

CODESTAR-Build,test,deploy the javaspring application in Ec2.

Working with Projects in AWS CodeStar:

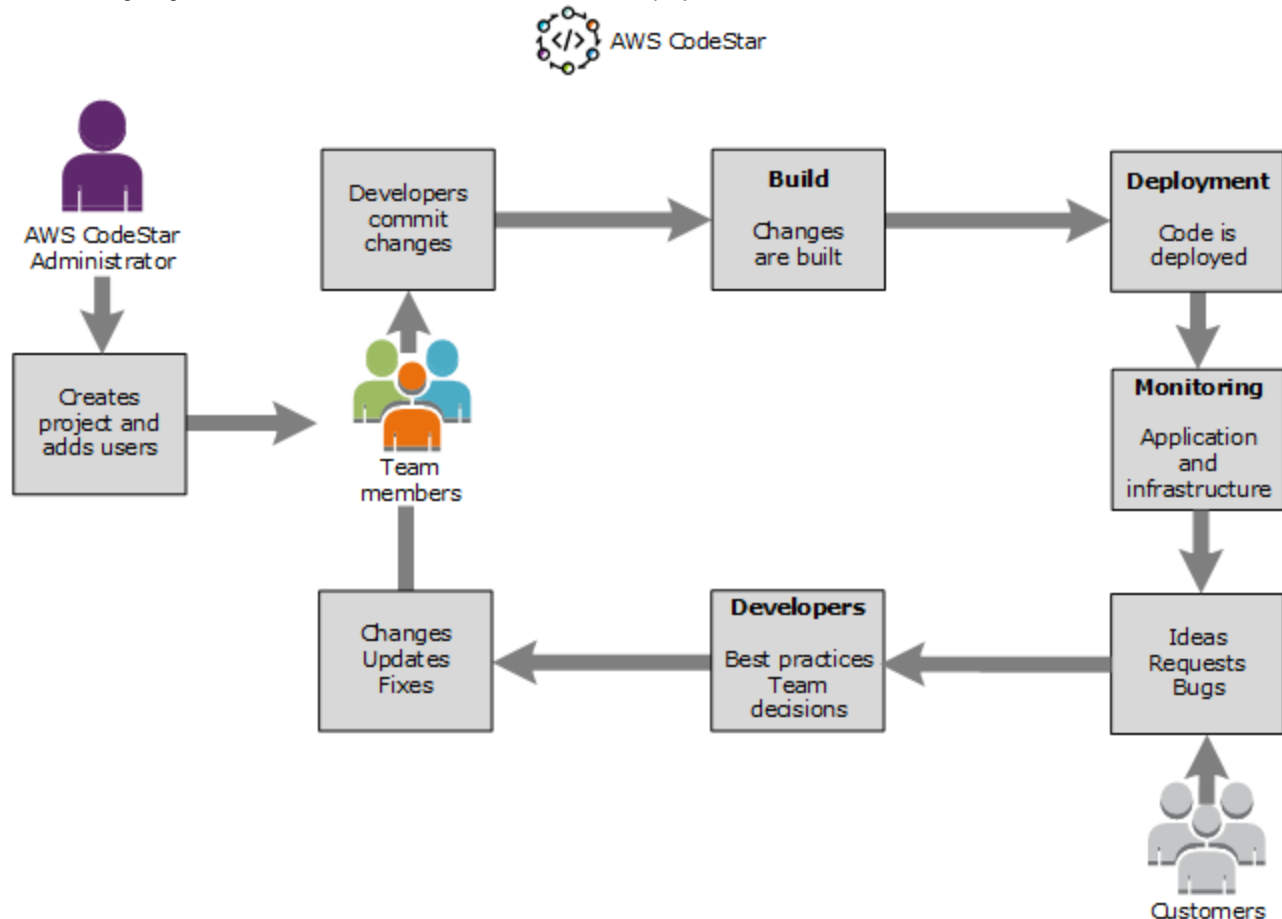
When you use an AWS CodeStar project template, you can quickly create a project that is already configured with the resources you need, including:

- Source repository
- Build environment
- Deployment and hosting resources
- Programming language

The template even includes sample source code so you can start working with your project right away.

After you have a project, you can add or remove resources, customize your project dashboard, and monitor progress.

The following diagram shows a basic workflow in an AWS CodeStar project.



1. A developer with the `AWSCodeStarFullAccess` policy applied creates a project and adds team members to it. Together they write, build, test, and deploy code.
2. The project dashboard provides tools that can be used in real time to view application activity and monitor builds, the flow of code through the deployment pipeline, and more.
3. The team uses the team wiki tile to share information, best practices, and links.
4. They integrate their issue-tracking software to help them track progress and tasks.
5. As customers provide requests and feedback, the team adds this information to the project and integrates it into their project planning and development. As the project grows, the team adds more team members to support their code base.

Create a Project(java 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.

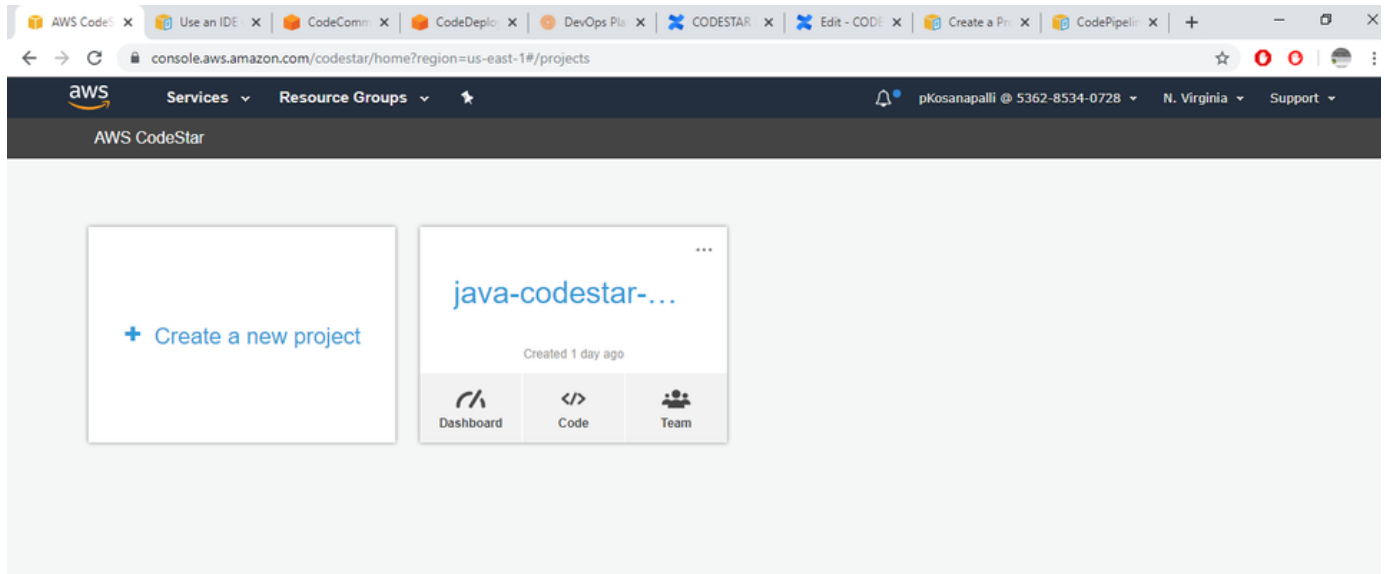
You must complete the steps in [How CodeStar - Getting started- how to create project in codestar in AWS](#) before you can complete the procedures in this topic.

Build, test, package& deploy the javaspring application in ec2:

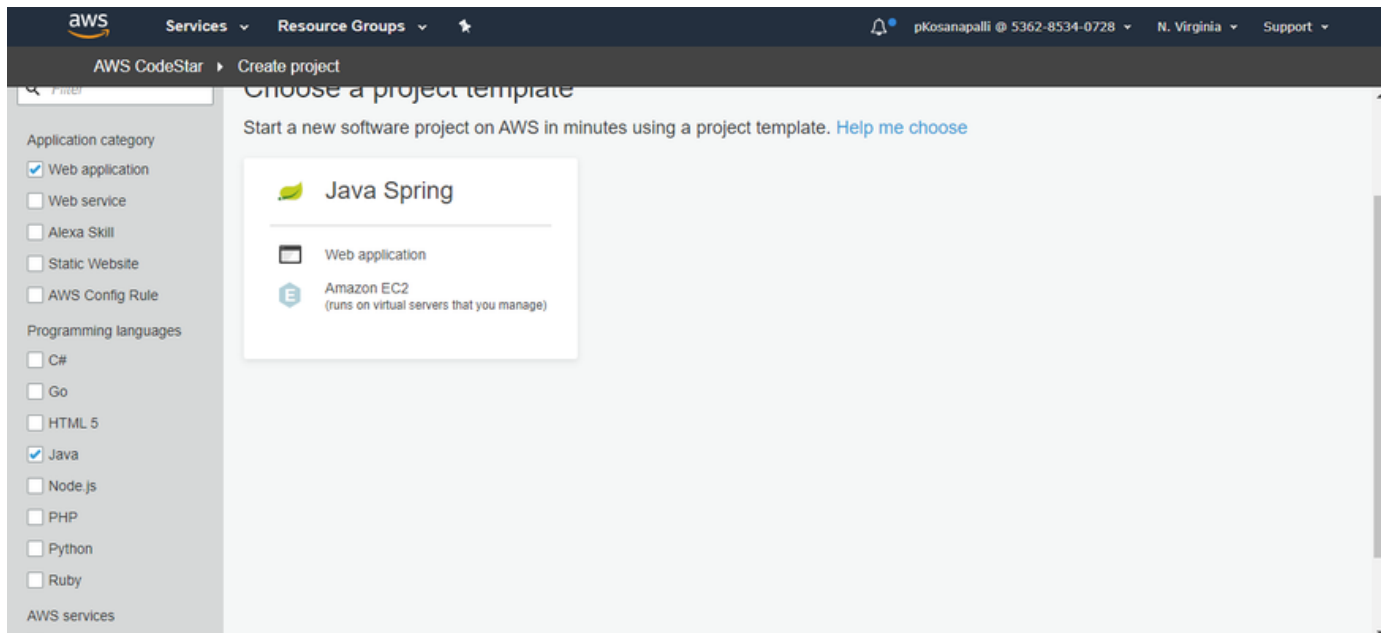
Use the AWS CodeStar console to create a project.

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**, **java** 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 *java-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.

Project details

Project name

java-codestar-ec2


Project ID ⓘ

java-codestar-e


Edit

Which repository do you want to use?

AWS CodeStar will store the project's source code with the service you choose here.



AWS CodeCommit
Highly available Git source control from AWS.
Includes encryption, IAM integration, and more.



GitHub
Creates a GitHub source repository for this project.
Requires an existing GitHub account.

Repository name

java-codestar-ec2

Previous

Next

- 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 documentation [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.

Review project details [Edit Amazon EC2 configuration](#)

AWS CodeStar includes all of the tools and services you need for a development project.
This project includes an AWS CodePipeline connected with the following tools:

Source Build Test Deploy Monitoring

AWS CodeCommit AWS CodeBuild AWS CodeDeploy Amazon CloudWatch

☒ AWS CodeStar would like permission to administer AWS resources on your behalf. [Learn more](#)

[Previous](#) [Create Project](#)

- 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**.

