete
    VPC vpc-0489083d0f9d200ea .................................................... complete
    Subnet 1 subnet-09e829ec5b13888f7 ............................................ complete
    Subnet 2 subnet-01e8378176d3fa13d ............................................ complete
    Security group sg-0dcc32db31de46fbe .......................................... complete
    Load balancer arn:aws:elasticloadbalancing:us-east-1:504802417114:loadbalancer/app/EC2Co-EcsEl-15DHL4ITW5RGU/8087501e8b76aff8 ... complete
```

Ensure the desired number of tasks are running



Clusters > helloworld-cluster > Service: helloworld_tomcat-container-service

Service : helloworld_tomcat-container-service

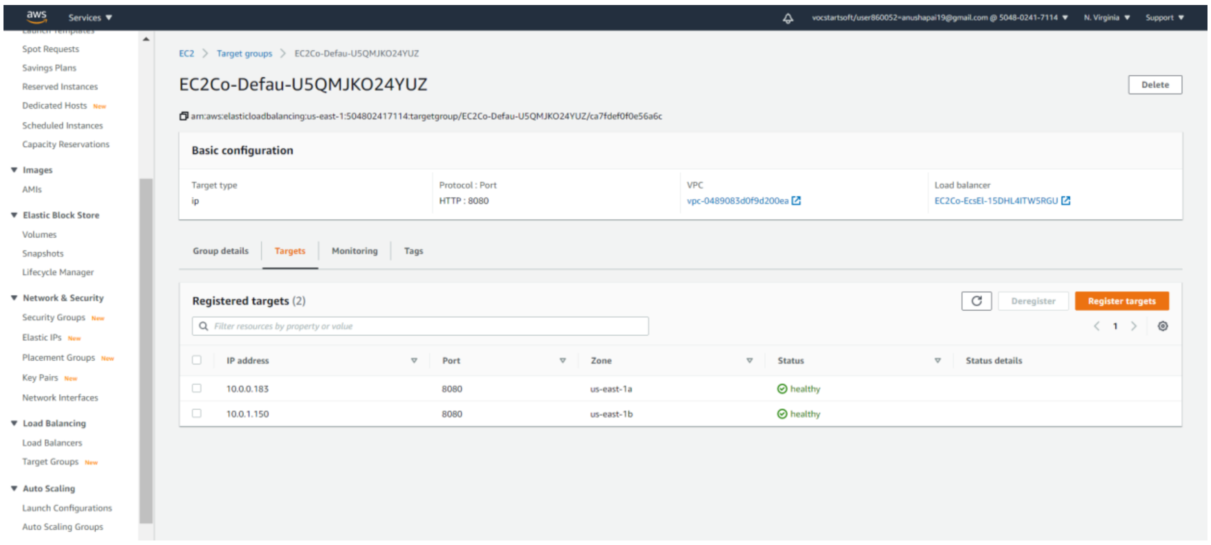
Cluster	helloworld-cluster	Desired count	2
Status	ACTIVE	Pending count	0
Task definition	helloworld-task:1	Running count	2
Service type	REPLICA		
Launch type	FARGATE		
Service role	AWSServiceRoleForECS		
Created By	arn:aws:iam::504802417114:role/vocstartsoft		

Last updated on October 5, 2020 12:36:03 PM (0m ago)

Task status: Running Stopped

Task	Task Definition	Last status	Desired status	Group	Launch type	Platform version
bbb10abc-2819-4b29-a026-913...	helloworld-task:1	RUNNING	RUNNING	service:helloworld_tomcat-conta...	FARGATE	1.3.0
