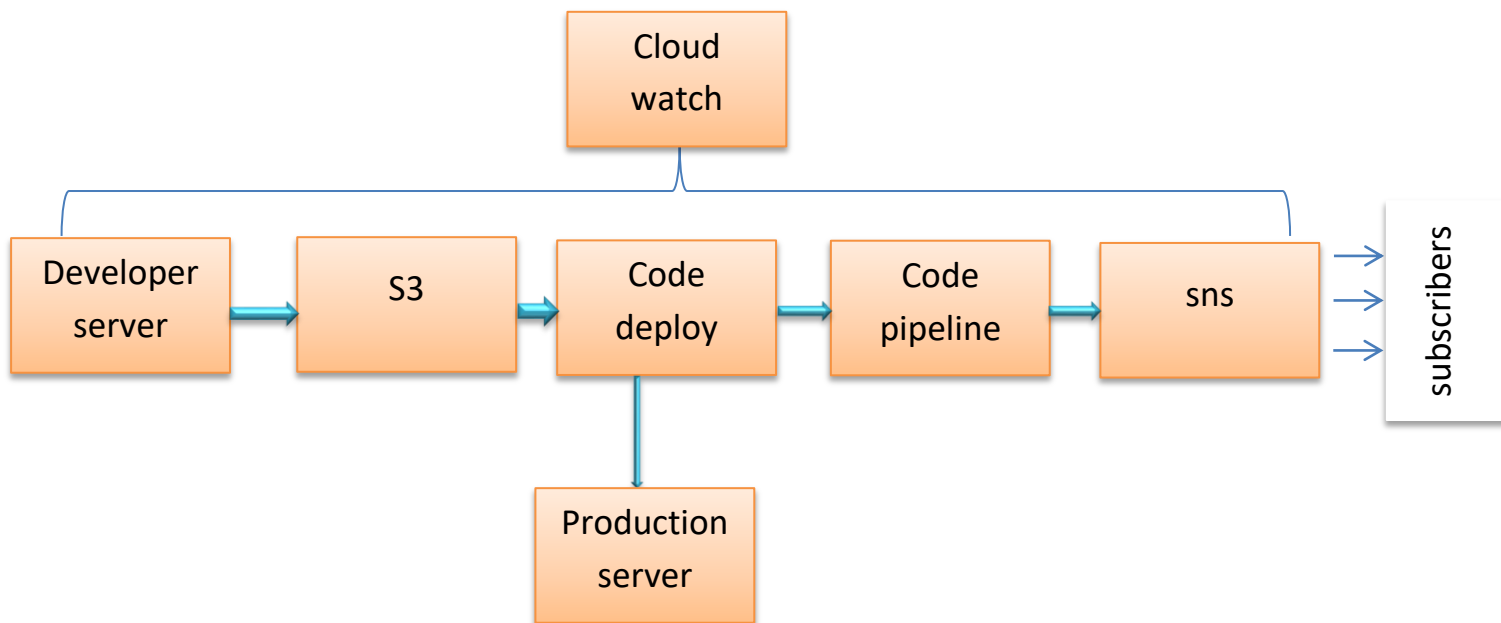


CODE DEPLOYED AND CODE PIPELINE



Code deployed

CodeDeploy is a deployment service that automates application deployments to Amazon EC2 instances, on-premises instances, serverless Lambda functions, or Amazon ECS services.

Steps to creat code deploy

Step1:create user

Iam--->add user--->name(any)---->acces key only tick---->next

The screenshot shows the 'Add user' console in the AWS IAM service. The page has a progress bar at the top with steps 1 through 5, where step 1 is currently active. The main section is titled 'Set user details' and includes a sub-header: 'You can add multiple users at once with the same access type and permissions. [Learn more](#)'. Below this, there is a 'User name*' field with the value 'seelan' and a link to 'Add another user'. The 'Select AWS access type' section explains that users can be configured for programmatic access or console access. Under 'Select AWS credential type*', the 'Access key - Programmatic access' option is selected with a checkbox, while the 'Password - AWS Management Console access' option is unselected. The bottom of the page features a '* Required' label, a 'Cancel' button, and a 'Next: Permissions' button.

```
Set permissions---->attach existing policy(administrator acces)---->next-->next-
--->next--->creat user
```

Add user

1

2

3

4

5

Set permissions

Add user to group

Copy permissions from existing user

Attach existing policies directly

Create policy

Filter policies

Search

Showing 809 results

	Policy name	Type	Used as
<input checked="" type="checkbox"/>	AdministratorAccess	Job function	None
<input type="checkbox"/>	AdministratorAccess-Amplify	AWS managed	None
<input type="checkbox"/>	AdministratorAccess-AWSElasticBeanstalk	AWS managed	None
<input type="checkbox"/>	AlexaForBusinessDeviceSetup	AWS managed	None
<input type="checkbox"/>	AlexaForBusinessFullAccess	AWS managed	None
<input type="checkbox"/>	AlexaForBusinessGatewayExecution	AWS managed	None
<input type="checkbox"/>	AlexaForBusinessLifesizeDelegatedAccessPolicy	AWS managed	None
<input type="checkbox"/>	AlexaForBusinessPolyDelegatedAccessPolicy	AWS managed	None
<input type="checkbox"/>	AlexaForBusinessReadOnlyAccess	AWS managed	None
<input type="checkbox"/>	AmazonAPIGatewayAdministrator	AWS managed	None
<input type="checkbox"/>	AmazonAPIGatewayInvokeFullAccess	AWS managed	None
<input type="checkbox"/>	AmazonAPIGatewayPushToCloudWatchLogs	AWS managed	None
<input type="checkbox"/>	AmazonAppFlowFullAccess	AWS managed	None

Cancel

Previous

Next: Tags

User created---->download ccv

Add user

12345


✓

Success

You successfully created the users shown below. You can view and download user security credentials. You can also email users instructions for signing in to the AWS Management Console. This is the last time these credentials will be available to download. However, you can create new credentials at any time.

Users with AWS Management Console access can sign-in at: <https://795071115491.signin.aws.amazon.com/console>

Download .csv

	User	Access key ID	Secret access key
▶	✓ seetan	AKIA3SHPOODRVLKWKSTK	 ***** Show

We get acces key and secret acces key

[illegible]

Step2:create ec2 role

Roles---->create role--->trusted entity type(aws service)---->use case(Ec2)---->next

Select trusted entity [info](#)

Trusted entity type

☒ AWS service
Allow AWS services like EC2, Lambda, or others to perform actions in this account.

☐ AWS account
Allow entities in other AWS accounts belonging to you or a 3rd party to perform actions in this account.

☐ Web identity
Allows users federated by the specified external web identity provider to assume this role to perform actions in this account.

☐ SAML 2.0 federation
Allow users federated with SAML 2.0 from a corporate directory to perform actions in this account.

☐ Custom trust policy
Create a custom trust policy to enable others to perform actions in this account.

Use case
Allow an AWS service like EC2, Lambda, or others to perform actions in this account.

Common use cases

☒ EC2
Allows EC2 instances to call AWS services on your behalf.

☐ Lambda
Allows Lambda functions to call AWS services on your behalf.

Use cases for other AWS services:
Choose a service to view use case

Cancel Next

Permission policies---->administrator acces-->next

Permissions policies (Selected 1/809) [info](#)

Choose one or more policies to attach to your new role.

Filter policies by property or policy name and press enter. 14 matches

administrator X Clear filters

Policy name	Type	Description
AWSSSODirectoryAdministrator	AWS managed	Administrator access for SSO Directory
DatabaseAdministrator	AWS managed - job function	Grants full access permissions to AWS services and actions required to set up and configure AWS database services.
AWSSSOMasterAccountAdministrator	AWS managed	Provides access within AWS SSO to manage AWS Organizations master and member accounts and cloud application
AWSCloud9Administrator	AWS managed	Provides administrator access to AWS Cloud9.
AWSSSOMemberAccountAdministrator	AWS managed	Provides access within AWS SSO to manage AWS Organizations member accounts and cloud application
SystemAdministrator	AWS managed - job function	Grants full access permissions necessary for resources required for application and development operations.
AdministratorAccess	AWS managed - job function	Provides full access to AWS services and resources.
AmazonAPIGatewayAdministrator	AWS managed	Provides full access to create/edit/delete APIs in Amazon API Gateway via the AWS Management Console.
AWSAppSyncAdministrator	AWS managed	Provides administrative access to the AppSync service, though not enough to access via the console.
NetworkAdministrator	AWS managed - job function	Grants full access permissions to AWS services and actions required to set up and configure AWS network resources.
AdministratorAccess-Amplify	AWS managed	Grants account administrative permissions while explicitly allowing direct access to resources needed by Amplify applications.
AdministratorAccess-AWSElasticBeanstalk	AWS managed	Grants account administrative permissions. Explicitly allows developers and administrators to gain direct access to resources they need to ma...
AWSAuditManagerAdministratorAccess	AWS managed	Provides administrative access to enable or disable AWS Audit Manager, update settings, and manage assessments, controls, and frameworks
AWSGrafanaAccountAdministrator	AWS managed	Provides access within Amazon Grafana to create and manage workspaces for the entire organization.

