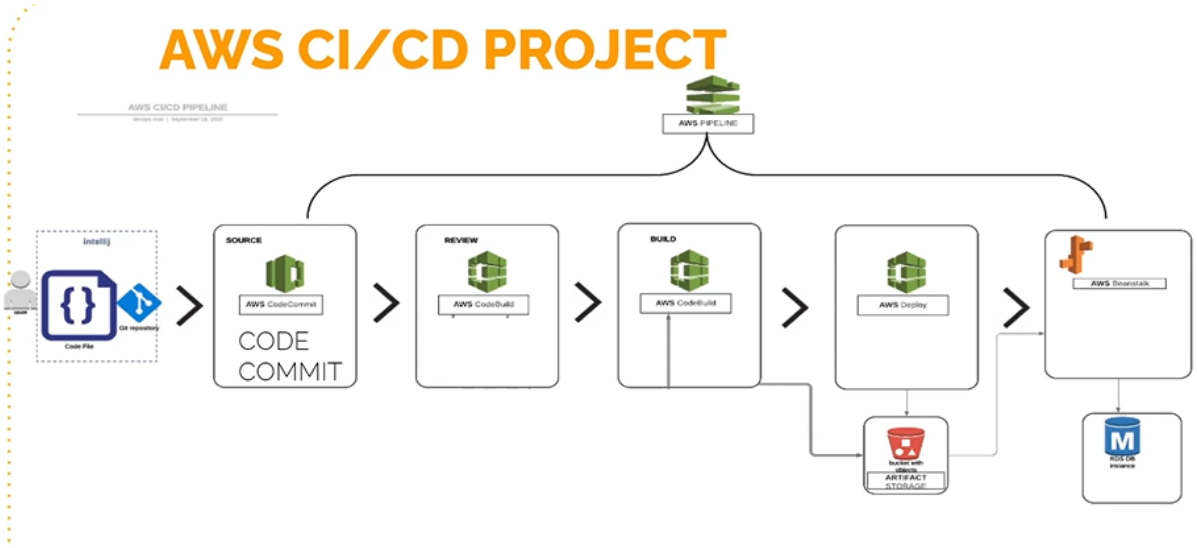
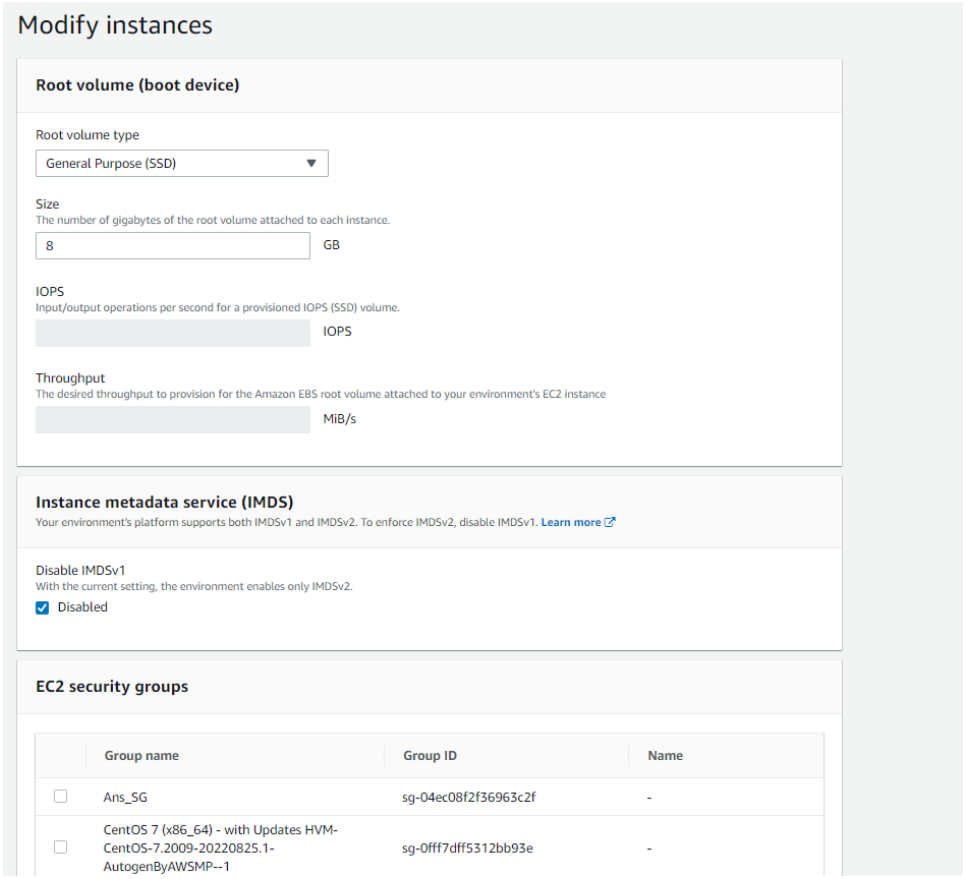
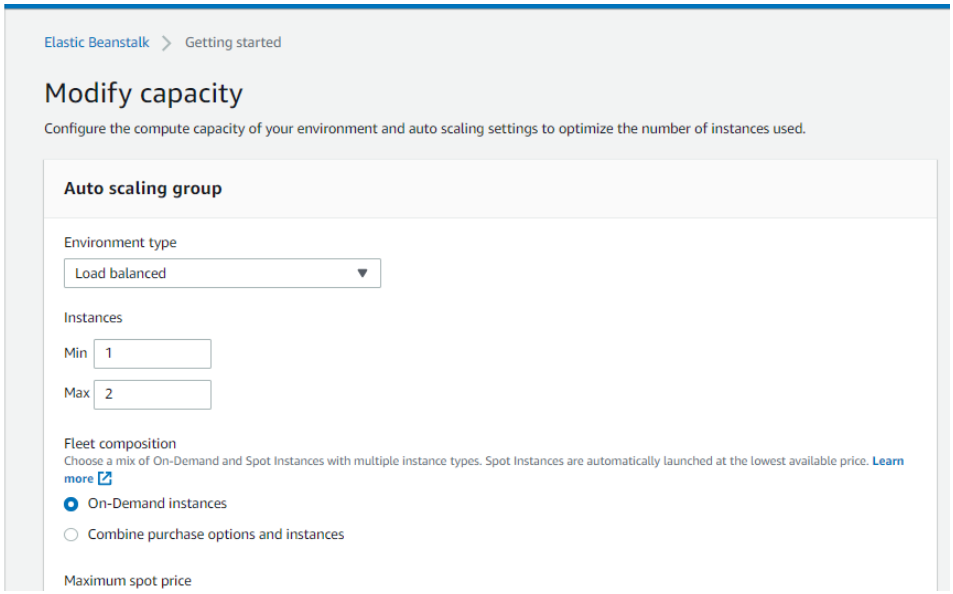
AWS CICD Project