f4fa810b-d6ed-432c-a9e0-3e15...	helloworld-task:1	RUNNING	RUNNING	service:helloworld_tomcat-conta...	FARGATE	1.3.0

Check the instance health state in the Target Group



vocstartsoft/user860052=anushapai19@gmail.com @ 5048-0241-7114 N. Virginia Support

EC2 > Target groups > EC2Co-Defau-U5QMJKO24YUZ

EC2Co-Defau-U5QMJKO24YUZ

arn:aws:elasticloadbalancing:us-east-1:504802417114:targetgroup/EC2Co-Defau-U5QMJKO24YUZ/ca7fdef0f0e56a6c

Basic configuration

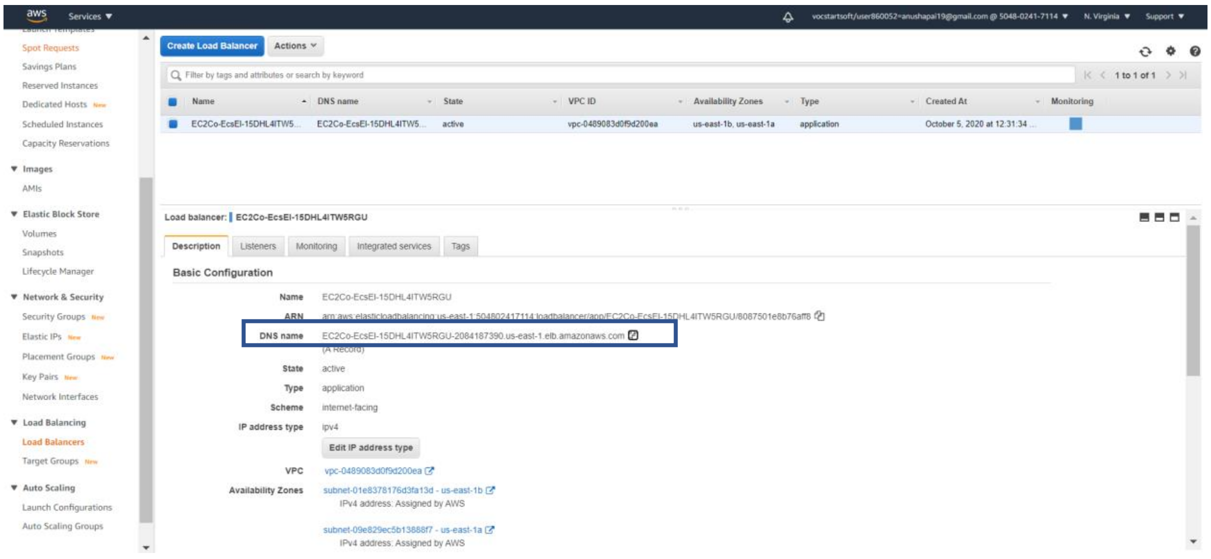
Target type	Protocol : Port	VPC	Load balancer
ip	HTTP : 8080	vpc-0489083d0f9d200ea	EC2Co-EcsEl-15DHL4ITW5RGU

Registered targets (2)

Filter resources by property or value

IP address	Port	Zone	Status	Status details
10.0.0.183	8080	us-east-1a	healthy	
10.0.1.150	8080	us-east-1b	healthy	

Grab the LB DNS and hit it to verify our Java web application is being served



vocstartsoft/user860052=anushapai19@gmail.com @ 5048-0241-7114 N. Virginia Support

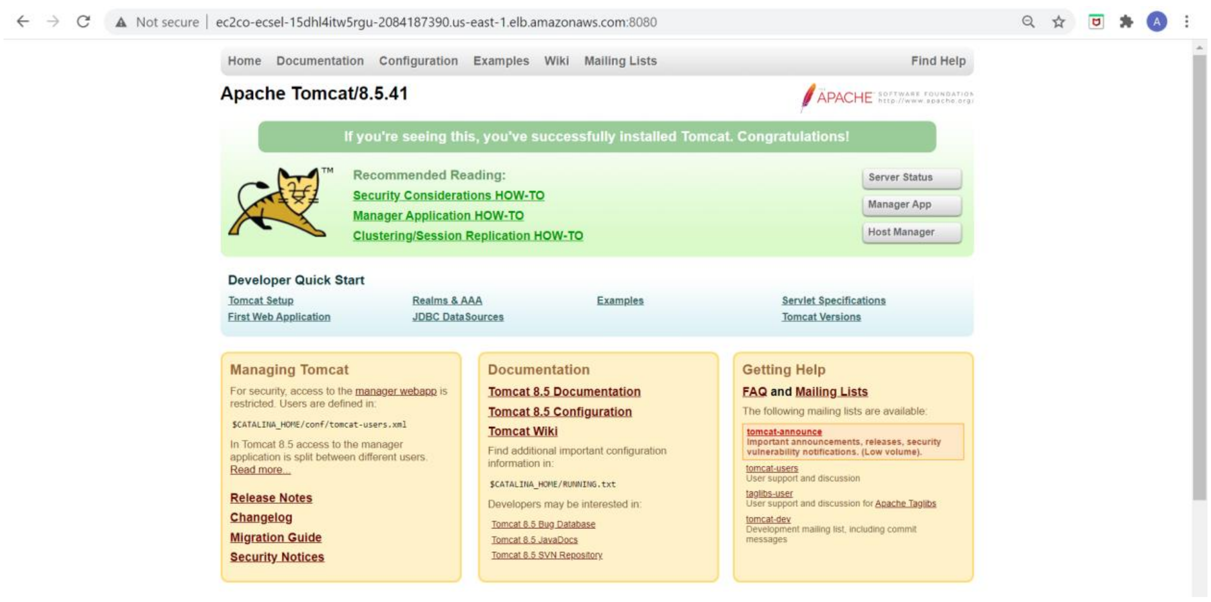
Filter by tags and attributes or search by keyword — 1 to 1 of 1

Name	DNS name	State	VPC ID	Availability Zones	Type	Created At	Monitoring
EC2Co-EcsEl-15DHL4ITW5...	EC2Co-EcsEl-15DHL4ITW5...	active	vpc-0489083d0f9d200ea	us-east-1b, us-east-1a	application	October 5, 2020 at 12:31:34 ...	

