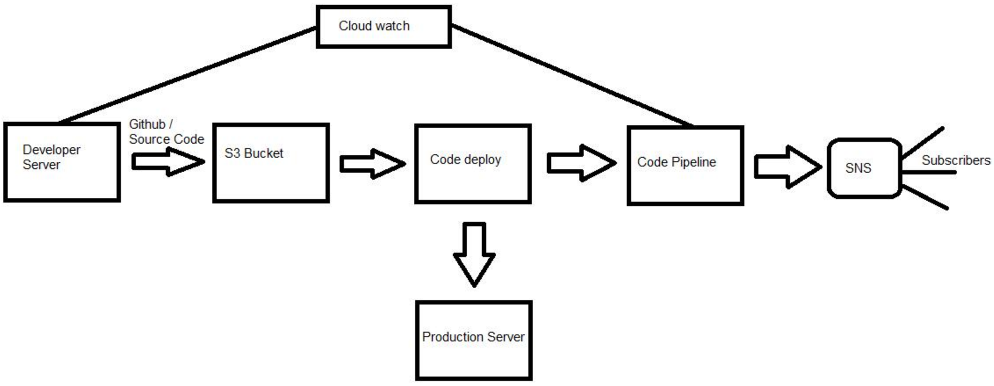
CODE DEPLOY & CODE PIPELINE

DIAGRAM



CODE DEPLOY:

AWS Code Deploy is a fully-managed deployment service provided by Amazon Web Services that automates software deployments to a variety of compute services such as Amazon EC2, AWS Fargate, AWS Lambda, and on-premises servers.

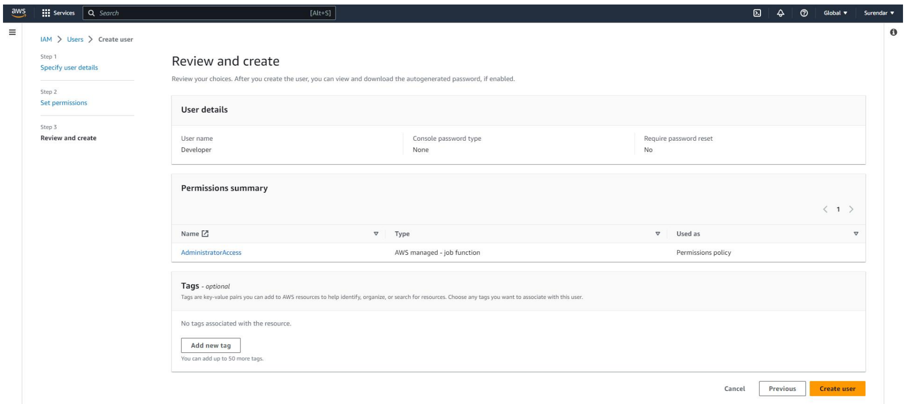
CODE PIPLELINE:

AWS Code Pipeline is a fully-managed continuous delivery service that automates the process of building,testing,and deploying applications.It helps to accelerate software development and delivery,while reducing risks and increasing the reliability of your applications.

STEPS TO CREATE CODE DEPLOY AND CODE PIPELINE:

STEP 1

In IAM.Create a IAM user as a {Developer} with Full Administrator Access.



Review and create

Review your choices. After you create the user, you can view and download the autogenerated password, if enabled.

User details

User name	Console password type	Require password reset
Developer	None	No

Permissions summary

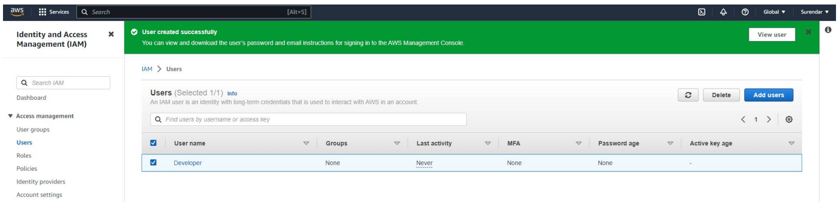
Name	Type	Used as
AdministratorAccess	AWS managed - job function	Permissions policy

Tags - *optional*

Tags are key-value pairs you can add to AWS resources to help identify, organize, or search for resources. Choose any tags you want to associate with this user.

No tags associated with the resource.

You can add up to 50 more tags.



User created successfully

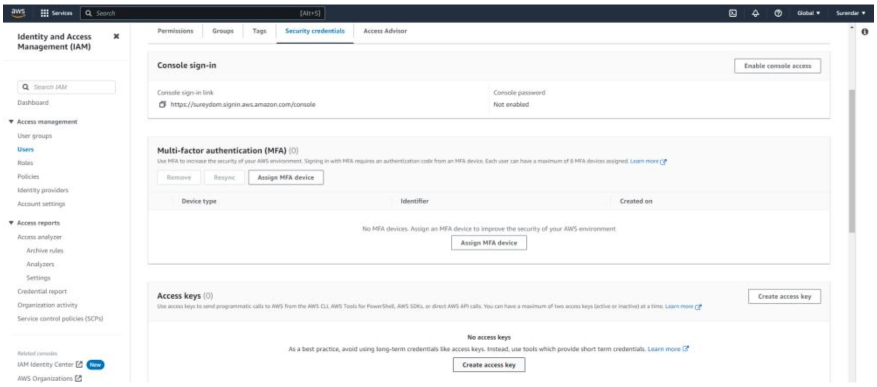
You can view and download the user's password and email instructions for signing in to the AWS Management Console.

Users (Selected 1/1) Info

An IAM user is an identity with long-term credentials that is used to interact with AWS in an account.

☑	User name	Groups	Last activity	MFA	Password age	Active key age
☑	Developer	None	Never	None	None	-

Add CLI (Command Line Interface) **AccessKeys to the IAM Developer User.**



Console sign-in

Enable console access

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

Console password: Not enabled

Multi-factor authentication (MFA) (0)

Use MFA to increase the security of your AWS environment. Signing in with MFA requires an authentication code from an MFA device. Each user can have a maximum of 8 MFA devices assigned. Learn more

Remove | Resync | Assign MFA device

Device type	Identifier	Created on

No MFA devices. Assign an MFA device to improve the security of your AWS environment

Assign MFA device

Access keys (0)

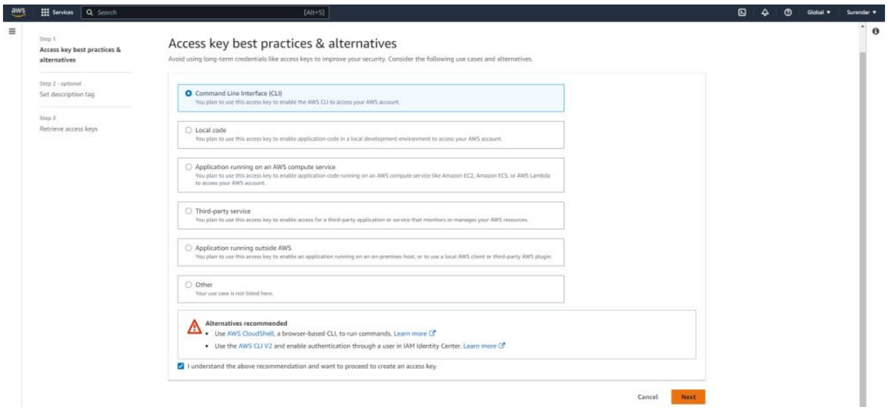
Use access keys to send programmatic calls to AWS from the AWS CLI, AWS Tools for PowerShell, AWS SDKs, or direct AWS API calls. You can have a maximum of two access keys (active or inactive) at a time. Learn more

Create access key

No access keys

As a best practice, avoid using long-term credentials like access keys. Instead, use tools which provide short term credentials. Learn more

Create access key



Step 1
Access key best practices & alternatives

Step 2 - optional
Set description tag

Step 3
Retrieve access keys

Access key best practices & alternatives

Avoid using long-term credentials like access keys to improve your security. Consider the following use cases and alternatives.

● Command Line Interface (CLI)
You plan to use this access key to enable the AWS CLI to access your AWS account.

○ Local code
You plan to use this access key to enable application code in a local development environment to access your AWS account.

○ Application running on an AWS compute service
You plan to use this access key to enable application code running on an AWS compute service like Amazon EC2, Amazon ECS, or AWS Lambda to access your AWS account.

○ Third-party service
You plan to use this access key to enable access for a third-party application or service that monitors or manages your AWS resources.

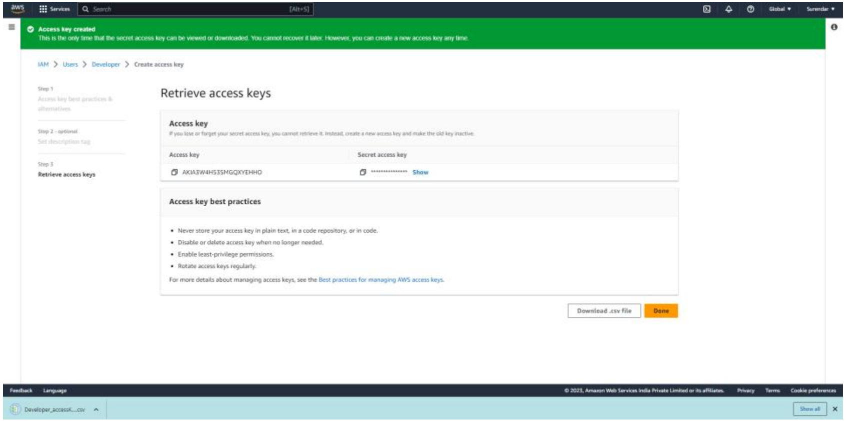
○ Application running outside AWS
You plan to use this access key to enable an application running on an on-premises host, or to use a local AWS client or third-party AWS plugin.

○ Other
Your use case is not listed here.

Alternatives recommended
- Use AWS CloudShell, a browser-based CLI, to run commands. Learn more
- Use the AWS CLI V2 and enable authentication through a user in IAM Identity Center. Learn more

☑ I understand the above recommendation and want to proceed to create an access key.

Cancel | Next



Access key created
This is the only time that the secret access key can be viewed or downloaded. You cannot recover it later. However, you can create a new access key any time.

IAM > Users > Developer > Create access key

Step 1
Access key best practices & alternatives

Step 2 - optional
Set description tag

Step 3
Retrieve access keys

Retrieve access keys

Access key

If you lose or forget your secret access key, you cannot retrieve it. Instead, create a new access key and make the old key inactive.

Access key	Secret access key
AKIA3W4H53SMGQXYEHHO	*************** Show

Access key best practices

- Never store your access key in plain text, in a code repository, or in code.
- Disable or delete access key when no longer needed.
- Enable least-privilege permissions.
- Rotate access keys regularly.

For more details about managing access keys, see the Best practices for managing AWS access keys.

Download .csv file | Done

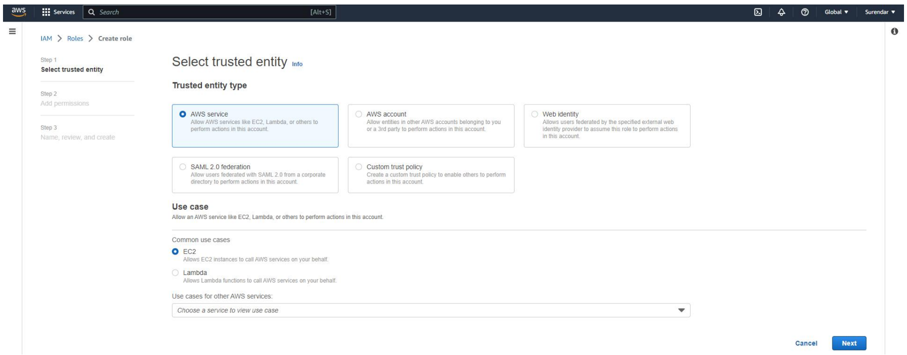
© 2023, Amazon Web Services India Private Limited or its affiliates.

Developer_accessK....csv

Create 2 new Roles as

1. **Full Admin Access,**
2. **2.Code Deploy Access.**

Choose EC2 use case - Next.



