CODESTAR-Build,test,deploy the python django application in Ec2.

pre-requires:

please read the complete java-spring application in ec2 CODESTAR-Build,test,deploy the javaspring application in Ec2.

In this documentation most of the steps are similar to above java-spring app documentation, so i'm just giving reference on for some of the steps.

Create a Project(Python-Django app in ec2) in AWS CodeStar:

You use the AWS CodeStar console to create a project. If you use a project template, it sets up the required resources for you. The template also includes sample code that you can use to start coding.

To create a project, sign in to the AWS Management Console with an IAM user that has the `AWSCodeStarFullAccess` policy or equivalent permissions.

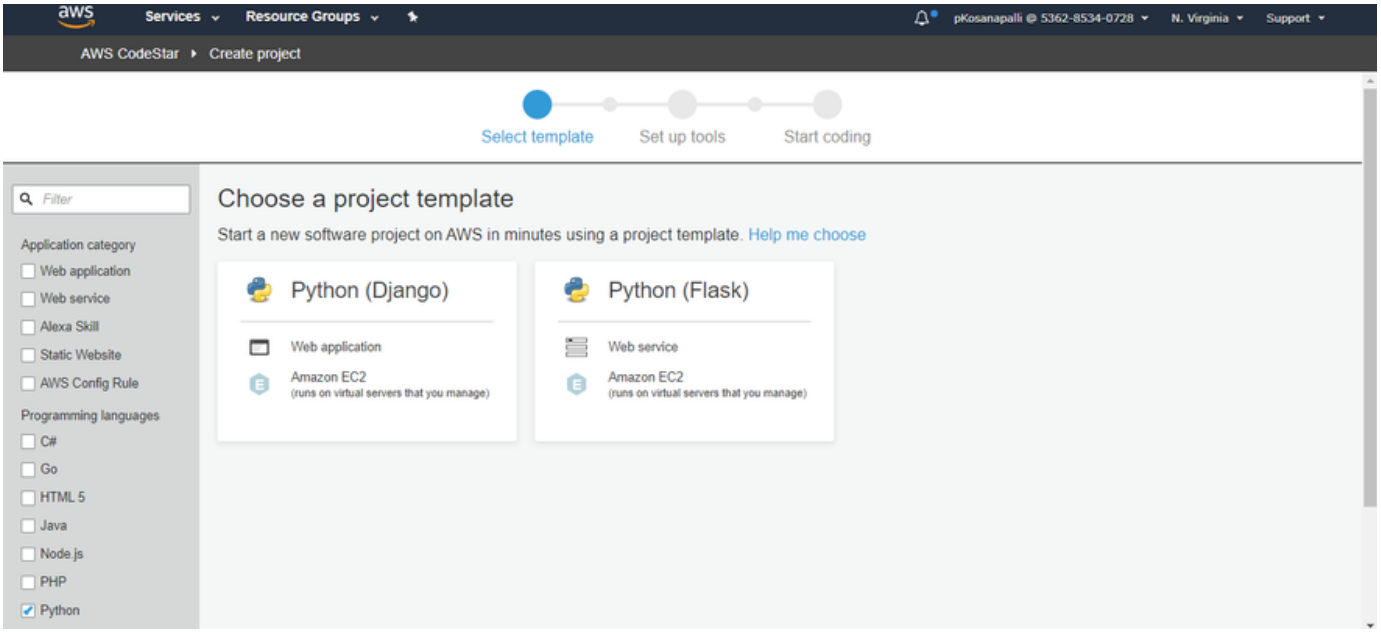
You must complete the steps in How CodeStar - Getting started- how to create project in codestar in AWS before you can complete the procedures in this topic.

Build, test, package& deploy the Python-Django application in ec2:

Use the AWS CodeStar console to create a project.

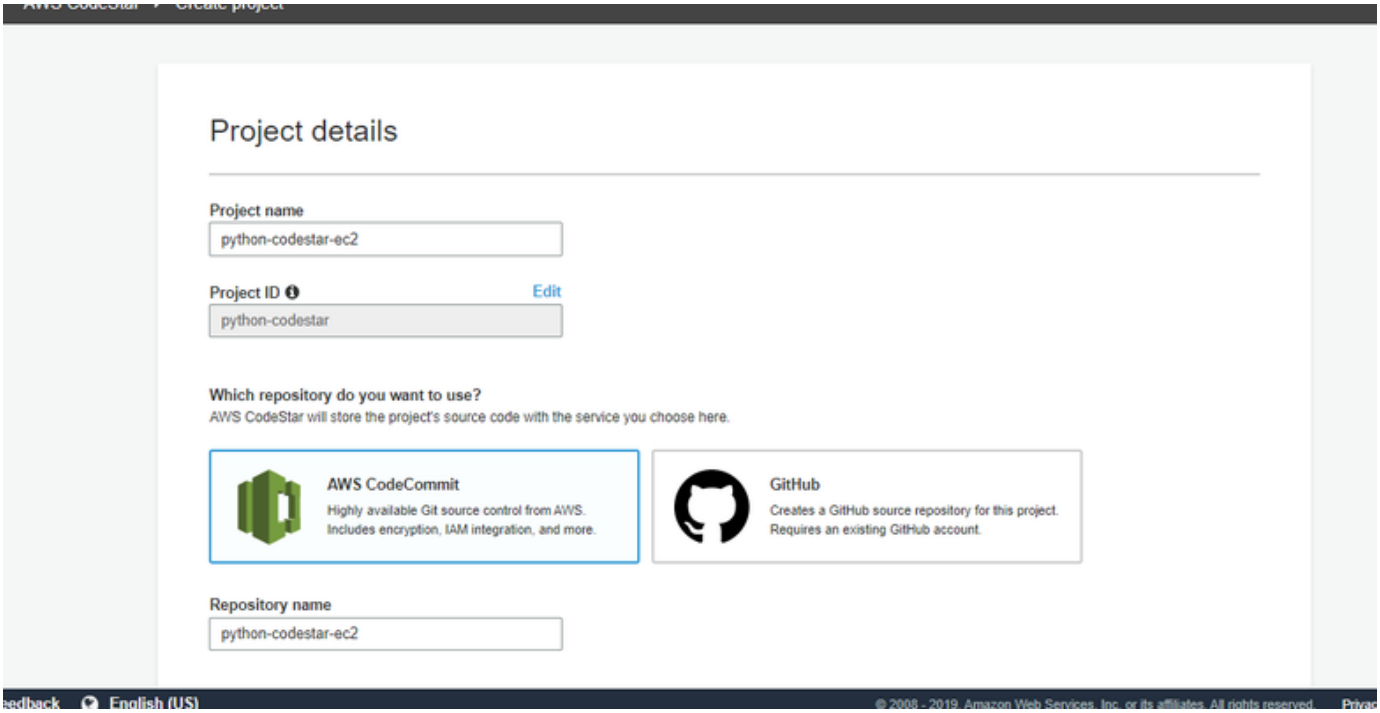
To create a project in AWS CodeStar

- Sign in to the AWS Management Console, and then open the AWS CodeStar console at https://console.aws.amazon.com/codestar/.
- On the **AWS CodeStar** page, choose **Create a new project**. (If you are the first user to create a project, choose **Start a project**.)
- On the **Choose a project template** page, choose the project type from the list of AWS CodeStar project templates. You can use the filter bar to narrow your choices. For example, for a web application project written in java spring to be deployed to Amazon EC2 instances, select the **Web application**, **python** and **Amazon EC2** check boxes. Then choose from the templates available for that set of options.



- In **Project name**, enter a name for the project, such as `My First Project`. The ID for the project is derived from this project name, but is limited to 15 characters.

  For example, the default ID for a project named `My First Project` is `python-codestar-ec2`. This project ID is the basis for the names of all resources associated with the project. AWS CodeStar uses this project ID as part of the URL for your code repository and for the names of related security access roles and policies in IAM. After the project is created, the project ID cannot be changed. To edit the project ID before you create the project, choose **Edit**. Project IDs must be unique for your AWS account in an AWS Region.



- Choose the repository provider, **AWS CodeCommit** or **GitHub**.
- If you chose **AWS CodeCommit**, for **Repository name**, accept the default AWS CodeCommit repository name, or enter a different one.
- If you chose **GitHub**, choose **Connect with GitHub.** go to this documention CODESTAR integration with github .
- Choose **Next**. You can see the project details i.e codecommit, codebuild, code deploy, codepipeline, cloudwatch. these services are default created.

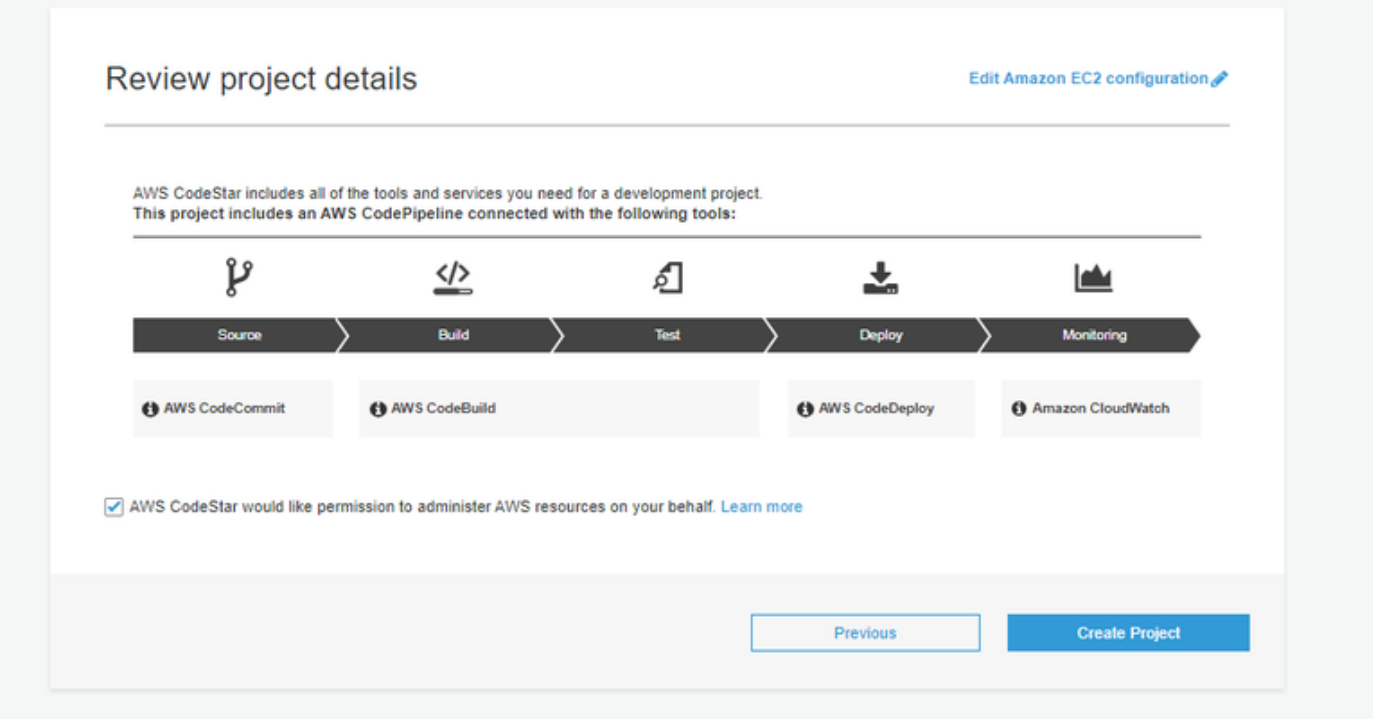
code commit- creates new repository.

codebuild - creates default codebuild service contains source, build environment(docker image,environment,service role),variables, build spec,artifacts, logs.

