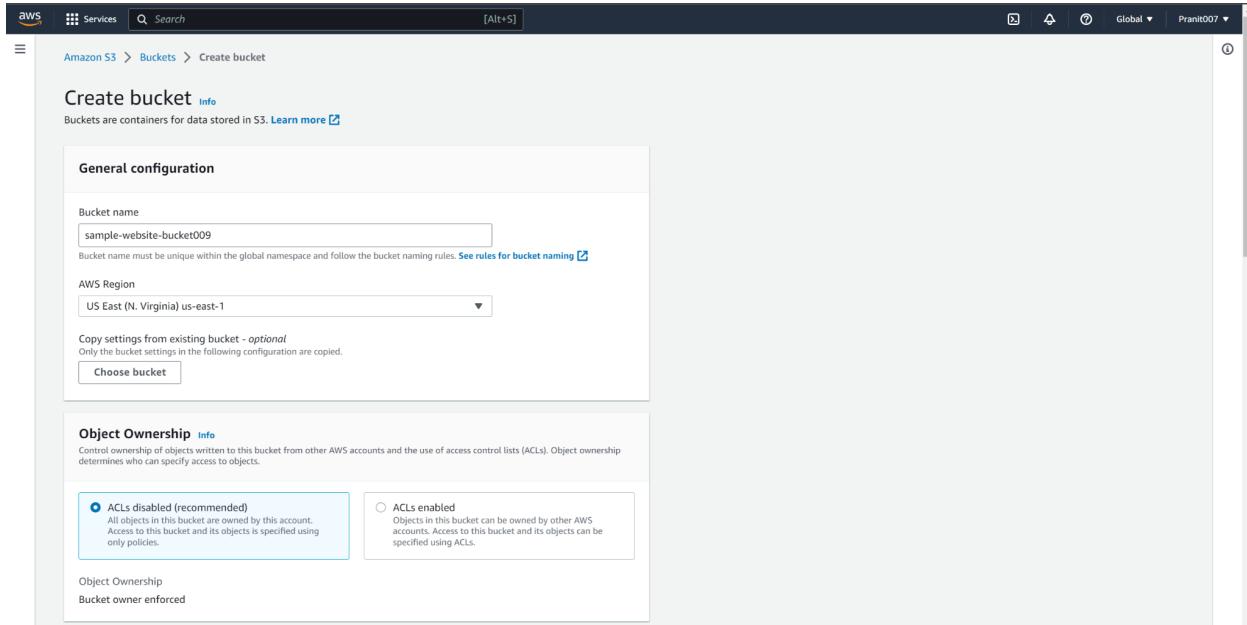


(1) Use codedeploy in order to deploy code to EC2 and code is in S3 bucket.

1. Create S3 bucket.



2. Enable Public Access(means unticked all the checkboxes)

The screenshot shows the 'Block Public Access settings for this bucket' section. It includes a note about public access being granted through various mechanisms like ACLs, policies, and access points. There are four checkboxes for different access types:

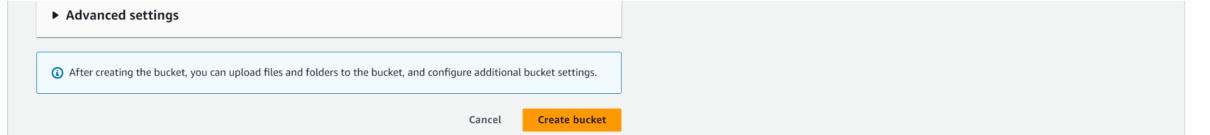
- Block all public access**: Turning this setting on is the same as turning on all four settings below. Each of the following settings are independent of one another.
- Block public access to buckets and objects granted through new access control lists (ACLs)**: S3 will block public access permissions 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.

A warning message states: "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." A checkbox below it says: "I acknowledge that the current settings might result in this bucket and the objects within becoming public." The 'Enable' checkbox is selected.

3. Enable bucket Versioning

The screenshot shows the 'Bucket Versioning' configuration. The 'Enable' radio button is selected. Below it is a 'Tags (0) - optional' section with a note about tracking storage costs and organizing buckets. The 'Add tag' button is visible. The 'Default encryption' section notes that server-side encryption is automatically applied to new objects. The 'Encryption type' dropdown shows 'Server-side encryption with Amazon S3 managed keys (SSE-S3)' selected. The 'Bucket Key' section notes that using an S3 Bucket key for SSE-KMS reduces encryption costs by lowering calls to AWS KMS. The 'Enable' radio button is selected here as well.

4. Now create the bucket



5.Go to IAM dashboard

6.Create a IAM role

The screenshot shows the AWS IAM Roles page. On the left, there's a navigation sidebar with options like 'Identity and Access Management (IAM)', 'Access management', 'Access reports', and 'Related consoles'. The main area displays a table of roles:

Role name	Trusted entities	Last activity
AWSCodePipelineServiceRole-us-east-1-CodeDeployPipeline	AWS Service: codepipeline	Yesterday
AWSCodePipelineServiceRole-us-east-1-Recommender-pipeline	AWS Service: codepipeline	Yesterday
AWSServiceRoleForAmazonSSM	AWS Service: ssm (Service-Linked Role)	1 hour ago
AWSServiceRoleForSupport	AWS Service: support (Service-Linked Role)	-
AWSServiceRoleForTrustedAdvisor	AWS Service: trustedadvisor (Service-Linked Role)	-
CodeDeployRole	AWS Service: codedeploy	6 hours ago
cwe-role-us-east-1-CodeDeployPipeline	AWS Service: events	-
cwe-role-us-east-1-Recommender-pipeline	AWS Service: events	-
EC2CodeDeploy	AWS Service: ec2	18 hours ago
s3-role	AWS Service: ec2	21 hours ago
s3_read_only	AWS Service: ec2	6 hours ago

Below the table, there's a section titled 'Roles Anywhere' with a 'Manage' button.

