

Semester	T.E. Semester V – Information Technology
Subject	Advance DevOps Lab
Subject Professor	Prof. Indu Anoop
In-charge	
Laboratory	

Student Name	Soham Anant Sawant	
Roll Number	21101B2002	
Grade and Subject Teacher's Signature		

Experiment	1	
Problem Statement	To understand the benefits of Cloud Infrastructure and Setup AWS Cloud9 IDE, Launch AWS Cloud9 IDE, and Perform Collaboration Demonstration.	
Resources / Apparatus Required	Hardware: Computer System	Software: Web Browser
Details	<p>Theory: AWS Cloud9 is a cloud-based integrated development environment (IDE) that lets you write, run, and debug your code with just a browser. It includes a code editor, debugger, and terminal. Cloud9 comes pre-packaged with essential tools for popular programming languages, including JavaScript, Python, PHP, and more, so you don't need to install files or configure your development machine to start new projects. Since your Cloud9 IDE is cloud-based, you can work on your projects from your office, home, or anywhere using an internet-connected machine. Cloud9 also provides a seamless experience for developing serverless applications enabling you to easily define resources, debug, and switch between local and remote execution of serverless applications. With Cloud9, you can quickly share your development environment with your team, enabling you to pair program and track each other's inputs in real time.</p> <p>Benefits:</p> <p>CODE WITH JUST A BROWSER</p> <p>AWS Cloud9 gives you the flexibility to run your development environment on a managed Amazon EC2 instance or any existing Linux server that</p>	

supports SSH. This means that you can write, run, and debug applications with just a browser, without needing to install or maintain a local IDE. The Cloud9 code editor and integrated debugger include helpful, time-saving features such as code hinting, code completion, and step-through debugging. The Cloud9 terminal provides a browser-based shell experience enabling you to install additional software, do a git push, or enter commands.

CODE TOGETHER IN REAL TIME

AWS Cloud9 makes collaborating on code easy. You can share your development environment with your team in just a few clicks and pair program together. While collaborating, your team members can see each other type in real time, and instantly chat with one another from within the IDE.

BUILD SERVERLESS APPLICATIONS WITH EASE

AWS Cloud9 makes it easy to write, run, and debug serverless applications. It preconfigures the development environment with all the SDKs, libraries, and plug-ins needed for serverless development. Cloud9 also provides an environment for locally testing and debugging AWS Lambda functions. This allows you to iterate on your code directly, saving you time and improving the quality of your code.

DIRECT TERMINAL ACCESS TO AWS

AWS Cloud9 comes with a terminal that includes sudo privileges to the managed Amazon EC2 instance that is hosting your development environment and a pre-authenticated AWS Command Line Interface. This makes it easy for you to quickly run commands and directly access AWS services

START NEW PROJECTS QUICKLY

AWS Cloud9 makes it easy for you to start new projects. Cloud9's development environment comes pre-packaged with tooling for over 40 programming languages, including Node.js, JavaScript, Python, PHP, Ruby, Go, and C++. This enables you to start writing code for popular application stacks within minutes by eliminating the need to install or configure files, SDKs, and plug-ins for your development machine. Because Cloud9 is cloud-based, you can easily maintain multiple development environments to isolate your project's resources.

For more info related to AWS-Cloud 9 you all can refer following Docs.
<https://docs.aws.amazon.com/cloud9/latest/user-guide/aws-cloud9-ug.pdf>

Code	<p>Steps:</p> <ol style="list-style-type: none"> 1. Login with your AWS root account. https://aws.amazon.com/console/ 2. Open IAM (Identity and Access Management) to Add two users. Provide manual password if you want and click on Next permission tab. 3. Click on Create group. Provide group name and attach AwsCloud9EnvironmentMember policy to group. Click on create group. 4. After that group is created click on next if u want to provide tag else click on Review for user settings and click on create user 5. Now Open your Browsers Incognito Window and login with IAM user1 which you configured before. [Cloud9 Environment should be created through IAM user account only and not root account due to security reasons) 6. Navigate to Cloud 9 service from Developer tools section 7. Click on Create Environment 8. Provide name for the Environment (For e.g.: WebAppIDE) and click on next. 9. Keep all the Default settings 10. Review the Environment name and Settings and click on Create Environment 11. Click on cloud9 IDE Environment tab. If you check at bottom side Cloud9 IDE there is AWS CLI for command operations: git version, IAM user1 details. Explore settings where you can update permissions of your teammates as from RW to R only or you can remove user too. 12. Now we will setup collaborative environment. Click on File you can create new file or choose from template, we can opt for html file to collaborate. Edit html file and save it 13. To share this file to collaborate with other members of your team click on Share option on Right Pane and username which you created in IAM before to Invite members and enable permissions as RW (Read and Write) and click on Done. Click OK for Security warning. 14. Now Open your Browsers Incognito Window and login with IAM user2 which you configured before. 15. After Successful login with IAM user2, open Cloud9 service from dashboard services and click on shared with you option from left side panel 16. Double click the IDE that was shared by IAM user1 , you will get same interface as your other member to collaborate in real time, also everyone within team can do group chats.
Output	<p>Output screenshots:</p> <ol style="list-style-type: none"> 1. Login with your AWS root account. https://aws.amazon.com/console/

The image consists of three vertically stacked screenshots of the AWS Management Console interface.

- Screenshot 1: Developer Tools**
This screenshot shows the "Developer Tools" section. It includes links to Cloud9 (A Cloud IDE for Writing, Running, and Debugging Code), CloudShell (A browser-based shell with AWS CLI access from the AWS Management Console), CodeArtifact (Secure, scalable, and cost-effective artifact management for software development), CodeBuild (Build and Test Code), CodeCommit (Store Code in Private Git Repositories), and CodeDeploy (Automate Code Deployments). A sidebar on the left lists "Recently visited", "Favorites", and "All services". The top navigation bar shows "aws Services Search for services, features, blogs, docs, and more [Alt+S]" and the bottom status bar shows "Feedback Looking for language selection? Find it in the new Unified Settings [?] © 2022, Amazon Internet Services Private Ltd. or its affiliates. Privacy Terms Cookie preferences EN 1:38 PM 23-Aug-22".
- Screenshot 2: End User Computing**
This screenshot shows the "End User Computing" section. It includes links to Directory Service (Host and Manage Active Directory), AWS Firewall Manager (Central management of firewall rules), GuardDuty (Intelligent Threat Detection to Protect Your AWS Accounts and Workloads), and IAM (Identity and Access Management). A sidebar on the left lists various computing services like Front-end Web & Mobile, Game Development, Internet of Things, Machine Learning, Management & Governance, Media Services, Migration & Transfer, Networking & Content Delivery, Quantum Technologies, Robotics, Satellite, Security, Identity, & Compliance, and Storage. The top navigation bar shows "aws Services Search for services, features, blogs, docs, and more [Alt+S]" and the bottom status bar shows "Investigate and analyze potential security issues [?] © 2022, Amazon Internet Services Private Ltd. or its affiliates. Privacy Terms Cookie preferences EN 1:38 PM 23-Aug-22".
- Screenshot 3: IAM**
This screenshot shows the "IAM" (Identity and Access Management) section. It includes links to IAM Identity Center (successor to AWS Single Sign-On), Inspector (Analyze Application Security), and Key Management Service (Securely Generate and Manage AWS Encryption Keys). A sidebar on the left lists various computing services like Front-end Web & Mobile, Game Development, Internet of Things, Machine Learning, Management & Governance, Media Services, Migration & Transfer, Networking & Content Delivery, Quantum Technologies, Robotics, Satellite, Security, Identity, & Compliance, and Storage. The top navigation bar shows "aws Services Search for services, features, blogs, docs, and more [Alt+S]" and the bottom status bar shows "Amazon Macie [?] © 2022, Amazon Internet Services Private Ltd. or its affiliates. Privacy Terms Cookie preferences EN 1:40 PM 23-Aug-22".

2. From services, Open **IAM** (Identity and Access Management) to Add two users [For example: soham_iam1, soham_iam2] Provide manual password if you want and click on Next permission tab.

The image consists of three vertically stacked screenshots of the AWS IAM Management Console. The top screenshot shows the 'Users' page with a message about the new users list experience. The middle screenshot shows the 'Add users' step where two users are being created: 'soham_iam1' and 'soham_iam2'. The bottom screenshot shows the 'Select AWS access type' step for the first user, with 'Password - AWS Management Console access' selected. A note at the bottom of the page says '3. Click on **Create group**. Provide group name and attach AwsCloud9EnvironmentMember policy to group. Click on create group.'

Introducing the new Users list experience
We've redesigned the Users list experience to make it easier to use. Let us know what you think.

Users (0) 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

No resources to display

User name*

soham_iam1

soham_iam2

Add another user

Select AWS access type

Select how these users will primarily access AWS. If you choose only programmatic access, it does NOT prevent users from accessing the console using an assumed role. Access keys and autogenerated passwords are provided in the last step. [Learn more](#)

Select AWS credential type*

- Access key - Programmatic access
Enables an **access key ID** and **secret access key** for the AWS API, CLI, SDK, and other development tools.
- Password - AWS Management Console access
Enables a **password** that allows users to sign-in to the AWS Management Console.

Console password*

- Autogenerated password
- Custom password

Show password

* Required

Cancel Next: Permissions

Feedback Looking for language selection? Find it in the new Unified Settings. © 2022, Amazon Internet Services Private Ltd. or its affiliates. Privacy Terms Cookie preferences EN 1:42 PM 23-Aug-22

Feedback Looking for language selection? Find it in the new Unified Settings. © 2022, Amazon Internet Services Private Ltd. or its affiliates. Privacy Terms Cookie preferences EN 1:44 PM 23-Aug-22

3. Click on **Create group**. Provide group name and attach AwsCloud9EnvironmentMember policy to group. Click on create group.

Screenshot of the AWS IAM Management Console showing the 'Add user' process. The 'Set permissions' step is selected, displaying options to 'Add users to group', 'Copy permissions from existing user', or 'Attach existing policies directly'. A 'Create group' button is visible. Below this, the 'Set permissions boundary' step is shown.

Create group

Group name: soham_cloud9

Filter policies: cloud9

Policy name	Type	Used as	Description
AWSCloud9Administrator	AWS managed	None	Provides administrator access to AWS Cloud9.
AWSCloud9Environment...	AWS managed	None	Provides the ability to be invited into AWS Cloud9 shared de...
AWSCloud9SSMInstance...	AWS managed	None	This policy will be used to attach a role on a InstanceProfile ...
AWSCloud9User	AWS managed	None	Provides permission to create AWS Cloud9 development en...

Buttons: Cancel, Create group

Text at the bottom: 4. After that group is created click on next if u want to provide tag else click on **Review** for user settings and click on **create users**. Copy the sign in access URL for IAM users that is generated.

Add tags (optional)

IAM tags are key-value pairs you can add to your user. Tags can include user information, such as an email address, or can be descriptive, such as a job title. You can use the tags to organize, track, or control access for this user. [Learn more](#)

Key	Value (optional)
<input type="text" value="Add new key"/>	<input type="text"/>

You can add 50 more tags.

Review

Review your choices. After you create the users, you can view and download autogenerated passwords and access keys.

User details

User names	soham_iam1 and soham_iam2
AWS access type	AWS Management Console access - with a password
Console password type	Custom
Require password reset	No
Permissions boundary	Permissions boundary is not set

Permissions summary

The users shown above will be added to the following groups.

Type	Name
Group	soham_cloud9

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.

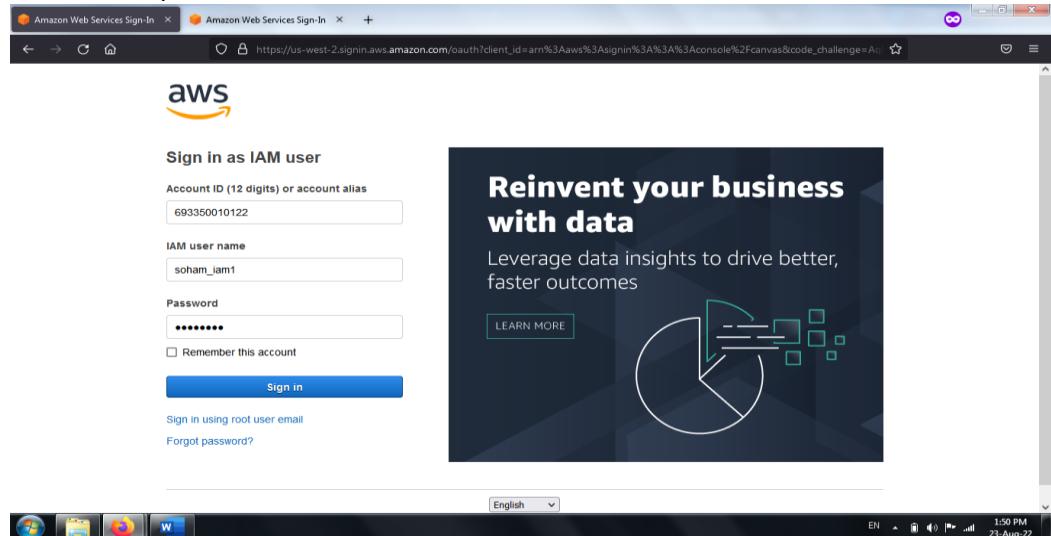
Users with AWS Management Console access can sign-in at: <https://693350010122.signin.aws.amazon.com/console>

Download .CSV

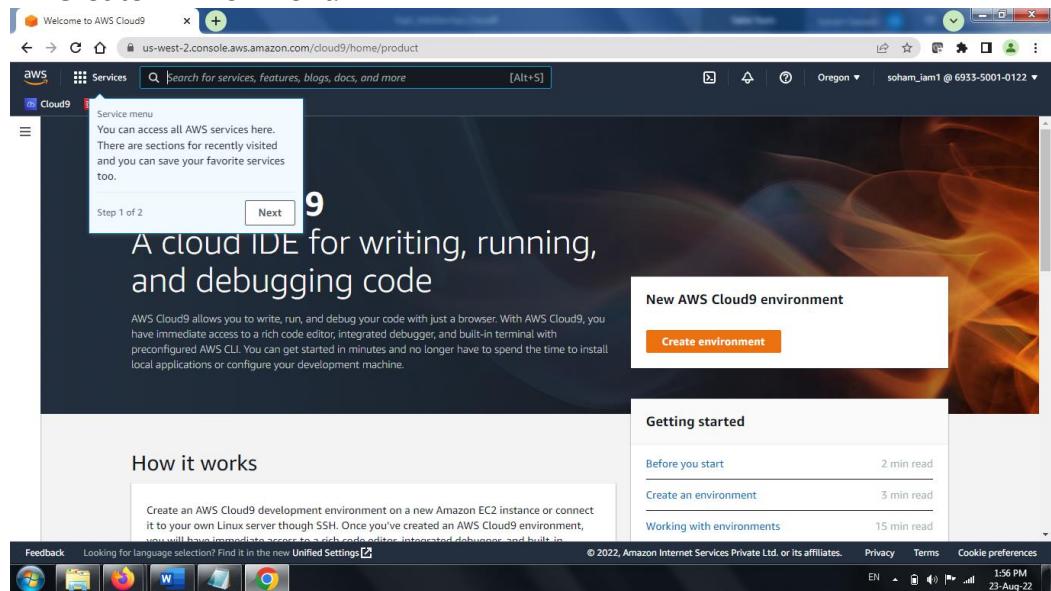
User	Email login instructions
soham_iam1	Send email
soham_iam2	Send email

Close

5. Log out from root account and paste the URL of copied IAM user sign in to Browsers address box and login as IAM user1 (i.e, soham_iam1) which was configured before. [Cloud9 Environment should be created through IAM user account only and not root account due to security reasons).



6. Navigate to Cloud 9 service from Developer tools section and Click on Create Environment.



7. Provide name for the Environment (For e.g.: WebAppIDE) and click on next. Keep all the Default settings. Review the Environment name and Settings and click on Create Environment.

Create a new environment

us-west-2.console.aws.amazon.com/cloud9/home/create

Name environment

Step 1 Name environment Step 2 Configure settings Step 3 Review

Environment name and description

Name
The name needs to be unique per user. You can update it at any time in your environment settings.

Description - Optional
This will appear on your environment's card in your dashboard. You can update it at any time in your environment settings.