IAM > Roles > Create role

Step 1
Select trusted entity

Step 2
Add permissions

Step 3
Name, review, and create

Select trusted entity

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.

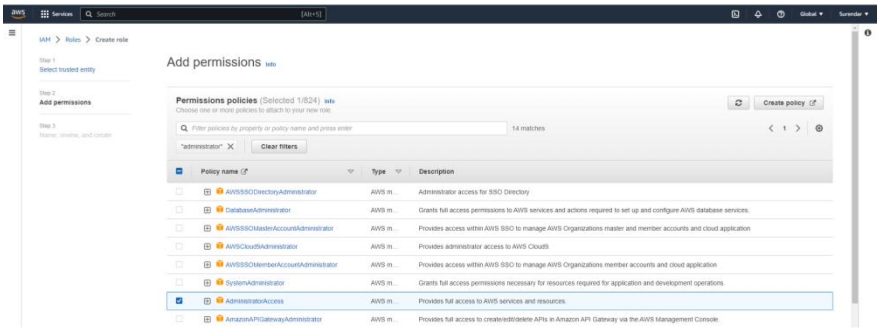
○ Lambda
Allows Lambda functions to call AWS services on your behalf.

Use cases for other AWS services:

Choose a service to view use case ▾

Cancel | Next

Add Permission (AdministratorAccess)



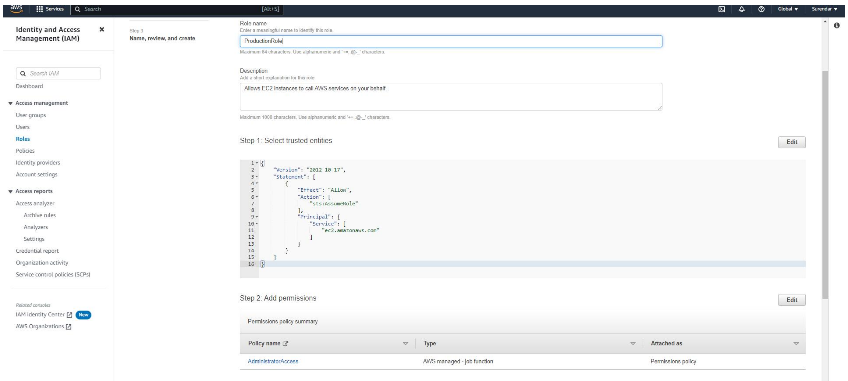
Add permissions Info

Permissions policies (Selected 1/824) 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"

	Policy name	Type	Description
□	AWSSSODirectoryAdministrator	AWS m...	Administrator access for SSO Directory
□	DatabaseAdministrator	AWS m...	Grants full access permissions to AWS services and actions required to set up and configure AWS database services.
□	AWSSSOMasterAccountAdministrator	AWS m...	Provides access within AWS SSO to manage AWS Organizations master and member accounts and cloud application
□	AWSCloud9Administrator	AWS m...	Provides administrator access to AWS Cloud9.
□	AWSSSOMemberAccountAdministrator	AWS m...	Provides access within AWS SSO to manage AWS Organizations member accounts and cloud application
□	SystemAdministrator	AWS m...	Grants full access permissions necessary for resources required for application and development operations.
☑	AdministratorAccess	AWS m...	Provides full access to AWS services and resources.
□	AmazonAPIGatewayAdministrator	AWS m...	Provides full access to create/edit/delete APIs in Amazon API Gateway via the AWS Management Console.

Set a Role name (ProdcutionRole) - Create Role.



Role name
Enter a meaningful name to identify this role.
ProductionRole
Maximum 64 characters. Use alphanumeric and '+=,.@-_' characters.

Description
Add a short explanation for this role.
Allows EC2 instances to call AWS services on your behalf.
Maximum 1000 characters. Use alphanumeric and '+=,.@-_' characters.

