Deployment of Java Applications on AWS EC2 instances

Using CodePipeline and CodeDeploy

Prerequisites:

Step1: Upload a sample application to your GitHub repository

Step2: Provision EC2 instances

Step3: Create an application and deployment group in CodeDeploy

Step4: Create a pipeline using CodePipeline

1. **Upload application to GitHub repository**

Push application to repository from a specific branch. Files to added to your repository for CodeDeploy and CodePipeline are appspec.yml and other scripts files to install dependencies.

Reference appspec.yml

version: 0.0

os: linux

files:

  - source: /

    destination: /home/ubuntu/

hooks:

  BeforeInstall:

    - location: /install\_dependencies.sh

      timeout: 300

      runas: root

    - location: /build.sh

      timeout: 300

      runas: root

    - location: /start.sh

      timeout: 300

      runas: root

1. **Provision EC2 instances**

Launch EC2 instances for your application deployment

* Tag the instance

Graphical user interface, text, application

Description automatically generated

* Select the AMI image for your instance

Graphical user interface, application

Description automatically generated

* Create a role for EC2 by attaching policy “AmazoneEC2RoleforAWSCodeDeploy” so that CodeDeploy-Agent can access the instance.

Graphical user interface, text, application

Description automatically generated

* Attach the role to the instance

Graphical user interface, text, application, email, Teams

Description automatically generated

* Create a security group with inbound rules which opens ports 22 for SSH and 8080 for application.

Graphical user interface, text, application, email

Description automatically generated

Graphical user interface, application

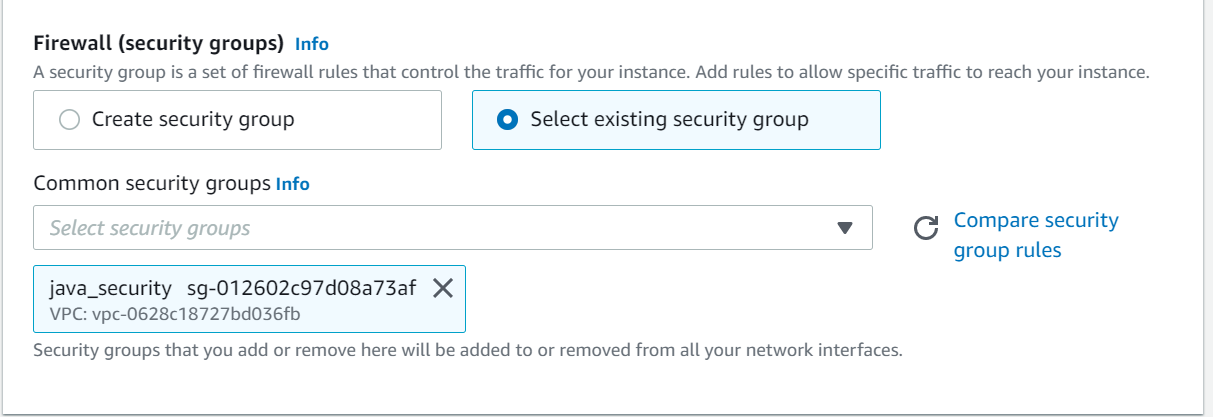
Description automatically generated

* Attach key pair login for SSH onto the EC2 instance

Graphical user interface, text, application, email

Description automatically generated

* Attach the security group for the instance



* Add user data as below

Graphical user interface, text, application, email

Description automatically generated

Things to modify for above script

Here bucket-name is the name of the Amazon S3 bucket that contains the CodeDeploy Resource Kit files for your region, region-identifier is the identifier of your region

For example: if your region is Asia pacific (Mumbai)

Replace bucket-name with aws-codedeploy-ap-south-2 and region-identifier with ap-south-2.

Finally, your link will be

<https://aws-codedeploy-ap-south-1.s3.ap-south-1.amazonaws.com/latest/install>

1. **Creating deployment and deployment group**

* Create a role for CodeDeploy with Policy attached as AWSCodeDeployRole

Graphical user interface, text, application, email

Description automatically generated

* Create deployment application with Compute platform as EC2/on-premises

Graphical user interface, text, application, email, Teams

Description automatically generated

* Create a deployment group

Graphical user interface, application

Description automatically generated

* Provide Deployment Group name and attach service role created for AWS CodeDeploy

Graphical user interface, text, application, email, Teams

Description automatically generated

* Choose the Deployment type either **in-place** or **blue/green.**

Graphical user interface, text, application

Description automatically generated

* Select the Environment Configuration as **Amazon EC2 instances** and add tags of your instances so that CodeDeploy deploys application on the tagged instances.

