



Subject- CMPE 281 – Cloud Technologies

Project 1 – FileStore

Student Name: Parvathi Pai

Student ID: 015293460

Table of Contents

Technologies Used.....	3
GitHub.....	4
Architecture.....	4
Steps for deployment.....	4
User Registration.....	4
Custom Login.....	6
File upload and download.....	7
File Updates	7
AWS resources.....	8
CloudFront and Bucket Policy.....	8
AWS Lambda, Cloud Watch and AWS SNS	8
Route 53.....	11
RDS	12
EC2 instances.....	13
Load balancer.....	13
Auto Scaling groups.....	14
Cognito	14
References.....	15

Technologies Used

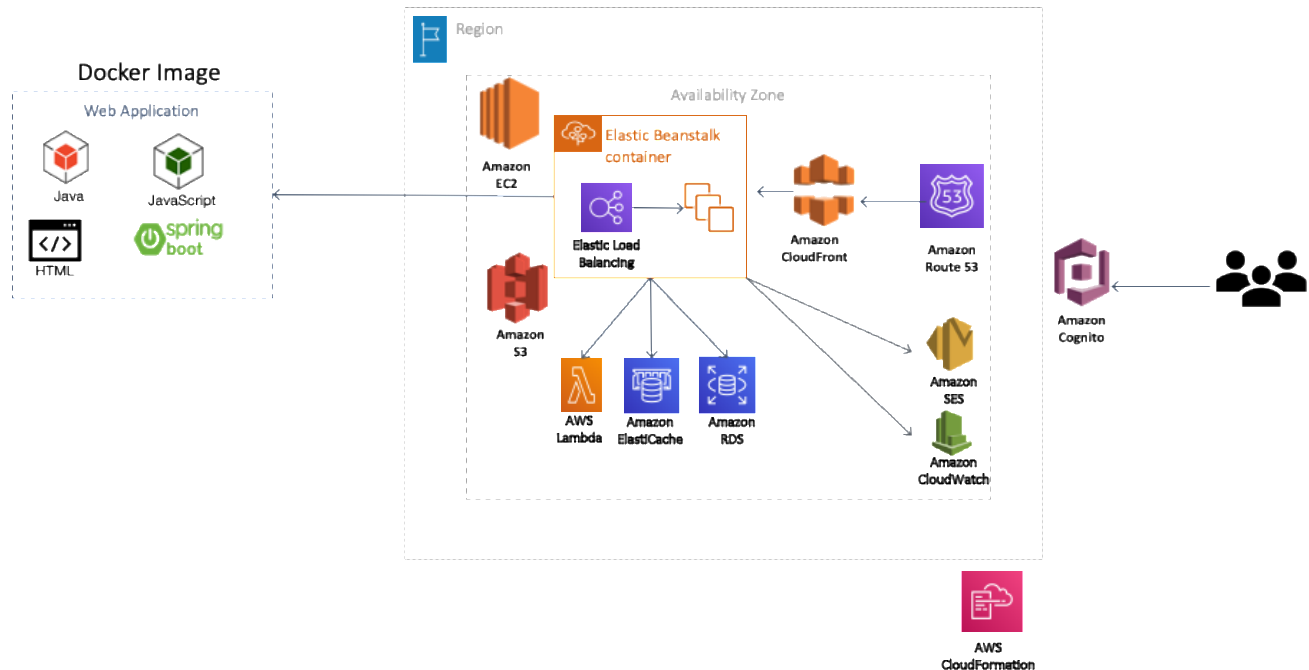
- Java 15, Gradle
- AWS account access
- Docker account
- AWS CLI and Docker CLI
- AWS Certificate Manager – generate certificate for domain proj1.parvathipai.com and associate with application load balancer in Beanstalk for SSL/TLS termination
- AWS Route 53 – hosted zone for parvathipai.com and ALIAS record for proj1.parvathipai.com
- AWS Cognito – user pool for users and admins logging into and registering to access the website and also for 1) validating security requirements including – password strength, standard attributes required for registration, account recovery and 2) registering an appclient for OAuth 2 authentication in Spring Security
- AWS RDS – for a micro free tier MySQL instance that hosts the database that the Spring application interacts with and records the logs of file uploads, etc.
- AWS S3 hosts a bucket that is partitioned by username to ensure users cannot access each other's uploads for modification or deletion
- AWS IAM configures users who can deploy the Spring application
- AWS EC2 Load Balancer is the SSL termination for requests and routes to 2 instances of the application via Beanstalk.
- AWS EC2 hosts the Spring application via a webserver listening on 80 (HTTP) and 443 (HTTPS) and connects the aforementioned services
- Spring application uses Spring Security, Spring-Starter-web, Thymeleaf, OAuth2, JPA etc. and is deployed via a Docker container in 3 environments – local development via IntelliJ IDEA, local deployment via beanstalk, and prod deployment via Beanstalk – with corresponding environment variables
- Spring application exposes a Thymeleaf page for uploading files and APIs for interacting with the application
- AWS VPC configures a bridge security group that connects to both the Beanstalk application and the RDS instance allowing only 3306 (MySQL) incoming traffic to go through.
- DataGrip to view AWS RDS