Step 1: Select trusted entities

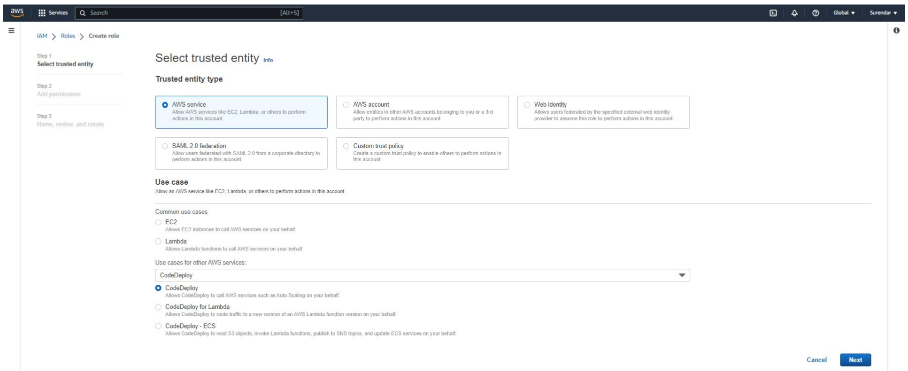
```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "sts:AssumeRole"
            ],
            "Principal": {
                "Service": [
                    "ec2.amazonaws.com"
                ]
            }
     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

Step 2: Add permissions

Permissions policy summary

Policy name	Type	Attached as
AdministratorAccess	AWS managed - job function	Permissions policy

Choose Codedeploy Use Case.



Select trusted entity Info

Trusted entity type

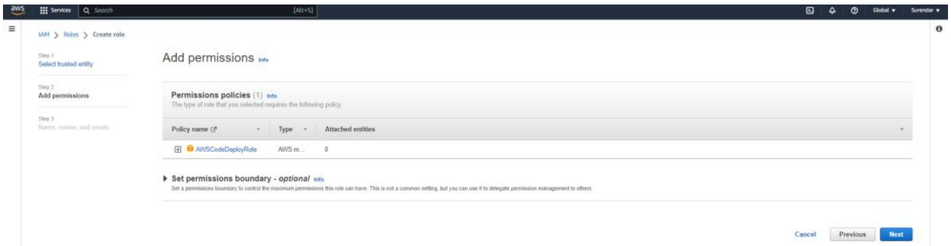
- ● AWS service — Allow AWS services like EC2, Lambda, or others to perform actions in this account.
- ○ AWS account — Allow entities in other AWS accounts belonging to you or a 3rd party to perform actions in this account.
- ○ Web identity — Allows users federated by the specified external web identity 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:
CodeDeploy
- ● CodeDeploy — Allows CodeDeploy to call AWS services such as Auto Scaling on your behalf.
- ○ CodeDeploy for Lambda — Allows CodeDeploy to route traffic to a new version of an AWS Lambda function version on your behalf.
- ○ CodeDeploy - ECS — Allows CodeDeploy to read S3 objects, invoke Lambda functions, publish to SNS topics, and update ECS services on your behalf.

Add Permission - AWSCodeDeployRole



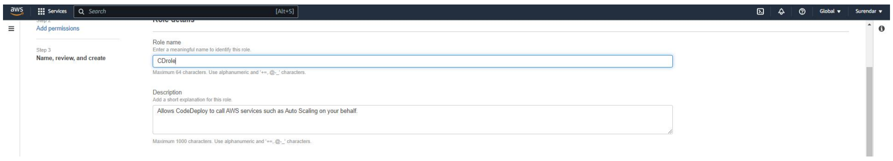
Add permissions Info

Permissions policies (1) Info
The type of role that you selected requires the following policy.

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

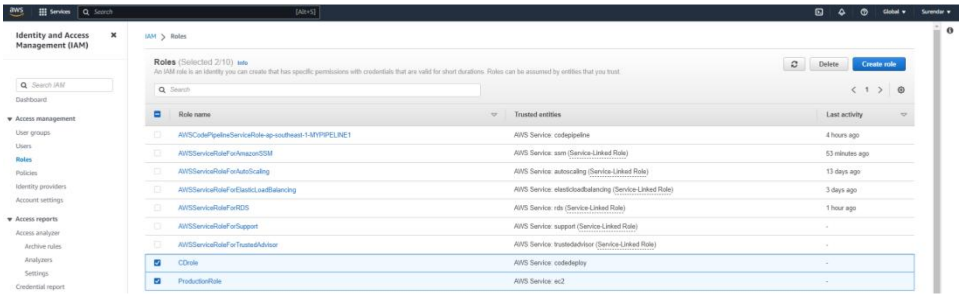
Set permissions boundary - *optional* Info
Set a permissions boundary to control the maximum permissions this role can have. This is not a common setting, but you can use it to delegate permission management to others.

Set a Role name (CDRole)



Role name
Enter a meaningful name to identify this role.
CDrole
Maximum 64 characters. Use alphanumeric and '+=,.@-_' characters.

Description
Add a short explanation for this role.
Allows CodeDeploy to call AWS services such as Auto Scaling on your behalf.
Maximum 1000 characters. Use alphanumeric and '+=,.@-_' characters.


Identity and Access Management (IAM)

IAM > Roles

Roles (Selected 2/10)

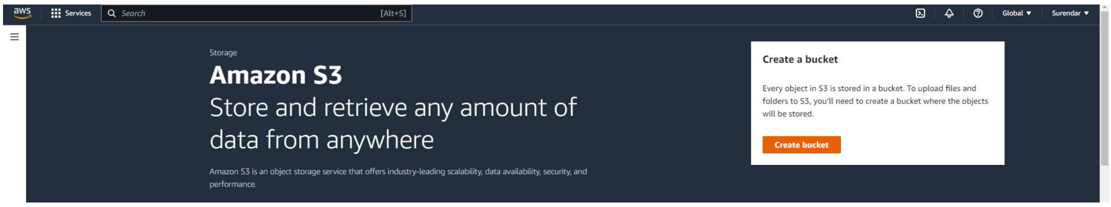
An IAM role is an identity you can create that has specific permissions with credentials that are valid for short durations. Roles can be assumed by entities that you trust.

	Role name	Trusted entities	Last activity
□	AWSCodePipelineServiceRole-ap-southeast-1-MYPIPELINE1	AWS Service: codepipeline	4 hours ago
□	AWSServiceRoleForAmazonSSM	AWS Service: ssm (Service-Linked Role)	53 minutes ago
□	AWSServiceRoleForAutoScaling	AWS Service: autoscaling (Service-Linked Role)	13 days ago
□	AWSServiceRoleForElasticLoadBalancing	AWS Service: elasticloadbalancing (Service-Linked Role)	3 days ago
□	AWSServiceRoleForRDS	AWS Service: rds (Service-Linked Role)	1 hour ago
□	AWSServiceRoleForSupport	AWS Service: support (Service-Linked Role)	-
□	AWSServiceRoleForTrustedAdvisor	AWS Service: trustedadvisor (Service-Linked Role)	-
☑	CDrole	AWS Service: codedeploy	-
☑	ProductionRole	AWS Service: ec2	-

Successfully Created 2 new Roles.

STEP 2

Create a new S3 Bucket


Storage

Amazon S3

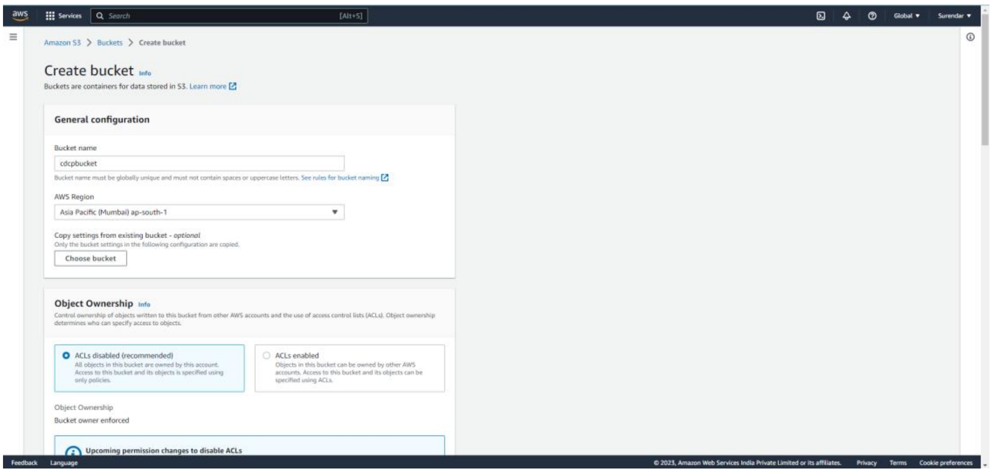
Store and retrieve any amount of data from anywhere

Amazon S3 is an object storage service that offers industry-leading scalability, data availability, security, and performance.

Create a bucket

Every object in S3 is stored in a bucket. To upload files and folders to S3, you'll need to create a bucket where the objects will be stored.

Set a Bucket name as CDCPbucket.


Amazon S3 > Buckets > Create bucket

Create bucket

Buckets are containers for data stored in S3. Learn more

General configuration

Bucket name: cdcpbucket

Bucket name must be globally unique and must not contain spaces or uppercase letters. See rules for bucket naming

AWS Region: Asia Pacific (Mumbai) ap-south-1

Copy settings from existing bucket - *optional*

Only the bucket settings in the following configuration are copied.

Object Ownership

Control ownership of objects written to this bucket from other AWS accounts and the use of access control lists (ACLs). Object ownership determines who can specify access to objects.

● ACLs disabled (recommended)
All objects in this bucket are owned by this account. Access to this bucket and its objects is specified using only policies.

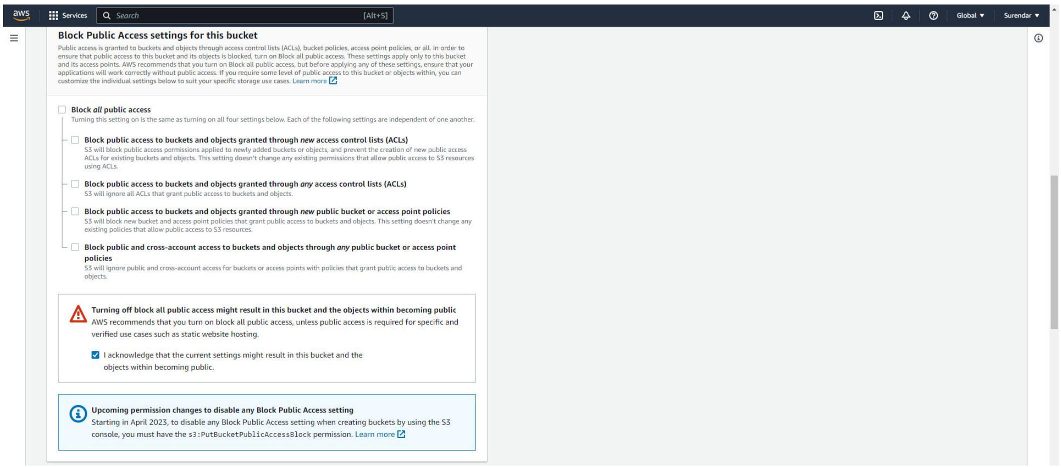
○ ACLs enabled
Objects in this bucket can be owned by other AWS accounts. Access to this bucket and its objects can be specified using ACLs.

Object Ownership
Bucket owner enforced

Upcoming permission changes to disable ACLs

© 2023, Amazon Web Services India Private Limited or its affiliates.

Unblock Public Access.


Block Public Access settings for this bucket

Public access is granted to buckets and objects through access control lists (ACLs), bucket policies, access point policies, or all. In order to ensure that public access to this bucket and its objects is blocked, turn on Block all public access. These settings apply only to this bucket and its access points. AWS recommends that you turn on Block all public access, but before applying any of these settings, ensure that your applications will work correctly without public access. If you require some level of public access to this bucket or objects within, you can customize the individual settings below to suit your specific storage use cases. Learn more

□ Block *all* public access
Turning this setting on is the same as turning on all four settings below. Each of the following settings are independent of one another.

- □ Block public access to buckets and objects granted through *new* access control lists (ACLs)
  S3 will block public access permissions applied to newly added buckets or objects, and prevent the creation of new public access ACLs for existing buckets and objects. This setting doesn't change any existing permissions that allow public access to S3 resources using ACLs.
- □ Block public access to buckets and objects granted through *any* access control lists (ACLs)
  S3 will ignore all ACLs that grant public access to buckets and objects.
- □ Block public access to buckets and objects granted through *new* public bucket or access point policies
  S3 will block new bucket and access point policies that grant public access to buckets and objects. This setting doesn't change any existing policies that allow public access to S3 resources.
- □ Block public and cross-account access to buckets and objects through *any* public bucket or access point policies
  S3 will ignore public and cross-account access for buckets or access points with policies that grant public access to buckets and objects.

Turning off block all public access might result in this bucket and the objects within becoming public

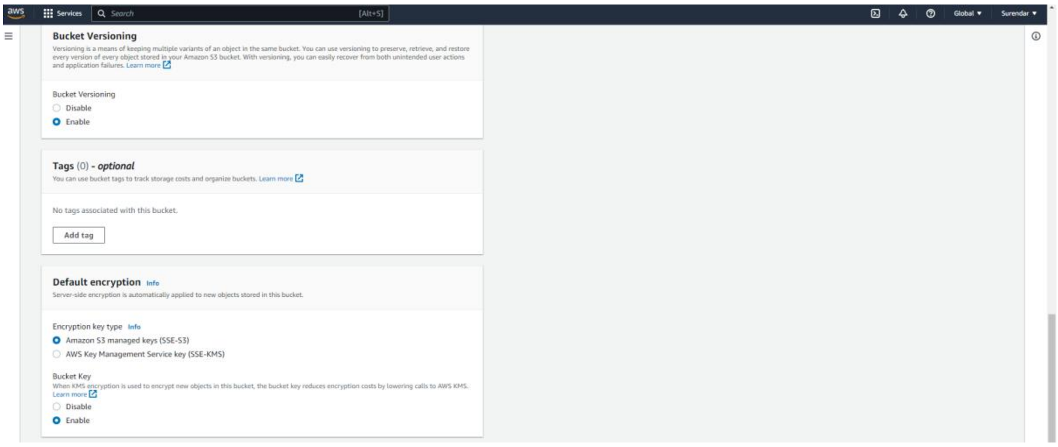
AWS recommends that you turn on block all public access, unless public access is required for specific and verified use cases such as static website hosting.

☑ I acknowledge that the current settings might result in this bucket and the objects within becoming public.

Upcoming permission changes to disable any Block Public Access setting

Starting in April 2023, to disable any Block Public Access setting when creating buckets by using the S3 console, you must have the `s3:PutBucketPublicAccessBlock` permission. Learn more

Enable Versioning - Create Bucket.



Bucket Versioning

Versioning is a means of keeping multiple variants of an object in the same bucket. You can use versioning to preserve, retrieve, and restore every version of every object stored in your Amazon S3 bucket. With versioning, you can easily recover from both unintended user actions and application failures. Learn more

Bucket Versioning

○ Disable

● Enable

Tags (0) - *optional*

You can use bucket tags to track storage costs and organize buckets. Learn more

No tags associated with this bucket.

Add tag

Default encryption Info

Server-side encryption is automatically applied to new objects stored in this bucket.

Encryption key type Info

● Amazon S3 managed keys (SSE-S3)

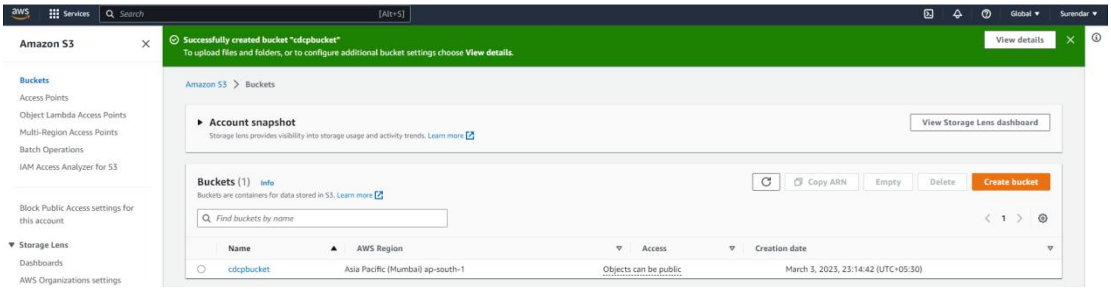
○ AWS Key Management Service key (SSE-KMS)

Bucket Key

When KMS encryption is used to encrypt new objects in this bucket, the bucket key reduces encryption costs by lowering calls to AWS KMS. Learn more

○ Disable

● Enable



Successfully created bucket "cdcpbucket"

To upload files and folders, or to configure additional bucket settings choose **View details**.

Amazon S3 > Buckets

Account snapshot

Storage lens provides visibility into storage usage and activity trends. Learn more

Buckets (1) Info

Buckets are containers for data stored in S3. Learn more

Name	AWS Region	Access	Creation date
cdcpbucket	Asia Pacific (Mumbai) ap-south-1	Objects can be public	March 3, 2023, 23:14:42 (UTC+05:30)

Amazon S3
Buckets
Access Points
Object Lambda Access Points
Multi-Region Access Points
Batch Operations
IAM Access Analyzer for S3
Block Public Access settings for this account
Storage Lens
Dashboards
AWS Organizations settings
View details
View Storage Lens dashboard
Copy ARN
Empty
Delete
Create bucket
Find buckets by name

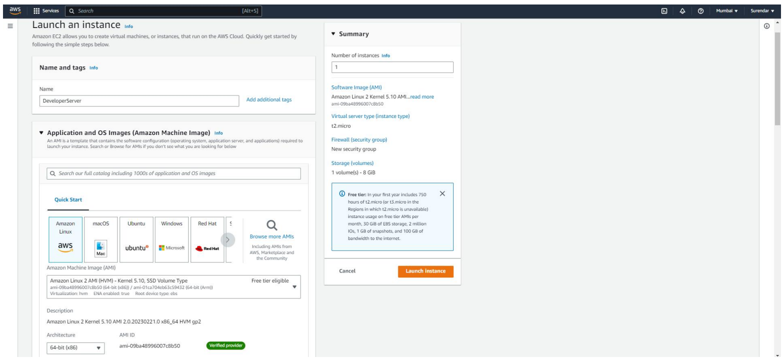
Successfully Created Amazon S3 Bucket for CDCP.

STEP 3

In EC2

Launch 2 Amazon Linux Instances for Developer server and Production Server.

Set a name as (DeveloperServer)



Launch an instance Info

Amazon EC2 allows you to create virtual machines, or instances, that run on the AWS Cloud. Quickly get started by following the simple steps below.

Name and tags Info

Name: DeveloperServer

Application and OS Images (Amazon Machine Image) Info

An AMI is a template that contains the software configuration (operating system, application server, and applications) required to launch your instance. Search or Browse for AMIs if you don't see what you are looking for below

Amazon Machine Image (AMI)

Amazon Linux 2 AMI (HVM) - Kernel 5.10, SSD Volume Type — Free tier eligible

ami-09ba48996007c8b50 (64-bit (x86)) / ami-01ca704eb63c59432 (64-bit (Arm))

Virtualization: hvm ENA enabled: true Root device type: ebs

Description

Amazon Linux 2 Kernel 5.10 AMI 2.0.20230221.0 x86_64 HVM gp2

Architecture: 64-bit (x86)

AMI ID: ami-09ba48996007c8b50 Verified provider

Summary

Number of instances Info: 1

Software Image (AMI)

Amazon Linux 2 Kernel 5.10 AMI...read more

ami-09ba48996007c8b50

Virtual server type (instance type)

t2.micro

Firewall (security group)

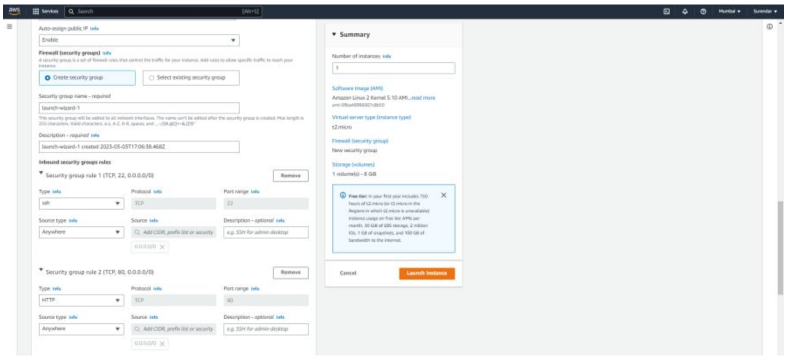
New security group

Storage (volumes)

1 volume(s) - 8 GiB

Free tier: In your first year includes 750 hours of t2.micro (or t3.micro in the Regions in which t2.micro is unavailable) instance usage on free tier AMIs per month, 30 GiB of EBS storage, 2 million IOs, 1 GB of snapshots, and 100 GB of bandwidth to the internet.

Add Inbound Security rules - SSH (Anywhere) & HTTP (Anywhere)



Auto-assign public IP Info: Enable

Firewall (security groups) Info

A security group is a set of firewall rules that control the traffic for your instance. Add rules to allow specific traffic to reach your instance.

● Create security group ○ Select existing security group

Security group name - required: launch-wizard-1

This security group will be added to all network interfaces. The name can't be edited after the security group is created. Max length is 255 characters. Valid characters: a-z, A-Z, 0-9, spaces, and ._-:/()#,@[]+=&;{}!$*

Description - required Info: launch-wizard-1 created 2023-03-03T17:06:38.468Z

Inbound security groups rules

Security group rule 1 (TCP, 22, 0.0.0.0/0)

Type: ssh Protocol: TCP Port range: 22

Source type: Anywhere Source: 0.0.0.0/0 Description - optional: e.g. SSH for admin desktop

Security group rule 2 (TCP, 80, 0.0.0.0/0)

Type: HTTP Protocol: TCP Port range: 80

Source type: Anywhere Source: 0.0.0.0/0 Description - optional: e.g. SSH for admin desktop

Summary

Number of instances Info: 1

Software Image (AMI)

Amazon Linux 2 Kernel 5.10 AMI...read more

ami-09ba48996007c8b50

Virtual server type (instance type)

t2.micro

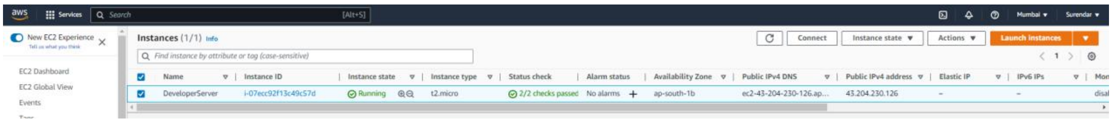
Firewall (security group)

New security group

Storage (volumes)

1 volume(s) - 8 GiB

Free tier: In your first year includes 750 hours of t2.micro (or t3.micro in the Regions in which t2.micro is unavailable) instance usage on free tier AMIs per month, 30 GiB of EBS storage, 2 million IOs, 1 GB of snapshots, and 100 GB of bandwidth to the internet.

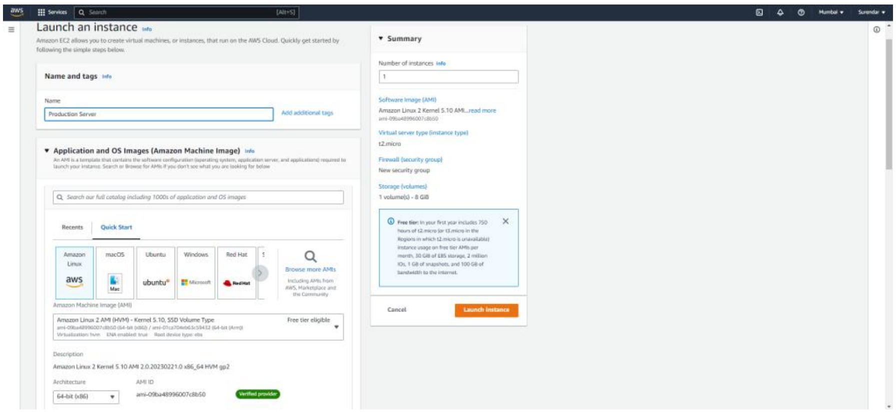


Instances (1/1)

Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public IPv4 DNS	Public IPv4 address	Elastic IP	IPv6 IPs	Mon
DeveloperServer	i-07ecc92f13c49c57d	Running	t2.micro	2/2 checks passed	No alarms	ap-south-1b	ec2-43-204-230-126.ap...	43.204.230.126	–	–	disal

Successfully Created a Amazon Linux instances for Developer Server.

Launch an another Instance as a Production server



Launch an instance

Amazon EC2 allows you to create virtual machines, or instances, that run on the AWS Cloud. Quickly get started by following the simple steps below.

Name and tags

Name: Production Server

Application and OS Images (Amazon Machine Image)

Amazon Machine Image (AMI): Amazon Linux 2 AMI (HVM) - Kernel 5.10, SSD Volume Type — Free tier eligible

Description: Amazon Linux 2 Kernel 5.10 AMI 2.0.20230221.0 x86_64 HVM gp2

Architecture: 64-bit (x86)

AMI ID: ami-09ba48996007c8b50

Summary

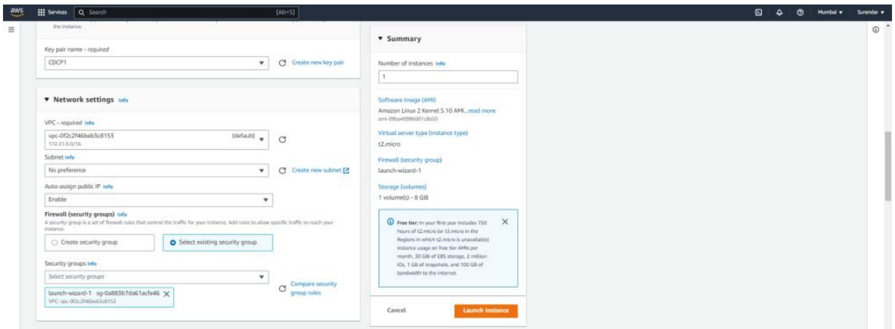
Number of instances: 1

Software Image (AMI): Amazon Linux 2 Kernel 5.10 AMI...read more

Virtual server type (instance type): t2.micro

Firewall (security group): New security group

Storage (volumes): 1 volume(s) - 8 GiB



Key pair name - required: CDCP1

Network settings

VPC - required: vpc-0f2c2f46beb3c8153 (default)

Subnet: No preference

Auto-assign public IP: Enable

Firewall (security groups): ○ Create security group ● Select existing security group

Security groups: launch-wizard-1 sg-0a883b7da61acfe46

Summary

Number of instances: 1

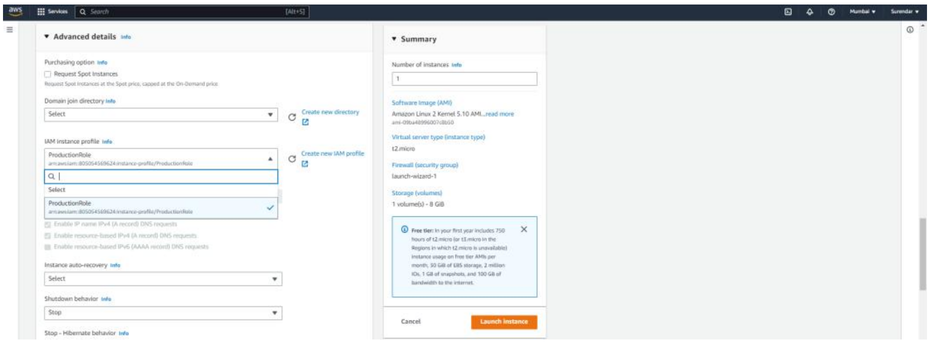
Software Image (AMI): Amazon Linux 2 Kernel 5.10 AMI...read more

Virtual server type (instance type): t2.micro

Firewall (security group): launch-wizard-1

Storage (volumes): 1 volume(s) - 8 GiB

In Advanced Details - IAM instance profile - select already created AdministratorAccess Role (ProductionRole).



Advanced details

Purchasing option: □ Request Spot Instances

Domain join directory: Select

IAM instance profile: ProductionRole
arn:aws:iam::805054569624:instance-profile/ProductionRole

Instance auto-recovery: Select

Shutdown behavior: Stop

Summary

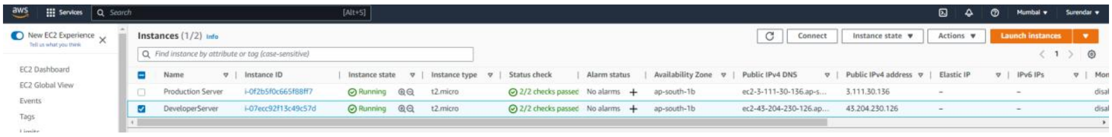
Number of instances: 1

Software Image (AMI): Amazon Linux 2 Kernel 5.10 AMI...read more

Virtual server type (instance type): t2.micro

Firewall (security group): launch-wizard-1

Storage (volumes): 1 volume(s) - 8 GiB



Instances (1/2)

Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public IPv4 DNS	Public IPv4 address	Elastic IP	IPv6 IPs	Mon
Production Server	i-0f2b5f0c665f88ff7	Running	t2.micro	2/2 checks passed	No alarms	ap-south-1b	ec2-3-111-30-136.ap-s...	3.111.30.136	–	–	disal
DeveloperServer	i-07ecc92f13c49c57d	Running	t2.micro	2/2 checks passed	No alarms	ap-south-1b	ec2-43-204-230-126.ap...	43.204.230.126	–	–	disal

Successfully Created 2 Amazon Linux Instances.

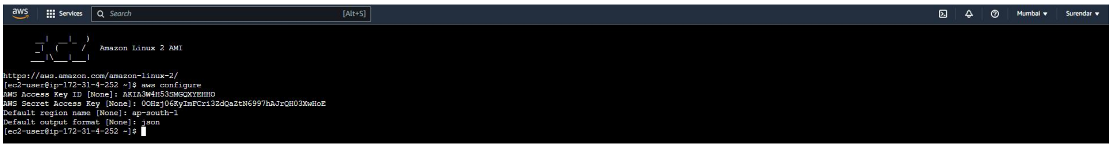
STEP 4

Connect DeveloperServer

Login IAM user in server

$aws configure

{Enter Login Accesskeys credentials}