Limit: 200 characters

Next step

AWS Cloud9

Your environments Shared with you Account environments How-to guide

Create a new environment

us-west-2.console.aws.amazon.com/cloud9/home/create

AWS Cloud9

Your environments Shared with you Account environments How-to guide

Create a new environment

us-west-2.console.aws.amazon.com/cloud9/home/create

AWS Cloud9

Your environments Shared with you Account environments How-to guide

Create a new environment

us-west-2.console.aws.amazon.com/cloud9/home/create

Review

Environment name and settings

Name
WebAppIDE

Description
This is ADEV Exp 1

Environment type
EC2

Instance type
t2.micro

Platform
Amazon Linux 2 (recommended)

Cost-saving setting
After 30 minutes (default)

IAM role
AWSServiceRoleForAWSCloud9 (generated)

We recommend the following best practices for using your AWS Cloud9 environment

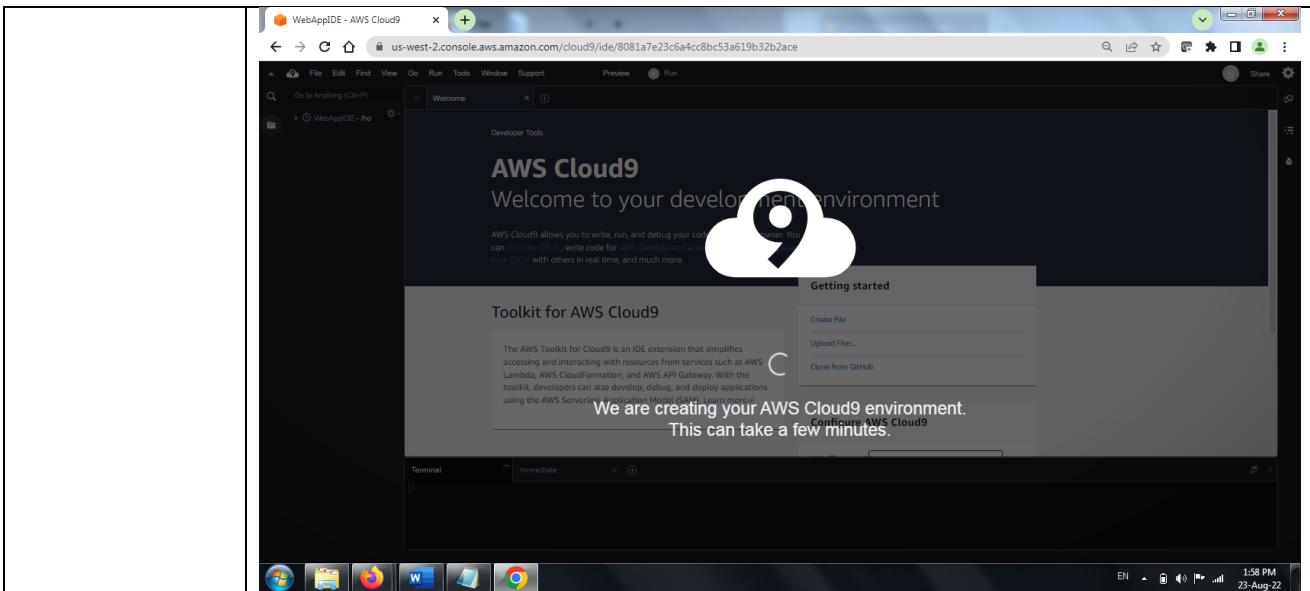
- Use source control and backup your environment frequently. AWS Cloud9 does not perform automatic backups.

EN 1:57 PM 23-Aug-22

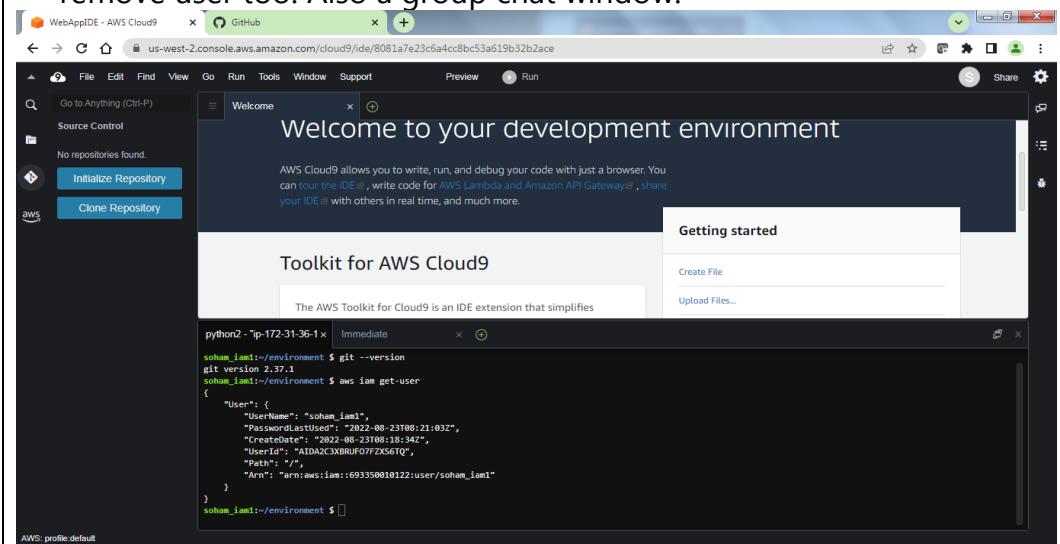
Feedback Looking for language selection? Find it in the new Unified Settings © 2022, Amazon Internet Services Private Ltd. or its affiliates. Privacy Terms Cookie preferences

EN 1:58 PM 23-Aug-22

Oregon soham_iam1 @ 6933-5001-0122



8. If you check at bottom side Cloud9 IDE there is AWS CLI for command operations: git version, IAM user details. Explore settings where you can update permissions of your teammates as from RW to R only or you can remove user too. Also a group chat window.



9. Now we will setup collaborative environment. Click on File you can create new file or choose from template, we can opt for html file to collaborate. Edit html file and save it (Ctrl+S).

The screenshot shows two instances of the AWS Cloud9 IDE running side-by-side.

Top Window:

- File Menu:** Shows options like New File, New From Template, Open..., Save, and Upload Local Files... under the 'File' tab.
- Code Editor:** An 'Untitled1.html' file is open, showing the following code:

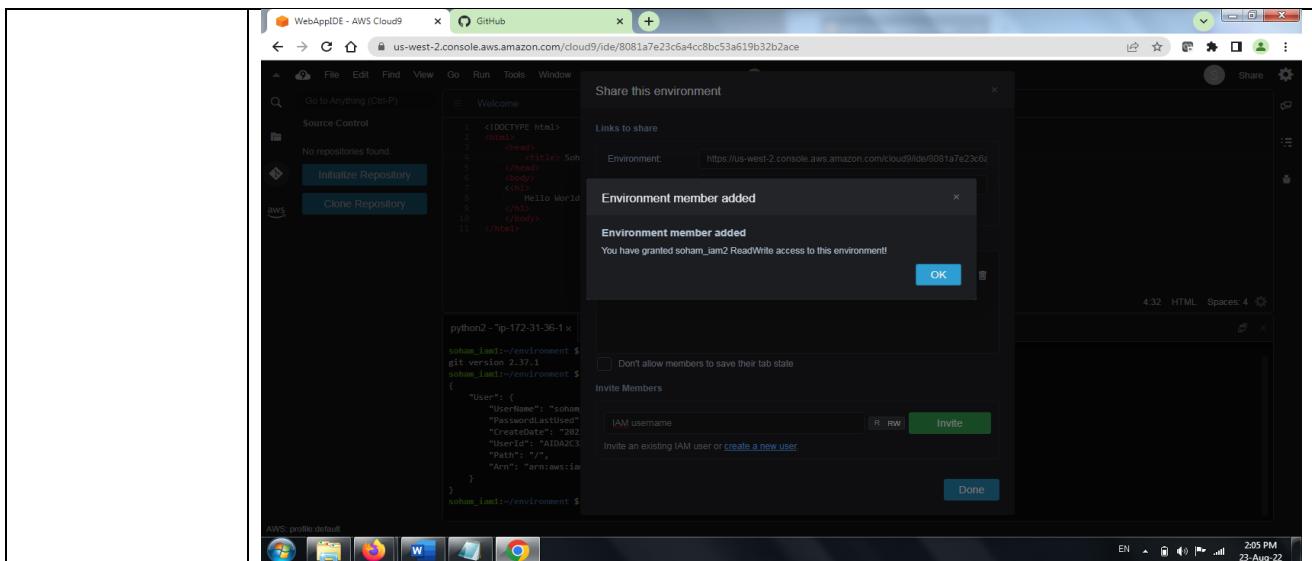

```
<!DOCTYPE html>
<html>
  <head>
    <title> Sphm_ADFV_Fntr </title>
  </head>
  <body>
    <h1> Hello World! </h1>
  </body>
</html>
```
- Terminal:** Shows command-line output for creating an IAM user named 'soham_iam1' and listing its details:


```
git version 2.37.1
soham_iam1:~/environment $ aws iam get-user
{
  "User": {
    "UserName": "soham_iam1",
    "PasswordLastUsed": "2022-08-23T08:21:03Z",
    "CreateDate": "2022-08-23T08:18:34Z",
    "UserId": "AIDA2Z3XMRUJ07F2X5G7Q",
    "Path": "/",
    "Arn": "arn:aws:iam::693350010122:user/soham_iam1"
  }
}
soham_iam1:~/environment $
```

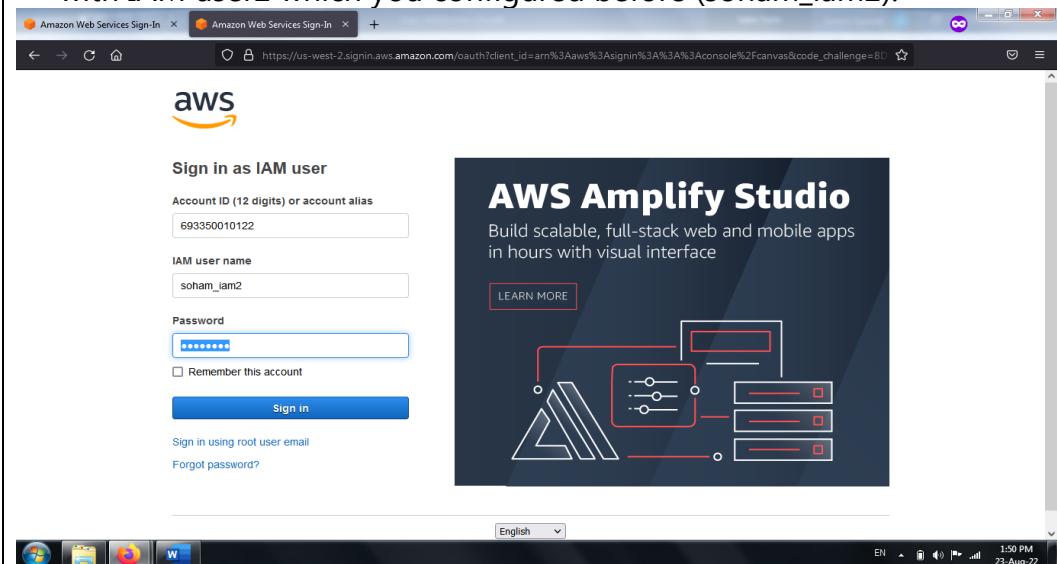
Bottom Window:

- File Menu:** Shows options like Go to Anything (Ctrl+P), Initialize Repository, and Clone Repository under the 'File' tab.
- Code Editor:** An 'Untitled1.html' file is open, showing the same HTML code as the top window.
- Save As Dialog:** A 'Save As' dialog is open, prompting for a filename ('first.html') and a folder ('WebAppIDE - /home/ec2-user/environment'). It includes checkboxes for 'Create folder' and 'Show files in tree'.
- Terminal:** Shows command-line output for creating an IAM user named 'soham_iam1' and listing its details, identical to the top window's terminal output.

10. To share this file to collaborate with other members of your team click on Share option on Top Right Pane and enter username of IAM user2 which you created in before (soham_iam2) to Invite members and enable permissions as RW (Read and Write) and click on Done. Click OK for Security warning. Click OK and Done.



11. Now Open your Browsers Incognito Window, paste URL of IAM user sign -in which was copied before into address box of browser and login with IAM user2 which you configured before (soham_iam2).



12. After Successful login with IAM user2, open Cloud9 service from dashboard services and click on shared with you option from left side panel.

Step 1 of 2 9 A cloud IDE for writing, running, and debugging code

AWS Cloud9 allows you to write, run, and debug your code with just a browser. With AWS Cloud9, you have immediate access to a rich code editor, integrated debugger, and built-in terminal with preconfigured AWS CLI. You can get started in minutes and no longer have to spend the time to install local applications or configure your development machine.

New AWS Cloud9 environment

Create environment

Getting started

Before you start 2 min read

Create an environment 3 min read

Working with environments 15 min read

Feedback Looking for language selection? Find it in the new Unified Settings.

© 2022, Amazon Internet Services Private Ltd. or its affiliates. Privacy Terms Cookie preferences

13. Double click the IDE/Click on Open IDE button of WebAppIDE that was shared by IAM user1, you will get same interface as your other member to collaborate in real time, also everyone within team can do group chats.

Shared with you

Shared with you (1)

Open IDE View details Edit Delete Create environment

WebAppIDE

Type EC2 Permissions Read-write

Description This is ADEV Exp 1

Owner Arn arn:aws:iam::693350010122:user/soham_jam1

Open IDE

Feedback Looking for language selection? Find it in the new Unified Settings.

File Edit Find View Go Run Tools Window Support

Welcome first.html

ENVIRONMENT MEMBERS ReadWrite soham_jam2 (online) You (online) R RW

GROUP CHAT Chat history is stored on the environment and can be both read and modified by ReadWrite members.

soham_jam2 Hi about a minute ago

You Hello less than a minute ago

Enter your message here

python2 ->ip-172-x Immediate

```
1 <!DOCTYPE html>
2 <html>
3   <head>
4     <title> Soham ADEV Exp1</title>
5   </head>
6   <body>
7     <h1>
8       Hello Soham IAM1
9     </h1>
10    </body>
11  </html>
```

432 HTML Spaces 4

soham_jam1:~/environment \$ aws iam get-user

```
{
  "User": {
    "UserName": "soham_jam1",
    "PasswordLastUsed": "2022-08-23T08:18:34Z",
    "CreateDate": "2022-08-23T08:18:34Z",
    "UserId": "AIDAC2X8RUFO7F2X56TQ",
    "Path": "/",
    "Arn": "arn:aws:iam::693350010122:user/soham_jam1"
  }
}
```

soham_jam1:~/environment \$

File Edit Find View Go Run Tools Window Support Preview Run

first.html

ENVIRONMENT MEMBERS ReadWrite You (online) soham_jam1 (online) R RW

GROUP CHAT Chat history is stored on the environment and can be both read and modified by ReadWrite members.

You Hi about a minute ago

soham_jam1 Hello less than a minute ago

Enter your message here

bash ->ip-172-x Immediate

soham_jam1:~/environment \$

Enter your message here

EN 2:11 PM 23-Aug-22

Conclusion

Learnt collaboration using AWS Cloud9 IDE.

Semester	T.E. Semester V – Information Technology
Subject	Advance DevOps Lab
Subject Professor In-charge	Prof. Indu Anoop
Laboratory	

Student Name	Soham Anant Sawant	
Roll Number	21101B2002	
Grade and Subject Teacher's Signature		

Experiment	5	
Problem Statement	To understand terraform lifecycle, core concepts/terminologies and install it on a Linux Machine	
Resources / Apparatus Required	Hardware: Computer System	Software: Web Browser
Details	<p>Terraform</p> <p>Terraform is an infrastructure as code (IaC) tool that allows you to build, change, and version infrastructure safely and efficiently. This includes low-level components such as compute instances, storage, and networking, as well as high-level components such as DNS entries, SaaS features, etc. Terraform can manage both existing service providers and custom in-house solutions.</p> <p>Key Features</p> <p>Infrastructure as Code:</p> <p>You describe your infrastructure using Terraform's high-level configuration language in human-readable, declarative configuration files. This allows you to create a blueprint that you can version, share, and reuse.</p> <p>Resource Graph</p> <p>Terraform builds a resource graph and creates or modifies non-dependent resources in parallel. This allows Terraform to build resources as efficiently as possible and gives you greater insight into your infrastructure.</p>	

Change Automation

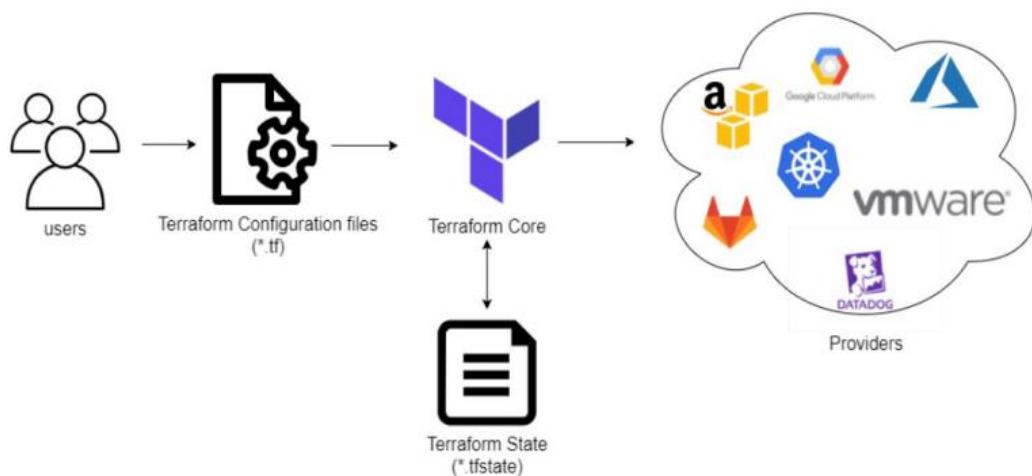
Terraform can apply complex changesets to your infrastructure with minimal human interaction. When you update configuration files, Terraform determines what changed and creates incremental execution plans that respect dependencies.

