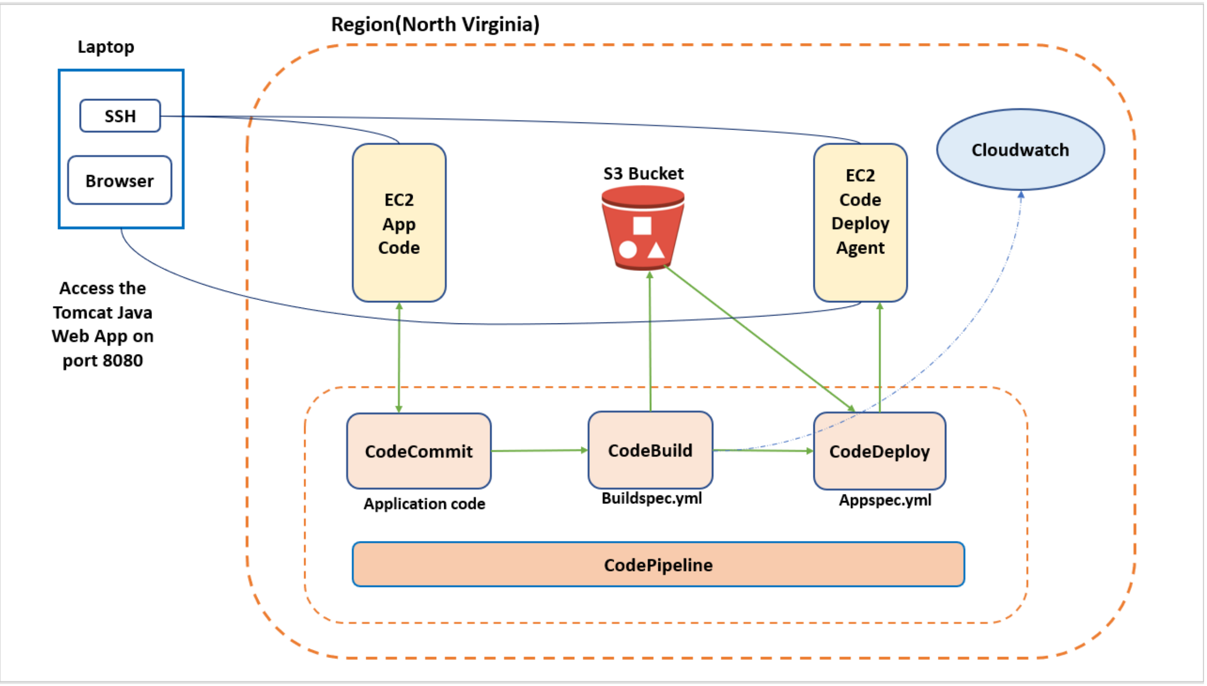
DevOps on AWS(CICD/CodePipeline)

Setup a CodeCommit Git repository to check in a sample codebase, specify build specifications using CodeBuild and deployment rules with CodeDeploy, and automate the whole process using CodePipeline.

FINAL OUTCOME

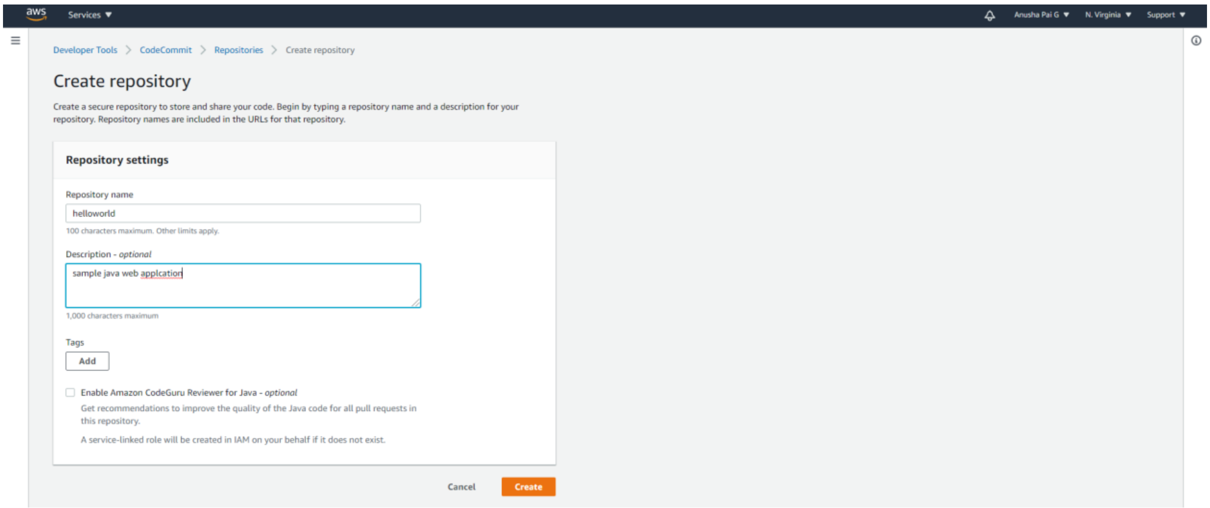


Steps:

1) Setup an Ubuntu 18.04 EC2 instance named “dev server” to manage GIT repo, install AWS CLI and configure region (SG : port 22)
2) Create an IAM role having full access to Code[commit|deploy|pipeline|S3|cloudwatch] and attach it to both the EC2 instances i.e “dev-server” and “prod-server”
3) Create a repo in CodeCommit called “helloworld”, create a new IAM user named “opsroot”(CodeCommit , console access) and login to the console using “opsroot” credentials, follow the prerequisite connection steps for SSH Linux
4) Create a build project using CodeBuild and an S3 bucket to push the artifacts
5) Spin up an Ubuntu 18.04 EC2 instance named “prod server” to deploy the codebase (SG : port 22, 8080) ,install AWS CLI and configure region, install CodeDeploy agent, tomcat8 to run the java application
6) Create a Code Deployment Group in CodeDeploy
7) Create a CodePipeline to automate the above work flow

CODECOMMIT – DEV SERVER

1)Create a repo to upload the code base


aws
Services ▼
Anusha Pai G ▼
N. Virginia ▼
Support ▼
Developer Tools > CodeCommit > Repositories > Create repository

Create repository

Create a secure repository to store and share your code. Begin by typing a repository name and a description for your repository. Repository names are included in the URLs for that repository.

Repository settings

Repository name: helloworld

100 characters maximum. Other limits apply.

Description - *optional*: sample java web applcation

1,000 characters maximum

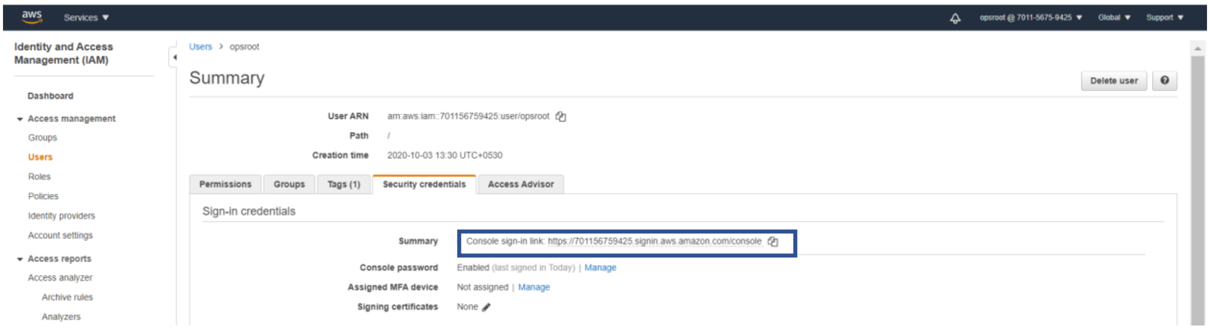
Tags

□ Enable Amazon CodeGuru Reviewer for Java - *optional*

Get recommendations to improve the quality of the Java code for all pull requests in this repository.

A service-linked role will be created in IAM on your behalf if it does not exist.

2)Create a new IAM user and login to the AWS console using the new user’s credentials



Summary

User ARN: arn:aws:iam::701156759425:user/opsroot

Path: /

Creation time: 2020-10-03 13:30 UTC+0530

Sign-in credentials

Summary: Console sign-in link: https://701156759425.signin.aws.amazon.com/console

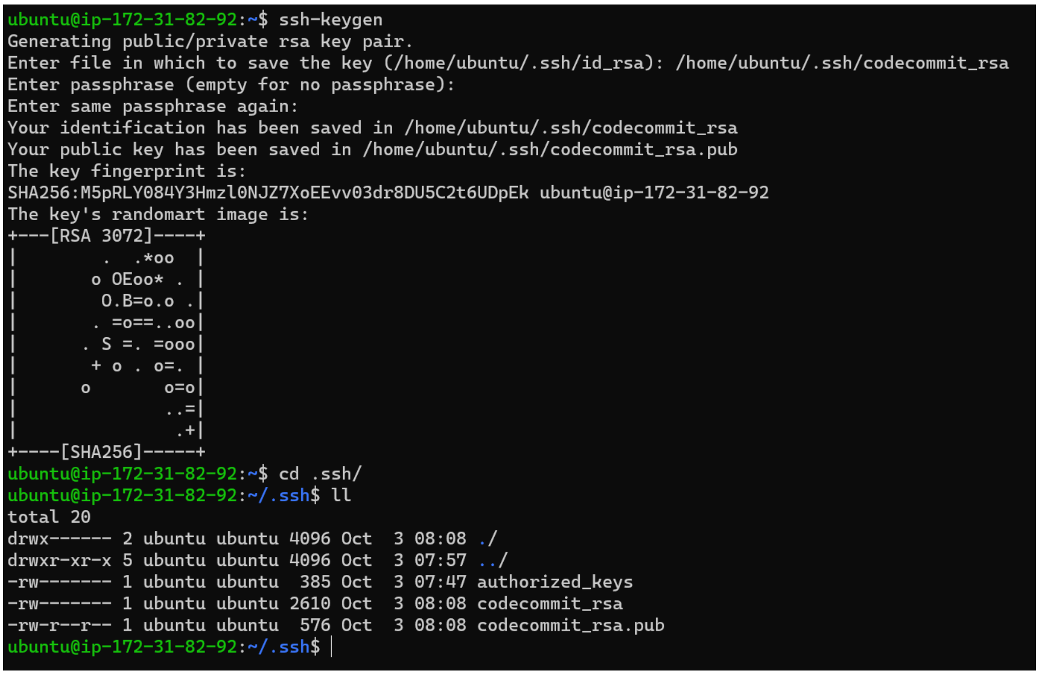
Console password: Enabled (last signed in Today) | Manage

Assigned MFA device: Not assigned | Manage

Signing certificates: None

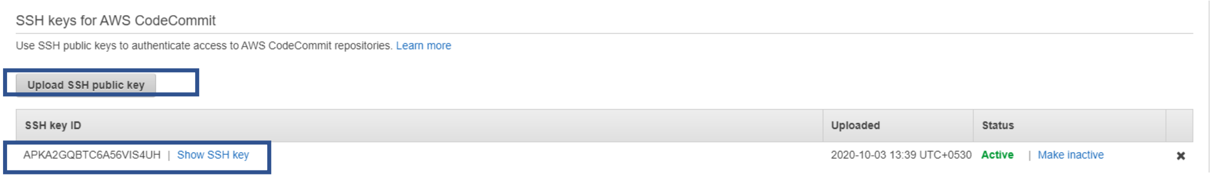
3)Follow the prerequisite to enable SSH based connection to CodeCommit repository

a)Generate a new SSH key pair



```
ubuntu@ip-172-31-82-92:~$ ssh-keygen
Generating public/private rsa key pair.
Enter file in which to save the key (/home/ubuntu/.ssh/id_rsa): /home/ubuntu/.ssh/codecommit_rsa
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /home/ubuntu/.ssh/codecommit_rsa
Your public key has been saved in /home/ubuntu/.ssh/codecommit_rsa.pub
The key fingerprint is:
SHA256:M5pRLY084Y3Hmzl0NJZ7XoEEvv03dr8DU5C2t6UDpEk ubuntu@ip-172-31-82-92
The key's randomart image is:
+---[RSA 3072]----+
|         .  .*oo |
|       o OEoo* . |
|        O.B=o.o .|
|      . =o==..oo|
|     . S =. =ooo|
|      + o . o=. |
|     o       o=o|
|             ..=|
|              .+|
+----[SHA256]-----+
ubuntu@ip-172-31-82-92:~$ cd .ssh/
ubuntu@ip-172-31-82-92:~/.ssh$ ll
total 20
drwx------ 2 ubuntu ubuntu 4096 Oct  3 08:08 ./
drwxr-xr-x 5 ubuntu ubuntu 4096 Oct  3 07:57 ../
-rw------- 1 ubuntu ubuntu  385 Oct  3 07:47 authorized_keys
-rw------- 1 ubuntu ubuntu 2610 Oct  3 08:08 codecommit_rsa
-rw-r--r-- 1 ubuntu ubuntu  576 Oct  3 08:08 codecommit_rsa.pub
ubuntu@ip-172-31-82-92:~/.ssh$
```

b)Register SSH Public key i.e copy the public key in ~/.ssh/codecommit_rsa.pub and upload the SSH public key in security credentials tab of the IAM “opsroot”. This will generate the SSH Key ID to authenticate to AWS CodeCommit repo

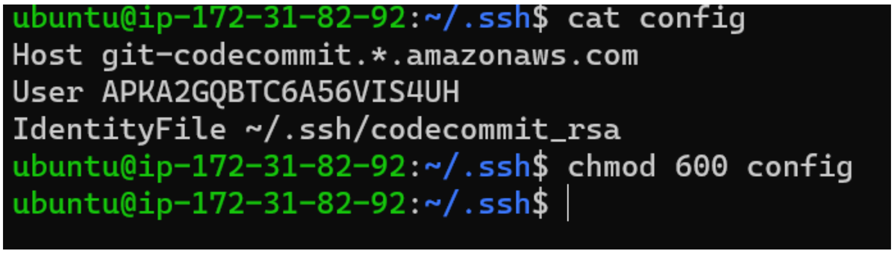


SSH keys for AWS CodeCommit

Use SSH public keys to authenticate access to AWS CodeCommit repositories. Learn more

