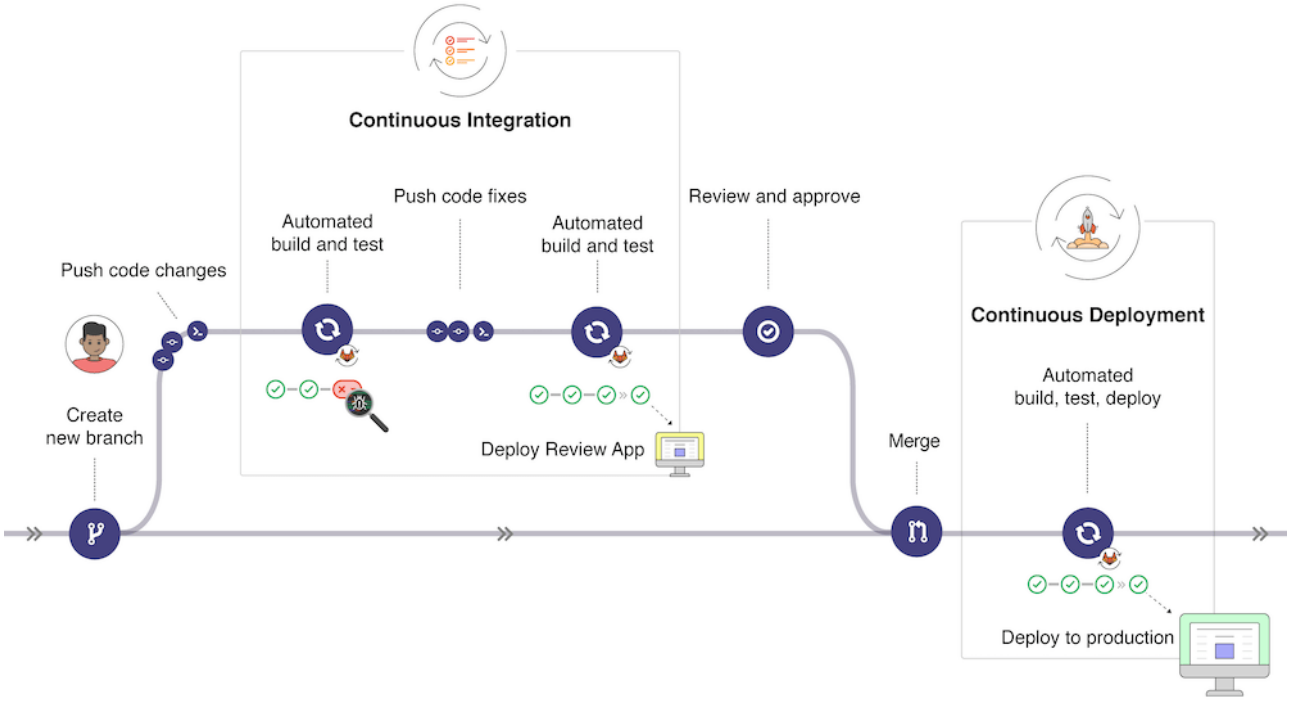
Introduction

Once we have developed and deployed the application/product, It needs to be continuously updated based on user feedback or the addition of new features. This process should be automated, as without automation we have to run the same development and deployment