7.Select EC2

The screenshot shows the 'Create role' wizard, Step 1: Select trusted entity. It has three steps: Step 1 (current), Step 2 (Add permissions), and Step 3 (Name, review, and create). The 'Trusted entity type' section contains five options:

- AWS service: Allows AWS services like EC2, Lambda, or others to perform actions in this account.
- AWS account: Allows 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: Allows 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.

The 'Use case' section shows 'EC2' selected, with a note: 'Allows EC2 instances to call AWS services on your behalf.' Other options include 'Lambda'.

At the bottom right are 'Cancel' and 'Next' buttons.

8.Search S3 and select AmazonS3ReadOnlyAccess policy.

The screenshot shows the AWS IAM 'Create role' wizard at Step 2: Add permissions. A search bar at the top right contains the text "s3*". Below it is a table titled "Permissions policies (863) Info" with columns for Policy name, Type, and Description. The table lists several S3-related policies, with "AmazonS3FullAccess" and "AmazonS3ReadOnlyAccess" being the most prominent. A note below the table says "Set permissions boundary - optional" and "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."

This screenshot is identical to the one above, showing the AWS IAM 'Create role' wizard at Step 2: Add permissions. The search bar still contains "s3*". The "AmazonS3ReadOnlyAccess" policy is now highlighted with a blue border and a checked checkbox, indicating it has been selected for the role.

9.Enter the role name

Name, review, and create

Role details

Role name
s3_read_only_access_for_ec2

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

```

1 = {
2   "Version": "2012-10-17",
3   "Statement": [
4     {
5       "Effect": "Allow",
6       "Action": [
7         "sts:AssumeRole"
8       ],
9       "Principal": {
10         "Service": [
11           "ec2.amazonaws.com"
12         ]
13     }
14   }
15 ]
16 }

```

Step 1: Select trusted entities

Step 2: Add permissions

10.Click create role

Identity and Access Management (IAM)

Roles (12) Info

Role name	Trusted entities	Last activity
AWSCodePipelineServiceRole-us-east-1-CodeDeployPipeline	AWS Service: codepipeline	Yesterday
AWSCodePipelineServiceRole-us-east-1-Recommender-pipeline	AWS Service: codepipeline	Yesterday
AWSServiceRoleForAmazonSSM	AWS Service: ssm (Service-Linked Role)	1 hour ago
AWSServiceRoleForSupport	AWS Service: support (Service-Linked Role)	-
AWSServiceRoleForTrustedAdvisor	AWS Service: trustedadvisor (Service-Linked Role)	-
CodeDeployRole	AWS Service: codedeploy	6 hours ago
cwe-role-us-east-1-CodeDeployPipeline	AWS Service: events	-
cwe-role-us-east-1-Recommender-pipeline	AWS Service: events	-
EC2CodeDeploy	AWS Service: ec2	18 hours ago
s3-role	AWS Service: ec2	21 hours ago
s3_read_only	AWS Service: ec2	6 hours ago
s3_read_only_access_for_ec2	AWS Service: ec2	-

Roles Anywhere

Authenticate your non AWS workloads and securely provide access to AWS services.

11.Now creating another IAM role which will get all the codedeploy service access.Search codedeploy.

Screenshot of the AWS IAM 'Create role' wizard Step 1: Select trusted entity.

The 'Trusted entity type' section shows the 'AWS service' option selected, with a note: "Allow AWS services like EC2, Lambda, or others to perform actions in this account." Other options include 'AWS account', 'Web identity', 'SAML 2.0 federation', and 'Custom trust policy'.
The 'Use case' section shows 'EC2' and 'Lambda' selected. A search bar shows 'CodeDeploy' and a dropdown menu lists 'CodeDeploy'.
The bottom right has 'Cancel' and 'Next' buttons.

Screenshot of the AWS IAM 'Create role' wizard Step 1: Select trusted entity.

The 'Trusted entity type' section shows the 'AWS service' option selected, with a note: "Allow AWS services like EC2, Lambda, or others to perform actions in this account." Other options include 'AWS account', 'Web identity', 'SAML 2.0 federation', and 'Custom trust policy'.
The 'Use case' section shows 'EC2' and 'Lambda' selected. A search bar shows 'CodeDeploy' and a dropdown menu lists 'CodeDeploy'.
The bottom right has 'Cancel' and 'Next' buttons.

The screenshot shows the 'Add permissions' step of the IAM role creation wizard. On the left, a sidebar lists steps: Step 1 (Select trusted entity), Step 2 (Add permissions, which is selected), and Step 3 (Name, review, and create). The main area displays a table titled 'Permissions policies (1)'. The table has columns for 'Policy name', 'Type', and 'Attached entities'. One row is shown: 'AWSCodeDeployRole' (AWS managed), with a count of 1 under 'Attached entities'. Below the table is a section titled 'Set permissions boundary - optional' with a note about controlling maximum permissions. At the bottom right are 'Cancel', 'Previous', and 'Next' buttons.