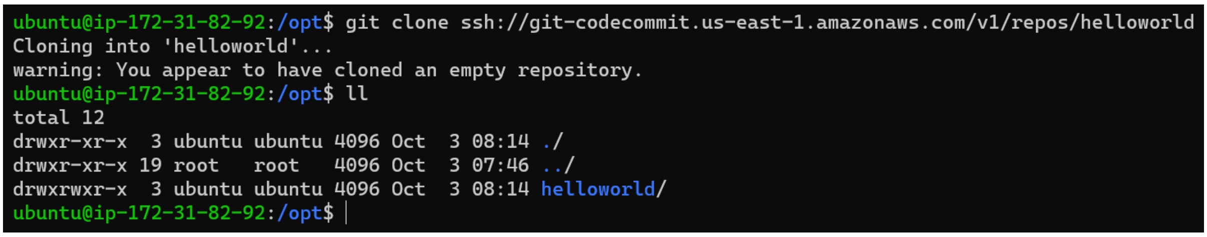
SSH key ID	Uploaded	Status	
APKA2GQBTC6A56VIS4UH | Show SSH key	2020-10-03 13:39 UTC+0530	Active | Make inactive	✖

c)Edit local SSH configuration , create a file named “config” with the below content



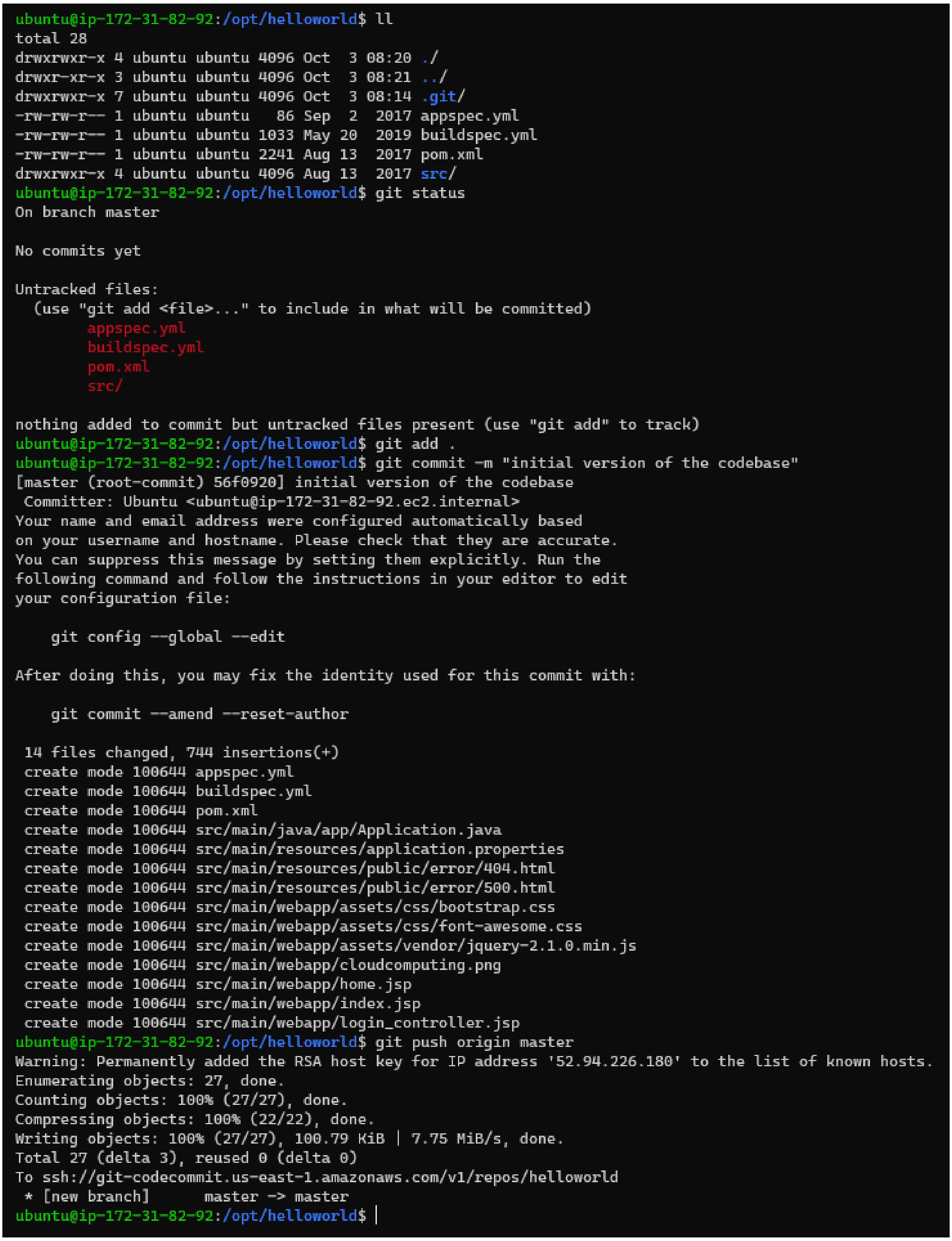
```
ubuntu@ip-172-31-82-92:~/.ssh$ cat config
Host git-codecommit.*.amazonaws.com
User APKA2GQBTC6A56VIS4UH
IdentityFile ~/.ssh/codecommit_rsa
ubuntu@ip-172-31-82-92:~/.ssh$ chmod 600 config
ubuntu@ip-172-31-82-92:~/.ssh$
```

d)Clone the repo

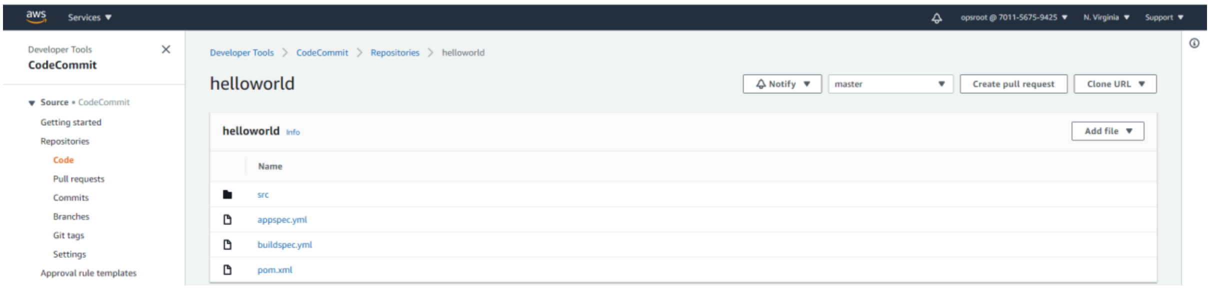


```
ubuntu@ip-172-31-82-92:/opt$ git clone ssh://git-codecommit.us-east-1.amazonaws.com/v1/repos/helloworld
Cloning into 'helloworld'...
warning: You appear to have cloned an empty repository.
ubuntu@ip-172-31-82-92:/opt$ ll
total 12
drwxr-xr-x  3 ubuntu ubuntu 4096 Oct  3 08:14 ./
drwxr-xr-x 19 root   root   4096 Oct  3 07:46 ../
drwxrwxr-x  3 ubuntu ubuntu 4096 Oct  3 08:14 helloworld/
ubuntu@ip-172-31-82-92:/opt$
```

4)Setup the Code



```
ubuntu@ip-172-31-82-92:/opt/helloworld$ ll
total 28
drwxrwxr-x 4 ubuntu ubuntu 4096 Oct  3 08:20 ./
drwxr-xr-x 3 ubuntu ubuntu 4096 Oct  3 08:21 ../
drwxrwxr-x 7 ubuntu ubuntu 4096 Oct  3 08:14 .git/
-rw-rw-r-- 1 ubuntu ubuntu   86 Sep  2  2017 appspec.yml
-rw-rw-r-- 1 ubuntu ubuntu 1033 May 20  2019 buildspec.yml
-rw-rw-r-- 1 ubuntu ubuntu 2241 Aug 13  2017 pom.xml
drwxrwxr-x 4 ubuntu ubuntu 4096 Aug 13  2017 src/
ubuntu@ip-172-31-82-92:/opt/helloworld$ git status
On branch master

No commits yet

Untracked files:
  (use "git add <file>..." to include in what will be committed)
        appspec.yml
        buildspec.yml
        pom.xml
        src/

nothing added to commit but untracked files present (use "git add" to track)
ubuntu@ip-172-31-82-92:/opt/helloworld$ git add .
ubuntu@ip-172-31-82-92:/opt/helloworld$ git commit -m "initial version of the codebase"
[master (root-commit) 56f0920] initial version of the codebase
 Committer: Ubuntu <ubuntu@ip-172-31-82-92.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. Run the
following command and follow the instructions in your editor to edit
your configuration file:

    git config --global --edit

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

    git commit --amend --reset-author

 14 files changed, 744 insertions(+)
 create mode 100644 appspec.yml
 create mode 100644 buildspec.yml
 create mode 100644 pom.xml
 create mode 100644 src/main/java/app/Application.java
 create mode 100644 src/main/resources/application.properties
 create mode 100644 src/main/resources/public/error/404.html
 create mode 100644 src/main/resources/public/error/500.html
 create mode 100644 src/main/webapp/assets/css/bootstrap.css
 create mode 100644 src/main/webapp/assets/css/font-awesome.css
 create mode 100644 src/main/webapp/assets/vendor/jquery-2.1.0.min.js
 create mode 100644 src/main/webapp/cloudcomputing.png
 create mode 100644 src/main/webapp/home.jsp
 create mode 100644 src/main/webapp/index.jsp
 create mode 100644 src/main/webapp/login_controller.jsp
ubuntu@ip-172-31-82-92:/opt/helloworld$ git push origin master
Warning: Permanently added the RSA host key for IP address '52.94.226.180' to the list of known hosts.
Enumerating objects: 27, done.
Counting objects: 100% (27/27), done.
Compressing objects: 100% (22/22), done.
Writing objects: 100% (27/27), 100.79 KiB | 7.75 MiB/s, done.
Total 27 (delta 3), reused 0 (delta 0)
To ssh://git-codecommit.us-east-1.amazonaws.com/v1/repos/helloworld
 * [new branch]      master -> master
ubuntu@ip-172-31-82-92:/opt/helloworld$
```



helloworld

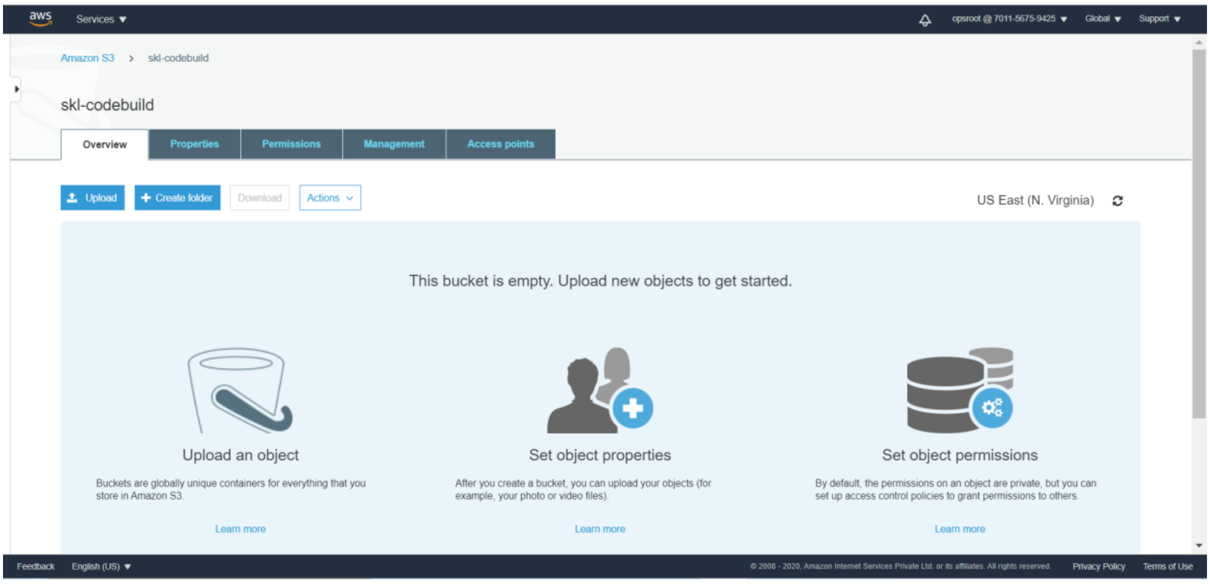
helloworld Info

Name
src
appspec.yml
buildspec.yml
pom.xml

CODE BUILD - CONSOLE

create a build environment i.e a docker image built temporarily for the source code to compile and generate an artifact which is uploaded to S3 and the build logs are logged into the CloudWatch Log group

1)Create an S3 bucket to hold the artifacts generated by CodeBuild Environment



skl-codebuild

This bucket is empty. Upload new objects to get started.

Upload an object

Buckets are globally unique containers for everything that you store in Amazon S3.

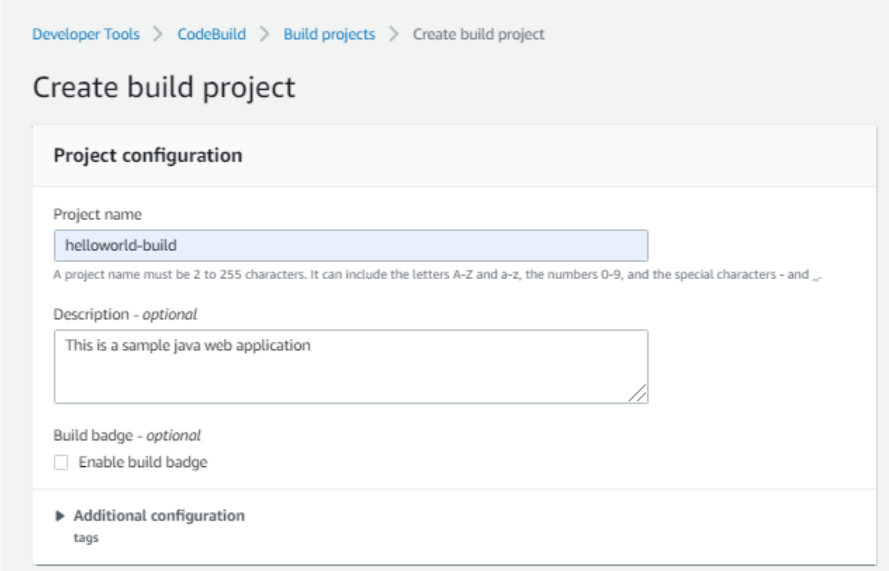
Set object properties

After you create a bucket, you can upload your objects (for example, your photo or video files).