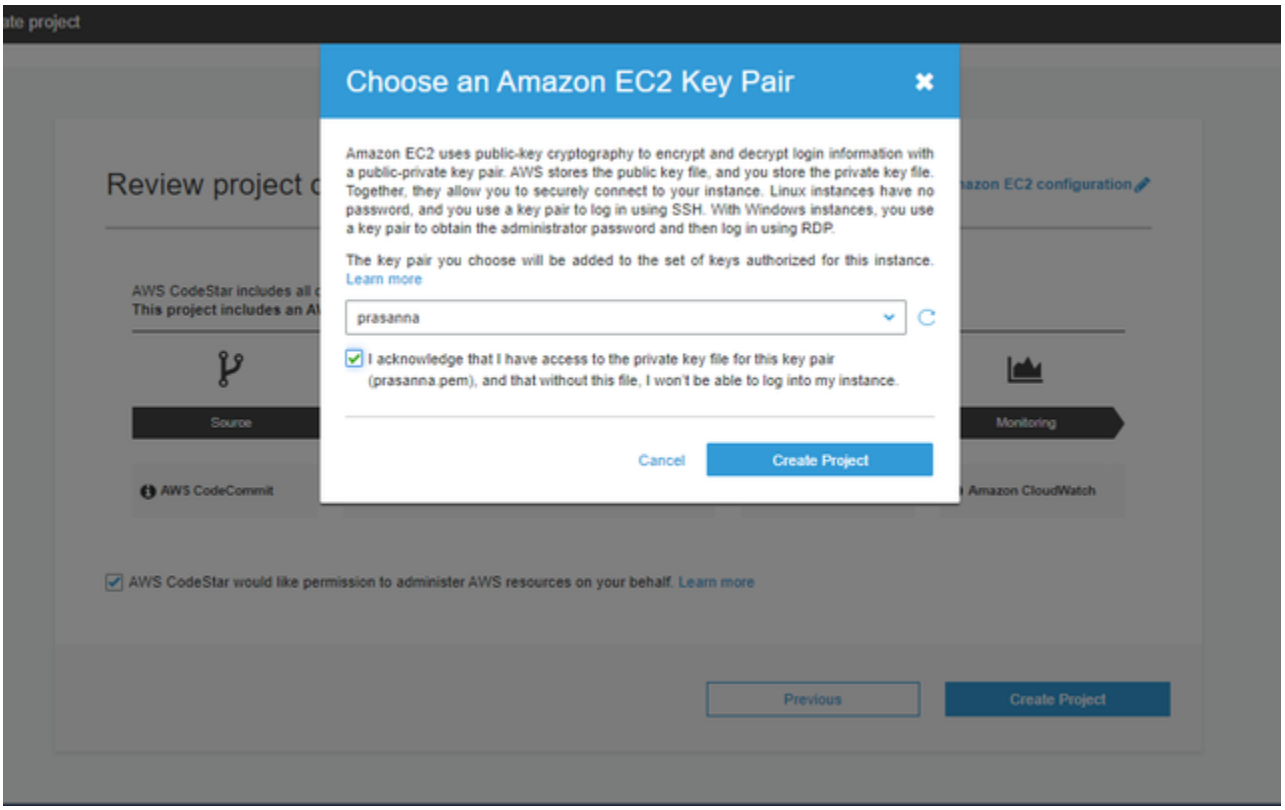
codedeploy - deployments, deployment groups (service role, compute platform.)

codepipeline- source,build,deploy.

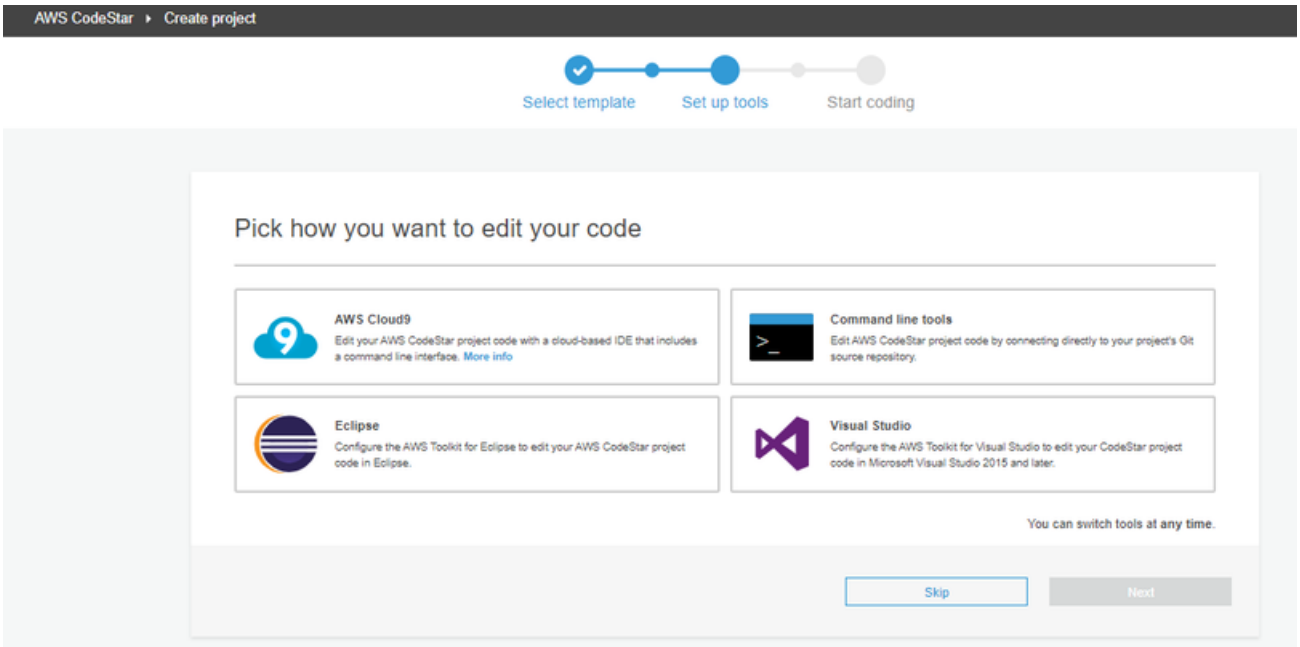
- Review the resources and configuration details.



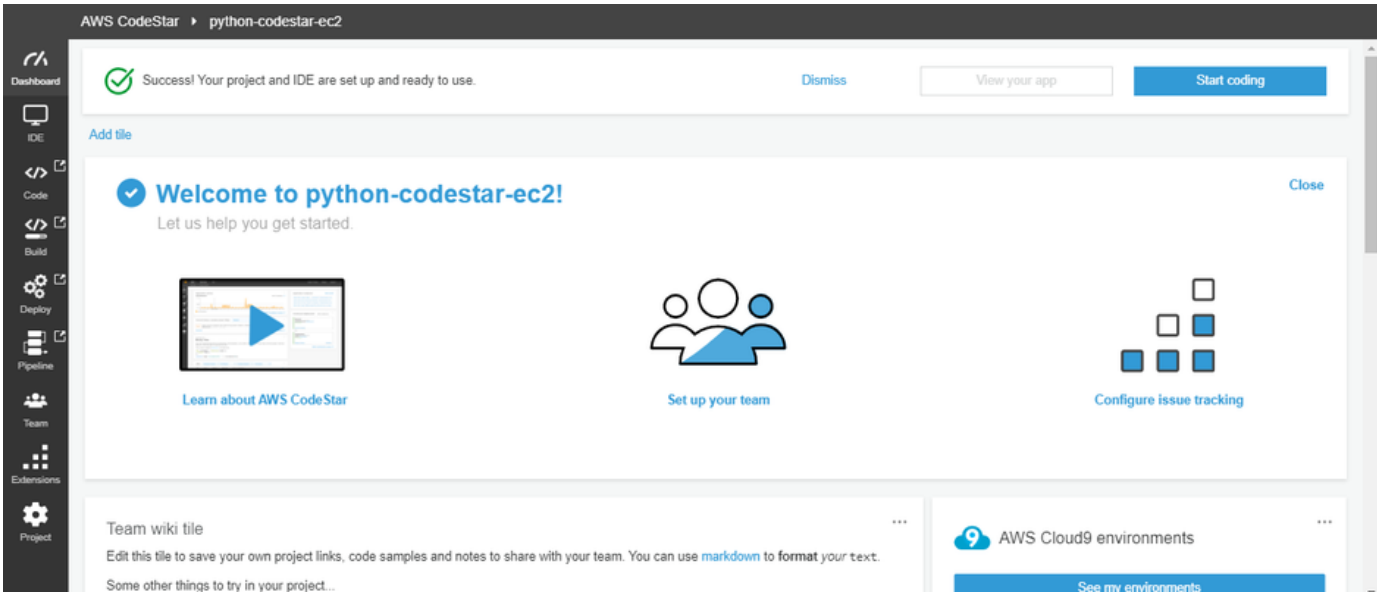
- In **Choose an Amazon EC2 Key Pair**, choose the Amazon EC2 key pair you created in keypairs.
- Select **I acknowledge that I have access to the private key file for this key pair**, and then choose **Create project**.



- It might take a few minutes to create the project (including the repository). After your project has a repository, you can use the **Set up tools** page to configure access to it, or you can choose **Skip** and configure access later.



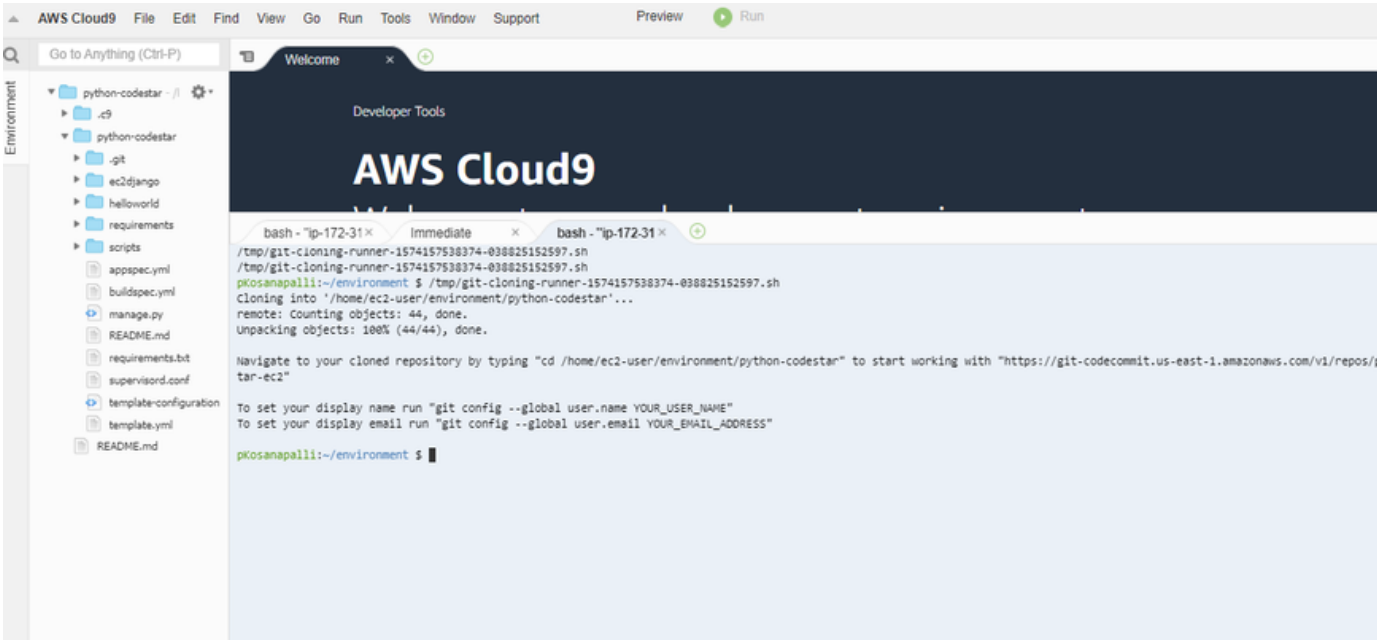
- After your project has been created, you can use the links on the **Welcome** tile to configure other items, such as your user profile in AWS CodeStar.