Set permissions boundary - optional [info](#)

Set a permissions boundary to control the maximum permissions this role can have. This is not a common setting, but you can use it to delegate permission management to others.

Cancel Previous Next

Role name(ec2role)---->create

Name, review, and create

Role details

Role name
ec2role

Description
Allows EC2 instances to call AWS services on your behalf

Step 1: Select trusted entities

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": "sts:AssumeRole",
      "Principal": {
        "Service": "ec2.amazonaws.com"
      }
    }
  ]
}
```

Step 2: Add permissions

Permissions policy selector

Ec2 role createdddd..

Step3:create code deploy role

```
Roles---->create role--->trusted entity type(aws service)---->use case(code deploy)----
>next
```

Step 1

Select trusted entity

Step 2

EC2 permissions

Step 3

EC2 permissions

Step 4

Custom, review, and create

Select trusted entity info

Trusted entity type

☒ **AWS service**
Allow AWS services like EC2, Lambda, or others to perform actions in this account.

☐ **AWS account**
Allow actions in other AWS accounts belonging to you or a third party to perform actions in this account.

☐ **Web identity**
Allow users federated by the specified external web identity provider to assume this role to perform actions in this account.

☐ **SAML 2.0 federation**
Allow users federated with SAML 2.0 from a corporate directory to perform actions in this account.

☐ **Custom trust policy**
Create a custom trust policy to enable others to perform actions in this account.

Use case

Allow an AWS service like EC2, Lambda, or others to perform actions in this account.

Common use cases

☐ **EC2**
Allow EC2 instances to call AWS services on your behalf.

☐ **Lambda**
Allow Lambda functions to call AWS services on your behalf.

Use cases for other AWS services:

CodeDeploy

☒ **CodeDeploy**
Allow CodeDeploy to call AWS services with an AWS IAM role on your behalf.

☐ **CodeDeploy for Lambda**
Allow CodeDeploy to make builds to a new version of an AWS Lambda function on your behalf.

☐ **CodeDeploy - ECS**
Allow CodeDeploy to read ECS objects, invoke Lambda functions, publish to SNS topics, and update EC2 instances on your behalf.

Permission policies---->default permission--->next

IAM > Roles > Create role

Step 1
Select trusted entity


Step 2
Add permissions

Step 3
Name, review, and create

Add permissions [Info](#)

Permissions policies (1) [Info](#)

The type of role that you selected requires the following policy.

Policy name ?	Type ?	Attached entities
 AWSCodeDeployRole	AWS m...	0

[▶ Set permissions boundary - optional](#) [Info](#)

Set a permissions boundary to control the maximum permissions this role can have. This is not a common setting, but you can use it to delegate permission management to others.

Cancel

Previous

Next

Role name(codedeployrole)---->create

The screenshot shows the AWS IAM console interface for creating a new role. The breadcrumb navigation at the top indicates the path: IAM > Roles > Create role. The process is divided into three steps: Step 1: Select trusted entity, Step 2: Add permissions, and Step 3: Name, review, and create. The current step is Step 1, which is titled 'Name, review, and create'. The 'Role name' field contains 'codeDeployRole'. The 'Description' field contains 'Allows CodeDeploy to call AWS services such as Auto Scaling on your behalf.' The 'Step 1: Select trusted entities' section shows a list of trusted entities, with the first one being 'codeDeployRole'.

IAM > Roles > Create role

Step 1
Select trusted entity

Step 2
Add permissions

Step 3
Name, review, and create

Role details

Role name
Enter a meaningful name to identify this role.

Maximum 64 characters. Use alphanumeric and "-", "@", "_" characters.

Description
Add a short explanation for this role.

Maximum 1000 characters. Use alphanumeric and "-", "@", "_" characters.

Step 1: Select trusted entities

```

1  {
2    "Version": "2012-10-17",
3    "Statement": [
4      {
5        "Sid": "",
6        "Effect": "Allow",
7        "Principal": {
8          "Service": [
9            "codeDeploy.amazonaws.com"
10           ]
11        },
12        "Action": [
13          "sts:AssumeRole"
14        ]
15      }
16    ]
17  }

```

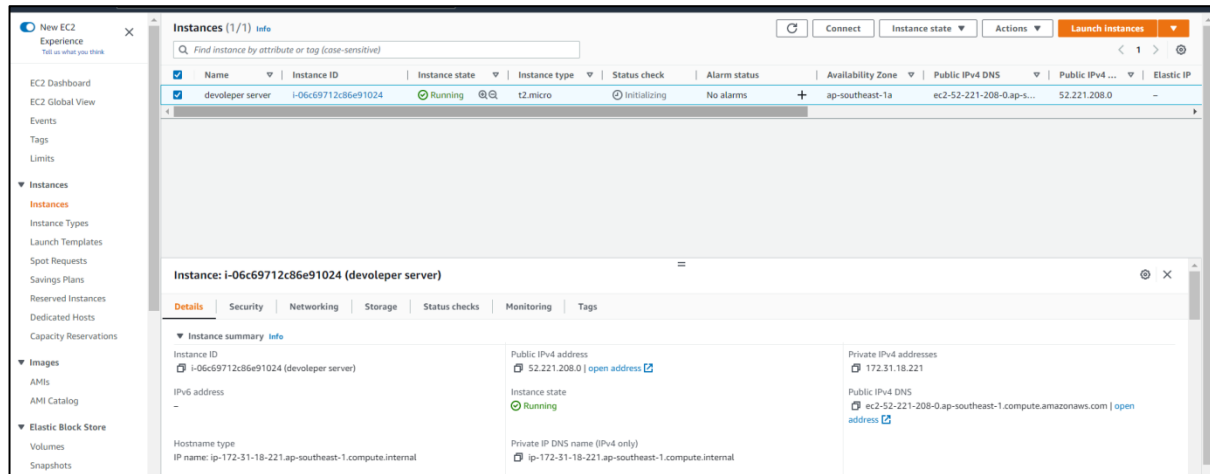
Edit