Set object permissions

By default, the permissions on an object are private, but you can set up access control policies to grant permissions to others.

2) Create a Build Project as below


Developer Tools > CodeBuild > Build projects > Create build project

Create build project

Project configuration

Project name

helloworld-build

A project name must be 2 to 255 characters. It can include the letters A-Z and a-z, the numbers 0-9, and the special characters - and _.

Description - *optional*

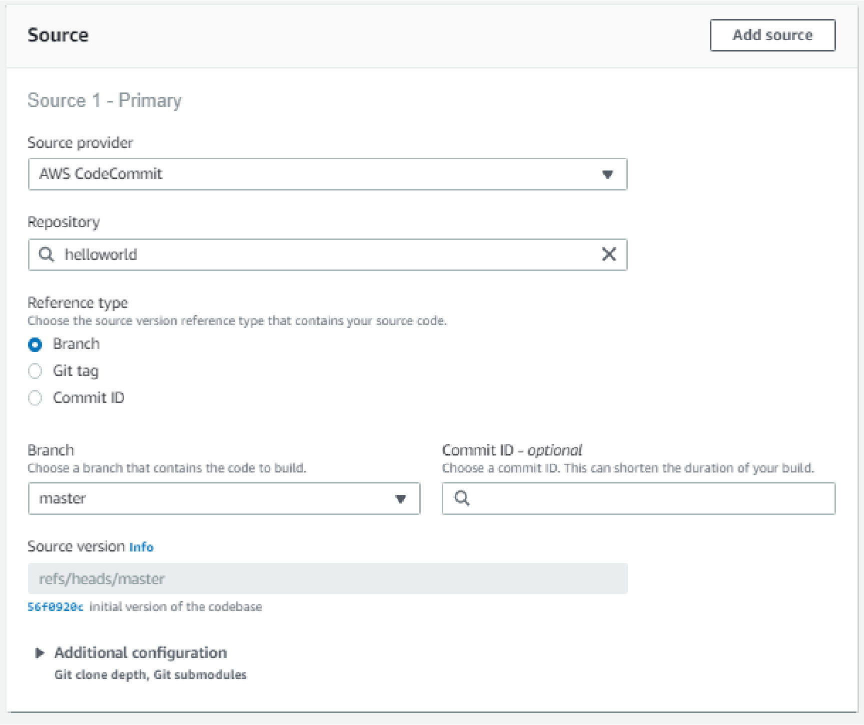
This is a sample java web application

Build badge - *optional*

□ Enable build badge

▶ Additional configuration

tags



Source 1 - Primary

Source provider

AWS CodeCommit ▼

Repository

helloworld

Reference type

Choose the source version reference type that contains your source code.

● Branch

○ Git tag

○ Commit ID

Branch

Choose a branch that contains the code to build.

master ▼

Commit ID - *optional*

Choose a commit ID. This can shorten the duration of your build.

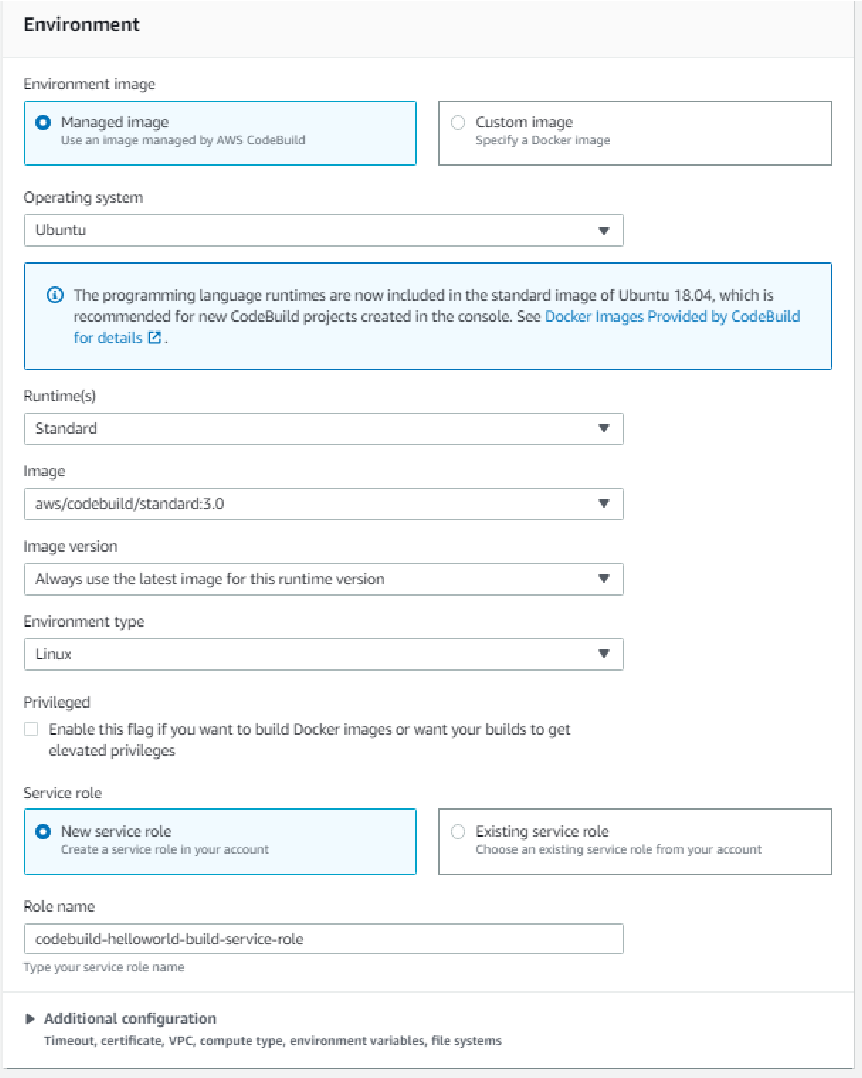
Source version Info

refs/heads/master

56f0920c initial version of the codebase

▶ Additional configuration

Git clone depth, Git submodules



Environment image

● Managed image
Use an image managed by AWS CodeBuild

○ Custom image
Specify a Docker image

Operating system

Ubuntu ▼

The programming language runtimes are now included in the standard image of Ubuntu 18.04, which is recommended for new CodeBuild projects created in the console. See Docker Images Provided by CodeBuild for details .

Runtime(s)

Standard ▼

Image

aws/codebuild/standard:3.0 ▼

Image version

Always use the latest image for this runtime version ▼

Environment type

Linux ▼

Privileged

□ Enable this flag if you want to build Docker images or want your builds to get elevated privileges

Service role

● New service role
Create a service role in your account

○ Existing service role
Choose an existing service role from your account

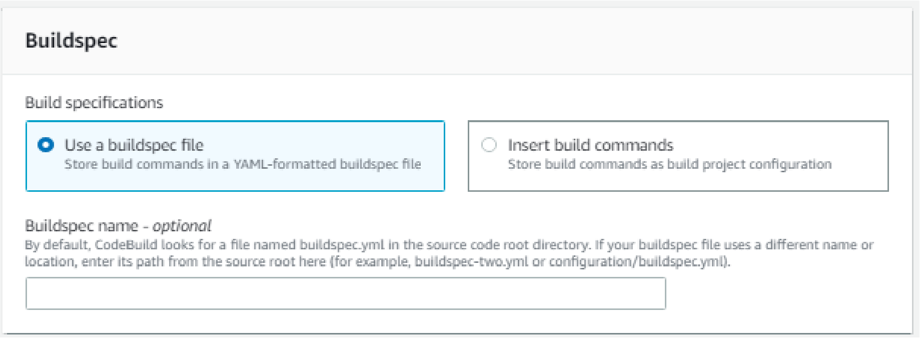
Role name

codebuild-helloworld-build-service-role

Type your service role name

▶ Additional configuration

Timeout, certificate, VPC, compute type, environment variables, file systems



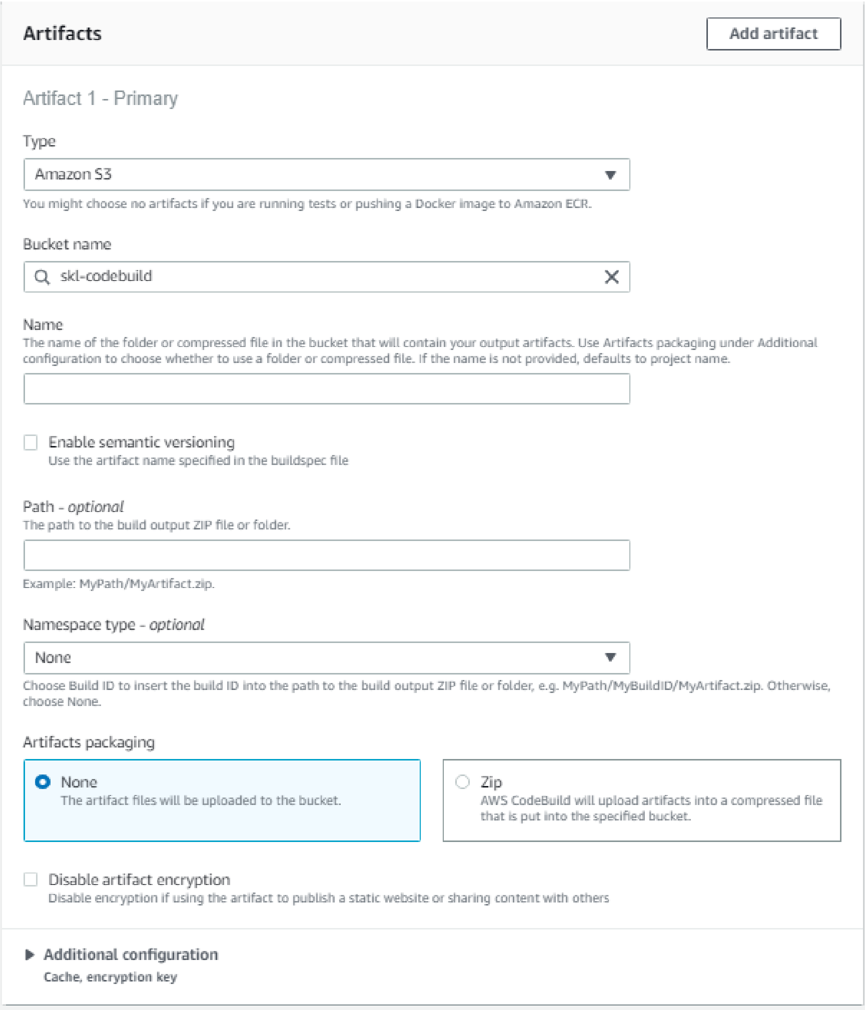
Buildspec

Build specifications

● Use a buildspec file
Store build commands in a YAML-formatted buildspec file

○ Insert build commands
Store build commands as build project configuration

Buildspec name - *optional*
By default, CodeBuild looks for a file named buildspec.yml in the source code root directory. If your buildspec file uses a different name or location, enter its path from the source root here (for example, buildspec-two.yml or configuration/buildspec.yml).



Artifacts

Artifact 1 - Primary

Type
Amazon S3
You might choose no artifacts if you are running tests or pushing a Docker image to Amazon ECR.

Bucket name
skl-codebuild

Name
The name of the folder or compressed file in the bucket that will contain your output artifacts. Use Artifacts packaging under Additional configuration to choose whether to use a folder or compressed file. If the name is not provided, defaults to project name.

□ Enable semantic versioning
Use the artifact name specified in the buildspec file

Path - *optional*
The path to the build output ZIP file or folder.

Example: MyPath/MyArtifact.zip.

Namespace type - *optional*
None
Choose Build ID to insert the build ID into the path to the build output ZIP file or folder, e.g. MyPath/MyBuildID/MyArtifact.zip. Otherwise, choose None.

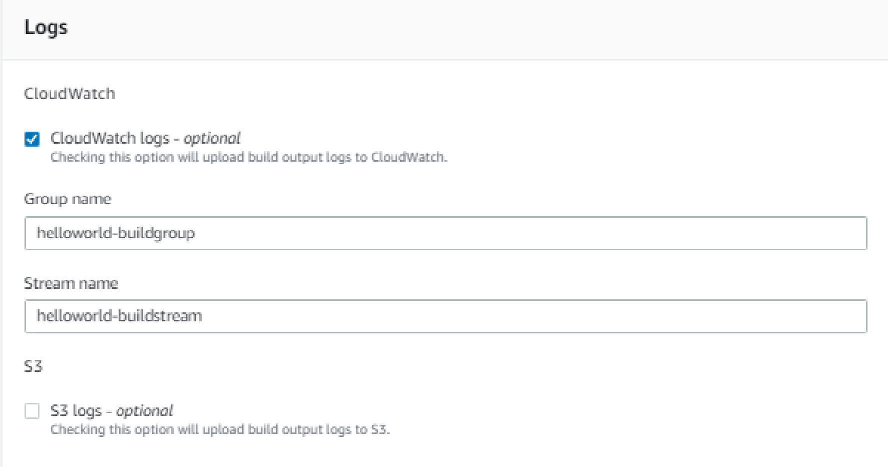
Artifacts packaging

● None
The artifact files will be uploaded to the bucket.

○ Zip
AWS CodeBuild will upload artifacts into a compressed file that is put into the specified bucket.

