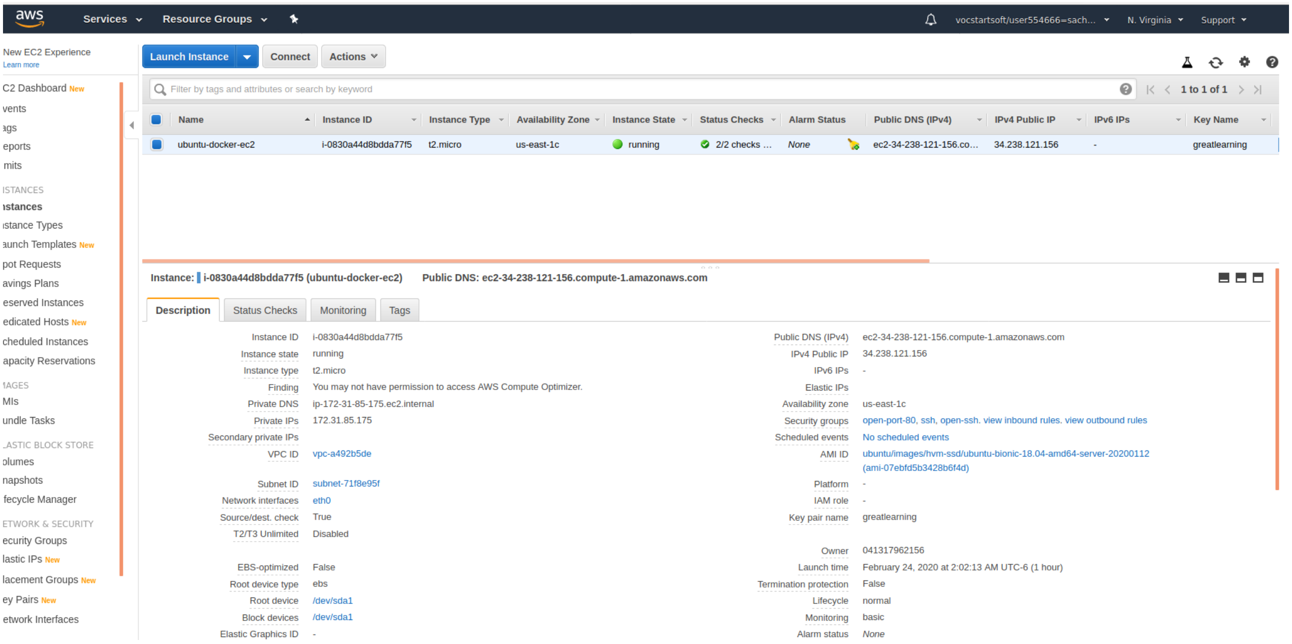
Project 3

Task 1
Local Docker Setup, Docker Image, Dockerfile and Push the image to cloud

Steps:

1. Create an ec2 ubuntu instance.

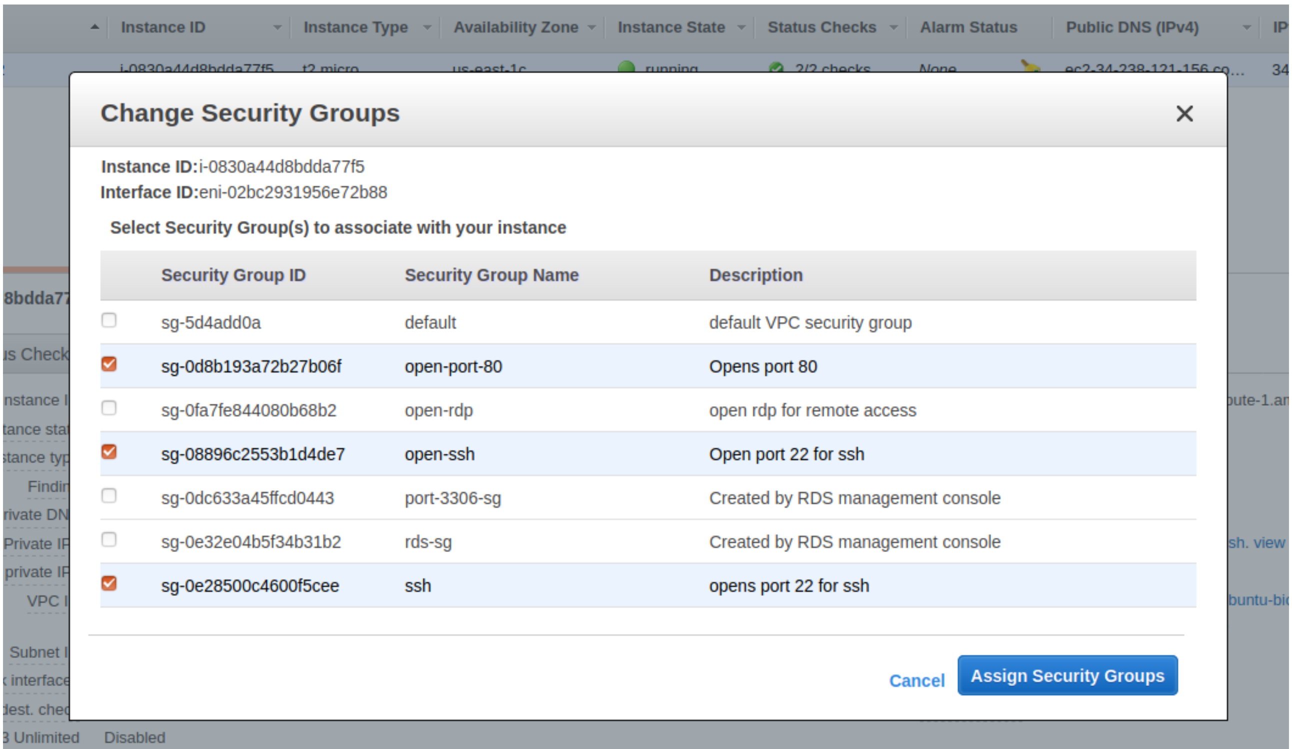


Name	Instance ID	Instance Type	Availability Zone	Instance State	Status Checks	Alarm Status	Public DNS (IPv4)	IPv4 Public IP	IPv6 IPs	Key Name
ubuntu-docker-ec2	i-0830a44d8bdda77f5	t2.micro	us-east-1c	running	2/2 checks ...	None	ec2-34-238-121-156.co...	34.238.121.156	-	greatlearning

Instance: i-0830a44d8bdda77f5 (ubuntu-docker-ec2) Public DNS: ec2-34-238-121-156.compute-1.amazonaws.com

Instance ID	i-0830a44d8bdda77f5	Public DNS (IPv4)	ec2-34-238-121-156.compute-1.amazonaws.com
Instance state	running	IPv4 Public IP	34.238.121.156
Instance type	t2.micro	IPv6 IPs	-
Finding	You may not have permission to access AWS Compute Optimizer.	Elastic IPs	
Private DNS	ip-172-31-85-175.ec2.internal	Availability zone	us-east-1c
Private IPs	172.31.85.175	Security groups	open-port-80, ssh, open-ssh. view inbound rules. view outbound rules
Secondary private IPs		Scheduled events	No scheduled events
VPC ID	vpc-a492b5de	AMI ID	ubuntu/images/hvm-ssd/ubuntu-bionic-18.04-amd64-server-20200112 (ami-07ebfd5b3428b6f4d)
Subnet ID	subnet-71f8e95f	Platform	-
Network interfaces	eth0	IAM role	-
Source/dest. check	True	Key pair name	greatlearning
T2/T3 Unlimited	Disabled		
EBS-optimized	False	Owner	041317962156
Root device type	ebs	Launch time	February 24, 2020 at 2:02:13 AM UTC-6 (1 hour)
Root device	/dev/sda1	Termination protection	False
Block devices	/dev/sda1	Lifecycle	normal
Elastic Graphics ID	-	Monitoring	basic
		Alarm status	None

2. Open port 22, 80, 8080 in security groups and assign it to this ec2 instance.



Instance ID:i-0830a44d8bdda77f5
Interface ID:eni-02bc2931956e72b88

Select Security Group(s) to associate with your instance

	Security Group ID	Security Group Name	Description
☐	sg-5d4add0a	default	default VPC security group
☑	sg-0d8b193a72b27b06f	open-port-80	Opens port 80
☐	sg-0fa7fe844080b68b2	open-rdp	open rdp for remote access
☑	sg-08896c2553b1d4de7	open-ssh	Open port 22 for ssh
☐	sg-0dc633a45ffcd0443	port-3306-sg	Created by RDS management console
☐	sg-0e32e04b5f34b31b2	rds-sg	Created by RDS management console
☑	sg-0e28500c4600f5cee	ssh	opens port 22 for ssh