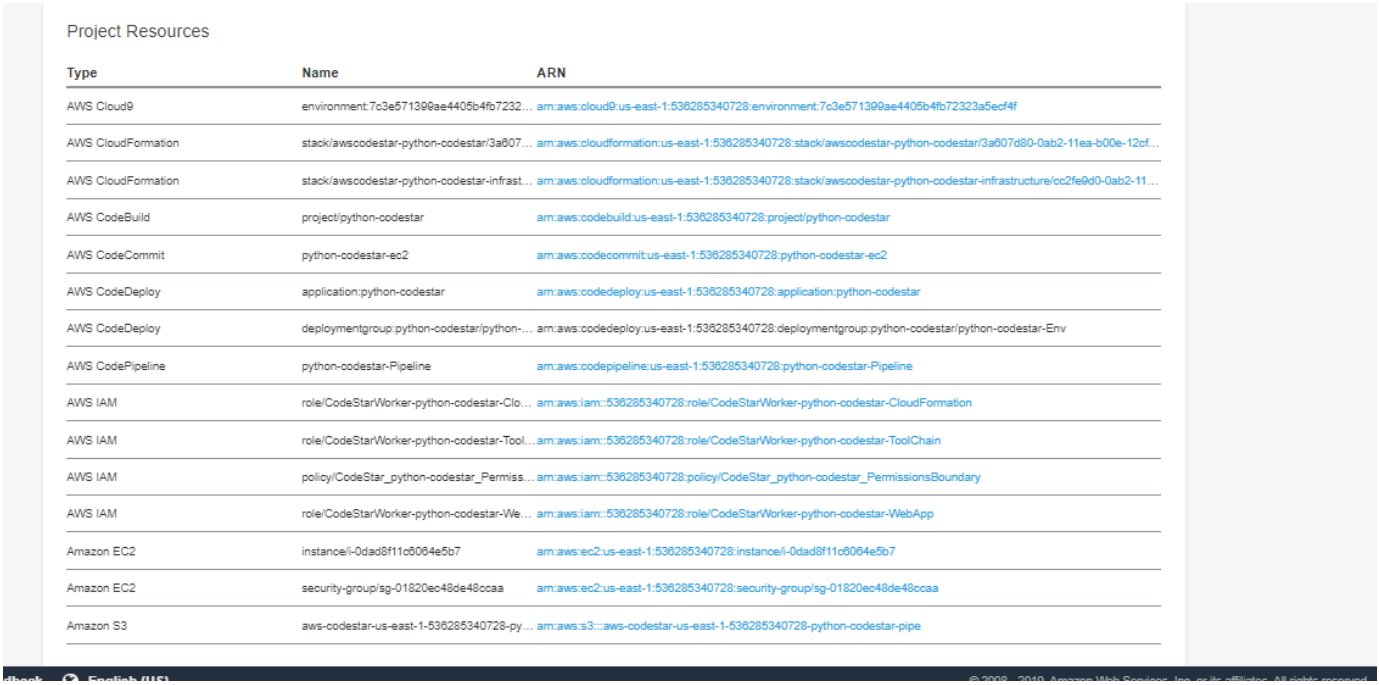
- While your project is being created, you can add team members or configure access to your project repository from the command line or your favorite IDE.
- You can connect your favorite IDE in any one of IDE's AWS cloud9, Eclipse, Vs code, AWS cli., for complete setup with IDE's, see the documentions here Codestar integrate with IDEs- ECLIPSE, CLOUD9, CLI,VS CODE.

Lets see the content of Welcome board of codestar project:

- first you can observe welcome page dashboard of the codestar project, which contains all collaborative services of aws and some of features of codestar.
- On the left side of dashboard you can observe some labels contains IDE, code, build, Deploy,pipeline, team,extensions, project see the below pics. more complete details refer the java-spring app. here im showing only python-django app in ec2.
- when ever you click the IDE, you can go to new page, that configure the favorite IDE setup.



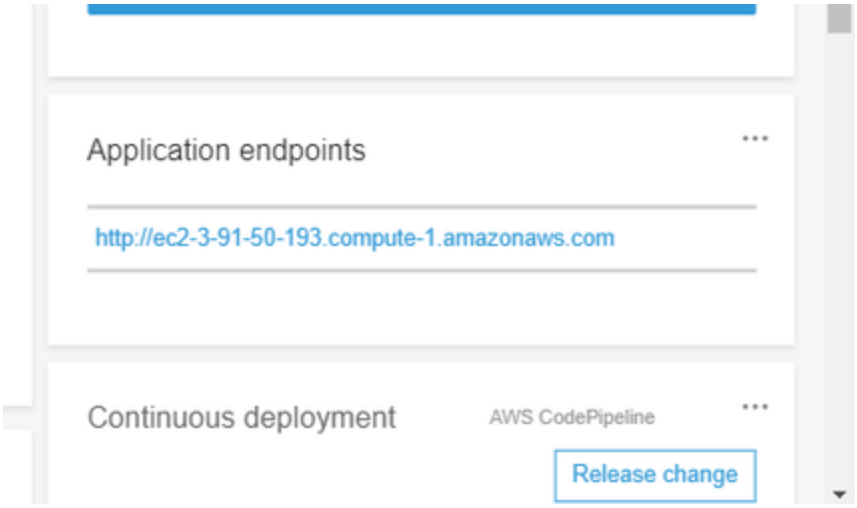
- On the left side of dashboard you can observe some labels name code, build, deploy you click the code, you can go to new page, codestar is configure with new repo name python-codestar-ec2 and build,deploy also.
- this are the resources using for codestarproject. All resource under code-star (s3, codepipeline,ec2) is configured with cloud-formation (stored under secret) for complete background of architecture of CODESTAR is documented separately- CODESTAR- background architecture.
- you can see the corresponding resource dashboard by clicking the corresponding urls in the pics.



Project Resources

Type	Name	ARN
AWS Cloud9	environment:7c3e571399ae4405b4fb7232...	arn:aws:cloud9:us-east-1:536285340728:environment:7c3e571399ae4405b4fb72323a5ecf4f
AWS CloudFormation	stack/awscodestar-python-codestar/3a607...	arn:aws:cloudformation:us-east-1:536285340728:stack/awscodestar-python-codestar/3a607d80-0ab2-11ea-b00e-12cf...
AWS CloudFormation	stack/awscodestar-python-codestar-infrast...	arn:aws:cloudformation:us-east-1:536285340728:stack/awscodestar-python-codestar-infrastructure/cc2fe9d0-0ab2-11...
AWS CodeBuild	project/python-codestar	arn:aws:codebuild:us-east-1:536285340728:project/python-codestar
AWS CodeCommit	python-codestar-ec2	arn:aws:codecommit:us-east-1:536285340728:python-codestar-ec2
AWS CodeDeploy	application:python-codestar	arn:aws:codedeploy:us-east-1:536285340728:application:python-codestar
AWS CodeDeploy	deploymentgroup:python-codestar/python-...	arn:aws:codedeploy:us-east-1:536285340728:deploymentgroup:python-codestar/python-codestar-Env
AWS CodePipeline	python-codestar-Pipeline	arn:aws:codepipeline:us-east-1:536285340728:python-codestar-Pipeline
AWS IAM	role/CodeStarWorker-python-codestar-Clo...	arn:aws:iam::536285340728:role/CodeStarWorker-python-codestar-CloudFormation
AWS IAM	role/CodeStarWorker-python-codestar-Tool...	arn:aws:iam::536285340728:role/CodeStarWorker-python-codestar-ToolChain
AWS IAM	policy/CodeStar_python-codestar_Permiss...	arn:aws:iam::536285340728:policy/CodeStar_python-codestar_PermissionsBoundary
AWS IAM	role/CodeStarWorker-python-codestar-We...	arn:aws:iam::536285340728:role/CodeStarWorker-python-codestar-WebApp
Amazon EC2	instance/i-0dad8f11c6064e5b7	arn:aws:ec2:us-east-1:536285340728:instance/i-0dad8f11c6064e5b7
Amazon EC2	security-group/sg-01820ec48de48ccaa	arn:aws:ec2:us-east-1:536285340728:security-group/sg-01820ec48de48ccaa
Amazon S3	aws-codestar-us-east-1-536285340728-py...	arn:aws:s3:::aws-codestar-us-east-1-536285340728-python-codestar-pipe

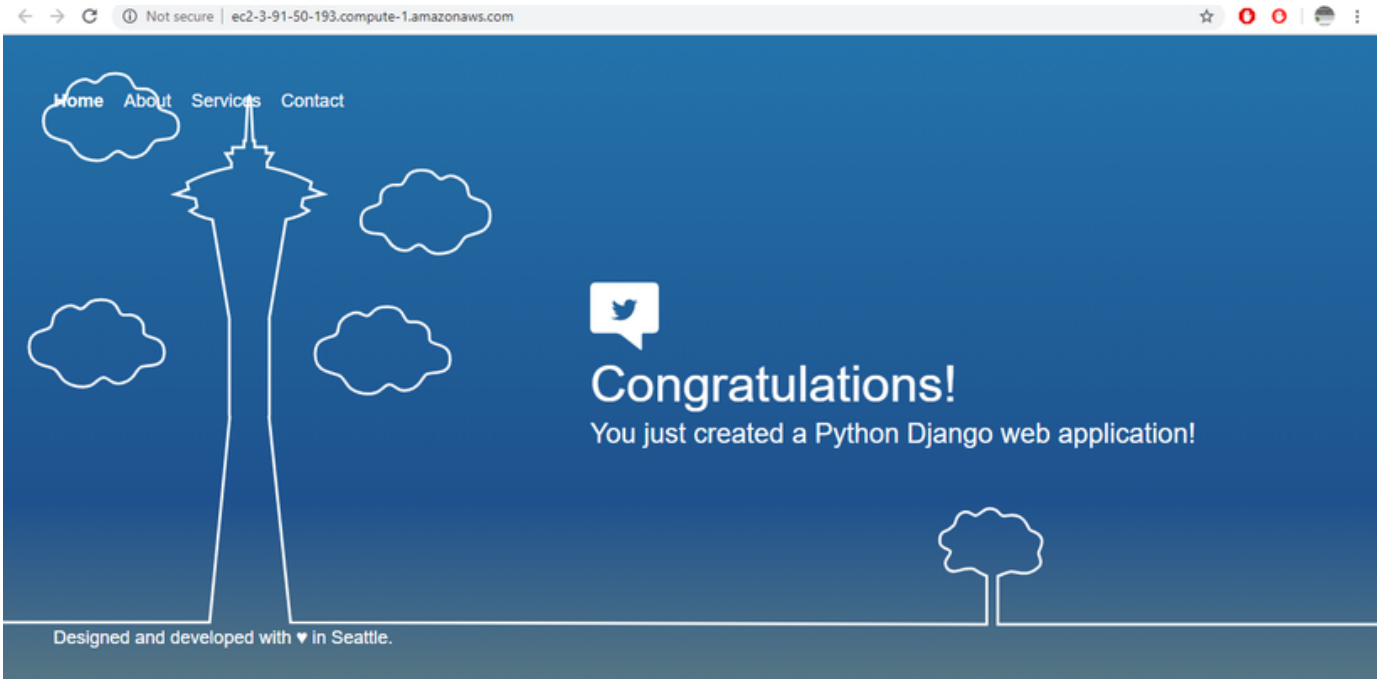
- Application endpoints- which are the configured for application output dashboard you can see the changes here by clicking the URL.



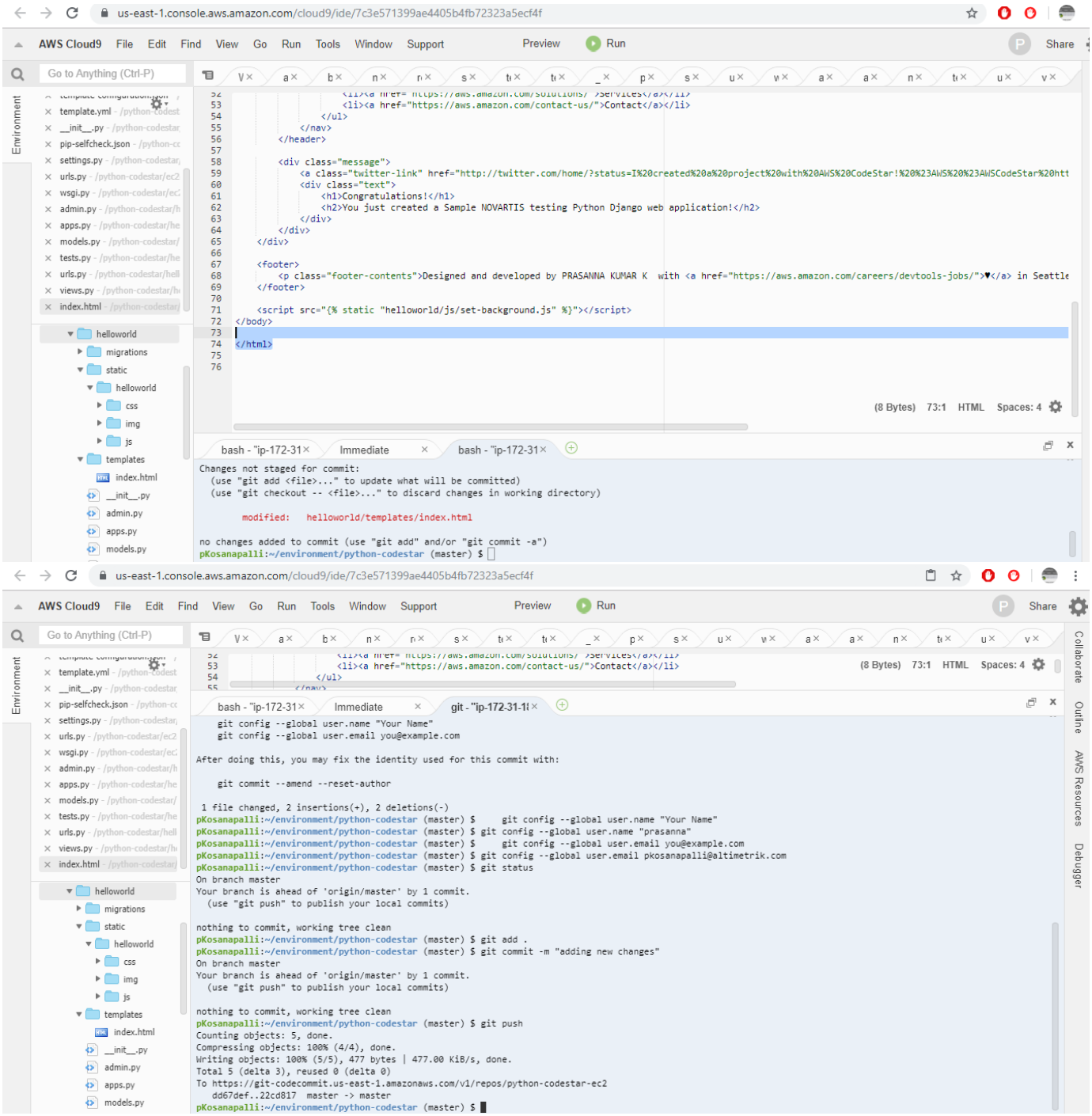
- Application activity - which can be monitor by the AWS cloudwatch for the application, click the cloudwatch details for more details.