```
code deploy role created..
```

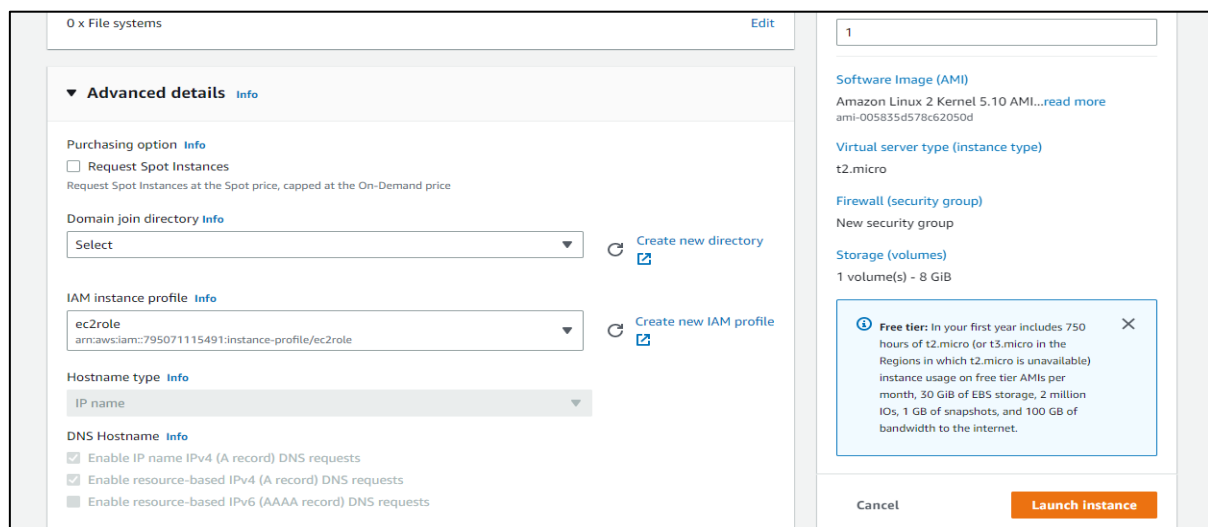
The screenshot shows the AWS IAM console's 'Roles' page. The 'Roles (Selected 2/18)' section is expanded, displaying a list of roles. The 'AWSCodePipelineServiceRole-1-deployrole' role is highlighted. The 'Trusted entities' column shows the role is trusted by 'AWS Service: codepipeline'. The 'Last activity' column shows the role was last used '10 hours ago'.

Step4:launch ec2 instance

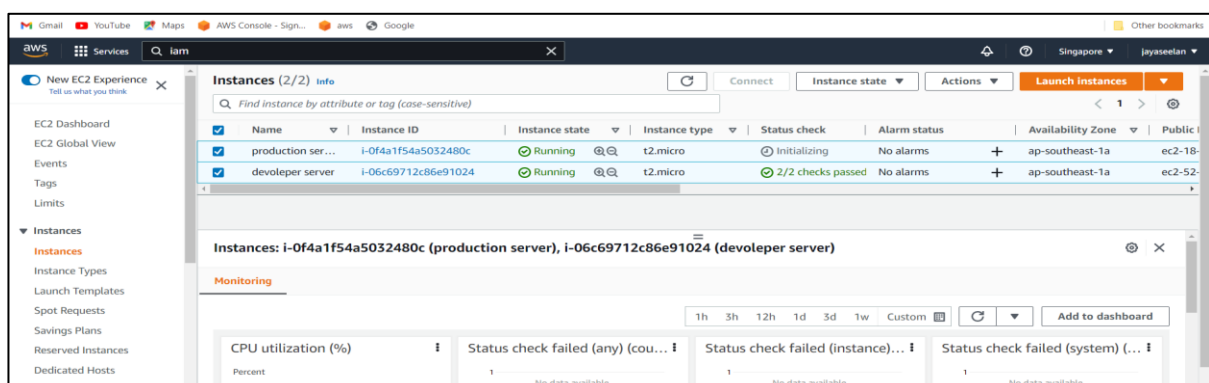
Step4.1: Name(developer server)---->linux os----->security group(ssh &http)--->launch instance



Step4.2: Name(production server)---->linux os----->security group(ssh &http)--->advanced details(iam role--->ec2 role attach)----->launch instance

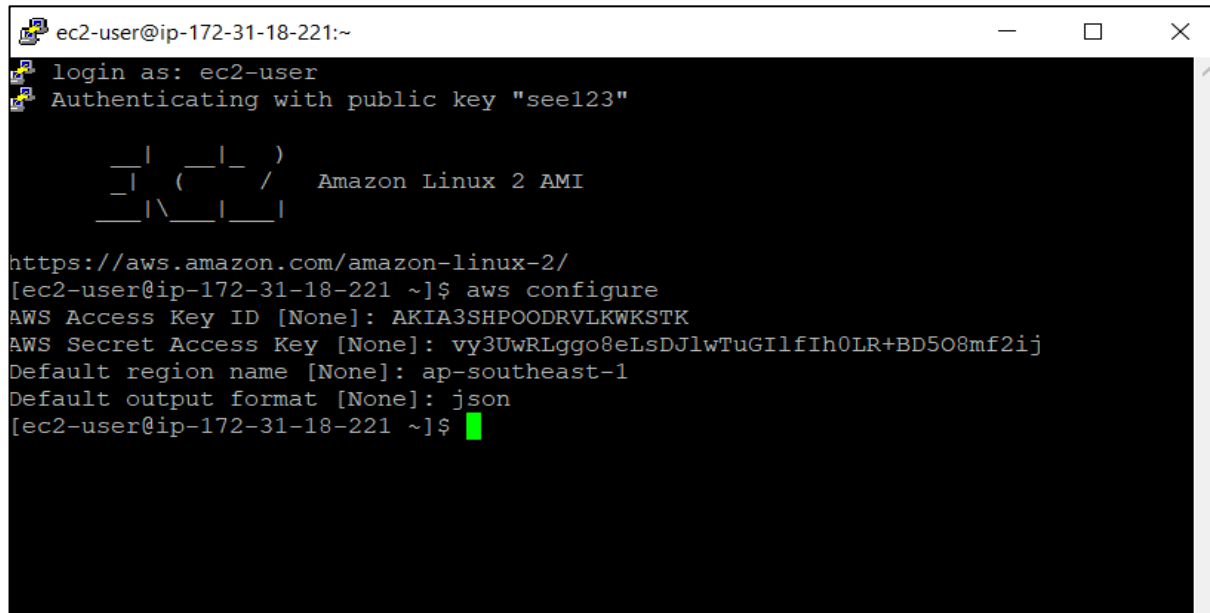


Developer and production server launched..



Step5:login developer server

Step5.1: Puuty open--->login(ec2-user)--->aws configure(acces key ,secret acces, key,zone,json)

A terminal window titled 'ec2-user@ip-172-31-18-221:~' showing the login process for 'ec2-user' using a public key 'see123'. It displays the Amazon Linux 2 AMI logo and the URL 'https://aws.amazon.com/amazon-linux-2/'. The user then runs 'aws configure', which prompts for the AWS Access Key ID (AKIA3SHPOODRVLKWKSTK), AWS Secret Access Key (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), Default region name (ap-southeast-1), and Default output format (json).

```
ec2-user@ip-172-31-18-221:~  
login as: ec2-user  
Authenticating with public key "see123"  
  
  _| _| _| _|  
  _| ( _| _| /  Amazon Linux 2 AMI  
  _| \ _| _| _|  
  
https://aws.amazon.com/amazon-linux-2/  
[ec2-user@ip-172-31-18-221 ~]$ aws configure  
AWS Access Key ID [None]: AKIA3SHPOODRVLKWKSTK  
AWS Secret Access Key [None]: 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  
Default region name [None]: ap-southeast-1  
Default output format [None]: json  
[ec2-user@ip-172-31-18-221 ~]$
```

Step5.2:make directory and put html content

#mkdir deploy_dir

#cd deploy_dir

#mkdir sampleapp


#cd sampleapp

#vi index.html

<html>

<h2> Sample App Version 1 </h2>

</html>

A terminal window titled 'ec2-user@ip-172-31-18-221:~/deploy_dir/sampleapp' showing the creation of 'index.html'. The user enters the HTML content: '<html>', '<h2> Sample App Version 1 </h2>', and '</html>'. The status bar at the bottom shows '"index.html" 3L, 47B' and '3,7 All'.

```
ec2-user@ip-172-31-18-221:~/deploy_dir/sampleapp  
<html>  
<h2> Sample App Version 1 </h2>  
</html>  
  