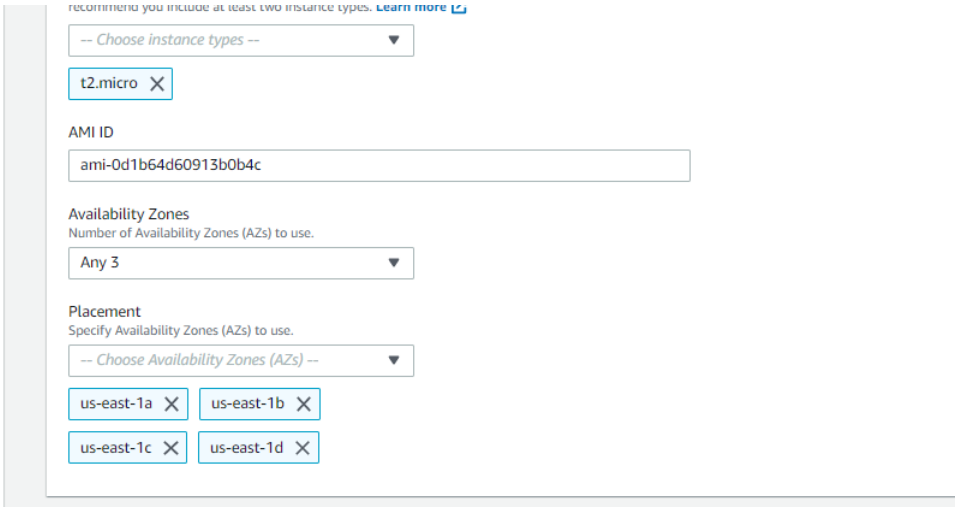
This project demonstrates the AWS CI/CD pipeline from committing code to deploy the application in Elastic beanstalk. A pipeline is setup to consume the code from AWS Code commit (repository + branch – this is based on CloudWatch logs), build the code using Maven on AWS code build (build spec, db details are in application. Prop file) and the artifacts to be uploaded in the S3 Bucket. All logs will be streamed to CloudWatch logs. The next step in the pipeline is to deploy the application in elastic beanstalk. Whenever a code commit happened on AWS code commit, a new pipeline will get triggered.

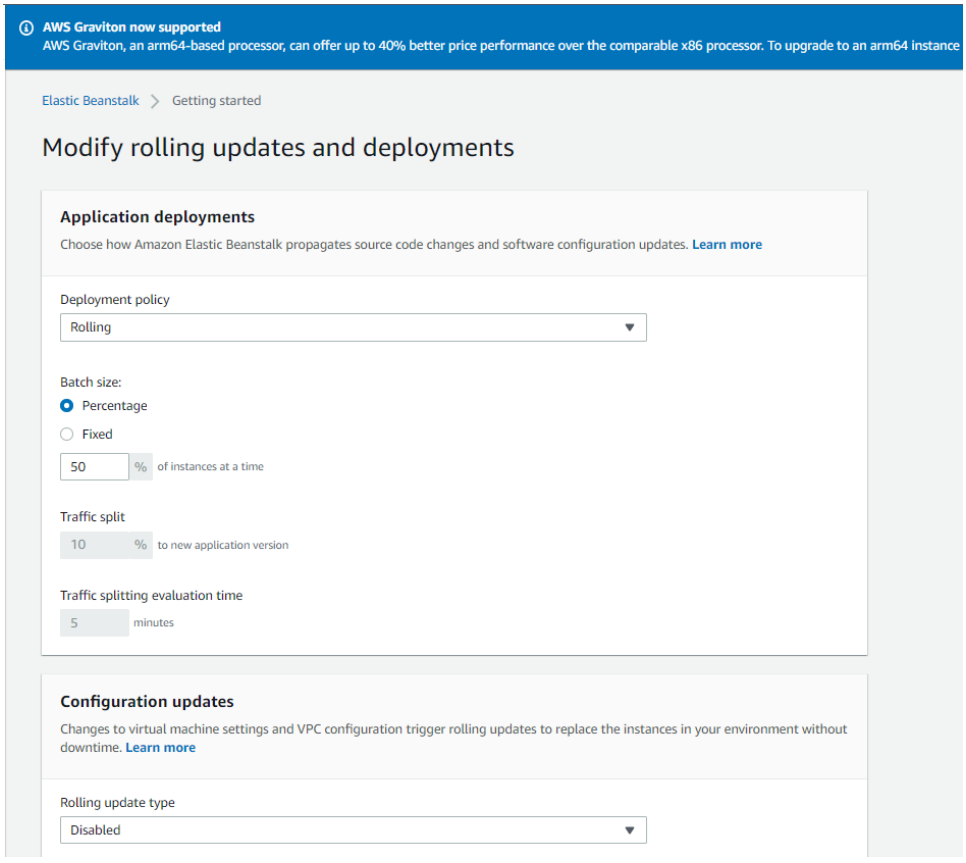


1. Create an Elasticbeanstalk service

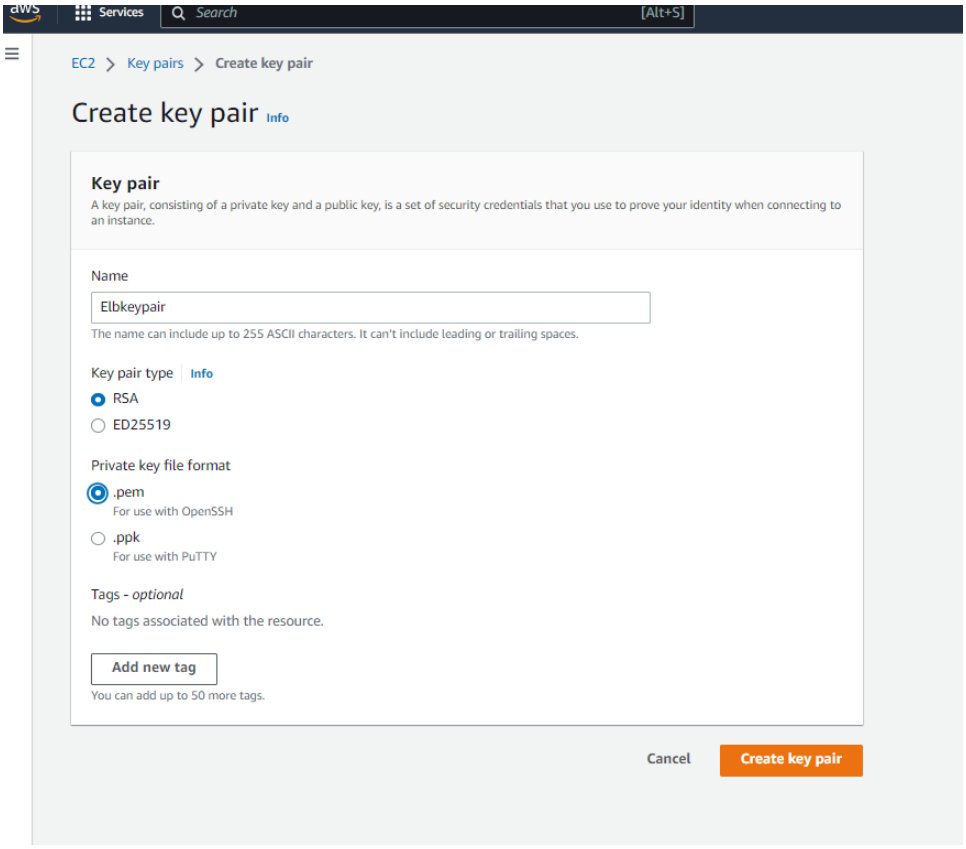

Modify instances
Root volume (boot device)
Root volume type
General Purpose (SSD)
Size
The number of gigabytes of the root volume attached to each instance.
8
GB
IOPS
Input/output operations per second for a provisioned IOPS (SSD) volume.
IOPS
Throughput
The desired throughput to provision for the Amazon EBS root volume attached to your environment's EC2 instance
MiB/s
Instance metadata service (IMDS)
Your environment's platform supports both IMDSv1 and IMDSv2. To enforce IMDSv2, disable IMDSv1. Learn more
Disable IMDSv1
With the current setting, the environment enables only IMDSv2.
Disabled
EC2 security groups
Group name
Group ID
Name
Ans_SG
sg-04ec08f2f36963c2f
-
CentOS 7 (x86_64) - with Updates HVM-CentOS-7.2009-20220825.1-AutogenByAWSMP--1
sg-0fff7dff5312bb93e
-