12. Write the IAM role name and then create role.

The screenshot shows the 'Name, review, and create' step of the IAM role creation wizard. The sidebar shows steps 1, 2, and 3. The main area is titled 'Role details'. It includes fields for 'Role name' (set to 'CodeDeployRole01'), 'Description' (containing a note about CodeDeploy calling AWS services like Auto Scaling), and a large text area showing the JSON policy document. The policy document is as follows:

```
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+}
```

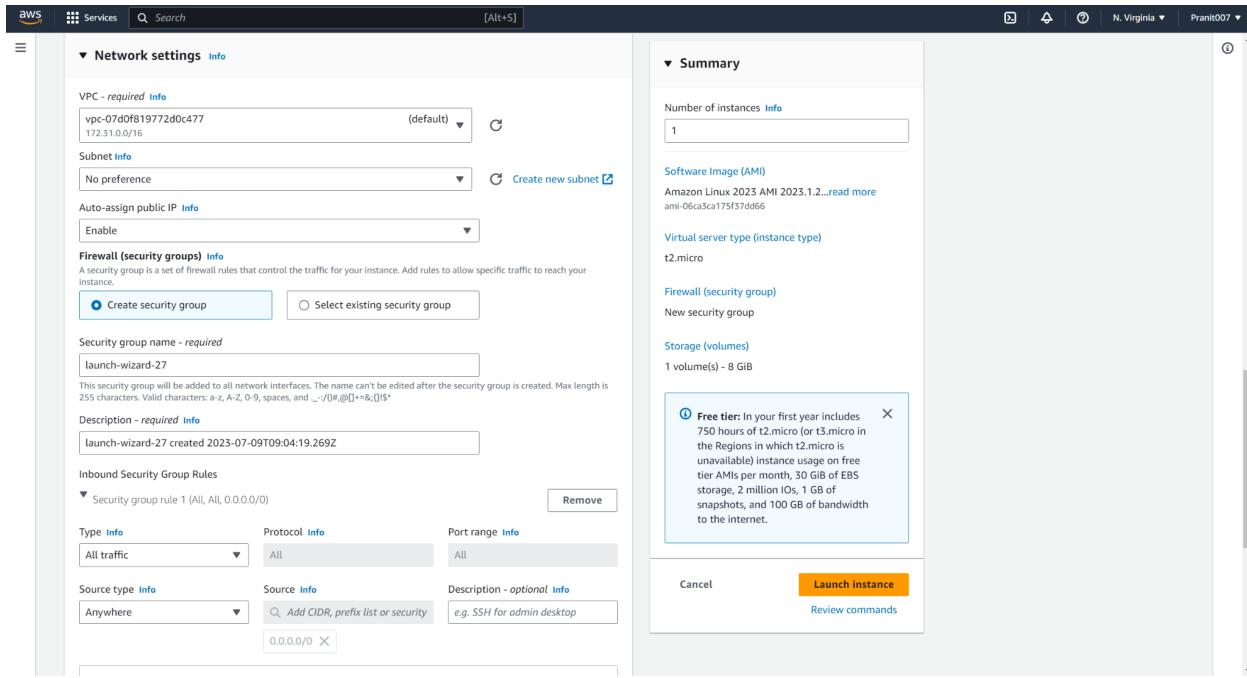
An 'Edit' button is located at the top right of the policy text area.

The screenshot shows the AWS IAM Roles page. A green header bar at the top indicates that 'Role CodeDeployRole01 created.' Below this, the main content area has a title 'Roles (13) Info'. It includes a search bar and buttons for 'Create role' and 'Delete'. A table lists 13 roles, each with a checkbox, 'Role name', 'Trusted entities', and 'Last activity'. The roles listed include AWSCodePipelineServiceRole, AWSCodePipelineServiceRoleForAmazonSSM, AWSCodePipelineServiceRoleForRecommender, AWSCodePipelineServiceRoleForSupport, AWSCodePipelineServiceRoleForTrustedAdvisor, CodeDeployRole, CodeDeployRole01, cwe-role-us-east-1-CodeDeployPipeline, cwe-role-us-east-1-Recommender-pipeline, EC2CodeDeploy, s3-role, s3_read_only, and s3_read_only_access_for_ec2.

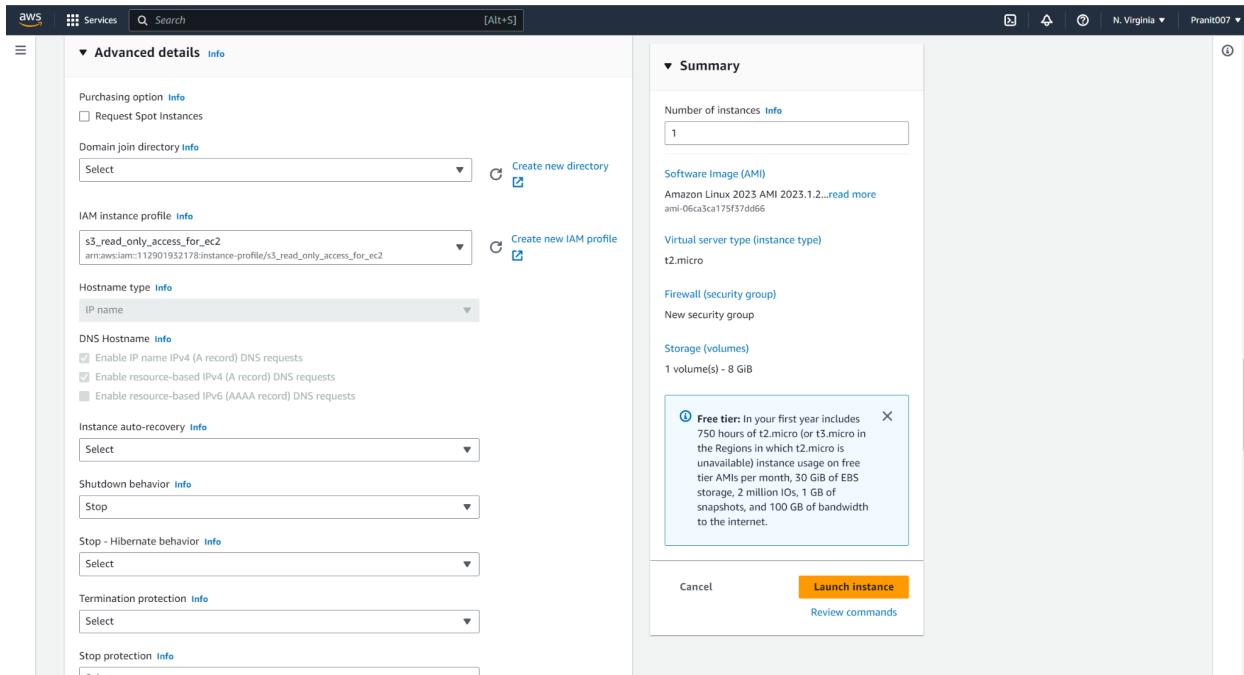
13. Now create a EC2 instance.