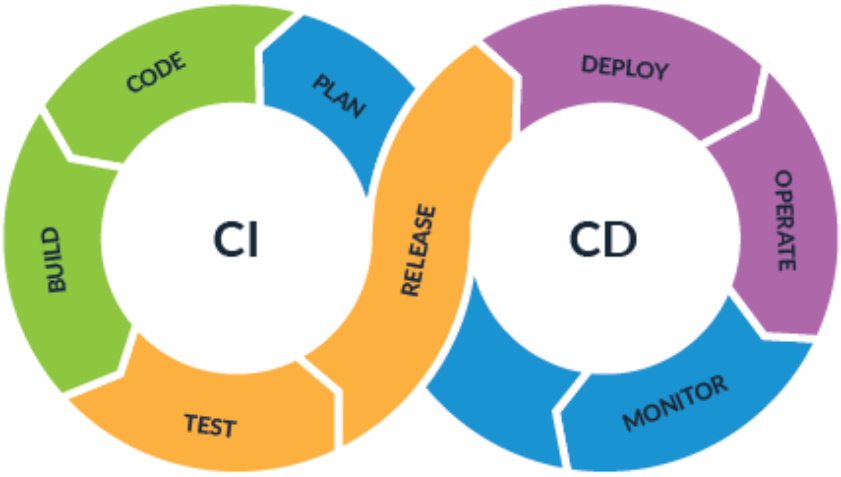
AWS CodeBuild

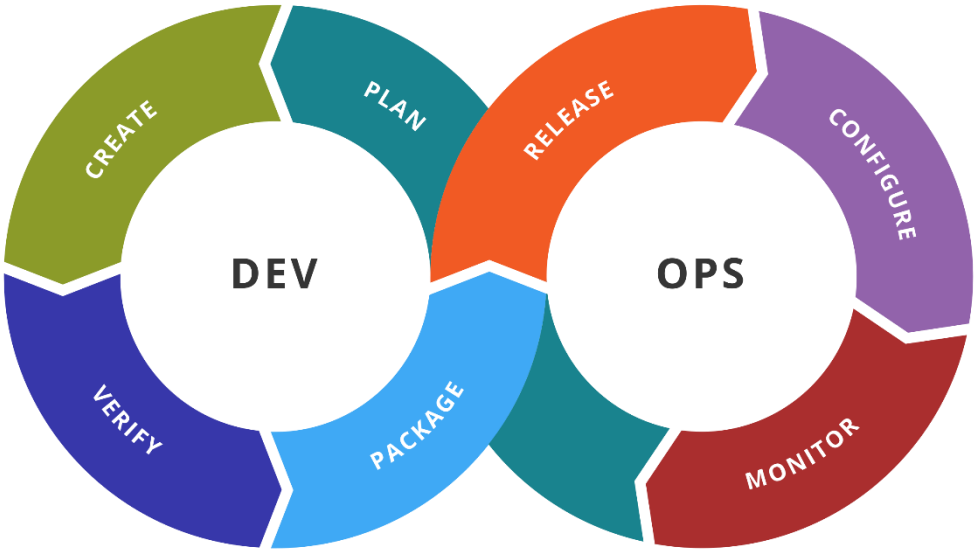
AWS CodeBuild is managed CI service which complies source code,runs tests, and packages the source code which can be used for deployment.

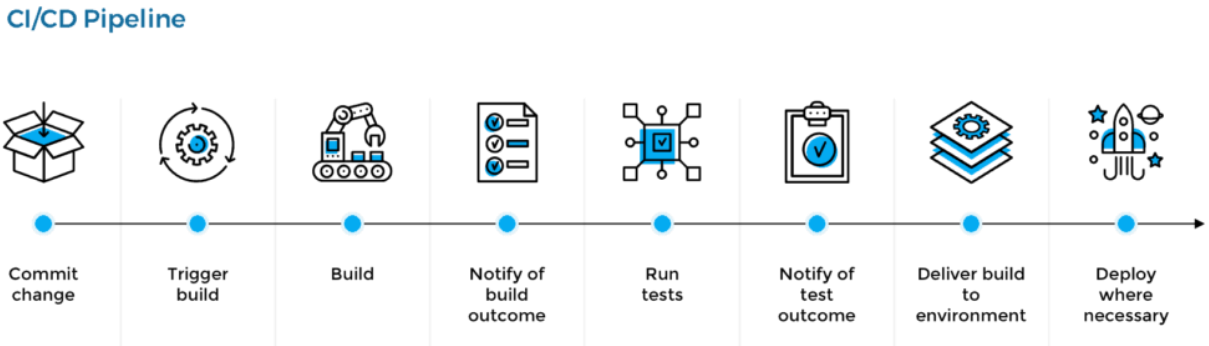
AWS CodePipeline

CodePipeline will automatically build and deploy our application when there are any changes in the code repository.