Build, test, package& deploy the Python-Django application in ec2:

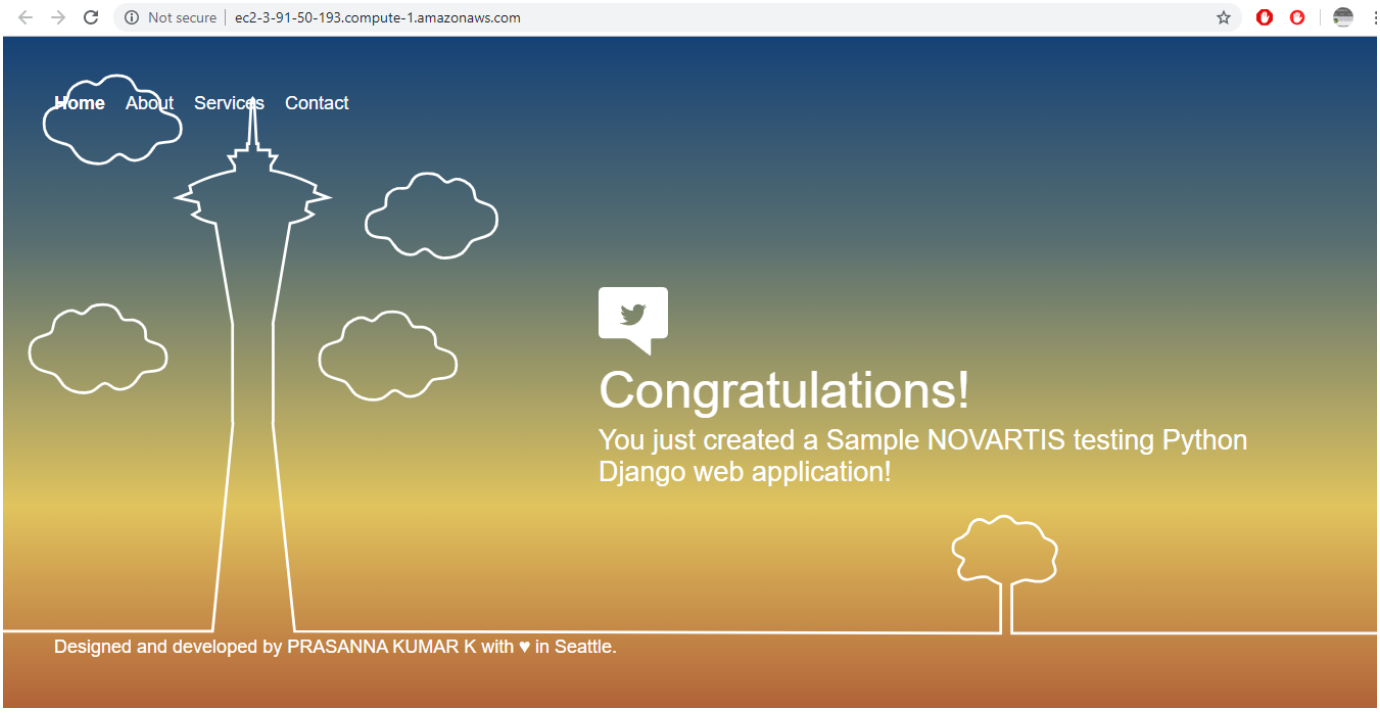
- Now we have created a python-django project in codestar.
- Go to project explore click the endpoint(url of application), redirect to new url of your application, this url can be configure based on our requirement in the DNS service.
- you can able to see the new python-django web application.



- Now go to the Cloud9 editor and modify the required changes and pull the changes to codestar. for more go to Codestar integrate with IDEs- ECLIPSE, CLOUD9, CLI,VS CODE.



- Now a new release is triggered in the code pipeline with latest changes. We have configured our code pipeline like this. Go to pipeline and check this.
- Go to the Url of the application, see the changes we made.



- Now we have done the build, test, deploy the python-django application.

How we build,test,deploy the Python-Django application in EC2:

- First lets see whats things in code commit(SCM), go to code commit & see the repo.

- lets see what it is

```
This sample code helps how simple Django web application deployed by AWS
CodeDeploy and AWS CloudFormation to an Amazon EC2 server.
This sample includes:
README.md - this file
appspec.yml - this file is used by AWS CodeDeploy when deploying the web
application to EC2
buildspec.yml - this file is used by AWS CodeBuild to build and test
your application
requirements/ - this directory contains requirements files that describe
the Python dependencies required for your Django application in
different environments
requirements.txt - this file is used to install production Python
dependencies
ec2django/ - this directory contains your Django project files. Note
that this directory contains a Django config file (settings.py) that
includes a pre-defined SECRET_KEY. Before running in a production
environment, you should replace this application key with one you
generate (see
https://docs.djangoproject.com/en/1.11/howto/deployment/checklist/#secre
t-key for details)
helloworld/ - this directory contains your Django application files
manage.py - this Python script is used to start your Django web
application
scripts/ - this directory contains scripts used by AWS CodeDeploy when
installing and deploying your application on the Amazon EC2 instance
supervisord.conf - this configuration file is used by Supervisor to
control your web application on the Amazon EC2 instance
template.yml - this file contains the description of AWS resources used
by AWS CloudFormation to deploy your infrastructure
template-configuration.json - this file contains the project ARN with
placeholders used for tagging resources with the project ID
```

- lets see the code individually etc.

```
{
  "Tags":
  {

"awscodestar:projectArn":"arn:$PARTITION$:codestar:$AWS_REGION$:$ACCOUNT
_ID$:project/$PROJECT_ID$"
  }
}
all resources under this codestar project will be created with this
naming configuration.
```

Code deploy-infrastructure automation with cloud formation(ec2& tomcat,java):-

- Once after selecting the application type and deployment, we need to configure the required resources for codestar project i.e s3, codepipeline, ec2 instance.For that purpose we need to create cloud formation stack for creating the ec2 instance and configurations,

```
this is cloud formation stack for creating the ec2 instance with
specified default parameters and security group also.
AWSTemplateFormatVersion: 2010-09-09
Conditions:
  UseSubnet: !Not [!Equals [!Ref 'SubnetId', subnet-none]]
  IsBurstableInstanceType: !Equals [!Select [0, !Split ['.', !Ref
InstanceType]], t2]
Transform:
- AWS::CodeStar
Parameters:
  ProjectId:
    Type: String
    Description: AWS CodeStar project ID used to name project resources
and create roles.
  InstanceType:
    Type: String
    Description: The type of Amazon EC2 Linux instances that will be
launched for this project.
  WebAppInstanceProfile:
     Type: String
     Description: The IAM role that will be created for the Amazon EC2
Linux instances.
  ImageId:
    Type: String
    Description: The Amazon EC2 Linux instance Amazon Machine Image
(AMI), which designates the configuration of the new instance.
  KeyPairName:
    Type: String
    Description: The name of an existing Amazon EC2 key pair in the
region where the project is created, which you can use to SSH into the
new Amazon EC2 Linux instances.
  VpcId:
    Type: String
    Description: The ID of the Amazon Virtual Private Cloud (VPC) to use
for Amazon EC2 instances.
  SubnetId:
    Type: String
    Description: The name of the VPC subnet to use for Amazon EC2
instances launched for this project.
  Stage:
    Type: String
    Description: The name for a project pipeline stage, such as Staging
or Prod, for which resources are provisioned and deployed.
    Default: ''
Resources:
  WebApp01:
```

```
    Description: The installation and configuration commands this
project will use to create instances that support this sample web
application.
    Properties:
      CreditSpecification:
        CPUCredits: !If [IsBurstableInstanceType, unlimited, !Ref
'AWS::NoValue']
      IamInstanceProfile: !Ref 'WebAppInstanceProfile'
      ImageId: !Ref 'ImageId'
      InstanceType: !Ref 'InstanceType'
      KeyName: !Ref 'KeyPairName'
      NetworkInterfaces:
      - AssociatePublicIpAddress: true
        DeviceIndex: 0
        GroupSet:
        - !Ref 'WebAppSG'
        SubnetId: !If
        - UseSubnet
        - !Ref 'SubnetId'
        - !Ref 'AWS::NoValue'
      Tags:
      - Key: Environment
        Value: !Sub '${ProjectId}-WebApp${Stage}'
      - Key: Name
        Value: !Sub '${ProjectId}-WebApp${Stage}'
      UserData:
        Fn::Base64:
          Fn::Sub: |
            #!/bin/bash -ex
            # Install the AWS CodeDeploy Agent.
            cd /home/ec2-user/
            wget
https://aws-codedeploy-${AWS::Region}.s3.amazonaws.com/latest/codedeploy
-agent.noarch.rpm
            yum -y install codedeploy-agent.noarch.rpm
            # Install the Amazon CloudWatch Logs Agent.
            wget
https://s3.amazonaws.com/aws-cloudwatch/downloads/latest/awslogs-agent-s
etup.py
            wget
https://s3.amazonaws.com/aws-codedeploy-us-east-1/cloudwatch/codedeploy_
logs.conf
            wget
https://s3.amazonaws.com/aws-codedeploy-us-east-1/cloudwatch/awslogs.con
f
            chmod +x ./awslogs-agent-setup.py
            python awslogs-agent-setup.py -n -r ${AWS::Region} -c
./awslogs.conf
            mkdir -p /var/awslogs/etc/config
            cp codedeploy_logs.conf /var/awslogs/etc/config/
```

```
            service awslogs restart
    Type: AWS::EC2::Instance
  WebAppSG:
    Description: The default Amazon EC2 security group that will be
created for the Amazon EC2 Linux instances.
    Type: AWS::EC2::SecurityGroup
    Properties:
      GroupDescription: Enable HTTP access via port 80 and SSH access
via port 22.
      SecurityGroupIngress:
      - IpProtocol: tcp
        FromPort: '80'
        ToPort: '80'
        CidrIp: 0.0.0.0/0
      - IpProtocol: tcp
        FromPort: '22'
        ToPort: '22'
```

```
          CidrIp: 0.0.0.0/0
        VpcId: !Ref 'VpcId'
```

- Now we have Ec2 instance for deployment, now we need to configure the instance for django app deployment . After instance created, this cloudformation stack created and this script runs includes(shell scripts).

```
version: 0.0
os: linux
files: (copy file from agent)
 - source: /ec2django/
   destination: /home/ec2-user/ec2django
 - source: /helloworld/
   destination: /home/ec2-user/helloworld
 - source: /manage.py
   destination: /home/ec2-user
 - source: /supervisord.conf
   destination: /home/ec2-user
 - source: /requirements.txt
   destination: /home/ec2-user
 - source: /requirements/
   destination: /home/ec2-user/requirements

permissions:
  - object: /home/ec2-user/manage.py
    owner: ec2-user
    mode: 644
    type:
      - file
  - object: /home/ec2-user/supervisord.conf
    owner: ec2-user
    mode: 644
    type:
      - file
hooks:
  AfterInstall:
    - location: scripts/install_dependencies
      timeout: 300
      runas: root
    - location: scripts/codestar_remote_access
      timeout: 300
      runas: root
    - location: scripts/start_server
      timeout: 300
      runas: root

  ApplicationStop:
    - location: scripts/stop_server
      timeout: 300
      runas: root
```

- The scripts are below install priority wise.