Load balancer: EC2Co-EcsEl-15DHL4ITW5RGU

Basic Configuration

Name	EC2Co-EcsEl-15DHL4ITW5RGU
ARN	arn:aws:elasticloadbalancing:us-east-1:504802417114:loadbalancer/app/EC2Co-EcsEl-15DHL4ITW5RGU/8087501e8b76aff8
DNS name	EC2Co-EcsEl-15DHL4ITW5RGU-2084187390.us-east-1.elb.amazonaws.com (A Record)
State	active
Type	application
Scheme	internet-facing
IP address type	ipv4
VPC	vpc-0489083d0f9d200ea
Availability Zones	subnet-01e8378176d3fa13d - us-east-1b IPv4 address: Assigned by AWS subnet-09e829ec5b13888f7 - us-east-1a IPv4 address: Assigned by AWS



Not secure | ec2co-ecsel-15dhl4itw5rgu-2084187390.us-east-1.elb.amazonaws.com:8080

Apache Tomcat/8.5.41

If you're seeing this, you've successfully installed Tomcat. Congratulations!

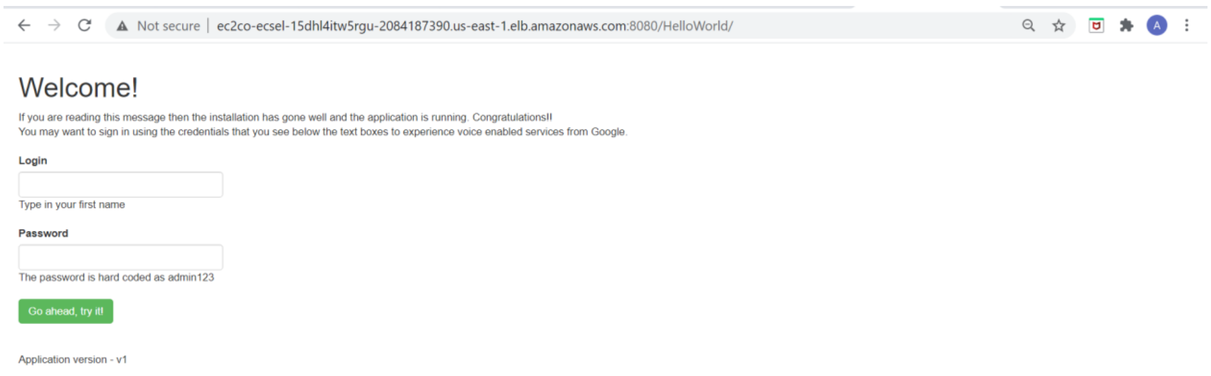
Recommended Reading:
Security Considerations HOW-TO
Manager Application HOW-TO
Clustering/Session Replication HOW-TO

Developer Quick Start
Tomcat Setup
First Web Application
Realms & AAA
JDBC DataSources
Examples
Servlet Specifications
Tomcat Versions

Managing Tomcat
For security, access to the manager webapp is restricted. Users are defined in:
$CATALINA_HOME/conf/tomcat-users.xml
In Tomcat 8.5 access to the manager application is split between different users. Read more...
Release Notes
Changelog
Migration Guide
Security Notices

Documentation
Tomcat 8.5 Documentation
Tomcat 8.5 Configuration
Tomcat Wiki
Find additional important configuration information in:
$CATALINA_HOME/RUNNING.txt
Developers may be interested in:
Tomcat 8.5 Bug Database
Tomcat 8.5 JavaDocs
Tomcat 8.5 SVN Repository

Getting Help
FAQ and Mailing Lists
The following mailing lists are available:
tomcat-announce
Important announcements, releases, security vulnerability notifications. (Low volume).
tomcat-users
User support and discussion
taglibs-user
User support and discussion for Apache Taglibs
tomcat-dev
Development mailing list, including commit messages



Not secure | ec2co-ecsel-15dhl4itw5rgu-2084187390.us-east-1.elb.amazonaws.com:8080/HelloWorld/

Welcome!

If you are reading this message then the installation has gone well and the application is running. Congratulations!!
You may want to sign in using the credentials that you see below the text boxes to experience voice enabled services from Google.

Login

Type in your first name

Password

The password is hard coded as admin123

Application version - v1