How does Terraform work?

Terraform works with two major components:

one is the **terraform core**: it takes the terraform configuration which is being provided by the user and then takes the terraform state which is managed by terraform itself. As such, this gets fed into the core that is responsible for figuring out what is that graph of our different resources for example how these different pieces relate to each other or what needs to be created/updated/destroyed, it does all the essential lifecycle management.

On the backside, terraform supports many different **providers**, such as: cloud providers (AWS,GCP,AZURE) and they also could be on-premise infrastructure (VMware, OpenStack.) But this support is not restricted or limited only to Infrastructure As A Service , terraform can also manage higher level like Platform As A Service(Kubernetes, Lambdas..)or even Software As A Service (DataDog, GitHub..)



All of these are important pieces of the infrastructure, they are all part of the logical end-to-end delivery.

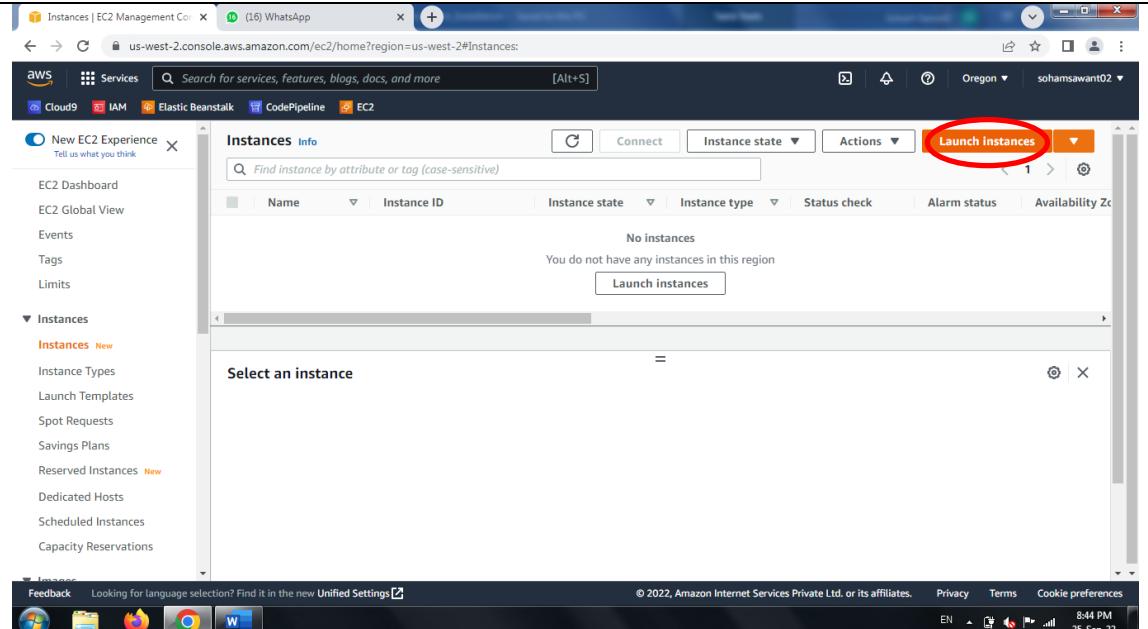
Terraform has over a hundred providers for different technologies, and each provider gives terraform users access to their resources. It also gives you the ability to create infrastructure at different levels.

Terraform Core Concepts:

Below are the core concepts/terminologies used in Terraform:

- **Variables:** Also used as input-variables, it is a key-value pair used by Terraform modules to allow customization.
- **Provider:** It is a plugin to interact with APIs of service and access its related resources.
- **Module:** It is a folder with Terraform templates where all the configurations are defined
- **State:** It consists of cached information about the infrastructure managed by Terraform and its related configurations.
- **Resources:** It refers to a block of one or more infrastructure objects (compute instances, virtual networks, etc.), which are used in configuring and managing the infrastructure.
- **Data Source:** It is implemented by providers to return information on external objects to terraform.
- **Output Values:** These are return values of a terraform module that can be used by other configurations.
- **Plan:** It is one of the stages where it determines what needs to be created, updated, or destroyed to move from the real/current state of the infrastructure to the desired state.
- **Apply:** It is one of the stages where it applies the changes in the real/current state of the infrastructure to move to the desired state.

Code



The screenshot shows the AWS EC2 'Launch an instance' page. At the top, a message states: 'You've been opted into the new launch experience. You can return to the previous version, but next time you log in, you'll automatically be opted into the new experience. [Find out more](#) or [send us feedback](#). Starting October 1, 2022, we will begin decommissioning the previous version.' A red circle highlights the 'Opt out to the old experience' button.

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: e.g. My Web Server

Software Image (AMI)

Amazon Linux 2 Kernel 5.10 AMI... [read more](#)
ami-08e2d37b6a0129927

Virtual server type (instance type)

t2.micro

Firewall (security group)

New security group

Launch instance

The screenshot shows the 'Step 1: Choose an Amazon Machine Image (AMI)' page of the launch wizard. It lists three AMI options:

- Ubuntu Server 20.04 LTS (HVM), SSD Volume Type** - ami-0c09c7eb16d3e8e70 (64-bit x86) / ami-02ba3245f79d1c136
 Select
- Ubuntu Server 20.04 LTS (HVM), EBS General Purpose (SSD) Volume Type** - Support available from Canonical (<http://www.ubuntu.com/cloud/services>).
 64-bit (x86)
 64-bit (Arm)
- Microsoft Windows Server 2022 Base** - ami-0df583d5f9d9e6cd8
 Select
64-bit (x86)

Feedback Looking for language selection? Find it in the new [Unified Settings](#)

© 2022, Amazon Internet Services Private Ltd. or its affiliates. Privacy Terms Cookie preferences

EN 8:47 PM 25-Sep-22

Step 2: Choose an Instance Type

Amazon EC2 provides a wide selection of instance types optimized to fit different use cases. Instances are virtual servers that can run applications. They have varying combinations of CPU, memory, storage, and networking capacity, and give you the flexibility to choose the appropriate mix of resources for your applications. [Learn more](#) about instance types and how they can meet your computing needs.

Family	Type	vCPUs	Memory (GiB)	Instance Storage (GB)	EBS-Optimized Available	Network Performance	IPv6 Support
t2	t2.nano	1	0.5	EBS only	-	Low to Moderate	Yes
t2	t2.micro <small>Free tier eligible</small>	1	1	EBS only	-	Low to Moderate	Yes
t2	t2.small	1	2	EBS only	-	Low to Moderate	Yes
t2	t2.medium	2	4	EBS only	-	Low to Moderate	Yes
t2	t2.large	2	8	EBS only	-	Low to Moderate	Yes

Step 3: Configure Instance Details

Configure the instance to suit your requirements. You can launch multiple instances from the same AMI, request Spot instances to take advantage of the lower pricing, assign an access management role to the instance, and more.

Number of instances: 1

Purchasing option: Request Spot Instances

Network: vpc-0e76d9fc9d10c151a (default)

Subnet: No preference (default subnet in any Availability Zone)

Auto-assign Public IP: Use subnet setting (Enable)

Hostname type: Use subnet setting (IP name)

DNS Hostname:

- Enable IP name IPv4 (A record) DNS requests
- Enable resource-based IPv4 (A record) DNS requests
- Enable resource-based IPv6 (AAAA record) DNS requests

Review and Launch

Step 4: Add Storage

Your instance will be launched with the following storage device settings. You can attach additional EBS volumes and instance store volumes to your instance, or edit the settings of the root volume. You can also attach additional EBS volumes after launching an instance, but not instance store volumes. [Learn more](#) about storage options in Amazon EC2.

Volume Type	Device	Snapshot	Size (GiB)	Volume Type	IOPS	Throughput (MB/s)	Delete on Termination	Encryption
Root	/dev/sda1	snap-043ff98c84a8774b1	8	General Purpose SSD (gp2)	100 / 3000	N/A	<input checked="" type="checkbox"/>	Not Encrypt

Add New Volume

Free tier eligible customers can get up to 30 GB of EBS General Purpose (SSD) or Magnetic storage. [Learn more](#) about free usage tier eligibility and usage restrictions.

Step 5: Add Tags

A tag consists of a case-sensitive key-value pair. For example, you could define a tag with key = Name and value = Webserver. A copy of a tag can be applied to volumes, instances or both. Tags will be applied to all instances and volumes. [Learn more](#) about tagging your Amazon EC2 resources.

Key	(128 characters maximum)	Value	(256 characters maximum)	Instances	Volumes	Network Interfaces
This resource currently has no tags						

Choose the Add tag button or [click to add a Name tag](#).
Make sure your [IAM policy](#) includes permissions to create tags.

Add Tag (Up to 50 tags maximum)

Review and Launch

Step 6: Configure Security Group

A security group is a set of firewall rules that control the traffic for your instance. On this page, you can add rules to allow specific traffic to reach your instance. For example, if you want to set up a web server and allow internet traffic to reach your instance, add rules that allow unrestricted access to the HTTP and HTTPS ports. You can create a new security group or select from an existing one below. [Learn more about Amazon EC2 security groups.](#)

Assign a security group: Create a new security group
 Select an existing security group

Security group name: launch-wizard-1

Description: launch-wizard-1 created 2022-09-25T20:52:34.417+05:30

Type	Protocol	Port Range	Source	Description
SSH	TCP	22	Custom 0.0.0.0/0	e.g. SSH for Admin Desktop

Add Rule

Warning
Rules with source of 0.0.0.0 allow all IP addresses to access your instance. We recommend setting security group rules to allow access from known IP addresses only.

Cancel Previous Review and Launch

Step 7: Review Instance Launch

Please review your instance launch details. You can always change them later.

AMI Details

Ubuntu Server 20.04 LTS (HVM)
Free tier eligible
Root Device Type: ebs

Instance Type

Instance Type	ECUs	vCPUs
t2.micro	-	1

Select an existing key pair or create a new key pair

A key pair consists of a **public key** that AWS stores, and a **private key file** that you store. Together, they allow you to connect to your instance securely. For Windows AMIs, the private key file is required to obtain the password used to log into your instance. For Linux AMIs, the private key file allows you to securely SSH into your instance. Amazon EC2 supports ED25519 and RSA key pair types.

Note: The selected key pair will be added to the set of keys authorized for this instance. Learn more about [removing existing key pairs from a public AMI](#).

Choose an existing key pair
Select a key pair
No key pairs found

No key pairs found
You don't have any key pairs. Please create a new key pair by selecting the [Create a new key pair](#) option above to continue.

Cancel Launch Instances

The screenshot shows the AWS EC2 Launch Instance Wizard. In the 'AMI Details' section, 'Ubuntu Server 20.04 LTS (HVM)' is selected. Under 'Instance Type', 't2.micro' is chosen. In the 'Key pair' section, 'Terraform' is entered into the 'Key pair name' field. A note states: 'You have to download the private key file (*.pem file) before you can continue. Store it in a secure and accessible location. You will not be able to download the file again after it's created.' Red arrows point from the 'Key pair name' input field, the note, and the 'Launch Instances' button to the right.

AMI Details

- Ubuntu Server 20.04 LTS (HVM)

Instance Type

- t2.micro

Key pair

- Key pair type: RSA
- Key pair name: Terraform

You have to download the private key file (*.pem file) before you can continue. Store it in a secure and accessible location. You will not be able to download the file again after it's created.

Launch Status

Your instances are now launching. The following instance launches have been initiated: i-0ed4fa9e3fc936ae. View launch log.

Get notified of estimated charges

How to connect to your instances

Here are some helpful resources to get you started

- How to connect to your Linux instance
- Amazon EC2: User Guide

The screenshot shows the AWS EC2 Management Console interface. In the top navigation bar, the URL is `us-west-2.console.aws.amazon.com/ec2/home?region=us-west-2#Instancessearch=i-0ed4fa9e3ffc936ae`. The main content area displays a table of instances. One instance is selected, and its details are shown in a modal window. The instance ID is `i-0ed4fa9e3ffc936ae`, and it has been renamed to `hostPC-terraform`. The modal includes a `Save` button, which is highlighted with a red arrow. Below the modal, the `Details` tab is selected, showing the instance summary. The `Connect` button in the top right of the modal is also highlighted with a red circle.

EC2 Management Console - us-west-2.console.aws.amazon.com/ec2/home?region=us-west-2#Instancessearch=i-0ed4fa9e3ffc936ae