"index.html" 3L, 47B 3,7 All
```

#vi appspec.yml

version: 0.0

os: linux

files:

- source: /index.html

destination: /var/www/html/

hooks:

BeforeInstall:

- location: scripts/httpd_install.sh

timeout: 300

runas: root

- location: scripts/httpd_start.sh

timeout: 300

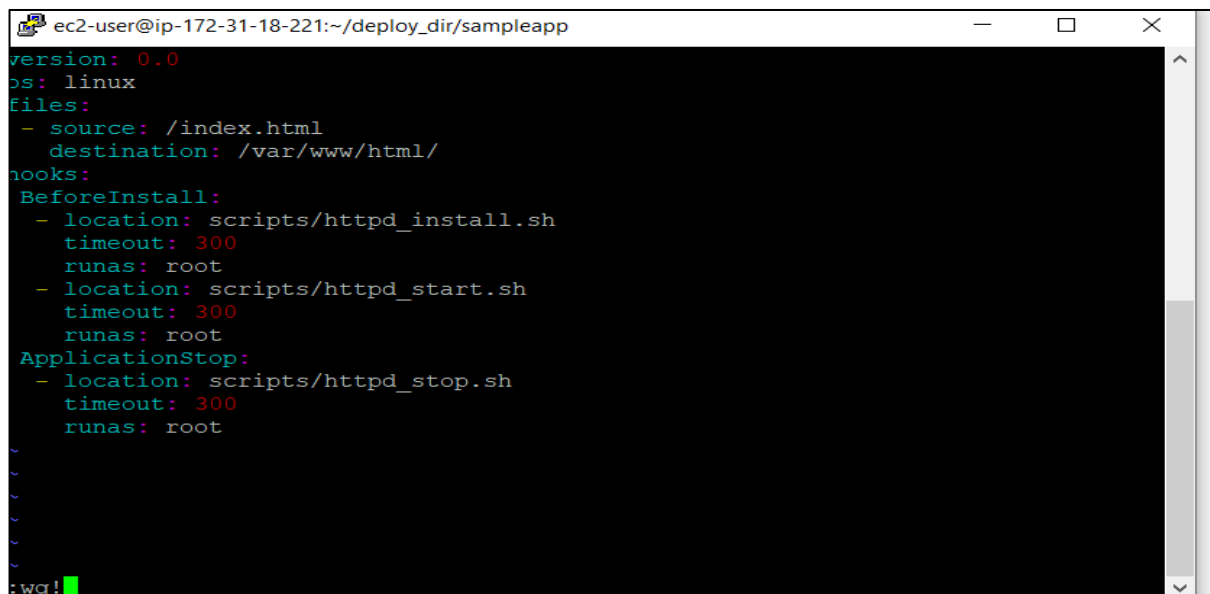
runas: root

ApplicationStop:

- location: scripts/httpd_stop.sh

timeout: 300

runas: root

A screenshot of a terminal window with a title bar showing 'ec2-user@ip-172-31-18-221:~/deploy_dir/sampleapp'. The terminal displays the content of the 'appspec.yml' file in a color-coded syntax. The text is as follows:

```
version: 0.0
os: linux
files:
  - source: /index.html
    destination: /var/www/html/
hooks:
  BeforeInstall:
    - location: scripts/httpd_install.sh
      timeout: 300
      runas: root
    - location: scripts/httpd_start.sh
      timeout: 300
      runas: root
  ApplicationStop:
    - location: scripts/httpd_stop.sh
      timeout: 300
      runas: root
```

At the bottom of the terminal, there is a prompt ':wq!' followed by a green cursor block. The terminal window has standard Linux window controls (minimize, maximize, close) on the right side.

```

ec2-user@ip-172-31-18-221:~/deploy_dir/sampleapp
login as: ec2-user
Authenticating with public key "see123"

  _|_  _|_  )
 _|_  ( _|_ /   Amazon Linux 2 AMI
 _|_ \ _|_ |

https://aws.amazon.com/amazon-linux-2/
[ec2-user@ip-172-31-18-221 ~]$ aws configure
AWS Access Key ID [None]: AKIA3SHPOODRVLKWKSTK
AWS Secret Access Key [None]: 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
Default region name [None]: ap-southeast-1
Default output format [None]: json
[ec2-user@ip-172-31-18-221 ~]$ mkdir deploy_dir
[ec2-user@ip-172-31-18-221 ~]$ cd deploy_dir
[ec2-user@ip-172-31-18-221 deploy_dir]$ mkdir sampleapp
[ec2-user@ip-172-31-18-221 deploy_dir]$ cd sampleapp
[ec2-user@ip-172-31-18-221 sampleapp]$ vi index.html
[ec2-user@ip-172-31-18-221 sampleapp]$ vi index.html
[ec2-user@ip-172-31-18-221 sampleapp]$ vi appspec.yml
[ec2-user@ip-172-31-18-221 sampleapp]$ LL
-bash: LL: command not found
[ec2-user@ip-172-31-18-221 sampleapp]$ ll
total 8
-rw-rw-r-- 1 ec2-user ec2-user 336 Jan  7 14:47 appspec.yml
-rw-rw-r-- 1 ec2-user ec2-user  47 Jan  7 14:44 index.html
[ec2-user@ip-172-31-18-221 sampleapp]$

```

#mkdir scripts

#cd scripts

#vi httpd_install.sh

```
#!/bin/bash
```

```
yum install -y httpd
```

#vi httpd_start.sh

```
#!/bin/bash
```

```
systemctl start httpd
```

#vi httpd_stop.sh

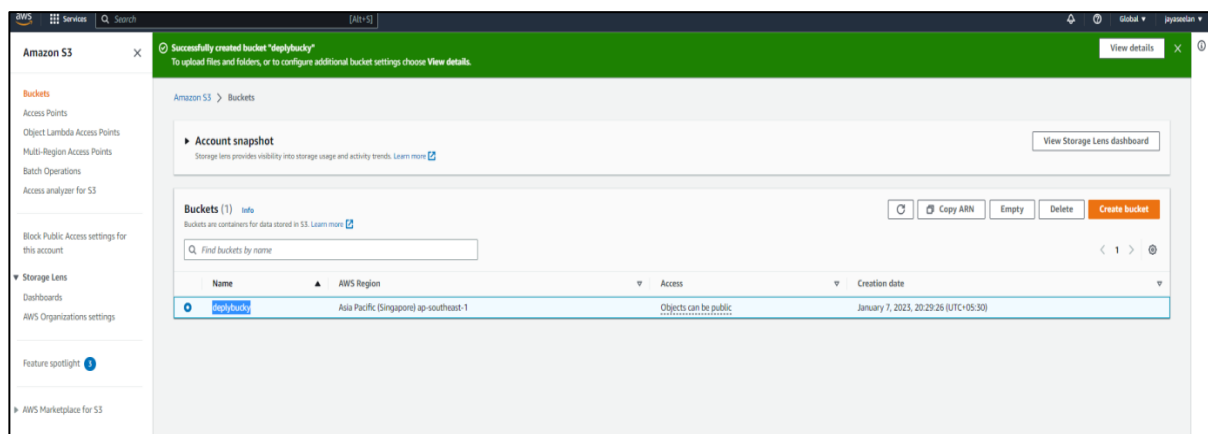
```
#!/bin/bash
```

```
systemctl stop httpd
```

now give permission **#chmod 777 ***


```
ec2-user@ip-172-31-18-221 sampleapp]$ mkdir scripts
ec2-user@ip-172-31-18-221 sampleapp]$ cd scripts
bash: cd: scripts: No such file or directory
ec2-user@ip-172-31-18-221 sampleapp]$ cd scripts
ec2-user@ip-172-31-18-221 scripts]$ vi httpd_install.sh
ec2-user@ip-172-31-18-221 scripts]$ vi httpd_start.sh
ec2-user@ip-172-31-18-221 scripts]$ vi httpd_stop.sh
ec2-user@ip-172-31-18-221 scripts]$ ll
total 12
-rw-rw-r-- 1 ec2-user ec2-user 33 Jan  7 14:52 httpd_install.sh
-rw-rw-r-- 1 ec2-user ec2-user 34 Jan  7 14:53 httpd_start.sh
-rw-rw-r-- 1 ec2-user ec2-user 33 Jan  7 14:53 httpd_stop.sh
ec2-user@ip-172-31-18-221 scripts]$ chmod 777 *
ec2-user@ip-172-31-18-221 scripts]$ ll
total 12
-rwxrwxrwx 1 ec2-user ec2-user 33 Jan  7 14:52 httpd_install.sh
-rwxrwxrwx 1 ec2-user ec2-user 34 Jan  7 14:53 httpd_start.sh
-rwxrwxrwx 1 ec2-user ec2-user 33 Jan  7 14:53 httpd_stop.sh
ec2-user@ip-172-31-18-221 scripts]$
```