3. ssh in this ec2 instance

4. sudo apt-get update

5. sudo apt install docker.io

6. sudo usermod -aG docker ubuntu

7. bash

8. exit

9. ssh again

10. docker version

11. sudo chown ubuntu:ubuntu -R /opt

12. ls -al /opt/

13. cd /opt/

14. mkdir myapp

15. cd myapp/

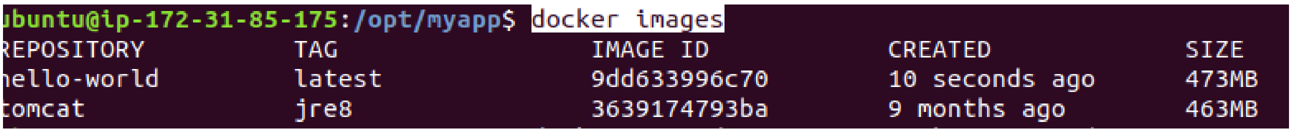
16. download war file
    wget https://storage.googleapis.com/skl-training/aws-codelabs/aws-intro/
    HelloWorld.war

17. create Dockerfile - touch Dockerfile

```
FROM tomcat:jre8
MAINTAINER Janit Sachdeva
COPY HelloWorld.war /usr/local/tomcat/webapps/
```

18. Build the docker image → docker build -t hello-world .

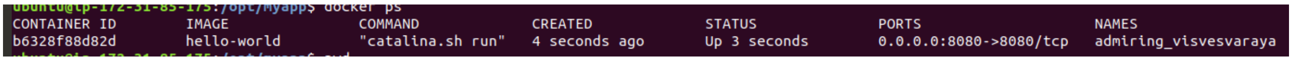
19. docker images



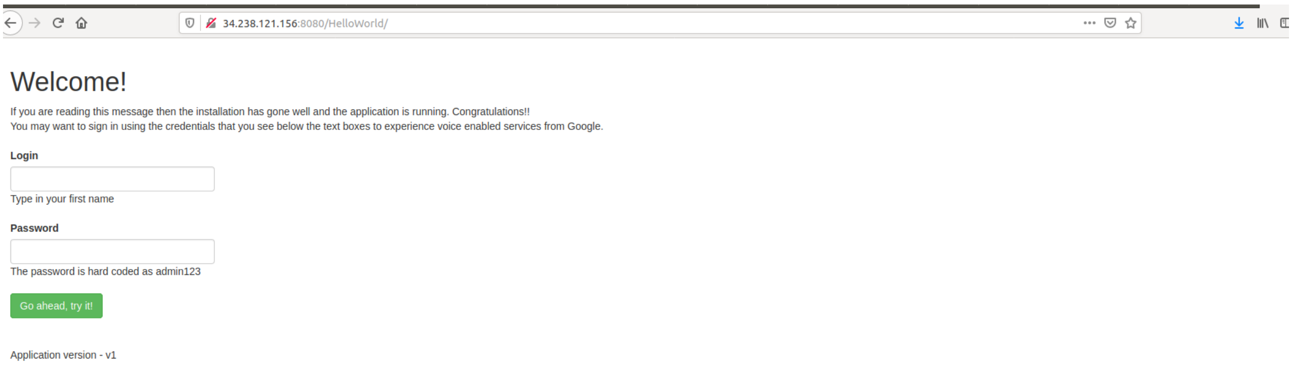
```
ubuntu@ip-172-31-85-175:/opt/myapp$ docker images
REPOSITORY          TAG                 IMAGE ID            CREATED             SIZE
hello-world         latest              9dd633996c70        10 seconds ago      473MB
tomcat              jre8                3639174793ba        9 months ago        463MB
```

20. docker run -d -p 8080:8080 hello-world

21. docker ps



```
ubuntu@ip-172-31-85-175:/opt/myapp$ docker ps
CONTAINER ID        IMAGE               COMMAND             CREATED             STATUS              PORTS                    NAMES
b6328f88d82d        hello-world         "catalina.sh run"   4 seconds ago       Up 3 seconds        0.0.0.0:8080->8080/tcp   admiring_visvesvaraya
```