Elastic Beanstalk
Getting started
Modify capacity
Configure the compute capacity of your environment and auto scaling settings to optimize the number of instances used.
Auto scaling group
Environment type
Load balanced
Instances
Min
1
Max
2
Fleet composition
Choose a mix of On-Demand and Spot Instances with multiple instance types. Spot Instances are automatically launched at the lowest available price. Learn more
On-Demand instances
Combine purchase options and instances
Maximum spot price

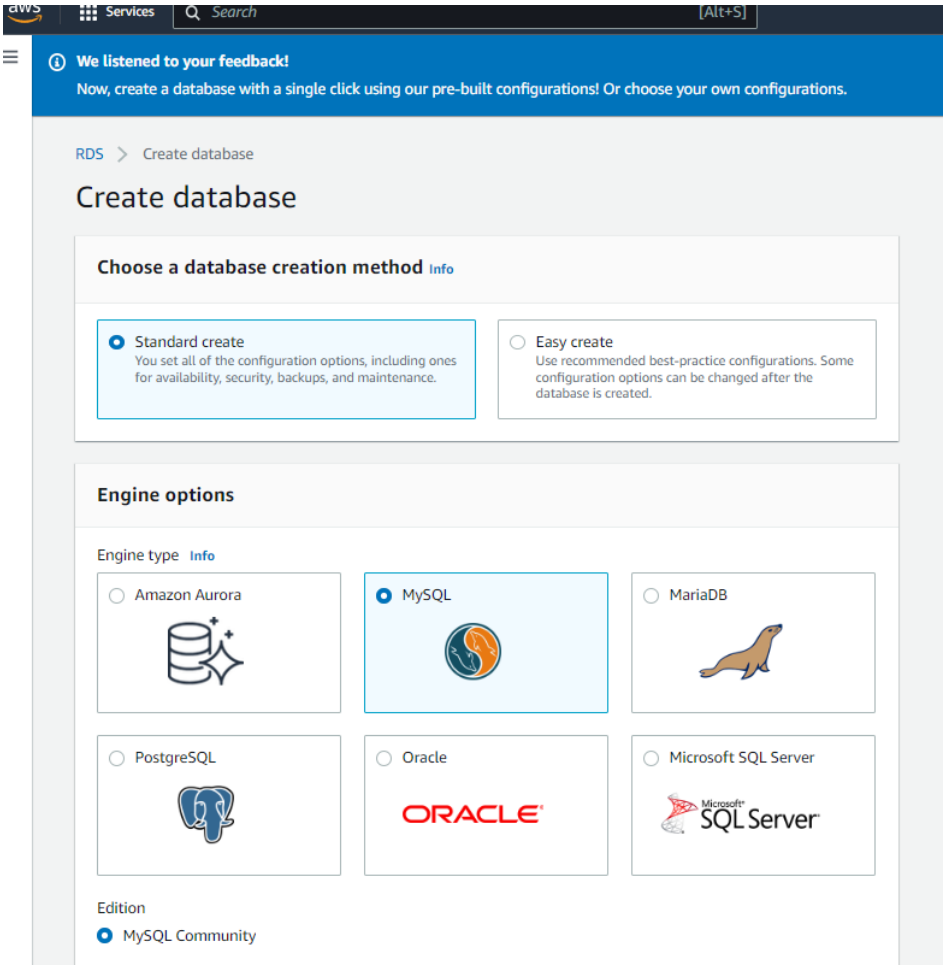


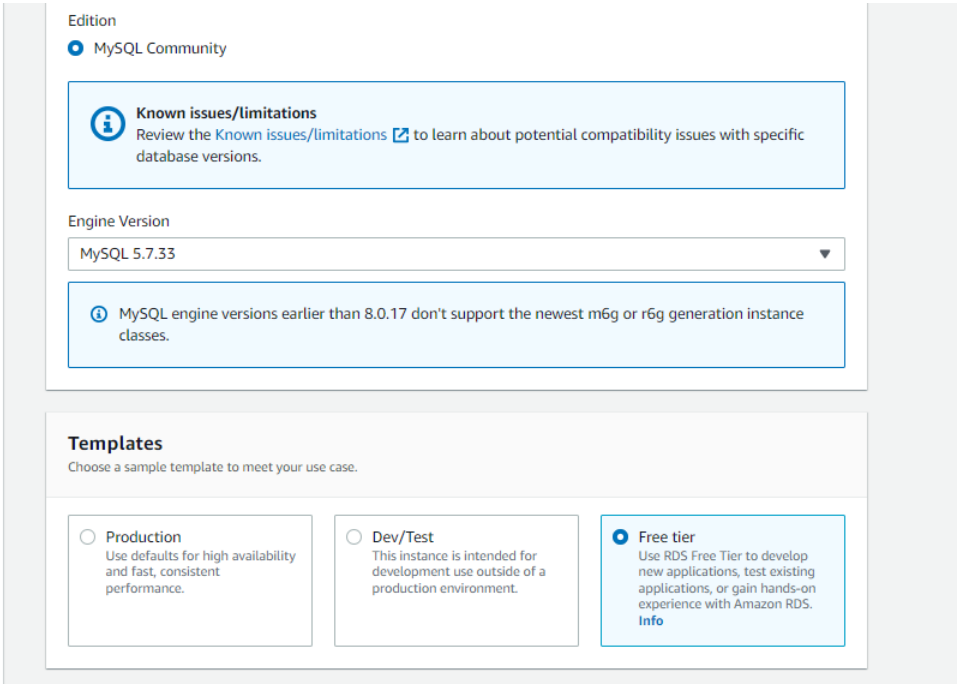


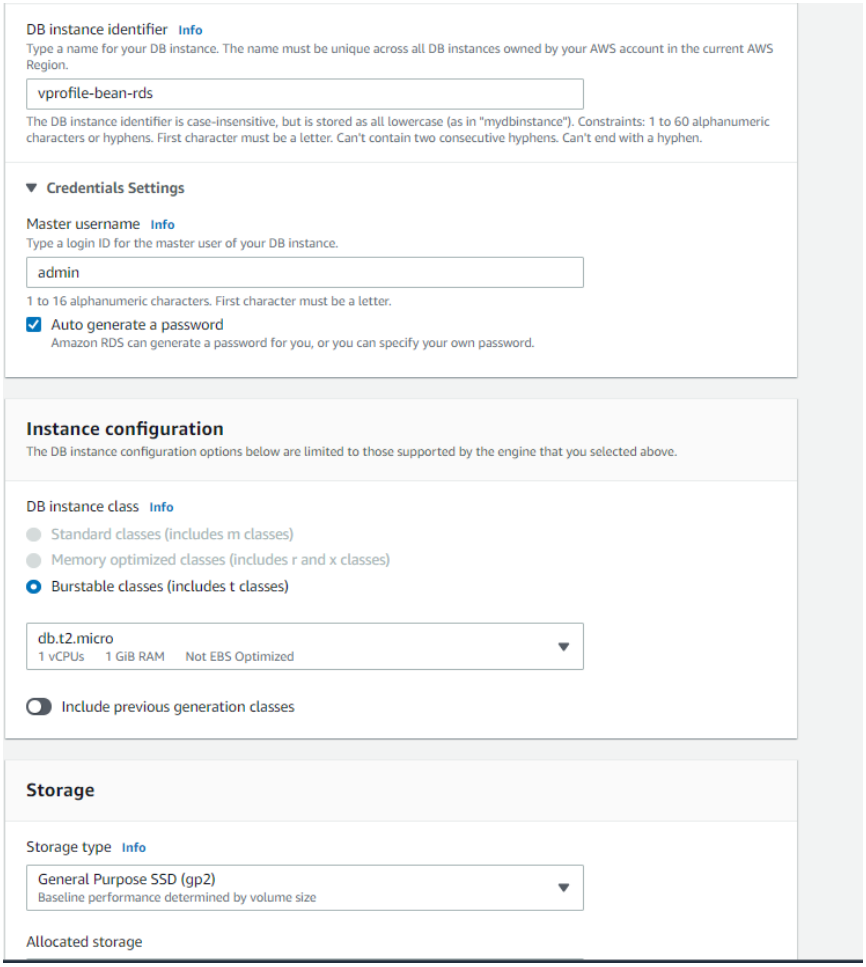
2. Create a key pair to use login ELB instance