```
       __|  __|_  )
       _|  (     /   Amazon Linux 2 AMI
      ___|\___|___|

https://aws.amazon.com/amazon-linux-2/
[ec2-user@ip-172-31-4-252 ~]$ aws configure
AWS Access Key ID [None]: AKIA3W4H53SMGQXYEHHO
AWS Secret Access Key [None]: 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
Default region name [None]: ap-south-1
Default output format [None]: json
[ec2-user@ip-172-31-4-252 ~]$ 
```

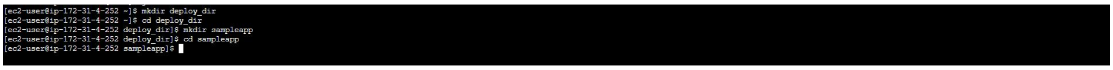
Create a multiple directories

$mkdir deploy_dir

$cd deploy_dir

$mkdir sampleapp

$cd sampleapp

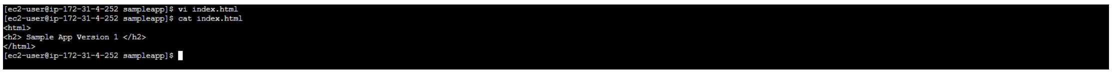


```
[ec2-user@ip-172-31-4-252 ~]$ mkdir deploy_dir
[ec2-user@ip-172-31-4-252 ~]$ cd deploy_dir
[ec2-user@ip-172-31-4-252 deploy_dir]$ mkdir sampleapp
[ec2-user@ip-172-31-4-252 deploy_dir]$ cd sampleapp
[ec2-user@ip-172-31-4-252 sampleapp]$ 
```

In Sampleapp create a index.html file using vi command

$vi index.html

Add a sample content to host in webserver.

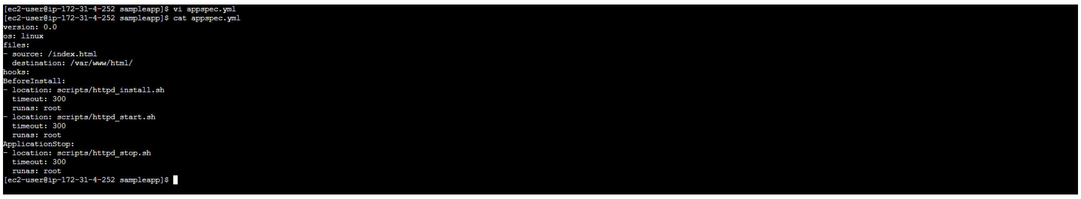