34.238.121.156:8080/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

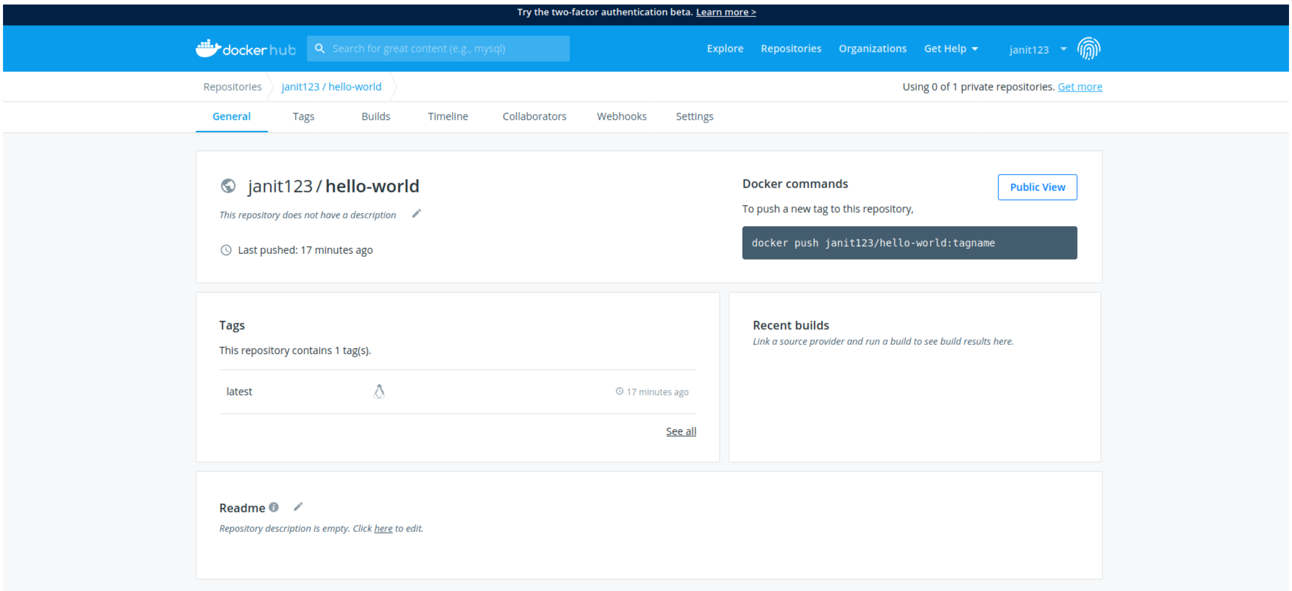
Password

The password is hard coded as admin123

Go ahead, try it!

Application version - v1

22. Create dockerhub repository



Try the two-factor authentication beta. Learn more >

Using 0 of 1 private repositories. Get more

janit123 / hello-world

This repository does not have a description

Last pushed: 17 minutes ago

Docker commands

To push a new tag to this repository,

```
docker push janit123/hello-world:tagname
```

Tags

This repository contains 1 tag(s).

latest   17 minutes ago

See all

Recent builds

Link a source provider and run a build to see build results here.

Readme

Repository description is empty. Click here to edit.