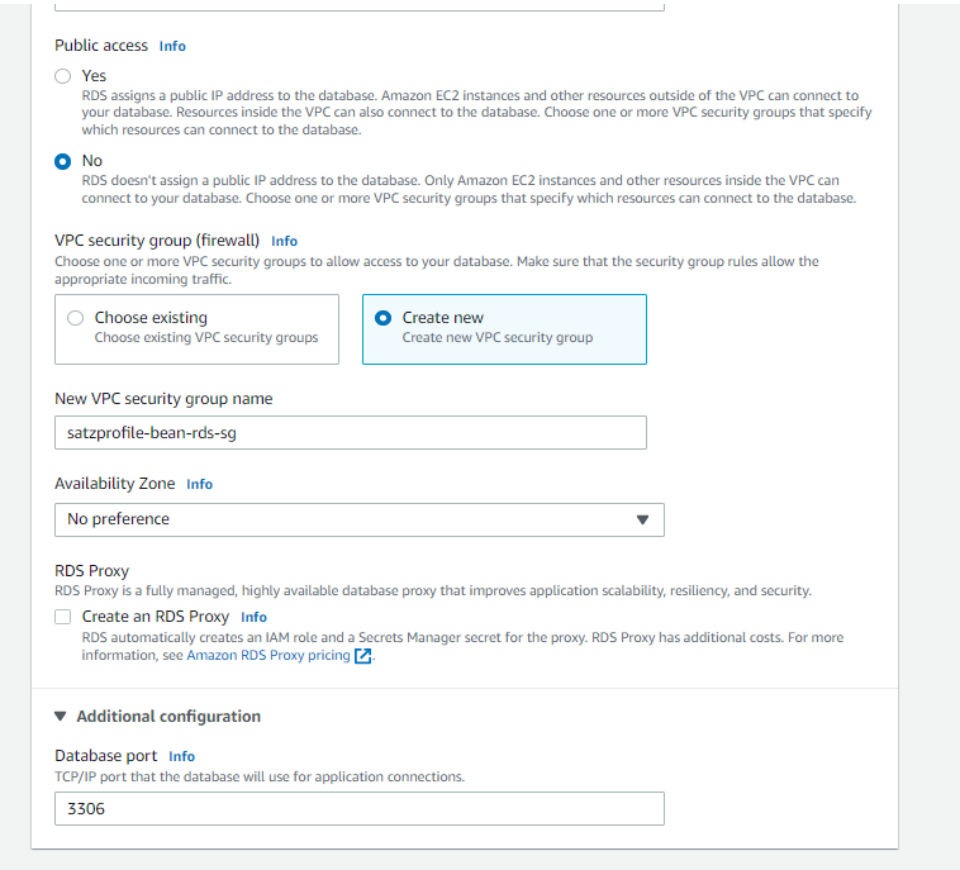


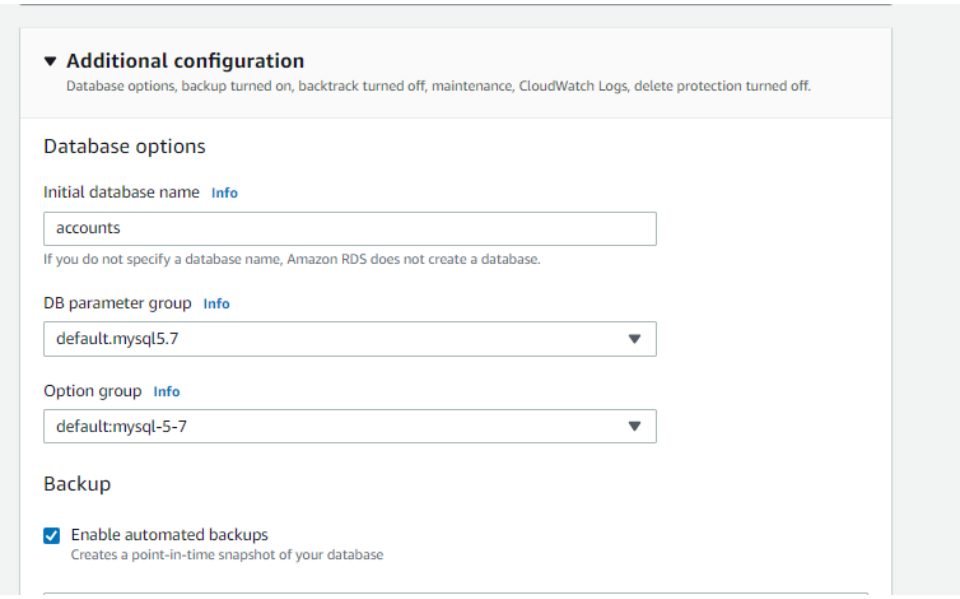
3. Create a RDS database


Services
Search
[Alt+S]
We listened to your feedback!
Now, create a database with a single click using our pre-built configurations! Or choose your own configurations.
RDS
Create database
Create database
Choose a database creation method Info
Standard create
You set all of the configuration options, including ones for availability, security, backups, and maintenance.
Easy create
Use recommended best-practice configurations. Some configuration options can be changed after the database is created.
Engine options
Engine type Info
Amazon Aurora
MySQL
MariaDB
PostgreSQL
Oracle
Microsoft SQL Server
Edition
MySQL Community


Edition
MySQL Community
Known issues/limitations
Review the Known issues/limitations to learn about potential compatibility issues with specific database versions.
Engine Version
MySQL 5.7.33
MySQL engine versions earlier than 8.0.17 don't support the newest m6g or r6g generation instance classes.
Templates
Choose a sample template to meet your use case.
Production
Use defaults for high availability and fast, consistent performance.
Dev/Test
This instance is intended for development use outside of a production environment.
Free tier
Use RDS Free Tier to develop new applications, test existing applications, or gain hands-on experience with Amazon RDS.
Info


DB instance identifier Info
Type a name for your DB instance. The name must be unique across all DB instances owned by your AWS account in the current AWS Region.
vprofile-bean-rds
The DB instance identifier is case-insensitive, but is stored as all lowercase (as in "mydbinstance"). Constraints: 1 to 60 alphanumeric characters or hyphens. First character must be a letter. Can't contain two consecutive hyphens. Can't end with a hyphen.
Credentials Settings
Master username Info
Type a login ID for the master user of your DB instance.
admin
1 to 16 alphanumeric characters. First character must be a letter.
Auto generate a password
Amazon RDS can generate a password for you, or you can specify your own password.
Instance configuration
The DB instance configuration options below are limited to those supported by the engine that you selected above.
DB instance class Info
Standard classes (includes m classes)
Memory optimized classes (includes r and x classes)
Burstable classes (includes t classes)
db.t2.micro
1 vCPUs 1 GiB RAM Not EBS Optimized
Include previous generation classes
Storage
Storage type Info
General Purpose SSD (gp2)
Baseline performance determined by volume size
Allocated storage





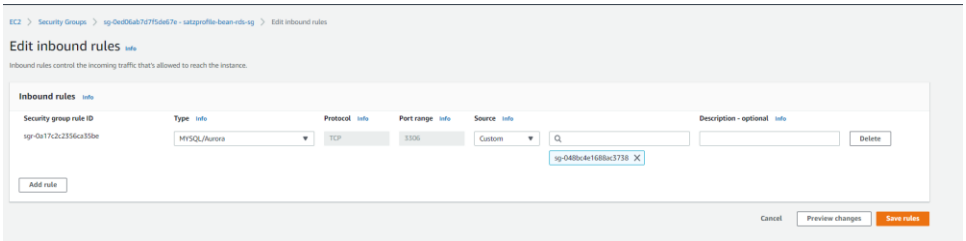
Make sure to Store the credentials

4. Now ELB will be ready, go to the EC2 instances, the corresponding security groups,

   Make sure to remove the inbound rule, that allow 22 traffic from anywhere. And allow only your IP

5. Now edit the inbound rule of RDS security group, and add ELB instance security group.

This will allow traffic from ELB to RDS



6. Using keypair, login to the ELB and preform below action.

   We are logging to an instance, download the code from git, and connect to the RDS instance..

