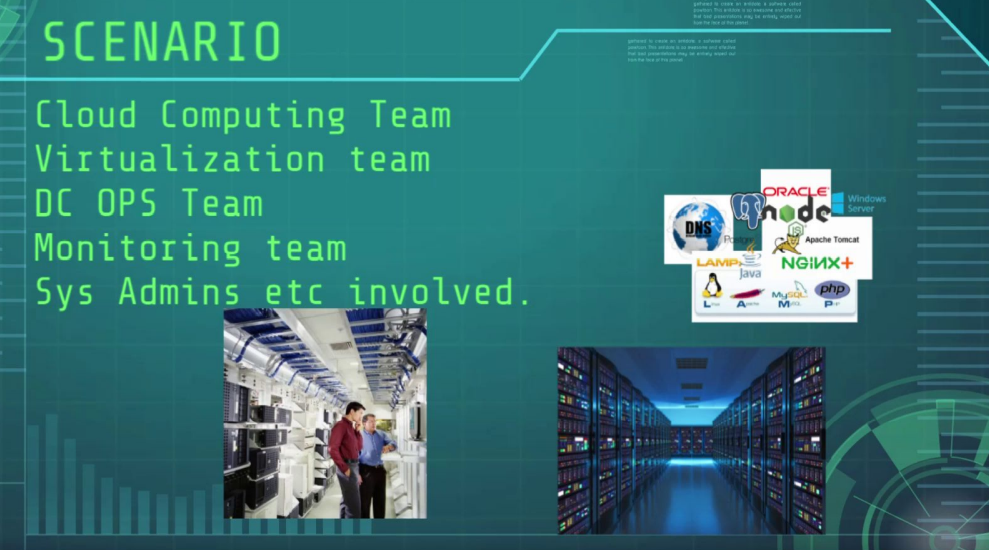
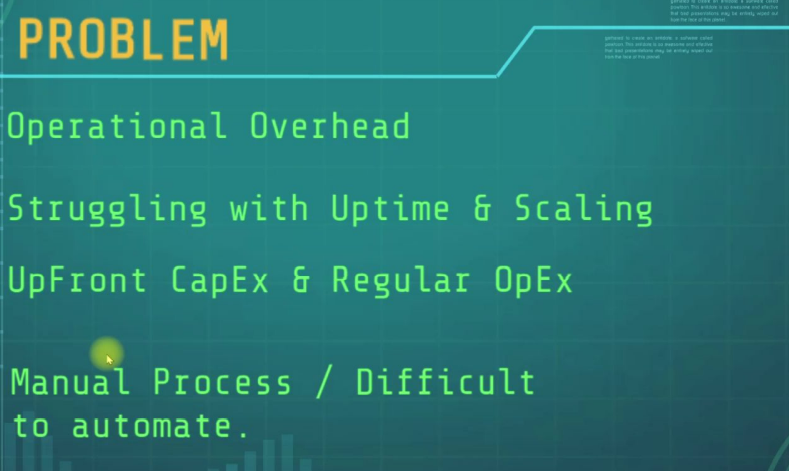
Refactoring with AWS is also known as as Re-Architecting your AWS. From previous project of Lift & Shift of our Artifacts throught S3 Buckets, We would:

* Re-Architect Services for AWS Cloud
* Architecture to boost agility or improve business community