/scripts/codestar_remoteaccess	**/scripts/install_dependencies**	/scripts/start_server

#!/bin/bash # Install AWS CodeStar remote access management. # Allows project members to remotely access Amazon EC2 instances running Linux associated with the project. wget -O /usr/local/bin/get_authorized_keys https://awscodestar-templates-common.s3.amazonaws.com/us-east-1/get_authorized_keys chmod 755 /usr/local/bin/get_authorized_keys sed -i '/AuthorizedKeysCommand /s/.*/AuthorizedKeysCommand \/usr\/local\/bin\/get_authorized_keys/g' /etc/ssh/sshd_config sed -i '/AuthorizedKeysCommandUser /s/.*/AuthorizedKeysCommandUser root/g' /etc/ssh/sshd_config /etc/init.d/sshd restart yum update -y aws-cfn-bootstrap yum install -y aws-cli # Install pip and python dev libraries. yum install -y python27-devel python27-pip gcc pip install boto3 pip install pycryptodome	#!/bin/bash yum update easy_install pip pip install supervisor==3.3.4 yum -y install python36 python36-virtualenv python36-pip cd /home/ec2-user virtualenv-3.6 environment source environment/bin/activate pip install -r requirements.txt	cd /home/ec2-user source environment/bin/activate python manage.py collectstatic --noinput /usr/local/bin/supervisord -c /home/ec2-user/supervisord.con

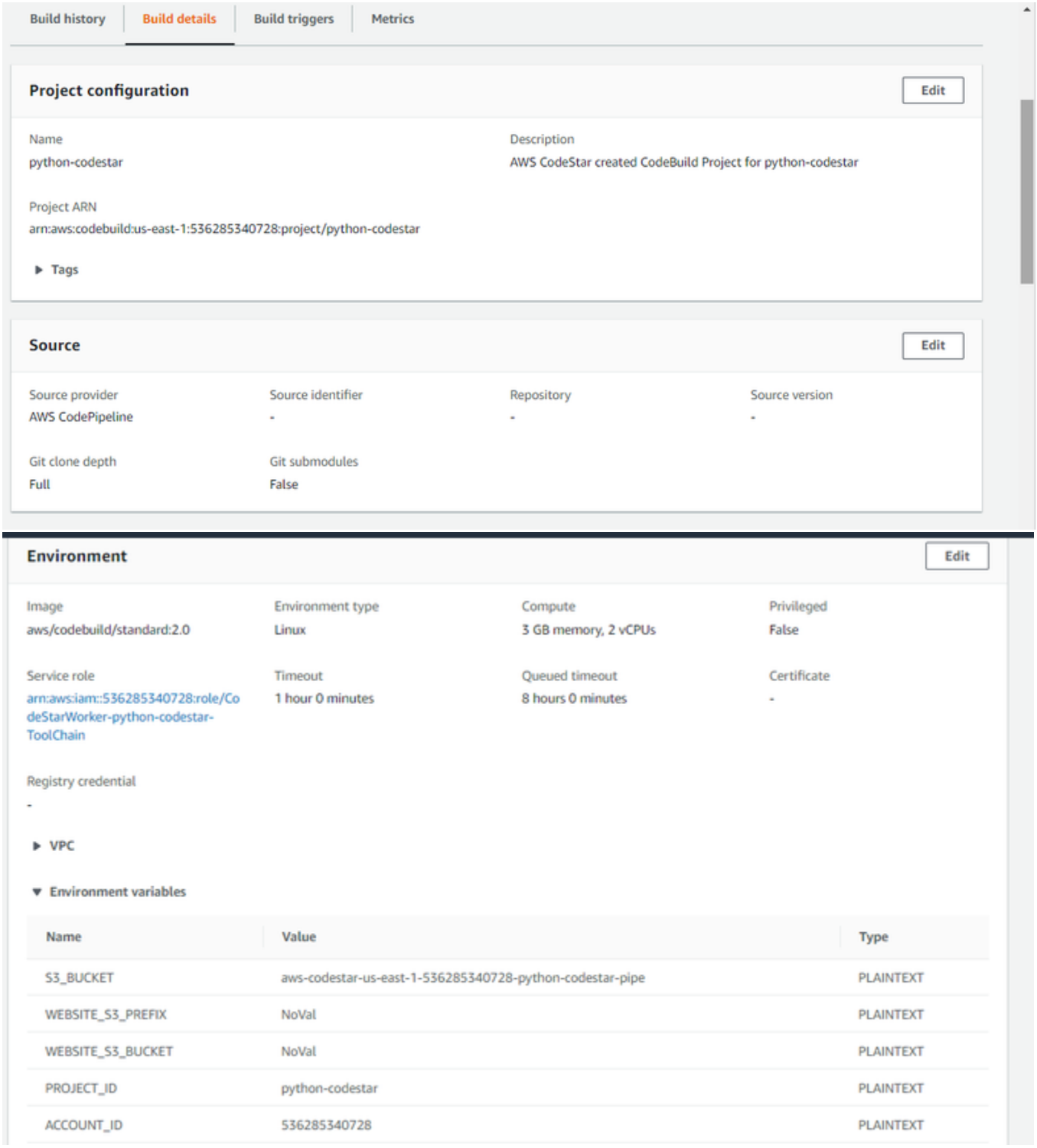
- Till now we have done deployment configurations in ec2 instance. What about build part, let see codebuild.

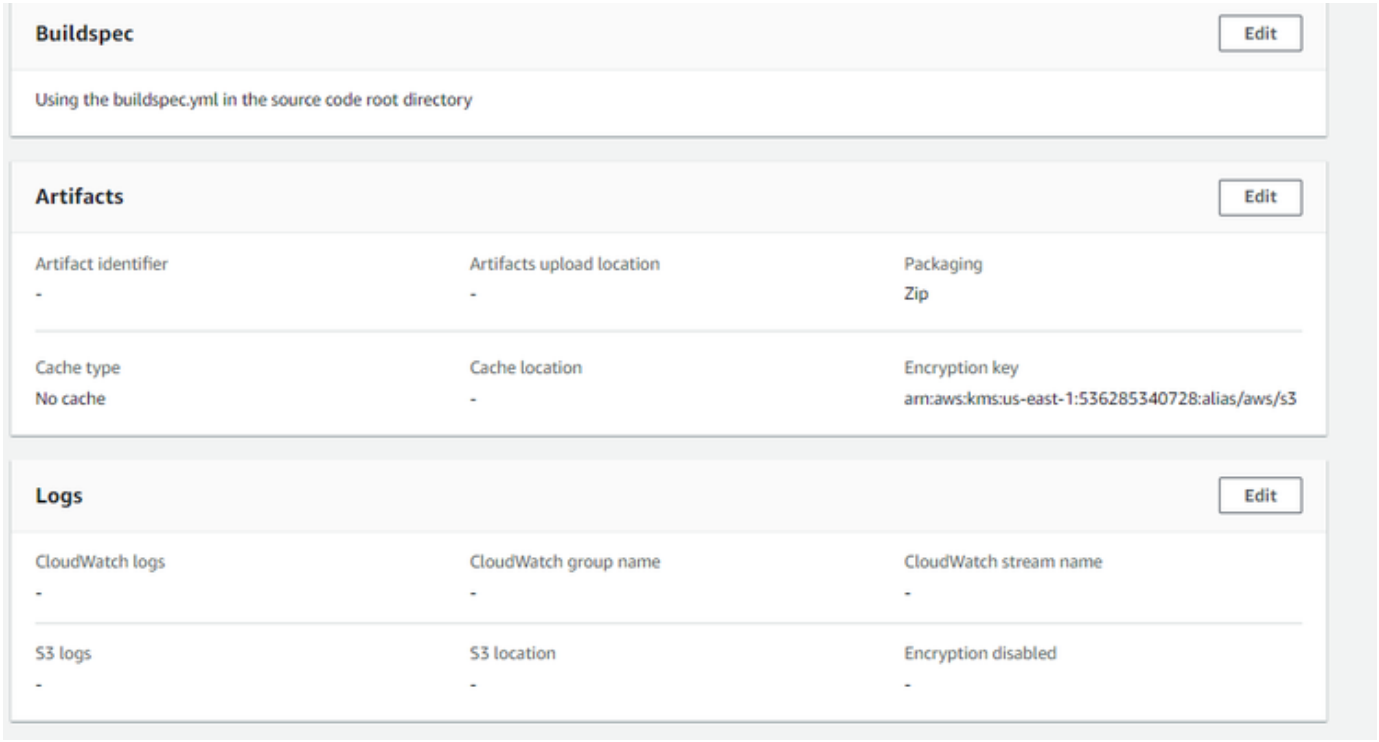
Code-build - build the python-django app specifications:-

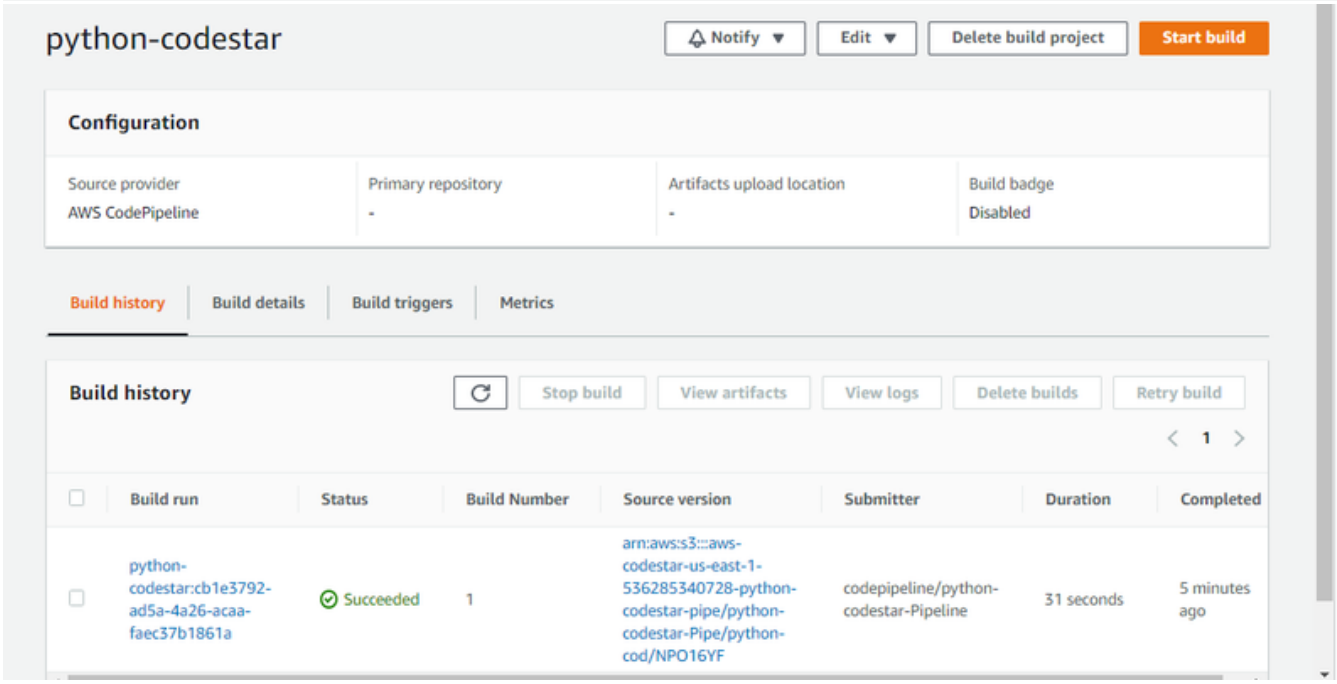
- Code-build- the code will be build in containers remote agent and store the artifacts in s3 instance. So lets see the specifications in code build.
- project configurations- related to name,Project ARN, description,tags.
- Source- its taken default code pipeline which is reflected by code-star.

```
version: 0.2
phases:
  install:
    runtime-versions:
      python: 3.7
    commands:
      # Install dependencies needed for running tests
      - pip install -r requirements/common.txt
  pre_build:
    commands:
      # Discover and run unit tests. For more information, see
<https://docs.djangoproject.com/en/2.0/topics/testing/overview/>
      - python manage.py test
  post_build:
    commands:
      # Do not remove this statement. This command is required for AWS
CodeStar projects.
      # Update the AWS Partition, AWS Region, account ID and project ID
in the project ARN in template-configuration.json file so AWS
CloudFormation can tag project resources.
      - sed -i.bak
's/\$PARTITION\$/'${PARTITION}'/g;s/\$AWS_REGION\$/'${AWS_REGION}'/g;s/\
$ACCOUNT_ID\$/'${ACCOUNT_ID}'/g;s/\$PROJECT_ID\$/'${PROJECT_ID}'/g'
template-configuration.json
artifacts:
  type: zip
  files:
    - 'template.yml'
    - 'ec2django/**/*'
    - 'helloworld/**/*'
    - 'scripts/**/*'
    - 'appspec.yml'
    - 'manage.py'
    - 'requirements/**/*'
    - 'requirements.txt'
    - 'supervisord.conf'
    - 'template-configuration.json'
```

- build environment, this python-django application will be build in a container(remote) with specification image, environment, compute etc.
- build spec- this will allow you to build and archive the application.
- Artifacts- this will store the archived files in the s3 bucket.
- logs - Cloudwatch is configure for monitoring the application.
- build steps is mentioned in the builspec.yml.
- This application is build with python command and give the command to run django server application
- you can able to see the build details and builds history.
- Artifacts will be store in a S3 bucket configure by cloud formation see the codestar project.


