

**Sri Lanka Institute of Information Technology Master of Science in Information Technology**

**Assignment**

MS19812090: D.E.Malawana

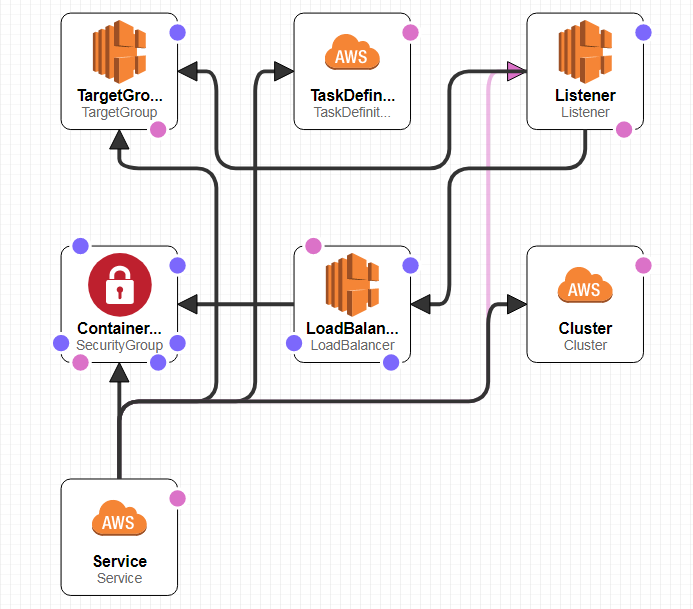
Module: Architecting Enterprise Cloud Solutions (SE6020)

Term: 3rd Semester

Date of Submission: 31st October 2020

**Faculty of Graduate Studies and Research Sri Lanka Institute of Information Technology**

Architecture Diagram



This diagram is generated by stack Designer page in the aws cloudformation .

Mainly there are following things,

Parameters:

* ServiceName
* VpcId
* IAMRole
* DockerImage
* SN1-subnet1
* SN2-subnet2

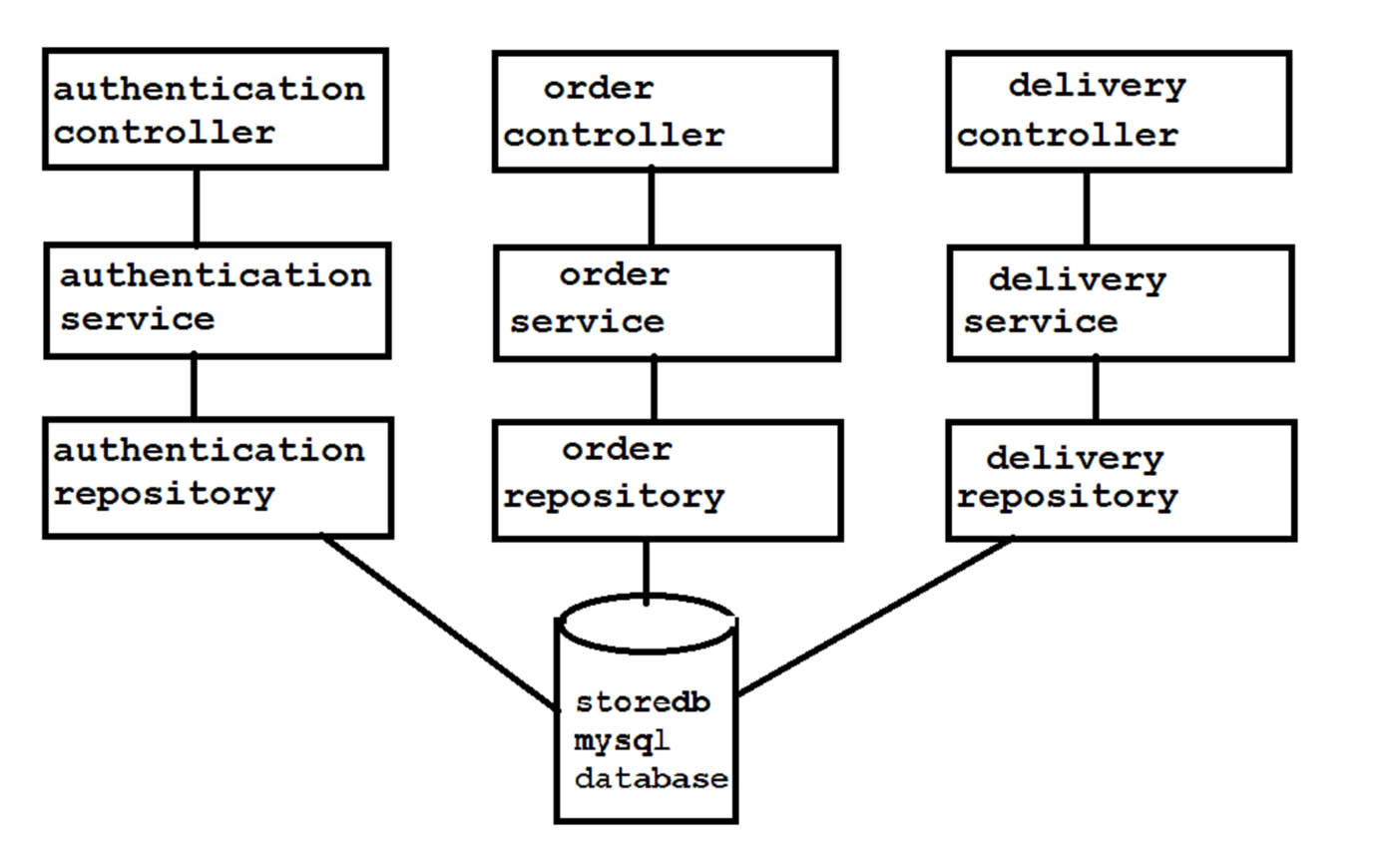
Resources:

* Cluster
* Security group
* Task definition
* Service
* Target group
* Load balancer
* Listener

In this project, Spring boot is used to implement microservices. Database is mysql.

There are three tables in the stored database. Those are authentication, order and delivery tables.

Microservice architecture diagram is shown below,



These services are running inside the ECR (Container registry).



Creating the containerized images of services,

1. Authentication service

Using following commands creates docker images and push into ECR. This commands are wrote in the ECR.cmd file. When it double click, all steps are run automatically.

* call aws ecr get-login-password --region us-east-2 | docker login --username AWS --password-stdin 593203563401.dkr.ecr.us-east-2.amazonaws.com
* call aws ecr create-repository --repository-name aecs-auth --image-scanning-configuration scanOnPush=true --region us-east-2
* call docker build -t aecs-auth .
* call docker tag aecs-port:latest 593203563401.dkr.ecr.us-east-2.amazonaws.com/aecs-auth:latest
* call docker push 593203563401.dkr.ecr.us-east-2.amazonaws.com/aecs-auth:latest

1. Order service

Using following commands creates docker images and push into ECR. This commands are wrote in the ECR.cmd file. When it double click, all steps are run automatically.

* call aws ecr get-login-password --region us-east-2 | docker login --username AWS --password-stdin 593203563401.dkr.ecr.us-east-2.amazonaws.com
* call aws ecr create-repository --repository-name aecs-order --image-scanning-configuration scanOnPush=true --region us-east-2
* call docker build -t aecs-order .
* call docker tag aecs-port:latest 593203563401.dkr.ecr.us-east-2.amazonaws.com/aecs-order :latest
* call docker push 593203563401.dkr.ecr.us-east-2.amazonaws.com/aecs-order :latest

1. Delivery service

Using following commands creates a Docker images and push into ECR. This commands are wrote in the ECR.cmd file. When it double click, all steps are run automatically.

* call aws ecr get-login-password --region us-east-2 | docker login --username AWS --password-stdin 593203563401.dkr.ecr.us-east-2.amazonaws.com
* call aws ecr create-repository --repository-name aecs-delivery --image-scanning-configuration scanOnPush=true --region us-east-2
* call docker build -t aecs-delivery .
* call docker tag aecs-port:latest 593203563401.dkr.ecr.us-east-2.amazonaws.com/aecs-delivery :latest
* call docker push 593203563401.dkr.ecr.us-east-2.amazonaws.com/aecs-delivery :latest

After create the ECR, flowing codes are run to create stack using cloud formation. These commands are in aecs.cmd file.

* call aws cloudformation create-stack --stack-name auth --template-body file://AECS\_Store\_infastructure.yaml --parameters ParameterKey=ServiceName,ParameterValue=auth-service --capabilities CAPABILITY\_NAMED\_IAM
* call aws cloudformation create-stack --stack-name order --template-body file://AECS\_Store\_infastructure.yaml --parameters ParameterKey=ServiceName,ParameterValue=order-service --capabilities CAPABILITY\_NAMED\_IAM
* call aws cloudformation create-stack --stack-name delivery --template-body file://AECS\_Store\_infastructure.yaml --parameters ParameterKey=ServiceName,ParameterValue=delivery-service --capabilities CAPABILITY\_NAMED\_IAM

Related details,

Used database details : RDS-Mysql

Endpoint: storedb.cf9twwftgada.us-east-2.rds.amazonaws.com

Port : 3306

Availability zone

us-east-2a