```
Bucket---->create bucket---->name(any)---->acl enabeled--->public acces--->bucket
versioning(enabled)--->create bucket
```



Step7:creat application using command(code deploy)

Step7.1:Open code deploy service--->application

Step7.2:application create using command--->cd scripts--->

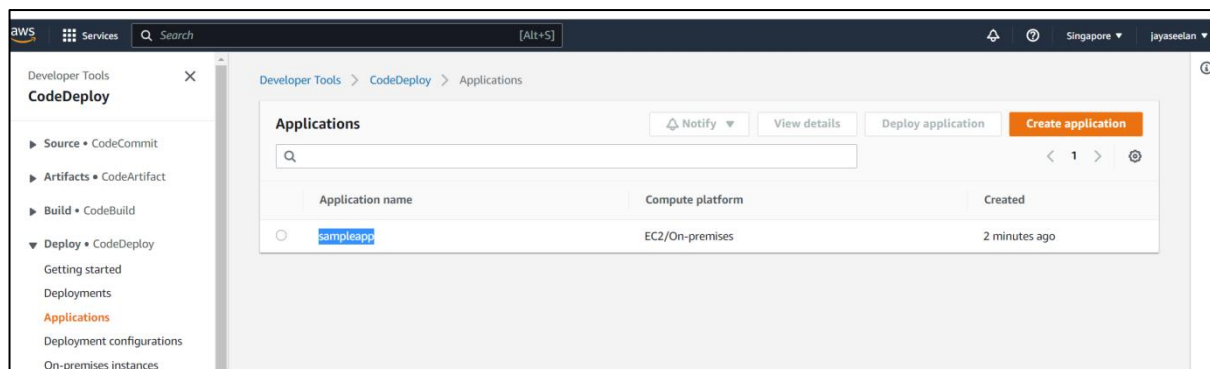
```
# aws deploy create-application --application-name sampleapp
```

```
[ec2-user@ip-172-31-18-221 scripts]$ chmod 777 *
[ec2-user@ip-172-31-18-221 scripts]$ ll
total 12
-rwxrwxrwx 1 ec2-user ec2-user 33 Jan  7 14:52 httpd_install.sh
-rwxrwxrwx 1 ec2-user ec2-user 34 Jan  7 14:53 httpd_start.sh
-rwxrwxrwx 1 ec2-user ec2-user 33 Jan  7 14:53 httpd_stop.sh
[ec2-user@ip-172-31-18-221 scripts]$
login as: ec2-user
Authenticating with public key "see123"
Last login: Sat Jan  7 14:37:45 2023 from 157.51.66.55

 _ _ | _ _ | _ )
 _ | ( _ _ /   Amazon Linux 2 AMI
 _ | \ _ | _ |

https://aws.amazon.com/amazon-linux-2/
[ec2-user@ip-172-31-18-221 ~]$
[ec2-user@ip-172-31-18-221 ~]$ aws deploy create-application --application-name sampleapp
{
  "applicationId": "16103d6b-5b99-492b-9bb5-5b9385230556"
}
[ec2-user@ip-172-31-18-221 ~]$
```

Step7.3:now check code deploy--->application--->one application will show



Step8:creating all files zip and put s3 bucket

```
#cd sampleapp----->
```

```
# aws deploy push --application-name sampleapp --s3-location s3://  
deploybucky/sampleapp.zip
```

```
login as: ec2-user
Authenticating with public key "see123"
Last login: Sat Jan  7 15:02:58 2023 from 157.51.66.55

 _ _ | _ _ | _ )
 _ | ( _ _ /   Amazon Linux 2 AMI
 _ | \ _ | _ |

https://aws.amazon.com/amazon-linux-2/
ec2-user@ip-172-31-18-221 ~]$ cd sampleapp
bash: cd: sampleapp: No such file or directory
ec2-user@ip-172-31-18-221 ~]$ ll
total 0
ec2-user@ip-172-31-18-221 sampleapp$ cd deploy_dir
ec2-user@ip-172-31-18-221 sampleapp$ ll
total 0
ec2-user@ip-172-31-18-221 sampleapp$ aws deploy push --application-name sampleapp --s3-location s3://deploybucky/sampleapp.zip
o deploy with this revision, run:
aws deploy create-deployment --application-name sampleapp --s3-location bucket=deploybucky,key=sampleapp.zip,bundleType=zip,eTag=204f657cd8c8867fda76f5falce9829a,version=.2PdpDylrFulN_WpbOJZ
Dag1jNIXMNR --deployment-group-name <deployment-group-name> --deployment-config-name <deployment-config-name> --description <description>
ec2-user@ip-172-31-18-221 sampleapp$
```

```
#cd deploy_dir----> # zip -r ../sampleapp.zip .
```

```
To deploy with this revision, run:
aws deploy create-deployment --application-name sampleapp --s3-location bucket=deplybucky,ke
ldxGljNIXnWR --deployment-group-name <deployment-group-name> --deployment-config-name <deplo
[ec2-user@ip-172-31-18-221 sampleapp]$ cd ..
[ec2-user@ip-172-31-18-221 deploy_dir]$ # zip -r ../sampleapp.zip .
[ec2-user@ip-172-31-18-221 deploy_dir]$ ll
total 0
drwxrwxr-x 3 ec2-user ec2-user 58 Jan  7 14:51 sampleapp
[ec2-user@ip-172-31-18-221 deploy_dir]$ zip -r ../sampleapp.zip .
  adding: sampleapp/ (stored 0%)
  adding: sampleapp/index.html (deflated 9%)
  adding: sampleapp/appspec.yml (deflated 53%)
  adding: sampleapp/scripts/ (stored 0%)
  adding: sampleapp/scripts/httpd_install.sh (stored 0%)
  adding: sampleapp/scripts/httpd_start.sh (stored 0%)
  adding: sampleapp/scripts/httpd_stop.sh (stored 0%)
[ec2-user@ip-172-31-18-221 deploy_dir]$
```

```
#cd.. ---->#aws s3 cp sampleapp.zip s3://aws280921(bucket nname--->deplybucky)
```

```
[ec2-user@ip-172-31-18-221 deploy_dir]$ ll
total 0
drwxrwxr-x 3 ec2-user ec2-user 58 Jan  7 14:51 sampleapp
[ec2-user@ip-172-31-18-221 deploy_dir]$ zip -r ../sampleapp.zip .
  adding: sampleapp/ (stored 0%)
  adding: sampleapp/index.html (deflated 9%)
  adding: sampleapp/appspec.yml (deflated 53%)
  adding: sampleapp/scripts/ (stored 0%)
  adding: sampleapp/scripts/httpd_install.sh (stored 0%)
  adding: sampleapp/scripts/httpd_start.sh (stored 0%)
  adding: sampleapp/scripts/httpd_stop.sh (stored 0%)
[ec2-user@ip-172-31-18-221 deploy_dir]$ cd ..
[ec2-user@ip-172-31-18-221 ~]$ aws s3 cp sampleapp.zip s3://deplybucky
upload: ./sampleapp.zip to s3://deplybucky/sampleapp.zip
[ec2-user@ip-172-31-18-221 ~]$
```

Finally it will upload,now check bucket object(2 zip format object shown)

The screenshot shows the Amazon S3 console interface. On the left is a navigation sidebar with options like Buckets, Access Points, and Storage Lens. The main area displays the 'deplybucky' bucket. Below the bucket name, there are tabs for Objects, Properties, Permissions, Metrics, Management, and Access Points. The 'Objects' tab is active, showing a list of two objects:

	Name	Type	Version ID	Last modified	Size	Storage class
<input type="checkbox"/>	sampleapp.zip	zip	ETUb8I7J5ObFwdKRRU4tRn1vVYQ51tW	January 7, 2023, 20:45:40 (UTC+05:30)	1.5 KB	Standard
<input type="checkbox"/>	sampleapp.zip	zip	.2PdpOylrFu1N_WpbOJZlDxGijNIXnWR	January 7, 2023, 20:41:00 (UTC+05:30)	885.0 B	Standard

Step9:login production server

```
#sudo -i

#yum install ruby -y

# wget https://aws-codedeploy-us-east-1.s3.amazonaws.com/latest/install

# chmod +x install

#./install auto

# service codedeploy-agent status
```

```

  ____  __
 / ___/  / /
/ /   /  / /
/ /___/  / /
\____/___/ /

Amazon Linux 2 AMI

https://aws.amazon.com/amazon-linux-2/
[ec2-user@ip-172-31-29-67 ~]$ sudo -i
[root@ip-172-31-29-67 ~]# yum install ruby -y
Loaded plugins: extras_suggestions, langpacks, priorities, update-motd
amzn2-core                               | 3.7 kB    00:00
Package ruby-2.0.0.648-36.amzn2.0.3.x86_64 already installed and latest version
Nothing to do
[root@ip-172-31-29-67 ~]# wget https://aws-codedeploy-us-east-1.s3.amazonaws.com/latest/install
--2023-01-07 15:21:19-- https://aws-codedeploy-us-east-1.s3.amazonaws.com/latest/install
Resolving aws-codedeploy-us-east-1.s3.amazonaws.com (aws-codedeploy-us-east-1.s3.amazonaws.com)... 52.216.113.99, 52.216.238.147, 52.217.64.4, ...
Connecting to aws-codedeploy-us-east-1.s3.amazonaws.com (aws-codedeploy-us-east-1.s3.amazonaws.com)|52.216.113.99|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 17816 (17K) [t]
Saving to: 'install'