With Continuous Integration and Continuous Delivery Pipeline, we can automate the complete workflow from building, testing, packaging, and deploying, which will be triggered when there are any changes to an existing application or we can say if there is any new commit to an existing code repository.




CREATE
PLAN
RELEASE
CONFIGURE
DEV
OPS
VERIFY
PACKAGE
MONITOR


CI/CD Pipeline
Commit change
Trigger build
Build
Notify of build outcome
Run tests
Notify of test outcome
Deliver build to environment
Deploy where necessary

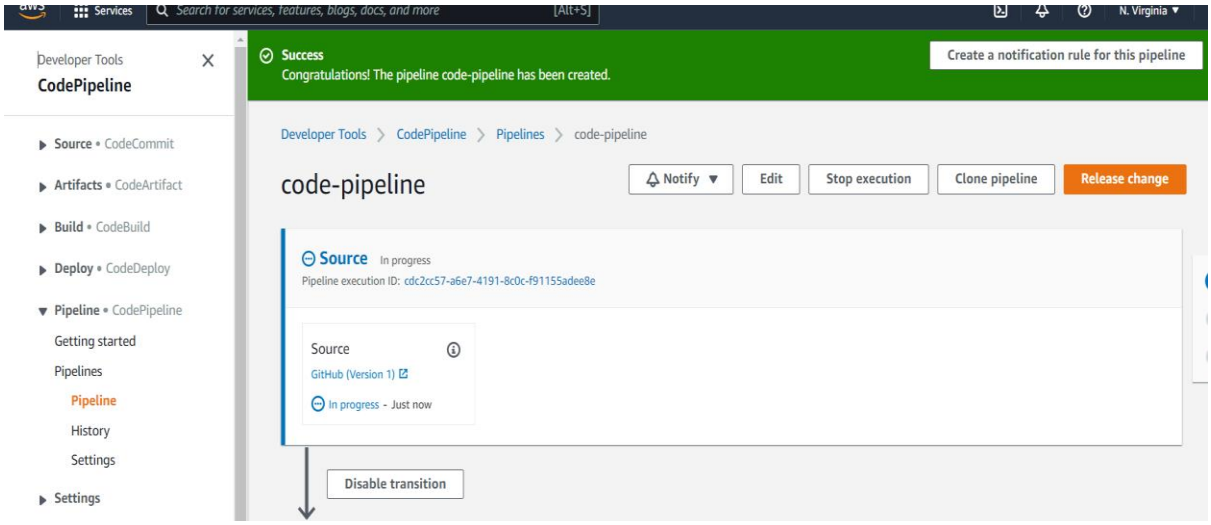
Step1 Choose pipeline setting

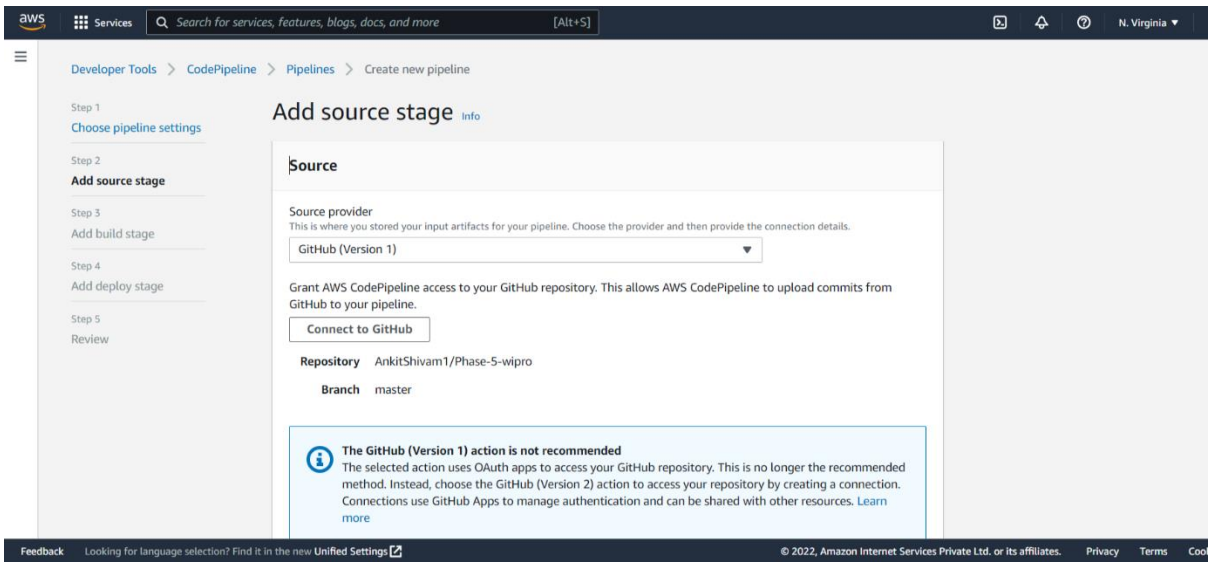
Step2 Add source stage

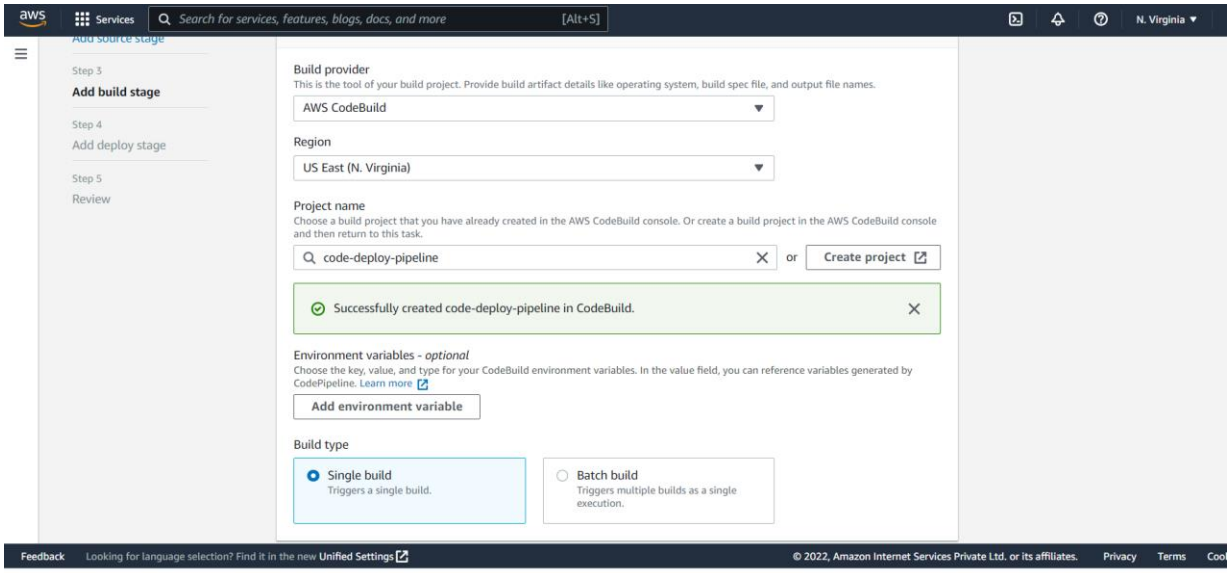
Step3 Add build stage

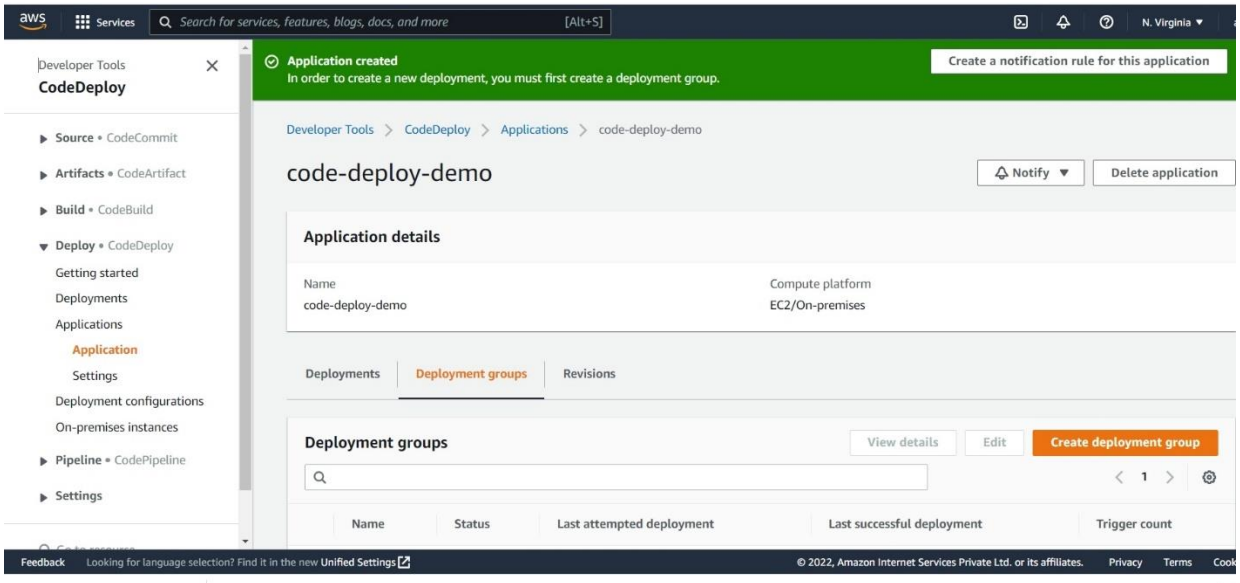
Step4 Add deploy stage

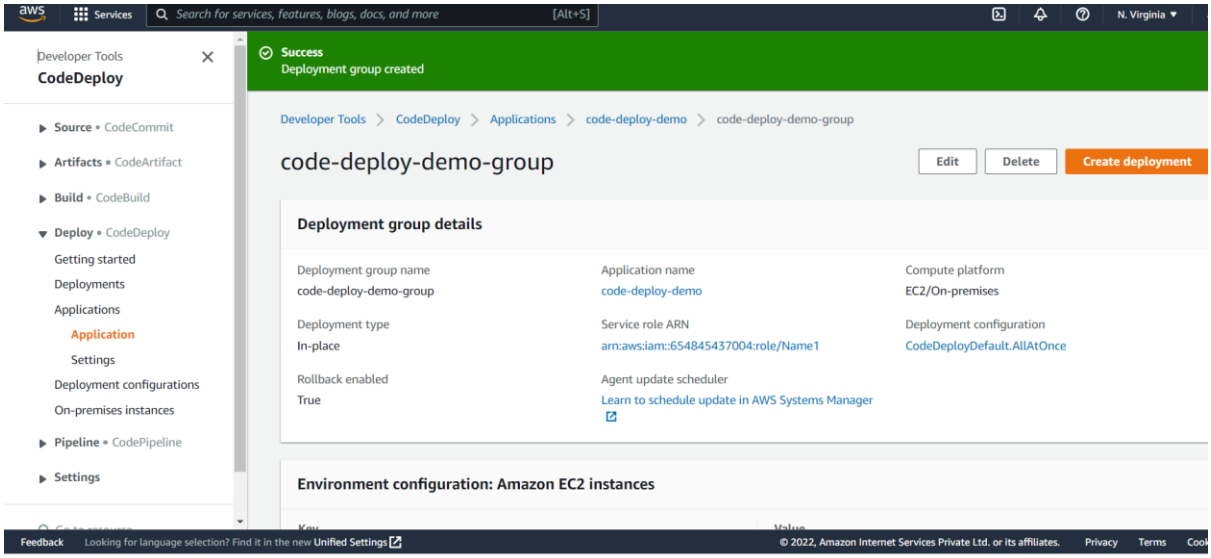
Step5 Review

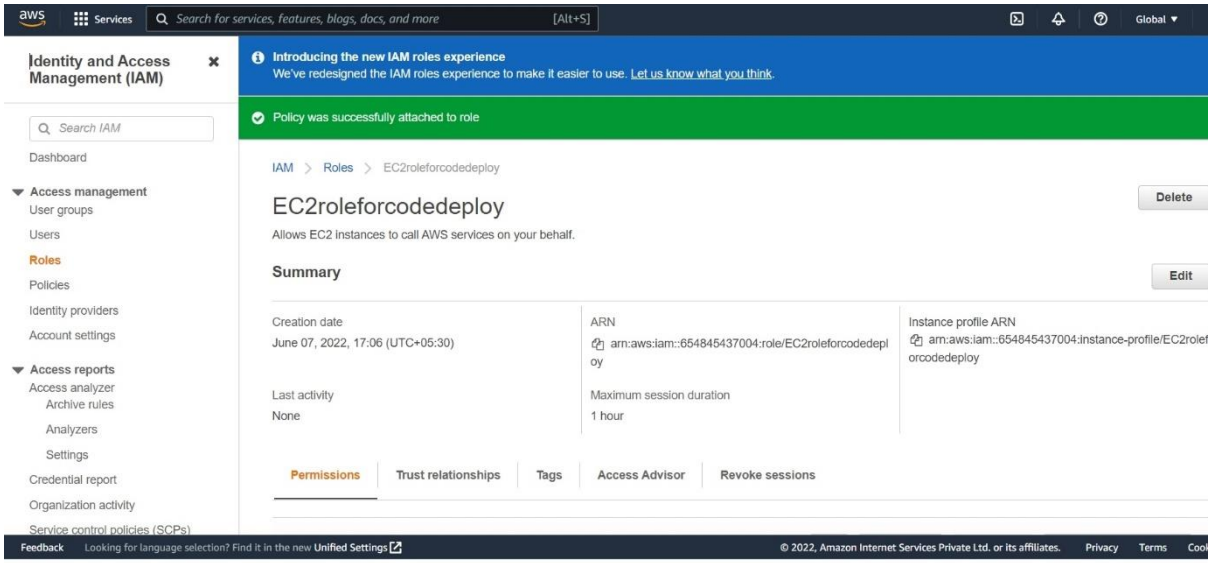


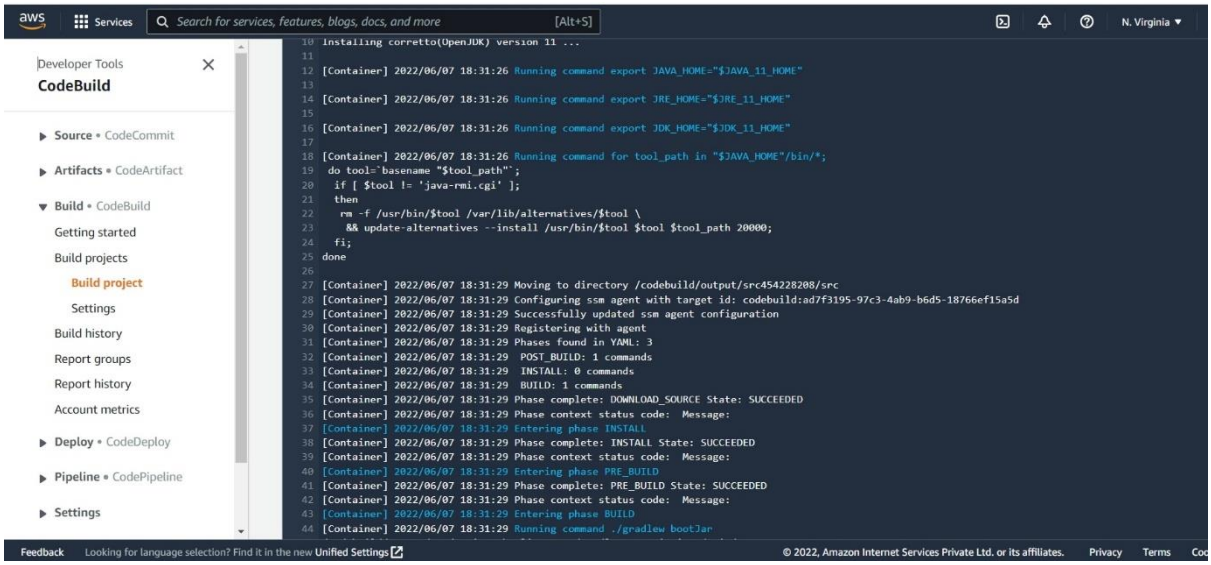



aws
Services
Search for services, features, blogs, docs, and more
[Alt+S]
N. Virginia
Step 3
Add build stage
Step 4