□ Disable artifact encryption
Disable encryption if using the artifact to publish a static website or sharing content with others

▶ **Additional configuration**
Cache, encryption key



Logs

CloudWatch

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

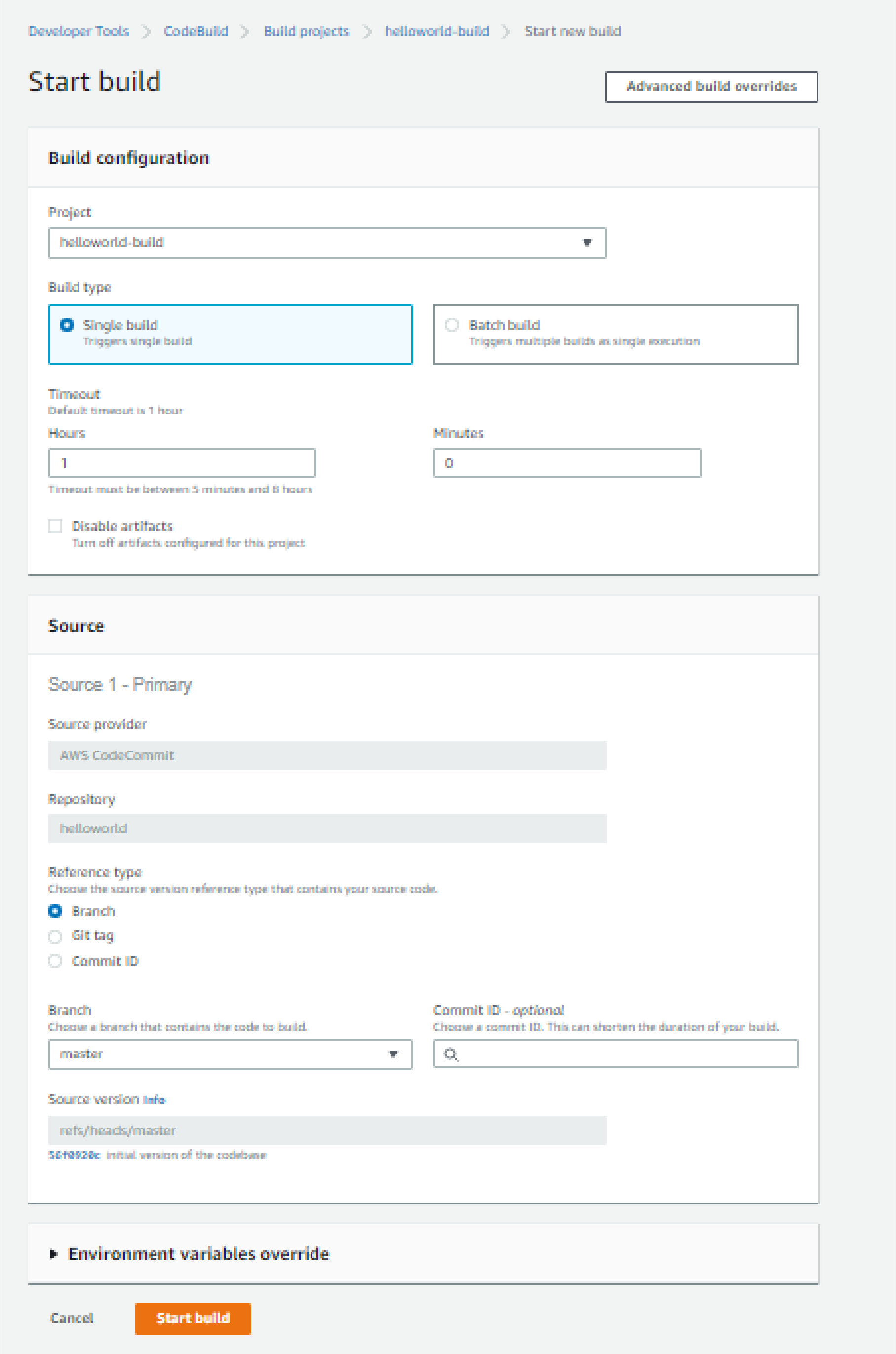
Group name
helloworld-buildgroup

Stream name
helloworld-buildstream

S3

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

3) Run the created build manually , verify the build events and ensure the artifact is pushed to the S3 bucket



Developer Tools > CodeBuild > Build projects > helloworld-build > Start new build

Start build

Advanced build overrides

Build configuration

Project

helloworld-build ▾

Build type

● Single build
Triggers single build

○ Batch build
Triggers multiple builds as single execution

Timeout
Default timeout is 1 hour

Hours: 1

Minutes: 0

Timeout must be between 5 minutes and 8 hours

□ Disable artifacts
Turn off artifacts configured for this project

Source

Source 1 - Primary

Source provider

AWS CodeCommit

Repository

helloworld

Reference type
Choose the source version reference type that contains your source code.

● Branch
○ Git tag
○ Commit ID

Branch
Choose a branch that contains the code to build.

master ▾

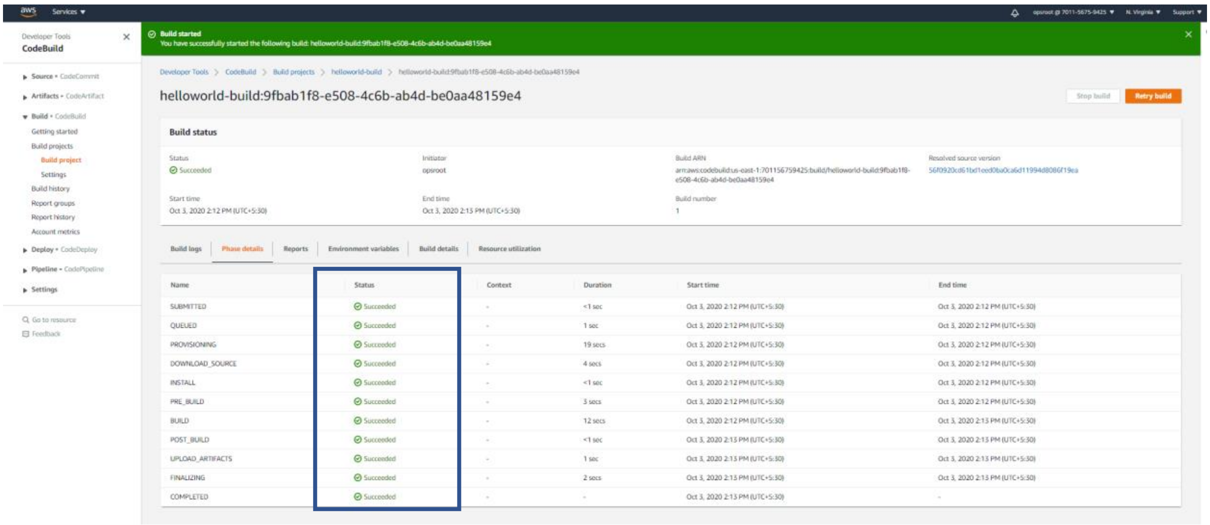
Commit ID - *optional*
Choose a commit ID. This can shorten the duration of your build.

Source version Info

refs/heads/master

5cf0920c initial version of the codebase

▸ **Environment variables override**



Build started
You have successfully started the following build: helloworld-build:9fbab1f8-e508-4c6b-ab4d-be0aa48159e4

helloworld-build:9fbab1f8-e508-4c6b-ab4d-be0aa48159e4

Build status

Status: Succeeded

Initiator: opsroot

Build ARN: arn:aws:codebuild:us-east-1:701156759425:build/helloworld-build:9fbab1f8-e508-4c6b-ab4d-be0aa48159e4

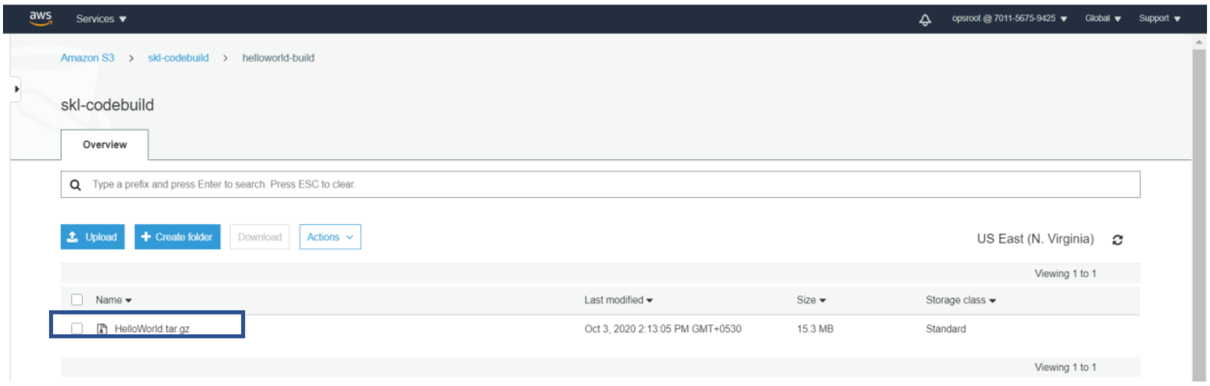
Resolved source version: 56f0920cd61bd1eed0ba0ca6d11994d8086f19ea

Start time: Oct 3, 2020 2:12 PM (UTC+5:30)

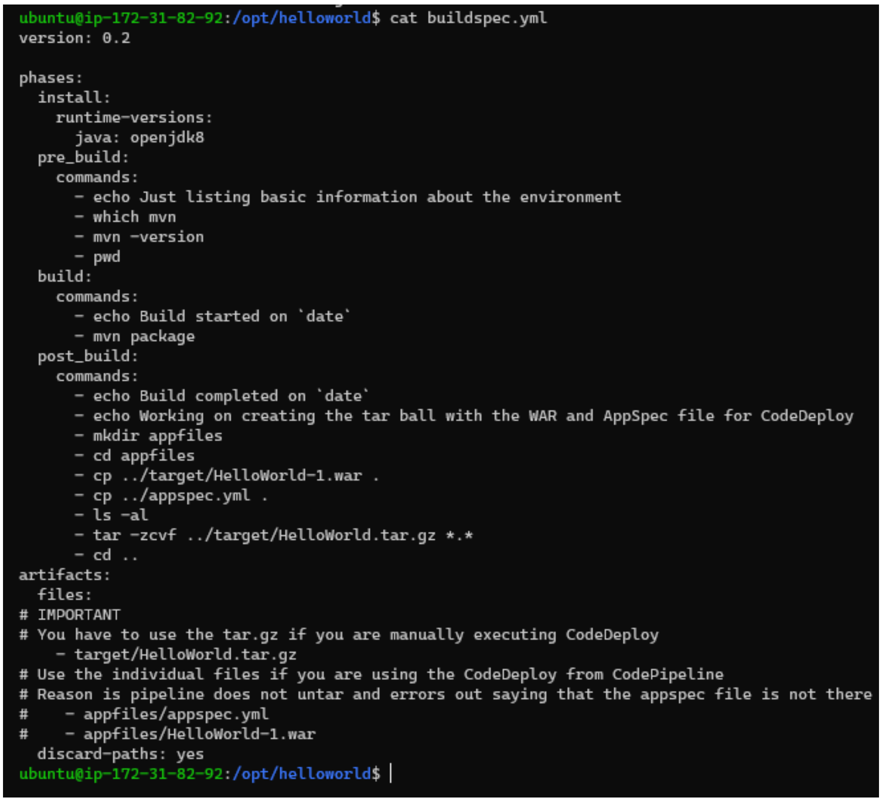
End time: Oct 3, 2020 2:13 PM (UTC+5:30)

Build number: 1

Name	Status	Context	Duration	Start time	End time
SUBMITTED	✓ Succeeded	-	<1 sec	Oct 3, 2020 2:12 PM (UTC+5:30)	Oct 3, 2020 2:12 PM (UTC+5:30)
QUEUED	✓ Succeeded	-	1 sec	Oct 3, 2020 2:12 PM (UTC+5:30)	Oct 3, 2020 2:12 PM (UTC+5:30)
PROVISIONING	✓ Succeeded	-	19 secs	Oct 3, 2020 2:12 PM (UTC+5:30)	Oct 3, 2020 2:12 PM (UTC+5:30)
DOWNLOAD_SOURCE	✓ Succeeded	-	4 secs	Oct 3, 2020 2:12 PM (UTC+5:30)	Oct 3, 2020 2:12 PM (UTC+5:30)
INSTALL	✓ Succeeded	-	<1 sec	Oct 3, 2020 2:12 PM (UTC+5:30)	Oct 3, 2020 2:12 PM (UTC+5:30)
PRE_BUILD	✓ Succeeded	-	3 secs	Oct 3, 2020 2:12 PM (UTC+5:30)	Oct 3, 2020 2:12 PM (UTC+5:30)
BUILD	✓ Succeeded	-	12 secs	Oct 3, 2020 2:12 PM (UTC+5:30)	Oct 3, 2020 2:13 PM (UTC+5:30)
POST_BUILD	✓ Succeeded	-	<1 sec	Oct 3, 2020 2:13 PM (UTC+5:30)	Oct 3, 2020 2:13 PM (UTC+5:30)
UPLOAD_ARTIFACTS	✓ Succeeded	-	1 sec	Oct 3, 2020 2:13 PM (UTC+5:30)	Oct 3, 2020 2:13 PM (UTC+5:30)
FINALIZING	✓ Succeeded	-	2 secs	Oct 3, 2020 2:13 PM (UTC+5:30)	Oct 3, 2020 2:13 PM (UTC+5:30)
COMPLETED	✓ Succeeded	-	-	Oct 3, 2020 2:13 PM (UTC+5:30)	-


aws
Services ▼
opsroot @ 7011-5675-9425 ▼
Global ▼
Support ▼
Amazon S3 > skl-codebuild > helloworld-build
skl-codebuild
Overview
Type a prefix and press Enter to search. Press ESC to clear.
Upload
Create folder
Download
Actions
US East (N. Virginia)
Viewing 1 to 1

Name	Last modified	Size	Storage class
□ HelloWorld.tar.gz	Oct 3, 2020 2:13:05 PM GMT+0530	15.3 MB	Standard