GitHub

Link to GitHub - <https://github.com/ParvathiRPai/FileStore>

Architecture

Displayed using for one **region** in the below diagram for simplicity but during deployment it's **made multiple regions**.



Steps for deployment

User Registration

- Register the account with amazon Cognito explanation video in one drive link - <https://1drv.ms/v/s!Atlj3hm1Xtus50PzYuEYQuqkg4M1>
- Enter the username, email and password and verify the email

Filestore application

[Log in with Amazon Cognito](#)

Sign up with a new account

Username

testuser4

Given name

Family name

Email

name@host.com

Password

✓ Password must contain a lower case letter

✓ Password must contain an upper case letter

✓ Password must contain a special character

✓ Password must contain a number

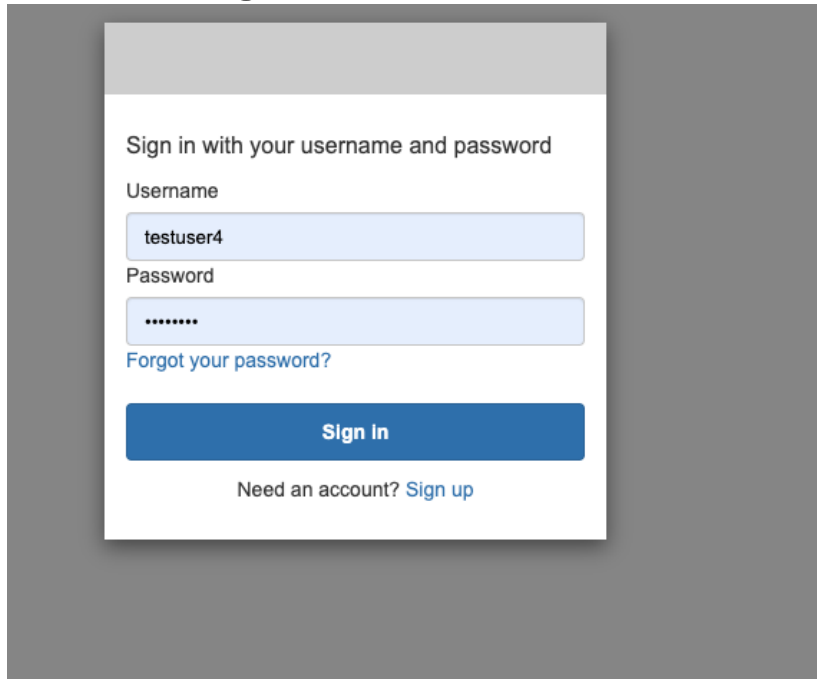
✓ Password must contain at least 8 characters

Sign up

Already have an account? [Sign in](#)

The user should enter the username and password created at the time of registration.

Custom Login



Sign in with your username and password

Username

testuser4

Password

.....