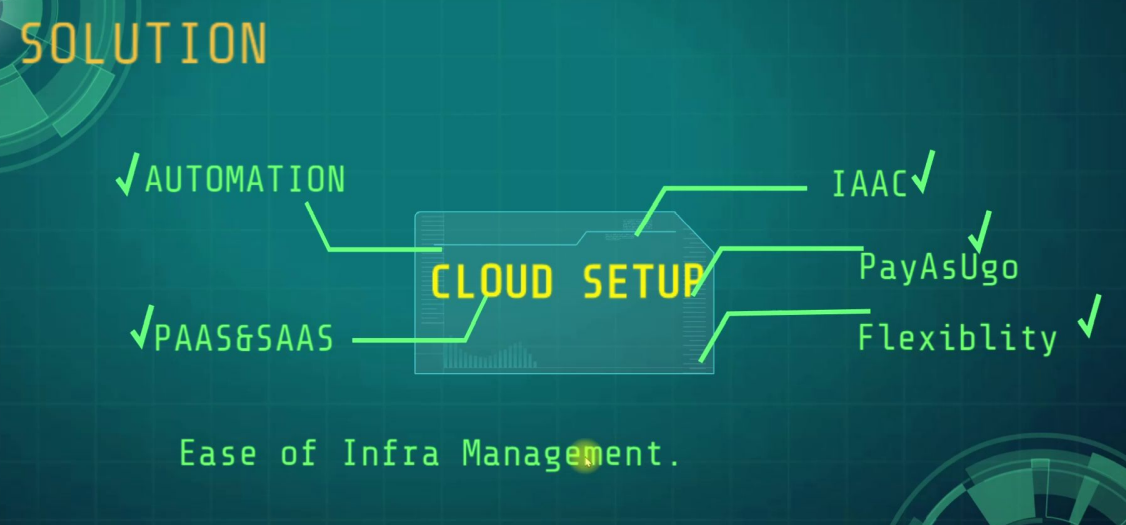




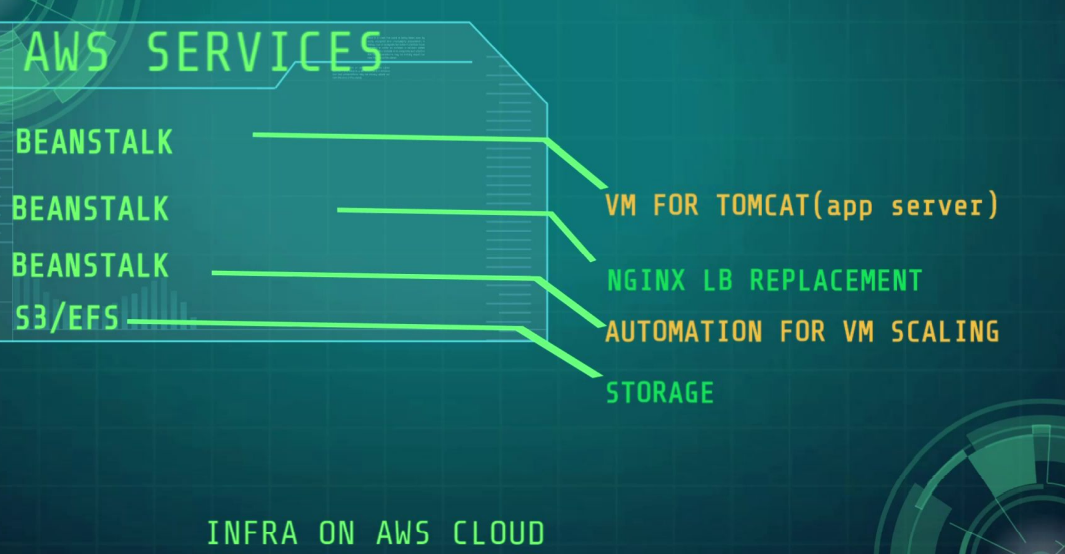


CapEx – Capital Expenditure

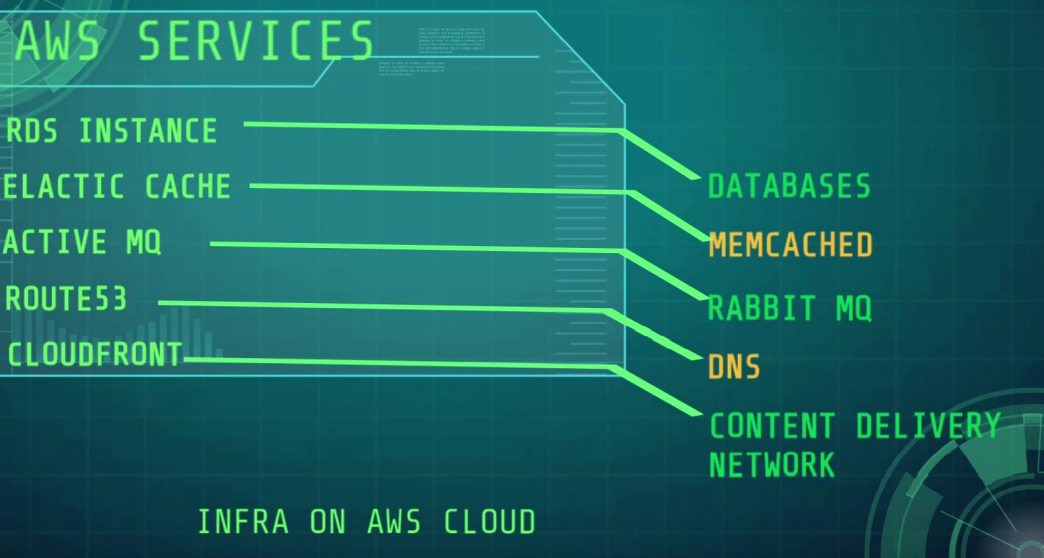