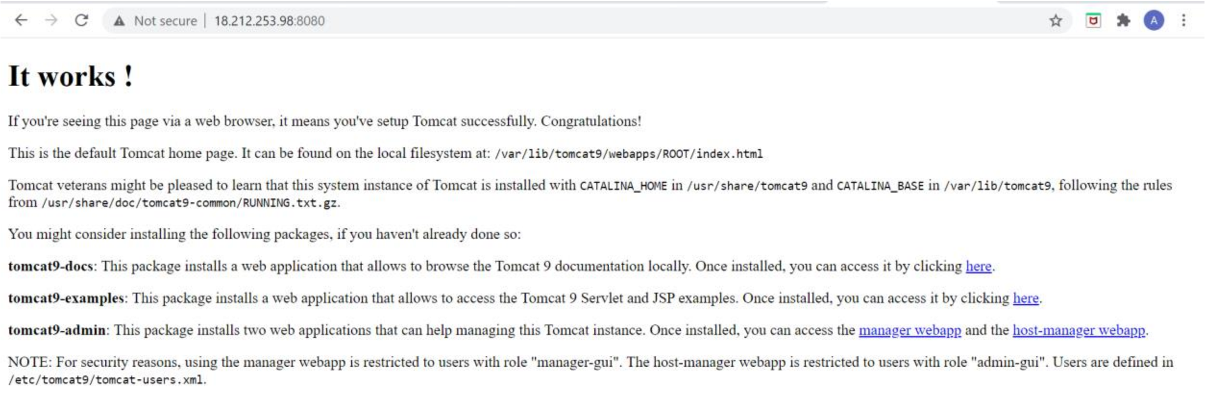


```
ubuntu@ip-172-31-82-92:/opt/helloworld$ cat buildspec.yml
version: 0.2

phases:
  install:
    runtime-versions:
      java: openjdk8
  pre_build:
    commands:
      - echo Just listing basic information about the environment
      - which mvn
      - mvn -version
      - pwd
  build:
    commands:
      - echo Build started on `date`
      - mvn package
  post_build:
    commands:
      - echo Build completed on `date`
      - echo Working on creating the tar ball with the WAR and AppSpec file for CodeDeploy
      - mkdir appfiles
      - cd appfiles
      - cp ../target/HelloWorld-1.war .
      - cp ../appspec.yml .
      - ls -al
      - tar -zcvf ../target/HelloWorld.tar.gz *.*
      - cd ..
artifacts:
  files:
# IMPORTANT
# You have to use the tar.gz if you are manually executing CodeDeploy
    - target/HelloWorld.tar.gz
# Use the individual files if you are using the CodeDeploy from CodePipeline
# Reason is pipeline does not untar and errors out saying that the appspec file is not there
#    - appfiles/appspec.yml
#    - appfiles/HelloWorld-1.war
  discard-paths: yes
ubuntu@ip-172-31-82-92:/opt/helloworld$
```

CODE DEPLOY - PROD SERVER

Work Flow: Application(name) -> deployment group(specify target resource and deployment method)-> deployment(where to find the codebase for deployment)

1)Spin up an EC2 instance and verify the contents of the Tomcat DocumentRoot i.e /var/lib/tomcat8/webapps, validate the default page is serving the request



It works !

If you're seeing this page via a web browser, it means you've setup Tomcat successfully. Congratulations!

This is the default Tomcat home page. It can be found on the local filesystem at: `/var/lib/tomcat9/webapps/ROOT/index.html`

Tomcat veterans might be pleased to learn that this system instance of Tomcat is installed with `CATALINA_HOME` in `/usr/share/tomcat9` and `CATALINA_BASE` in `/var/lib/tomcat9`, following the rules from `/usr/share/doc/tomcat9-common/RUNNING.txt.gz`.

You might consider installing the following packages, if you haven't already done so:

tomcat9-docs: This package installs a web application that allows to browse the Tomcat 9 documentation locally. Once installed, you can access it by clicking here.

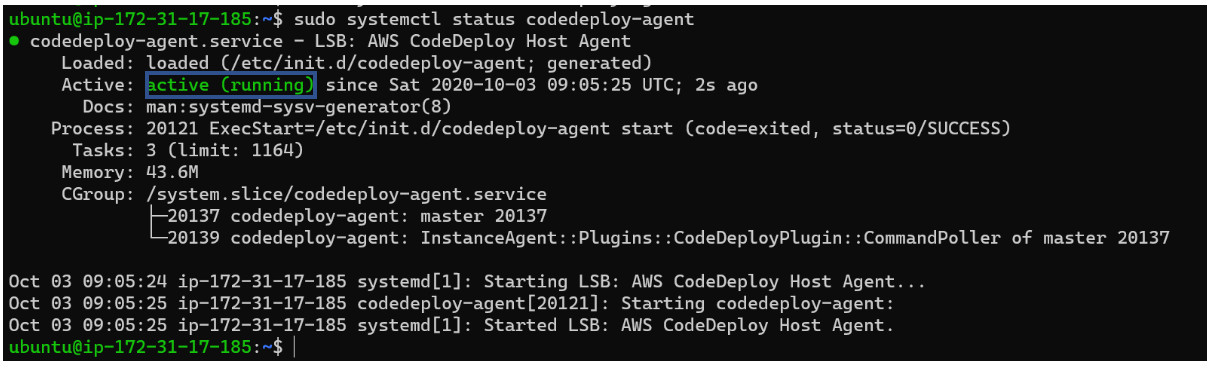
tomcat9-examples: This package installs a web application that allows to access the Tomcat 9 Servlet and JSP examples. Once installed, you can access it by clicking here.

tomcat9-admin: This package installs two web applications that can help managing this Tomcat instance. Once installed, you can access the manager webapp and the host-manager webapp.

NOTE: For security reasons, using the manager webapp is restricted to users with role "manager-gui". The host-manager webapp is restricted to users with role "admin-gui". Users are defined in `/etc/tomcat9/tomcat-users.xml`.

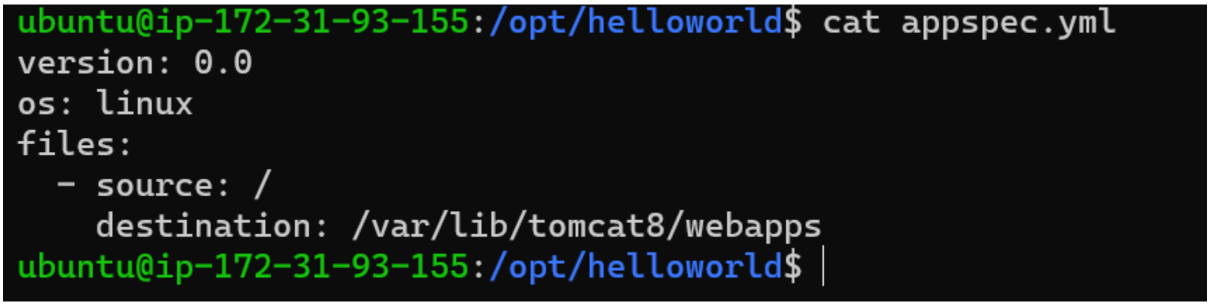
2)Install codedeploy-agent

a) sudo apt install ruby
b) cd /home/ubuntu
c) wget https://aws-codedeploy-us-east-1.s3.us-east-1.amazonaws.com/latest/install
d) chmod 700 install
e) sudo ./install auto
f) systemctl enable codedeploy-agent
g) systemctl start codedeploy-agent
h) systemctl status codedeploy-agent (ensure the status is active and running)



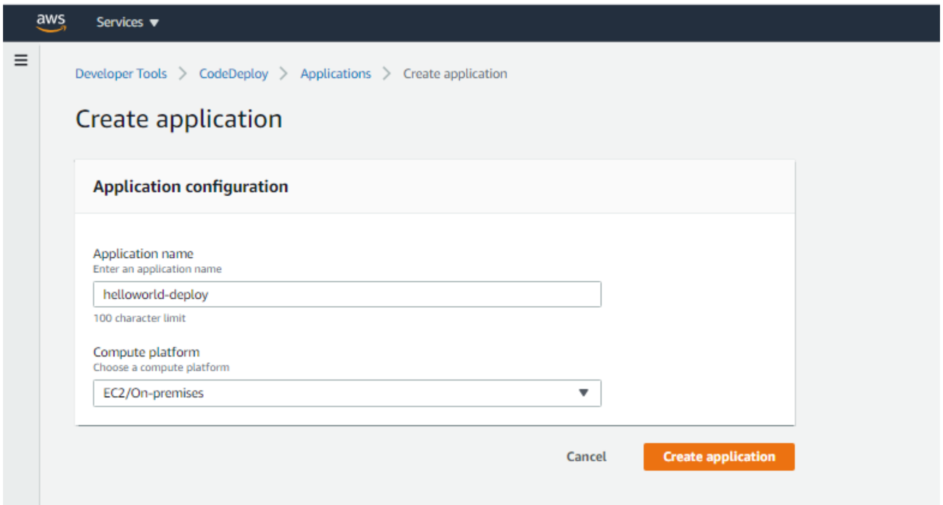
```
ubuntu@ip-172-31-17-185:~$ sudo systemctl status codedeploy-agent
● codedeploy-agent.service - LSB: AWS CodeDeploy Host Agent
     Loaded: loaded (/etc/init.d/codedeploy-agent; generated)
     Active: active (running) since Sat 2020-10-03 09:05:25 UTC; 2s ago
       Docs: man:systemd-sysv-generator(8)
    Process: 20121 ExecStart=/etc/init.d/codedeploy-agent start (code=exited, status=0/SUCCESS)
      Tasks: 3 (limit: 1164)
     Memory: 43.6M
     CGroup: /system.slice/codedeploy-agent.service
             ├─20137 codedeploy-agent: master 20137
             └─20139 codedeploy-agent: InstanceAgent::Plugins::CodeDeployPlugin::CommandPoller of master 20137

Oct 03 09:05:24 ip-172-31-17-185 systemd[1]: Starting LSB: AWS CodeDeploy Host Agent...
Oct 03 09:05:25 ip-172-31-17-185 codedeploy-agent[20121]: Starting codedeploy-agent:
Oct 03 09:05:25 ip-172-31-17-185 systemd[1]: Started LSB: AWS CodeDeploy Host Agent.
ubuntu@ip-172-31-17-185:~$
```

Below is the appspec file available in the root of the Git repo



```
ubuntu@ip-172-31-93-155:/opt/helloworld$ cat appspec.yml
version: 0.0
os: linux
files:
  - source: /
    destination: /var/lib/tomcat8/webapps
ubuntu@ip-172-31-93-155:/opt/helloworld$
```

a)Create Application


aws
Services ▼
Developer Tools > CodeDeploy > Applications > Create application

Create application

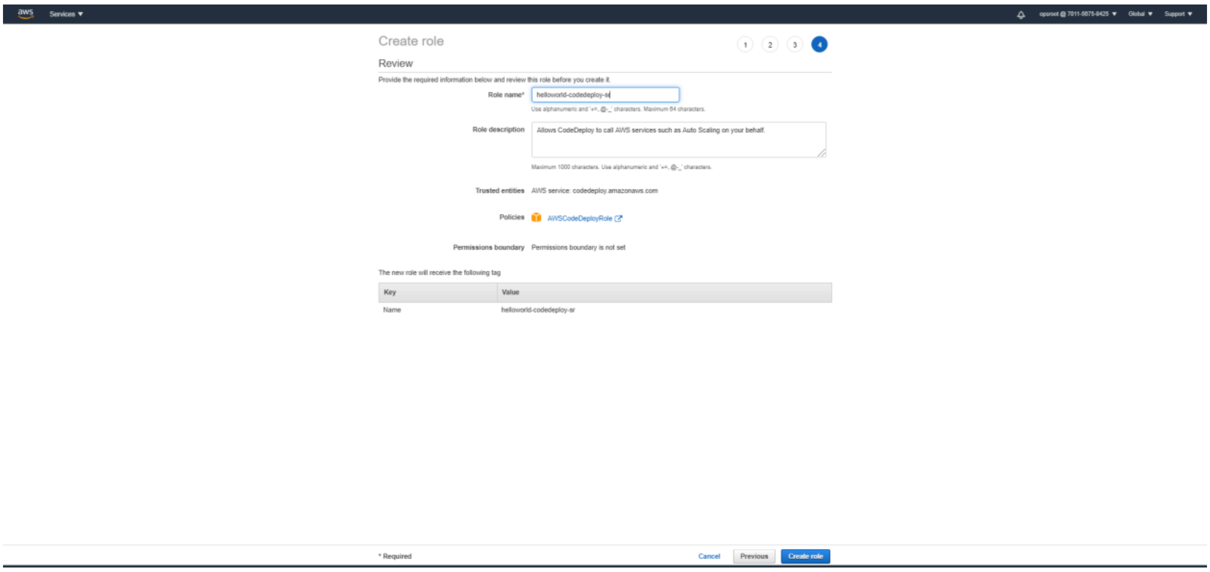
Application configuration

Application name
Enter an application name
helloworld-deploy
100 character limit

Compute platform
Choose a compute platform
EC2/On-premises ▼

Cancel
Create application

b)Create a Service Role for CodeDeploy manually and specify the ARN in the search field



Create role

Review

Provide the required information below and review this role before you create it.

Role name* helloworld-codedeploy-sr
Use alphanumeric and '+=,.@-_' characters. Maximum 64 characters.

Role description Allows CodeDeploy to call AWS services such as Auto Scaling on your behalf.
Maximum 1000 characters. Use alphanumeric and '+=,.@-_' characters.