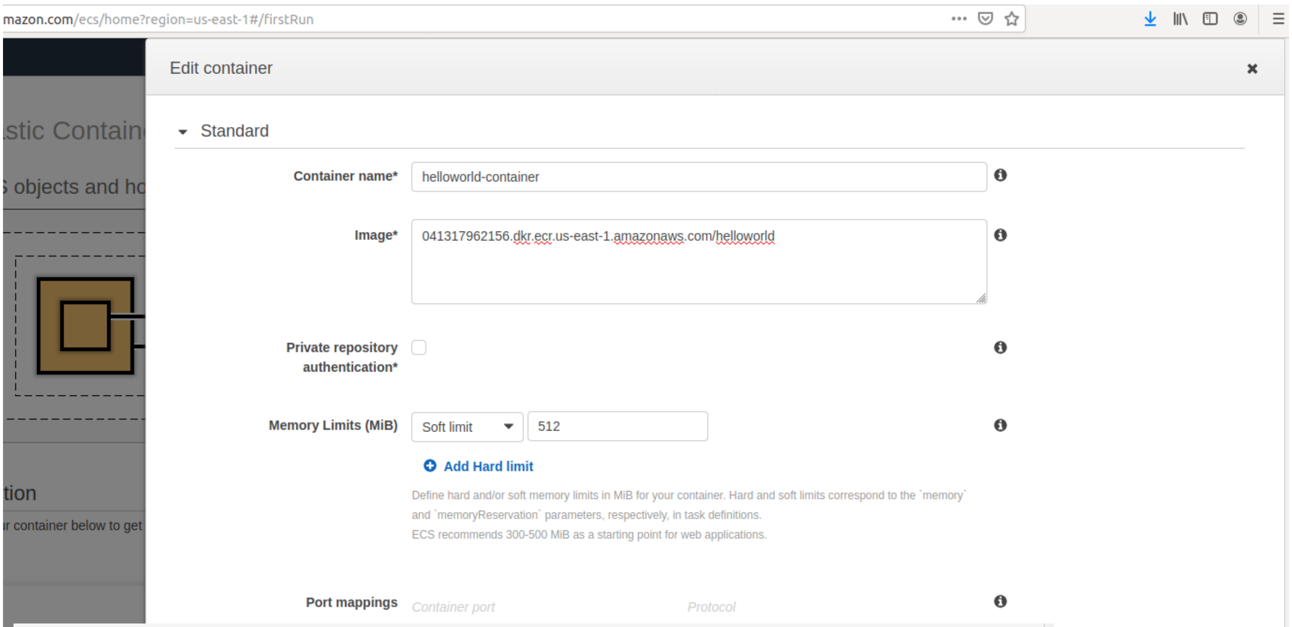
23. docker login

24. Tag the image -- **sudo docker image tag hello-world janit123/hello-world**

25. Push the image to dockerhub public repo --- **sudo docker image push janit123/hello-world**

Task 2:
Setup Fargate ECS and run the Docker Image

1. Create Container----- **helloworld-container**



Container name*: helloworld-container

Image*: 041317962156.dkr.ecr.us-east-1.amazonaws.com/helloworld

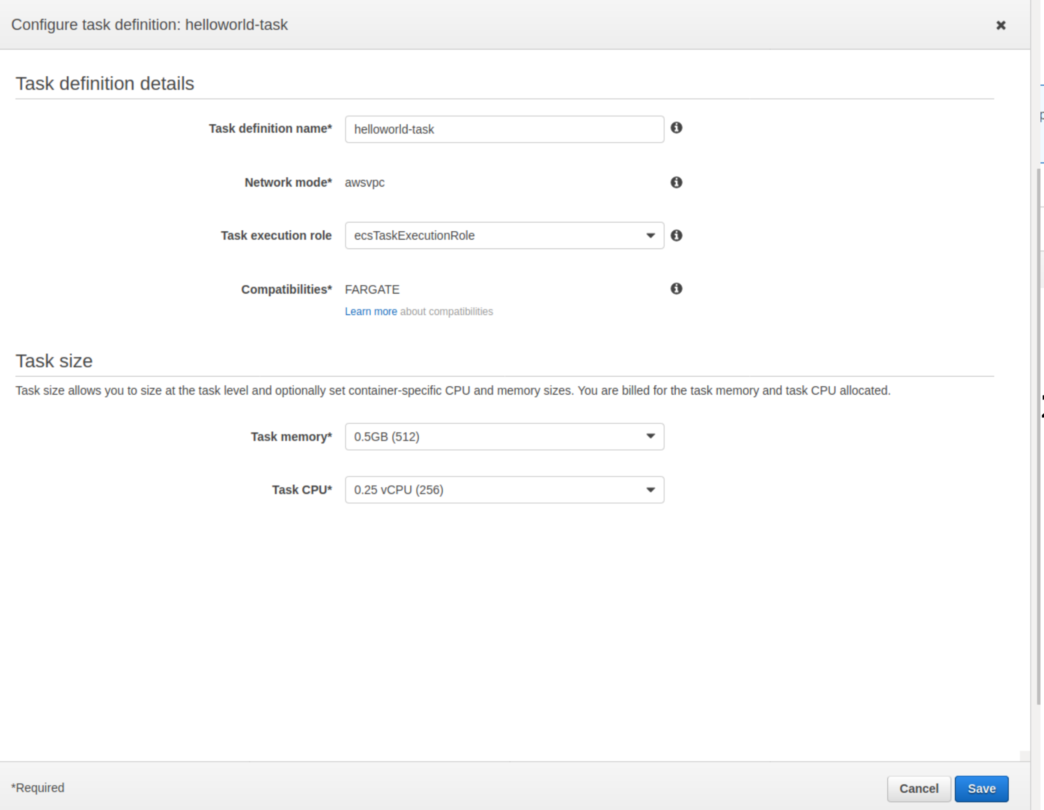
Private repository authentication*: □

Memory Limits (MiB): Soft limit 512