Graphical user interface, text, application, email

Description automatically generated

* Select Agent Configuration so that CodeDeploy will install CodeDeploy-Agent on all instances.

Graphical user interface, text, application, email

Description automatically generated

* Configure the deployment by choosing how fast your application is to deploy on instances. Basically, go with Default.DeployAllAtOnce.

Graphical user interface, text, application, Teams

Description automatically generated

* If required assign Load Balancer either Application or Classic Load Balancer.

Graphical user interface, application

Description automatically generated

* Also, can add triggers when a deployment starts or fails.

Graphical user interface, text, application, email, Teams

Description automatically generated

1. **Creating CodePipeline**

* Create pipeline with role attached or can choose option create role, so that a new role will be created and assigned to the pipeline. Do not forget to tick the below checkbox which permits role creation for CodePipeline.

Graphical user interface, text, application, email

Description automatically generated

Graphical user interface, text, application, email

Description automatically generated

* choose artifact storage either as default location or AWS S3 bucket.

Graphical user interface, application, Teams

Description automatically generated

* Next add source code configuration for our deployment you can choose your source code. connect to your source code through login prompt page and grant authorization access.

Graphical user interface, text, application, email

Description automatically generated

Graphical user interface, text, application

Description automatically generated

* Connect to your source code through login prompt page and grant authorization access.

Graphical user interface, text, application, email

Description automatically generated

* Select the repository and branch of your source code
* Choose the GitHub webhooks or AWS CodePipeline to check periodically changes in the repository.

Graphical user interface, text, application, email

Description automatically generated

* Next choose build stage or can skip it.

Graphical user interface, text, application, email

Description automatically generated

* Next add deployment stage by choosing AWS CodeDeploy, region, Application name and Deployment Group.

A picture containing text

Description automatically generated

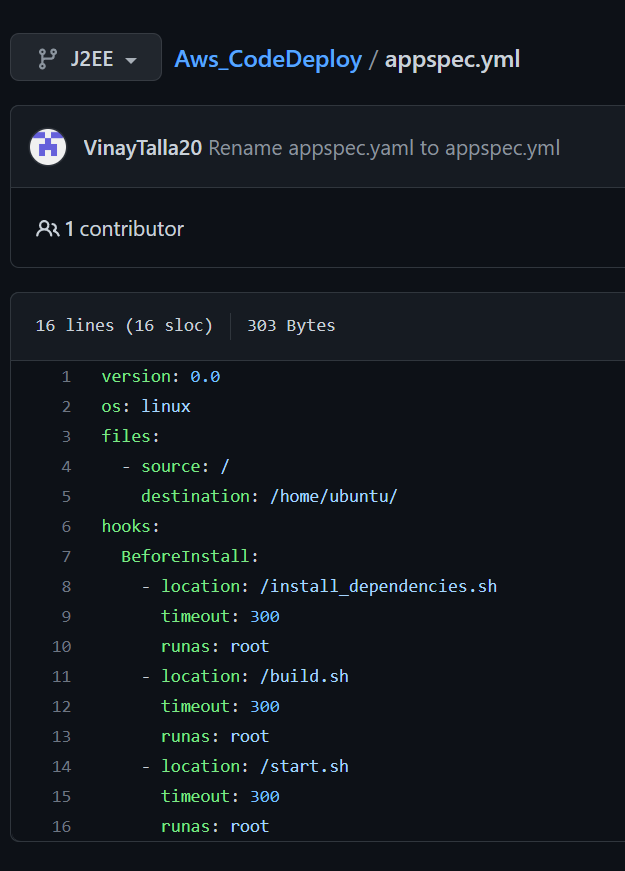
* Review the changes and confirm to create pipeline.

Graphical user interface, application, Teams

Description automatically generated

Pre-configuration files needed for AWS CodePipeline to run set of sequence commands will be provided in appspec.yml

* Add appspec.yml file to your root location of repository



AWS CodePipeline will initiate the hooks as per the sequence

Scripts files for reference

Text

Description automatically generated

Text

Description automatically generated

A screenshot of a computer

Description automatically generated with medium confidence