```
[ec2-user@ip-172-31-4-252 sampleapp]$ vi index.html
[ec2-user@ip-172-31-4-252 sampleapp]$ cat index.html
<html>
<h2> Sample App Version 1 </h2>
</html>
[ec2-user@ip-172-31-4-252 sampleapp]$ 
```

In Sampleapp create another yml file using vi command

$vi appspec.yml

Add automation sample yml content to install httpd service.

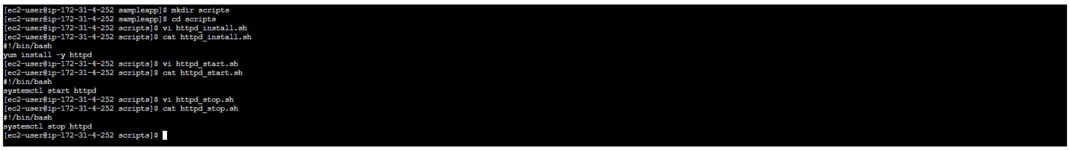


```
[ec2-user@ip-172-31-4-252 sampleapp]$ vi appspec.yml
[ec2-user@ip-172-31-4-252 sampleapp]$ cat appspec.yml
version: 0.0
os: linux
files:
- source: /index.html
  destination: /var/www/html/
hooks:
BeforeInstall:
- location: scripts/httpd_install.sh
  timeout: 300
  runas: root
- location: scripts/httpd_start.sh
  timeout: 300
  runas: root
ApplicationStop:
- location: scripts/httpd_stop.sh
  timeout: 300
  runas: root
[ec2-user@ip-172-31-4-252 sampleapp]$ 
```

Create a scripts directory using mkdir

Create a scripting for automation of install,start,stop process for httpd service.

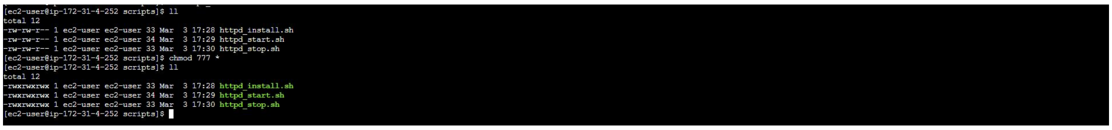
$vi httpd_install.sh , $vi httpd_start.sh , $vi httpd_stop.sh



```
[ec2-user@ip-172-31-4-252 sampleapp]$ mkdir scripts
[ec2-user@ip-172-31-4-252 sampleapp]$ cd scripts
[ec2-user@ip-172-31-4-252 scripts]$ vi httpd_install.sh
[ec2-user@ip-172-31-4-252 scripts]$ cat httpd_install.sh
#!/bin/bash
yum install -y httpd
[ec2-user@ip-172-31-4-252 scripts]$ vi httpd_start.sh
[ec2-user@ip-172-31-4-252 scripts]$ cat httpd_start.sh
#!/bin/bash
systemctl start httpd
[ec2-user@ip-172-31-4-252 scripts]$ vi httpd_stop.sh
[ec2-user@ip-172-31-4-252 scripts]$ cat httpd_stop.sh
#!/bin/bash
systemctl stop httpd
[ec2-user@ip-172-31-4-252 scripts]$ 
```

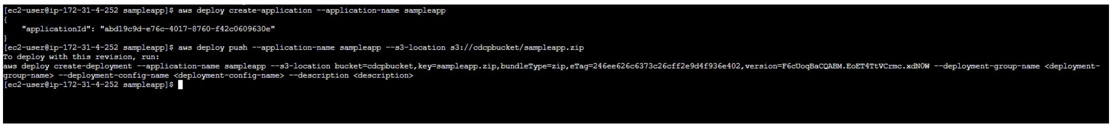
Change permission values of all files inside scripts dir

$chmod 777 *



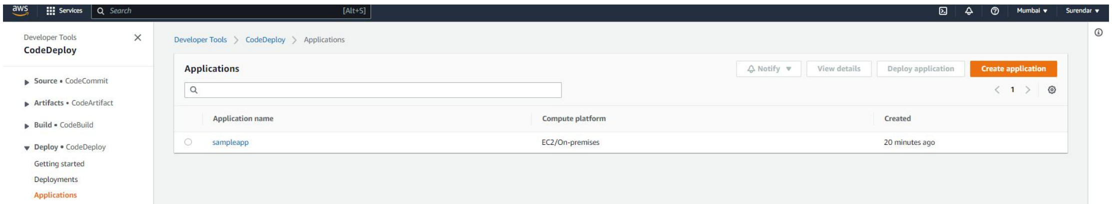
```
[ec2-user@ip-172-31-4-252 scripts]$ ll
total 12
-rw-rw-r-- 1 ec2-user ec2-user 33 Mar  3 17:28 httpd_install.sh
-rw-rw-r-- 1 ec2-user ec2-user 34 Mar  3 17:29 httpd_start.sh
-rw-rw-r-- 1 ec2-user ec2-user 33 Mar  3 17:30 httpd_stop.sh
[ec2-user@ip-172-31-4-252 scripts]$ chmod 777 *
[ec2-user@ip-172-31-4-252 scripts]$ ll
total 12
-rwxrwxrwx 1 ec2-user ec2-user 33 Mar  3 17:28 httpd_install.sh
-rwxrwxrwx 1 ec2-user ec2-user 34 Mar  3 17:29 httpd_start.sh
-rwxrwxrwx 1 ec2-user ec2-user 33 Mar  3 17:30 httpd_stop.sh
[ec2-user@ip-172-31-4-252 scripts]$
```

Create a new deploy application and push the scripts code into S3 Bucket storage using command in DeveloperServer CLI.



```
[ec2-user@ip-172-31-4-252 sampleapp]$ aws deploy create-application --application-name sampleapp
{
    "applicationId": "abd19c9d-e76c-4017-8760-f42c0609630e"
}
[ec2-user@ip-172-31-4-252 sampleapp]$ aws deploy push --application-name sampleapp --s3-location s3://cdcpbucket/sampleapp.zip
To deploy with this revision, run:
aws deploy create-deployment --application-name sampleapp --s3-location bucket=cdcpbucket,key=sampleapp.zip,bundleType=zip,eTag=246ee626c6373c26cff2e9d4f936e402,version=F6cUoqBaCQABM.EoET4TtVCrmc.xdN0W --deployment-group-name <deployment-group-name> --deployment-config-name <deployment-config-name> --description <description>
[ec2-user@ip-172-31-4-252 sampleapp]$
```

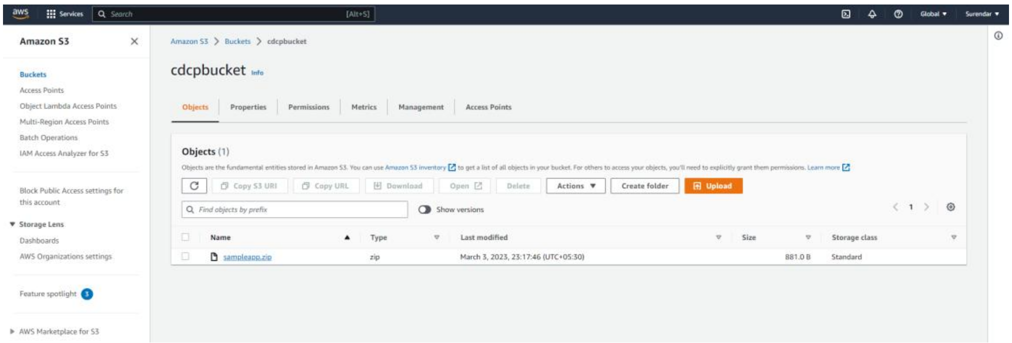
$aws deploy create-application --application-name sampleapp



Applications

Application name	Compute platform	Created
○ sampleapp	EC2/On-premises	20 minutes ago

$aws deploy push --application-name sampleapp --s3-location s3://cdcpbucket/sampleapp.zip



cdcpbucket

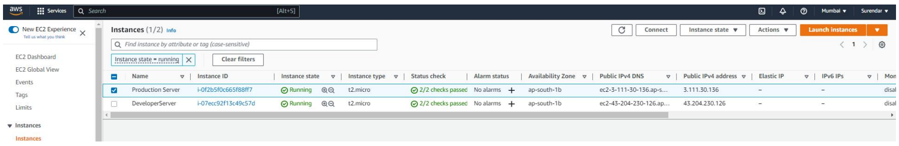
Objects (1)

Objects are the fundamental entities stored in Amazon S3. You can use Amazon S3 inventory to get a list of all objects in your bucket. For others to access your objects, you'll need to explicitly grant them permissions. Learn more

Name	Type	Last modified	Size	Storage class
sampleapp.zip	zip	March 3, 2023, 23:17:46 (UTC+05:30)	881.0 B	Standard

STEP 5

Connect ProductionServer



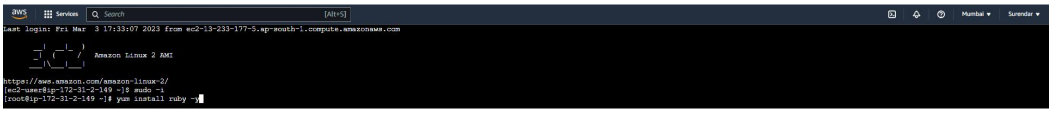
Instances (1/2)

	Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public IPv4 DNS	Public IPv4 address	Elastic IP	IPv6 IPs	Mon
☑	Production Server	i-0f2b5f0c665f88ff7	Running	t2.micro	2/2 checks passed	No alarms	ap-south-1b	ec2-3-111-30-136.ap-s...	3.111.30.136	–	–	disal
□	DeveloperServer	i-07ecc92f13c49c57d	Running	t2.micro	2/2 checks passed	No alarms	ap-south-1b	ec2-43-204-230-126.ap...	43.204.230.126	–	–	disal

Change to Root User

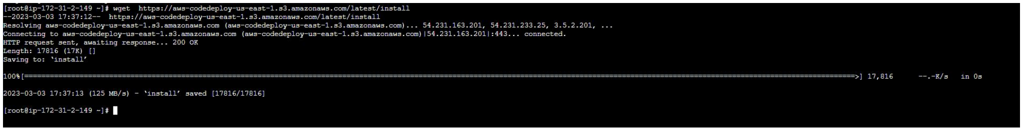
$sudo -i

#yum install ruby -y



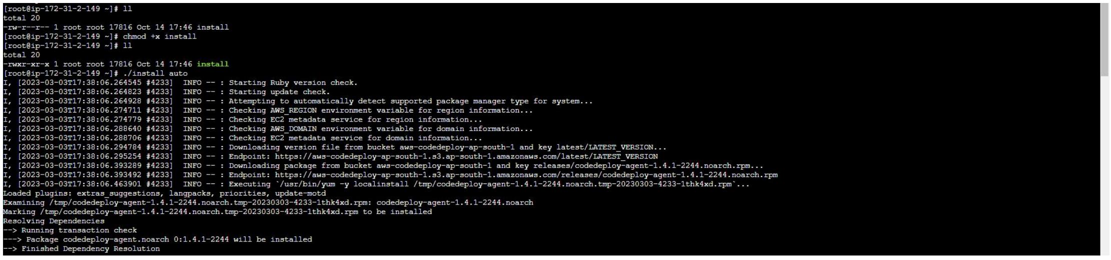
```
Last login: Fri Mar  3 17:33:07 2023 from ec2-13-233-177-5.ap-south-1.compute.amazonaws.com