[Forgot your password?](#)

Sign In

Need an account? [Sign up](#)

AWS Cognito verified the user details as shown below
In the Test user pool, the user account will be created

Users > **a93e8edd-e529-4097-87d6-4c7cba3a3270**

[Add to group](#)

[Reset password](#)

[Enable SMS MFA](#)

[Disable user](#)

Groups -

Account Status Enabled / CONFIRMED

SMS MFA Status Disabled

Last Modified Oct 24, 2020 8:22:33 PM

Created Oct 24, 2020 8:22:21 PM

sub a93e8edd-e529-4097-87d6-4c7cba3a3270

email_verified true

email parvathirpai@gmail.com

File upload and download

One drive link showing the working of upload and download -

<https://1drv.ms/u/s!Atlj3hm1Xtus50Z42mOLSvlceJXK>

After login the user goes to the page to upload and download and delete the file –

Filestore application

Hello, **testuser4**!

Your authorities are: **ROLE_USER SCOPE_openid**

This section is only visible only for ROLE_USER.

This section is only visible only to authenticated users.

Filestore application

Your user name is **testuser4** Your first name is **Test4** Your last name is **User** Your email is **testuser4@example.com**

File to upload: No file chosen

File Updates

One drive link showing file updates - <https://1drv.ms/v/s!Atlj3hm1Xtus50cAwC3ngc6alLKG>

File upload and delete and view in AWS RDS –

In this project I have made use of DataGrip software to analyze the Amazon RDS database

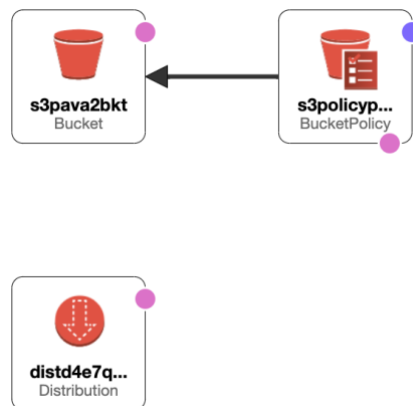
	id	description	email	first_name	last_modified_at	last_name	uploaded_at	user_name
1	1	Test Description	testuser4@example.com	Test4	2020-10-23 22:00:55	User	2020-10-23 22:00:55	testuser4
2	2	Test Description	testuser4@example.com	Test4	2020-10-23 23:10:02	User	2020-10-23 23:10:02	testuser4
3	3	Test Description	testuser4@example.com	Test4	2020-10-24 00:27:49	User	2020-10-24 00:27:49	testuser4
4	4	Test Description	testuser4@example.com	Test4	2020-10-24 00:59:13	User	2020-10-24 00:59:13	testuser4
5	5	Test Description	testuser4@example.com	Test4	2020-10-24 23:57:59	User	2020-10-24 23:57:59	testuser4
6	6	Test Description	testuser4@example.com	Test4	2020-10-25 00:01:50	User	2020-10-25 00:01:50	testuser4

AWS resources

- I have made use of the same AWS resources as that of HW-2
- Made use of cloud formation template to create highly available, scalable solution with DR measures.
- The link to template is here - <https://github.com/ParvathiRPai/aws-cloud>
- Video explanation one drive link - <https://1drv.ms/v/s!Atlj3hm1Xtus50Tpstwi5E8DF7d3>
- Cloud formation bucket policy –

CloudFront and Bucket Policy

Added cloud front to block the IP address of Cuba and downloading of the file is done via the cloud front. Only authorized users can access the AWS buckets and after 75 days the contents of AWS resources is added in AWS Glacier.



AWS Lambda, Cloud Watch and AWS SNS

For this project I have created a trigger between AWS lambda and S3 when a file is deleted a lambda trigger is created and a log will be posted in the cloud watch and notification will be sent via AWS SNS.

One drive explanation link -

<https://1drv.ms/v/s!Atlj3hm1Xtus50WQ3eiZKejh-vRE>

Uploaded the docker image generated through gradle on Elastic Beanstalk and the application is running in the EC2 instance

[Elastic Beanstalk](#) > [Environments](#)

All environments

Filter results matching the display values


< 1 >

	Environment name	Health	Application name	Date created	Last modified	URL	Running versions	Platform	Platform state	Tier name
<input type="radio"/>	testenv	Ok	filestore	2020-10-23 15:03:44 UTC-0700	2020-10-24 17:00:56 UTC-0700	testenv.eba-ia8vjdb.us-west-2.elasticbeanstalk.com	app-201024_165857	Docker running on 64bit Amazon Linux 2	Supported	WebServer

testenv
testenv.eba-ia8vjdb.us-west-2.elasticbeanstalk.com [\(e-bsammjkbwh\)](#)
Application name: filestore