OpEx – Operating Expenditure

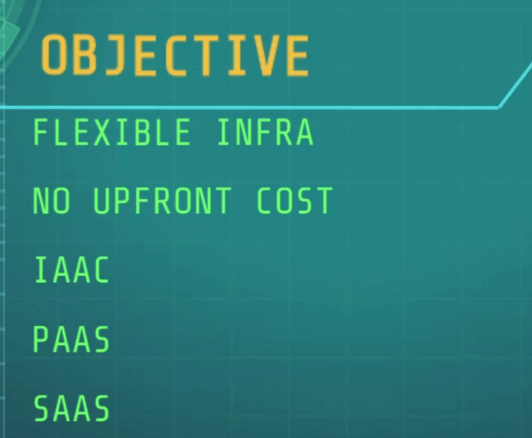


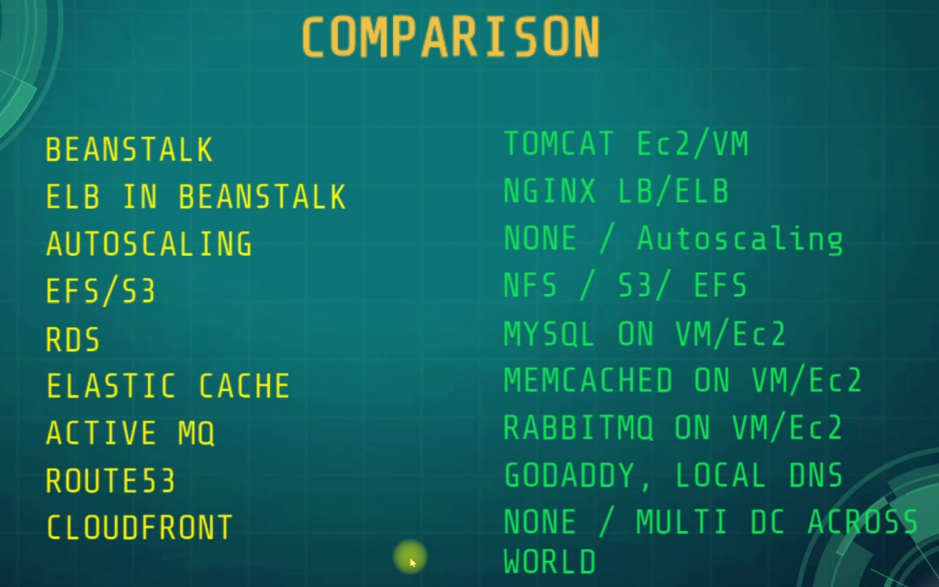
FOR THE FRONTEND:



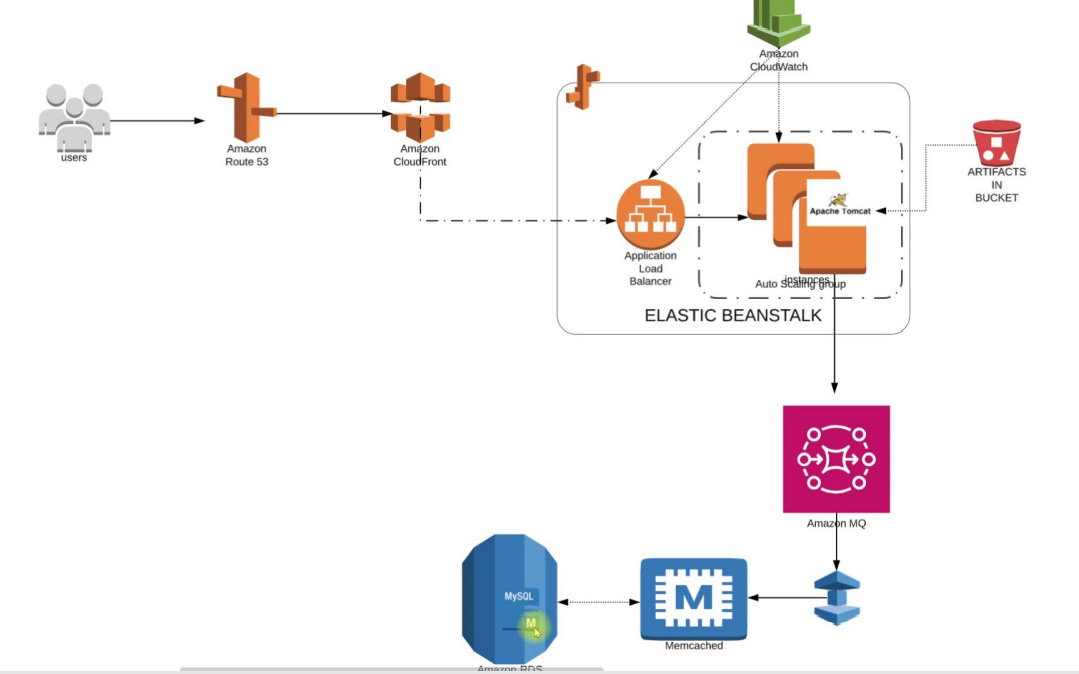
FOR THE BACKEND:

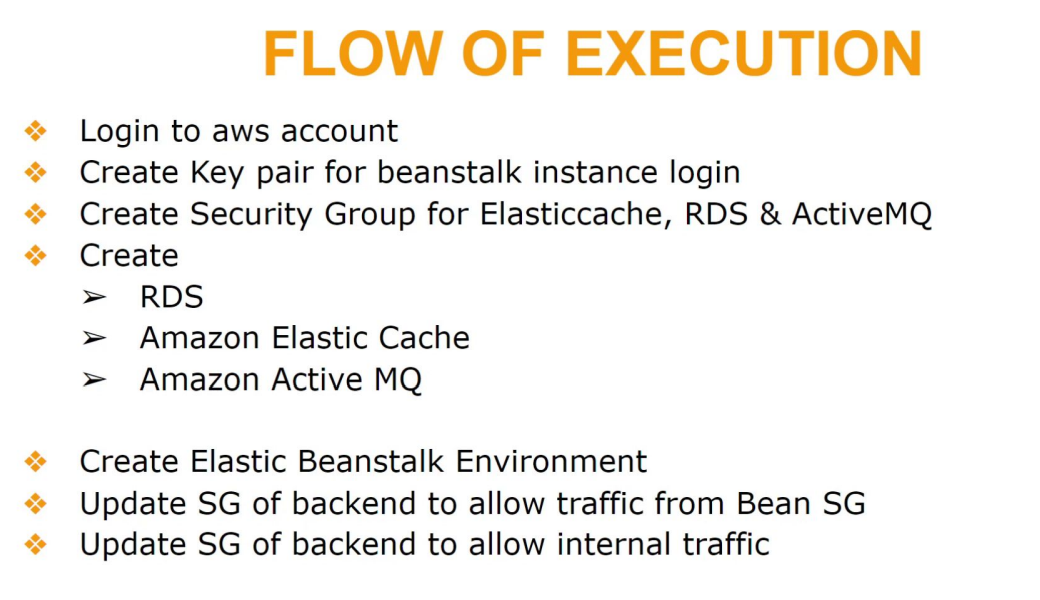


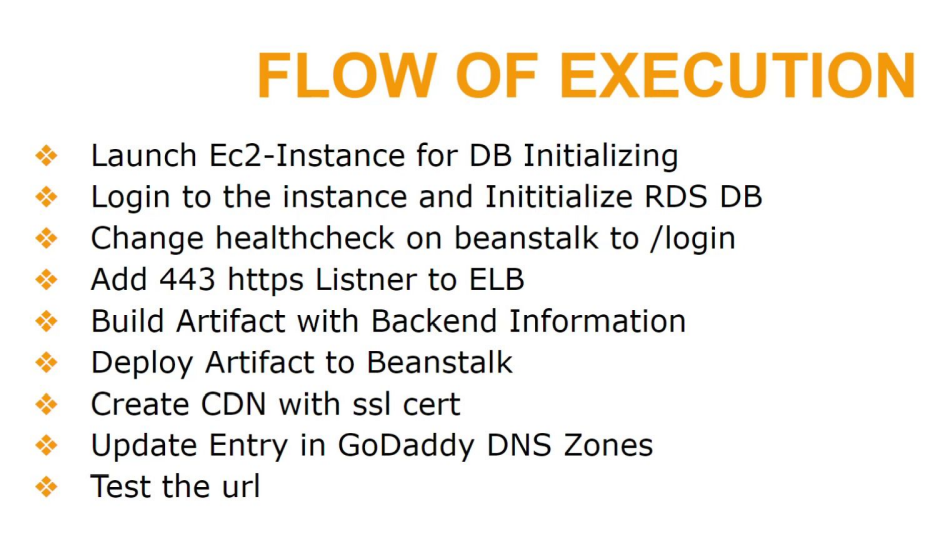












STEPS TO EXECUTION:

1.) Login to AWS Console

2.)Create Key pair for beanstalk instance login: (Name=vprofile-bean-key)

3.)Create Security Group for Elasticcache, RDS, & ActiveMQ

Name=vprofile-backend-Sg …….. inbound rule=All traffic allowed from its own security group backend

4.)Create RDS, Amazon Elastic Cache, Amazon Active MQ

RDS:

* Go to RDS service …… click on create DBsubnet groups(Name=vprofile-rds-sub-group, Availability zones=select all, Subnets=All subnets)
* Go to parameter groups: (parameter group family=mysql5.7 …. Name=vprofile-rds-para-grp ……… click create)
* Click on Databases which is on the left pane …… click create(standard create, MySQL) ….. version=mysql5.7.22 ……. Templates=Freetier ….. DB instance identifier=vprofile-rds-mysql ……… DB instance class=Burstable t2 micro …… Multi A-Z deployment(create) ……. Subnet group=Select your own created subnet ….. public Access=No …… Security Group=Select your Backend-SG …… port No=3306 ….. initial database name=accounts …… DB parameter group=Select yours …. click create database …… click on view credential details above and make sure you copy to a text file