Trusted entities AWS service: codedeploy.amazonaws.com

Policies AWSCodeDeployRole

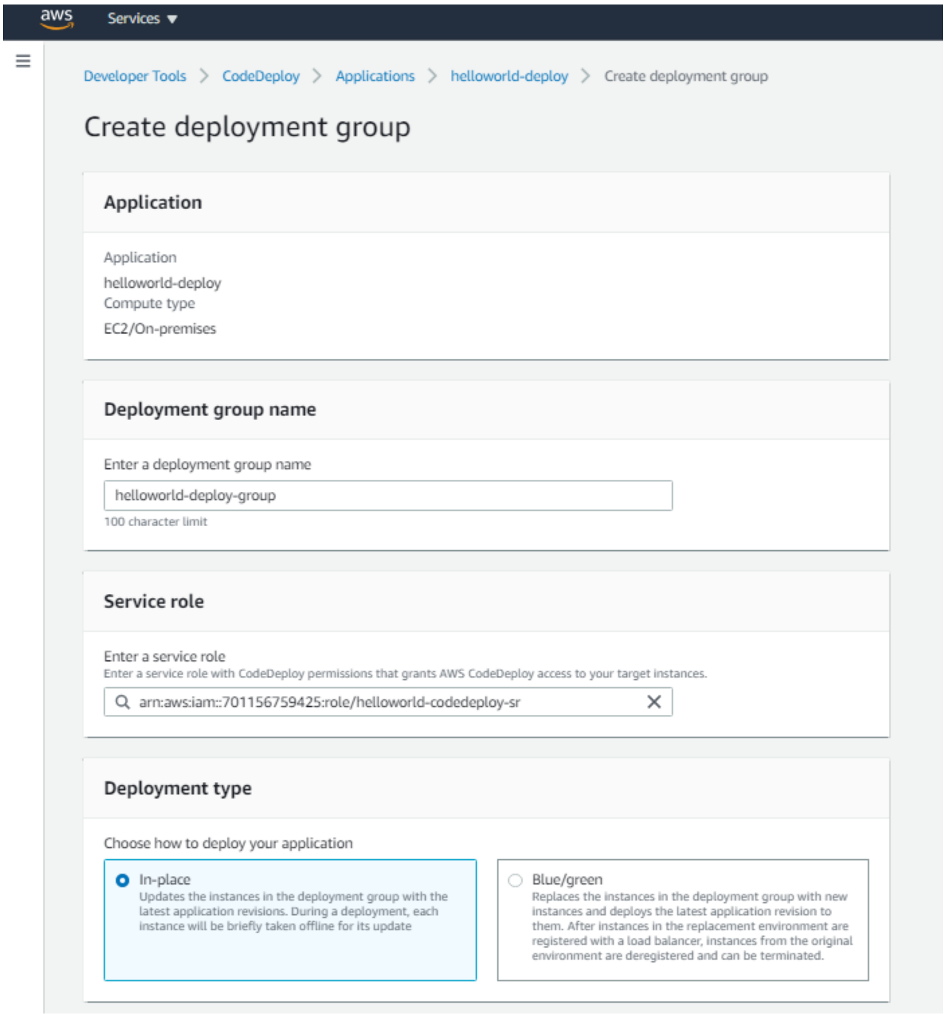
Permissions boundary Permissions boundary is not set

The new role will receive the following tag

Key	Value
Name	helloworld-codedeploy-sr

* Required

c)Create deployment group for the application



aws
Services ▼
≡
Developer Tools > CodeDeploy > Applications > helloworld-deploy > Create deployment group

Create deployment group

Application

Application
helloworld-deploy
Compute type
EC2/On-premises

Deployment group name

Enter a deployment group name
helloworld-deploy-group
100 character limit

Service role

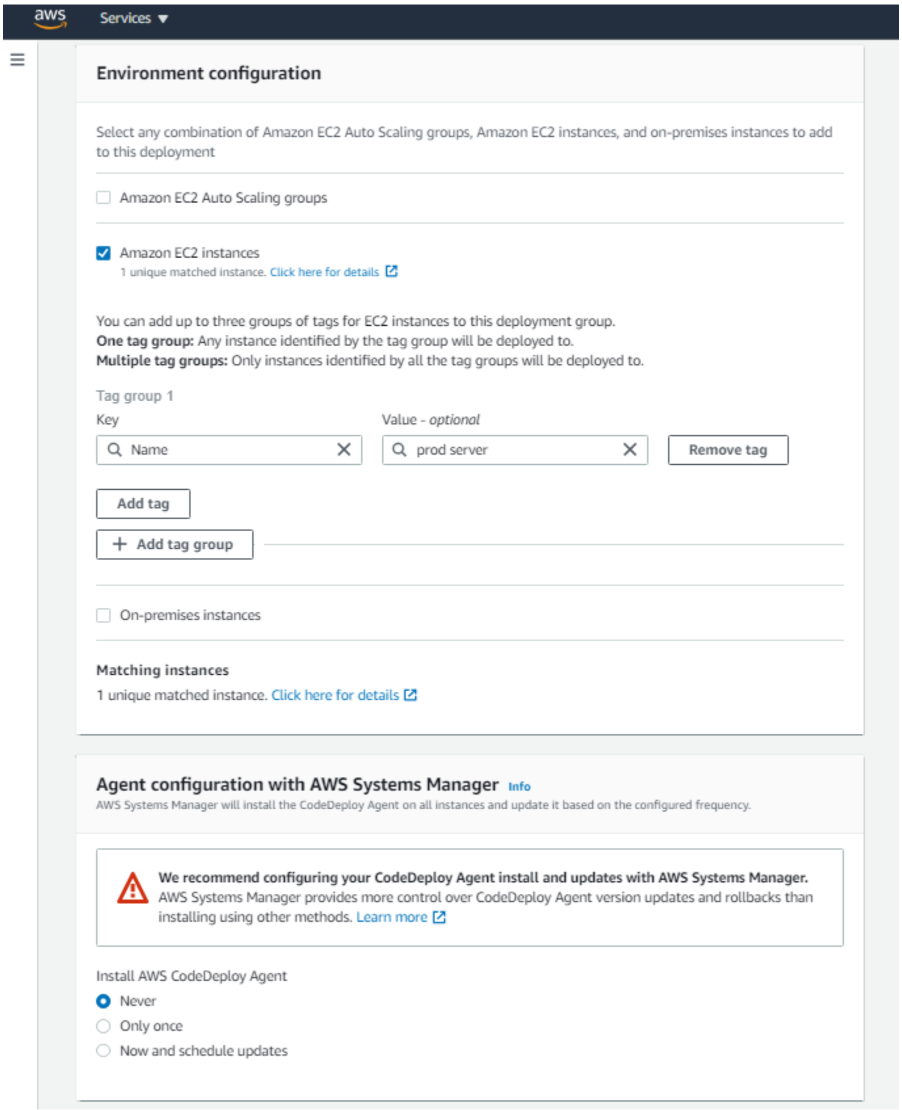
Enter a service role
Enter a service role with CodeDeploy permissions that grants AWS CodeDeploy access to your target instances.
arn:aws:iam::701156759425:role/helloworld-codedeploy-sr

Deployment type

Choose how to deploy your application

● In-place
Updates the instances in the deployment group with the latest application revisions. During a deployment, each instance will be briefly taken offline for its update

○ Blue/green
Replaces the instances in the deployment group with new instances and deploys the latest application revision to them. After instances in the replacement environment are registered with a load balancer, instances from the original environment are deregistered and can be terminated.



Environment configuration

Select any combination of Amazon EC2 Auto Scaling groups, Amazon EC2 instances, and on-premises instances to add to this deployment

□ Amazon EC2 Auto Scaling groups

☑ Amazon EC2 instances
1 unique matched instance. Click here for details

You can add up to three groups of tags for EC2 instances to this deployment group.
One tag group: Any instance identified by the tag group will be deployed to.
Multiple tag groups: Only instances identified by all the tag groups will be deployed to.

Tag group 1

Key	Value - *optional*	
Name	prod server	Remove tag

□ On-premises instances

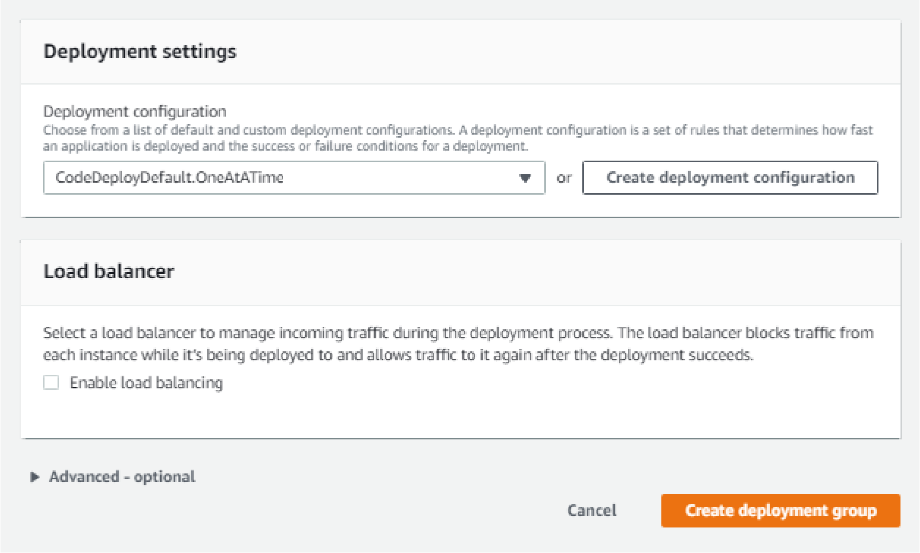
Matching instances
1 unique matched instance. Click here for details

Agent configuration with AWS Systems Manager Info

AWS Systems Manager will install the CodeDeploy Agent on all instances and update it based on the configured frequency.

We recommend configuring your CodeDeploy Agent install and updates with AWS Systems Manager.
AWS Systems Manager provides more control over CodeDeploy Agent version updates and rollbacks than installing using other methods. Learn more

Install AWS CodeDeploy Agent
● Never
○ Only once
○ Now and schedule updates



Deployment settings

Deployment configuration
Choose from a list of default and custom deployment configurations. A deployment configuration is a set of rules that determines how fast an application is deployed and the success or failure conditions for a deployment.

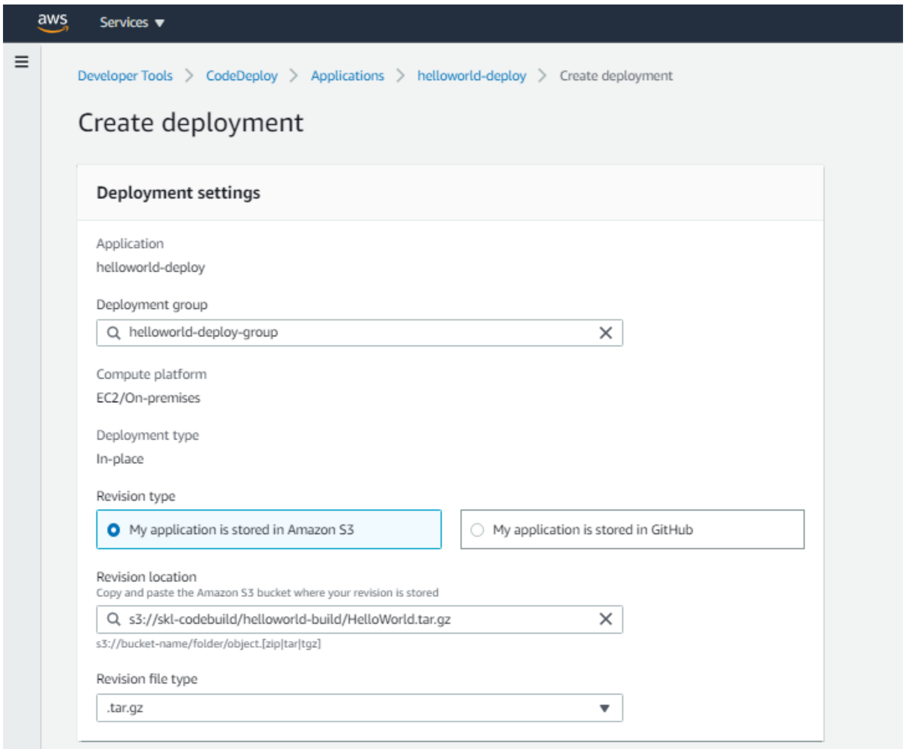
CodeDeployDefault.OneAtATime ▾ or Create deployment configuration

Load balancer

Select a load balancer to manage incoming traffic during the deployment process. The load balancer blocks traffic from each instance while it's being deployed to and allows traffic to it again after the deployment succeeds.

□ Enable load balancing

c) Create Deployment



Developer Tools > CodeDeploy > Applications > helloworld-deploy > Create deployment

Create deployment

Deployment settings

Application
helloworld-deploy

Deployment group
helloworld-deploy-group

Compute platform
EC2/On-premises

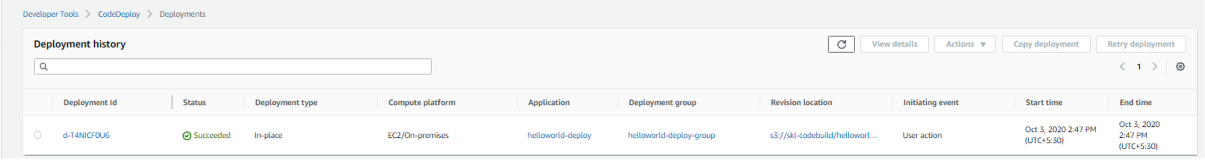
Deployment type
In-place

Revision type
● My application is stored in Amazon S3
○ My application is stored in GitHub

Revision location
Copy and paste the Amazon S3 bucket where your revision is stored
s3://skl-codebuild/helloworld-build/HelloWorld.tar.gz
s3://bucket-name/folder/object.[zip|tar|tgz]

Revision file type
.tar.gz

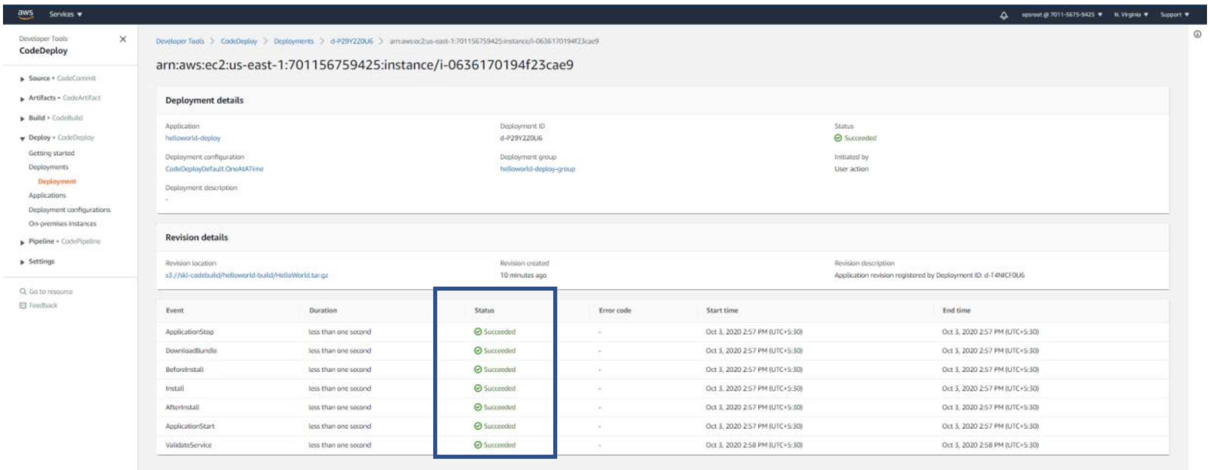
d)Verify the event status for the deployment