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

https://aws.amazon.com/amazon-linux-2/
[ec2-user@ip-172-31-2-149 ~]$ sudo -i
[root@ip-172-31-2-149 ~]# yum install ruby -y
```

Download codedeploy agent service in productionserver using wget command in path /install



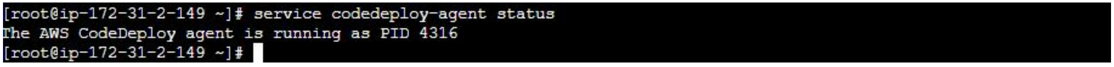
```
[root@ip-172-31-2-149 ~]# wget  https://aws-codedeploy-us-east-1.s3.amazonaws.com/latest/install
--2023-03-03 17:37:12--  https://aws-codedeploy-us-east-1.s3.amazonaws.com/latest/install
Resolving aws-codedeploy-us-east-1.s3.amazonaws.com (aws-codedeploy-us-east-1.s3.amazonaws.com)... 54.231.163.201, 54.231.233.25, 3.5.2.201, ...
Connecting to aws-codedeploy-us-east-1.s3.amazonaws.com (aws-codedeploy-us-east-1.s3.amazonaws.com)|54.231.163.201|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 17816 (17K) []
Saving to: 'install'

100%[==================================================================================================================================>] 17,816      --.-K/s   in 0s

2023-03-03 17:37:13 (125 MB/s) - 'install' saved [17816/17816]

[root@ip-172-31-2-149 ~]#
```

#chmod +x install
# ./install auto



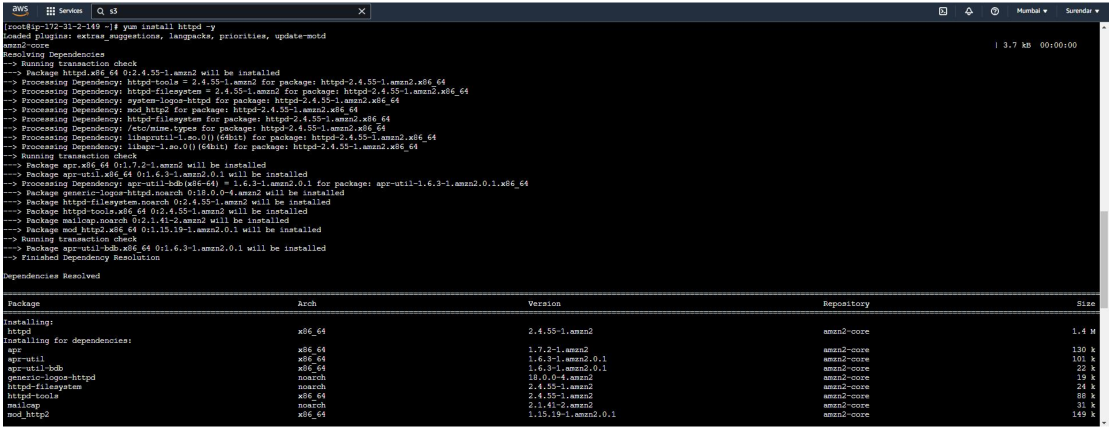
```
[root@ip-172-31-2-149 ~]# ll
total 20
-rw-r--r-- 1 root root 17816 Oct 14 17:46 install
[root@ip-172-31-2-149 ~]# chmod +x install
[root@ip-172-31-2-149 ~]# ll
total 20
-rwxr-xr-x 1 root root 17816 Oct 14 17:46 install
[root@ip-172-31-2-149 ~]# ./install auto
I, [2023-03-03T17:38:06.264545 #4233]  INFO -- : Starting Ruby version check.
I, [2023-03-03T17:38:06.264823 #4233]  INFO -- : Starting update check.
I, [2023-03-03T17:38:06.264928 #4233]  INFO -- : Attempting to automatically detect supported package manager type for system...
I, [2023-03-03T17:38:06.274711 #4233]  INFO -- : Checking AWS_REGION environment variable for region information...
I, [2023-03-03T17:38:06.274779 #4233]  INFO -- : Checking EC2 metadata service for region information...
I, [2023-03-03T17:38:06.288640 #4233]  INFO -- : Checking AWS_DOMAIN environment variable for domain information...
I, [2023-03-03T17:38:06.288706 #4233]  INFO -- : Checking EC2 metadata service for domain information...
I, [2023-03-03T17:38:06.294784 #4233]  INFO -- : Downloading version file from bucket aws-codedeploy-ap-south-1 and key latest/LATEST_VERSION...
I, [2023-03-03T17:38:06.295254 #4233]  INFO -- : Endpoint: https://aws-codedeploy-ap-south-1.s3.ap-south-1.amazonaws.com/latest/LATEST_VERSION
I, [2023-03-03T17:38:06.393289 #4233]  INFO -- : Downloading package from bucket aws-codedeploy-ap-south-1 and key releases/codedeploy-agent-1.4.1-2244.noarch.rpm...
I, [2023-03-03T17:38:06.393492 #4233]  INFO -- : Endpoint: https://aws-codedeploy-ap-south-1.s3.ap-south-1.amazonaws.com/releases/codedeploy-agent-1.4.1-2244.noarch.rpm
I, [2023-03-03T17:38:06.463901 #4233]  INFO -- : Executing `/usr/bin/yum -y localinstall /tmp/codedeploy-agent-1.4.1-2244.noarch.tmp-20230303-4233-1thk4xd.rpm`...
Loaded plugins: extras_suggestions, langpacks, priorities, update-motd
Examining /tmp/codedeploy-agent-1.4.1-2244.noarch.tmp-20230303-4233-1thk4xd.rpm: codedeploy-agent-1.4.1-2244.noarch
Marking /tmp/codedeploy-agent-1.4.1-2244.noarch.tmp-20230303-4233-1thk4xd.rpm to be installed
Resolving Dependencies
--> Running transaction check
---> Package codedeploy-agent.noarch 0:1.4.1-2244 will be installed
--> Finished Dependency Resolution
```

Check codedeploy agent status using command
#service codedeploy-agent status



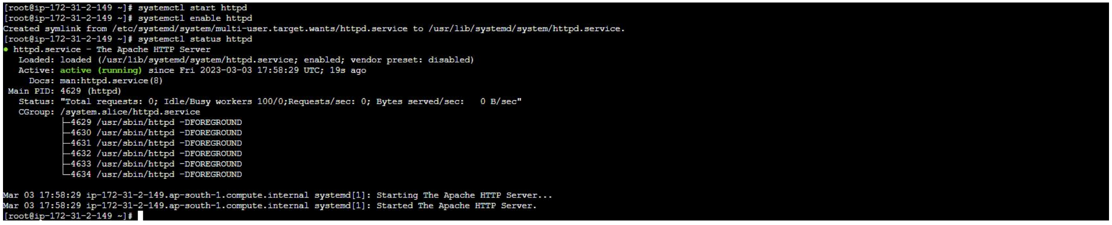
```
[root@ip-172-31-2-149 ~]# service codedeploy-agent status
The AWS CodeDeploy agent is running as PID 4316
[root@ip-172-31-2-149 ~]#
```

Install httpd service
#yum install httpd -y



```
[root@ip-172-31-2-149 ~]# yum install httpd -y
Loaded plugins: extras_suggestions, langpacks, priorities, update-motd
amzn2-core                                                                                                                                    | 3.7 kB  00:00:00
Resolving Dependencies
--> Running transaction check
---> Package httpd.x86_64 0:2.4.55-1.amzn2 will be installed
--> Processing Dependency: httpd-tools = 2.4.55-1.amzn2 for package: httpd-2.4.55-1.amzn2.x86_64
--> Processing Dependency: httpd-filesystem = 2.4.55-1.amzn2 for package: httpd-2.4.55-1.amzn2.x86_64
--> Processing Dependency: system-logos-httpd for package: httpd-2.4.55-1.amzn2.x86_64
--> Processing Dependency: mod_http2 for package: httpd-2.4.55-1.amzn2.x86_64
--> Processing Dependency: httpd-filesystem for package: httpd-2.4.55-1.amzn2.x86_64
--> Processing Dependency: /etc/mime.types for package: httpd-2.4.55-1.amzn2.x86_64
--> Processing Dependency: libaprutil-1.so.0()(64bit) for package: httpd-2.4.55-1.amzn2.x86_64
--> Processing Dependency: libapr-1.so.0()(64bit) for package: httpd-2.4.55-1.amzn2.x86_64
--> Running transaction check
---> Package apr.x86_64 0:1.7.2-1.amzn2 will be installed
---> Package apr-util.x86_64 0:1.6.3-1.amzn2.0.1 will be installed
--> Processing Dependency: apr-util-bdb(x86-64) = 1.6.3-1.amzn2.0.1 for package: apr-util-1.6.3-1.amzn2.0.1.x86_64
---> Package generic-logos-httpd.noarch 0:18.0.0-4.amzn2 will be installed
---> Package httpd-filesystem.noarch 0:2.4.55-1.amzn2 will be installed
---> Package httpd-tools.x86_64 0:2.4.55-1.amzn2 will be installed
---> Package mailcap.noarch 0:2.1.41-2.amzn2 will be installed
---> Package mod_http2.x86_64 0:1.15.19-1.amzn2.0.1 will be installed
--> Running transaction check
---> Package apr-util-bdb.x86_64 0:1.6.3-1.amzn2.0.1 will be installed
--> Finished Dependency Resolution

Dependencies Resolved

==================================================================================================================================================================
 Package                              Arch                            Version                                     Repository                                 Size
==================================================================================================================================================================
Installing:
 httpd                                x86_64                          2.4.55-1.amzn2                              amzn2-core                                1.4 M
Installing for dependencies:
 apr                                  x86_64                          1.7.2-1.amzn2                               amzn2-core                                130 k
 apr-util                             x86_64                          1.6.3-1.amzn2.0.1                           amzn2-core                                101 k
 apr-util-bdb                         x86_64                          1.6.3-1.amzn2.0.1                           amzn2-core                                 22 k
 generic-logos-httpd                  noarch                          18.0.0-4.amzn2                              amzn2-core                                 19 k
 httpd-filesystem                     noarch                          2.4.55-1.amzn2                              amzn2-core                                 24 k
 httpd-tools                          x86_64                          2.4.55-1.amzn2                              amzn2-core                                 88 k
 mailcap                              noarch                          2.1.41-2.amzn2                              amzn2-core                                 31 k
 mod_http2                            x86_64                          1.15.19-1.amzn2.0.1                         amzn2-core                                149 k
```

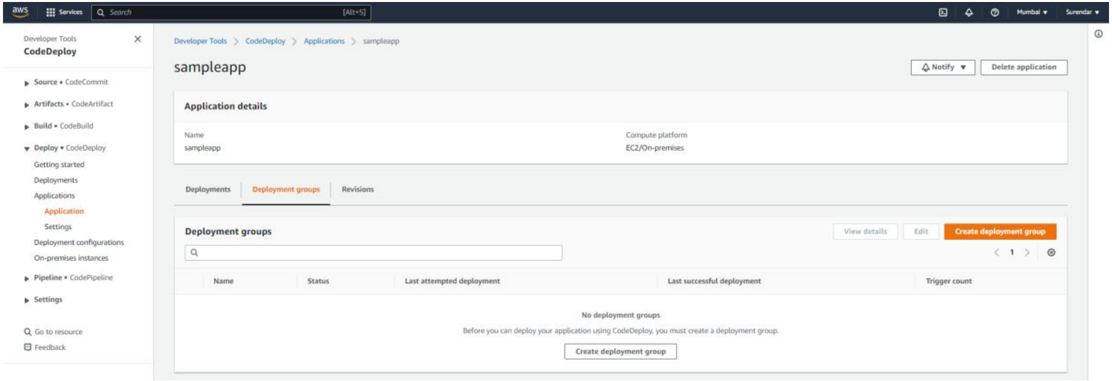
Start / Enable httpd service
#systemctl start httpd
#systemctl enablet httpd
#systemctl status httpd