AMAZON ELASTIC CACHE

* Search for Elastic cache
* click parameter group on the left pane ….. click create parameter (Name=vprofile-memcached-para-grp … family=memcached 1.4 or 1.5 ….. click on create)
* Click on subnet groups on the left pane …. create subnet group (Name=vprofile-memcached-sub-grp) …… selected subnets (click on manage and make sure all is selected) … create
* Go to Elastic cache main page and click on get started …. under create a cluster drop down, click on create a memcached cluster (Location=AWS Cloud ……. Name=vprofile-elasticcache-service ….. Engine version=1.4.5 ….. parameter group=Select yours you created earlier ……. port=11211 …. Node type=t2.micro …. number of nodes=1 ……subnet group setting=Select yours you created …… Availability zone=no preference …. click next ….. security groups, click manage and select backend-SG …. Tag (Name=vprofile-elasticache-svc ….. click next …. review your settings and click create)

AMAZON ACTIVE MQ

* Search for AmazonMQ under services ….. click on get started ….. broker engine type=RabbitMQ ….. Deployment mode=Single-Instance broker (If it is production you need to select cluster deployment) ….. click next …… Broker name=vprofile-rmq …… Broker instance type=t3 micro ….. RabbitMQ access (Username=rabbit password=rabbit) use any username and pass of your choice and make sure you remember it or else you can’t see it again ….. Additional setting (Broker engine version=3.9.16, Network and security=Private access, Select your existing security group you created) …… Add Tag (Name=vprofile-rmq01) ….. Review your settings and click create broker

**DB Initialization (RDS) through EC2 instance:**

* Go to RDS service and check if it is ready, then copy the endpoint url which you will find when you scroll down to connectivity & security paste it somewhere(word or notepad)
* Go to EC2 instance and click on create …… Name=mysql-client, Ubuntu 18, keypair=use same one you create (vprofile-bean-key), create a different security group (mysqlClient-SG, Inbound rules=22 allowed from my ip), additional information: provision with codes below

#!/bin/bash

sudo apt update

sudo apt install mysql-client –y (Click Launch Instance)

* Get the public IP of the Instance and ssh into it.(ssh –i Downloads/vprofile-bean-key.pem ubuntu@public Ip)
* systemctl start mysql-client
* systemctl enable mysql-client
* systemctl status mysql-client
* mysql –h (paste the RDS endpoint copied earlier) –u rabbit –prabbit
* If it doesn’t connection, go to the security group of the backend instance (Mysql/Aurora tcp 3306 allow from mysql security group)
* mysql –h (paste the RDS endpoint copied earlier) –u rabbit –prabbit
* show databases;
* clone the source code: git clone <https://github.com/devopshydclub/vprofile-project.git>
* ls
* cd vprofile-project/ …… git branch –a …….. git checkout aws-Refactor ….. ls
* cd src/main/resources … ls
* mysql –h (paste the RDS endpoint copied earlier) –u rabbit –prabbit accounts < db\_backup.sql
* mysql –h (paste the RDS endpoint copied earlier) –u rabbit –prabbit accounts
* show tables;
* exit (RDS is initialized)

Create the Beanstalk Environment

* Go to Amazon RDS and fetch its endpoint and port no (Copy and Paste in a Notepad)
* Go to Amazon ActiveMQ and fetch its endpoint from the {// to .com} only and port no (Copy and Paste in a Notepad)
* Go to Amazon ElasticCache and fetch its endpoint and port no (Copy and Paste in a Notepad)
* Go to Elastic Beanstalk service ….. click on create application (Application Name:vprofile-java-app, Tag—Project=vprofile, Platform=Tomcat, Application code=sample application, click on configure more options (Presets=Custom configuration, click on edit for Instances🡺 Root volume type=Container default, Ec2 Security groups=backend-SG, click save ………. Click on edit for Capacity🡺 Environment type=Load Balanced, Instances=2min & 8max, instance type=t2 micro, Availability zones=leave as default, metric=Networkout ……. Click on Edit Rolling Updates and Deployments🡺 Deployment policy=Rolling, Batch size=50%, everything else should be left as default, click save …….. Click on edit Security🡺Service role=aws-elasticbeanstalk-service-role, EC2 keypair= vprofile-bean-key, IAM instance profile=aws-elasticbeanstalk-ec2-role if it is not there, leave it empty, click save ………………… Click on edit Notification🡺 put your email …….. Click on edit Tag🡺 project=vprofile, click save ……… you are done, click create app.
* After which you have clicked create App, If you see Terminated, it means it the first time you created it in that region. And you missed out on some key settings. Repeat the steps again.