Build history
Build details
Build triggers
Metrics
Project configuration
Edit
Name
python-codestar
Description
AWS CodeStar created CodeBuild Project for python-codestar
Project ARN
arn:aws:codebuild:us-east-1:536285340728:project/python-codestar
Tags
Source
Edit
Source provider
AWS CodePipeline
Source identifier
-
Repository
-
Source version
-
Git clone depth
Full
Git submodules
False
Environment
Edit
Image
aws/codebuild/standard:2.0
Environment type
Linux
Compute
3 GB memory, 2 vCPUs
Privileged
False
Service role
arn:aws:iam::536285340728:role/CodeStarWorker-python-codestar-ToolChain
Timeout
1 hour 0 minutes
Queued timeout
8 hours 0 minutes
Certificate
-
Registry credential
-
VPC
Environment variables
Name
Value
Type
S3_BUCKET
aws-codestar-us-east-1-536285340728-python-codestar-pipe
PLAINTEXT
WEBSITE_S3_PREFIX
NoVal
PLAINTEXT
WEBSITE_S3_BUCKET
NoVal
PLAINTEXT
PROJECT_ID
python-codestar
PLAINTEXT
ACCOUNT_ID
536285340728
PLAINTEXT

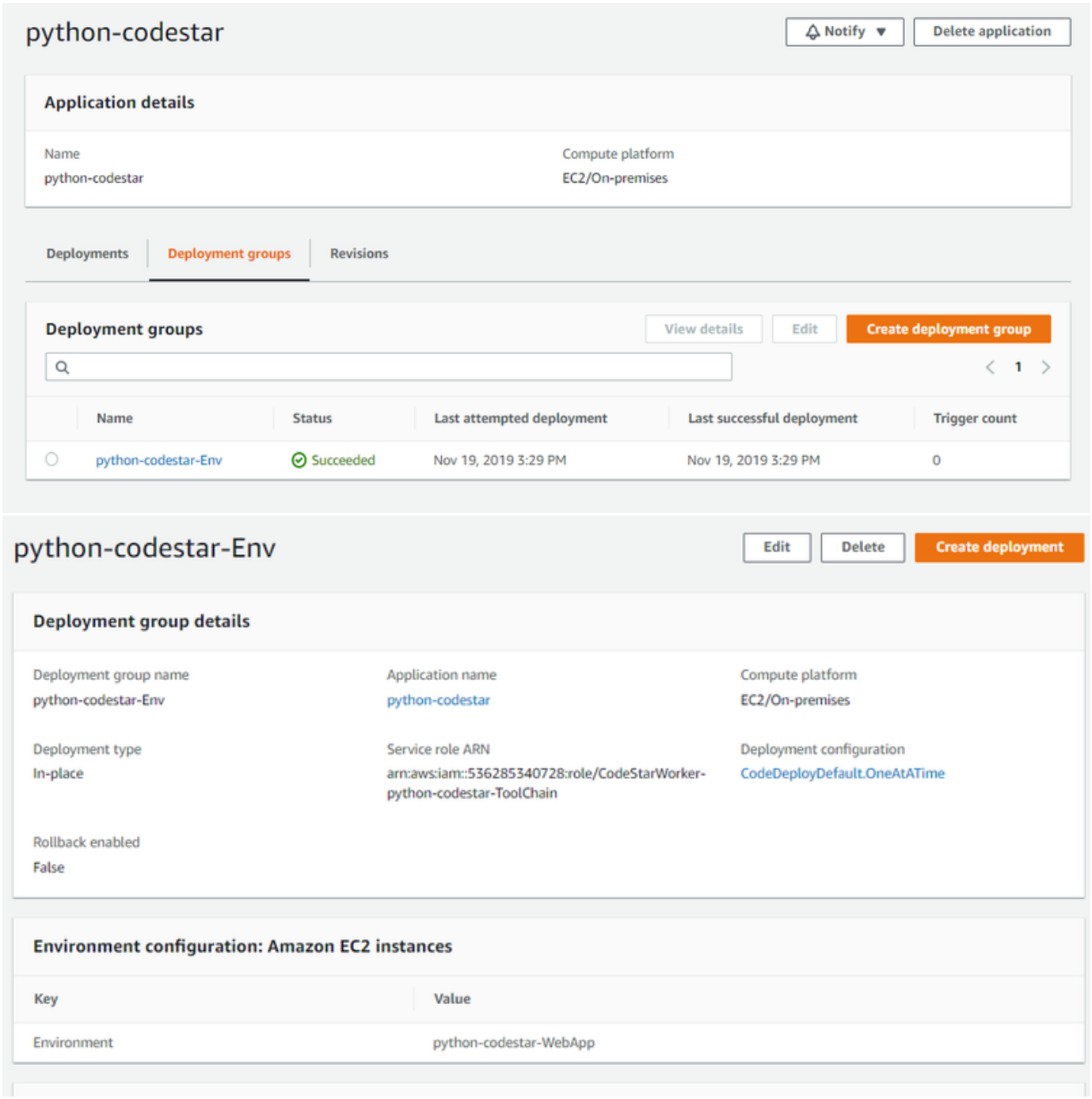


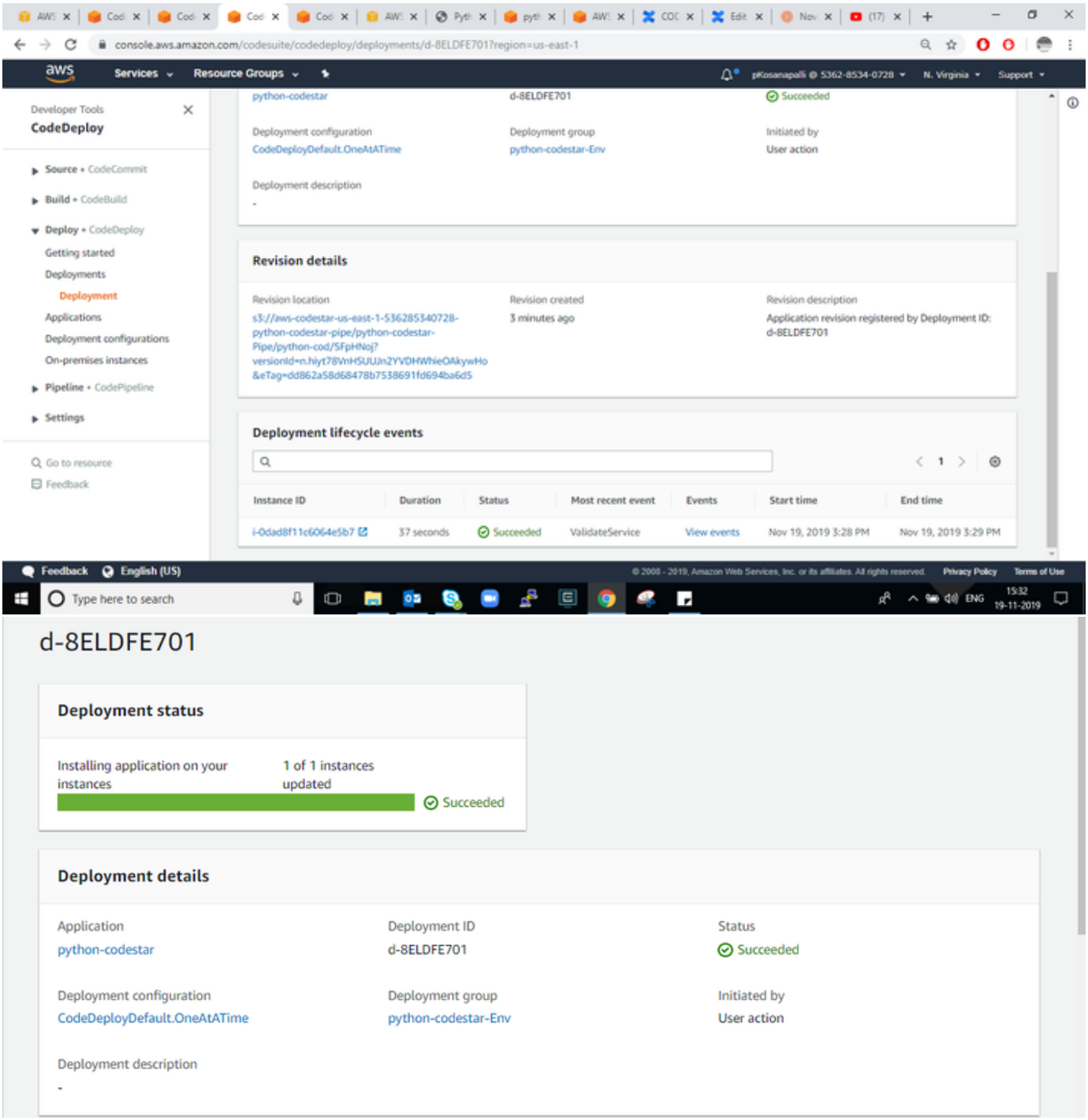


- for code quality coverage, sonar is generating the results to sonar instance.

Code deploy- configurations:

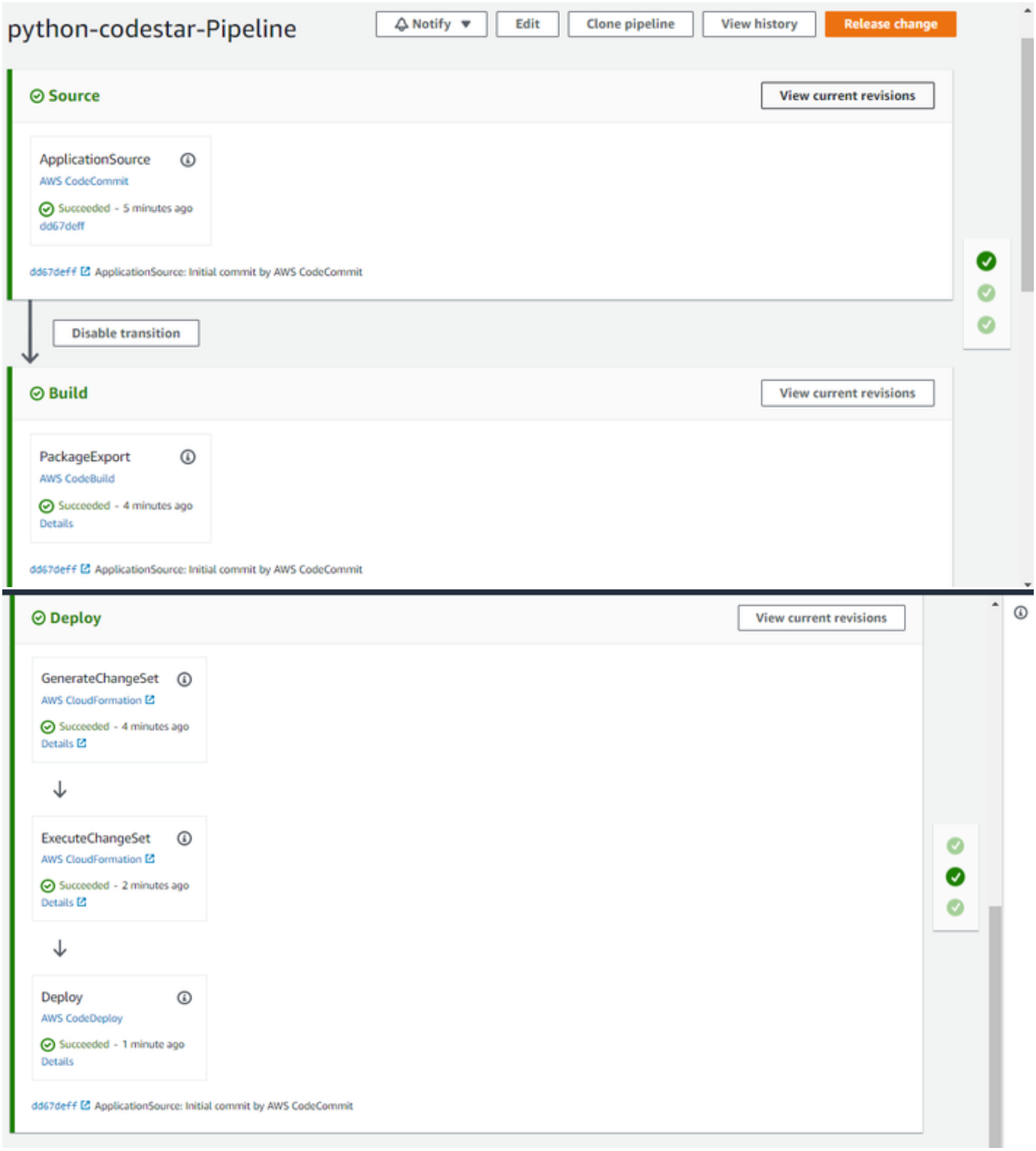
- lets go to code deploy configurations, deployment group details i.e compute platform
- deployment group details- compute platform(tag of cloudformation ec2).
- deployment revisions are taking from s3 bucket (archived files of build).
- Go to each deployment id, you can see the configurations.