Choose an Amazon EC2 Key Pair

Amazon EC2 uses public-key cryptography to encrypt and decrypt login information with a public-private key pair. AWS stores the public key file, and you store the private key file. Together, they allow you to securely connect to your instance. Linux instances have no password, and you use a key pair to log in using SSH. With Windows instances, you use a key pair to obtain the administrator password and then log in using RDP.

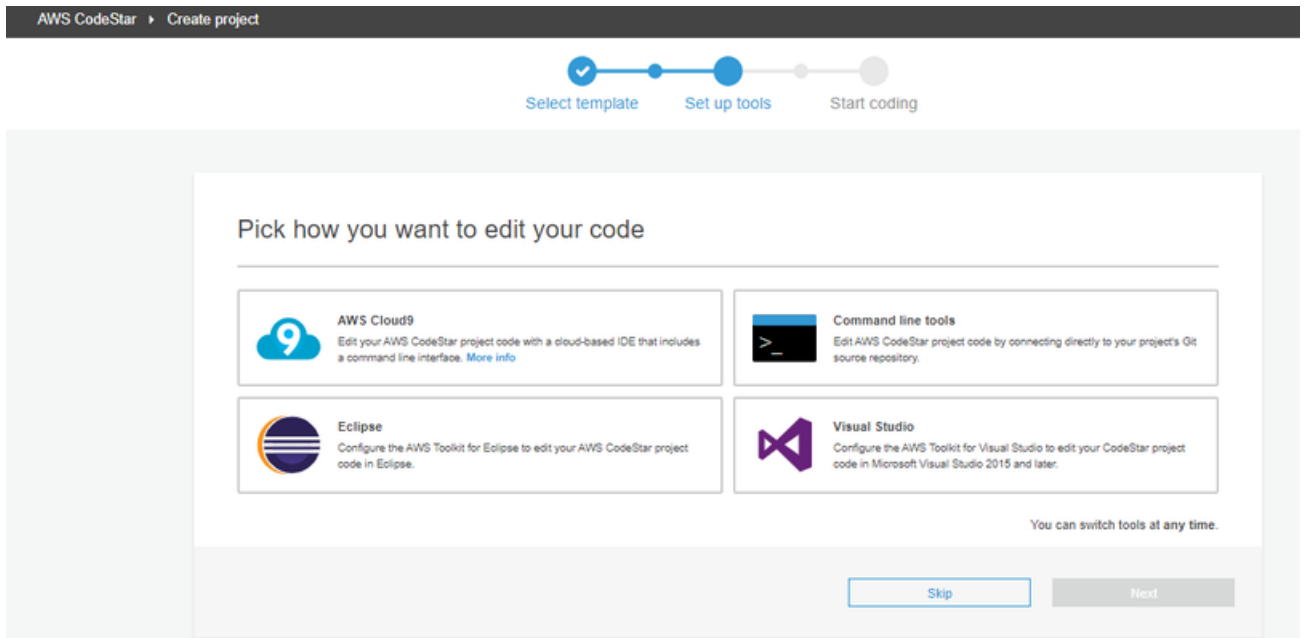
The key pair you choose will be added to the set of keys authorized for this instance. [Learn more](#)

prasanna

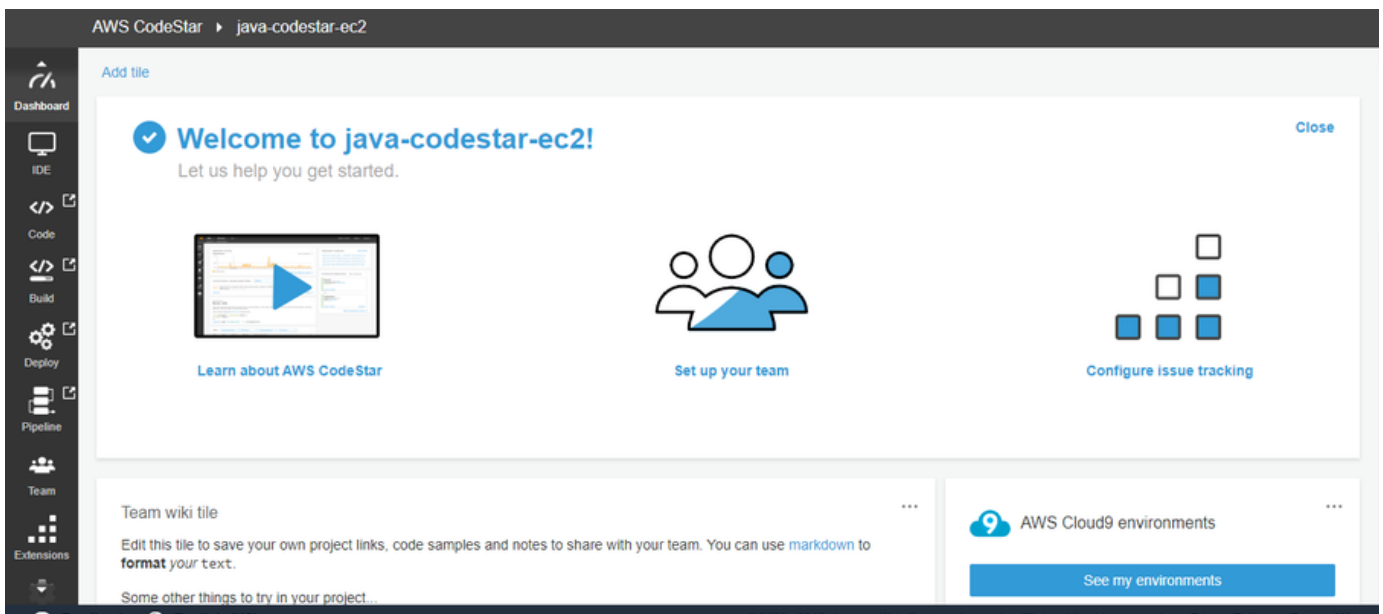
☒ I acknowledge that I have access to the private key file for this key pair (prasanna.pem), and that without this file, I won't be able to log into my instance.

[Cancel](#) [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.



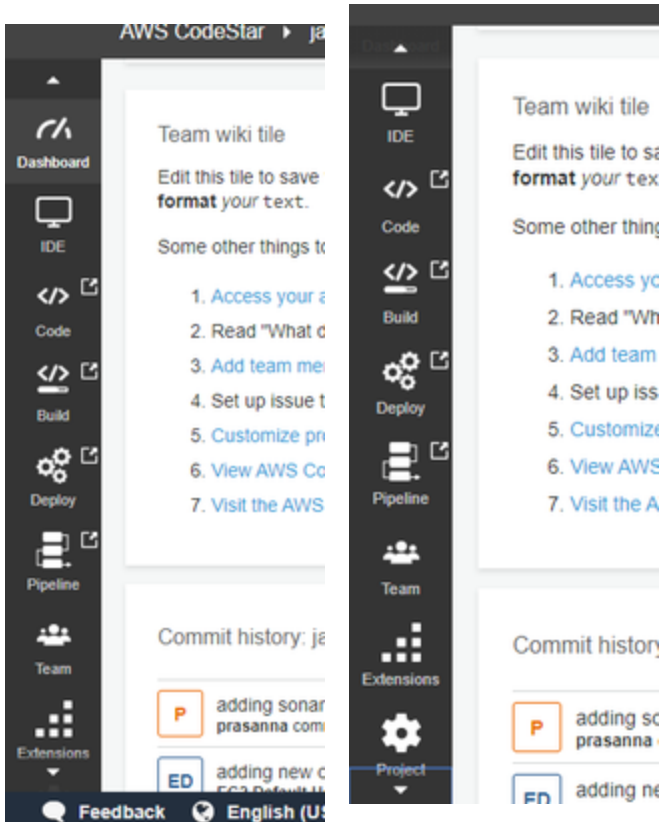
- After your project has been created, you can use the links on the **Welcome** tile to configure other items, such as your user profile in AWS CodeStar.



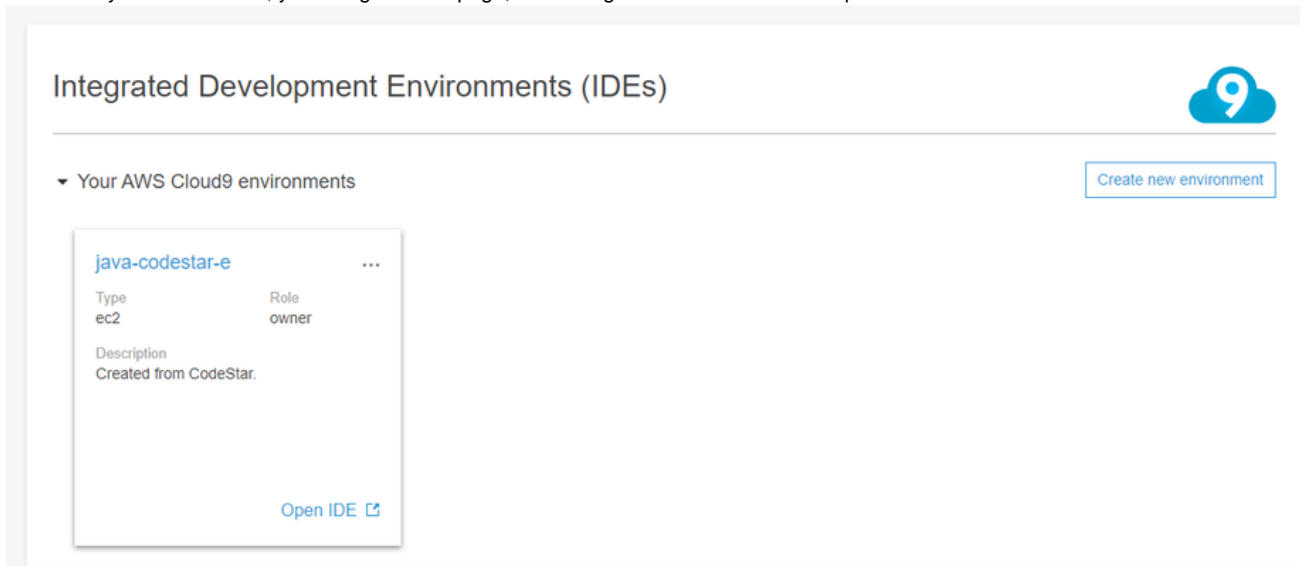
- While your project is being created, you can add team members or configure access to your project repository from the command line or your favorite IDE.
- You can connect your favorite IDE in any one of IDE's AWS cloud9, Eclipse, Vs code, AWS cli., for complete setup with IDE's, see the documentations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

Lets see the content of Welcome board of codestar project:

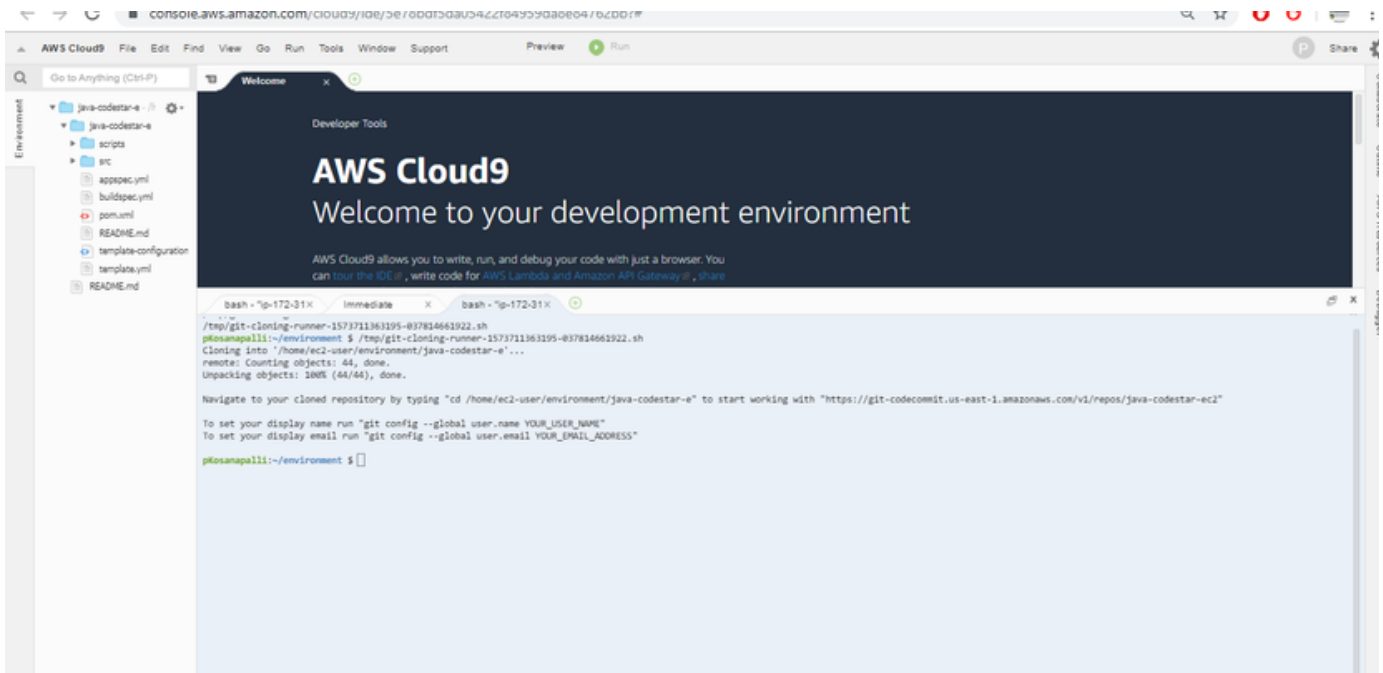
- first you can observe welcome page dashboard of the codestar project, which contains all collaborative services of aws and some of features of codestar.
- On the left side of dashboard you can observe some labels contains IDE, code, build, Deploy,pipeline, team,extensions, project see the below pics.



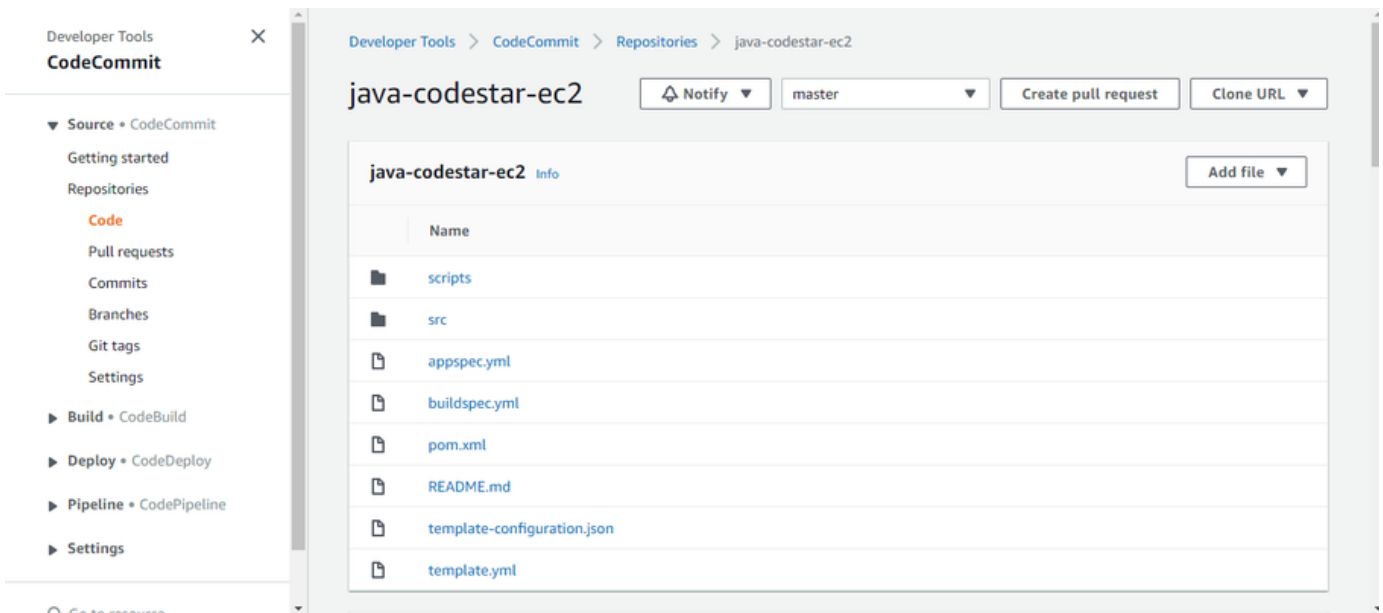
- when ever you click the IDE, you can go to new page, that configure the favorite IDE setup.



- You can observe the different IDEs setup, right now im configured the cloud9 environment for my code editing. 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 documentations here [Code star integrate with IDEs- ECLIPSE, CLOUD9, VS CODE.](#)
- im opted cloud9 IDE, you can open by clicking open IDE environment.



- Now a new instance created & configured for our environment, it contains CLI and repository of your codestar. Through out our document we are using Cloud9 editor.
- Go and see what are things in the instance. this is similar to ubuntu machine in AWS.
- On the left side of dashboard you can observe some labels name code, you click the code, you can go to new page, codestar is configure with new repo name javaspring-ec2.



- Now you can clone the code in IDE and make changes, push it to repo. Same you can observe in the repository.
- On the left side of dashboard you can observe some labels name Build, you click the Build, takes to new page, codestar is configure with new codebuild name javaspring-ec2.
- See the codebuild - creates default codebuild service contains source, build environment(docker image,environment,service role),variables, build spec,artifacts, logs, build history.

Developer Tools **CodeBuild**

- Source • CodeCommit
- Build • CodeBuild
 - Getting started
 - Build projects
 - Build project**
 - Settings
 - Build history
 - Account metrics
- Deploy • CodeDeploy
- Pipeline • CodePipeline
- Settings

Go to resource Feedback

Developer Tools > CodeBuild > Build projects > java-codestar-e

java-codestar-e

Notify Edit Delete build project **Start build**

Configuration

Source provider AWS CodePipeline	Primary repository -	Artifacts upload location -	Build badge Disabled
-------------------------------------	-------------------------	--------------------------------	-------------------------

Build history | Build details | Build triggers | Metrics

Build history Stop build View artifacts View logs Delete builds Retry build

< 1 >

<input type="checkbox"/>	Build run	Status	Build Number	Source version	Submitter	Duration	Completed
<input type="checkbox"/>	java-codestar-e			arn:aws:s3:::aws-codestar-us-east-1-			

- On the left side of dashboard you can observe some labels name Deploy, you click the Deploy, takes to new page, codestar is configure with new codedeploy name javaspring-ec2. Codedeploy contains - deployments, deployment groups (service role, compute platform.)

Developer Tools **CodeDeploy**

- Source • CodeCommit
- Build • CodeBuild
- Deploy • CodeDeploy
 - Getting started
 - Deployments
 - Applications
 - Application**
 - Settings
 - Deployment configurations
 - On-premises instances
- Pipeline • CodePipeline
- Settings

Go to resource

Developer Tools > CodeDeploy > Applications > java-codestar-e

java-codestar-e

Notify Delete application

Application details

Name java-codestar-e	Compute platform EC2/On-premises
-------------------------	-------------------------------------

Deployments | **Deployment groups** | Revisions

Deployment groups View details Edit **Create deployment group**

Q

	Name	Status	Last attempted deployment	Last successful deployment	Trigger count
<input type="radio"/>	java-codestar-e-Env	✔ Succeeded	Nov 14, 2019 2:40 PM	Nov 14, 2019 2:40 PM	0

- On the left side of dashboard you can observe some labels name Pipeline, you click the pipeline, takes to new page, codestar is configure with new codepipeline name javaspring-ec2. Codepipeline contains integrations of code commit, codebuild, code deploy.

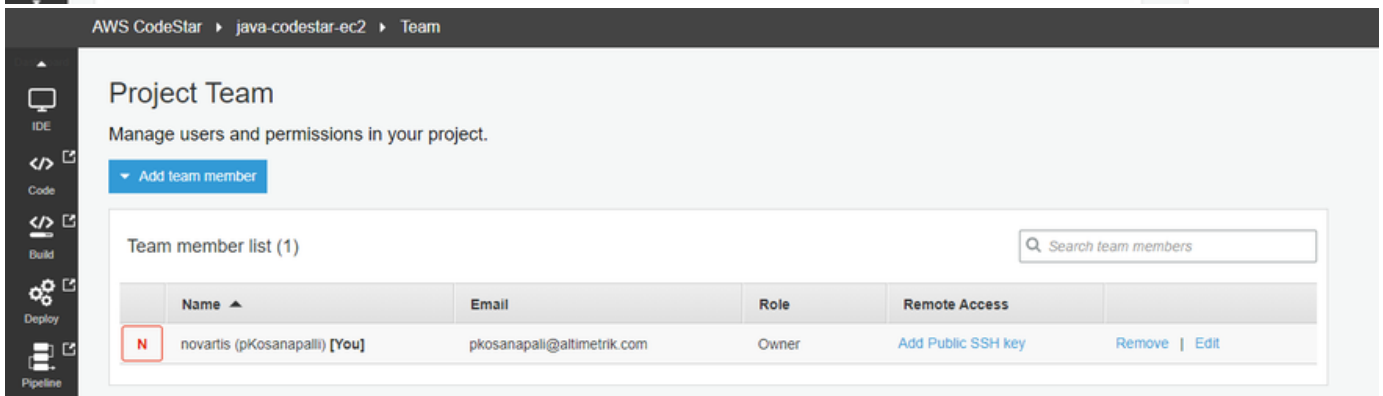
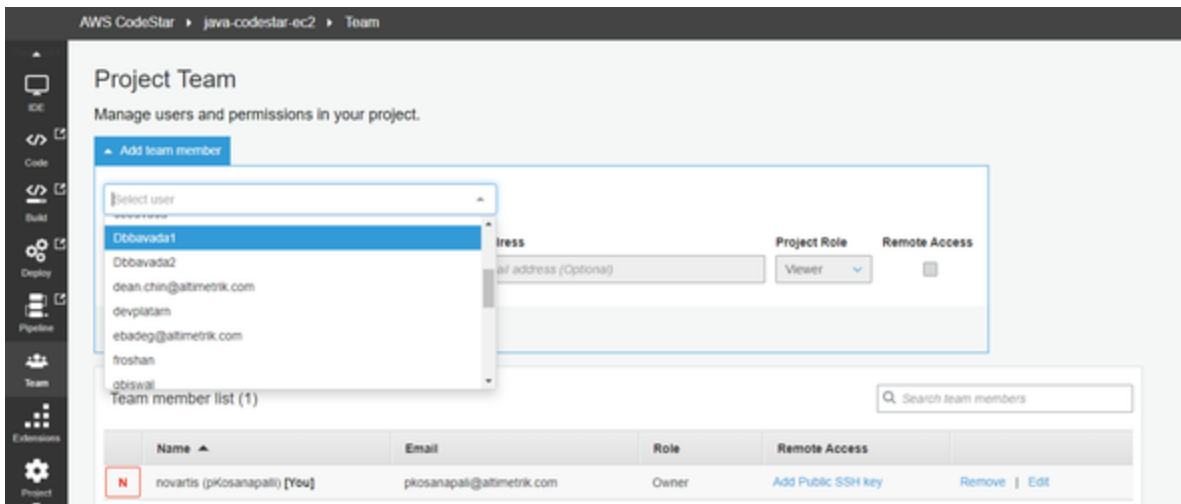
The image displays three sequential screenshots of the AWS CodePipeline console, showing the execution of a pipeline named 'java-codestar-e-Pipeline'. The left sidebar shows the 'CodePipeline' service selected under 'Developer Tools'.