Refresh


Actions

Health

Ok

Causes

Running version
app-201024_165857

Upload and deploy

Platform

Docker running on 64bit Amazon Linux 2/3.2.0

Change

Recent events

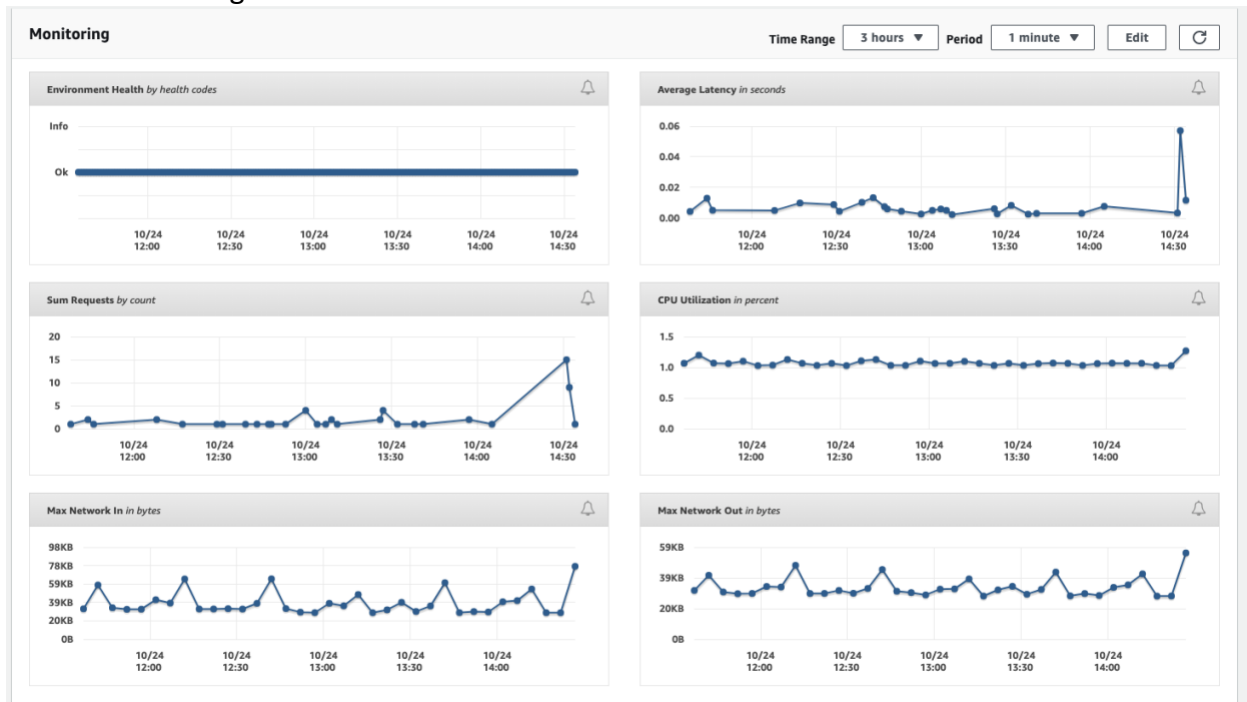
Show all

< 1 >

Time	Type	Details
2020-10-24 17:02:32 UTC-0700	INFO	Environment health has transitioned from Info to Ok. Application update completed 84 seconds ago and took 44 seconds.
2020-10-24 17:00:56 UTC-0700	INFO	Environment update completed successfully.
2020-10-24 17:00:56 UTC-0700	INFO	New application version was deployed to running EC2 instances.
2020-10-24 17:00:49 UTC-0700	INFO	Instance deployment completed successfully.
2020-10-24 17:00:32 UTC-0700	INFO	Environment health has transitioned from Ok to Info. Application update in progress on 1 instance. 0 out of 1 instance completed (running for 18 seconds).