100%[=====>] 17,816      74.2KB/s   in 0.2s

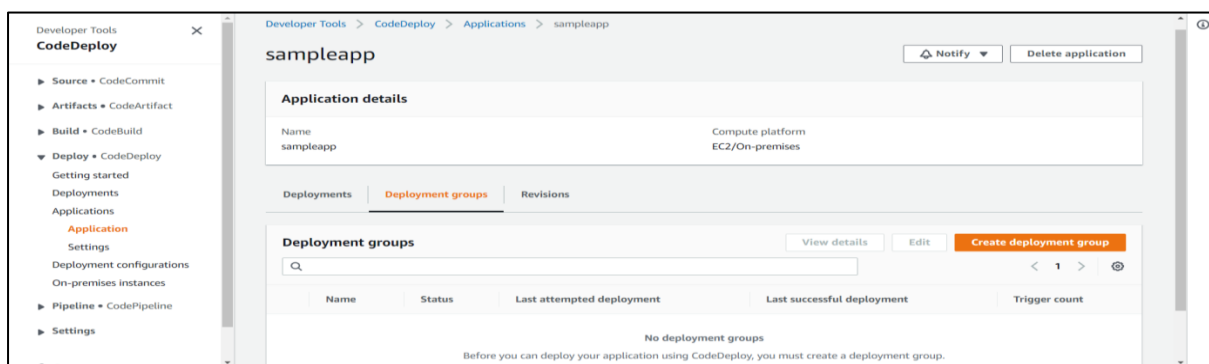
2023-01-07 15:21:20 (74.2 KB/s) - 'install' saved [17816/17816]

[root@ip-172-31-29-67 ~]# ll
total 20
-rw-r--r-- 1 root root 17816 Oct 14 17:46 install
[root@ip-172-31-29-67 ~]# ls
install
[root@ip-172-31-29-67 ~]# chmod +x install
[root@ip-172-31-29-67 ~]# ./install auto
[root@ip-172-31-29-67 ~]# #
I, [2023-01-07T15:22:01.601831 #3763] INFO -- : Starting Ruby version check.
I, [2023-01-07T15:22:01.602083 #3763] INFO -- : Starting update check.
I, [2023-01-07T15:22:01.602186 #3763] INFO -- : Attempting to automatically detect supported package manager type for system...
I, [2023-01-07T15:22:01.610824 #3763] INFO -- : Checking AWS_REGION environment variable for region information...
I, [2023-01-07T15:22:01.610897 #3763] INFO -- : Checking EC2 metadata service for region information...
I, [2023-01-07T15:22:01.626339 #3763] INFO -- : Checking AWS_DOMAIN environment variable for domain information...
I, [2023-01-07T15:22:01.626409 #3763] INFO -- : Checking EC2 metadata service for domain information...
I, [2023-01-07T15:22:01.633917 #3763] INFO -- : Downloading version file from bucket aws-codedeploy-ap-southeast-1 and key latest/LATEST_VERSION...
I, [2023-01-07T15:22:01.634330 #3763] INFO -- : Endpoint: https://aws-codedeploy-ap-southeast-1.s3.ap-southeast-1.amazonaws.com/latest/LATEST_VERSION
I, [2023-01-07T15:22:01.709595 #3763] INFO -- : Running version matches target version, skipping install
I, [2023-01-07T15:22:01.709735 #3763] INFO -- : Update check complete.
I, [2023-01-07T15:22:01.709761 #3763] INFO -- : Stopping updater.
[root@ip-172-31-29-67 ~]# service codedeploy-agent status
The AWS CodeDeploy agent is running as PID 3390
[root@ip-172-31-29-67 ~]#
```

Finally get PID

Step10:Create Code Deployment

Step10.1:Code deploy---->application---->create code deployment group-->