   Can use any instance in the same VPC. It is not required to do it in ELB insatnce.

Become root and install git, mysql

```
 http://docs.aws.amazon.com/elasticbeanstalk/latest/dg/customize-containers-ec2.
html

[ec2-user@ip-172-31-31-138 ~]$ sudo -i
[root@ip-172-31-31-138 ~]# yum install git mysql -y
```

Clone the repo,

```
Complete!
[root@ip-172-31-31-138 ~]# git clone https://github.com/devopshydclub/vprofile-r
epo.git
```

```
Complete!
[root@ip-172-31-31-138 ~]# git clone https://github.com/devopshydclub/vprofile-r
epo.git
Cloning into 'vprofile-repo'...
remote: Enumerating objects: 1707, done.
remote: Counting objects: 100% (213/213), done.
remote: Compressing objects: 100% (70/70), done.
remote: Total 1707 (delta 176), reused 143 (delta 143), pack-reused 1494
Receiving objects: 100% (1707/1707), 42.34 MiB | 27.54 MiB/s, done.
Resolving deltas: 100% (615/615), done.
[root@ip-172-31-31-138 ~]# cd vprofile-repo/
[root@ip-172-31-31-138 vprofile-repo]# ls
ansible      Jenkinsfile  README.md  tomcat-setup.sh
context.xml  pom.xml      src        tomcat-users.xml
[root@ip-172-31-31-138 vprofile-repo]# git checkout vp-rem
branch 'vp-rem' set up to track 'origin/vp-rem'.
Switched to a new branch 'vp-rem'
[root@ip-172-31-31-138 vprofile-repo]# ls src/main/resources/db_backup.sql
src/main/resources/db_backup.sql
[root@ip-172-31-31-138 vprofile-repo]#
```

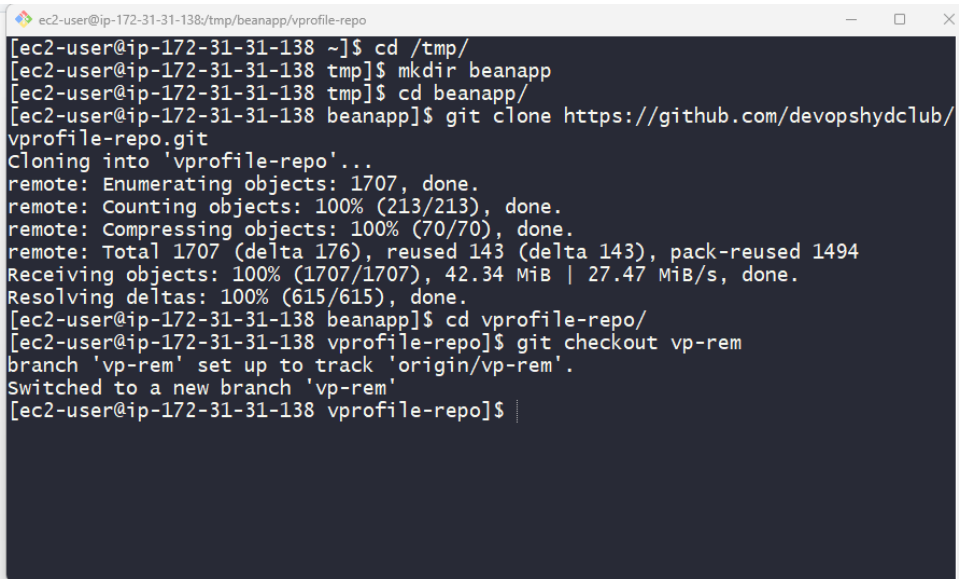
Now connect to the RDS instance,

```
[root@ip-172-31-31-138 vprofile-repo]# mysql -h vprofile-bean-rds.cs20txafmaaq.u
s-east-1.rds.amazonaws.com -u admin -pQzYcEtyLG7aq3rzzFJw accounts
ERROR 1045 (28000): Access denied for user 'admin'@'172.31.31.138' (using passwo
rd: YES)
[root@ip-172-31-31-138 vprofile-repo]# mysql -h vprofile-bean-rds.cs20txafmaaq.u
s-east-1.rds.amazonaws.com -u admin -ptQzYcEtyLG7aq3rzzFJw accounts
Welcome to the MariaDB monitor.  Commands end with ; or \g.
Your MySQL connection id is 27
Server version: 5.7.33-log Source distribution

Copyright (c) 2000, 2018, Oracle, MariaDB Corporation Ab and others.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

MySQL [accounts]>
```

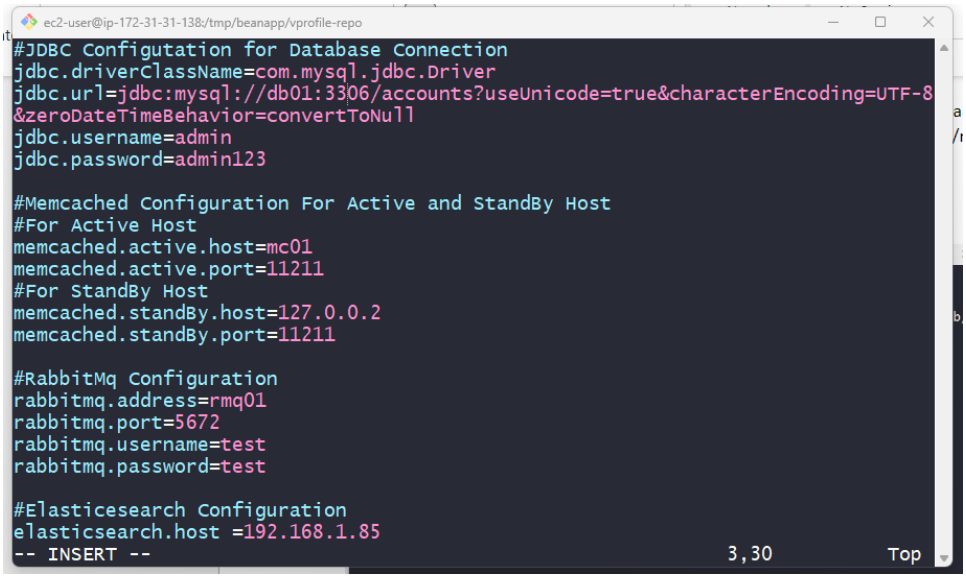
8.. Deploy Schemas

[root@ip-172-31-31-138 vprofile-repo]# mysql -h vprofile-bean-rds.cs20txafmaaq.us-east-1.rds.amazonaws.com -u admin -ptQzYcEtyLG7aq3rzzFJw accounts < src/main/resources/db_backup.sql