Source Stage: The first screenshot shows the 'Source' stage with a single action named 'ApplicationSource' using 'AWS CodeCommit' as the provider. The action status is 'Succeeded - 3 hours ago' with revision 'c34e55d2'. A 'Disable transition' button is visible below the action.

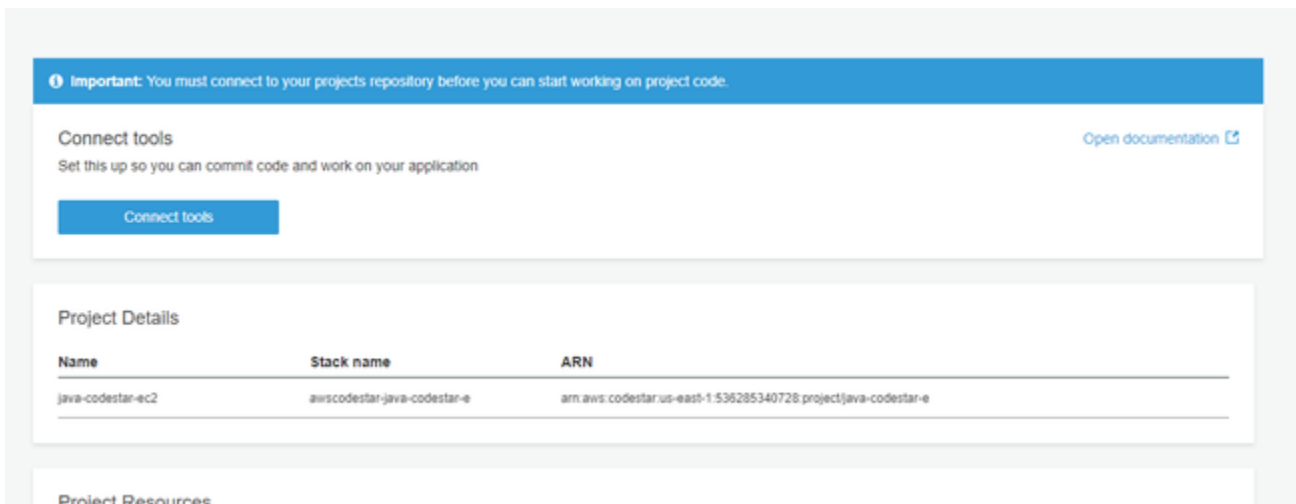
Build Stage: The second screenshot shows the 'Build' stage with a single action named 'PackageExport' using 'AWS CodeBuild' as the provider. The action status is 'Succeeded - 3 hours ago' with revision 'c34e55d2'. A 'Disable transition' button is visible below the action.

Deploy Stage: The third screenshot shows the 'Deploy' stage with three actions: 'GenerateChangeSet' (AWS CloudFormation), 'ExecuteChangeSet' (AWS CloudFormation), and 'Deploy' (AWS CodeDeploy). All three actions are in a 'Succeeded - 3 hours ago' state with revision 'c34e55d2'. A 'Disable transition' button is visible below the 'Deploy' action.

- On the left side of dashboard you can observe some labels name Team, you click the Team, takes to new page, codestar is configure with new feature, add team mates and restrict them the role.



- On the left side of dashboard you can observe some labels name extensions, you click the extensions, codestar is configure with new feature, integrations i.e, github, jira. for complete setup for integration see this [codestar- How to integrate with JIRA](#).
- On the left side of dashboard you can observe some labels name project details, you click this, you can observe the project details, project resources, connected tools.



Project Resources

Type	Name	ARN
AWS Cloud9	environment:5e78bdf5da05422f84959da8...	arn:aws:cloud9:us-east-1:536285340728:environment:5e78bdf5da05422f84959da8e84762bb
AWS CloudFormation	stack/awscodestar-java-codestar-e/fa8968...	arn:aws:cloudformation:us-east-1:536285340728:stack/awscodestar-java-codestar-e/fa896890-05ff-11ea-8929-12b44...
AWS CloudFormation	stack/awscodestar-java-codestar-e-infrastr...	arn:aws:cloudformation:us-east-1:536285340728:stack/awscodestar-java-codestar-e-infrastructure/a01a8280-0600-11...
AWS CodeBuild	project:java-codestar-e	arn:aws:codebuild:us-east-1:536285340728:project:java-codestar-e
AWS CodeCommit	java-codestar-ec2	arn:aws:codecommit:us-east-1:536285340728:java-codestar-ec2
AWS CodeDeploy	deploymentgroup:java-codestar-e/java-cod...	arn:aws:codedeploy:us-east-1:536285340728:deploymentgroup:java-codestar-e/java-codestar-e-Env
AWS CodeDeploy	application:java-codestar-e	arn:aws:codedeploy:us-east-1:536285340728:application:java-codestar-e
AWS CodePipeline	java-codestar-e-Pipeline	arn:aws:codepipeline:us-east-1:536285340728:java-codestar-e-Pipeline

AWS IAM	role/CodeStarWorker-java-codestar-e-Tool...	arn:aws:iam::536285340728:role/CodeStarWorker-java-codestar-e-ToolChain
AWS IAM	role/CodeStarWorker-java-codestar-e-Web...	arn:aws:iam::536285340728:role/CodeStarWorker-java-codestar-e-WebApp
AWS IAM	policy/CodeStar_java-codestar-e_Permissi...	arn:aws:iam::536285340728:policy/CodeStar_java-codestar-e_PermissionsBoundary
AWS IAM	role/CodeStarWorker-java-codestar-e-Clou...	arn:aws:iam::536285340728:role/CodeStarWorker-java-codestar-e-CloudFormation
Amazon EC2	instance/i-05fd4633c533c4e5	arn:aws:ec2:us-east-1:536285340728:instance/i-05fd4633c533c4e5
Amazon EC2	security-group/sg-050bf849e4230a47d	arn:aws:ec2:us-east-1:536285340728:security-group/sg-050bf849e4230a47d
Amazon S3	aws-codestar-us-east-1-536285340728-jav...	arn:aws:s3::aws-codestar-us-east-1-536285340728-java-codestar-e-pipe

- 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.
- come back to the dashboard of codestar javaspring app project. you can see dashboard with some blog views, which are configured to view title. they are as below.

AWS CodeStar ▶ java-codestar-ec2

Team wiki tile

Edit this tile to save your own project links, code samples and notes to share with your team. You can use [markdown](#) to **format** your text.

Some other things to try in your project...

- [1. Access your application](#)
- [2. Read "What do I do next?" in README.md in project source repository](#)
- [3. Add team members](#)
- [4. Set up issue tracking \(under "Extensions"\)](#)
- [5. Customize project dashboard](#)
- [6. View AWS CodeStar documentation](#)
- [7. Visit the AWS CodeStar forum](#)

AWS Cloud9 environments

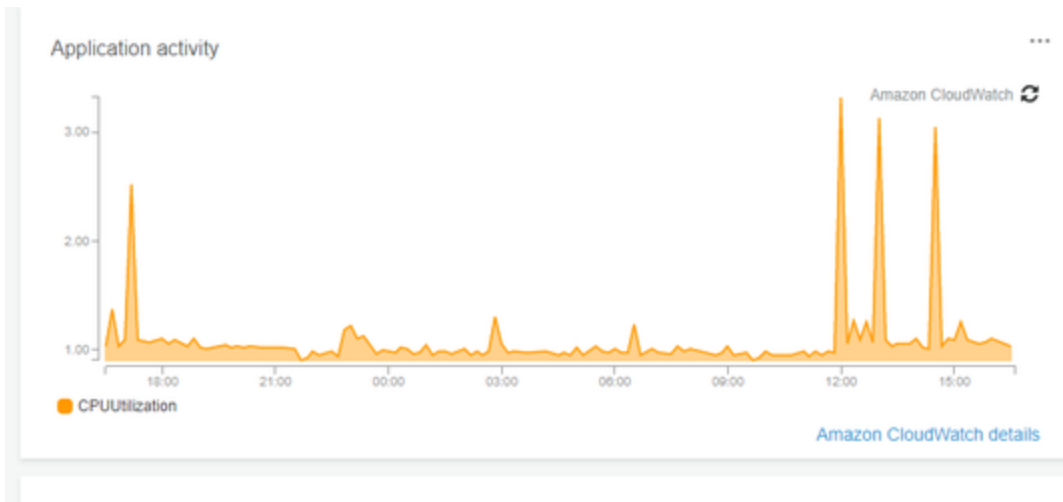
[See my environments](#)

Application endpoints

<http://ec2-34-229-174-236.compute-1.amazonaws.com>

Continuous deployment AWS CodePipeline

- AWS cloud9 environment you can redirect to IDE dashboard.
- 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.



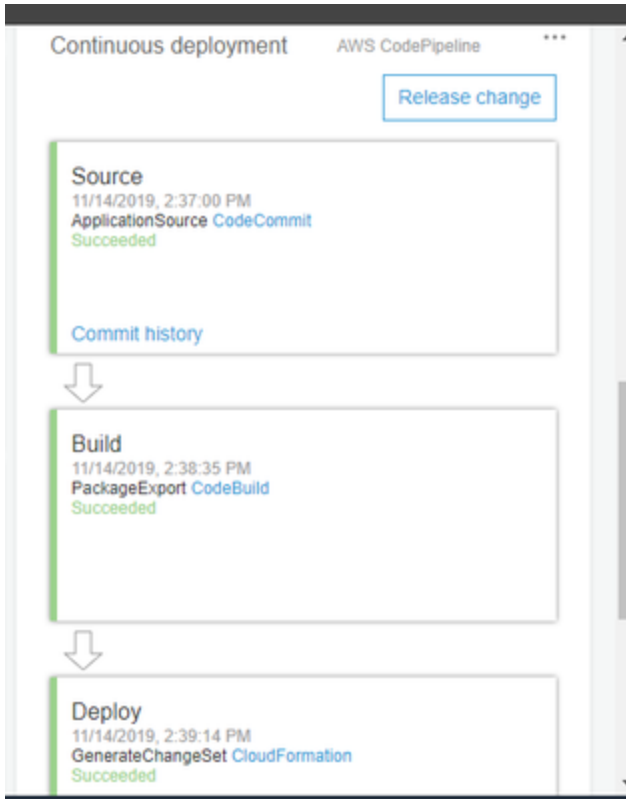
- commit history- view latest changes of codecommit in this project.