Instances (1/1) Info

Name: hostPC-terraform

Instance state: Running

Instance type: t2.micro

Status check: Initializing

Alarm status: No alarms

Availability Zone: us-west-2b

Public IPv4 address: 35.87.76.55 | open address

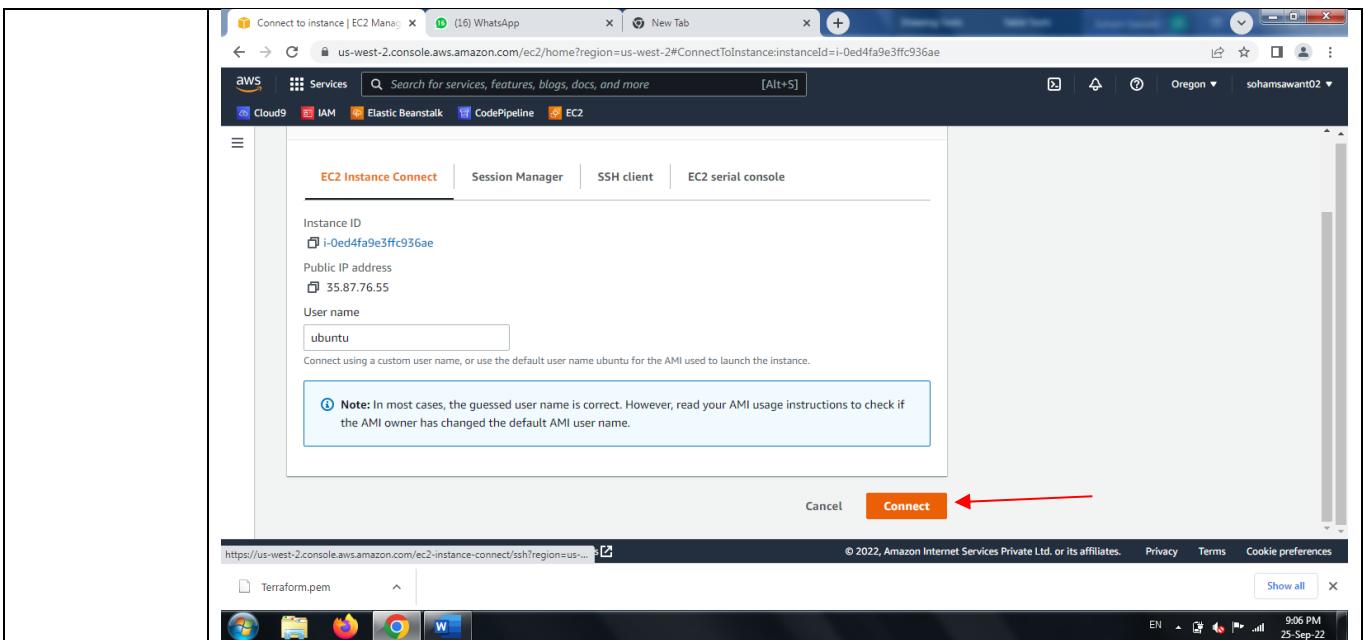
Private IPv4 addresses: 172.31.26.243

Public IPv4 DNS: ec2-35-87-76-55.us-west-

Details | Security | Networking | Storage | Status checks | Monitoring | Tags

Feedback Looking for language selection? Find it in the new Unified Settings

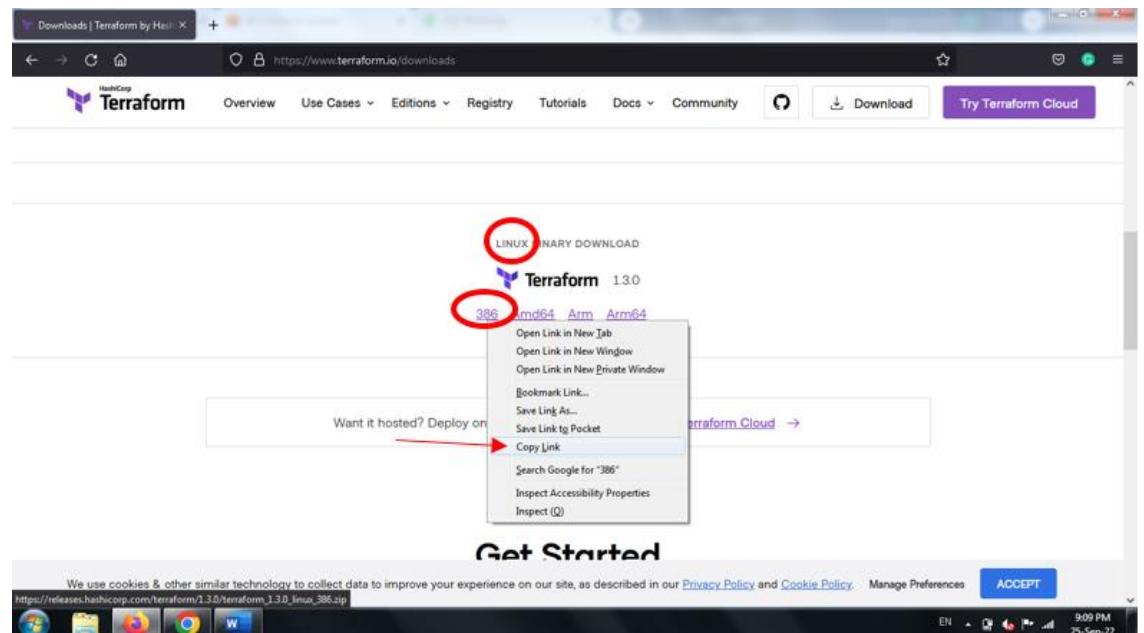
EN 9:06 PM 25-Sep-22



Terraform Installation Steps on Ubuntu20

Step: 1 (Ensure root user privileges)Terraform uses HashiCorp Configuration Language (HCL) to manage environments of Operators and Infrastructure teams. To **download** go to site <https://www.terraform.io/downloads.html>

Select the appropriate package for your operating system and architecture. Copy link address



Use wget command to download from the link

```
wget
```

```
https://releases.hashicorp.com/terraform/1.2.9/terraform\_1.2.9\_linux\_386.zip
```

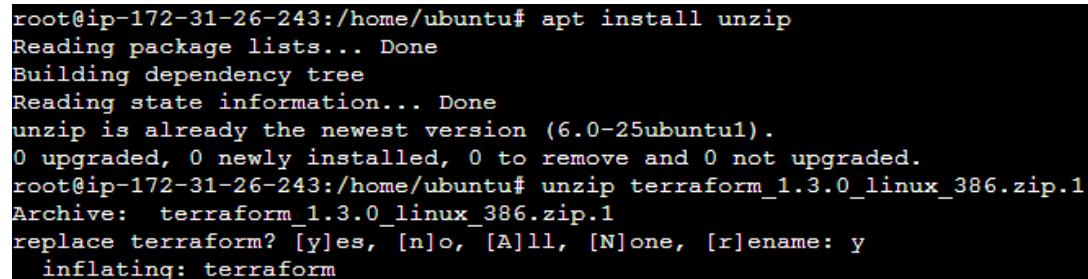


```
aws Services Search for services, features, blogs, docs, and more [Alt+S] Cloud9 IAM Elastic Beanstalk CodePipeline EC2
ubuntu@ip-172-31-26-243:~$ sudo su
root@ip-172-31-26-243:~/home/ubuntu$ wget https://releases.hashicorp.com/terraform/1.3.0/terraform_1.3.0_linux_386.zip
--2022-09-25 15:44:30-- https://releases.hashicorp.com/terraform/1.3.0/terraform_1.3.0_linux_386.zip
Resolving releases.hashicorp.com (releases.hashicorp.com)... 54.192.76.88, 54.192.76.105, 54.192.76.14, ...
Connecting to releases.hashicorp.com (releases.hashicorp.com)|54.192.76.88|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 18285619 (17M) [application/zip]
Saving to: 'terraform_1.3.0_linux_386.zip.1'

terraform_1.3.0_linux_386.zip.1      100%[=====] 17.44M --.-KB/s   in 0.1s
2022-09-25 15:44:31 (148 MB/s) - 'terraform_1.3.0_linux_386.zip.1' saved [18285619/18285619]
```

Step:2 unzip the archive by using below command

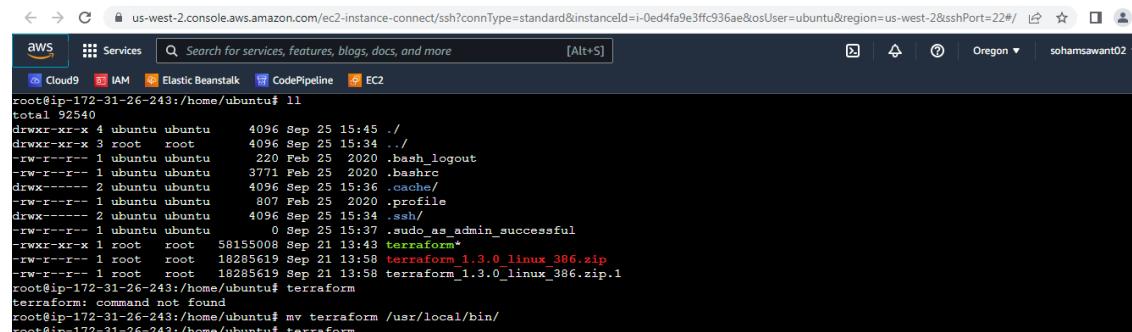
```
unzip terraform_1.2.9_linux_386.zip
```



```
root@ip-172-31-26-243:/home/ubuntu$ apt install unzip
Reading package lists... Done
Building dependency tree
Reading state information... Done
unzip is already the newest version (6.0-25ubuntu1).
0 upgraded, 0 newly installed, 0 to remove and 0 not upgraded.
root@ip-172-31-26-243:/home/ubuntu$ unzip terraform_1.3.0_linux_386.zip.1
Archive:  terraform_1.3.0_linux_386.zip.1
replace terraform? [y]es, [n]o, [A]ll, [N]one, [r]ename: y
    inflating: terraform
```

The archive will extract a single binary called **terraform**
and Move the terraform binary to a directory included in your system's PATH in
my case `usr/local/bin/`

```
mv terraform /usr/local/bin/
```



```
aws Services Search for services, features, blogs, docs, and more [Alt+S] Cloud9 IAM Elastic Beanstalk CodePipeline EC2
root@ip-172-31-26-243:~/home/ubuntu$ ll
total 92540
drwxr-xr-x  4 ubuntu  ubuntu       4096 Sep 25 15:45 .
drwxr-xr-x  3 root   root        4096 Sep 25 15:34 ..
-rw-r--r--  1 ubuntu  ubuntu     220 Feb 25 2020 .bash_logout
-rw-r--r--  1 ubuntu  ubuntu    3771 Feb 25 2020 .bashrc
drwx----- 2 ubuntu  ubuntu     4096 Sep 25 15:36 .cache/
-rw-r--r--  1 ubuntu  ubuntu     807 Feb 25 2020 .profile
drwx----- 2 ubuntu  ubuntu     4096 Sep 25 15:34 .ssh/
-rw-r--r--  1 ubuntu  ubuntu      0 Sep 25 15:37 .sudo_as_admin_successful
-rw-r--r--  1 root   root    58155008 Sep 21 13:43 terraform*
-rw-r--r--  1 root   root    18285619 Sep 21 13:58 terraform_1.3.0_linux_386.zip
-rw-r--r--  1 root   root    18285619 Sep 21 13:58 terraform_1.3.0_linux_386.zip.1
root@ip-172-31-26-243:~/home/ubuntu$ terraform
terraform: command not found
root@ip-172-31-26-243:~/home/ubuntu$ mv terraform /usr/local/bin/
root@ip-172-31-26-243:~/home/ubuntu$ terraform
```

Step 4: To check whether Terraform is installed, run:

```
terraform -v
```

or

type

`terraform`

to see options list for terraform

```
root@ip-172-31-26-243:/home/ubuntu# terraform -v
Terraform v1.3.0
on linux_386
root@ip-172-31-26-243:/home/ubuntu#
```

i-0ed4fa9e3ffc936ae (hostPC-terraform)

```
root@ip-172-31-26-243:/home/ubuntu# terraform
Usage: terraform [global options] <subcommand> [args]

The available commands for execution are listed below.
The primary workflow commands are given first, followed by
less common or more advanced commands.

Main commands:
  init          Prepare your working directory for other commands
  validate      Check whether the configuration is valid
  plan          Show changes required by the current configuration
  apply         Create or update infrastructure
  destroy       Destroy previously-created infrastructure

All other commands:
  console        Try Terraform expressions at an interactive command prompt
  fmt            Reformat your configuration in the standard style
  force-unlock  Release a stuck lock on the current workspace
  get            Install or upgrade remote Terraform modules
  graph          Generate a Graphviz graph of the steps in an operation
  import         Associate existing infrastructure with a Terraform resource
  login          Obtain and save credentials for a remote host
  logout         Remove locally-stored credentials for a remote host
  output         Show output values from your root module
  providers     Show the providers required for this configuration

  refresh        Update the state to match remote systems
  show           Show the current state or a saved plan
  state          Advanced state management
  taint          Mark a resource instance as not fully functional
  test           Experimental support for module integration testing
  untaint       Remove the 'tainted' state from a resource instance
  version        Show the current Terraform version
  workspace     Workspace management

Global options (use these before the subcommand, if any):
  -chdir=DIR    Switch to a different working directory before executing the
                given subcommand.
  -help          Show this help output, or the help for a specified subcommand.
  -version       An alias for the "version" subcommand.
```

Conclusion Successfully understood terraform and it's installation.

Semester	T.E. Semester V – Information Technology
Subject	Advance DevOps Lab
Subject Professor In-charge	Prof. Indu Anoop
Laboratory	

Student Name	Soham Anant Sawant	
Roll Number	21101B2002	
Grade and Subject Teacher's Signature		

Experiment	6	
Problem Statement	To build, change, destroy AWS/GCP/ Microsoft Azure/DigitalOcean infrastructure using terraform	
Resources / Apparatus Required	Hardware: Computer System	Software: Web Browser
Details	<p>Terraform</p> <p>Terraform is an infrastructure as code (IaC) tool that allows you to build, change, and version infrastructure safely and efficiently. This includes low-level components such as compute instances, storage, and networking, as well as high-level components such as DNS entries, SaaS features, etc. Terraform can manage both existing service providers and custom in-house solutions.</p> <p>Key Features</p> <p>Infrastructure as Code:</p> <p>You describe your infrastructure using Terraform's high-level configuration language in human-readable, declarative configuration files. This allows you to create a blueprint that you can version, share, and reuse.</p> <p>Resource Graph</p> <p>Terraform builds a resource graph and creates or modifies non-dependent resources in parallel. This allows Terraform to build resources as efficiently as possible and gives you greater insight into your infrastructure.</p>	

Change Automation

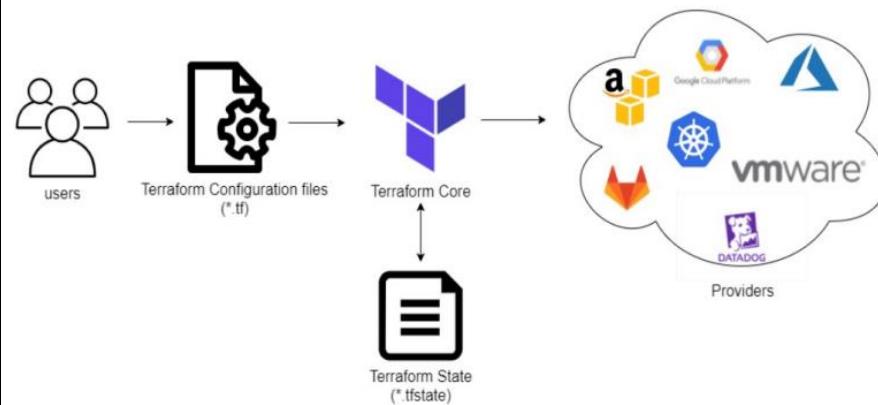
Terraform can apply complex changesets to your infrastructure with minimal human interaction. When you update configuration files, Terraform determines what changed and creates incremental execution plans that respect dependencies.

How does Terraform work?

Terraform works with two major components:

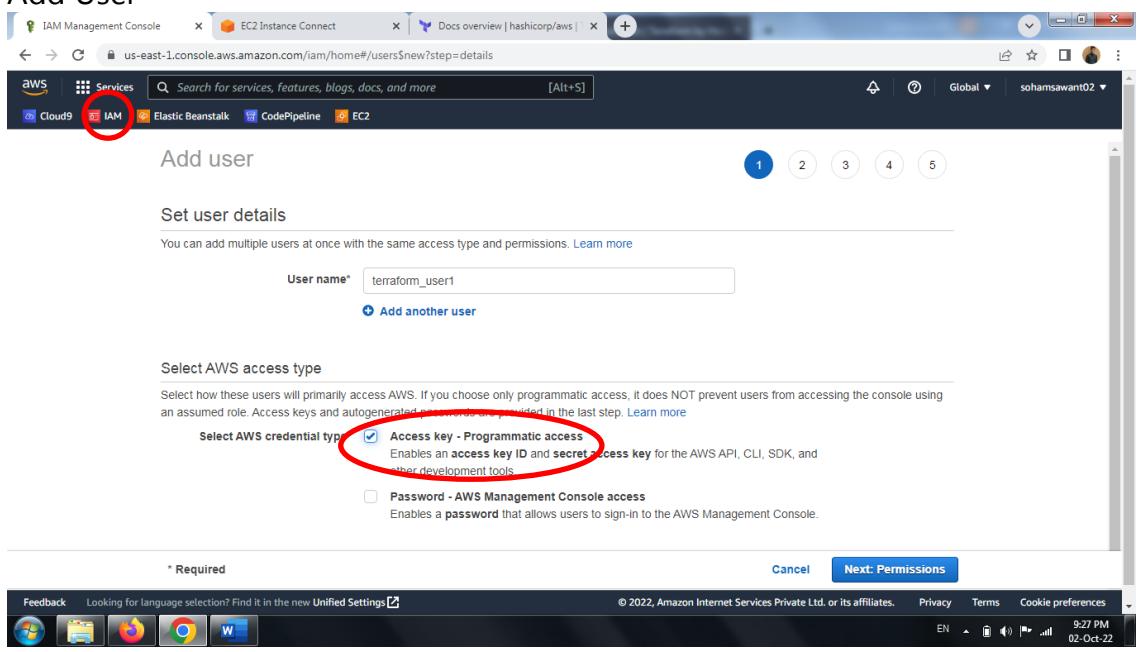
one is the **terraform core**: it takes the terraform configuration which is being provided by the user and then takes the terraform state which is managed by terraform itself. As such, this gets fed into the core that is responsible for figuring out what is that graph of our different resources for example how these different pieces relate to each other or what needs to be created/updated/destroyed, it does all the essential lifecycle management.

On the backside, terraform supports many different **providers**, such as: cloud providers (AWS,GCP,AZURE) and they also could be on-premise infrastructure (VMware, OpenStack.) But this support is not restricted or limited only to Infrastructure As A Service , terraform can also manage higher level like Platform As A Service(Kubernetes, Lambdas..)or even Software As A Service (DataDog, GitHub..)