The screenshot shows the 'Launch an instance' wizard in the AWS EC2 service. The left sidebar shows the navigation path: EC2 > Instances > Launch an Instance. The main content area is divided into several sections: 'Name and tags' (with a 'Name' field containing 'Web-server'), 'Application and OS Images (Amazon Machine Image)' (with a search bar), 'Quick Start' (listing various AMI icons like Amazon Linux, macOS, Ubuntu, Windows, Red Hat, SUSE, and SUSI), and a summary section. The summary section includes fields for 'Number of instances' (set to 1), 'Software Image (AMI)' (Amazon Linux 2023 AMI 2023.1.2...), 'Virtual server type (instance type)' (t2.micro), 'Firewall (security group)' (New security group), and 'Storage (volumes)' (1 volume(s) - 8 GiB). A callout box highlights the 'Free tier' information: '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 I/Os, 1 GB of snapshots, and 100 GB of bandwidth to the internet.' At the bottom right are 'Cancel', 'Launch instance' (in orange), and 'Review commands' buttons.

14. Select a Key pair and add a security group with type “all traffic” and source type “anywhere”



15. Add the IAM role s3_read_only_access_for_ec2 and launch instance.



16. Now connect to the instance and run these commands sequentially to install codedeploy agent.

1.sudo su

2.sudo yum update

3.sudo yum install ruby -y

4.wget

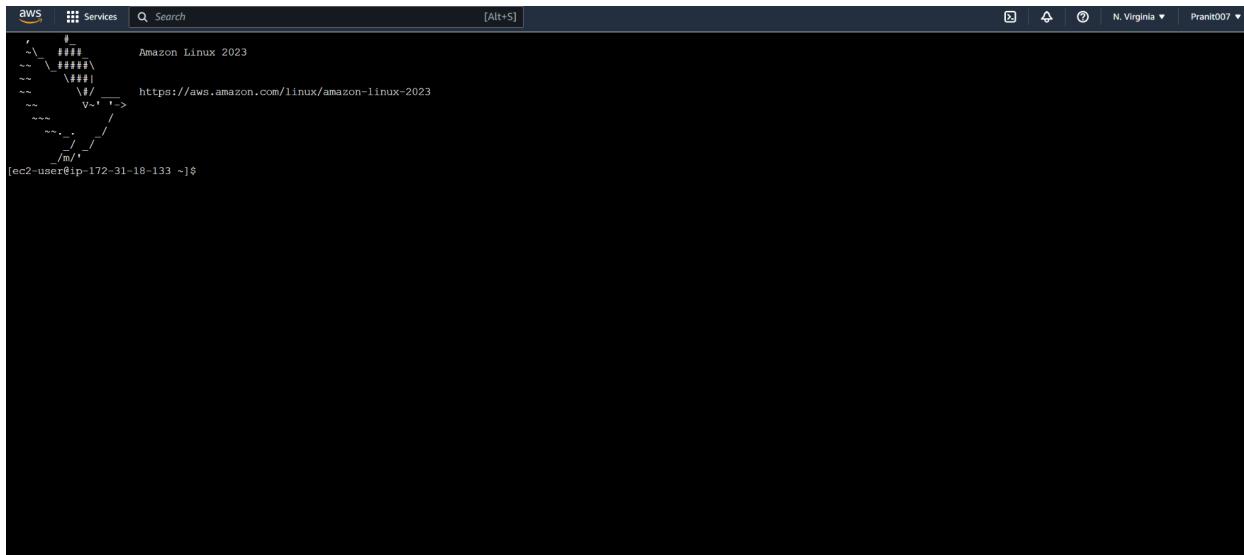
https://aws-codedeploy-us-east-1.s3.us-east-1.amazonaws.com/latest/install

5.chmod +x ./install

6.sudo ./install auto

7.sudo service codedeploy-agent start

8.sudo service codedeploy-agent status



The screenshot shows a terminal window with a scroll history. The last few lines show the execution of 'sudo yum update'.

```
Last login: Sun Jul  9 09:12:41 2023 from 18.206.107.28
[ec2-user@ip-172-31-18-133 ~]$ sudo yum update
Last metadata expiration check: 0:07:26 ago on Sun Jul  9 09:06:50 2023.
Dependencies resolved.
Nothing to do.
Complete!
[root@ip-172-31-18-133 ec2-user]# sudo yum install ruby -y
Last metadata expiration check: 0:07:37 ago on Sun Jul  9 09:06:50 2023.
Dependencies resolved.
```

Package	Architecture	Version	Repository	Size
Installing:				
ruby3_2	x86_64	3.2.2-180.amzn2023.0.1	amazonlinux	45 k
Installing dependencies:				
ruby3_2-default-gems	noarch	3.2.2-180.amzn2023.0.1	amazonlinux	36 k
ruby3_2-labs	x86_64	3.2.2-180.amzn2023.0.1	amazonlinux	4.0 M
ruby3_2-rubygem-io-console	x86_64	0.6.0-180.amzn2023.0.1	amazonlinux	28 k
ruby3_2-rubygem-json	x86_64	2.6.3-180.amzn2023.0.1	amazonlinux	56 k
ruby3_2-rubygem-psych	x86_64	5.0.1-180.amzn2023.0.1	amazonlinux	54 k
Installing weak dependencies:				
ruby3_2-rubygem-bigdecimal	x86_64	3.1.3-180.amzn2023.0.1	amazonlinux	71 k
ruby3_2-rubygem-bundler	noarch	2.4.10-180.amzn2023.0.1	amazonlinux	387 k
ruby3_2-rubygem-doc	noarch	6.5.0-180.amzn2023.0.1	amazonlinux	463 k
ruby3_2-rubygems	noarch	3.4.10-180.amzn2023.0.1	amazonlinux	259 k