9

Health Monitoring of elastic beanstalk -



Deployment of EC2 through elastic beanstalk –

Time	Type	Details
2020-10-23 17:57:44 UTC-0700	INFO	Environment health has transitioned from Info to Ok. Application update completed 47 seconds ago and took 31 seconds.
2020-10-23 17:56:44 UTC-0700	INFO	Environment health has transitioned from Ok to Info. Application update in progress on 1 instance. 0 out of 1 instance completed (running for 28 seconds).
2020-10-23 17:56:41 UTC-0700	INFO	Environment update completed successfully.
2020-10-23 17:56:41 UTC-0700	INFO	New application version was deployed to running EC2 instances.
2020-10-23 17:56:37 UTC-0700	INFO	Instance deployment completed successfully.
2020-10-23 17:56:10 UTC-0700	INFO	Deploying new version to instance(s).
2020-10-23 17:56:06 UTC-0700	INFO	Environment update is starting.
2020-10-23 17:46:44 UTC-0700	INFO	Environment health has transitioned from Info to Ok. Application update completed 44 seconds ago and took 31 seconds.
2020-10-23 17:45:44 UTC-0700	INFO	Environment health has transitioned from Ok to Info. Application update in progress on 1 instance. 0 out of 1 instance completed (running for 25 seconds).
2020-10-23 17:45:28 UTC-0700	INFO	Environment update completed successfully.
2020-10-23 17:45:28 UTC-0700	INFO	New application version was deployed to running EC2 instances.
2020-10-23 17:45:22 UTC-0700	INFO	Instance deployment completed successfully.
2020-10-23 17:44:57 UTC-0700	INFO	Deploying new version to instance(s).
2020-10-23 17:44:52 UTC-0700	INFO	Environment update is starting.
2020-10-23 17:41:44 UTC-0700	INFO	Environment health has transitioned from Info to Ok. Application restart completed 50 seconds ago and took 7 seconds.
2020-10-23 17:40:44 UTC-0700	INFO	Environment health has transitioned from Ok to Info. Application restart in progress (running for 7 seconds).
2020-10-23 17:40:43 UTC-0700	INFO	Restarted application server on all ec2 instances.
2020-10-23 17:40:38 UTC-0700	INFO	Instance deployment completed successfully.
2020-10-23 17:40:36 UTC-0700	INFO	restartAppServer is starting.
2020-10-23 17:37:45 UTC-0700	INFO	Environment health has transitioned from Info to Ok. Configuration update completed 64 seconds ago and took 2 minutes.

Route 53

Website hosted zone. My route - <https://proj1.parvathipai.com>

Route 53 > Hosted zones

Hosted zones [Info](#)

Hosted zones (1) [Refresh](#) [View details](#) [Edit](#) [Delete](#) [Create hosted zone](#)

Automatic mode is the current search behavior optimized for best filter results. [To change modes go to settings.](#)

[<](#) [1](#) [>](#) [⊙](#)

	Domain name	Type	Created by	Record count	Description	Hosted zone ID
<input type="radio"/>	parvathipai.com	Public	Route 53	4	HostedZone created by Route53 Registrar	Z03543683PMF5BN2VRF4B

parvathipai.com [Info](#) [Delete](#) [Test record](#) [Configure query logging](#)

Hosted zone details [Edit](#)

[Records \(4\)](#) Hosted zone tags (0)

Records (4) [Info](#)

Automatic mode is the current search behavior optimized for best filter results. [To change modes go to settings.](#)

[Type](#) [Routing policy](#) [Alias](#) [<](#) [1](#) [>](#) [⊙](#)

<input type="checkbox"/>	Record name	Type	Routing policy	Differentiator	Alias	Value/Route traffic to	TTL (seconds)	Health check	Evaluate target health	Record ID
<input type="checkbox"/>	parvathipai.com	NS	Simple	-	No	ns-1937.awsdns-50.co.uk. ns-608.awsdns-12.net. ns-1166.awsdns-17.org. ns-208.awsdns-26.com.	172800	-	-	-
<input type="checkbox"/>	parvathipai.com	SOA	Simple	-	No	ns-1937.awsdns-50.co.uk. awsdns-hostmaster.amazon.com. 1 7200 900 1209600 86400	900	-	-	-
<input type="checkbox"/>	proj1.parvathipai.com	A	Simple	-	Yes	testenv.eba-la8vjdb.us-west-2.elasticbeanstalk.com.	-	-	Yes	-
<input type="checkbox"/>	_0a269bd0cedd82ebc85dd762f1be54ac.proj1.parvathipai.com	CNAME	Simple	-	No	_062f83c90f6aff1983d1b5421e4a1dcb.zbkrxrfvj.acm-validations.aws.	300	-	-	-