Add deploy stage
Step 5
Review
Build provider
This is the tool of your build project. Provide build artifact details like operating system, build spec file, and output file names.
AWS CodeBuild
Region
US East (N. Virginia)
Project name
Choose a build project that you have already created in the AWS CodeBuild console. Or create a build project in the AWS CodeBuild console and then return to this task.
code-deploy-pipeline
or
Create project
Successfully created code-deploy-pipeline in CodeBuild.
Environment variables - optional
Choose the key, value, and type for your CodeBuild environment variables. In the value field, you can reference variables generated by CodePipeline. Learn more
Add environment variable
Build type
Single build
Triggers a single build.
Batch build
Triggers multiple builds as a single execution.
Feedback
Looking for language selection? Find it in the new Unified Settings
© 2022, Amazon Internet Services Private Ltd. or its affiliates.
Privacy
Terms


aws
Services
Search for services, features, blogs, docs, and more
[Alt+S]
N. Virginia
Developer Tools
CodeDeploy
Source • CodeCommit
Artifacts • CodeArtifact
Build • CodeBuild
Deploy • CodeDeploy
Getting started
Deployments
Applications
Application
Settings
Deployment configurations
On-premises instances
Pipeline • CodePipeline
Settings
Application created
In order to create a new deployment, you must first create a deployment group.
Create a notification rule for this application
Developer Tools > CodeDeploy > Applications > code-deploy-demo
code-deploy-demo
Notify
Delete application
Application details
Name
code-deploy-demo
Compute platform
EC2/On-premises
Deployments
Deployment groups
Revisions
Deployment groups
View details
Edit
Create deployment group
Name
Status
Last attempted deployment
Last successful deployment
Trigger count
Feedback
Looking for language selection? Find it in the new Unified Settings
© 2022, Amazon Internet Services Private Ltd. or its affiliates.
Privacy
Terms


Services
Search for services, features, blogs, docs, and more
[Alt+S]
N. Virginia
Developer Tools
CodeDeploy