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

2. Create Task



Configure task definition: helloworld-task
Task definition details

Task definition name*: helloworld-task

Network mode*: awsvpc

Task execution role: ecsTaskExecutionRole

Compatibilities*: FARGATE

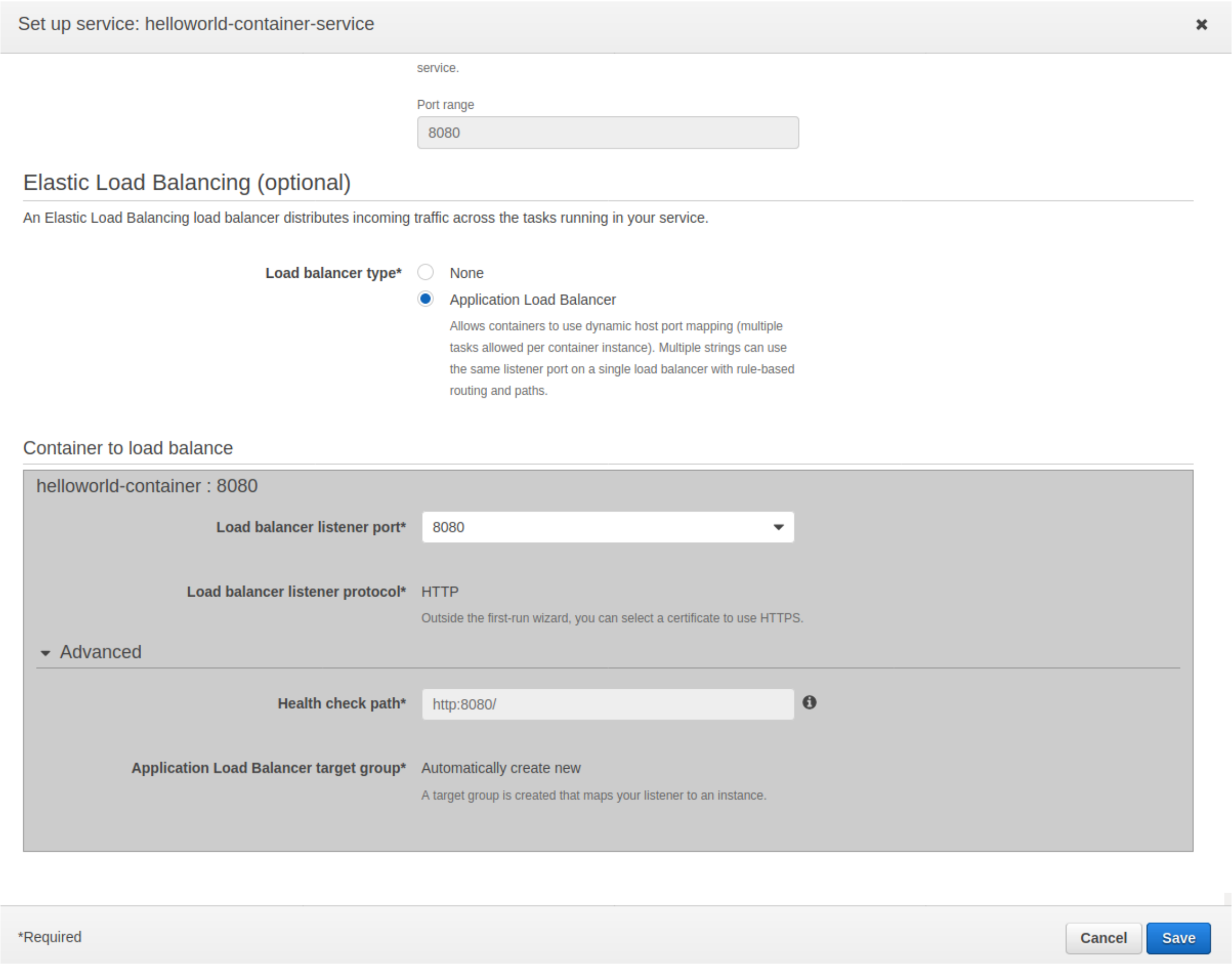
Learn more about compatibilities
Task size

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)

3. Create Service



Set up service: helloworld-container-service

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-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