Developer Tools > CodeDeploy > Deployments

Deployment history

	Deployment Id	Status	Deployment type	Compute platform	Application	Deployment group	Revision location	Initiating event	Start time	End time
○	d-T4NICF0U6	✓ Succeeded	In-place	EC2/On-premises	helloworld-deploy	helloworld-deploy-group	s3://skl-codebuild/helloworl…	User action	Oct 3, 2020 2:47 PM (UTC+5:30)	Oct 3, 2020 2:47 PM (UTC+5:30)



Developer Tools > CodeDeploy > Deployments > d-P29Y220U6 > arn:aws:ec2:us-east-1:701156759425:instance/i-0636170194f23cae9

arn:aws:ec2:us-east-1:701156759425:instance/i-0636170194f23cae9

Deployment details

Application
helloworld-deploy

Deployment ID
d-P29Y220U6

Status
✓ Succeeded

Deployment configuration
CodeDeployDefault.OneAtATime

Deployment group
helloworld-deploy-group

Initiated by
User action

Deployment description
-

Revision details

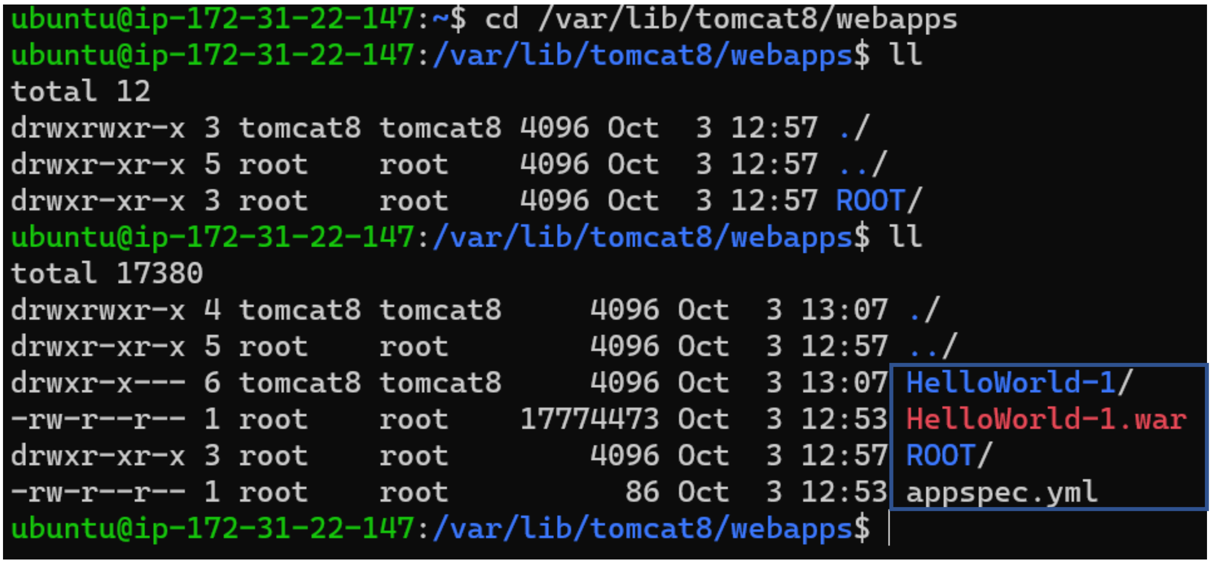
Revision location
s3://skl-codebuild/helloworld-build/HelloWorld.tar.gz

Revision created
10 minutes ago

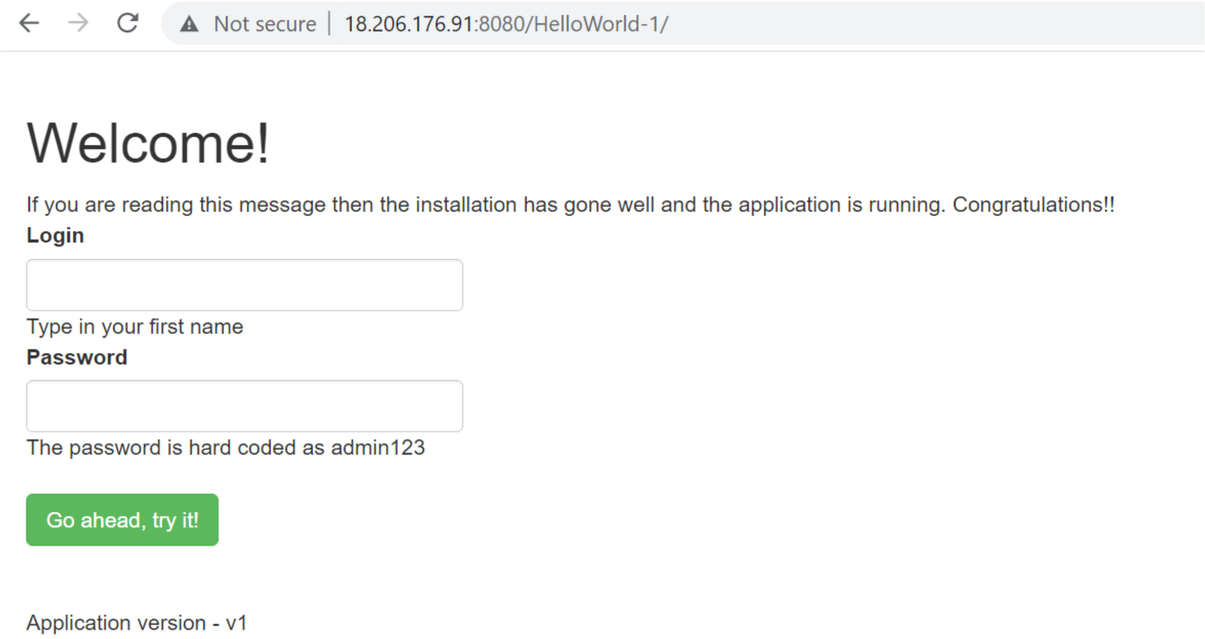
Revision description
Application revision registered by Deployment ID: d-T4NICF0U6

Event	Duration	Status	Error code	Start time	End time
ApplicationStop	less than one second	✓ Succeeded	-	Oct 3, 2020 2:57 PM (UTC+5:30)	Oct 3, 2020 2:57 PM (UTC+5:30)
DownloadBundle	less than one second	✓ Succeeded	-	Oct 3, 2020 2:57 PM (UTC+5:30)	Oct 3, 2020 2:57 PM (UTC+5:30)
BeforeInstall	less than one second	✓ Succeeded	-	Oct 3, 2020 2:57 PM (UTC+5:30)	Oct 3, 2020 2:57 PM (UTC+5:30)
Install	less than one second	✓ Succeeded	-	Oct 3, 2020 2:57 PM (UTC+5:30)	Oct 3, 2020 2:57 PM (UTC+5:30)
AfterInstall	less than one second	✓ Succeeded	-	Oct 3, 2020 2:57 PM (UTC+5:30)	Oct 3, 2020 2:57 PM (UTC+5:30)
ApplicationStart	less than one second	✓ Succeeded	-	Oct 3, 2020 2:57 PM (UTC+5:30)	Oct 3, 2020 2:57 PM (UTC+5:30)
ValidateService	less than one second	✓ Succeeded	-	Oct 3, 2020 2:58 PM (UTC+5:30)	Oct 3, 2020 2:58 PM (UTC+5:30)

The deployment has compiled the codebase and pushed to the location specified in the appspec.yml file i.e the DocumentRoot of Java web application /var/lib/tomcat8/webapps



```
ubuntu@ip-172-31-22-147:~$ cd /var/lib/tomcat8/webapps
ubuntu@ip-172-31-22-147:/var/lib/tomcat8/webapps$ ll
total 12
drwxrwxr-x 3 tomcat8 tomcat8 4096 Oct  3 12:57 ./
drwxr-xr-x 5 root    root    4096 Oct  3 12:57 ../
drwxr-xr-x 3 root    root    4096 Oct  3 12:57 ROOT/
ubuntu@ip-172-31-22-147:/var/lib/tomcat8/webapps$ ll
total 17380
drwxrwxr-x 4 tomcat8 tomcat8     4096 Oct  3 13:07 ./
drwxr-xr-x 5 root    root        4096 Oct  3 12:57 ../
drwxr-x--- 6 tomcat8 tomcat8     4096 Oct  3 13:07 HelloWorld-1/
-rw-r--r-- 1 root    root    17774473 Oct  3 12:53 HelloWorld-1.war
drwxr-xr-x 3 root    root        4096 Oct  3 12:57 ROOT/
-rw-r--r-- 1 root    root          86 Oct  3 12:53 appspec.yml
ubuntu@ip-172-31-22-147:/var/lib/tomcat8/webapps$
```



Not secure | 18.206.176.91:8080/HelloWorld-1/

Welcome!

If you are reading this message then the installation has gone well and the application is running. Congratulations!!

Login

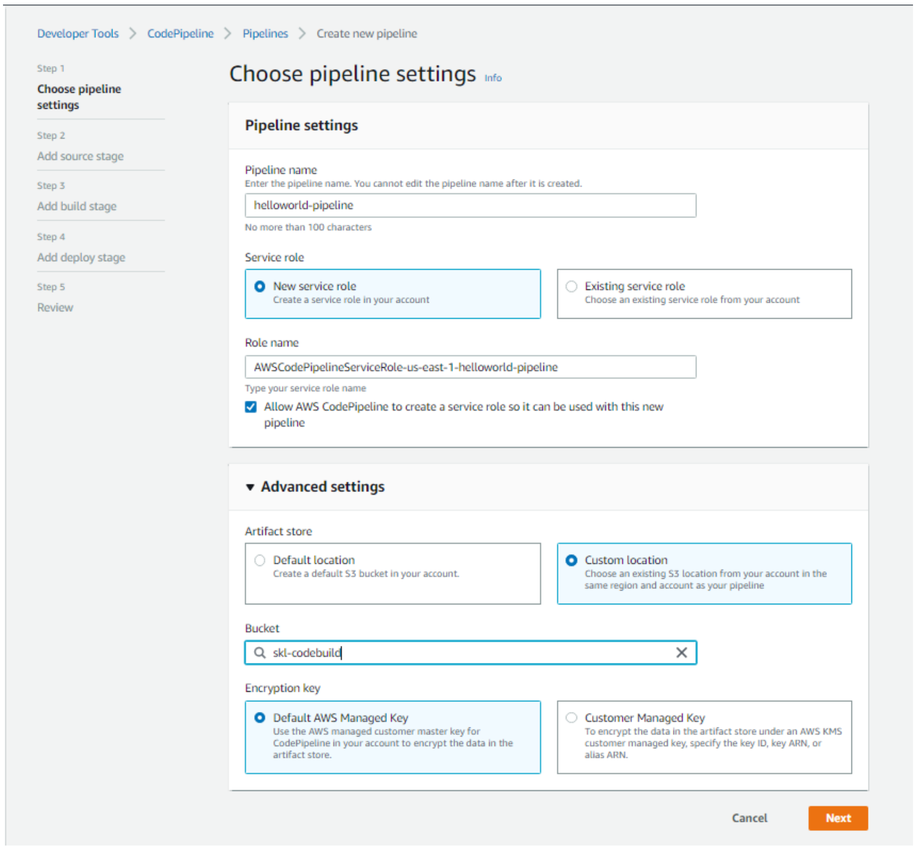
Type in your first name

Password

The password is hard coded as admin123

Application version - v1

CODE PIPELINE



Choose pipeline settings

Pipeline settings

Pipeline name
Enter the pipeline name. You cannot edit the pipeline name after it is created.
helloworld-pipeline
No more than 100 characters

Service role

● New service role
Create a service role in your account

○ Existing service role
Choose an existing service role from your account

Role name
AWSCodePipelineServiceRole-us-east-1-helloworld-pipeline
Type your service role name

☑ Allow AWS CodePipeline to create a service role so it can be used with this new pipeline

Advanced settings

Artifact store

○ Default location
Create a default S3 bucket in your account.

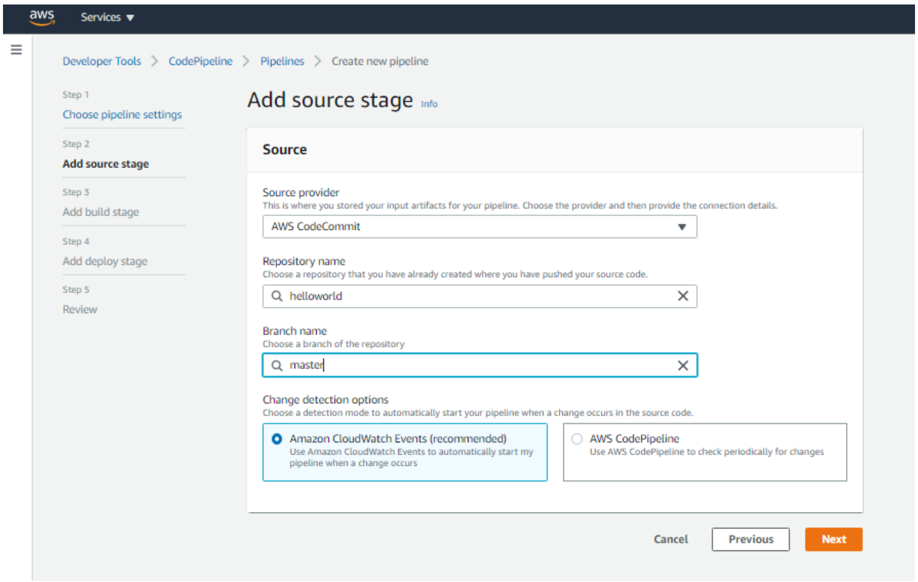
● Custom location
Choose an existing S3 location from your account in the same region and account as your pipeline

Bucket
skl-codebuild

Encryption key

● Default AWS Managed Key
Use the AWS managed customer master key for CodePipeline in your account to encrypt the data in the artifact store.

○ Customer Managed Key
To encrypt the data in the artifact store under an AWS KMS customer managed key, specify the key ID, key ARN, or alias ARN.