Commit history: java-codestar-ec2 master

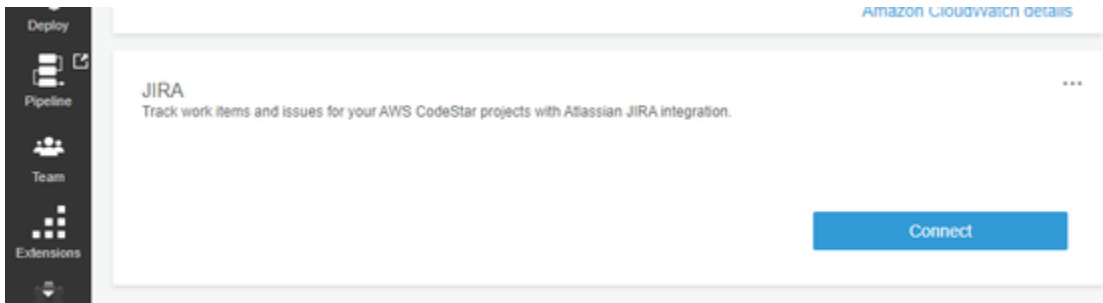
P	adding sonar prasanna committed 3 hours ago	c34a55d
ED	adding new content EC2 Default User committed 4 hours ago	47f7da7
AC	Initial commit by AWS CodeCommit AWS CodeCommit committed 1 day ago	e556040

[Connect](#) [AWS CodeCommit details](#)

- continuous deployment- dashboard view for code pipeline.



- you can configure the Jira issue with code star. for more details see [codestar- How to integrate with JIRA.](#)

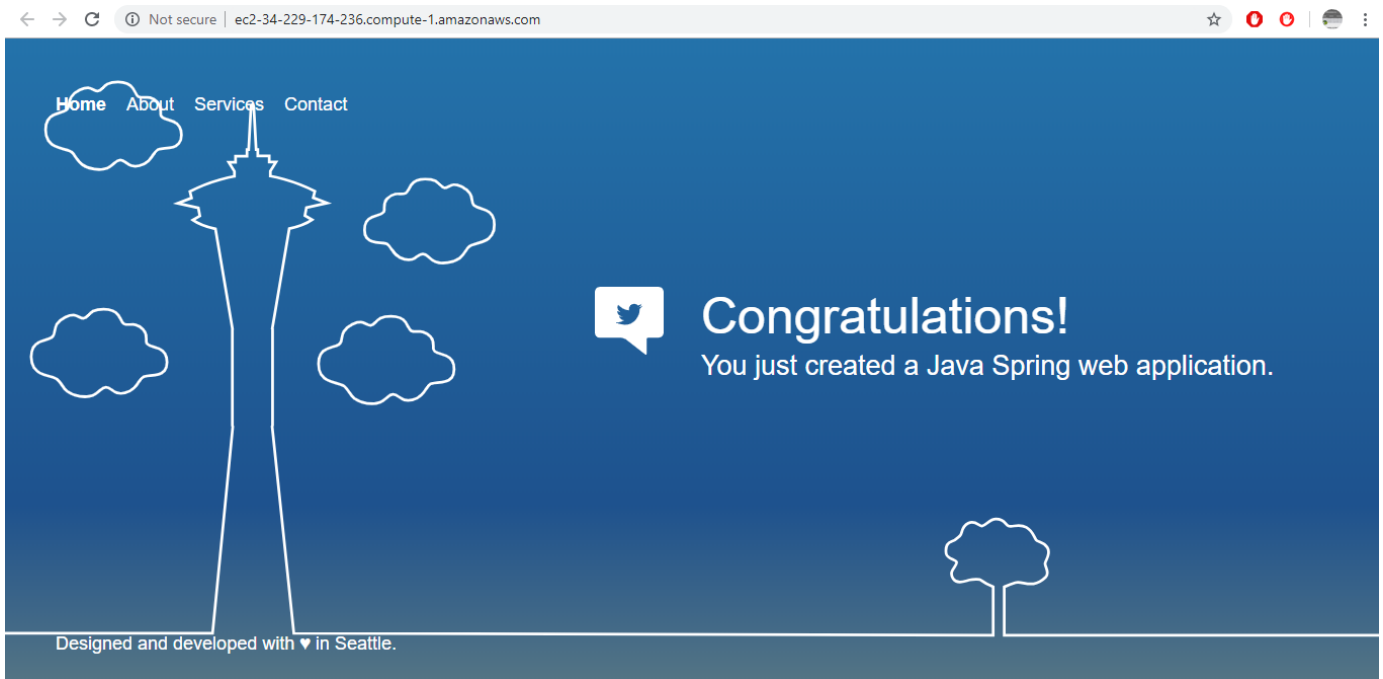


Build, test, package& deploy the javaspring application in ec2:

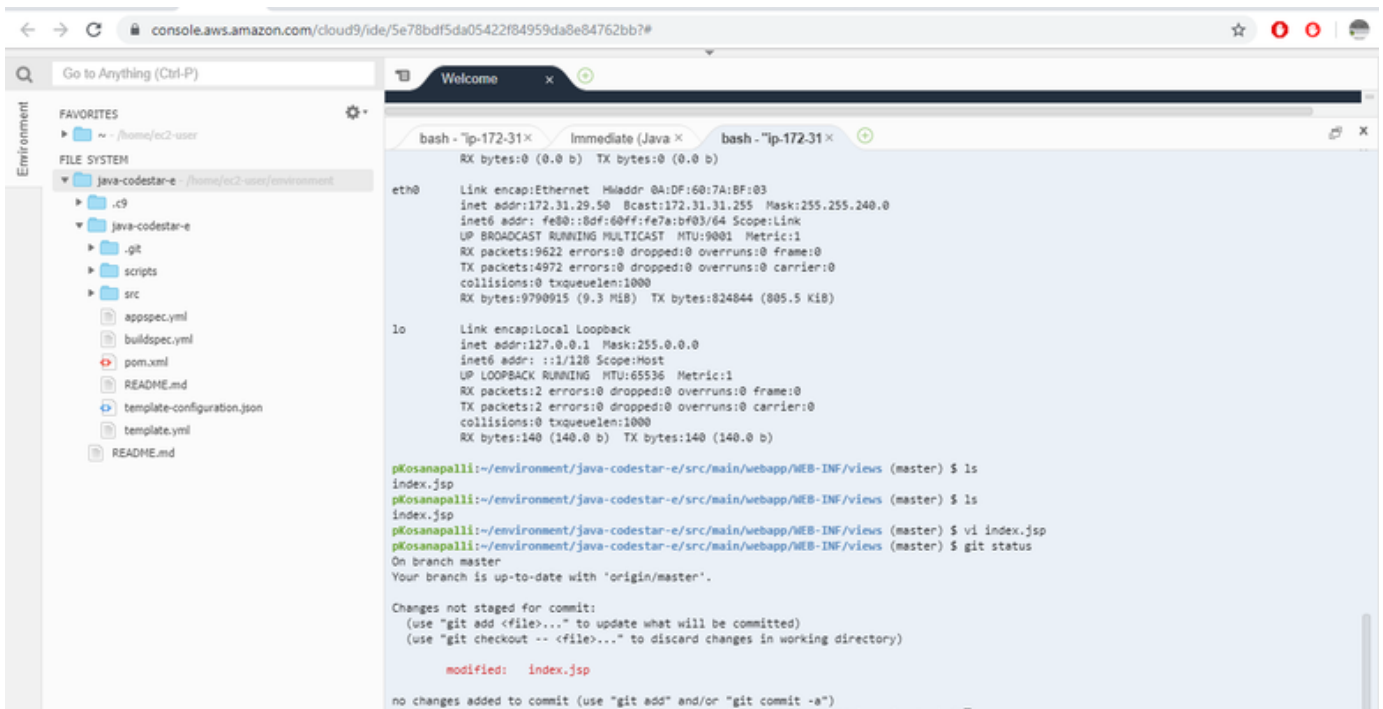
- Now we have created a javaspring project in codestar.
- Go to project explore click the endpoint(url of application), redirect to new url of your application, this url can be configure based on our requirement in the DNS service.

`http://ec2-34-229-174-236.compute-1.amazonaws.com/`

- you can able to see the new java-spring 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,VS CODE.](#)



The screenshot shows an IDE with a terminal window and a CodeCommit diff view. The terminal window displays the following commands and output:

```
modified: index.jsp

no changes added to commit (use "git add" and/or "git commit -a")
pkosanapalli:~/environment/java-codestar-e/src/main/webapp/WEB-INF/views (master) $ git add .
pkosanapalli:~/environment/java-codestar-e/src/main/webapp/WEB-INF/views (master) $ git commit -m "adding new content"
[master 47f7da7] adding new content
Committer: EC2 Default User <ec2-user@ip-172-31-29-50.ec2.internal>
Your name and email address were configured automatically based
on your username and hostname. Please check that they are accurate.
You can suppress this message by setting them explicitly:

    git config --global user.name "Your Name"
    git config --global user.email you@example.com

After doing this, you may fix the identity used for this commit with:

    git commit --amend --reset-author

1 file changed, 2 insertions(+), 2 deletions(-)
pkosanapalli:~/environment/java-codestar-e/src/main/webapp/WEB-INF/views (master) $ git config --global user.name "prasanna"
pkosanapalli:~/environment/java-codestar-e/src/main/webapp/WEB-INF/views (master) $ git config --global user.email pkosanapalli@altimetrik.co
pkosanapalli:~/environment/java-codestar-e/src/main/webapp/WEB-INF/views (master) $ git commit -m "adding new content"
On branch master
Your branch is ahead of 'origin/master' by 1 commit.
(use "git push" to publish your local commits)

nothing to commit, working tree clean
pkosanapalli:~/environment/java-codestar-e/src/main/webapp/WEB-INF/views (master) $ git push
Counting objects: 8, done.
Compressing objects: 100% (6/6), done.
Writing objects: 100% (8/8), 652 bytes | 217.00 KiB/s, done.
Total 8 (delta 3), reused 0 (delta 0)
To https://git-codecommit.us-east-1.amazonaws.com/v1/repos/java-codestar-ec2
e556040..47f7da7 master -> master
pkosanapalli:~/environment/java-codestar-e/src/main/webapp/WEB-INF/views (master) $
```

The CodeCommit diff view shows the changes in the file `src/main/webapp/WEB-INF/views/index.jsp`. The diff is displayed in a table format with line numbers and changes:

Line	Old	New
46		
47	<div class="text">	<div class="text">
48	<h1>Congratulations!</h1>	<h1>Congratulations!</h1>
49	- <h2>You just created a Java Spring web application.</h2>	+ <h2>You just created a sample Novartis testing java Spring web application.</h2>
50	</div>	</div>
51	</div>	</div>
52	</div>	</div>
53	<div>	<div>
54	<div>	<div>
55	- <p class="footer-content">Designed and developed with 	+ <p class="footer-content">Designed and developed by prasanna
56	</p>	</p>
57	</div>	</div>
58	<script src="{setBackgroundJs}"></script>	<script src="{setBackgroundJs}"></script>

- 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.