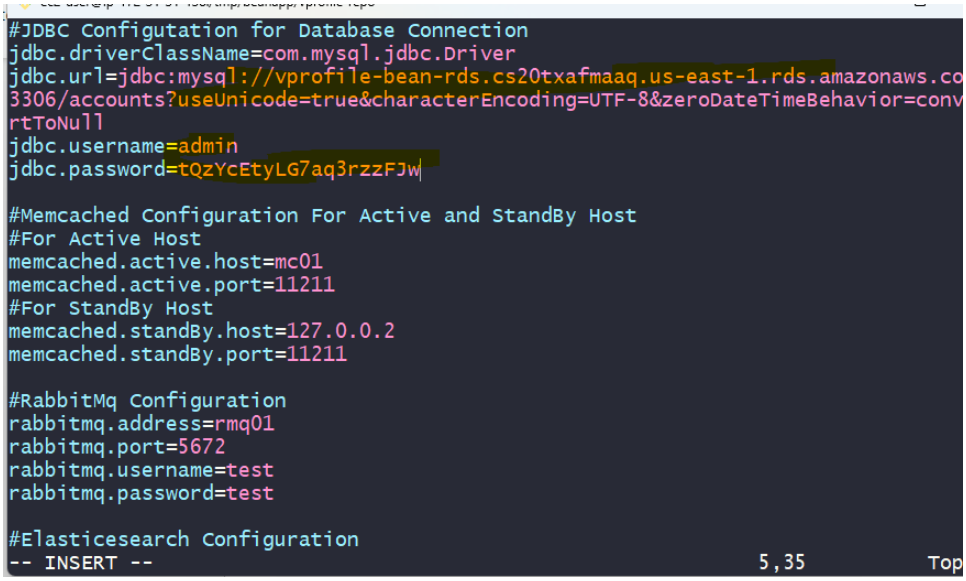
Logout and now go to the ec2-instance.



9. Get into the src/main/resources/application.properties

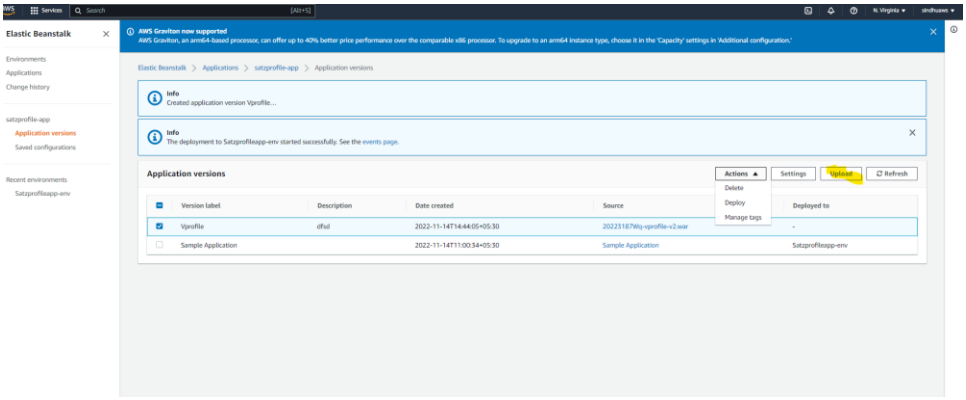


10. Update the RDS url, username and password,



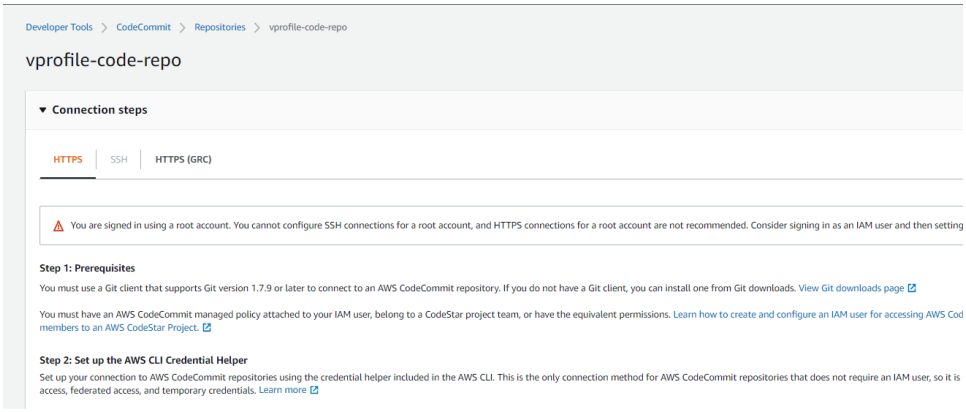
11. Install maven on the machine with root, refer
https://docs.aws.amazon.com/neptune/latest/userguide/iam-auth-connect-prerq.html

12. Download the war file to desktop, Now go to ELB and upload the artifacts, then deploy it to the environments,

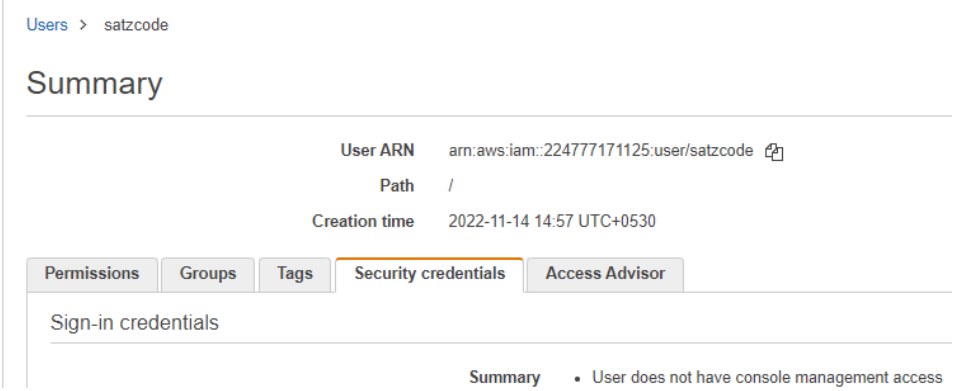


13. Code Commit:

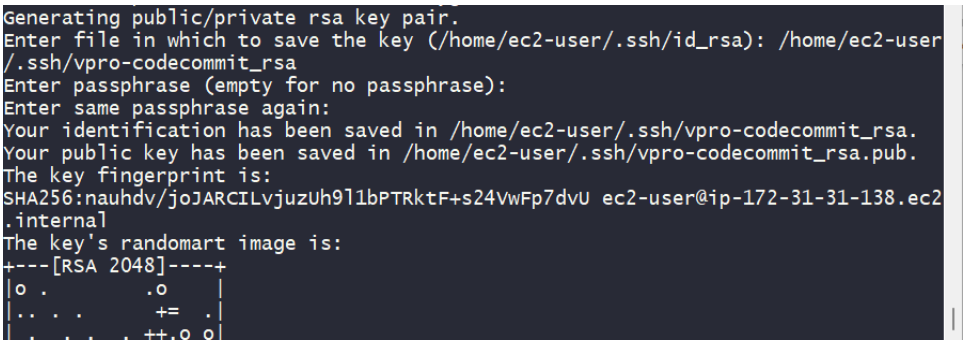
Create a code commit repository



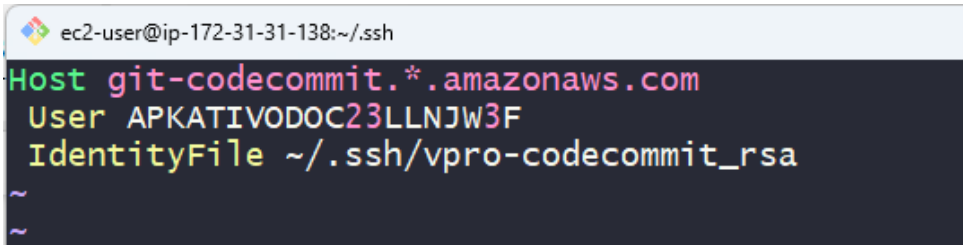
14. Create a new user with AWSCODECOMMIT full access assigned to only the repoistory