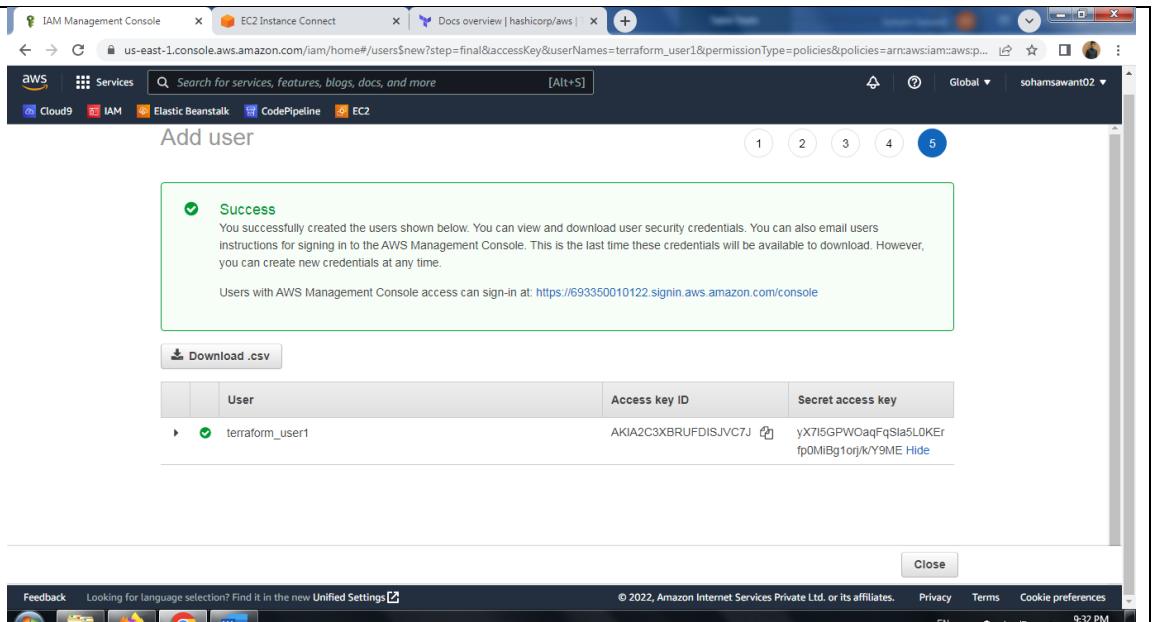
All of these are important pieces of the infrastructure, they are all part of the logical end-to-end delivery.

Terraform has over a hundred providers for different technologies, and each provider gives terraform users access to their resources. It also gives you the ability to create infrastructure at different levels.

	<p>Terraform Workflow:</p> <p>These are the list of steps we are going to perform</p> <ol style="list-style-type: none"> 1. Create a file and save it as main.tf 2. Execute the command terraform init to initialize 3. Execute the command terraform plan to check what change would be made. (Should always do it) 4. If you are happy with the changes it is claiming to make, then execute terraform apply to commit and start the build
Code	<p>Pre-requisite: An AMI of ubuntu 20 system with terraform installed.</p> <p>Add User</p>  <p>The screenshot shows the AWS IAM Management Console with the 'Add user' wizard open. The top navigation bar has tabs for IAM, Services, EC2 Instance Connect, and Docs overview. The 'IAM' tab is highlighted with a red circle. The main page title is 'Add user'. Under 'Set user details', there's a note about adding multiple users. The 'User name' field contains 'terraform_user1'. Below it is a link to 'Add another user'. Under 'Select AWS access type', there are two options: 'Access key - Programmatic access' (checked) and 'Password - AWS Management Console access'. A red circle highlights the checked checkbox for programmatic access. At the bottom, there are 'Required' fields, a 'Cancel' button, and a 'Next: Permissions' button, which is also highlighted with a red circle.</p>

The screenshot illustrates the process of creating a new AWS user. It consists of three main sections:

- Step 1: Set permissions**
 - The 'Attach existing policies directly' button is highlighted with a red circle.
 - A red arrow points to the 'AdministratorAccess' policy in the list, indicating it is selected.
- Step 2: Review**
 - User details:
 - User name: terraform_user1
 - AWS access type: Programmatic access - with an access key
 - Permissions boundary: Permissions boundary is not set
 - Permissions summary:
 - The following policies will be attached to the user shown above.
 - Type: Managed policy
 - Name: AdministratorAccess
- Step 3: Create user**
 - Final step before creation.



Steps to build, change, destroy AWS infrastructure using Terraform

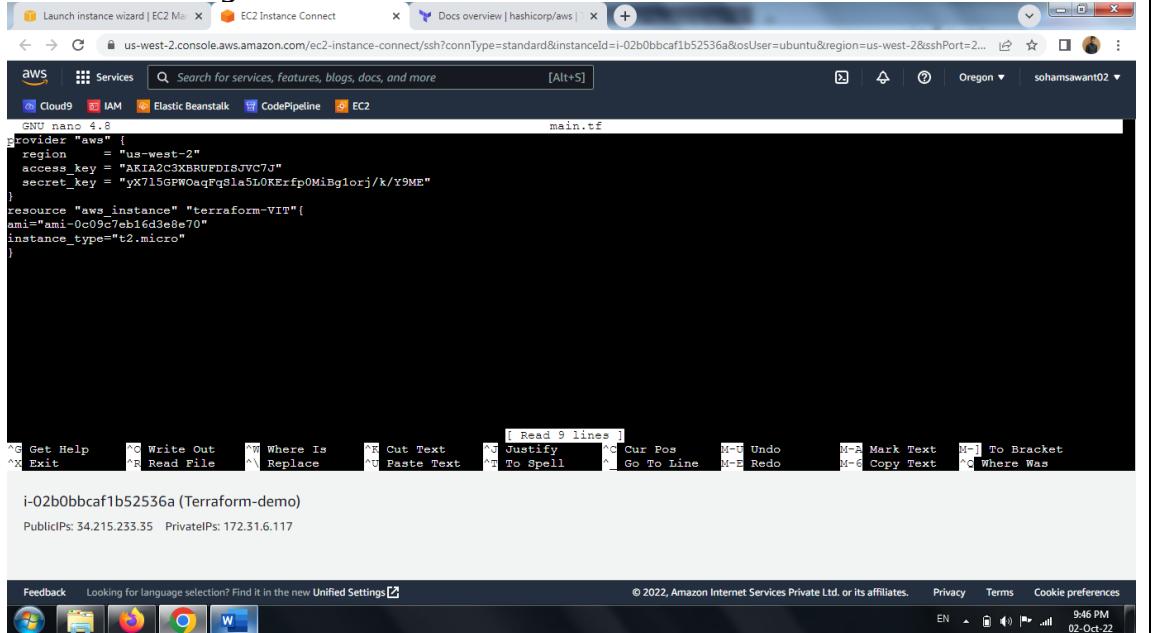
Step: 1 : To BUILD an AWS infrastructure

1.1 Write your **main.tf** file

Use command to create a file and edit it

```
touch main.tf
nano main.tf
```

Edit to following contents and Press Ctrl+S then Ctrl+X



Replace the access key and secret key values of the new IAM user which needs to be created in the region mentioned. Also replace the ami value to the virtual system's ami value [From launch instances portal]

1.2 Initialize the terraform

Write the command

```
terraform init
```

A screenshot of a terminal window titled "EC2 Instance Connect" on a web browser. The URL is "us-west-2.console.aws.amazon.com/ec2-instance-connect/ssh?connType=standard&instanceId=i-02b0bbcaf1b52536a&osUser=ubuntu®ion=us-west-2&sshPort=2...". The terminal session shows the user running the "terraform init" command. The output indicates that the provider plugins are being initialized, including Hashicorp/aws v4.33.0, and a lock file ".terraform.lock.hcl" is created. A message at the end states "Terraform has been successfully initialized!".

```
root@ip-172-31-6-117:/home/ubuntu# nano main.tf
root@ip-172-31-6-117:/home/ubuntu# terraform init

Initializing the backend...

Initializing provider plugins...
- Finding latest version of hashicorp/aws...
- Installing hashicorp/aws v4.33.0...
- Installed hashicorp/aws v4.33.0 (signed by HashiCorp)

Terraform has created a lock file .terraform.lock.hcl to record the provider selections it made above. Include this file in your version control repository so that Terraform can guarantee to make the same selections by default when you run "terraform init" in the future.

Terraform has been successfully initialized!

You may now begin working with Terraform. Try running "terraform plan" to see any changes that are required for your infrastructure. All Terraform commands should now work.

If you ever set or change modules or backend configuration for Terraform, rerun this command to reinitialize your working directory. If you forget, other commands will detect it and remind you to do so if necessary.

root@ip-172-31-6-117:/home/ubuntu#
```

i-02b0bbcaf1b52536a (Terraform-demo)
PublicIPs: 34.215.233.35 PrivateIPs: 172.31.6.117

Feedback Looking for language selection? Find it in the new Unified Settings © 2022, Amazon Internet Services Private Ltd. or its affiliates. Privacy Terms Cookie preferences EN 9:38 PM 02-Oct-22

1.3 Execute plan phase to understand what changes to be done.

```
terraform plan -lock=false
```

A screenshot of a terminal window titled "EC2 Instance Connect" on a web browser. The URL is "us-west-2.console.aws.amazon.com/ec2-instance-connect/ssh?connType=standard&instanceId=i-02b0bbcaf1b52536a&osUser=ubuntu®ion=us-west-2&sshPort=2...". The terminal session shows the user running the "terraform plan -lock=false" command. The output shows the execution plan, indicating actions such as creating an AWS instance. The plan is generated using the selected providers.

```
commands will detect it and remind you to do so if necessary.
root@ip-172-31-6-117:/home/ubuntu# terraform plan -lock=false

Terraform used the selected providers to generate the following execution plan. Resource actions are indicated with the following symbols:
+ create

Terraform will perform the following actions:

# aws_instance.terraform-VIT will be created
+ resource "aws_instance" "terraform-VIT" {
    ami                               = "ami-0c09c7eb16d3e870"
    arn                             = (known after apply)
    associate_public_ip_address      = (known after apply)
    availability_zone                = (known after apply)
    cpu_core_count                  = (known after apply)
    cpu_threads_per_core            = (known after apply)
    disable_api_stop                = (known after apply)
    disable_api_termination          = (known after apply)
    ebs_optimized                   = (known after apply)
    get_password_data               = false
    host_id                         = (known after apply)
    host_resource_group_arn          = (known after apply)
    id                             = (known after apply)
    instance_initiated_shutdown_behavior = (known after apply)
    instance_state                  = (known after apply)
```

i-02b0bbcaf1b52536a (Terraform-demo)
PublicIPs: 34.215.233.35 PrivateIPs: 172.31.6.117

Feedback Looking for language selection? Find it in the new Unified Settings © 2022, Amazon Internet Services Private Ltd. or its affiliates. Privacy Terms Cookie preferences EN 9:39 PM 02-Oct-22

```

aws Cloud9 IAM Elastic Beanstalk CodePipeline EC2
+ enable_resource_name_dns_aaa_record = (known after apply)
+ hostname_type                   = (known after apply)
}

+ root_block_device {
  + delete_on_termination = (known after apply)
  + device_name          = (known after apply)
  + encrypted            = (known after apply)
  + iops                 = (known after apply)
  + kms_key_id           = (known after apply)
  + tags                 = (known after apply)
  + throughput            = (known after apply)
  + volume_id             = (known after apply)
  + volume_size           = (known after apply)
  + volume_type            = (known after apply)
}
}

Plan: 1 to add, 0 to change, 0 to destroy.

Note: You didn't use the -out option to save this plan, so Terraform can't guarantee to take exactly these actions if you run "terraform apply" now.
root@ip-172-31-6-117:/home/ubuntu#
```

i-02b0bbcfa1b52536a (Terraform-demo)
PublicIPs: 34.215.233.35 PrivateIPs: 172.31.6.117

1.4 Apply the actions which were planned in apply phase

terraform apply -lock=false

```

aws Cloud9 IAM Elastic Beanstalk CodePipeline EC2
root@ip-172-31-6-117:/home/ubuntu# terraform apply -lock=false
Terraform used the selected providers to generate the following execution plan. Resource actions are indicated with the following symbols:
  * create

Terraform will perform the following actions:

# aws_instance.terraform-VIT will be created
* resource "aws_instance" "terraform-VIT" {
  + ami                                = "ami-0c09c7eb16d3e8e70"
  + arn                                = (known after apply)
  + associate_public_ip_address         = (known after apply)
  + availability_zone                  = (known after apply)
  + cpu_core_count                     = (known after apply)
  + cpu_threads_per_core              = (known after apply)
  + disable_api_stop                  = (known after apply)
  + disable_api_termination           = (known after apply)
  + ebs_optimized                      = (known after apply)
  + get_password_data                = false
  + host_id                            = (known after apply)
  + host_resource_group_arn           = (known after apply)
  + id                                 = (known after apply)
  + instance_initiated_shutdown_behavior = (known after apply)
  + instance_state                    = (known after apply)
  + instance_type                     = "t2.micro"
}

i-02b0bbcfa1b52536a (Terraform-demo)
PublicIPs: 34.215.233.35 PrivateIPs: 172.31.6.117
```

```

+ kms_key_id          = (known after apply)
+ tags                = (known after apply)
+ throughput          = (known after apply)
+ volume_id           = (known after apply)
+ volume_size         = (known after apply)
+ volume_type         = (known after apply)

}

Plan: 1 to add, 0 to change, 0 to destroy.

Do you want to perform these actions?
Terraform will perform the actions described above.
Only 'yes' will be accepted to approve.

Enter a value: yes

aws instance.terraform-VIT: Creating...
aws instance.terraform-VIT: Still creating... [10s elapsed]
aws instance.terraform-VIT: Still creating... [20s elapsed]
aws instance.terraform-VIT: Still creating... [30s elapsed]
aws instance.terraform-VIT: Creation complete after 32s [id=i-0dea07b69f464c00d]

Apply complete! Resources: 1 added, 0 changed, 0 destroyed.
root@ip-172-31-6-117:/home/ubuntu#

```

i-02b0bbcaf1b52536a (Terraform-demo)
PublicIPs: 34.215.233.35 PrivateIPs: 172.31.6.117

Type 'yes' to confirm to apply.

Step 2: Confirm the infrastructure created

Go to EC2 console to check if a new instance is created as per the code written in main.tf file.

Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone
Terraform-demo	i-02b0bbcaf1b52536a	Running	t2.micro	2/2 checks passed	No alarms	us-west-2c
-	i-0dea07b69f464c00d	Running	t2.micro	Initializing	No alarms	us-west-2c

Step 3: CHANGE the infrastructure created using terraform

Modify main.tf to include instance name.