Developer Tools

CodePipeline

Source • CodeCommit

Build • CodeBuild

Deploy • CodeDeploy

Pipeline • CodePipeline

Getting started

Pipelines

Pipeline

History

Settings

Go to resource

Feedback

Developer Tools > CodePipeline > Pipelines > java-codestar-e-Pipeline

java-codestar-e-Pipeline

Notify

Edit

Clone pipeline

View history

Release change

Source

View current revisions

ApplicationSource

AWS CodeCommit

Succeeded - 1 minute ago

47f7da72

47f7da72 ApplicationSource: adding new contant

Disable transition

Build

View current revisions

Feedback English (US)

© 2008 - 2019, Amazon Web Services, Inc. or its affiliates. All rights reserved. Privacy Policy Terms of Use

Type here to search

12:06 14-11-2019

Developer Tools

CodePipeline

Source • CodeCommit

Build • CodeBuild

Deploy • CodeDeploy

Pipeline • CodePipeline

Getting started

Pipelines

Pipeline

History

Settings

Go to resource

Feedback

Developer Tools > CodePipeline > Pipelines > java-codestar-e-Pipeline > Execution history

Execution history

View details

Execution ID

Status

Source revisions

Duration

Completed

Trigger

b01b067d-e641-4102-9f12-8516abc7ab4c

In progress

ApplicationSource – 47f7da72: adding new contant

0 seconds

-

CloudWatch Event - am:aws:events:us-east-1:536285340728:rule/aws-codestar-java-codestar-e-SourceEvent

088b90a7-b7b1-4491-aae7-e9197b6a3a90

Succeeded

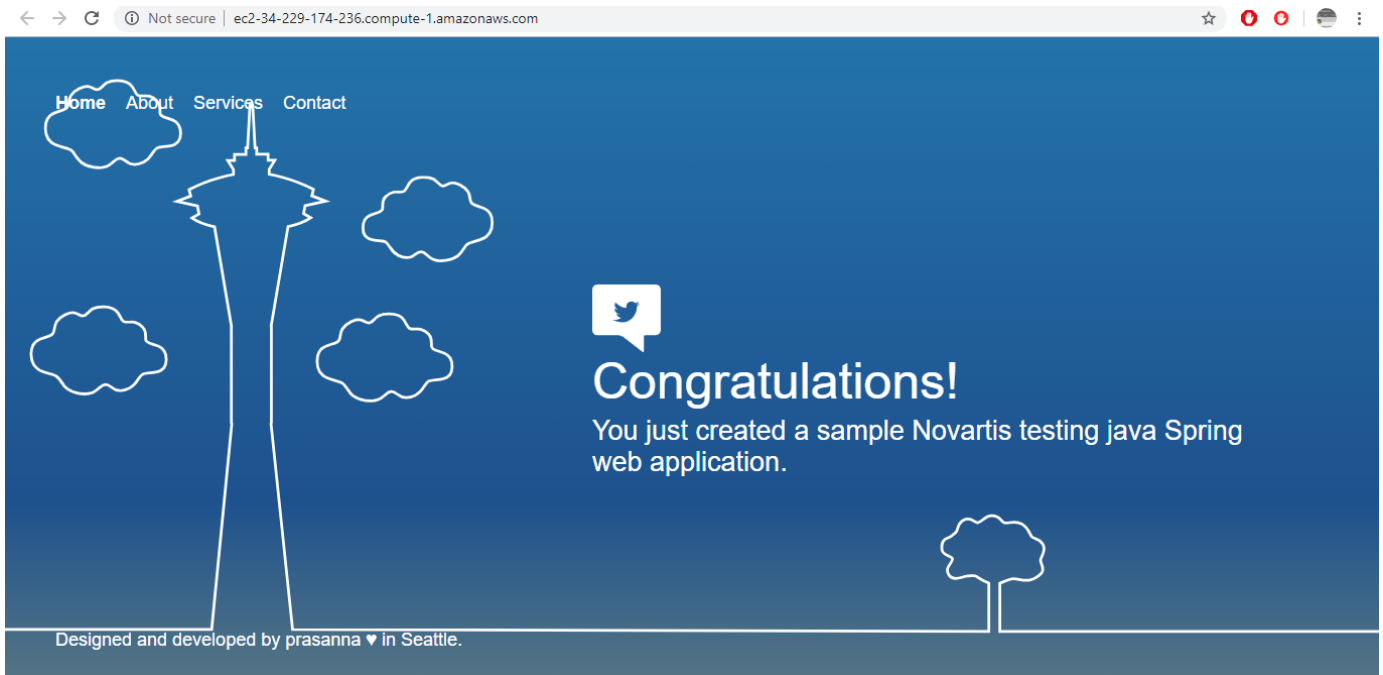
ApplicationSource – e5560404: Initial commit by AWS CodeCommit

4 minutes 40 seconds

Nov 13, 2019 4:03 PM

CreatePipeline - am:aws:sts::536285340728:assume-role/aws-codestar-servic-

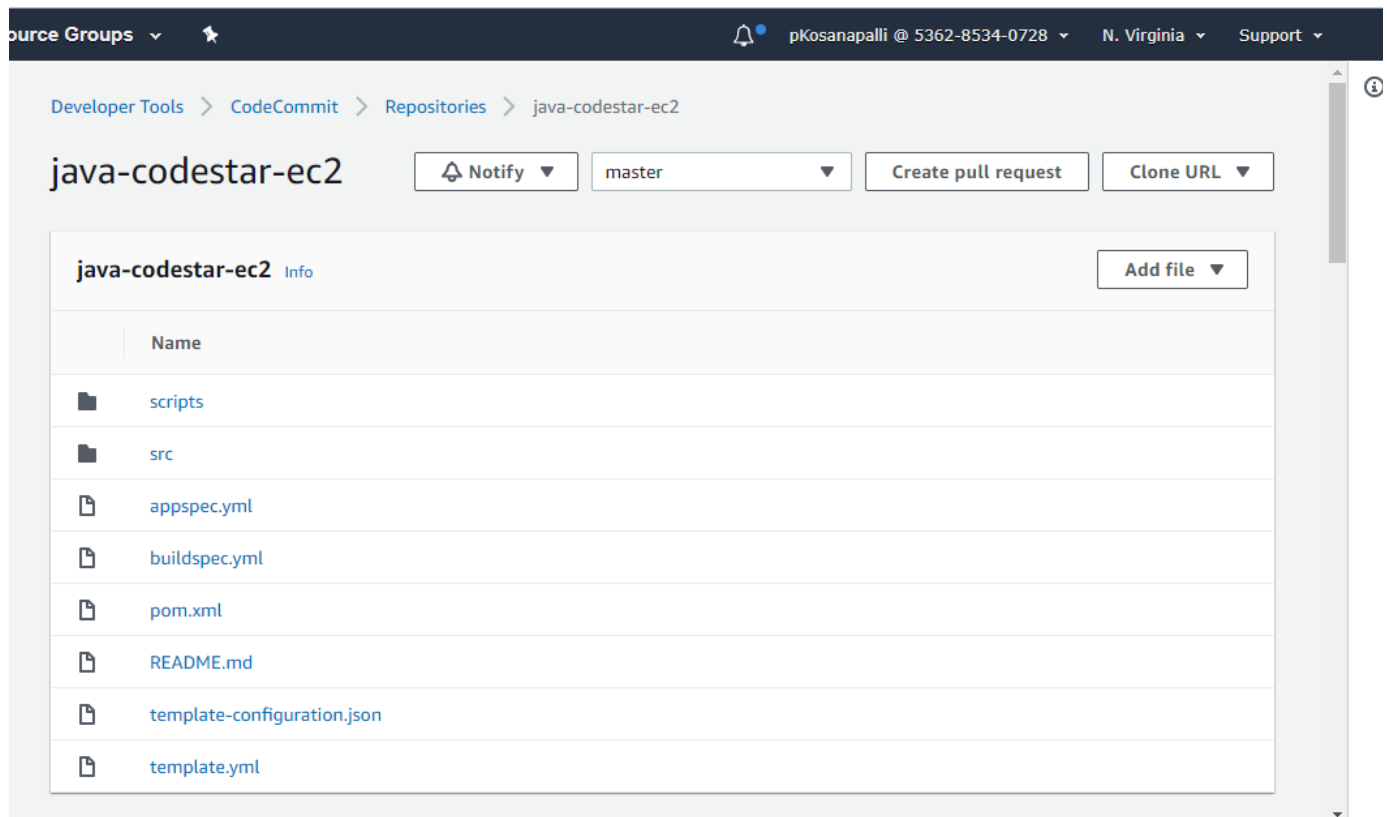
- Go to the Url of the application, see the changes we made.



- Now we have done the build, test, deploy the java-spring application.

How we build, test, deploy the javaspring 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 contains a simple Java web application deployed by AWS CodeDeploy and AWS CloudFormation to an Amazon EC2 server.

What's Here

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 the web application

pom.xml - this file is the Maven Project Object Model for the web application

src/main - this directory contains your Java service source files

src/test - this directory contains your Java service unit test files

scripts/ - this directory contains scripts used by AWS CodeDeploy when installing and deploying your 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.

Name: java-codestar-e

Project ARN:

arn:aws:codebuild:us-east-1:536285340728:project/java-codestar-e

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-setup.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.conf
          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 java app deployment in tomcat. After instance created this script runs includes(shell scripts).

```
version: 0.0 (this will deploy the s3 jar file in tomcat)
os: linux
files:
  - source: target/ROOT.war (build code is stored)
    destination: /home/ec2-user/javaapp (deployment target in ec2 instance)
hooks:
  AfterInstall:
    - location: scripts/install_dependencies(installing tomcat and java)

      timeout: 300
      runas: root
    - location: scripts/codestar_remote_access(configuring the remote access)
      timeout: 300
      runas: root
    - location: scripts/start_server(service restart)
      timeout: 300
      runas: root
```

- The scripts are below install priority wise.

java-codestar-ec2/scripts/codestar_remoteaccess	java-codestar-ec2/scripts/install_dependencies
<pre>#!/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://aws-codestar-templates-common.s3.amazonaws.com/us-east-1/get_authorized_keys chmod 755 /usr/local/bin/get_authorized_keys sed -i '/AuthorizedKeysCommand /s/.*AuthorizedKeysCommand Vusr/local/bin/get_authorized_keys/g' /etc/ssh/ssh_config sed -i '/AuthorizedKeysCommandUser /s/.*AuthorizedKeysCommandUser root/g' /etc/ssh/ssh_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</pre>	<pre>#!/bin/bash cd /home/ec2-user/javaapp wget https://s3.amazonaws.com/dhqs-mirror-iad/af.tar.gz mkdir /opt/tomcat tar xvf apache-tomcat-8*tar.gz -C /opt/tomcat --strip cd /opt/tomcat sed -i 's port="8080" port="80" g' conf/server.xml rm -rf webapps/ROOT mv /home/ec2-user/javaapp/ROOT.war webapps/ yum -y install java-1.8.0-openjdk-1.8.0.171 alternatives --set java /usr/lib/jvm/jre-1.8.0-openjdk</pre>

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

Code-build - build the java app specifications:-

- Code-build- the code will be build in containers 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.

Project configuration

Edit

Name
java-codestar-e

Description
AWS CodeStar created CodeBuild Project for java-codestar-e

Project ARN
arn:aws:codebuild:us-east-1:536285340728:project/java-codestar-e

Tags

Source

Edit

Source provider	Source identifier	Repository	Source version
AWS CodePipeline	-	-	-
Git clone depth	Git submodules		
Full	False		

- build environment, this java-spring application will be build in a container(remote) with specification image, environment, compute etc.

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-java-codestar-e-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-java-codestar-e-pipe	PLAINTEXT
WEBSITE_S3_PREFIX	NoVal	PLAINTEXT
WEBSITE_S3_BUCKET	NoVal	PLAINTEXT
PROJECT_ID	java-codestar-e	PLAINTEXT
ACCOUNT_ID	536285340728	PLAINTEXT
PARTITION	aws	PLAINTEXT

- 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.