15. Generate ssh-keygen,



15. Create a config file with the below information under .ssh directory,



From terminal using below cmd, we can validate the connectivity to Code commit repo

ssh -v git-codecommit.us-east-2.amazonaws.com

Now update the remote repo url in your local file,

Cat .git/config

Now get the list of branches in the folder,

Git branch –a | grep –v HEAD | cut –d '/' -f3 | grep –v master > /tmp/branches

//To check out all the branches in git

for i in `cat /tmp/branches` ' do git checkout $I; done

// remove the old remote origin

git remote rm origin

//To add the new origin

git remote add origin ssh://git-codecommit.us-east-1.amazonaws.com/v1/repos/vprofile-code-repo

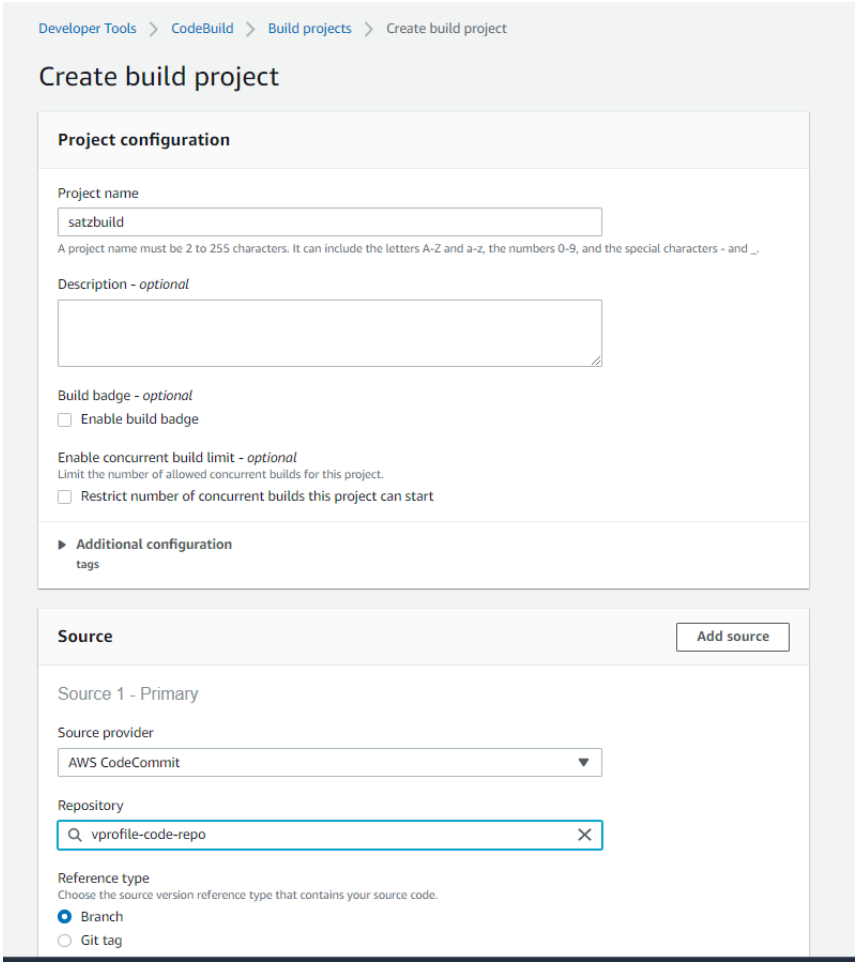
// git push to push all the branches on code commit repo.

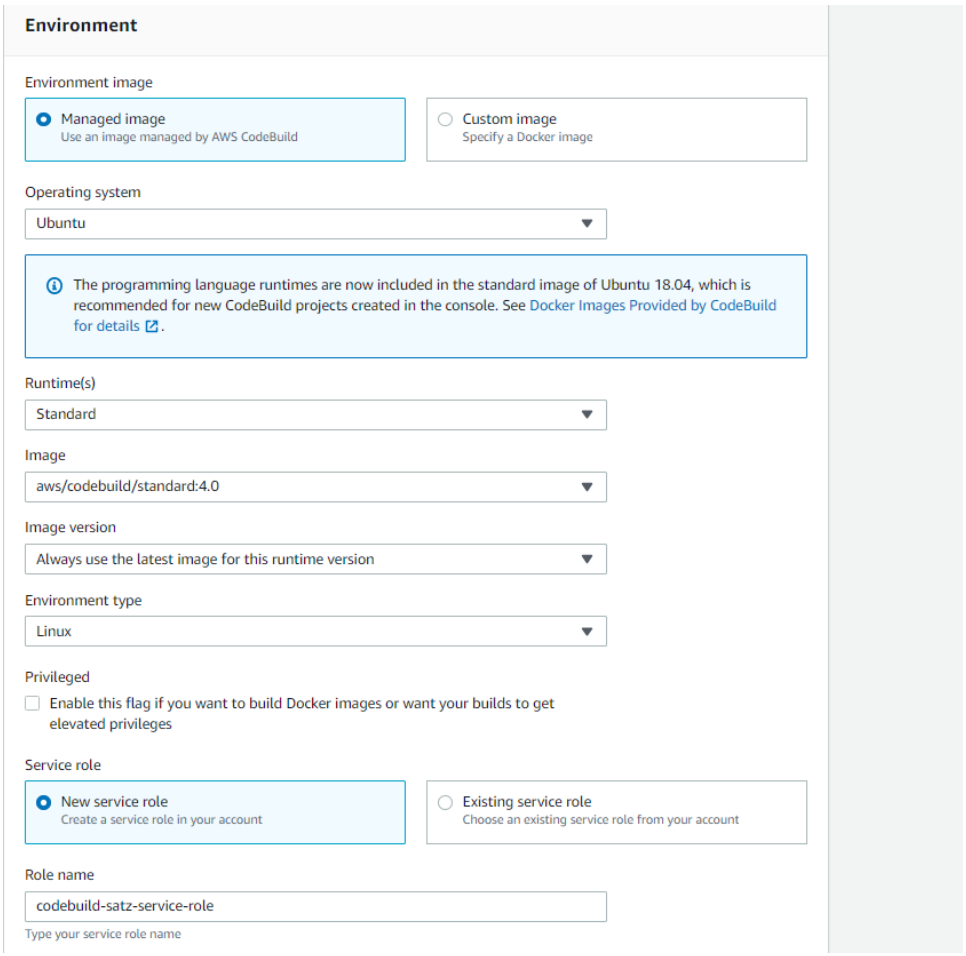
git push origin –all

// To push all the tags,

Git push –tags

16. Create a new build in CodeCommit,


Developer Tools > CodeBuild > Build projects > Create build project
Create build project
Project configuration
Project name
satzbuild
A project name must be 2 to 255 characters. It can include the letters A-Z and a-z, the numbers 0-9, and the special characters - and _.
Description - optional
Build badge - optional
Enable build badge
Enable concurrent build limit - optional
Limit the number of allowed concurrent builds for this project.
Restrict number of concurrent builds this project can start
Additional configuration
tags
Source
Add source
Source 1 - Primary
Source provider
AWS CodeCommit
Repository
vprofile-code-repo
Reference type
Choose the source version reference type that contains your source code.
Branch
Git tag



17. Refer build spec.yml file in directory, that should be added to below buildspec.

Also, S3 needs to be created and mention s3 in below.

Buildspec

Build specifications

- ○ Use a buildspec file
  Store build commands in a YAML-formatted buildspec file
- ◉ Insert build commands
  Store build commands as build project configuration

Build commands

```
  #post_build:
    #commands:
      # - command
      # - command
#reports:
  #report-name-or-arn:
    #files:
      # - location
      # - location
    #base-directory: location
    #discard-paths: yes
    #file-format: JunitXml | CucumberJson
#artifacts:
  #files:
    # - location
    # - location
  #name: $(date +%Y-%m-%d)
  #discard-paths: yes
  #base-directory: location
#cache:
  #paths:
    # - paths
```

Switch to single line

Logs

CloudWatch

☑ CloudWatch logs - *optional*
Checking this option will upload build output logs to CloudWatch.