python-codestar
Notify
Delete application
Application details
Name
python-codestar
Compute platform
EC2/On-premises
Deployments
Deployment groups
Revisions
Deployment groups
View details
Edit
Create deployment group
1
Name
Status
Last attempted deployment
Last successful deployment
Trigger count
python-codestar-Env
Succeeded
Nov 19, 2019 3:29 PM
Nov 19, 2019 3:29 PM
0
python-codestar-Env
Edit
Delete
Create deployment
Deployment group details
Deployment group name
python-codestar-Env
Application name
python-codestar
Compute platform
EC2/On-premises
Deployment type
In-place
Service role ARN
arn:aws:iam::536285340728:role/CodeStarWorker-python-codestar-ToolChain
Deployment configuration
CodeDeployDefault.OneAtATime
Rollback enabled
False
Environment configuration: Amazon EC2 instances
Key
Value
Environment
python-codestar-WebApp



Code pipeline- configurations:

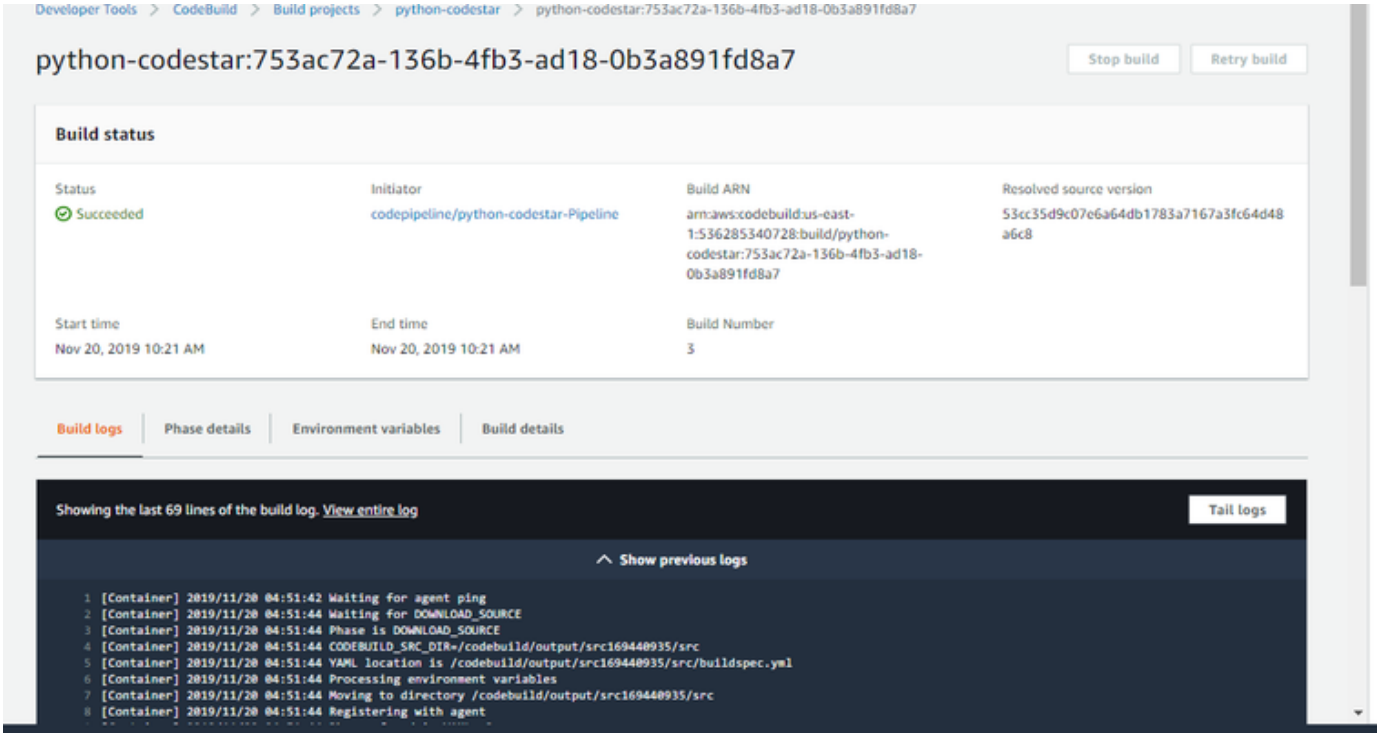
- Go to code star project code-pipeline see the stages.
- you can click each stage, it will be redirect.
- AWS cloudformation for creating ec2 instance.
- AWS code deploy- code will be deploy here.
- click the release change in pipeline, the trigger will be started.
- The triggers can be also configure based on over requirement at specific stage level code build/code-deploy.
- we can add the stages at any specified point and need to configure the task.

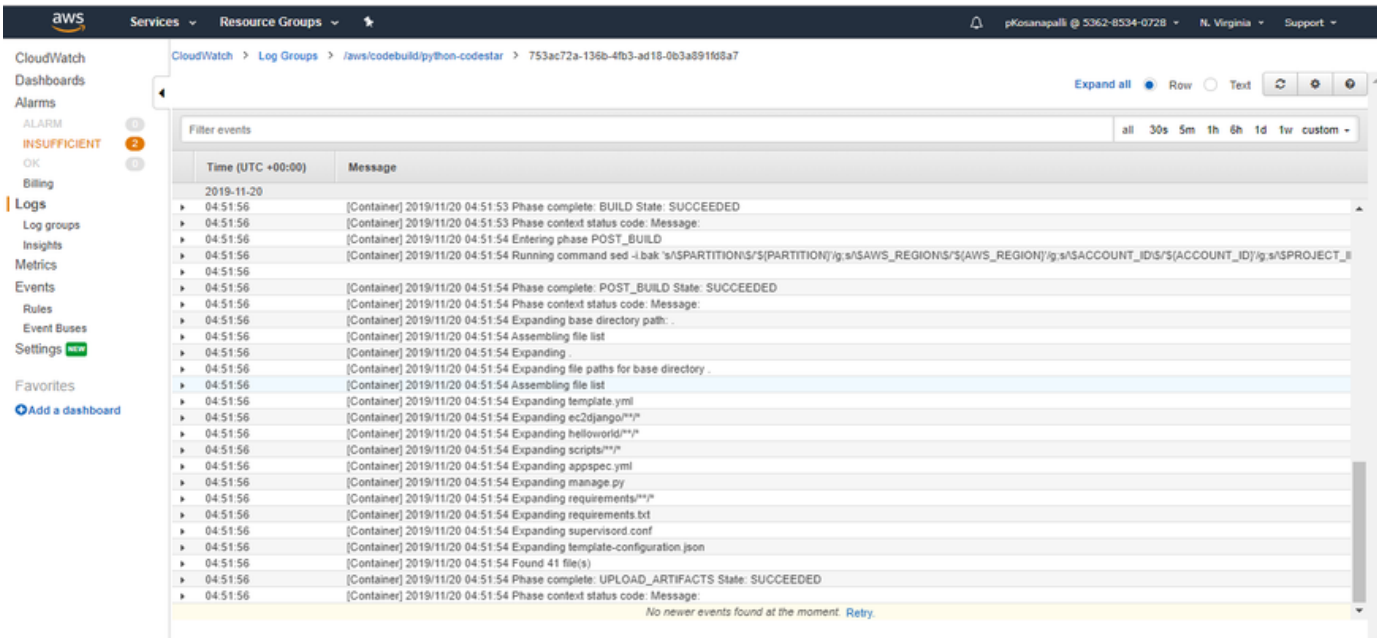


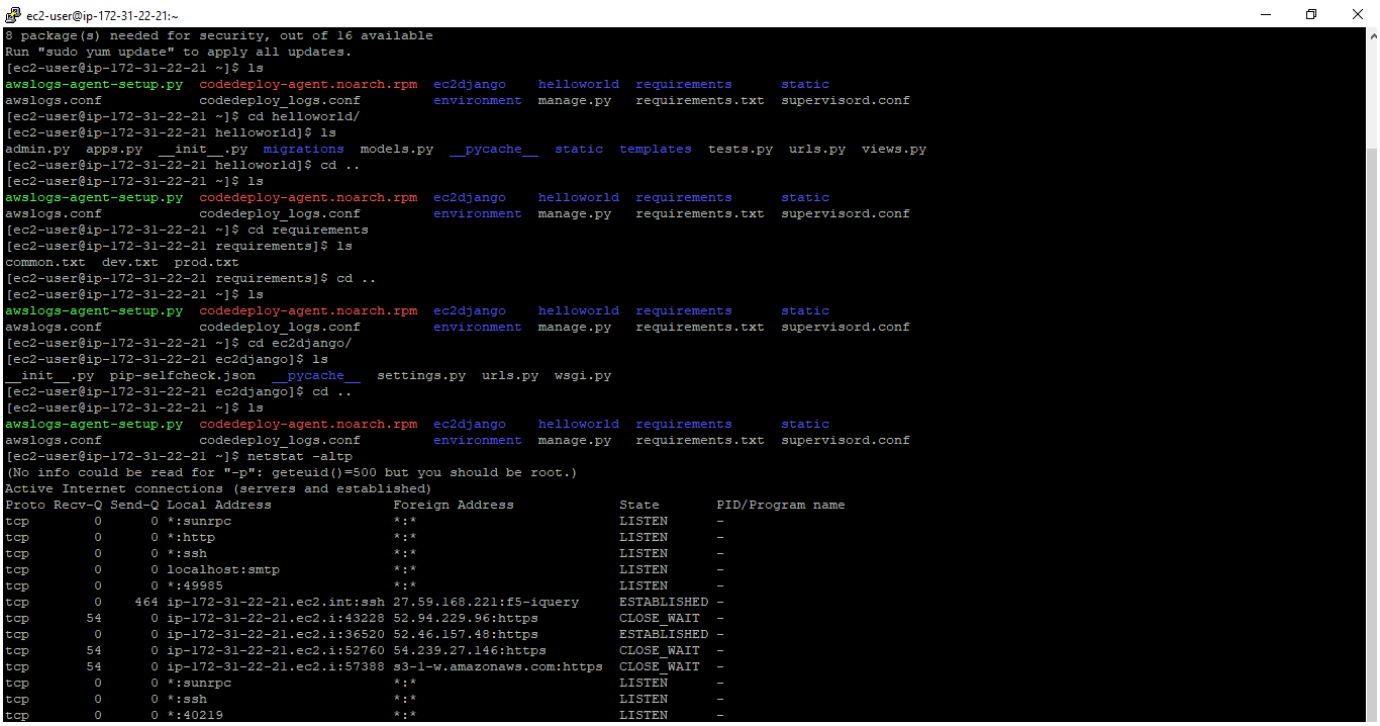
Note: What's next ? How to debug the application