Instances | EC2 Management Console **EC2 Instance Connect** **Docs overview | hashicorp/aws** **[+]**

Services **Search for services, features, blogs, docs, and more** **[Alt+S]**

Cloud9 **IAM** **Elastic Beanstalk** **CodePipeline** **EC2**

```
GNU nano 4.8                               main.tf
provider "aws" {
  region      = "us-west-2"
  access_key  = "AKIA2C3XBRUFDI8JVC7J"
  secret_key  = "yX7l5GPWOaqFqSla5LOErfpOMiBglorj/k/y9ME"
}
resource "aws_instance" "terraform-VIT"
ami="ami-0c09c7eb16d3e8e70"
instance_type="t2.micro"
tags={
  Name="Terraform-DemoInfra"
}
```

Get Help **Write Out** **Where Is** **Cut Text** **Justify** **Cur Pos** **Undo** **Mark Text** **To Bracket**
Exit **Read File** **Replace** **Paste Text** **To Spell** **Go To Line** **Redo** **Copy Text** **Where Was**

i-02b0bbcfa1b52536a (Terraform-demo)
Public IPs: 34.215.233.35 Private IPs: 172.31.6.117

Feedback Looking for language selection? Find it in the new [Unified Settings](#) **© 2022, Amazon Internet Services Private Ltd. or its affiliates.** **Privacy** **Terms** **Cookie preferences**

EN **9:56 PM** **02-Oct-22**

Repeat steps from 1.2. terraform init, plan and apply

Instances | EC2 Management Console **EC2 Instance Connect** **Docs overview | hashicorp/aws** **[+]**

Services **Search for services, features, blogs, docs, and more** **[Alt+S]**

Cloud9 **IAM** **Elastic Beanstalk** **CodePipeline** **EC2**

Terraform used the selected providers to generate the following execution plan. Resource actions are indicated with the following symbols:
~ update in-place

Terraform will perform the following actions:

```
# aws_instance.terraform-VIT will be updated in-place
~ resource "aws_instance" "terraform-VIT" {
  id                      = "i-0dea07b69f464c00d"
  ~ tags                  = [
    + "Name" = "Terraform-DemoInfra"
  ]
  ~ tags_all              = [
    + "Name" = "Terraform-DemoInfra"
  ]
  # (29 unchanged attributes hidden)
  # (7 unchanged blocks hidden)
}

Plan: 0 to add, 1 to change, 0 to destroy.
```

Note: You didn't use the -out option to save this plan, so Terraform can't guarantee to take exactly these actions if you run "terraform apply" now.

i-02b0bbcfa1b52536a (Terraform-demo)
Public IPs: 34.215.233.35 Private IPs: 172.31.6.117

Feedback Looking for language selection? Find it in the new [Unified Settings](#) **© 2022, Amazon Internet Services Private Ltd. or its affiliates.** **Privacy** **Terms** **Cookie preferences**

EN **9:58 PM** **02-Oct-22**

Instances | EC2 Management Console **EC2 Instance Connect** **Docs overview | hashicorp/aws**

Search for services, features, blogs, docs, and more [Alt+S]

Cloud9 IAM Elastic Beanstalk CodePipeline EC2

```

id = "i-0dea07b69f464c00d"
tags = [
  {
    "Name" = "Terraform-DemoInfra"
  }
]
tags_all = [
  {
    "Name" = "Terraform-DemoInfra"
  }
]
# (29 unchanged attributes hidden)
# (7 unchanged blocks hidden)
]

Plan: 0 to add, 1 to change, 0 to destroy.

Do you want to perform these actions?
Terraform will perform the actions described above.
Only 'yes' will be accepted to approve.

Enter a value: yes

aws instance.terraform-VIT: Modifying... [id=i-0dea07b69f464c00d]
aws instance.terraform-VIT: Modifications complete after 1s [id=i-0dea07b69f464c00d]

Apply complete! Resources: 0 added, 1 changed, 0 destroyed.
root@ip-172-31-6-117:/home/ubuntu#

```

i-02b0bbcaf1b52536a (Terraform-demo)
Public IPs: 34.215.233.35 Private IPs: 172.31.6.117

Feedback Looking for language selection? Find it in the new Unified Settings © 2022, Amazon Internet Services Private Ltd. or its affiliates. Privacy Terms Cookie preferences

EN 9:58 PM 02-Oct-22

Instances | EC2 Management Console **EC2 Instance Connect** **Docs overview | hashicorp/aws**

Search for services, features, blogs, docs, and more [Alt+S]

Cloud9 IAM Elastic Beanstalk CodePipeline EC2

New EC2 Experience Tell us what you think

Instances (1/2) Info

Find instance by attribute or tag (case-sensitive)

Instance state = running

Clear filters

Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability zone
Terraform-demo	i-02b0bbcaf1b52536a	Running	t2.micro	2/2 checks passed	No alarms	us-west-2
Terraform-DemoInfra	i-0dea07b69f464c00d	Running	t2.micro	2/2 checks passed	No alarms	us-west-2

Launch instances

Instance: i-0dea07b69f464c00d (Terraform-DemoInfra)

Details Security Networking Storage Status checks Monitoring Tags

Instance summary

Instance ID i-0dea07b69f464c00d (Terraform-DemoInfra)	Public IPv4 address 34.216.4.211 open address	Private IPv4 addresses 172.31.7.127
IPv6 address -	Instance state Running	Public IPv4 DNS ec2-34-216-4-211.us-west-2.compute.amazonaws.com open address

Waiting for us-west-2.console.aws.amazon.com...

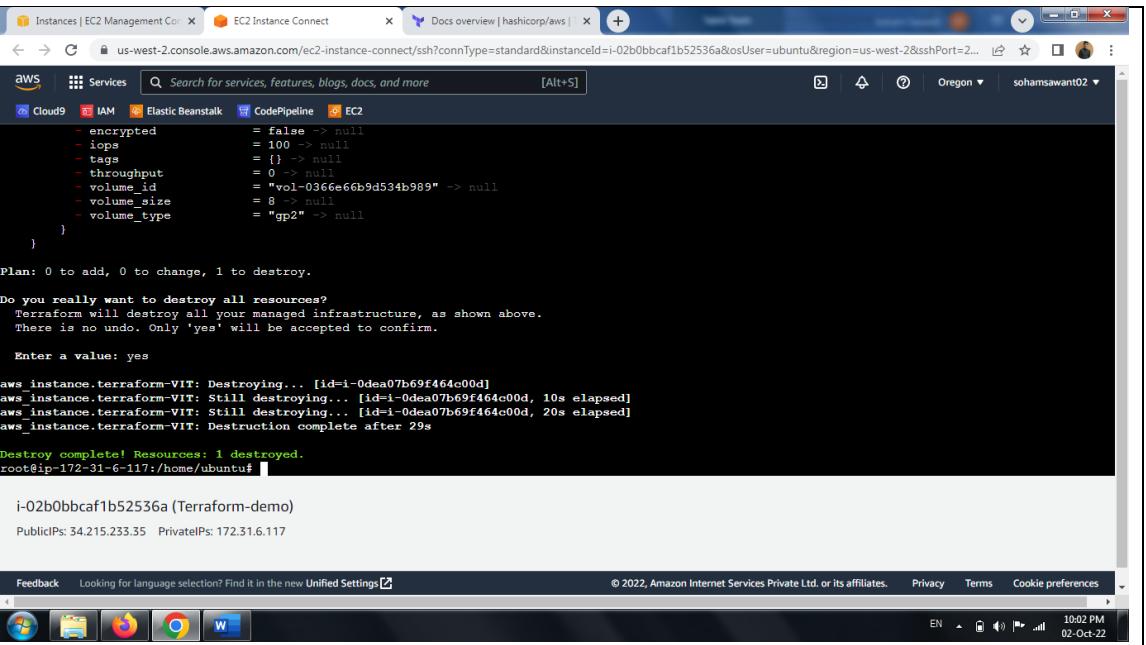
© 2022, Amazon Internet Services Private Ltd. or its affiliates. Privacy Terms Cookie preferences

EN 9:59 PM 02-Oct-22

Resource successfully changed to include instance name.

Step 4: DESTROY the built infrastructure

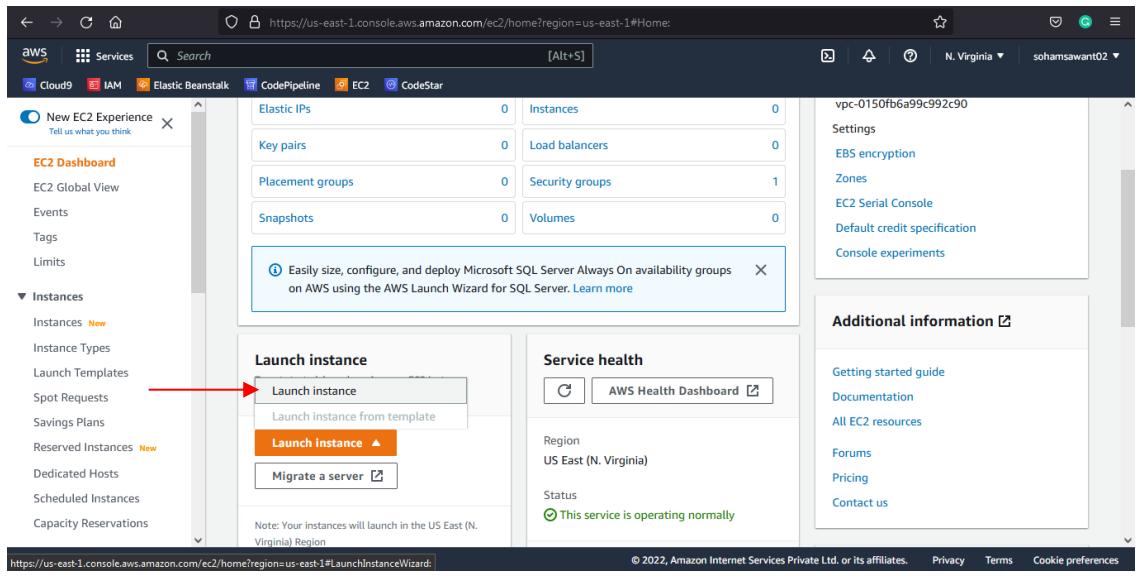
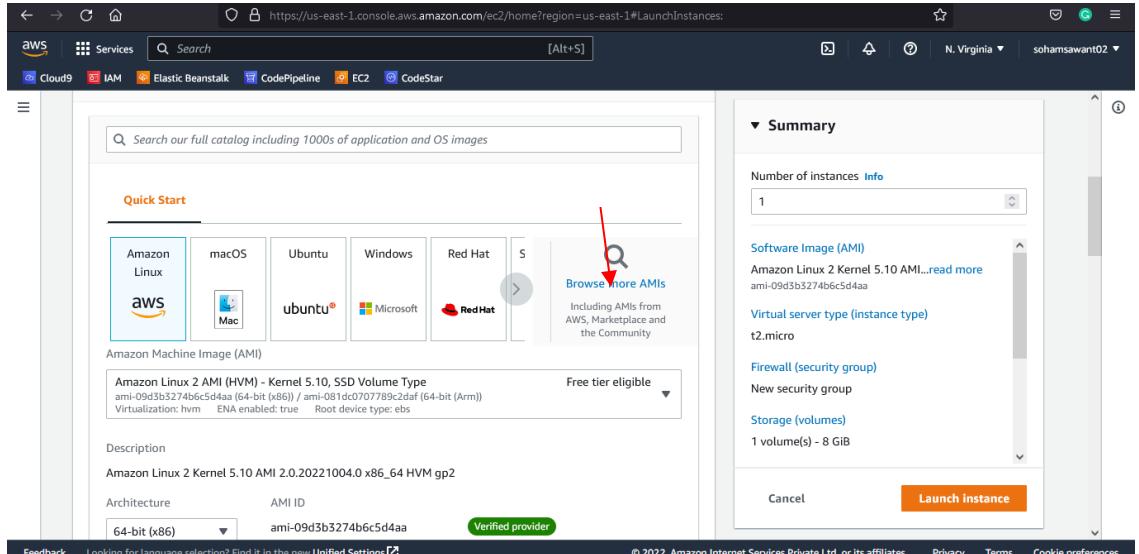
terraform destroy

	 <pre> resource "aws_instance" "Terraform-demo" { ami = "ami-0366e66b9d534b989" instance_type = "t2.micro" key_name = "ubuntu" vpc_security_group_ids = ["sg-0a1f1f1f1f1f1f1f1"] } output "instance_id" { value = aws_instance.Terraform-demo.id } </pre> <p>Do you really want to destroy all resources? Terraform will destroy all your managed infrastructure, as shown above. There is no undo. Only 'yes' will be accepted to confirm.</p> <p>Enter a value: yes</p> <pre> aws instance.terraform=VIT: Destroying... [id=i-0dea07b69f464c00d] aws instance.terraform=VIT: Still destroying... [id=i-0dea07b69f464c00d, 10s elapsed] aws instance.terraform=VIT: Still destroying... [id=i-0dea07b69f464c00d, 20s elapsed] aws instance.terraform=VIT: Destruction complete after 29s </pre> <p>Destroy complete! Resources: 1 destroyed.</p>
Conclusion	Successfully implemented through code how to build, change, destroy AWS infrastructure using terraform

Semester	T.E. Semester V – Information Technology
Subject	Advance DevOps Lab
Subject Professor In-charge	Prof. Indu Anoop
Laboratory	

Student Name	Soham Anant Sawant	
Roll Number	21101B2002	
Grade and Subject Teacher's Signature		

Experiment	Self-Learning Assignment 2	
Problem Statement	Setting up Splunk on AWS	
Resources / Apparatus Required	Hardware: Computer System	Software: Web Browser
Details	<p>Splunk</p> <p>Full-stack, real-time, analytics-driven monitoring for AWS</p> <p>See all your insights in one place. Splunk Observability takes the complexity out of monitoring your AWS, hybrid cloud environment. Get complete, instant visibility with contextual insights across your infrastructure, applications and customer experience to anticipate problems before customers notice, and know where to look when a problem does occur.</p> <p>Features:</p> <ul style="list-style-type: none"> • Simplified monitoring for complex AWS environments • Visualize your entire hybrid technology landscape • Improve uptime, reliability and performance • Power ad hoc investigations for real-time answers • Keep control of your data with no vendor lock-in <p>Advantages</p> <ul style="list-style-type: none"> • Simple to implement and scale • Continually index all of your IT data in real time. • Automatically discover useful information embedded in your data, so you don't have to identify it yourself. 	

	<ul style="list-style-type: none"> Search your physical and virtual IT infrastructure for literally anything of interest and get results in seconds.
Code	<p>Go to EC2 and Launch the instance.</p>  <p>Click on browse more AMI.</p>  <p>Select Splunk Enterprise.</p>

Quickstart AMIs (0)
My AMIs (0)
AWS Marketplace AMIs (16)
Community AMIs (17)

Refine results

Splunk (16 results) showing 1 - 16

Sort By: Relevance

Categories

Infrastructure Software (16)
DevOps (5)
Business Applications (2)

Publisher

Splunk Enterprise

By Splunk | Ver 9.0.0

★★★★★ 14 AWS reviews | 329 external reviews

The Splunk Enterprise AMI accelerates the speed at which organizations deploy Splunk Enterprise in AWS. Splunk Enterprise is the leading platform for Operational Intelligence, delivering an easy, fast, and secure way to search, analyze and visualize the massive streams of machine data generated by...

Splunk SOAR

By Splunk | Ver 5.3.5

Select

Click on Continue.

Splunk Enterprise

Splunk | Ver 9.0.0

★★★★★ 14 AWS reviews | 329 external reviews

Bring Your Own License | Free Tier

Overview Product details Pricing Usage Support

The Splunk Enterprise AMI accelerates the speed at which organizations deploy Splunk Enterprise in AWS..

Typical total price \$0.085/Hr

Total pricing per instance for services hosted on c5.large in us-east-1.
See additional pricing information.

Latest version 9.0.0

Delivery methods Amazon Machine Image

Operating systems Amazon Linux 2

Video Product Video

Categories Security Migration

Continue

Select instance type t2.micro.

Instance type

t2.micro

Family: t2 1 vCPU 1 GiB Memory

Free tier eligible

Compare instance types

The AMI vendor recommends using a c5.large instance (or larger) for the best experience with this product.

Key pair (login)

You can use a key pair to securely connect to your instance. Ensure that you have access to the selected key pair before you launch the instance.

Key pair name - required

Select Create new key pair

Network settings

Edit

Summary

Number of instances Info

1

Software Image (AMI)

Splunk Enterprise

ami-07f60d8d1e8ae7835

Virtual server type (instance type)

t2.micro

Firewall (security group)

New security group

Storage (volumes)

1 volume(s) - 20 GiB

Cancel

Launch Instance

Storage 20 GB.

The screenshot shows the AWS EC2 'Launch Instances' page. In the 'Configure storage' section, a red arrow points to the 'Root volume (Not encrypted)' size input field, which is set to 20 GiB. The summary panel on the right indicates 1 instance, using the Splunk Enterprise AMI (ami-07f60d8d1e8ae7835) and t2.micro instance type.

Launch instance.

The screenshot shows the 'Create security group' step. A red arrow points to the 'Launch instance' button at the bottom right of the summary panel. The security group 'Splunk Enterprise-9.0.0-AutogenByAWSMP--1' is being created with several rules allowing traffic from anywhere on various ports (SSH, CUSTOMTCP, HTTPS).

Create new Key Pair.

The screenshot shows the 'Create new key pair' page. A red arrow points to the 'Create key pair' button at the bottom right. The key pair name is 'Splunk'. The key pair type is RSA, and the private key file format is .pem. The summary panel on the right shows the instance configuration: 1 instance, using the Splunk Enterprise AMI (ami-07f60d8d1e8ae7835) and t2.micro instance type.

Launch Instance

The screenshot shows the 'Launch Instance' wizard in the AWS Management Console. The configuration includes:

- Key pair (login)**: Splunk
- Network settings**: VPC: vpc-0150fb6a99c992c90, Subnet: No preference (Default subnet in any availability zone), Auto-assign public IP: Enabled
- Software Image (AMI)**: Splunk Enterprise ami-07f60d8d1e8ae7835
- Virtual server type (instance type)**: t2.micro
- Firewall (security group)**: New security group
- Storage (volumes)**: 1 volume(s) - 8 GiB

A red arrow points to the **Launch instance** button at the bottom right.

Successfully initiated the launch of instance.

The screenshot shows the EC2 Instances page. It displays a success message: "Successfully initiated launch of instance (i-0c5d95a3984c6c66e)". A red arrow points to this message. Below it, there's a "Launch log" section with two entries: "Initializing requests" and "Subscribing to Marketplace AMI", both marked as "Succeeded".

Go to running instance and click on Elastic IPs.

The screenshot shows the EC2 Instances page. On the left sidebar, under the **Network & Security** section, there is a link labeled **Elastic IPs**, which is highlighted with a red arrow.

Allocate Elastic IP Address.