Group name

satzprofile-cicd-project

Stream name

build logs

S3

☑ S3 logs - *optional*
Checking this option will upload build output logs to S3.

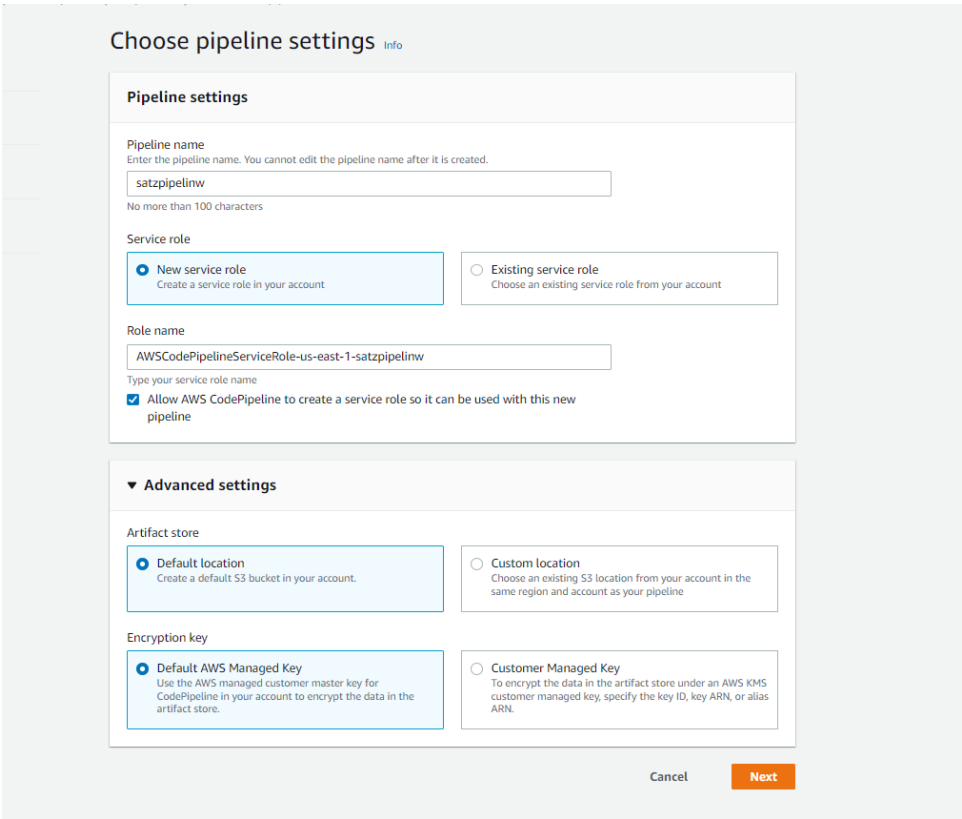
Bucket

🔍

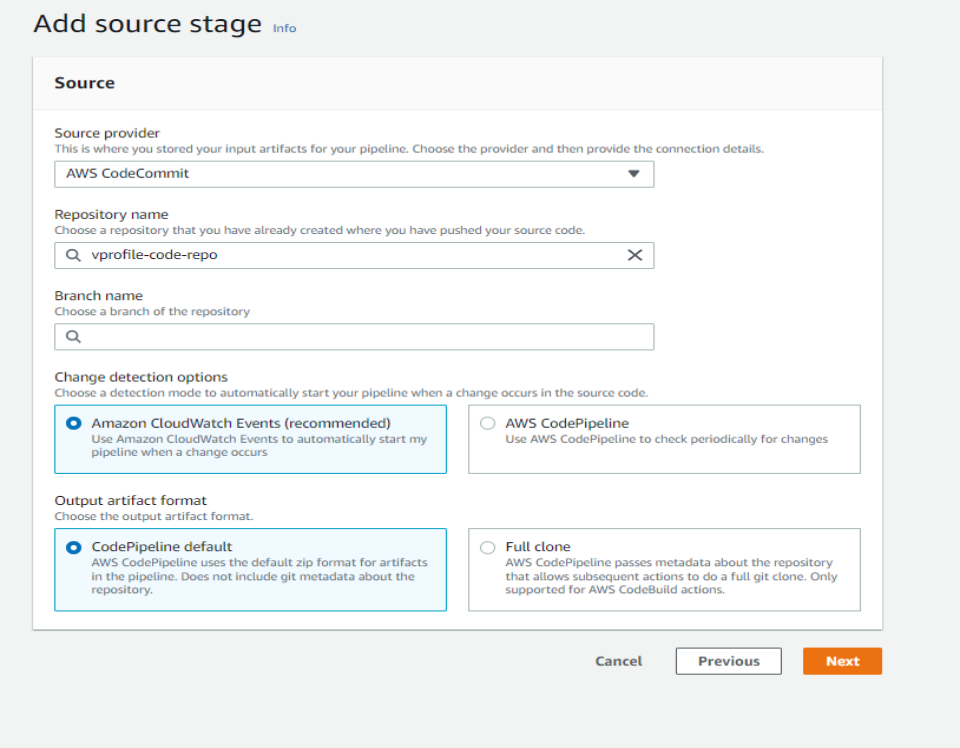
18. Once all are configured, start the build.. Upto this build generation is done.

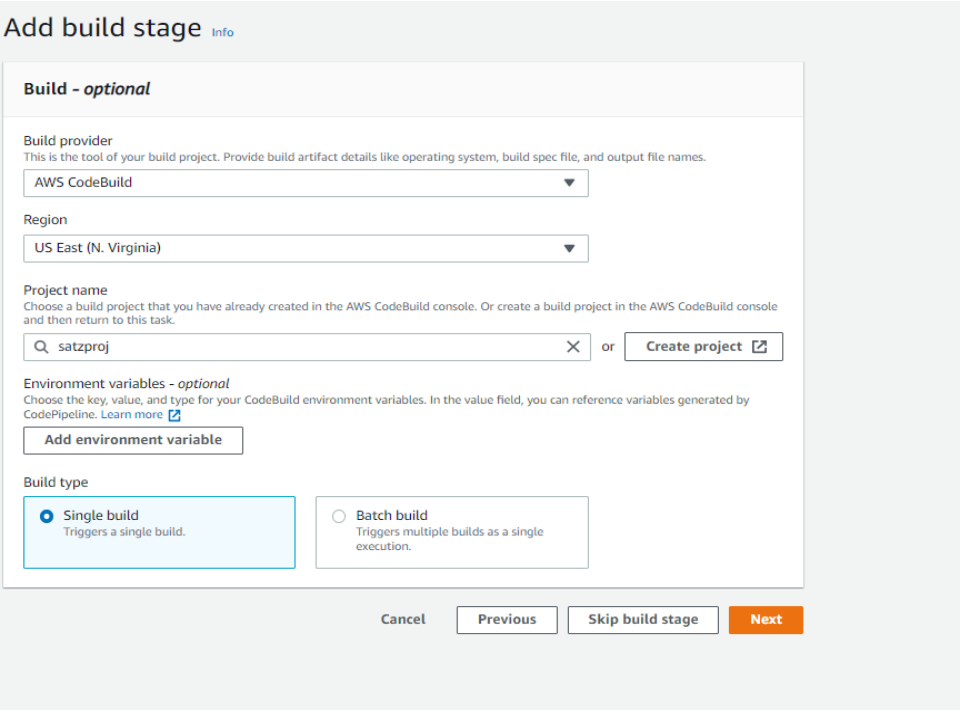
Pipeline,

Create a new pipeline



19. Select the repo, branch name. If any new commits occur this pipeline will get triggered.





Then select the beanstalk environment and create a pipeline.

In between the pipeline, we can add approvals, unit test etc.