Source • CodeCommit
Artifacts • CodeArtifact
Build • CodeBuild
Deploy • CodeDeploy
Getting started
Deployments
Applications
Application
Settings
Deployment configurations
On-premises instances
Pipeline • CodePipeline
Settings
Success
Deployment group created
Developer Tools > CodeDeploy > Applications > code-deploy-demo > code-deploy-demo-group
code-deploy-demo-group
Edit
Delete
Create deployment
Deployment group details
Deployment group name
code-deploy-demo-group
Application name
code-deploy-demo
Compute platform
EC2/On-premises
Deployment type
In-place
Service role ARN
arn:aws:iam::654845437004:role/Name1
Deployment configuration
CodeDeployDefault.AllAtOnce
Rollback enabled
True
Agent update scheduler
Learn to schedule update in AWS Systems Manager
Environment configuration: Amazon EC2 instances
Feedback
Looking for language selection? Find it in the new Unified Settings
© 2022, Amazon Internet Services Private Ltd. or its affiliates.
Privacy
Terms


Services
Search for services, features, blogs, docs, and more
[Alt+S]
Global
Identity and Access Management (IAM)
Search IAM
Dashboard
Access management
User groups
Users
Roles
Policies
Identity providers
Account settings
Access reports
Access analyzer
Archive rules
Analyzers
Settings
Credential report
Organization activity
Service control policies (SCPs)
Introducing the new IAM roles experience
We've redesigned the IAM roles experience to make it easier to use. Let us know what you think.
Policy was successfully attached to role
IAM > Roles > EC2roleforcodedeploy
EC2roleforcodedeploy
Delete
Allows EC2 instances to call AWS services on your behalf.
Summary
Edit
Creation date
June 07, 2022, 17:06 (UTC+05:30)
ARN
arn:aws:iam::654845437004:role/EC2roleforcodedeploy
Instance profile ARN
arn:aws:iam::654845437004:instance-profile/EC2roleforcodedeploy
Last activity
None
Maximum session duration
1 hour
Permissions
Trust relationships
Tags
Access Advisor
Revoke sessions
Feedback
Looking for language selection? Find it in the new Unified Settings
© 2022, Amazon Internet Services Private Ltd. or its affiliates.
Privacy
Terms