UPDATE THE SECURITY GROUPS & ELB

* Go to EC2, you should see two instances created by Beanstalk
* Go to backend security group and click on inbound rules and:

add rule for beanstalk(custom tcp 3306 from security group of vprofile-java-app-prod)

add rule for Elastic Cache(custom tcp 11211 from security group of vprofile-java-app-prod)

add rule for AmazonMQ(custom tcp 5671 from security group of vprofile-java-app-prod) ……… Save Rules

* Go to Elastic Beanstalk, click on Environment on the left pane, click on your environment you created, click on configuration and scroll down to Load balancer then click on Edit, click on add listener(port=443, HTTPS, set your SSL certificate then click on add) …… scroll down to Processes and click on the box to check mark it, click on Action and edit, scroll down to health check(path=/login, scroll down and enable stickiness, then click save) ….. after everything, click on apply. The Health status should be on Severe, after deploying our Artifact it would change to OK!

BUILDING AND DEPLOYING OUR ARTIFACT

* cd /c/ ……. your git dir
* clone source code: git clone <https://github.com/devopshydclub/vprofile-project.git>
* ls … cd vprofile-project/ ,,,,,, git pull (in case of changes)
* git branch –a …to show all branches
* git checkout aws-refactor
* make sure you have maven and jdk installed
* ls
* cd src/main/resources/ ……. to application dir
* ls
* vim application.properties

Edit the line with yellow with Beanstalk URL details : (jdbc.url=jdbc:mysql:// (RDS endpoint):3306/accounts?useUnicode=true&characterEncoding=UTF-8&zeroDateTimeBehavior=convertToNull)

memcached.active.host=(Elastic Cache endpoint)

rabbitmq.address=(ActiveMQ endpoint) …….. save and quit

* cd vprofile-project/ …… ls
* mvn install
* ls target/ ……….this is were our artifact lies
* Go to Elastic Beanstalk …… click on Application versions on the left pane ……. click on upload (Version label=vprofile-v2.5, click on choose file and go to the dir where you saved the artifact, select and upload)
* Select the uploaded artifact, click on action and click deploy
* Click on Environment on the left pane and check the health status, it should be ok now.
* Go to beanstalk and click on configuration on the left pane, click Edit for Load Balancer and scroll down to processes, check mark it and click on Actions and click edit, scroll down to enable stickiness and mark, save and next apply
* copy the Beanstalk endpoint and paste in a browser to check status of all services if done well
* Go to Godaddy on domain setting, go to DNS Management and click Add record:

(CNAME HOST=vprofile ENDPOINT=beanstalk endpoint copy it)

* Check: <https://vprofile.(your> domain name)

Cloud front: (It is the Content Delivery Network for AWS)

Used for distributing your content around the world

* Go to Cloud front, click on create distribution, …… Origin domain=<https://vprofile.(your> domain name), protocol=Match viewer …… scroll down to Viewer protocol policy=HTTP and HTTPS, ………… Allowed Http methods=Get,Head,Options,Put,Post,Patch,Delete ……. scroll down to settings(use all edge locations, scroll down to alternate domain name(<https://vprofile.(your> domain name)), custom SSL certificate=choose your main cert from the list, scroll down to security policy=TLSv1, …….. Click on create Distribution