The screenshot shows a terminal window with a scroll history. The last few lines show the execution of 'wget' and 'chmod'.

```
Completed!
[root@ip-172-31-18-133 ec2-user]# wget https://aws-codedeploy-us-east-1.s3.us-east-1.amazonaws.com/latest/install
2023-07-09 09:14:56 --2023-07-09 09:14:56-- https://aws-codedeploy-us-east-1.s3.us-east-1.amazonaws.com/latest/install
Resolving aws-codedeploy-us-east-1.s3.us-east-1.amazonaws.com (aws-codedeploy-us-east-1.s3.us-east-1.amazonaws.com) ... 52.216.16.117, 52.216.142.240, 52.216.200.134, ...
Connecting to aws-codedeploy-us-east-1.s3.us-east-1.amazonaws.com (aws-codedeploy-us-east-1.s3.us-east-1.amazonaws.com) [52.216.16.117]:443...
HTTP request sent, awaiting response... 200 OK
Length: 17892 (17K)
Saving to: 'install'

install          100%[=====] 17.47K --.-KB/s   in 0.001s

2023-07-09 09:14:56 (31.6 MB/s) - 'install' saved [17892/17892]

[root@ip-172-31-18-133 ec2-user]# chmod +x ./install
[root@ip-172-31-18-133 ec2-user]# sudo ./install auto
I: [2023-07-09T09:15:23.891118 #26217] INFO -- : Starting Ruby version check.
W: [2023-07-09T09:15:23.891328 #26217] WARNING -- : The current version in /usr/bin/ruby3.2 is 3.2.2, . Attempting to install anyway.
I: [2023-07-09T09:15:23.891338 #26217] INFO -- : Starting to detect supported package manager type for system...
I: [2023-07-09T09:15:23.892056 #26217] INFO -- : Attempting to automatically detect supported package manager type for system information...
I: [2023-07-09T09:15:23.896004 #26217] INFO -- : Checking AWS REGION environment variable for region information...
I: [2023-07-09T09:15:23.896139 #26217] INFO -- : Checking EC2 metadata service for region information...
I: [2023-07-09T09:15:23.909611 #26217] INFO -- : Checking AWS IAMADMIN environment variable for domain information...
I: [2023-07-09T09:15:23.909750 #26217] INFO -- : Checking EC2 metadata service for domain information...
I: [2023-07-09T09:15:23.916558 #26217] INFO -- : Downloading version file from bucket aws-codedeploy-us-east-1 and key latest/LATEST VERSION...
I: [2023-07-09T09:15:23.916778 #26217] INFO -- : Endpoint: https://aws-codedeploy-us-east-1.s3.us-east-1.amazonaws.com/latest/LATEST VERSION
I: [2023-07-09T09:15:24.035513 #26217] INFO -- : Downloading package from bucket aws-codedeploy-us-east-1 and key releases/codedeploy-agent-1.6.0-49.noarch.rpm...
I: [2023-07-09T09:15:24.035734 #26217] INFO -- : Endpoint: https://aws-codedeploy-us-east-1.s3.us-east-1.amazonaws.com/releases/codedeploy-agent-1.6.0-49.noarch.rpm
I: [2023-07-09T09:15:24.138373 #26217] INFO -- : Executing '/usr/bin/yum -y localinstall /tmp/codedeploy-agent-1.6.0-49.noarch.rpm'...
Last metadata expiration check: 0:08:34 ago on Sun Jul  9 09:06:50 2023.
Dependencies resolved.
```

Package	Architecture	Version	Repository	Size
Installing:				
codedeploy-agent	noarch	1.6.0-49	@commandline	2.7 M

```

AWS Services Search [Alt+S]
Installed size: 13 M
Downloading Packages:
Running transaction check
Transaction check succeeded.
Running transaction test
Transaction test succeeded.
Running transaction
Preparing :
  Running scriptlet: codedeploy-agent-1.6.0-49.noarch

pre hook : 1
Checking if there is already a process named codedeploy-agent running.

  Installing : codedeploy-agent-1.6.0-49.noarch
    Running scriptlet: codedeploy-agent-1.6.0-49.noarch

post hook : 1
Check if there is a codedeployagent config file.
Start codedeploy-agent in post hook if this is a first install.

  Verifying : codedeploy-agent-1.6.0-49.noarch

Installed:
  codedeploy-agent-1.6.0-49.noarch

Complete!
I: [2023-07-09T09:15:27.590364 #26217] INFO -- : Update check complete.
I: [2023-07-09T09:15:27.590531 #26217] INFO -- : Stopping updater.
[root@ip-172-31-18-133 ec2-user]# sudo service codedeploy-agent start
Starting codedeploy-agent:[root@ip-172-31-18-133 ec2-user]# sudo service codedeploy-agent status
/opt/codedeploy-agent/vendor/gems/glib-2.11.0/lib/glib/commands/help_modules/global_help_format.rb:37: warning: Passing safe_level with the 2nd argument of ERB.new is deprecated. Do not use it, and specify other arguments as keyword arguments.
/opt/codedeploy-agent/vendor/gems/glib-2.11.0/lib/glib/commands/help_modules/global_help_format.rb:37: warning: Passing trim_mode with the 3rd argument of ERB.new is deprecated. Use keyword arguments instead.
/opt/codedeploy-agent/vendor/gems/glib-2.11.0/lib/glib/commands/command_help_format.rb:27: warning: Passing safe_level with the 2nd argument of ERB.new is deprecated. Do not use it, and specify other arguments as keyword arguments.
/opt/codedeploy-agent/vendor/gems/glib-2.11.0/lib/glib/commands/help_modules/command_help_format.rb:27: warning: Passing trim_mode with the 3rd argument of ERB.new is deprecated. Use keyword arguments instead.
The AWS CodeDeploy agent is running as PID 26351
[root@ip-172-31-18-133 ec2-user]#