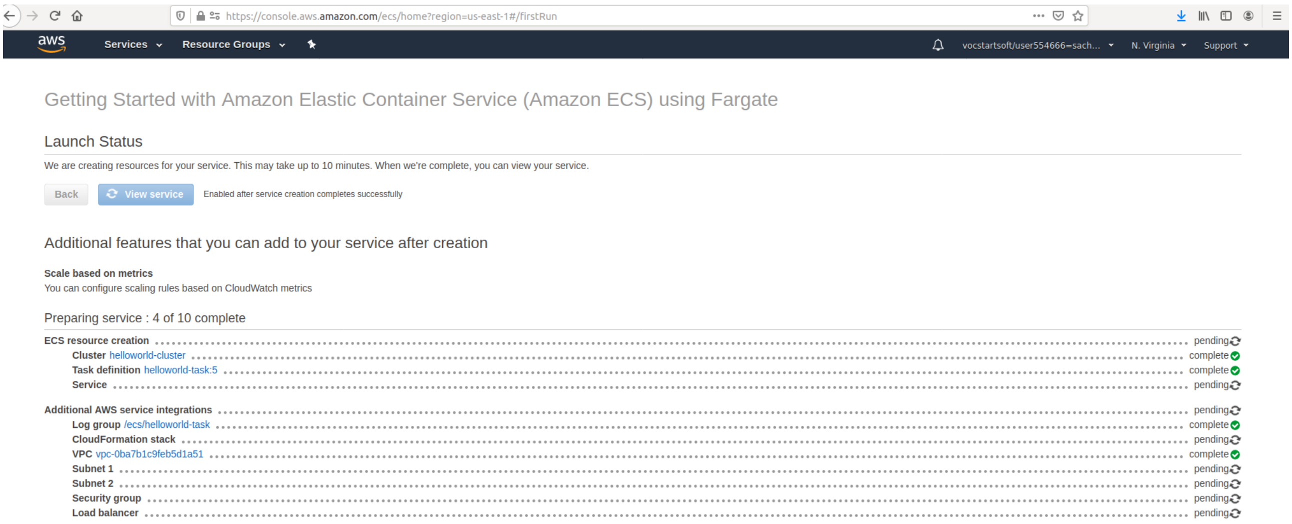
Health check path*: http:8080/

Application Load Balancer target group*: Automatically create new

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

*Required

4. Create Cluster and save


https://console.aws.amazon.com/ecs/home?region=us-east-1#/firstRun
aws
Services
Resource Groups
vocstartsoft/user554666=sach...
N. Virginia
Support
Back
View service

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.

Enabled after service creation completes successfully

Additional features that you can add to your service after creation

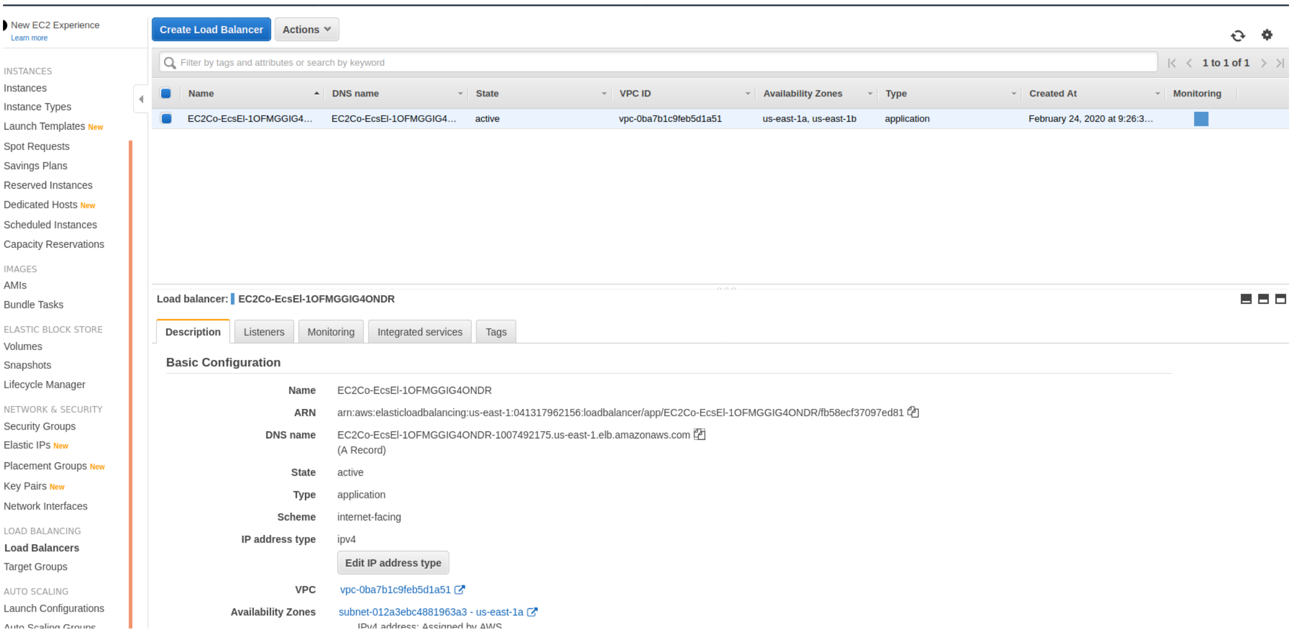
Scale based on metrics
You can configure scaling rules based on CloudWatch metrics