RDS

SQL database

RDS > Databases > testdbinstance3

testdbinstance3

Modify

Actions

Summary

DB identifier

testdbinstance3

CPU

1.86%

Info

Available

Class

db.t2.micro

Role

Instance

Current activity

11 Connections

Engine

MySQL Community

Region & AZ

us-west-2c

Connectivity & security

Monitoring

Logs & events

Configuration

Maintenance & backups

Tags

Connectivity & security

Endpoint & port

Endpoint

testdbinstance3.cb0tzhehmaym.us-west-2.rds.amazonaws.com

Port

3306

Networking

Availability zone

us-west-2c

VPC

vpc-8b7c1df3

Subnet group

default-vpc-8b7c1df3

Subnets

subnet-b6f13ceb

subnet-fdc98cd6

subnet-30a95b48

subnet-169e825d

Security

VPC security groups

beanstalk-rds-bridge (sg-0fb7a362fd1c327cc)

(active)

launch-wizard-1 (sg-0ddff0e3c5027baf94)

(active)

Public accessibility

Yes

Certificate authority

rds-ca-2019

Certificate authority date

Aug 22nd, 2024

View in DataGrip

<Filter Criteria>

id	description	email	first_name	last_modified_at	last_name	uploaded_at	user_name
1	Test Description	testuser4@example.com	Test4	2020-10-23 22:00:55	User	2020-10-23 22:00:55	testuser4
2	Test Description	testuser4@example.com	Test4	2020-10-23 23:10:02	User	2020-10-23 23:10:02	testuser4
3	Test Description	testuser4@example.com	Test4	2020-10-24 00:27:49	User	2020-10-24 00:27:49	testuser4
4	Test Description	testuser4@example.com	Test4	2020-10-24 00:59:13	User	2020-10-24 00:59:13	testuser4
5	Test Description	testuser4@example.com	Test4	2020-10-24 23:57:59	User	2020-10-24 23:57:59	testuser4
6	Test Description	testuser4@example.com	Test4	2020-10-25 00:01:50	User	2020-10-25 00:01:50	testuser4

12

EC2 instances

EC2 > Instances > i-0f94778b3d0520123

Instance summary for i-0f94778b3d0520123 (testenv) [Info](#)

Updated less than a minute ago

[Refresh](#) [Connect](#) [Instance state](#)

Instance ID i-0f94778b3d0520123 (testenv)	Public IPv4 address 18.237.216.154 open address	Private IPv4 addresses 172.31.15.151
Instance state Running	Public IPv4 DNS ec2-18-237-216-154.us-west-2.compute.amazonaws.com open address	Private IPv4 DNS ip-172-31-15-151.us-west-2.compute.internal
Instance type t2.micro	Elastic IP addresses -	VPC ID vpc-8b7c1df3
IAM Role aws-elasticbeanstalk-ec2-role	Subnet ID subnet-b6f13ceb	

AWS Compute Optimizer
Opt-in to AWS Compute Optimizer for recommendations. [Learn more](#)

Details | Security | Networking | Storage | Monitoring | Tags

Instance details [Info](#)

Platform Linux/UNIX	AMI ID ami-05a1bbda6ce073d6	Monitoring disabled
Platform details Linux/UNIX	AMI name aws-elasticbeanstalk-ec2-2.0.20200928.64bit-eb_docker_amazon_linux_2-hvm-2020-10-02T11-20	Termination protection Disabled
Launch time Fri Oct 23 2020 15:30:14 GMT-0700 (Pacific Daylight Time) (1 day)	AMI location amazon/aws-elasticbeanstalk-ec2-2.0.20200928.64bit-eb_docker_amazon_linux_2-hvm-2020-10-02T11-20	Lifecycle normal
Stop-hibernate behavior disabled	AMI Launch index 0	Key pair name -

Load balancer

[Create Load Balancer](#) [Actions](#)

Filter by tags and attributes or search by keyword

Name	DNS name	State	VPC ID	Availability Zones	Type	Created At	Monitoring
awseb-e-b-AWSEBLoa-PW...	awseb-e-b-AWSEBLoa-PW...		vpc-8b7c1df3	us-west-2c, us-west-2b...	classic	October 23, 2020 at 3:04:04 ...	