```

17. Now go to s3 bucket and upload a website.

Amazon S3 > Buckets > sample-website-bucket009

sample-website-bucket009 [Info](#)

[Objects](#) [Properties](#) [Permissions](#) [Metrics](#) [Management](#) [Access Points](#)

Objects (0)

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](#)

[C](#) [Copy S3 URI](#) [Copy URL](#) [Download](#) [Open](#) [Delete](#) [Actions](#) [Create folder](#) [Upload](#)

[Show versions](#)

Name	Type	Last modified	Size	Storage class
No objects You don't have any objects in this bucket.				
Upload				

The screenshot shows the AWS S3 'Upload' interface. At the top, there's a navigation bar with 'Services', a search bar, and user information ('Pranit007'). Below it, the path 'Amazon S3 > Buckets > sample-website-bucket009 > Upload' is visible. The main area is titled 'Upload' with a 'Info' link. A note says 'Add the files and folders you want to upload to S3. To upload a file larger than 150GB, use the AWS CLI, AWS SDK or Amazon S3 REST API.' with a 'Learn more' link. A large blue box allows dragging and dropping files. Below it, a table lists 'Files and folders (1 Total, 5.1 MB)'. One item, 'sample_website.zip', is shown with details: Name, Folder, Type (application/x-zip-compressed), and Size (5.1 MB). There are 'Remove', 'Add files', and 'Add folder' buttons. A 'Destination' section shows 'Destination' as 's3://sample-website-bucket009'. Under 'Permissions', it says 'Grant public access and access to other AWS accounts.' Under 'Properties', it says 'Specify storage class, encryption settings, tags, and more.' At the bottom are 'Cancel' and a prominent orange 'Upload' button.

18.Now search codedeploy and create a application

The screenshot shows the AWS CodeDeploy 'Create application' interface. The top navigation bar includes 'Services', a search bar, and user information ('Pranit007'). The path 'Developer Tools > CodeDeploy > Applications > Create application' is shown. The main title is 'Create application'. A 'Application configuration' section contains fields for 'Application name' (set to 'My-App'), 'Compute platform' (set to 'EC2/On-premises'), and 'Tags' (with an 'Add tag' button). At the bottom are 'Cancel' and a prominent orange 'Create application' button.

19.Create deployment group.

Application

Application
My-App
Compute type
EC2/On-premises

Deployment group name

Enter a deployment group name
MyApp-group
100 character limit

Service role

Enter a service role
Enter a service role with CodeDeploy permissions that grants AWS CodeDeploy access to your target instances.
arn:aws:iam::112901932178:role/CodeDeployRole01

Deployment type

Choose how to deploy your application
 In-place Blue/green

20.In environment configuration select EC2 and attached ec2 instance previously created.

Choose how to deploy your application

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

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

Environment configuration

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

Amazon EC2 Auto Scaling groups

Amazon EC2 instances
1 unique matched instance. [Click here for details](#)

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

Tag group 1

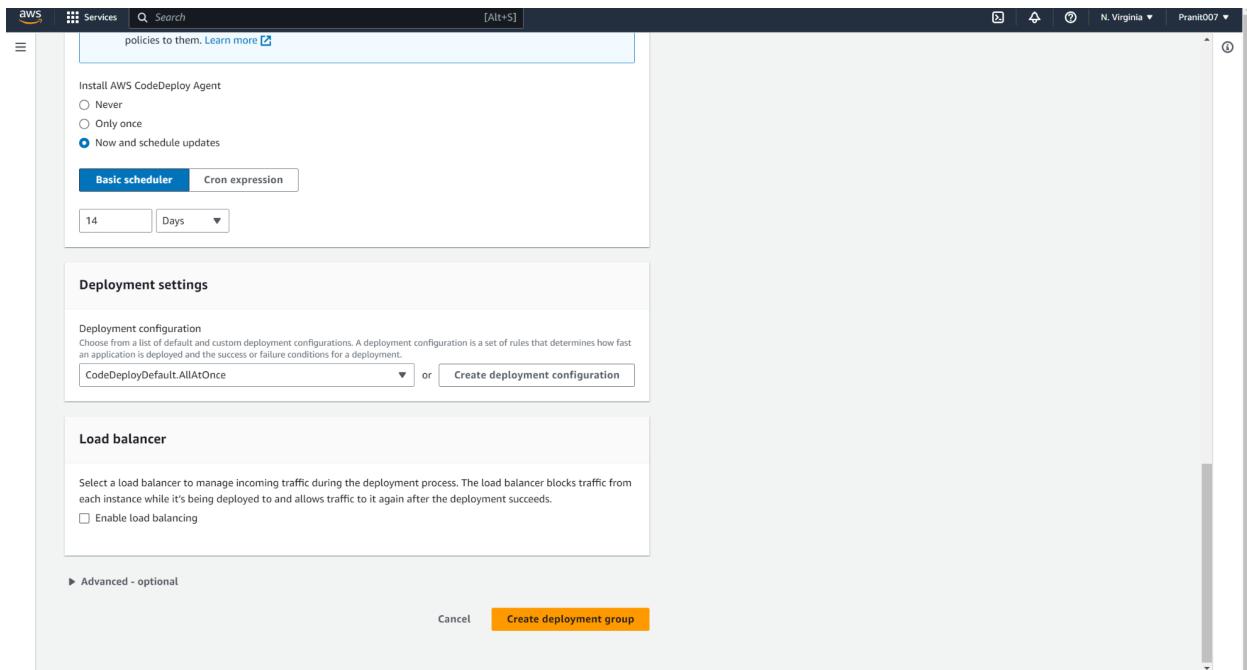
Key	Value - optional
<input type="text"/> Name	<input type="text"/> Web-server
Remove tag	
Add tag	
+ Add tag group	