Buildspec

Edit

Using the buildspec.yml in the source code root directory

Artifacts

Edit

Artifact identifier	Artifacts upload location	Packaging
-	-	Zip
Cache type	Cache location	Encryption key
No cache	-	arn:aws:kms:us-east-1:536285340728:alias/aws/s3

Logs

Edit

CloudWatch logs	CloudWatch group name	CloudWatch stream name
-	-	-
S3 logs	S3 location	Encryption disabled
-	-	-

- build steps is mentioned in the builspec.yml.
- This application is build with maven tool and sonar code quality. give the following commands to run.


```

version: 0.2

phases:
  install:
    runtime-versions:
      java: openjdk8
    commands:
      # Upgrade AWS CLI to the latest version
      - pip install --upgrade awscli
  pre_build:
    commands:
      - mvn clean compile test
  build:
    commands:
      - mvn package
      - mvn sonar:sonar
-Dsonar.host.url=http://novartis.devops.altimetrik.io:9000
-Dsonar.login=admin -Dsonar.password=admin

  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:
      - 'appspec.yml'
      - 'template.yml'
      - 'scripts/*'
      - 'target/ROOT.war'
      - 'template-configuration.json'

```

- lets look the code maven commands passing for build, test, package, code quality and archiving the files.

```

<project xmlns="http://maven.apache.org/POM/4.0.0"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://maven.apache.org/POM/4.0.0
http://maven.apache.org/maven-v4_0_0.xsd">
  <modelVersion>4.0.0</modelVersion>
  <groupId>com.aws.codestar.projecttemplates</groupId>

```

```

<artifactId>HelloWorld</artifactId>
<version>1.0</version>
<packaging>war</packaging>
<name>Sample Spring MVC Application Using AWS CodeStar</name>
<properties>
    <junit.platform.version>1.2.0</junit.platform.version>
    <junit.jupiter.version>5.2.0</junit.jupiter.version>
</properties>
<dependencies>
    <dependency>
        <groupId>org.springframework</groupId>
        <artifactId>spring-context</artifactId>
        <version>5.0.7.RELEASE</version>
    </dependency>
</dependencies>
<build>
    <pluginManagement>
        <plugins>
            <plugin>
                <groupId>org.apache.maven.plugins</groupId>
                <artifactId>maven-war-plugin</artifactId>
                <version>3.2.2</version>
                <configuration>

<warSourceDirectory>src/main/webapp</warSourceDirectory>
                <warName>ROOT</warName>
                <failOnMissingWebXml>false</failOnMissingWebXml>

            </plugin>
            <plugin>
                <groupId>org.apache.maven.plugins</groupId>
                <artifactId>maven-surefire-plugin</artifactId>
                <version>2.22.0</version>
                <dependencies>
                    <dependency>
                        <groupId>org.junit.platform</groupId>

<artifactId>junit-platform-surefire-provider</artifactId>
                        <version>${junit.platform.version}</version>

                    </dependency>
                </dependencies>
            </plugin>
        </plugins>
    </pluginManagement>

```

```

        <finalName>ROOT</finalName>
    </build>
</project>

```

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

sonarqube | Projects | Issues | Rules | Quality Profiles | Quality Gates | Administration

Search for projects, sub-projects and files...

Perspective: Overall Status | Sort by: Last analysis date | Search by project name or key | 32 projects

My Favorites | All

Filters

Quality Gate

Passed 32 | Warning 0 | Failed 0

☆ Sample Spring MVC Application Using AWS CodeStar **Passed**

Last analysis: November 14, 2019, 1:04 PM

0 **A** Bugs | 0 **A** Vulnerabilities | 2 **A** Code Smells | 0.0% Coverage | 0.0% Duplications

200 XS XML, Java, ...

sonarqube | Projects | Issues | Rules | Quality Profiles | Quality Gates | Administration

Search for projects, sub-projects and files...

☆ Sample Spring MVC Application ... | master

November 14, 2019, 1:04 PM | Version 1.0

Overview | Issues | Measures | Code | Activity | Administration

Quality Gate **Passed**

Bugs | Vulnerabilities

0 **A** Bugs | 0 **A** Vulnerabilities

Code Smells

45min **A** Debt | 2 Code Smells

started 1 minute ago

Coverage

0.0% Coverage | 1 Unit Tests

XS 200 Lines of Code | XML 104 | Java 93 | JavaScript 3

No tags

Activity

November 14, 2019 | 1.0 | Show More

Quality Gate (Default) Novartis

Quality Profiles (Java) Sonar way | (JavaScript) Sonar way | (XML) Sonar way

Key com.aws.codestar.projecttemplates:HelloW

- you can able to see the build details and builds history.

java-codestar-e

Notify Edit Delete build project Start build

Configuration

Source provider AWS CodePipeline	Primary repository -	Artifacts upload location -	Build badge Disabled
-------------------------------------	-------------------------	--------------------------------	-------------------------

Build history Build details Build triggers Metrics

Build history

Stop build View artifacts View logs Delete builds Retry build

Build run	Status	Build Number	Source version	Submitter	Duration	Completed
java-codestar-e-dd5e0553-faf7-4137-a00d-e820b8774842	Succeeded	4	arn:aws:s3::aws-codestar-us-east-1-536285340728-java-codestar-e-pipe/java-codestar-e-pipe/java-codes/kOWHhNb	codepipeline/java-codestar-e-Pipeline	1 minute 12 seconds	4 days ago
java-codestar-e-2fb5a5c7-0d70-48cc-b13e-c1d6d41d4724	Succeeded	3	arn:aws:s3::aws-codestar-us-east-1-536285340728-java-codestar-e-pipe/java-codestar-e-pipe/java-codes/g78yn5Y	codepipeline/java-codestar-e-Pipeline	1 minute 12 seconds	4 days ago

Developer Tools > CodeBuild > Build projects > java-codestar-e > java-codestar-e-dd5e0553-faf7-4137-a00d-e820b8774842

java-codestar-e-dd5e0553-faf7-4137-a00d-e820b8774842

Stop build Retry build

Build status

Status Succeeded	Initiator codepipeline/java-codestar-e-Pipeline	Build ARN arn:aws:codebuild:us-east-1:536285340728:build/java-codestar-e-dd5e0553-faf7-4137-a00d-e820b8774842	Resolved source version c34a55d229220a24e30c7ee73d01dc2122e10d72
Start time Nov 14, 2019 2:37 PM	End time Nov 14, 2019 2:38 PM	Build Number 4	

Build logs Phase details Environment variables Build details

Showing the last 1000 lines of the build log. View entire log Tail logs

Show previous logs

```

1 Downloading from central: https://repo.maven.apache.org/maven2/org/codehaus/plexus/plexus-interpolation/1.12/plexus-interpolation-1.12.jar
2 Progress (1): 889 B
3
4 Downloaded from central: https://repo.maven.apache.org/maven2/org/codehaus/plexus/plexus-interpolation/1.12/plexus-interpolation-1.12.jar (889 B at 99 kB/s)
5 Downloading from central: https://repo.maven.apache.org/maven2/org/codehaus/plexus/plexus-components/1.1.14/plexus-components-1.1.14.jar
6 Progress (1): 2.2/5.8 kB
7 Progress (1): 5.0/5.8 kB
8 Progress (1): 5.8 kB
9
10 Downloaded from central: https://repo.maven.apache.org/maven2/org/codehaus/plexus/plexus-components/1.1.14/plexus-components-1.1.14.jar (5.8 kB at 1.1 MB/s)

```

© 2008 - 2019, Amazon Web Services, Inc. or its affiliates. All rights reserved. Privacy Policy Terms of

- Artifacts will be store in a S3 bucket configure by cloud formation see the codestar project resources.

Code deploy- configurations:

- lets go to code deploy configurations, deployment group details i.e compute platform

Developer Tools > CodeDeploy > Applications > java-codestar-e

java-codestar-e

[Notify](#) [Delete application](#)

Application details

Name	Compute platform
java-codestar-e	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
<input type="radio"/>	java-codestar-e-Env	✓ Succeeded	Nov 14, 2019 2:40 PM	Nov 14, 2019 2:40 PM	0

- deployment group details- compute platform(tag of cloudformation ec2).

Developer Tools > CodeDeploy > Applications > java-codestar-e > java-codestar-e-Env

java-codestar-e-Env

[Edit](#) [Delete](#) [Create deployment](#)

Deployment group details

Deployment group name	Application name	Compute platform
java-codestar-e-Env	java-codestar-e	EC2/On-premises
Deployment type	Service role ARN	Deployment configuration
In-place	arn:aws:iam::536285340728:role/CodeStarWorker-java-codestar-e-ToolChain	CodeDeployDefault.OneAtATime
Rollback enabled		
False		

Environment configuration: Amazon EC2 instances

Key	Value
Environment	java-codestar-e-WebApp

Triggers

Name	Events	Type
------	--------	------

- deployment revisions are taking from s3 bucket (archived files of build).

Deployment group deployment history							View details	Actions ▾
<input type="text" value="Q"/>								
	Deployment Id	Status	Revision location	Initiating event	Start time			
<input type="radio"/>	d-D055FIXW0	Succeeded	s3://aws-codestar-us-east-1-53...	User action	Nov 14, 2019 2:39 PM			
<input type="radio"/>	d-ZU35EN9PD	Succeeded	s3://aws-codestar-us-east-1-53...	User action	Nov 14, 2019 1:07 PM			
<input type="radio"/>	d-CLHNP2UW0	Succeeded	s3://aws-codestar-us-east-1-53...	User action	Nov 14, 2019 12:09 PM			
<input type="radio"/>	d-N2KQIVAW0	Succeeded	s3://aws-codestar-us-east-1-53...	User action	Nov 13, 2019 4:02 PM			

- Go to each deployment id, you can see the configurations.

d-D055FIXW0

Deployment status

Installing application on your instances

1 of 1 instances updated

Succeeded

Deployment details

Application java-codestar-e	Deployment ID d-D055FIXW0	Status Succeeded
Deployment configuration CodeDeployDefault.OneAtATime	Deployment group java-codestar-e-Env	Initiated by User action
Deployment description -		

Revision details

Revision location s3://aws-codestar-us-east-1-536285340728-java-codestar-e-pipe/java-codestar-e-Pipe/java-codes/aoCsyA3?versionId=cdvhWBgrdS1THfTA8nogv2ZDsdq5GZPY&eTag=b5d0bb2fb0a7f5e42b3ca01e7d58d742-2	Revision created 4 days ago	Revision description Application revision registered by Deployment ID: d-D055FIXW0
---	--------------------------------	---

Code pipeline- configurations:

- Go to code star project code-pipeline see the stages.

Developer Tools > CodePipeline > Pipelines > java-codestar-e-Pipeline

java-codestar-e-Pipeline

[Notify](#) [Edit](#) [Clone pipeline](#) [View history](#) [Release change](#)

Source

[View current revisions](#)

ApplicationSource

AWS CodeCommit

✓ Succeeded - 4 days ago

c34a55d2

c34a55d2 [ApplicationSource: adding sonar](#)

[Disable transition](#)

Build

[View current revisions](#)

PackageExport

AWS CodeBuild

✓ Succeeded - 4 days ago

[Details](#)

c34a55d2 [ApplicationSource: adding sonar](#)

Deploy

[View current revisions](#)

GenerateChangeSet

AWS CloudFormation

✓ Succeeded - 4 days ago

[Details](#)

↓

ExecuteChangeSet

AWS CloudFormation

✓ Succeeded - 4 days ago

[Details](#)

↓

Deploy

AWS CodeDeploy

✓ Succeeded - 4 days ago

[Details](#)

c34a55d2 [ApplicationSource: adding sonar](#)

- you can click each stage, it will be redirect.
- AWS cloudformation for creating ec2 instance.

pipeline-changeset

Overview

Change set ID: `arn:aws:cloudformation:us-east-1:536285340728:changeSet/pipeline-changeset/2c931299-bdbf-43d1-ae04-45f1e904b9bf`

Description:

Created time: 2019-11-14 14:38:39 UTC+0550

Status: **CREATE_COMPLETE**

Stack name: `awscodestar-java-codestar-e-infrastructure`

Change set input

Changes

The changes CloudFormation will make if you execute this change set.