Add source stage

Source

Source provider
This is where you stored your input artifacts for your pipeline. Choose the provider and then provide the connection details.
AWS CodeCommit

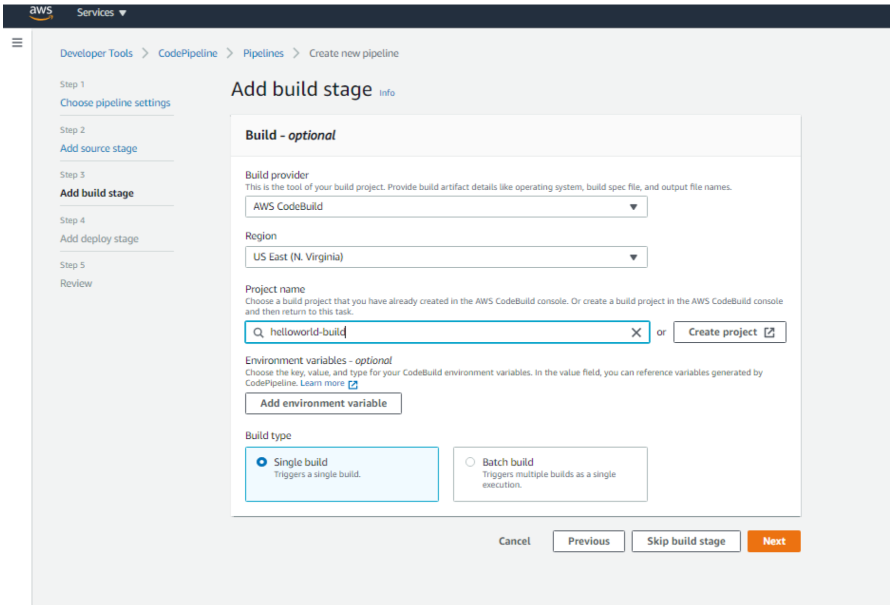
Repository name
Choose a repository that you have already created where you have pushed your source code.
helloworld

Branch name
Choose a branch of the repository
master

Change detection options
Choose a detection mode to automatically start your pipeline when a change occurs in the source code.

● Amazon CloudWatch Events (recommended)
Use Amazon CloudWatch Events to automatically start my pipeline when a change occurs

○ AWS CodePipeline
Use AWS CodePipeline to check periodically for changes



Add build stage Info

Build - *optional*

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) ▼

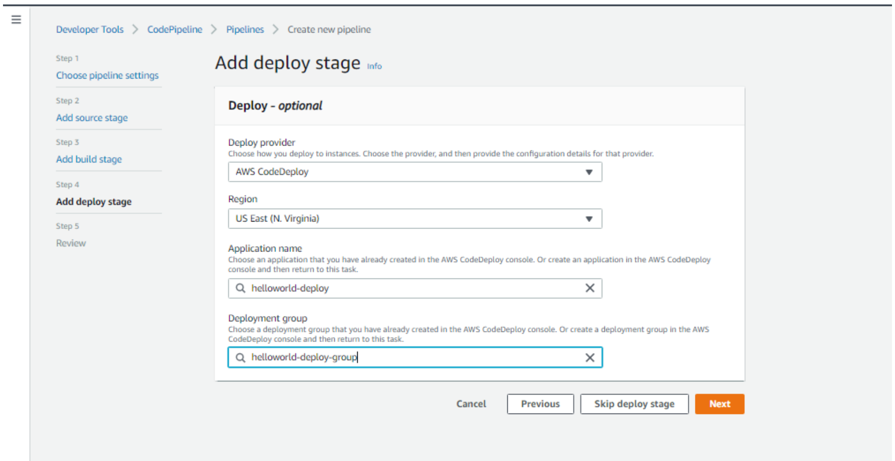
Project name
Choose a build project that you have already created in the AWS CodeBuild console. Or create a build project in the AWS CodeBuild console and then return to this task.
helloworld-build

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

Build type

● Single build
Triggers a single build.

○ Batch build
Triggers multiple builds as a single execution.



Add deploy stage Info

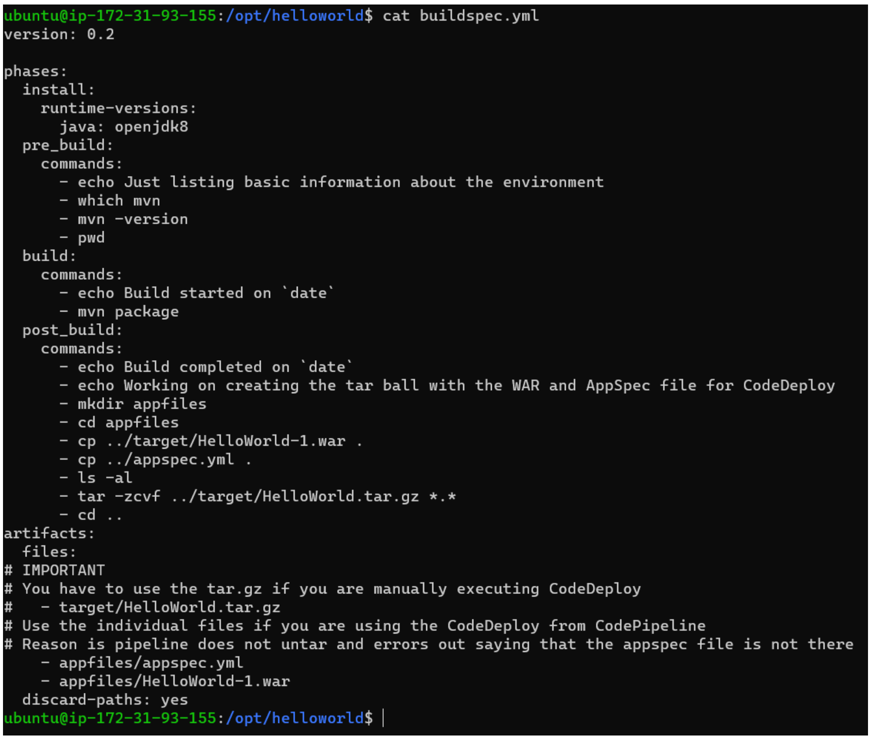
Deploy - *optional*

Deploy provider
Choose how you deploy to instances. Choose the provider, and then provide the configuration details for that provider.
AWS CodeDeploy ▼

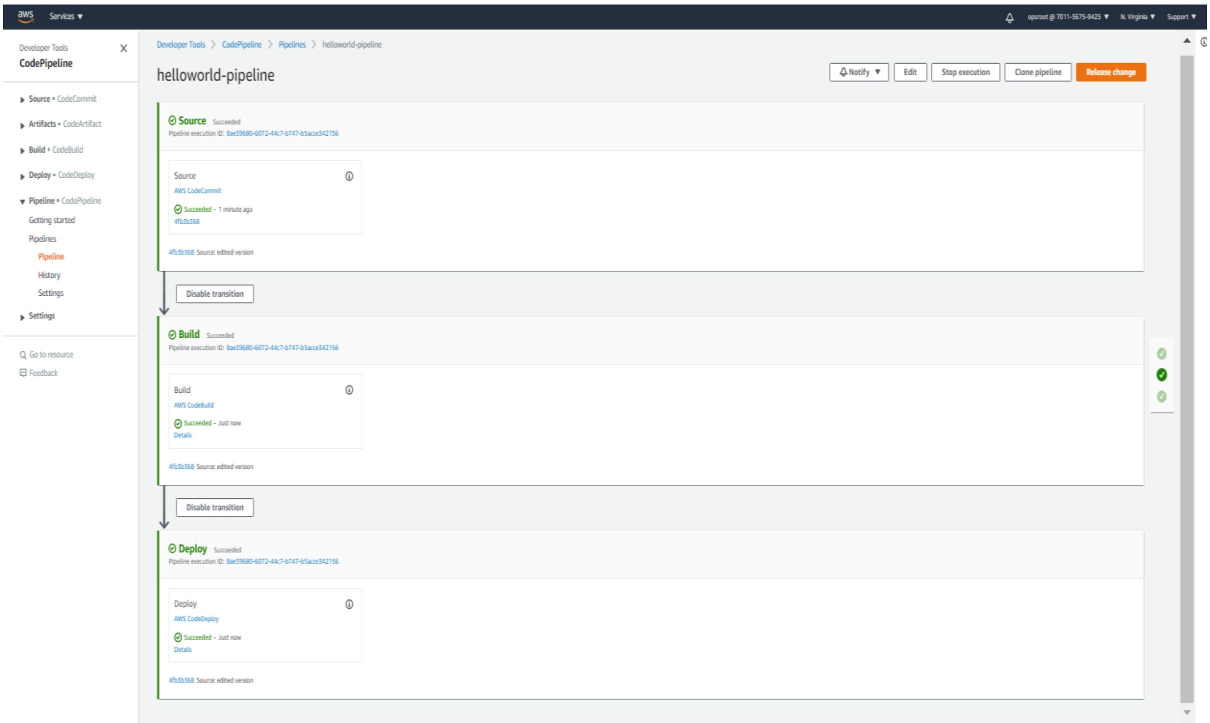
Region
US East (N. Virginia) ▼

Application name
Choose an application that you have already created in the AWS CodeDeploy console. Or create an application in the AWS CodeDeploy console and then return to this task.
helloworld-deploy

Deployment group
Choose a deployment group that you have already created in the AWS CodeDeploy console. Or create a deployment group in the AWS CodeDeploy console and then return to this task.
helloworld-deploy-group



```
ubuntu@ip-172-31-93-155:/opt/helloworld$ cat buildspec.yml
version: 0.2

phases:
  install:
    runtime-versions:
      java: openjdk8
  pre_build:
    commands:
      - echo Just listing basic information about the environment
      - which mvn
      - mvn -version
      - pwd
  build:
    commands:
      - echo Build started on `date`
      - mvn package
  post_build:
    commands:
      - echo Build completed on `date`
      - echo Working on creating the tar ball with the WAR and AppSpec file for CodeDeploy
      - mkdir appfiles
      - cd appfiles
      - cp ../target/HelloWorld-1.war .
      - cp ../appspec.yml .
      - ls -al
      - tar -zcvf ../target/HelloWorld.tar.gz *.*
      - cd ..
artifacts:
  files:
# IMPORTANT
# You have to use the tar.gz if you are manually executing CodeDeploy
#    - target/HelloWorld.tar.gz
# Use the individual files if you are using the CodeDeploy from CodePipeline
# Reason is pipeline does not untar and errors out saying that the appspec file is not there
    - appfiles/appspec.yml
    - appfiles/HelloWorld-1.war
  discard-paths: yes
ubuntu@ip-172-31-93-155:/opt/helloworld$ 
```



helloworld-pipeline

✓ Source Succeeded
Pipeline execution ID: 8ae39680-6072-44c7-b747-b5acce342156

Source
AWS CodeCommit
✓ Succeeded - 1 minute ago
4fb3b368

4fb3b368 Source: edited version

✓ Build Succeeded
Pipeline execution ID: 8ae39680-6072-44c7-b747-b5acce342156

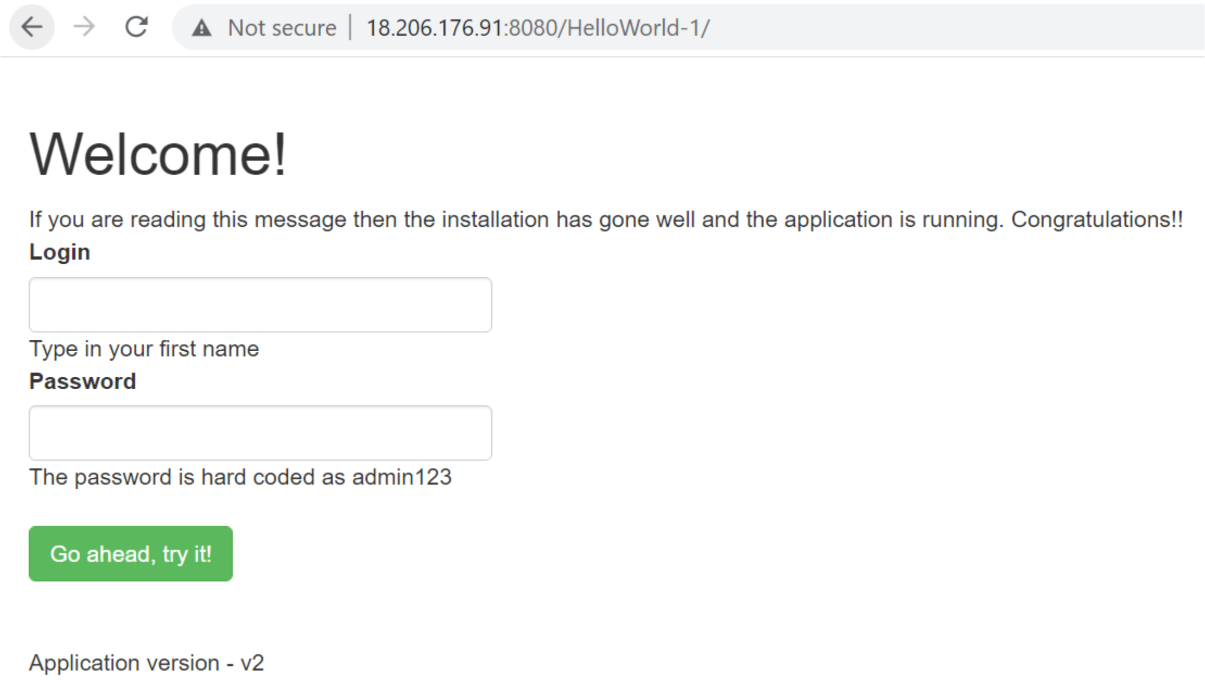
Build
AWS CodeBuild
✓ Succeeded - Just now
Details

4fb3b368 Source: edited version

✓ Deploy Succeeded
Pipeline execution ID: 8ae39680-6072-44c7-b747-b5acce342156

Deploy
AWS CodeDeploy
✓ Succeeded - Just now
Details

4fb3b368 Source: edited version



18.206.176.91:8080/HelloWorld-1/

Welcome!

If you are reading this message then the installation has gone well and the application is running. Congratulations!!

Login

Type in your first name

Password

The password is hard coded as admin123

Application version - v2