On-premises instances

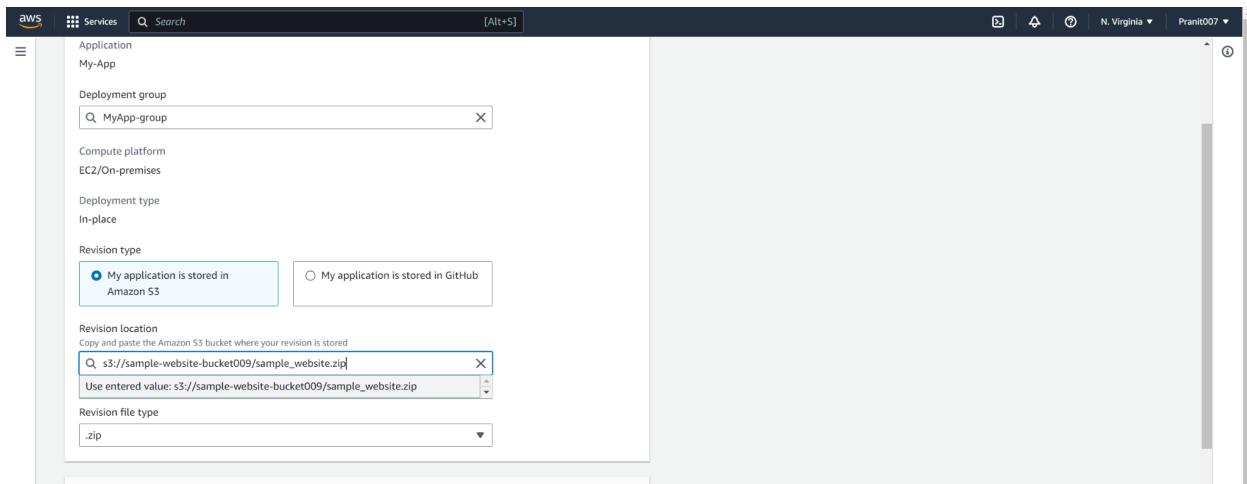
Matching instances

1 unique matched instance. [Click here for details](#)

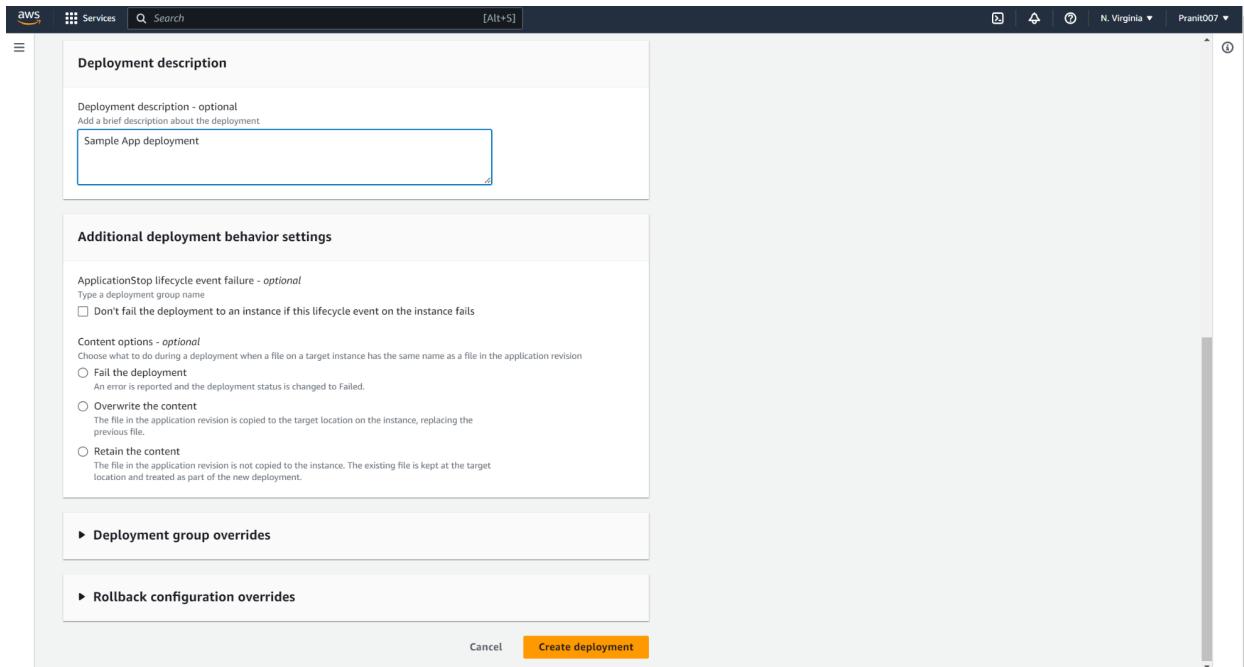
21.Unticked the load balancer and create deployment group.