Load balancer: **awseb-e-b-AWSEBLoa-PWDGIG080PCX**

Description | Instances | Health check | Listeners | Monitoring | Tags | Migration

Basic Configuration

Name	awseb-e-b-AWSEBLoa-PWDGIG080PCX	Creation time	October 23, 2020 at 3:04:04 PM UTC-7
DNS name	awseb-e-b-AWSEBLoa-PWDGIG080PCX-1364866426.us-west-2.elb.amazonaws.com (A Record)	Hosted zone	Z1H1FL5HABSF5
Type	Classic (Migrate Now)	Status	1 of 1 instances in service
Scheme	Internet-facing	VPC	vpc-8b7c1df3
Availability Zones	subnet-169e825d - us-west-2b, subnet-30a95b48 - us-west-2a, subnet-b6f13ceb - us-west-2c		

Auto Scaling groups

EC2 > Auto Scaling groups > awseb-e-bsammjkbwh-stack-AWSEBAutoScalingGroup-10G1GEHFLEAV5

Details | Activity | Automatic scaling | Instance management | Monitoring | Instance refresh

Group details

Desired capacity	1	Auto Scaling group name	awseb-e-bsammjkbwh-stack-AWSEBAutoScalingGroup-10G1GEHFLEAV5
Minimum capacity	1	Date created	Fri Oct 23 2020 15:04:19 GMT-0700 (Pacific Daylight Time)
Maximum capacity	4	Amazon Resource Name (ARN)	arn:aws:autoscaling:us-west-2:192075818472:autoScalingGroup:3379fb24-a1c4-4a20-9892-9e35efb42aca:autoScalingGroupName/awseb-e-bsammjkbwh-stack-AWSEBAutoScalingGroup-10G1GEHFLEAV5

Launch configuration

Launch configuration	awseb-e-bsammjkbwh-stack-AWSEBAutoScalingLaunchConfiguration-W4MD1GAWAD06	AMI ID	ami-05a1bbdab6ce073d6	Security groups	awseb-e-bsammjkbwh-stack-AWSEBSecurityGroup-AHJ3HCGTVUSV beanstalk-rds-bridge
Instance type	t2.micro	Key pair name	-	Create time	Fri Oct 23 2020 15:29:45 GMT-0700 (Pacific Daylight Time)
Storage (volumes)	-				

[View details in the launch configuration console](#)

Cognito

App client settings

What identity providers and OAuth 2.0 settings should be used for your app clients?

Each of your app clients can use different identity providers and OAuth 2.0 settings. You must enable at least one identity provider for each app client. [Learn more about identity providers.](#)

App client testappclient1

ID 51an9opmenne1oha1vufu142dk

Enabled Identity Providers ☒ Select all

☒ Cognito User Pool

Sign in and sign out URLs

Enter your callback URLs below that you will include in your sign in and sign out requests. Each field can contain multiple URLs by entering a comma after each URL.

Callback URL(s)

<http://localhost:8080/login/oauth2/code/cognito>, <https://proj1.parvathipai.com/login/oauth2/code/cognito>

Sign out URL(s)

<http://localhost:8080/logout>, <https://proj1.parvathipai.com/logout>

OAuth 2.0

Select the OAuth flows and scopes enabled for this app. [Learn more about flows and scopes.](#)

Allowed OAuth Flows

☒ Authorization code grant ☐ Implicit grant ☐ Client credentials

Allowed OAuth Scopes

☐ phone ☒ email ☒ openid ☐ aws.cognito.signin.user.admin ☐ profile

Hosted UI

The hosted UI provides an OAuth 2.0 authorization server with built-in webpages that can be used to sign up and sign in users using the domain you created. [Learn more about the hosted UI](#)

[Launch Hosted UI](#)

References

- <https://spring.io/guides>
- AWS documentation
- AWS java SDK