SJSU SAN JOSÉ STATE UNIVERSITY

Subject- CMPE 281 – Cloud Technologies
Project 1 – FileStore
Student Name: Parvathi Pai
Student ID: 015293460

Table of Contents

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**.
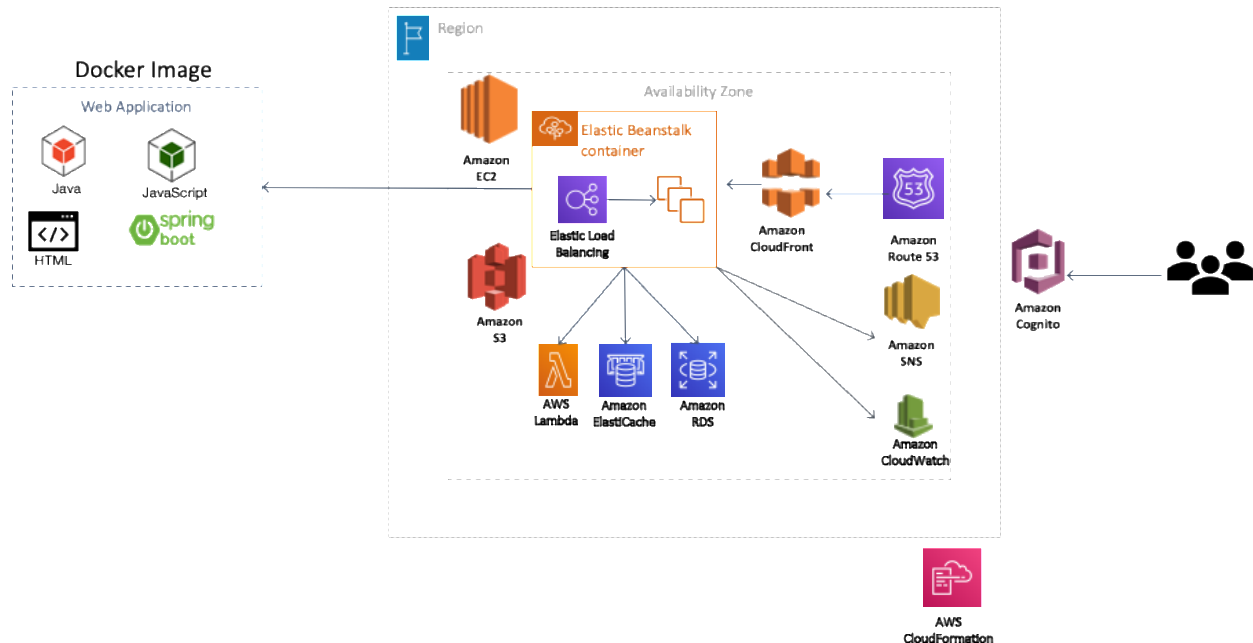
Steps for deployment

User Registration

- Register the account with amazon Cognito explanation video in one drive link – https://1drv.ms/v/s!Atlj3hm1Xtus50PzYuEYQugkg4M1

- 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
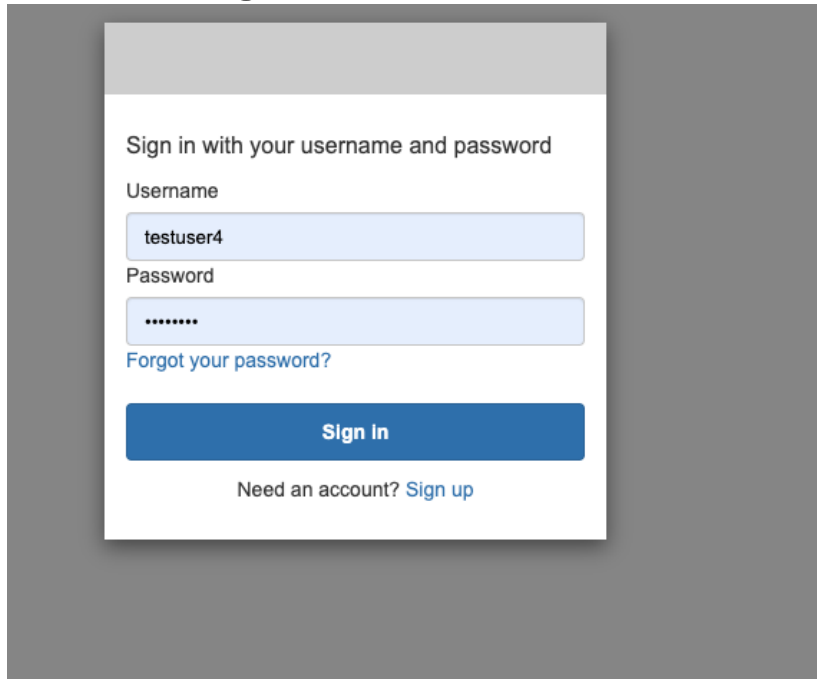
Sign up