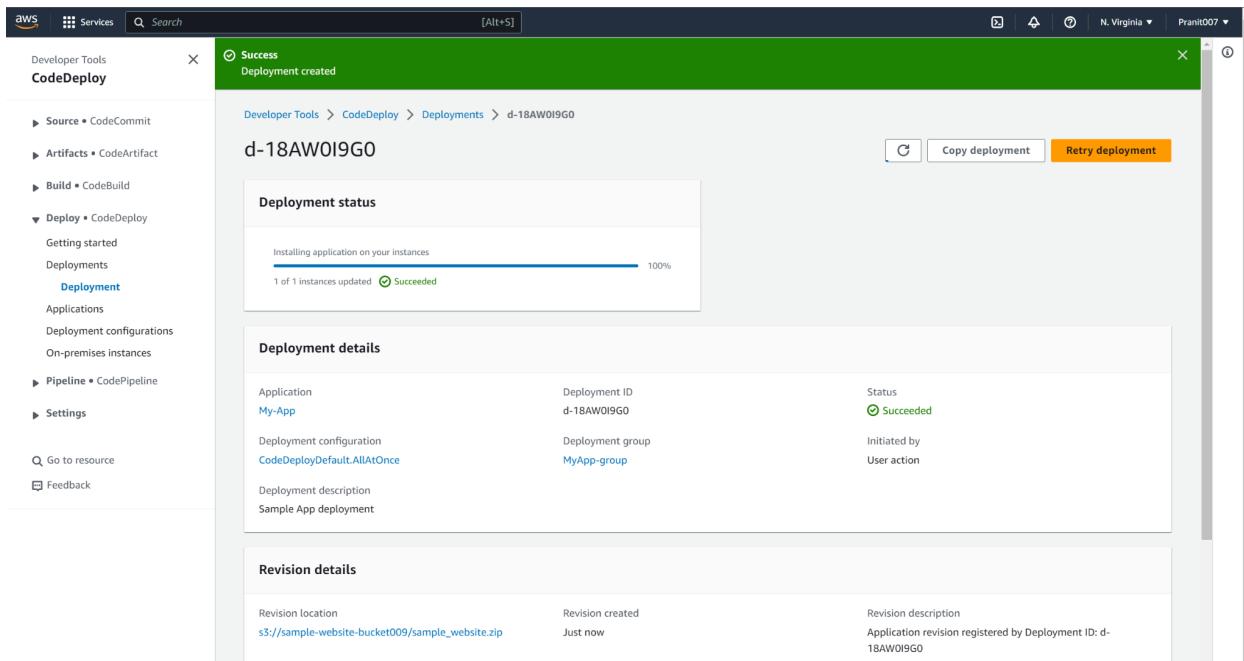
22. Now create deployment and copy the s3 url and pasted at Revision location.



23. Then create deployment.



24.After sometime deployment status will show succeeded.



25.Now go to instances and select the Web-server ec2 instance and copy the public IP.

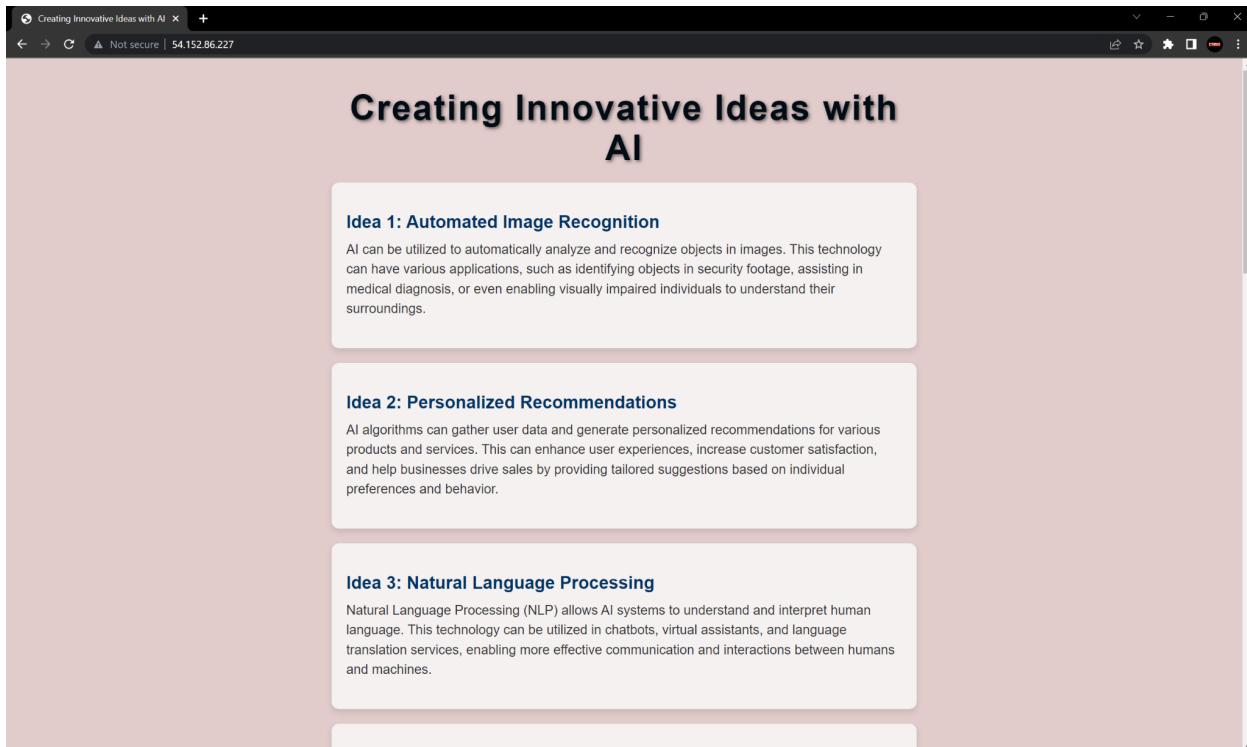
Instance: i-093ec4edaa771af0c (Web-server)

Details Security Networking Storage Status checks Monitoring Tags

▼ Instance summary Info

Instance ID	Public IPv4 address
i-093ec4edaa771af0c (Web-server)	54.152.86.227 open address
IPv6 address	Instance state
-	Running

26.Paste the Public IP on the browser.



27.Pasted the public IP on mobile and accessed the website.