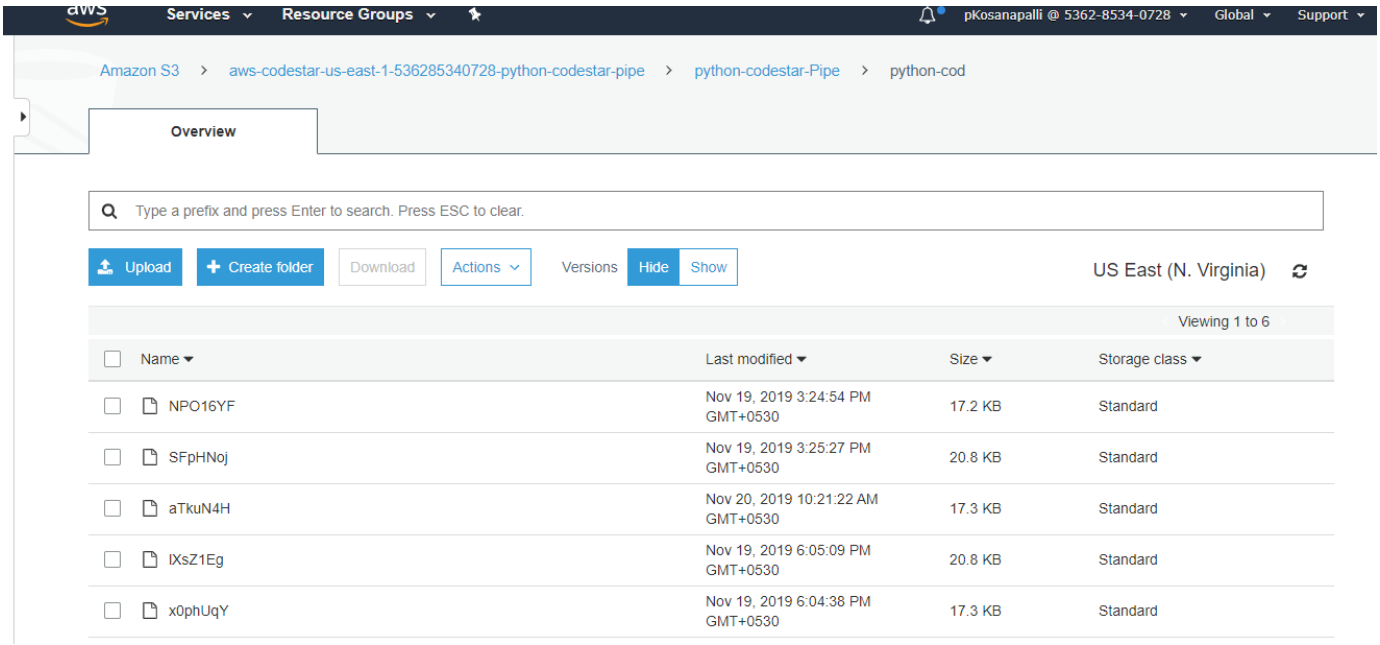
How to debug the code star project application:

- Go to project dashboard project resources click the corresponding resources.
- click the ec2 instaces, redirect console and try to login to the instance.
- Here we can observe the tomcat web-apps root directory, that is the war file of what we accessing the application. we can debugg if any issues facing in the application.
- Code build will store the artifacts in s3. The same artifacts will be taken as source for deployment purpose in code deploy.
- For any role based access go to IAM configurations, debug if any issues faced.
- go to cloudwatch for monitoring the code-build.


Developer Tools > CodeBuild > Build projects > python-codestar > python-codestar:753ac72a-136b-4fb3-ad18-0b3a891fd8a7
python-codestar:753ac72a-136b-4fb3-ad18-0b3a891fd8a7
Stop build
Retry build
Build status
Status
Succeeded
Initiator
codepipeline/python-codestar-Pipeline
Build ARN
arn:aws:codebuild:us-east-1:536285340728:build/python-codestar:753ac72a-136b-4fb3-ad18-0b3a891fd8a7
Resolved source version
53cc35d9c07e6a64db1783a7167a3fc64d48a6c8
Start time
Nov 20, 2019 10:21 AM
End time
Nov 20, 2019 10:21 AM
Build Number
3
Build logs
Phase details
Environment variables
Build details
Showing the last 69 lines of the build log. View entire log
Tail logs
Show previous logs
1 [Container] 2019/11/20 04:51:42 Waiting for agent ping
2 [Container] 2019/11/20 04:51:44 Waiting for DOWNLOAD_SOURCE
3 [Container] 2019/11/20 04:51:44 Phase is DOWNLOAD_SOURCE
4 [Container] 2019/11/20 04:51:44 CODEBUILD_SRC_DIR=/codebuild/output/src169440935/src
5 [Container] 2019/11/20 04:51:44 YAML location is /codebuild/output/src169440935/src/buildspec.yml
6 [Container] 2019/11/20 04:51:44 Processing environment variables
7 [Container] 2019/11/20 04:51:44 Moving to directory /codebuild/output/src169440935/src
8 [Container] 2019/11/20 04:51:44 Registering with agent


aws
Services
Resource Groups
pKosanapalli @ 5362-8534-0728
N. Virginia
Support
CloudWatch
Dashboards
Alarms
ALARM 0
INSUFFICIENT 2
OK 0
Billing
Logs
Log groups
Insights
Metrics
Events
Rules
Event Buses
Settings NEW
Favorites
Add a dashboard
CloudWatch > Log Groups > /aws/codebuild/python-codestar > 753ac72a-136b-4fb3-ad18-0b3a891fd8a7
Expand all
Row
Text
Filter events
all 30s 5m 1h 6h 1d 1w custom
Time (UTC +00:00)
Message
2019-11-20
04:51:56 [Container] 2019/11/20 04:51:53 Phase complete: BUILD State: SUCCEEDED
04:51:56 [Container] 2019/11/20 04:51:53 Phase context status code: Message:
04:51:56 [Container] 2019/11/20 04:51:54 Entering phase POST_BUILD
04:51:56 [Container] 2019/11/20 04:51:54 Running command sed -i.bak 's/\$PARTITION\$/'${PARTITION}'/g;s/\$AWS_REGION\$/'${AWS_REGION}'/g;s/\$ACCOUNT_ID\$/'${ACCOUNT_ID}'/g;s/\$PROJECT_
04:51:56
04:51:56 [Container] 2019/11/20 04:51:54 Phase complete: POST_BUILD State: SUCCEEDED
04:51:56 [Container] 2019/11/20 04:51:54 Phase context status code: Message:
04:51:56 [Container] 2019/11/20 04:51:54 Expanding base directory path: .
04:51:56 [Container] 2019/11/20 04:51:54 Assembling file list
04:51:56 [Container] 2019/11/20 04:51:54 Expanding .
04:51:56 [Container] 2019/11/20 04:51:54 Expanding file paths for base directory .
04:51:56 [Container] 2019/11/20 04:51:54 Assembling file list
04:51:56 [Container] 2019/11/20 04:51:54 Expanding template.yml
04:51:56 [Container] 2019/11/20 04:51:54 Expanding ec2django/**/*
04:51:56 [Container] 2019/11/20 04:51:54 Expanding helloworld/**/*
04:51:56 [Container] 2019/11/20 04:51:54 Expanding scripts/**/*
04:51:56 [Container] 2019/11/20 04:51:54 Expanding appspec.yml
04:51:56 [Container] 2019/11/20 04:51:54 Expanding manage.py
04:51:56 [Container] 2019/11/20 04:51:54 Expanding requirements/**/*
04:51:56 [Container] 2019/11/20 04:51:54 Expanding requirements.txt
04:51:56 [Container] 2019/11/20 04:51:54 Expanding supervisord.conf
04:51:56 [Container] 2019/11/20 04:51:54 Expanding template-configuration.json
04:51:56 [Container] 2019/11/20 04:51:54 Found 41 file(s)
04:51:56 [Container] 2019/11/20 04:51:54 Phase complete: UPLOAD_ARTIFACTS State: SUCCEEDED
04:51:56 [Container] 2019/11/20 04:51:54 Phase context status code: Message:
No newer events found at the moment. Retry.


ec2-user@ip-172-31-22-21:~
8 package(s) needed for security, out of 16 available
Run "sudo yum update" to apply all updates.
[ec2-user@ip-172-31-22-21 ~]$ ls
awslogs-agent-setup.py codedeploy-agent.noarch.rpm ec2django helloworld requirements static
awslogs.conf codedeploy_logs.conf environment manage.py requirements.txt supervisord.conf
[ec2-user@ip-172-31-22-21 ~]$ cd helloworld/
[ec2-user@ip-172-31-22-21 helloworld]$ ls
admin.py apps.py __init__.py migrations models.py __pycache__ static templates tests.py urls.py views.py
[ec2-user@ip-172-31-22-21 helloworld]$ cd ..
[ec2-user@ip-172-31-22-21 ~]$ ls
awslogs-agent-setup.py codedeploy-agent.noarch.rpm ec2django helloworld requirements static
awslogs.conf codedeploy_logs.conf environment manage.py requirements.txt supervisord.conf
[ec2-user@ip-172-31-22-21 ~]$ cd requirements
[ec2-user@ip-172-31-22-21 requirements]$ ls
common.txt dev.txt prod.txt
[ec2-user@ip-172-31-22-21 requirements]$ cd ..
[ec2-user@ip-172-31-22-21 ~]$ ls
awslogs-agent-setup.py codedeploy-agent.noarch.rpm ec2django helloworld requirements static
awslogs.conf codedeploy_logs.conf environment manage.py requirements.txt supervisord.conf
[ec2-user@ip-172-31-22-21 ~]$ cd ec2django/
[ec2-user@ip-172-31-22-21 ec2django]$ ls
__init__.py pip-selfcheck.json __pycache__ settings.py urls.py wsgi.py
[ec2-user@ip-172-31-22-21 ec2django]$ cd ..
[ec2-user@ip-172-31-22-21 ~]$ ls
awslogs-agent-setup.py codedeploy-agent.noarch.rpm ec2django helloworld requirements static
awslogs.conf codedeploy_logs.conf environment manage.py requirements.txt supervisord.conf
[ec2-user@ip-172-31-22-21 ~]$ netstat -altp
(No info could be read for "-p": geteuid()=500 but you should be root.)
Active Internet connections (servers and established)
Proto Recv-Q Send-Q Local Address Foreign Address State PID/Program name
tcp 0 0 *:sunrpc *:* LISTEN -
tcp 0 0 *:http *:* LISTEN -
tcp 0 0 *:ssh *:* LISTEN -
tcp 0 0 localhost:smtp *:* LISTEN -
tcp 0 0 *:49985 *:* LISTEN -
tcp 0 464 ip-172-31-22-21.ec2.int:ssh 27.59.168.221:f5-iquery ESTABLISHED -
tcp 54 0 ip-172-31-22-21.ec2.i:43228 52.94.229.96:https CLOSE_WAIT -
tcp 0 0 ip-172-31-22-21.ec2.i:36520 52.46.157.48:https ESTABLISHED -
tcp 54 0 ip-172-31-22-21.ec2.i:52760 54.239.27.146:https CLOSE_WAIT -
tcp 54 0 ip-172-31-22-21.ec2.i:57388 s3-1-w.amazonaws.com:https CLOSE_WAIT -
tcp 0 0 *:sunrpc *:* LISTEN -
tcp 0 0 *:ssh *:* LISTEN -
tcp 0 0 *:40219 *:* LISTEN -


Services
Resource Groups
pKosanapalli @ 5362-8534-0728
Global
Support
Amazon S3 > aws-codestar-us-east-1-536285340728-python-codestar-pipe > python-codestar-Pipe > python-cod
Overview
Type a prefix and press Enter to search. Press ESC to clear.
Upload
Create folder
Download
Actions
Versions
Hide
Show
US East (N. Virginia)
Viewing 1 to 6
Name
Last modified
Size
Storage class
NPO16YF
Nov 19, 2019 3:24:54 PM GMT+0530
17.2 KB
Standard
SFpHNoj
Nov 19, 2019 3:25:27 PM GMT+0530
20.8 KB
Standard
aTkuN4H
Nov 20, 2019 10:21:22 AM GMT+0530
17.3 KB
Standard
lXsZ1Eg
Nov 19, 2019 6:05:09 PM GMT+0530
20.8 KB
Standard
x0phUqY
Nov 19, 2019 6:04:38 PM GMT+0530
17.3 KB
Standard

that done !!!