```
Deployment group name(any)--->service role select--->deployment type( in place)---
>select amazon ec2 instance--->tag group1--->key(name)---->value(production server)----
>load balancer disable--->create deployment group
```

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

Developer Tools > CodeDeploy > Applications > sampleapp > Create deployment group

Create deployment group

Application

Application
sampleapp
Compute type
EC2/On-premises

Deployment group name

Enter a deployment group name

seelan deploy

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.

Q am:awsiam:79507111549:role/codedeployrole

Services
s3

Developer Tools
CodeDeploy

Source • CodeCommit

Artifacts • CodeArtifact

Build • CodeBuild

Deploy • CodeDeploy

Getting started

Deployments

Applications

Application

Settings

Deployments configurations

On-premises instances

Pipeline • CodePipeline

Settings

Go to resource

Feedback

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 instances. [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	
Q Name	Q production server	Remove tag

Add tag

Add fan removal

Deployment group created..

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

Go to resource

Feedback

Success

Deployment group created

Developer Tools > CodeDeploy > Applications > sampleapp > seelandedeploy

seelandedeploy

EditDeleteCreate deployment

Deployment group details

Deployment group name	Application name	Compute platform
seelandedeploy	sampleapp	EC2/On-premises
Deployment type	Service role ARN	Deployment configuration
In-place	arn:aws:iam::795071115491:role/codedeployrole	CodeDeployDefault.AllAtOnce
Rollback enabled	Agent update scheduler	
False	Learn to schedule update in AWS Systems Manager	

Environment configuration: Amazon EC2 instances

Key	Value
Name	production server

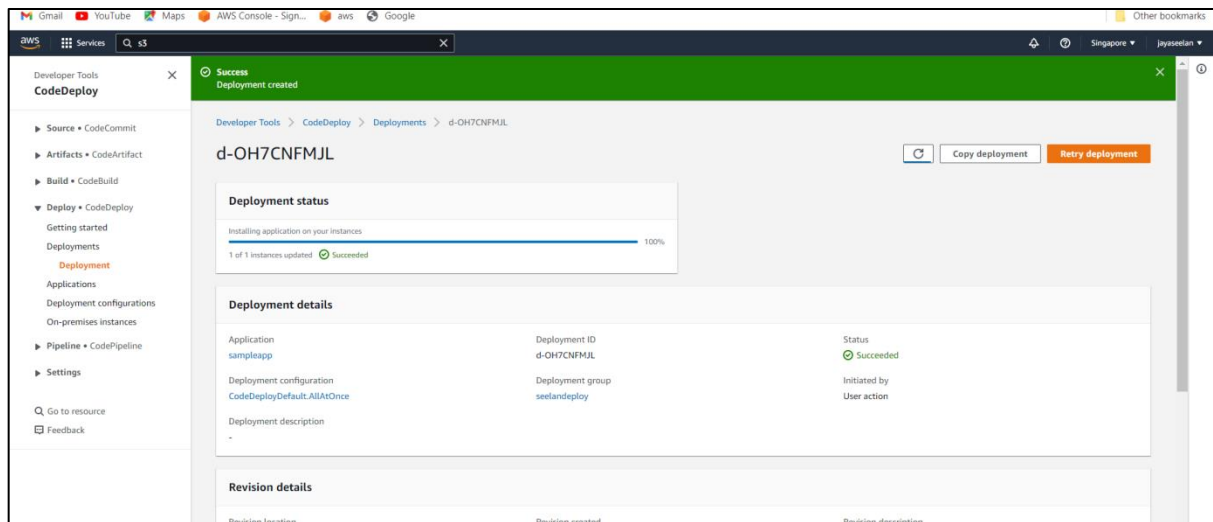
Triggers

Step10.2:create code deployment --->it already show application name,deployment group name--->revision type(my application stored in amazon s3)---->select s3 bucket location--->create deployment

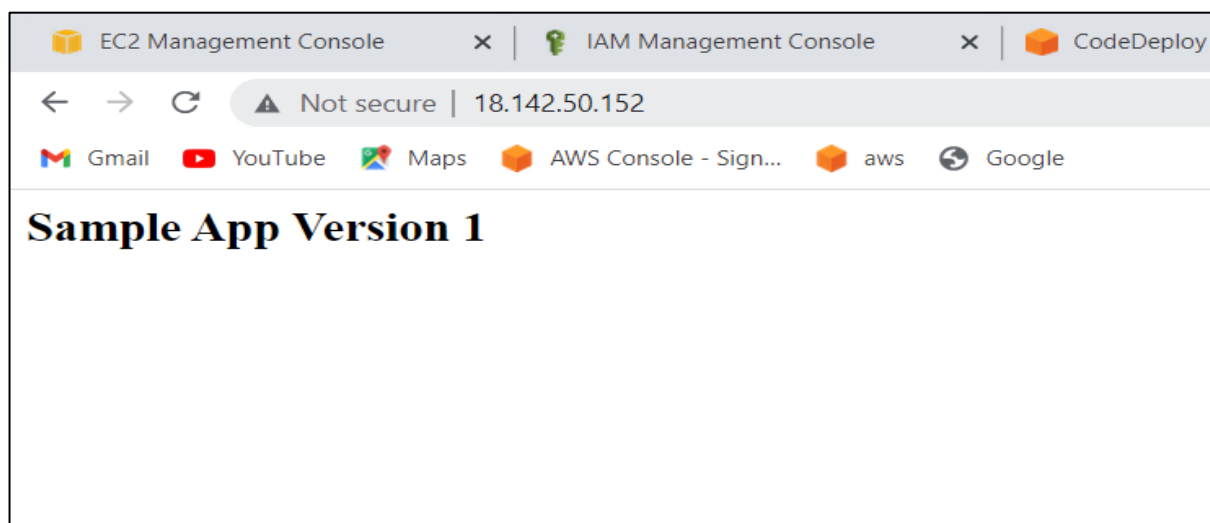
The screenshot shows the AWS CodeDeploy console interface. On the left is a navigation menu with options like Source, Artifacts, Build, Deploy, and Settings. The main area displays the 'Create deployment' page for the 'sampleapp' application group. Under 'Deployment settings', the 'Revision location' is set to an S3 bucket path, and the 'Revision file type' is set to '.zip'.

Deployment created..

It will show deployment status 100% then only it is success..

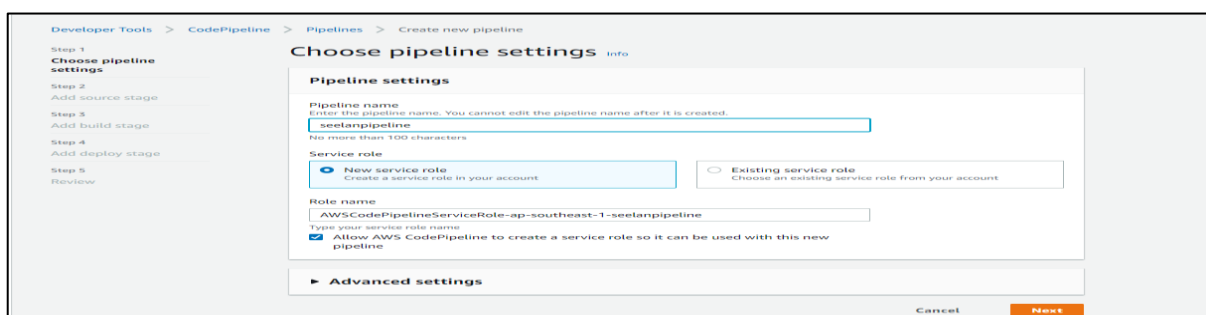


Now check html content--->put production server public ip in chrome text page



Step11:Create Code pipeline

Create pipeline--->name(any)---->next



Source provider--->select(amazon s3)--->next

The screenshot shows the 'Add source stage' step in the AWS CodePipeline console. The left sidebar lists the steps: Step 1 (Choose pipeline settings), Step 2 (Add source stage), Step 3 (Add build stage), Step 4 (Add deploy stage), and Step 5 (Review). The main area is titled 'Add source stage' with an 'Info' link. Below the title is a 'Source' section with a 'Source provider' dropdown menu. The text below the dropdown says: 'This is where you stored your input artifacts for your pipeline. Choose the provider and then provide the connection details.' At the bottom right, there are 'Cancel', 'Previous', and 'Next' buttons.

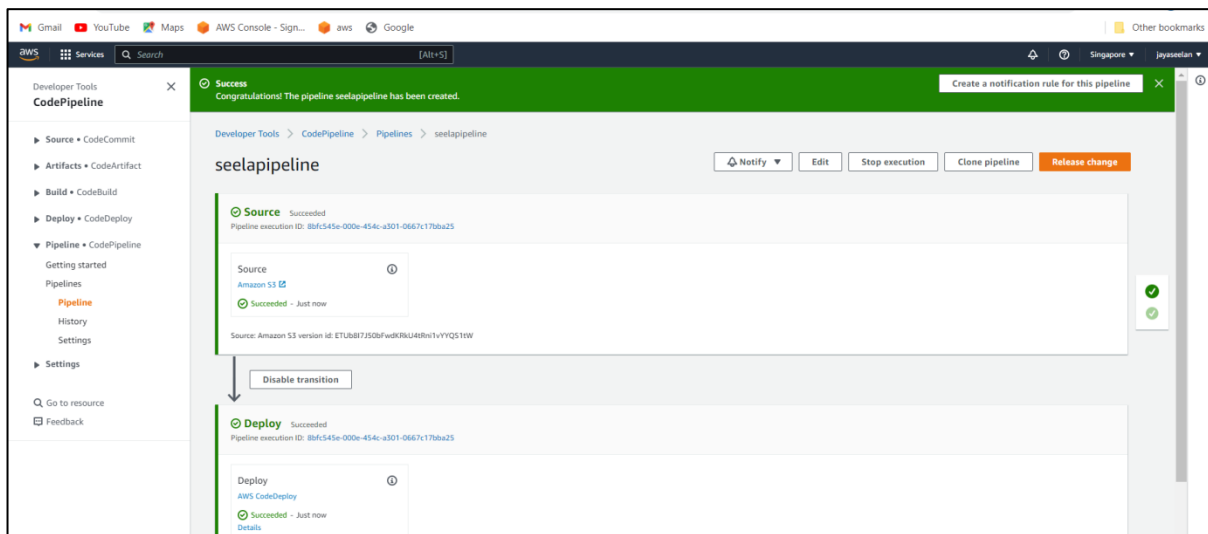
Bucket(select)--->select s3 object key--->next--->skip build stage

The screenshot shows the 'Add source stage' step in the AWS CodePipeline console. The left sidebar lists the steps: Step 1 (Choose pipeline settings), Step 2 (Add source stage), Step 3 (Add build stage), Step 4 (Add deploy stage), and Step 5 (Review). The main area is titled 'Add source stage' with an 'Info' link. Below the title is a 'Source' section. The 'Source provider' dropdown is set to 'Amazon S3'. Below it is a 'Bucket' dropdown set to 'deplybucky'. Below that is an 'S3 object key' text input field containing 'sampleapp.zip'. Below the text input is a note: 'Enter the object key. You can include a file path without the delimiter character (/) at the beginning. Include the file extension. Example: SampleApp.zip'. Below the note are two radio button options for 'Change detection options': 'Amazon CloudWatch Events (recommended)' (selected) and 'AWS CodePipeline'. At the bottom right, there are 'Cancel', 'Previous', and 'Next' buttons.

Add deploy stage--->deploy provider(aws code deploy)--->application(select)--->deployment group select--->next--->review--->create..

The screenshot shows the 'Add deploy stage' step in the AWS CodePipeline console. The left sidebar lists the steps: Step 1 (Choose pipeline settings), Step 2 (Add source stage), Step 3 (Add build stage), Step 4 (Add deploy stage), and Step 5 (Review). The main area is titled 'Add deploy stage' with an 'Info' link. Below the title is a blue information box that says: 'You cannot skip this stage. Pipelines must have at least two stages. Your second stage must be either a build or deployment stage. Choose a provider for either the build stage or deployment stage.' Below the information box is a 'Deploy' section. The 'Deploy provider' dropdown is set to 'AWS CodeDeploy'. Below it is a 'Region' dropdown set to 'Asia Pacific (Singapore)'. Below that is an 'Application name' text input field containing 'sampleapp'. Below the text input is a 'Deployment group' text input field containing 'seelandedeploy'. At the bottom right, there are 'Cancel', 'Previous', and 'Next' buttons.

code pipeline created success..



Step11:check code pipeline work

Developer server go--->edit index file

#cd sampelapp

#vi index.html ---->change html content

<html>

<h2> Sample App Version 2 </h2>

</html>