▼ Filter				Viewing 2
Action	Logical ID	Physical ID	Resource Type	
Remove	SyncResources1573716912950	java-codestar-e	AWS::CodeStar::SyncResources	
Add	SyncResources1573722520830		AWS::CodeStar::SyncResources	

awscodestar-java-codestar-e-infrastructure

Other Actions Update Stack

Stack name: `awscodestar-java-codestar-e-infrastructure`

Stack ID: `arn:aws:cloudformation:us-east-1:536285340728:stack/awscodestar-java-codestar-e-infrastructure/a01a8280-0600-11ea-9ccf-0aff04b5e65f`

Status: **UPDATE_COMPLETE**

Status reason:

Termination protection: Disabled

Drift status: **NOT_CHECKED** [View details](#)

Last drift check time:

IAM role: `CodeStarWorker-java-codestar-e-CloudFormation (arn:aws:iam:536285340728:role/CodeStarWorker-java-codestar-e-CloudFormation)`

Description

Outputs

Resources

To view detailed drift information for specific resources, visit the [Drift Details page](#).

Logical ID	Physical ID	Type	Drift Status	Status	Status Reason
SyncResources157372252...	java-codestar-e	AWS::CodeStar::SyncResources	NOT_CHECKED	CREATE_COMPLETE	
WebApp01	i-058df463c533c4e5	AWS::EC2::Instance	NOT_CHECKED	CREATE_COMPLETE	
WebAppSG	sg-050bf849e4230a47d	AWS::EC2::SecurityGroup	NOT_CHECKED	CREATE_COMPLETE	

- 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 instances, redirect console and try to login to the instance.

Launch Instance

Connect

Actions

Instance ID: i-05fd4633c533c4e5

Add filter

1 to 1 of 1

Name	Instance ID	Instance Type	Availability Zone	Instance State	Status Checks	Alarm Status	Public DNS (IPv4)
java-codestar-e-WebApp	i-05fd4633c533c4e5	t2.micro	us-east-1d	running	2/2 checks ...	None	ec2-34-229-174-2

root@ip-172-31-24-125:/opt/tomcat/webapps

```
[root@ip-172-31-24-125 tomcat]# ls
bin BUILDING.txt conf CONTRIBUTING.md lib LICENSE logs NOTICE README.md RELEASE-NOTES RUNNING.txt temp webapps work
[root@ip-172-31-24-125 tomcat]# cd webapps/
[root@ip-172-31-24-125 webapps]# ls
docs examples host-manager manager ROOT ROOT.war
[root@ip-172-31-24-125 webapps]#
```

ec2-user@ip-172-31-24-125:/opt/tomcat

```
bin BUILDING.txt conf CONTRIBUTING.md lib LICENSE logs NOTICE README.md RELEASE-NOTES RUNNING.txt temp webapps work
[root@ip-172-31-24-125 tomcat]# cd webapps/
[root@ip-172-31-24-125 webapps]# ls
docs examples host-manager manager ROOT ROOT.war
[root@ip-172-31-24-125 webapps]# cd ..
[root@ip-172-31-24-125 tomcat]# ls
bin BUILDING.txt conf CONTRIBUTING.md lib LICENSE logs NOTICE README.md RELEASE-NOTES RUNNING.txt temp webapps work
[root@ip-172-31-24-125 tomcat]# cd ..
[root@ip-172-31-24-125 opt]# ls
aws codedeploy-agent tomcat
[root@ip-172-31-24-125 opt]# cd aws/
[root@ip-172-31-24-125 aws]# ls
bash: ls: command not found
[root@ip-172-31-24-125 aws]# ls
awscli apitools bin credential-file-path.template
[root@ip-172-31-24-125 aws]# cd ..
[root@ip-172-31-24-125 opt]# ls
aws codedeploy-agent tomcat
[root@ip-172-31-24-125 opt]# cd codedeploy-agent/
[root@ip-172-31-24-125 codedeploy-agent]# ls
bin certs codedeploy-agent-1.1.0.gemspec deployment-root Gemfile lib LICENSE state vendor
[root@ip-172-31-24-125 codedeploy-agent]# cd deployment-root/
[root@ip-172-31-24-125 deployment-root]# ls
d6eb1a1d-8d0d-4f0d-8050-67b03329c1d1 deployment-instructions deployment-logs ongoing-deployment
[root@ip-172-31-24-125 deployment-root]# ls -al
total 24
drwxr-xr-x 6 root root 4096 Nov 13 10:32 .
drwxr-xr-x 8 root root 4096 Nov 13 10:32 ..
drwxr-xr-x 4 root root 4096 Nov 14 06:39 d6eb1a1d-8d0d-4f0d-8050-67b03329c1d1
drwxr-xr-x 2 root root 4096 Nov 14 06:39 deployment-instructions
drwxr-xr-x 2 root root 4096 Nov 13 10:32 deployment-logs
drwxr-xr-x 2 root root 4096 Nov 14 06:39 ongoing-deployment
[root@ip-172-31-24-125 deployment-root]# cd d6eb1a1d-8d0d-4f0d-8050-67b03329c1d1/
[root@ip-172-31-24-125 d6eb1a1d-8d0d-4f0d-8050-67b03329c1d1]# ls
d-CLNNP2UW0 d-N2KQIVAW0
[root@ip-172-31-24-125 d6eb1a1d-8d0d-4f0d-8050-67b03329c1d1]# cd d-N2KQIVAW0/
[root@ip-172-31-24-125 d-N2KQIVAW0]# ls
bash: ls: command not found
[root@ip-172-31-24-125 d-N2KQIVAW0]# ls
bundle.tar deployment-archive logs
[root@ip-172-31-24-125 d-N2KQIVAW0]# cd ..
[root@ip-172-31-24-125 d6eb1a1d-8d0d-4f0d-8050-67b03329c1d1]# exit
```

- Here we can observe the tomcat web-apps root directory, that is the war file of what we accessing the application. we can debug 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.

Amazon S3 > aws-codestar-us-east-1-536285340728-java-codestar-e-pipe > java-codestar-e-Pipe > java-codes

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 8

Name	Last modified	Size	Storage class
0otTFuF	Nov 13, 2019 3:58:51 PM GMT+0530	15.2 KB	Standard
Gi7QsHU	Nov 13, 2019 3:59:54 PM GMT+0530	5.1 MB	Standard
VZyund3	Nov 14, 2019 12:05:45 PM GMT+0530	15.2 KB	Standard
Z8oy0S6	Nov 14, 2019 1:04:45 PM GMT+0530	5.1 MB	Standard
aoCeyA3	Nov 14, 2019 2:38:12 PM GMT+0530	5.1 MB	Standard
g7Bys5Y	Nov 14, 2019 1:03:31 PM GMT+0530	15.3 KB	Standard
kOWWhitb	Nov 14, 2019 2:36:56 PM GMT+0530	15.3 KB	Standard
v8Q2Pd9	Nov 14, 2019 12:06:52 PM GMT+0530	5.1 MB	Standard

Viewing 1 to 8

- For any role based access go to IAM configurations, debug if any issues faced.

Identity and Access Management (IAM)

Roles > CodeStarWorker-java-codestar-e-ToolChain

Summary

Role ARN: arn:aws:iam::536285340728:role/CodeStarWorker-java-codestar-e-ToolChain

Role description: Edit

Instance Profile ARNs:

Path: /

Creation time: 2019-11-13 15:57 UTC+0530

Maximum CLI/API session duration: 1 hour Edit

Permissions Trust relationships Tags Access Advisor Revoke sessions

Permissions policies (9 policies applied)

Attach policies Add inline policy

Policy name	Policy type
AWSCodeCommitFullAccess	AWS managed policy
AWSLambdaFullAccess	AWS managed policy

Show 7 more

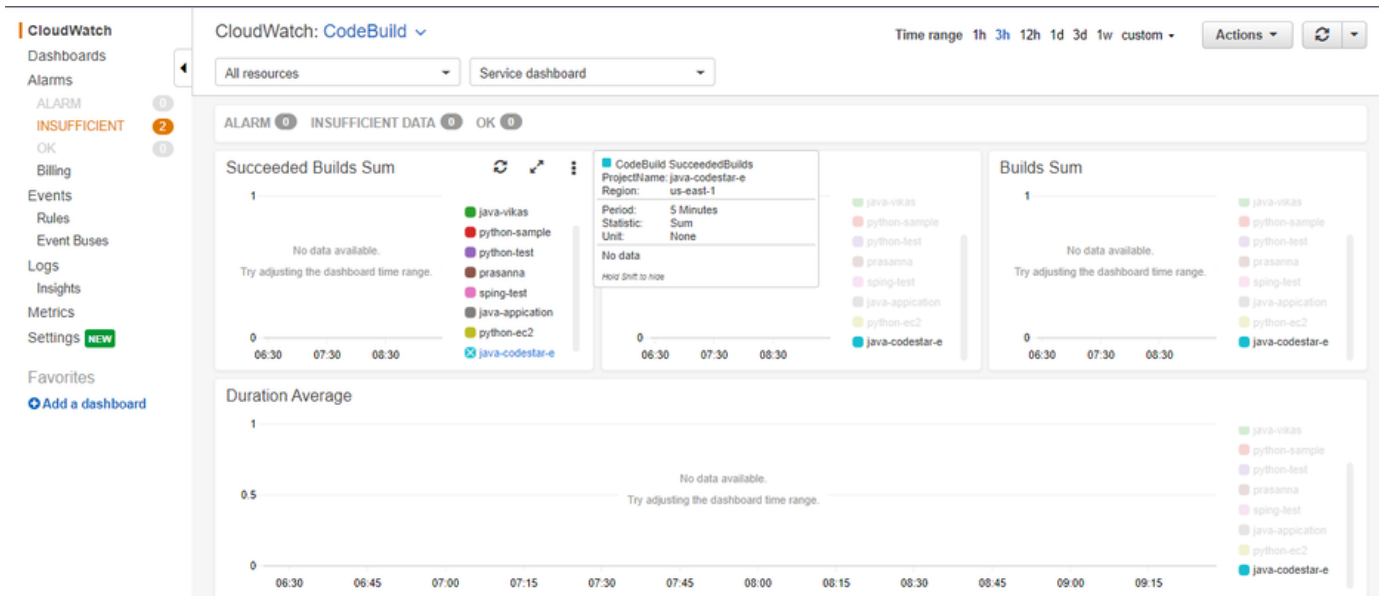
Permissions boundary (set)

Set a permissions boundary to control the maximum permissions this role can have. This is not a common setting but can be used to delegate permission management to others. [Learn more](#)

Change boundary Remove boundary

CodeStar_java-codestar-e_PermissionsBoundary (Managed policy)

- go to cloudwatch for monitoring the code-build.



that done !!!

whats next ???

lets try the java-spring application in fargate and python django too.