The screenshot shows the AWS EC2 console with the 'Elastic IP addresses' page. On the left, there's a navigation sidebar with options like 'Dedicated Hosts', 'Scheduled Instances', 'Capacity Reservations', 'Images', 'AMIs', 'AMI Catalog', 'Elastic Block Store', 'Volumes', 'Snapshots', 'Lifecycle Manager', 'Network & Security', 'Security Groups', 'Elastic IPs' (which is selected), 'Placement Groups', 'Key Pairs', and 'Network Interfaces'. The main area displays a table of 'Elastic IP addresses' with columns for 'Name', 'Allocated IPv4 add...', 'Type', 'Allocation ID', and 'Reverse DNS'. At the top right of the table, there's a 'Actions' dropdown and a prominent orange 'Allocate Elastic IP address' button. A red arrow points to this button.

Allocate

This screenshot shows the 'Allocate' step in the AWS EC2 console. It includes sections for 'Global static IP addresses' (with a note about AWS Global Accelerator) and 'Tags - optional' (with a note about tags). There's a 'Create accelerator' button and a 'Cancel' button. At the bottom right, there's an orange 'Allocate' button, which has a red arrow pointing to it.

Associate this Elastic IP Address.

The screenshot shows the AWS EC2 console after successfully allocating an Elastic IP address. A green notification bar at the top says 'Elastic IP address allocated successfully. Elastic IP address 34.193.144.252'. To the right of this message is an orange 'Associate this Elastic IP address' button, which has a red arrow pointing to it. Below this, the 'Elastic IP addresses' table shows one entry: '34.193.144.252' with 'Public IP' type and 'eipalloc-0654448d624f89318' as the Allocation ID. The table has columns for 'Name', 'Allocated IPv4 add...', 'Type', 'Allocation ID', and 'Reverse DNS'. At the bottom, there's a 'Summary' section.

Associate Elastic IP Address.

The screenshot shows the AWS EC2 console under the 'Elastic IP addresses' section. A single elastic IP address, '34.193.144.252', is listed as 'Allocated' with a 'Public IP' type. In the 'Actions' dropdown menu, the 'Associate Elastic IP address' option is highlighted with a red arrow. Below the table, there's a summary section for the selected IP.

Select the instance and click on Associate.

The screenshot shows the 'Associate Elastic IP address' dialog. It asks for the 'Resource type' (set to 'Instance') and the 'Instance' (set to 'i-0c5d95a3984c6c66e'). It also includes a note about reassociation and a checkbox for 'Allow this Elastic IP address to be reassigned'. A red arrow points to the 'Associate' button at the bottom right.

Elastic IP Address associated successfully.

The screenshot shows the 'Elastic IP addresses' page again. A green success message at the top states: 'Elastic IP address 34.193.144.252 has been associated with instance i-0c5d95a3984c6c66e'. The table now shows the association details: the IP is listed as 'Allocated' with a 'Public IP' type, and its 'Allocation ID' is 'eipalloc-0654448d624f89318'. The 'Associate' button from the previous step is no longer visible.

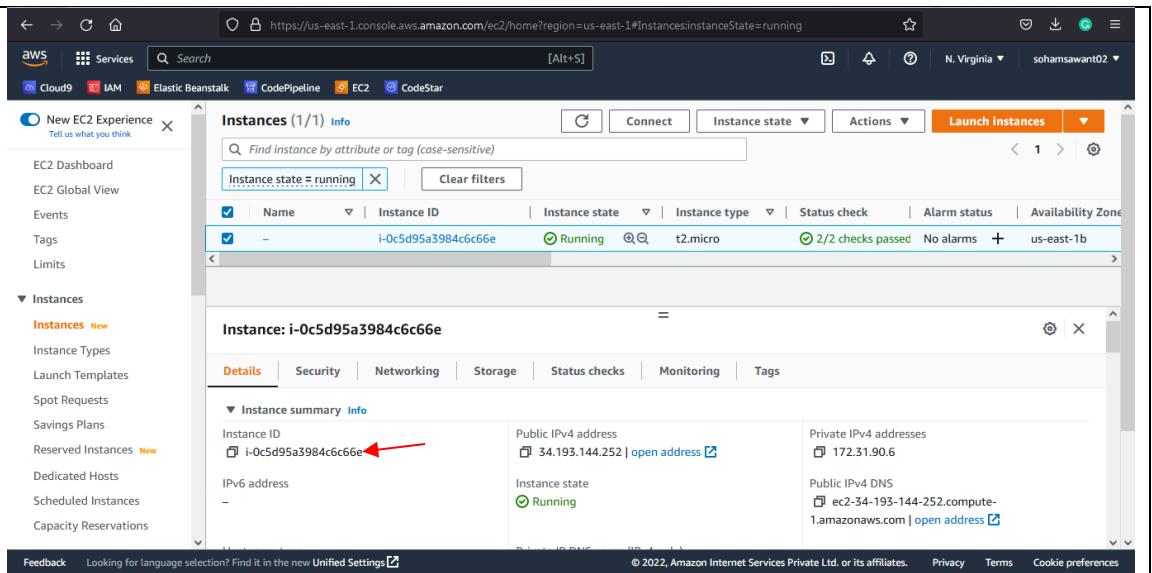
Copy Elastic IP Address of the instance.

The screenshot shows the AWS EC2 Management console. In the left sidebar, under 'Instances', the 'Instances' tab is selected. In the main content area, there is a table titled 'Instances (1/1)'. One row is shown with the instance ID i-0c5d95a3984c6c66e, which is running on a t2.micro instance type. On the right side of the instance details, there is a section titled 'Elastic IP addresses' with a single entry: '34.193.144.252 [Public IP]'. A red arrow points to this entry.

Go to ElasticIP:8000 in the browser.

The screenshot shows a web browser window with the URL 34.193.144.252:8000/en-US/account/login?return_to=%2Fen-US%2F. The page has a dark background with white text. At the top, it says 'splunk>enterprise'. Below that is a sign-in form with fields for 'Username' and 'Password', and a 'Sign In' button. There is also a link for 'First time signing in?'.

Copy the Instance ID.

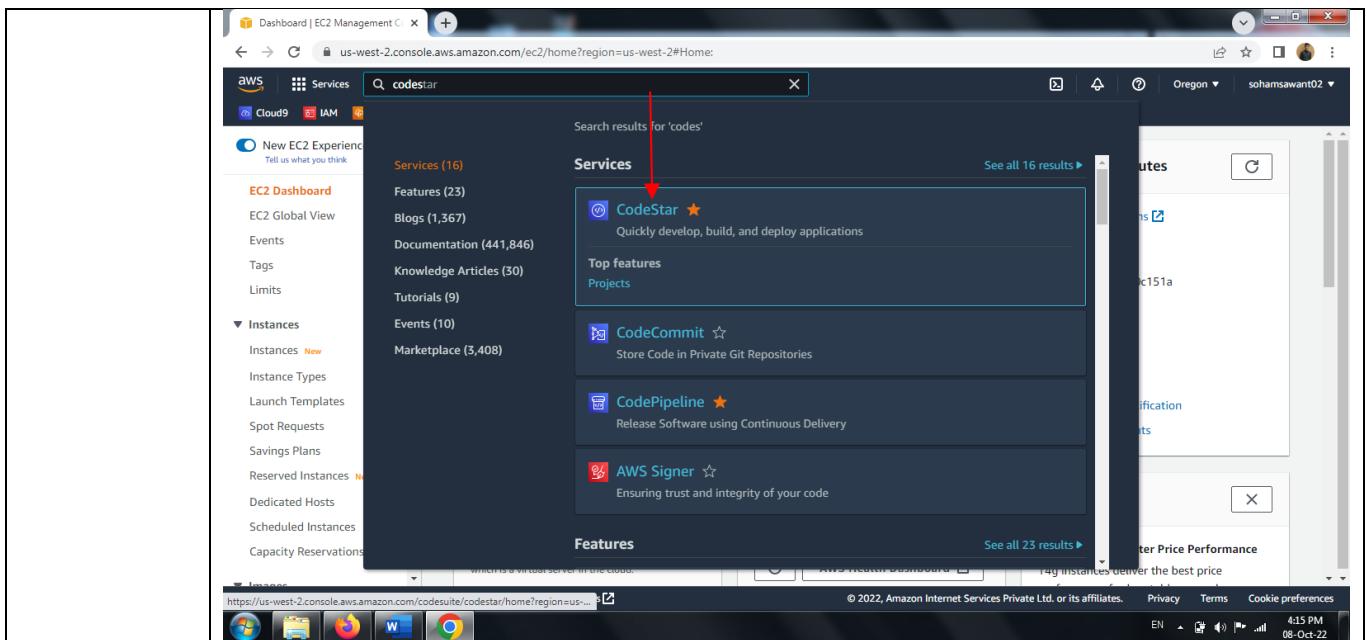
	 <p>Instance: i-0c5d95a3984c6c66e</p> <p>Details Security Networking Storage Status checks Monitoring Tags</p> <p>Instance ID: i-0c5d95a3984c6c66e (highlighted with a red arrow)</p> <p>Public IPv4 address: 34.193.144.252 open address</p> <p>Private IPv4 addresses: 172.31.90.6</p> <p>Public IPv4 DNS: ec2-34-193-144-252.compute-1.amazonaws.com open address</p>
Conclusion	Successfully set up Splunk on AWS.

Semester	T.E. Semester V – Information Technology
Subject	Advance DevOps Lab
Subject Professor In-charge	Prof. Indu Anoop
Laboratory	

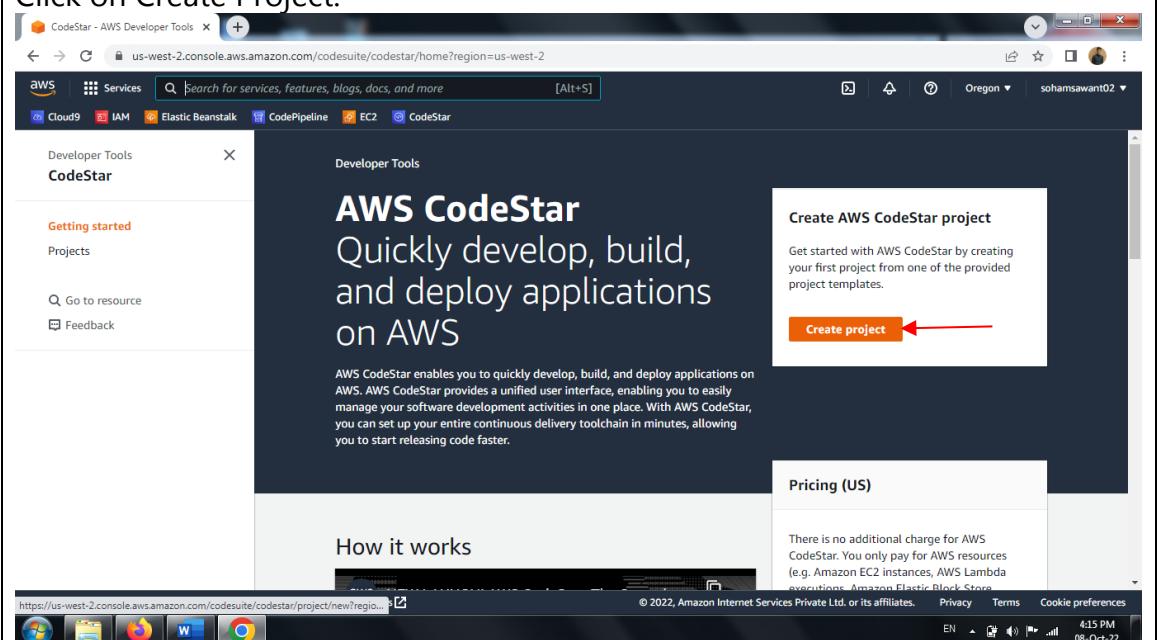
Student Name	Soham Anant Sawant	
Roll Number	21101B2002	
Grade and Subject Teacher's Signature		

Experiment	Self-Learning Topic 1	
Problem Statement	Understanding and implementing AWS Codestar, its features and difference with AWS Cloud9 IDE	
Resources / Apparatus Required	Hardware: Computer System	Software: Web Browser
Details	<p>Theory of AWS Code star:</p> <p>What is AWS Codestar?</p> <p>→ AWS CodeStar is a cloud-based development service that provides the tools you need to quickly develop, build, and deploy applications on AWS. With AWS CodeStar, you can set up your entire continuous delivery toolchain in minutes, allowing you to start releasing code faster. AWS CodeStar makes it easy for your whole team to work together securely, with built-in role-based policies that allow you to easily manage access and add owners, contributors, and viewers to your projects. Each AWS CodeStar project comes with a unified project dashboard and integration with Atlassian JIRA software, a third-party issue tracking and project management tool. With the AWS CodeStar project dashboard, you can easily track your entire software development process, from a backlog work item to production code deployment.</p> <p>Features:</p> <ul style="list-style-type: none"> → Project templates → Team access management → Hosted Git repository → Fully managed build service 	

	<ul style="list-style-type: none"> ➔ Automated continuous delivery pipeline ➔ Automated deployments ➔ IDE integrations ➔ Central project dashboard <p>Difference with AWS Cloud9 IDE:</p> <table border="1"> <thead> <tr> <th>AWS CODESTAR</th><th>AWS CLOUD9</th></tr> </thead> <tbody> <tr> <td>AWS CodeStar is a software development tool that enables you to quickly develop, build, and deploy applications on AWS. With CodeStar, you can setup your continuous delivery toolchain in minutes, allowing you to start releasing code faster.</td><td>Cloud9 is a cloud IDE for writing, running, and debugging code. Cloud9 comes pre-packaged with essential tools for many popular programming languages (JavaScript, Python, PHP, etc.) so you don't have to tinker with installing various compilers and toolchains.</td></tr> <tr> <td>With services like AWS CodeStar developers can very easily develop, debug, and deploy serverless applications in the cloud.</td><td>With services like AWS Cloud9 developers can very easily develop, debug, and deploy serverless applications in the cloud.</td></tr> <tr> <td>We can use the AWS CodeStar console to create AWS Cloud9 development environments for projects that store their code in Code Commit.</td><td>We can use AWS Cloud9 to make code changes and develop software in an AWS CodeStar project.</td></tr> <tr> <td>AWS CodeStar provides a number of project templates to help you quickly start developing applications for deployment on Amazon EC2, AWS Lambda, and AWS Elastic Beanstalk with support for many popular programming languages including Java, JavaScript, Python, Ruby, and PHP.</td><td>The IDE offers a rich code editing experience with support for several programming languages and runtime debuggers, as well as a built-in terminal</td></tr> </tbody> </table>	AWS CODESTAR	AWS CLOUD9	AWS CodeStar is a software development tool that enables you to quickly develop, build, and deploy applications on AWS. With CodeStar, you can setup your continuous delivery toolchain in minutes, allowing you to start releasing code faster.	Cloud9 is a cloud IDE for writing, running, and debugging code. Cloud9 comes pre-packaged with essential tools for many popular programming languages (JavaScript, Python, PHP, etc.) so you don't have to tinker with installing various compilers and toolchains.	With services like AWS CodeStar developers can very easily develop, debug, and deploy serverless applications in the cloud.	With services like AWS Cloud9 developers can very easily develop, debug, and deploy serverless applications in the cloud.	We can use the AWS CodeStar console to create AWS Cloud9 development environments for projects that store their code in Code Commit.	We can use AWS Cloud9 to make code changes and develop software in an AWS CodeStar project.	AWS CodeStar provides a number of project templates to help you quickly start developing applications for deployment on Amazon EC2, AWS Lambda, and AWS Elastic Beanstalk with support for many popular programming languages including Java, JavaScript, Python, Ruby, and PHP.	The IDE offers a rich code editing experience with support for several programming languages and runtime debuggers, as well as a built-in terminal
AWS CODESTAR	AWS CLOUD9										
AWS CodeStar is a software development tool that enables you to quickly develop, build, and deploy applications on AWS. With CodeStar, you can setup your continuous delivery toolchain in minutes, allowing you to start releasing code faster.	Cloud9 is a cloud IDE for writing, running, and debugging code. Cloud9 comes pre-packaged with essential tools for many popular programming languages (JavaScript, Python, PHP, etc.) so you don't have to tinker with installing various compilers and toolchains.										
With services like AWS CodeStar developers can very easily develop, debug, and deploy serverless applications in the cloud.	With services like AWS Cloud9 developers can very easily develop, debug, and deploy serverless applications in the cloud.										
We can use the AWS CodeStar console to create AWS Cloud9 development environments for projects that store their code in Code Commit.	We can use AWS Cloud9 to make code changes and develop software in an AWS CodeStar project.										
AWS CodeStar provides a number of project templates to help you quickly start developing applications for deployment on Amazon EC2, AWS Lambda, and AWS Elastic Beanstalk with support for many popular programming languages including Java, JavaScript, Python, Ruby, and PHP.	The IDE offers a rich code editing experience with support for several programming languages and runtime debuggers, as well as a built-in terminal										
Code	<p>Screenshots of implementation</p> <p>Go to CodeStar.</p>										



Click on Create Project.