```
[root@ip-172-31-2-149 ~]# systemctl start httpd
[root@ip-172-31-2-149 ~]# systemctl enable httpd
Created symlink from /etc/systemd/system/multi-user.target.wants/httpd.service to /usr/lib/systemd/system/httpd.service.
[root@ip-172-31-2-149 ~]# systemctl status httpd
● httpd.service - The Apache HTTP Server
   Loaded: loaded (/usr/lib/systemd/system/httpd.service; enabled; vendor preset: disabled)
   Active: active (running) since Fri 2023-03-03 17:58:29 UTC; 19s ago
     Docs: man:httpd.service(8)
 Main PID: 4629 (httpd)
   Status: "Total requests: 0; Idle/Busy workers 100/0;Requests/sec: 0; Bytes served/sec:   0 B/sec"
   CGroup: /system.slice/httpd.service
           ├─4629 /usr/sbin/httpd -DFOREGROUND
           ├─4630 /usr/sbin/httpd -DFOREGROUND
           ├─4631 /usr/sbin/httpd -DFOREGROUND
           ├─4632 /usr/sbin/httpd -DFOREGROUND
           ├─4633 /usr/sbin/httpd -DFOREGROUND
           └─4634 /usr/sbin/httpd -DFOREGROUND

Mar 03 17:58:29 ip-172-31-2-149.ap-south-1.compute.internal systemd[1]: Starting The Apache HTTP Server...
Mar 03 17:58:29 ip-172-31-2-149.ap-south-1.compute.internal systemd[1]: Started The Apache HTTP Server.
[root@ip-172-31-2-149 ~]#
```

STEP 6

In Codedeploy - Application - In sampleapp - Create deployment group



sampleapp

Application details

Name: sampleapp

Compute platform: EC2/On-premises

Deployment groups

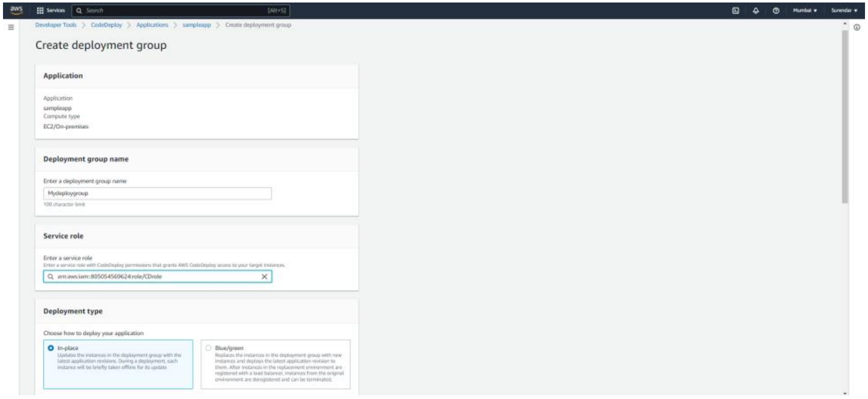
Name	Status	Last attempted deployment	Last successful deployment	Trigger count

No deployment groups

Before you can deploy your application using CodeDeploy, you must create a deployment group.

Create deployment group

Set deploymentgroup name (Mydeploygroup) - Service role Select (CDrole) - Deployment type (in-place)


aws
Services
Search
[Alt+S]
Mumbai ▾
Surendar ▾
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: Mydeploygroup

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::805054569624:role/CDrole

Deployment type

Choose how to deploy your application

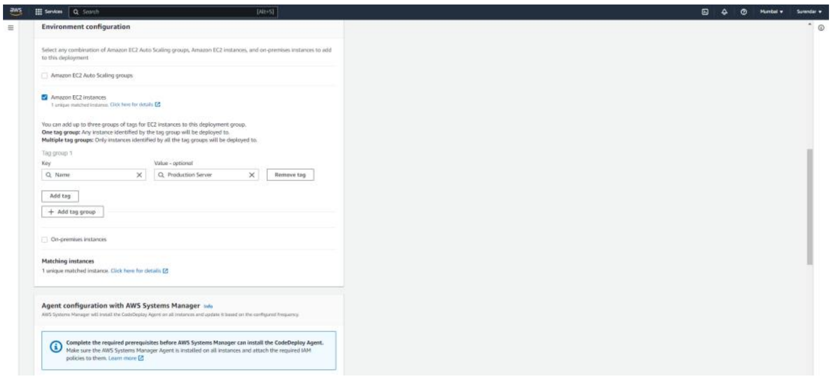
● In-place
Updates the instances in the deployment group with the latest application revisions. During a deployment, each instance will be briefly taken offline for its update

○ 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.

In environment Configuration

Select the Productionserver Tag

Key: Name   Value: ProductionServer.



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	Production Server

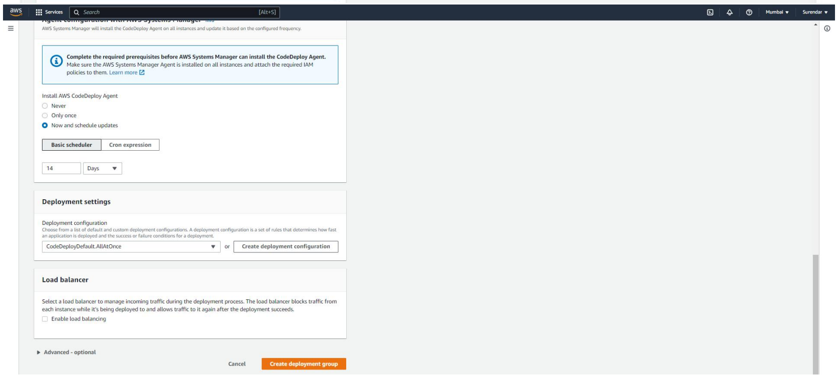
□ On-premises instances

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.

Complete the required prerequisites before AWS Systems Manager can install the CodeDeploy Agent. Make sure the AWS Systems Manager Agent is installed on all instances and attach the required IAM policies to them. Learn more



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

Complete the required prerequisites before AWS Systems Manager can install the CodeDeploy Agent. Make sure the AWS Systems Manager Agent is installed on all instances and attach the required IAM policies to them. Learn more

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

14 Days

Deployment settings

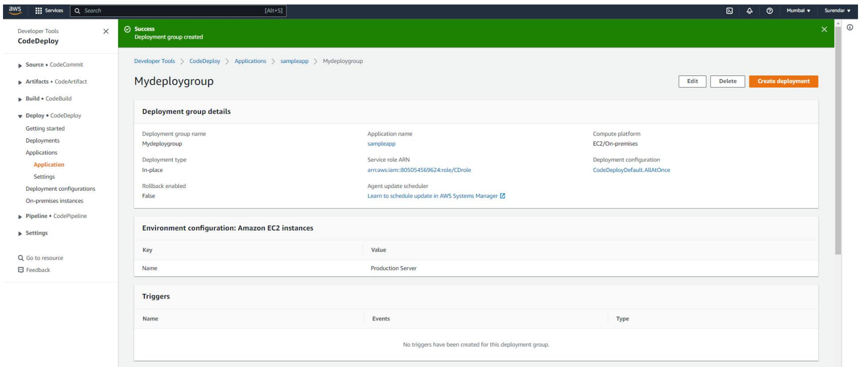
Deployment configuration
Choose from a list of default and custom deployment configurations. A deployment configuration is a set of rules that determines how fast an application is deployed and the success or failure conditions for a deployment.

CodeDeployDefault.AllAtOnce

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



Success
Deployment group created

Mydeploygroup

Deployment group details

Deployment group name
Mydeploygroup

Application name
sampleapp

Compute platform
EC2/On-premises

Deployment type
In-place

Service role ARN
arn:aws:iam::805054569624:role/CDrole

Deployment configuration
CodeDeployDefault.AllAtOnce

Rollback enabled
False

Agent update scheduler
Learn to schedule update in AWS Systems Manager

Environment configuration: Amazon EC2 instances

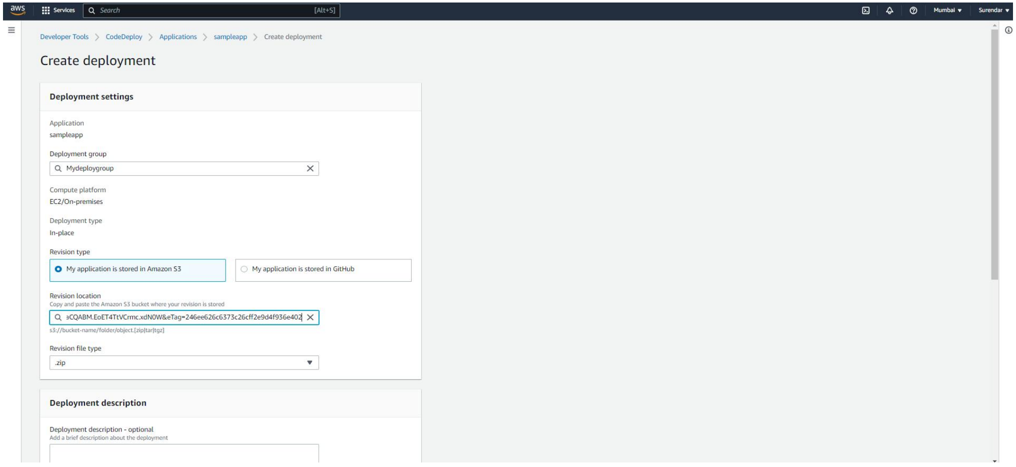
Key	Value
Name	Production Server

Triggers

Name	Events	Type
No triggers have been created for this deployment group.		

Successfully Created Deployment Group.

In Deployment group create Deployment
Choose Deployment Group name
Revision type - My application is stored in Amazon S3 - Choose the S3 Bucket Location.


aws
Services
Search
[Alt+S]
Mumbai ▾
Surendar ▾
Developer Tools > CodeDeploy > Applications > sampleapp > Create deployment

Create deployment

Deployment settings

Application
sampleapp

Deployment group
Mydeploygroup

Compute platform
EC2/On-premises

Deployment type
In-place

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

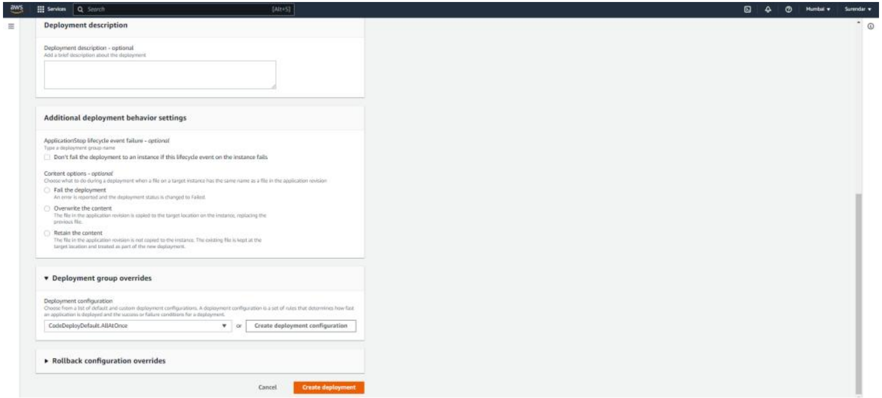
Revision location
Copy and paste the Amazon S3 bucket where your revision is stored
aCQABM.EoET4TtVCrmc.xdN0W&eTag=246ee626c6373c26cff2e9d4f936e402
s3://bucket-name/folder/object.[zip|tar|tgz]

Revision file type
.zip

Deployment description

Deployment description - optional
Add a brief description about the deployment

Create deployment



Deployment description

Deployment description - optional
Add a brief description about the deployment

Additional deployment behavior settings

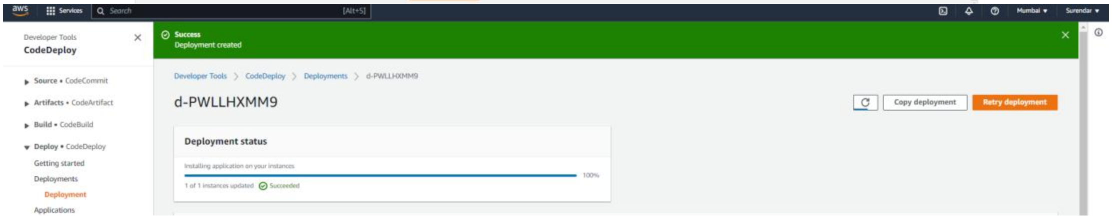
ApplicationStop lifecycle event failure - *optional*
Type a deployment group name
□ Don't fail the deployment to an instance if this lifecycle event on the instance fails

Content options - *optional*
Choose what to do during a deployment when a file on a target instance has the same name as a file in the application revision
○ Fail the deployment
An error is reported and the deployment status is changed to Failed.
○ Overwrite the content
The file in the application revision is copied to the target location on the instance, replacing the previous file.
○ Retain the content
The file in the application revision is not copied to the instance. The existing file is kept at the target location and treated as part of the new deployment.

▾ Deployment group overrides

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.AllAtOnce ▾ or Create deployment configuration

▸ Rollback configuration overrides



Success
Deployment created

d-PWLLHXMM9

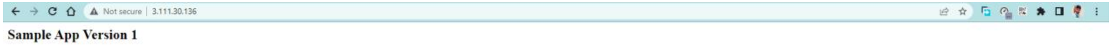
Deployment status

Installing application on your instances 100%
1 of 1 instances updated ✓ Succeeded

Successfully Created Deployment

Note: If error occurs check the yml content align and steps is done correctly.

Copy Productionserver IPv4 address and Paste in chrome tab.



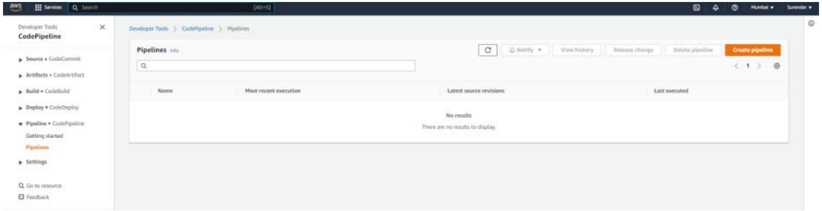
Not secure | 3.111.30.136

Sample App Version 1

Successfully Hosted Website.

STEP 7

In CodePipeline - pipelines - create Pipeline.



Developer Tools > CodePipeline > Pipelines

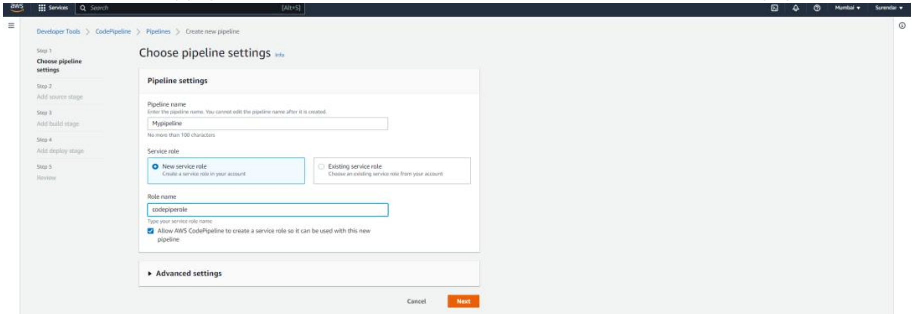
Pipelines Info

Name	Most recent execution	Latest source revisions	Last executed

No results

There are no results to display.

Set pipleline name (Mypipeline) - New Service - Set a Role Name (codepipeline) - Next.



Developer Tools > CodePipeline > Pipelines > Create new pipeline

Choose pipeline settings Info

Pipeline settings

Pipeline name

Enter the pipeline name. You cannot edit the pipeline name after it is created.

Mypipeline

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

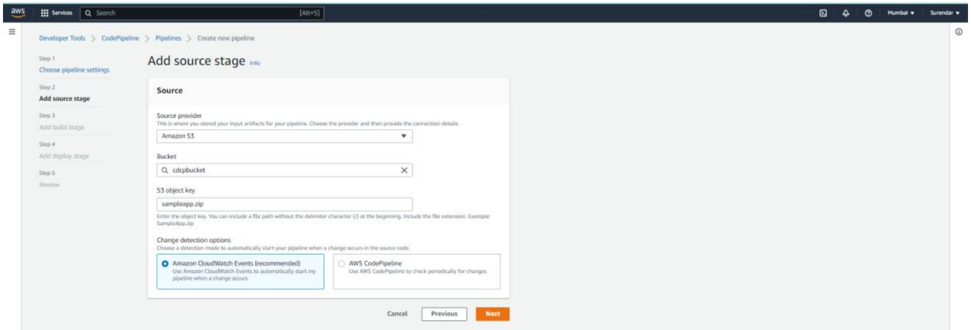
Role name

codepiperole

Type your service role name

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

Add Source stage - Amazon S3 - select Bucketname - Choose Cloudwatch events - next.



Developer Tools > CodePipeline > Pipelines > Create new pipeline

Add source stage Info

Source

Source provider

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

Amazon S3

Bucket

cdcpbucket

S3 object key

sampleapp.zip

Enter the object key. You can include a file path without the delimiter character (/) at the beginning. Include the file extension. Example: SampleApp.zip

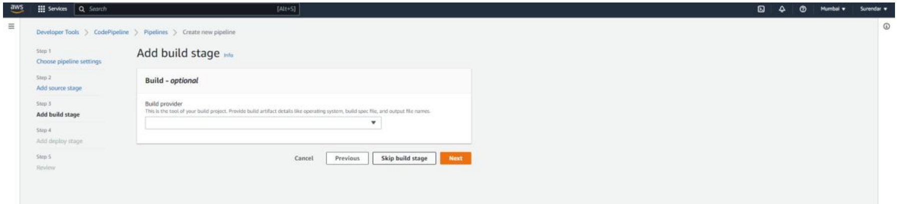
Change detection options

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

Skip Add build stage - next.



Developer Tools > CodePipeline > Pipelines > Create new pipeline

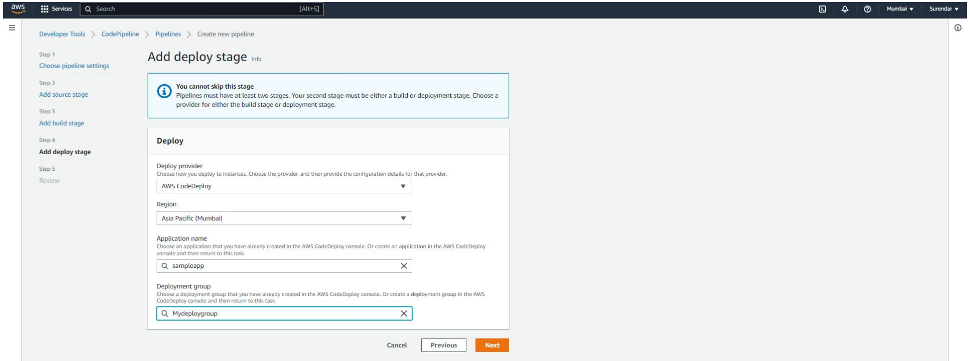
Add build stage Info

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.

Add deploy stage - Select Codedeploy - Select Mumbai region - Application name - Deployment group.



Developer Tools > CodePipeline > Pipelines > Create new pipeline

Step 1
Choose pipeline settings

Step 2
Add source stage

Step 3
Add build stage

Step 4
Add deploy stage

Step 5
Review

Add deploy stage Info

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.

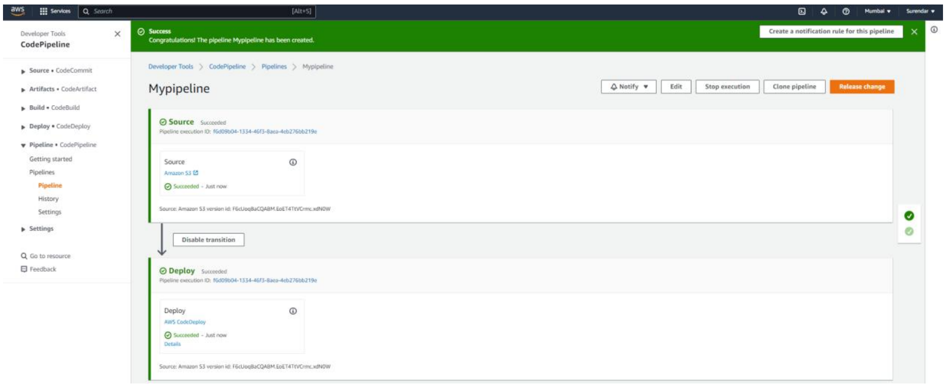
Deploy

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

Region
Asia Pacific (Mumbai)

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.
sampleapp

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.
Mydeploygroup



Success
Congratulations! The pipeline Mypipeline has been created.

Developer Tools > CodePipeline > Pipelines > Mypipeline

Mypipeline

Source Succeeded
Pipeline execution ID: f6d09b04-1334-46f3-8aea-4eb276bb219e

Source
Amazon S3
Succeeded - Just now

Source: Amazon S3 version id: F6cUoqBaCQABM.EoET4TtVCrmc.xdN0W

Deploy Succeeded
Pipeline execution ID: f6d09b04-1334-46f3-8aea-4eb276bb219e

Deploy
AWS CodeDeploy
Succeeded - Just now
Details

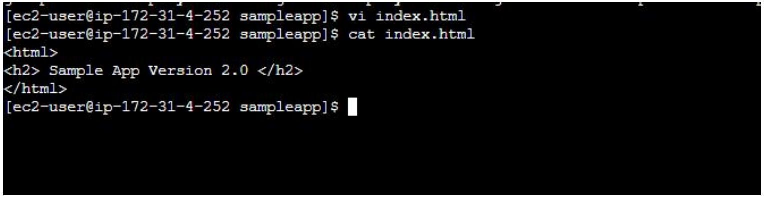
Source: Amazon S3 version id: F6cUoqBaCQABM.EoET4TtVCrmc.xdN0W

Successfully Created Pipeline

STEP 8

In Developerserver

In Sampleapp - Change Sample content in index.html