Already have an account? Sign in

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

Custom Login

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

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 –

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

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.
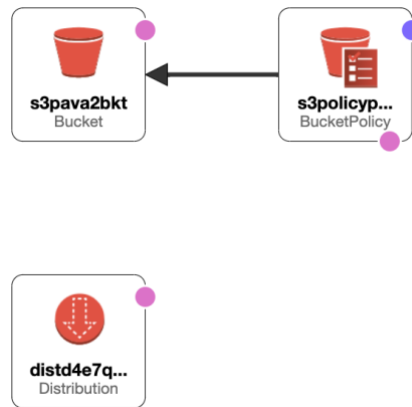
AWS Lambda, Cloud Watch and AWS SNS

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

AWS Lambda

Gets triggered when file in S3 gets deleted

AWS CloudWatch
Logs get created when files in S3 gets deleted

Elastic Beanstalk

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

Health Monitoring of elastic beanstalk -
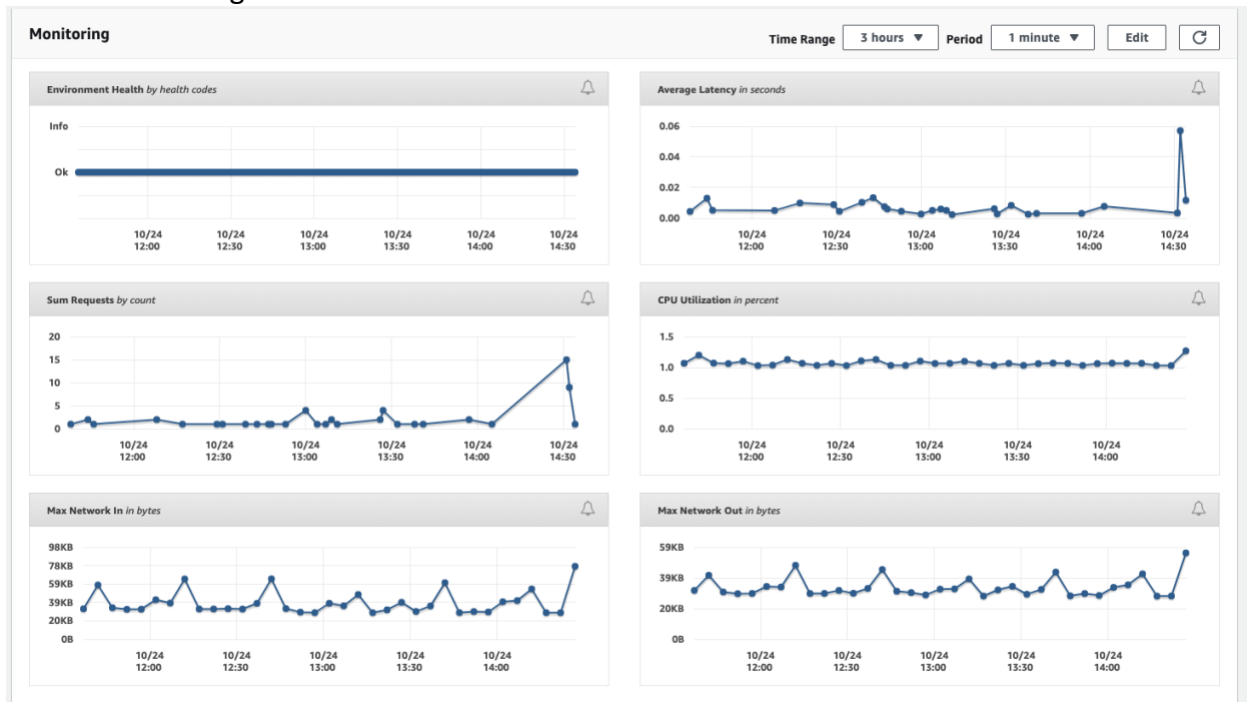
Deployment of EC2 through elastic beanstalk –

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

Hosted zones Info

Hosted zones (1)

Automatic mode is the current search behavior optimized for best filter results. To change modes go to settings.

| View details | Edit | Delete | **Create hosted zone** |

Q Filter hosted zones by property or value

< 1 >

	Domain name ▽	Type ▽	Created by ▽	Record count ▽	Description ▽	Hosted zone ID ▽
○	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.

| Edit | Delete | Import zone file | **Create record** |

Q Filter records by property or value

Type ▼   Routing policy ▼   Alias ▼

< 1 >

	Record name ▽	Type ▽	Routing policy ▽	Differe ntiator ▽	Alias ▽	Value/Route traffic to ▽	TTL (second s) ▽	Health check ▽	Evaluate target health ▽	Record ID ▽
☐	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	-	-	-
☐	parvathipai.com	SOA	Simple	-	No	ns-1937.awsdns-50.co.uk. awsdns-hostmaster.amazon.com. 1 7200 900 1209600 86400	900	-	-	-
☐	proj1.parvathipai.com	A	Simple	-	Yes	testenv.eba-ia8vjidb.us-west-2.elasticbeanstalk.com.	-	-	Yes	-
☐	_0a269bd0eced82ebc 85dd762f1be54ac.pr oj1.parvathipai.com	CNAME	Simple	-	No	_062f83c90f6aff1983d1b5421 e4a1dcb.zbkrxsrfvj.acm-validations.aws.	300	-	-	-

RDS
SQL database

View in DataGrip

EC2 instances

Load balancer

Auto Scaling groups

Cognito
App client settings

References

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