```
ec2-user@ip-172-31-18-221:~/deploy_dir/sampleapp
<html>
<h2> Sample App Version 2 </h2>
</html>
```

Delete sampleapp.zip

```
Amazon Linux 2 AMI

https://aws.amazon.com/amazon-linux-2/
[ec2-user@ip-172-31-18-221 ~]$ cd deplo_dir
-bash: cd: deplo_dir: No such file or directory
[ec2-user@ip-172-31-18-221 ~]$ cd deploy_dir
-bash: cd: deploy_dir: No such file or directory
[ec2-user@ip-172-31-18-221 ~]$ cd deploy_dir
[ec2-user@ip-172-31-18-221 deploy_dir]$ ll
total 0
drwxrwxr-x 3 ec2-user ec2-user 58 Jan 7 14:51 sampleapp
[ec2-user@ip-172-31-18-221 deploy_dir]$ cd sampleapp
[ec2-user@ip-172-31-18-221 sampleapp]$ ls
appspec.yml index.html scripts
[ec2-user@ip-172-31-18-221 sampleapp]$ vi index.html
[ec2-user@ip-172-31-18-221 sampleapp]$ cd ..
[ec2-user@ip-172-31-18-221 deploy_dir]$ ll
total 0
drwxrwxr-x 3 ec2-user ec2-user 58 Jan 7 15:50 sampleapp
[ec2-user@ip-172-31-18-221 deploy_dir]$ cd ..
[ec2-user@ip-172-31-18-221 ~]$ ll
total 4
drwxrwxr-x 3 ec2-user ec2-user 23 Jan 7 14:41 deploy_dir
-rw-rw-r-- 1 ec2-user ec2-user 1551 Jan 7 15:13 sampleapp.zip
[ec2-user@ip-172-31-18-221 ~]$
```



```
#rm -rv sampleapp.zip
```

```
[ec2-user@ip-172-31-18-221 ~]$ ll
total 4
drwxrwxr-x 3 ec2-user ec2-user  23 Jan  7 14:41 deploy_dir
-rw-rw-r-- 1 ec2-user ec2-user 1551 Jan  7 15:13 sampleapp.zip
[ec2-user@ip-172-31-18-221 ~]$ rm -rv sampleapp.zip
removed 'sampleapp.zip'
[ec2-user@ip-172-31-18-221 ~]$
```

Now delete s3 bucket object



If a folder is selected for deletion, all objects in the folder will be deleted, and any new objects added while the delete action is in progress might also be deleted. If an object is selected for deletion, any new objects with the same name that are uploaded before the delete action is completed will also be deleted.

- Deleting the specified objects can't be undone.

[Learn more](#)

Specified objects

< 1 >

Name	Version ID	Type	Last modified	Size
 sampleapp.zip	ETUb8I7J50bFwdKRkU4tRni1vYYQS1tW	zip	January 7, 2023, 20:45:40 (UTC+05:30)	1.5 KB
 sampleapp.zip	.2PdpDylrFu1N_WpbOJZIDxGijNIXnWR	zip	January 7, 2023, 20:41:00 (UTC+05:30)	885.0 B

Permanently delete objects?

To confirm deletion, type *permanently delete* in the text input field.

Cancel Delete objects

Now again put object in bucket using command

```
#Cd sampleapp
```

```
aws deploy push --application-name sampleapp --s3-location s3://deplybuckey/sampleapp.zip
```

```
/home/ec2-user/deploy_dir/appspec.yml was not found
[ec2-user@ip-172-31-18-221 deploy_dir]$ aws deploy push --application-name sampleapp --s3-location s3://deplybuckey/sampleapp.zip

/home/ec2-user/deploy_dir/appspec.yml was not found
[ec2-user@ip-172-31-18-221 deploy_dir]$ cd ..
[ec2-user@ip-172-31-18-221 ~]$ cd deploy_dir
[ec2-user@ip-172-31-18-221 deploy_dir]$ cd sampleapp
[ec2-user@ip-172-31-18-221 sampleapp]$ aws deploy push --application-name sampleapp --s3-location s3://deplybuckey/sampleapp.zip
To deploy with this revision, run:
aws deploy create-deployment --application-name sampleapp --s3-location bucket=deplybuckey,key=sampleapp.zip,bundleType=zip,eTag=9934011c3aa5c9abc2e27985816c749c,version=yE9tw3WRH8AVpbeUKxSd
ng0QesNMU5j --deployment-group-name <deployment-group-name> --deployment-config-name <deployment-config-name> --description <description>
[ec2-user@ip-172-31-18-221 sampleapp]$
```

cd..

deply_dir----> #zip -r ../sampleapp.zip .

```
ng8QPesNMJS] --deployment-group-name <deployment-group-name> --deployment-config-name
[ec2-user@ip-172-31-18-221 sampleapp]$ cd deploy_dir
-bash: cd: deploy_dir: No such file or directory
[ec2-user@ip-172-31-18-221 sampleapp]$ cd ..
[ec2-user@ip-172-31-18-221 deploy_dir]$ zip -r ../sampleapp.zip .
  adding: sampleapp/ (stored 0%)
  adding: sampleapp/appspec.yml (deflated 53%)
  adding: sampleapp/scripts/ (stored 0%)
  adding: sampleapp/scripts/httpd_install.sh (stored 0%)
  adding: sampleapp/scripts/httpd_start.sh (stored 0%)
  adding: sampleapp/scripts/httpd_stop.sh (stored 0%)
  adding: sampleapp/index.html (deflated 9%)
[ec2-user@ip-172-31-18-221 deploy_dir]$
```

cd..

aws s3 cp sampleapp.zip s3://deplybucky

```
Last login: Sat Jan  7 15:48:08 2023 from 157.51.66.55

  _ |  ( _ | _ )
  _ |  ( _ | _ ) /   Amazon Linux 2 AMI
  _ | \ _ | _ |

https://aws.amazon.com/amazon-linux-2/
[ec2-user@ip-172-31-18-221 ~]$ cd deploy_dir
[ec2-user@ip-172-31-18-221 deploy_dir]$ cd sampleapp
[ec2-user@ip-172-31-18-221 sampleapp]$ aws s3 cp sampleapp.zip s3://deplybucky

The user-provided path sampleapp.zip does not exist.
[ec2-user@ip-172-31-18-221 sampleapp]$ cd ..
[ec2-user@ip-172-31-18-221 deploy_dir]$ cd ..
[ec2-user@ip-172-31-18-221 ~]$ aws s3 cp sampleapp.zip s3://deplybucky
upload: ./sampleapp.zip to s3://deplybucky/sampleapp.zip
[ec2-user@ip-172-31-18-221 ~]$ ll
total 4
drwxrwxr-x 3 ec2-user ec2-user  23 Jan  7 14:41 deploy_dir
-rw-rw-r-- 1 ec2-user ec2-user 1551 Jan  7 16:01 sampleapp.zip
[ec2-user@ip-172-31-18-221 ~]$ ls
```

It show sampleapp.zip file

Now check bucket object will shown --->it is update version

The screenshot shows the Amazon S3 console interface. On the left is a navigation sidebar with options like Buckets, Access Points, and Storage Lens. The main area displays the 'deplybucky' bucket. Below the bucket name, there are tabs for Objects, Properties, Permissions, Metrics, Management, and Access Points. The 'Objects' tab is active, showing a list of objects. The list has columns for Name, Type, Version ID, Last modified, Size, and Storage class. Two objects are listed: 'sampleapp.zip' with Version ID '5xBRz_vU05IAKwYLAaf3Pe29ERhte14' and 'sampleapp.zip' with Version ID 'yE9tw3WRH8AVpbeUKx5dngBQPesNMJSj'. Both were last modified on January 7, 2023, and are 1.5 KB in size. The console also shows a search bar and various action buttons like 'Copy S3 URI', 'Copy URL', 'Download', 'Open', 'Delete', 'Create folder', and 'Upload'.

Name	Type	Version ID	Last modified	Size	Storage class
sampleapp.zip	zip	5xBRz_vU05IAKwYLAaf3Pe29ERhte14	January 7, 2023, 21:34:53 (UTC+05:30)	1.5 KB	Standard
sampleapp.zip	zip	yE9tw3WRH8AVpbeUKx5dngBQPesNMJSj	January 7, 2023, 21:29:42 (UTC+05:30)	885.0 B	Standard

Step12:now code pipline work

Copy production server public ip and put chrome it will shown update html content --->it will success..