```
[ec2-user@ip-172-31-4-252 sampleapp]$ vi index.html
[ec2-user@ip-172-31-4-252 sampleapp]$ cat index.html
<html>
<h2> Sample App Version 2.0 </h2>
</html>
[ec2-user@ip-172-31-4-252 sampleapp]$
```

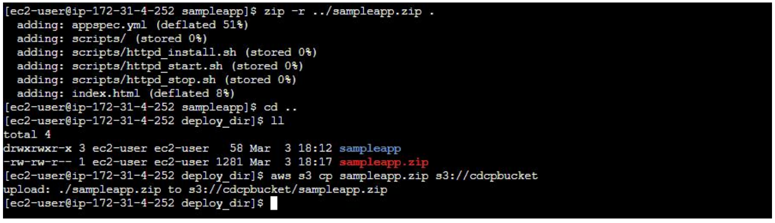
Zip the sampleapp file and save in deploy_dir using command

$zip -r ../sampleapp.zip .

$cd ..

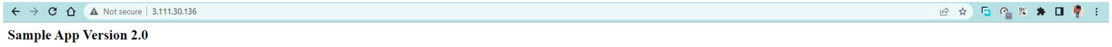
Copy the sampleapp.zip in S3 bucket

$aws s3 cp sampleapp.zip s3://cdcpbucket



```
[ec2-user@ip-172-31-4-252 sampleapp]$ zip -r ../sampleapp.zip .
  adding: appspec.yml (deflated 51%)
  adding: scripts/ (stored 0%)
  adding: scripts/httpd_install.sh (stored 0%)
  adding: scripts/httpd_start.sh (stored 0%)
  adding: scripts/httpd_stop.sh (stored 0%)
  adding: index.html (deflated 8%)
[ec2-user@ip-172-31-4-252 sampleapp]$ cd ..
[ec2-user@ip-172-31-4-252 deploy_dir]$ ll
total 4
drwxrwxr-x 3 ec2-user ec2-user   58 Mar  3 18:12 sampleapp
-rw-rw-r-- 1 ec2-user ec2-user 1281 Mar  3 18:17 sampleapp.zip
[ec2-user@ip-172-31-4-252 deploy_dir]$ aws s3 cp sampleapp.zip s3://cdcpbucket
upload: ./sampleapp.zip to s3://cdcpbucket/sampleapp.zip
[ec2-user@ip-172-31-4-252 deploy_dir]$
```

Copy productionserver IPv4 address and paste in chrome tab



Not secure | 3.111.30.136

Sample App Version 2.0

Sample Content Changes updated Successfully
Both Code deploy and Code pipeline Working Properly.