aws
Services
Search for services, features, blogs, docs, and more
[Alt+S]
N. Virginia
Developer Tools
CodeBuild
Source • CodeCommit
Artifacts • CodeArtifact
Build • CodeBuild
Getting started
Build projects
Build project
Settings
Build history
Report groups
Report history
Account metrics
Deploy • CodeDeploy
Pipeline • CodePipeline
Settings
Installing corretto(OpenJDK) version 11 ...
[Container] 2022/06/07 18:31:26 Running command export JAVA_HOME="$JAVA_11_HOME"
[Container] 2022/06/07 18:31:26 Running command export JRE_HOME="$JRE_11_HOME"
[Container] 2022/06/07 18:31:26 Running command export JDK_HOME="$JDK_11_HOME"
[Container] 2022/06/07 18:31:26 Running command for tool_path in "$JAVA_HOME"/bin/*;
do tool=`basename "$tool_path"`;
if [$tool != 'java-rmi.cgi'];
then
rm -f /usr/bin/$tool /var/lib/alternatives/$tool \
&& update-alternatives --install /usr/bin/$tool $tool $tool_path 20000;
fi;
done
[Container] 2022/06/07 18:31:29 Moving to directory /codebuild/output/src454228208/src
[Container] 2022/06/07 18:31:29 Configuring ssm agent with target id: codebuild:ad7f3195-97c3-4ab9-b6d5-18766ef15a5d
[Container] 2022/06/07 18:31:29 Successfully updated ssm agent configuration
[Container] 2022/06/07 18:31:29 Registering with agent
[Container] 2022/06/07 18:31:29 Phases found in YAML: 3
[Container] 2022/06/07 18:31:29 POST_BUILD: 1 commands
[Container] 2022/06/07 18:31:29 INSTALL: 0 commands
[Container] 2022/06/07 18:31:29 BUILD: 1 commands
[Container] 2022/06/07 18:31:29 Phase complete: DOWNLOAD_SOURCE State: SUCCEEDED
[Container] 2022/06/07 18:31:29 Phase context status code: Message:
[Container] 2022/06/07 18:31:29 Entering phase INSTALL
[Container] 2022/06/07 18:31:29 Phase complete: INSTALL State: SUCCEEDED
[Container] 2022/06/07 18:31:29 Phase context status code: Message:
[Container] 2022/06/07 18:31:29 Entering phase PRE_BUILD
[Container] 2022/06/07 18:31:29 Phase complete: PRE_BUILD State: SUCCEEDED
[Container] 2022/06/07 18:31:29 Phase context status code: Message:
[Container] 2022/06/07 18:31:29 Entering phase BUILD
[Container] 2022/06/07 18:31:29 Running command ./gradlew bootJar
Feedback
Looking for language selection? Find it in the new Unified Settings
© 2022, Amazon Internet Services Private Ltd. or its affiliates.
Privacy
Terms

Steps to push above project on Github

1.Go to github account and create a new repository as Step 5_Testing in a DevOps Lifecycle

2.Open git bash in folder which is to push on github

3.Initialize this repository using **git init** command

4. Adding all files to staging area using **git add .** command

5. Commit the changes using the

git commit -m "first commit step5" command.

6. Update the changes made in this local repository into remote repository using command

git remote add origin https://github.com/TShivansh/Step-5_Testing-in-a-DevOps-Lifecycle.git

7.Then use git push -u origin main command to push the project to Github

GitHub Repository link:

https://github.com/TShivansh/Step-5_Testing-in-a-DevOps-Lifecycle