Preparing service : 4 of 10 complete

```
ECS resource creation .................................... pending
    Cluster helloworld-cluster ........................... complete ✓
    Task definition helloworld-task:5 .................... complete ✓
    Service .............................................. pending

Additional AWS service integrations ...................... pending
    Log group /ecs/helloworld-task ....................... complete ✓
    CloudFormation stack ................................. pending
    VPC vpc-0ba7b1c9feb5d1a51 ............................ complete ✓
    Subnet 1 ............................................. pending
    Subnet 2 ............................................. pending
    Security group ....................................... pending
    Load balancer ........................................ pending
```

5. Load Balancer and Target Group created



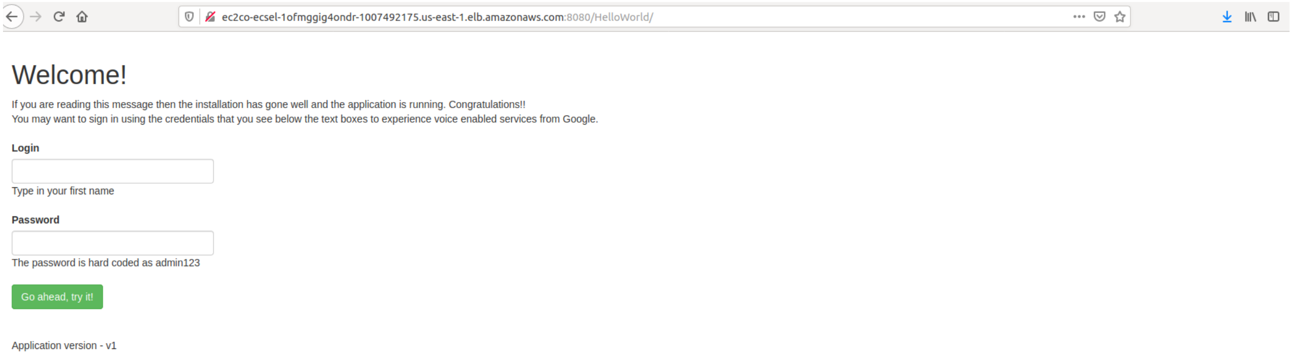
Name	DNS name	State	VPC ID	Availability Zones	Type	Created At	Monitoring
EC2Co-EcsEl-1OFMGGIG4...	EC2Co-EcsEl-1OFMGGIG4...	active	vpc-0ba7b1c9feb5d1a51	us-east-1a, us-east-1b	application	February 24, 2020 at 9:26:3...	

Load balancer: EC2Co-EcsEl-1OFMGGIG4ONDR

Basic Configuration

Name	EC2Co-EcsEl-1OFMGGIG4ONDR
ARN	arn:aws:elasticloadbalancing:us-east-1:041317962156:loadbalancer/app/EC2Co-EcsEl-1OFMGGIG4ONDR/fb58ecf37097ed81
DNS name	EC2Co-EcsEl-1OFMGGIG4ONDR-1007492175.us-east-1.elb.amazonaws.com (A Record)
State	active
Type	application
Scheme	internet-facing
IP address type	ipv4
VPC	vpc-0ba7b1c9feb5d1a51
Availability Zones	subnet-012a3ebc4881963a3 - us-east-1a IPv4 address: Assigned by AWS

6. Open web application in browser using load balancer DNS



ec2co-ecsel-1ofmggig4ondr-1007492175.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