Click on create service role.

CodeStar - AWS Developer Tools

us-west-2.console.aws.amazon.com/codesuite/codestar/project/new?region=us-west-2

Services

Cloud9 IAM Elastic Beanstalk CodePipeline EC2 CodeStar

CodeStar > Projects > Create project

Step 1 Choose a project template

Step 2 Set up your project

Step 3 Review

Create service role

AWS Codestar uses an IAM service role to manage resources on your behalf. You must create a service role in order to create and delete projects.

Create service role

Choose a project template

Templates

node Hello World Skill Application type

node Hello World Skill Application type

node Go Application type

Feedback Looking for language selection? Find it in the new Unified Settings

© 2022, Amazon Internet Services Private Ltd. or its affiliates. Privacy Terms Cookie preferences

Select Node.js

Success

You have successfully created aws-codestar-service-role. Allow a few moments for the changes to take effect.

node Hello World Skill Application type Alexa skill AWS service AWS Lambda Runs serverless

node Node.js Application type Web application AWS service AWS Lambda Runs serverless

node Express.js Application type Web service AWS service AWS Lambda Runs serverless

node Express.js Application type Web service AWS service AWS EC2 Runs on virtual servers that you manage

node Express.js Application type Web service AWS service AWS Elastic Beanstalk Runs in a managed application environment

node Node.js Application type Web application AWS service AWS EC2 Runs on virtual servers that you manage

Give Project Name

CodeStar - AWS Developer Tools

us-west-2.console.aws.amazon.com/codesuite/codestar/project/new?region=us-west-2

Success
You have successfully created aws-codestar-service-role. Allow a few moments for the changes to take effect.

Step 1
Choose a project template

Step 2
Set up your project

Step 3
Review

Set up your project

Project details

Project name: nodejs website

Project ID: nodejs-website

Project repository

Select a repository provider:

- CodeCommit
- GitHub

Feedback Looking for language selection? Find it in the new Unified Settings

© 2022, Amazon Internet Services Private Ltd. or its affiliates. Privacy Terms Cookie preferences

EN 4:17 PM 08-Oct-22

Go to EC2 dashboard and click on Key Pairs and Create key pair.

CodeStar - AWS Developer Tools

us-west-2.console.aws.amazon.com/ec2/home?region=us-west-2#KeyPairs:

Key pairs

New EC2 Experience Tell us what you think

EC2 Dashboard

EC2 Global View

Events

Tags

Limits

Instances

- Instances New
- Instance Types
- Launch Templates
- Spot Requests
- Savings Plans
- Reserved Instances New
- Dedicated Hosts
- Scheduled Instances
- Capacity Reservations

Images

Feedback Looking for language selection? Find it in the new Unified Settings

Actions

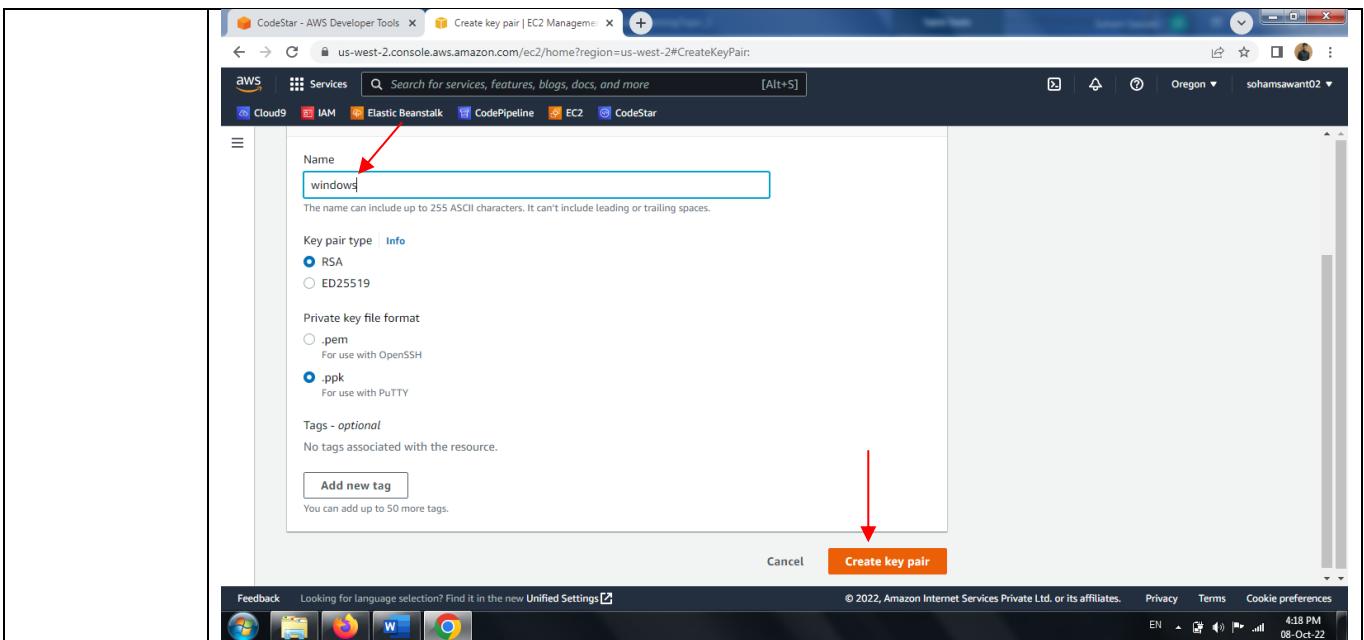
Create key pair

No key pairs to display

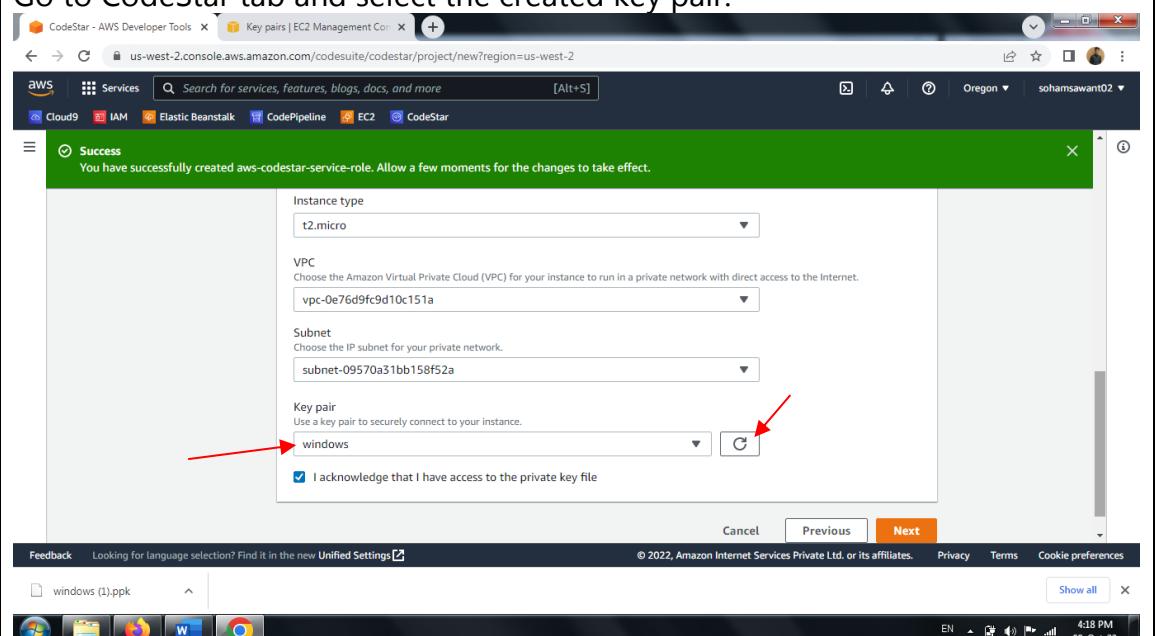
© 2022, Amazon Internet Services Private Ltd. or its affiliates. Privacy Terms Cookie preferences

EN 4:17 PM 08-Oct-22

Give name as windows and create.



Go to CodeStar tab and select the created key pair.



Click on Create Project.

CodeStar - AWS Developer Tools | Key pairs | EC2 Management Con... | +

us-west-2.console.aws.amazon.com/codesuite/codestar/project/new?region=us-west-2

Services Search for services, features, blogs, docs, and more [Alt+S]

Cloud9 IAM Elastic Beanstalk CodePipeline EC2 CodeStar

Success You have successfully created aws-codestar-service-role. Allow a few moments for the changes to take effect.

Project repository

Repository provider	Repository name	Repository description
AWS CodeCommit	nodejs-website	-

EC2 Configuration

Instance type	VPC	Subnet
t2.micro	vpc-0e76d9fc9d10c151a	subnet-09570a31bb158f52a
Key pair	windows	

Cancel Previous Create project

Feedback Looking for language selection? Find it in the new Unified Settings Show all

© 2022, Amazon Internet Services Private Ltd. or its affiliates. Privacy Terms Cookie preferences

windows (1).ppk

EN 4:19 PM 08-Oct-22

Wait till processing.

CodeStar - AWS Developer Tools | Key pairs | EC2 Management Con... | +

us-west-2.console.aws.amazon.com/codesuite/codestar/projects/nodejs-website?region=us-west-2&project-resources-meta=eyJmIjp7InRleHQiOiJlfSwicyI6eyJwcm9wZ... Show all

Services Search for services, features, blogs, docs, and more [Alt+S]

Cloud9 IAM Elastic Beanstalk CodePipeline EC2 CodeStar

Developer Tools CodeStar

Getting started Projects

Project Team Settings

Go to resource Feedback

Project provisioning AWS CodeStar is setting up your project's resources. This may take a few minutes.

nodejs website

Overview IDE Repository Pipeline Monitoring Issues

Next steps

Develop with AWS Cloud9 The AWS Cloud9 cloud-based IDE is fully integrated into AWS developer tools and ready to use in seconds.

Set up AWS Cloud9

Develop with other IDEs We also have integrations with Visual Studio, Eclipse, and the command line interface.

Set up other IDEs

Track issues Add issue tracking to your development workflow.

Set up issue tracking

Add people to your project Add team members to this project with IAM users.

Add team members

Feedback Looking for language selection? Find it in the new Unified Settings Show all

© 2022, Amazon Internet Services Private Ltd. or its affiliates. Privacy Terms Cookie preferences

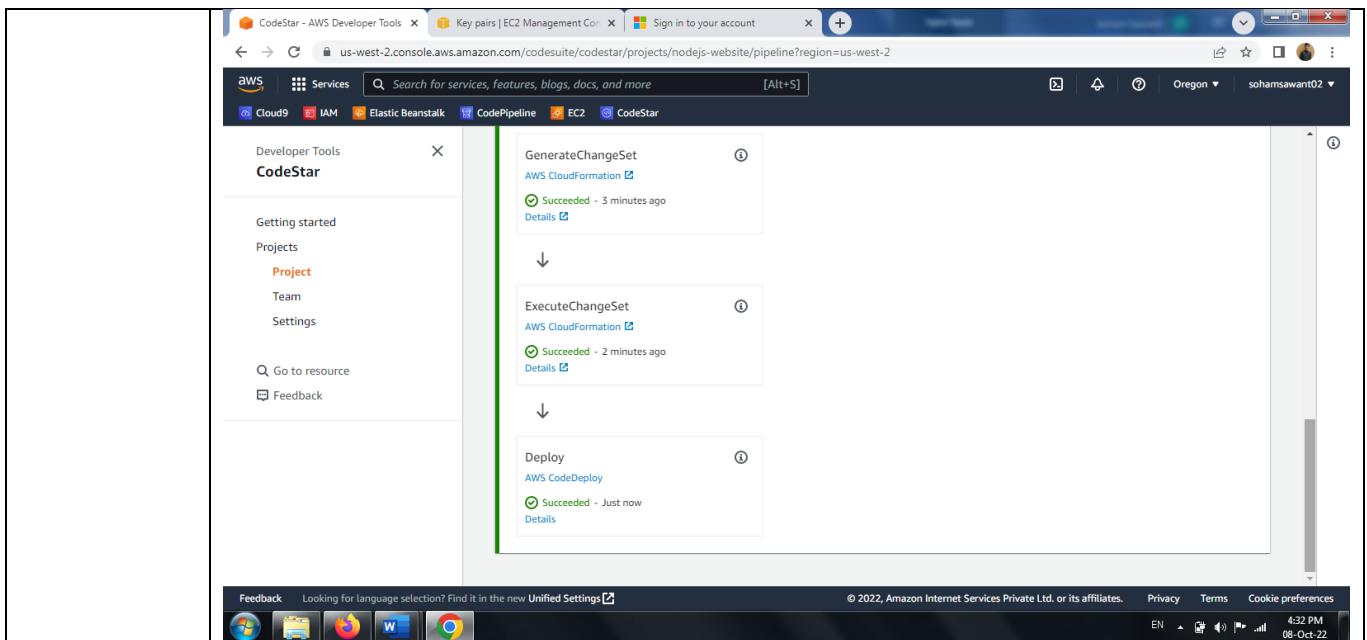
windows (1).ppk

EN 4:20 PM 08-Oct-22

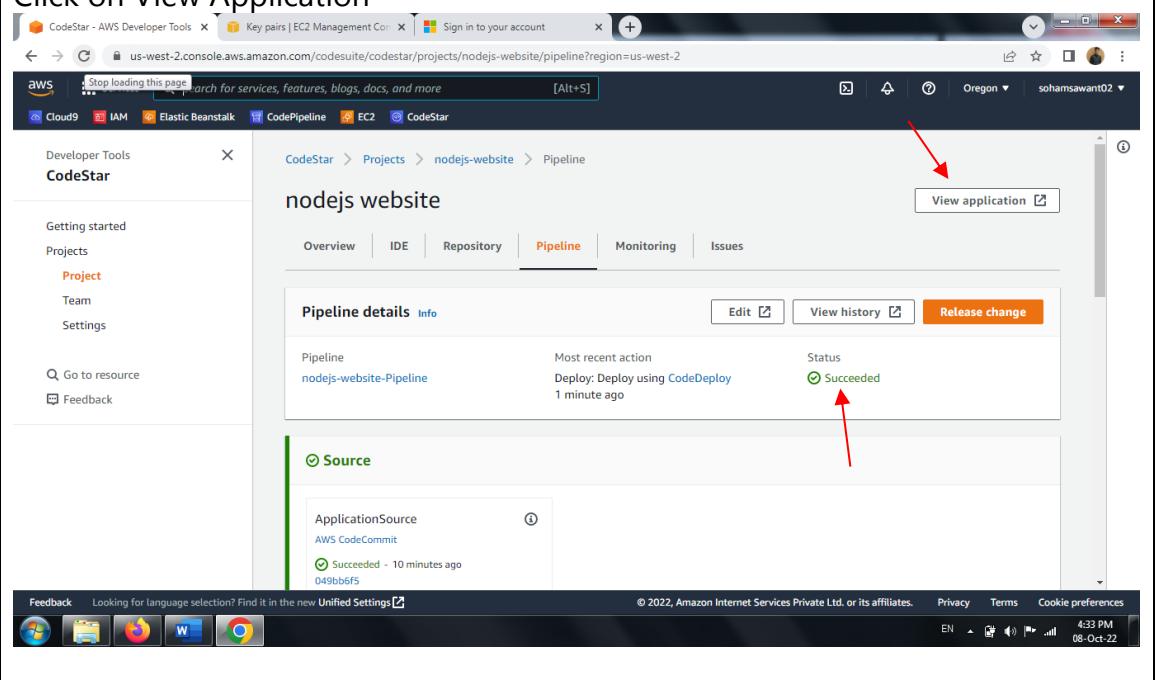
The screenshot shows the AWS CodeStar console. A green banner at the top says "Project provisioned". The main area displays the "nodejs website" project with tabs for Overview, IDE, Repository, Pipeline, Monitoring, and Issues. The Overview tab is selected. Below the tabs, there's a "Next steps" section with links to "Set up AWS Cloud9", "Develop with other IDEs", "Track issues", and "Add people to your project". The status bar at the bottom shows the URL <https://us-west-2.console.aws.amazon.com/codesuite/codestar/projects/nodejs-website/environments?region=us-west-2>.

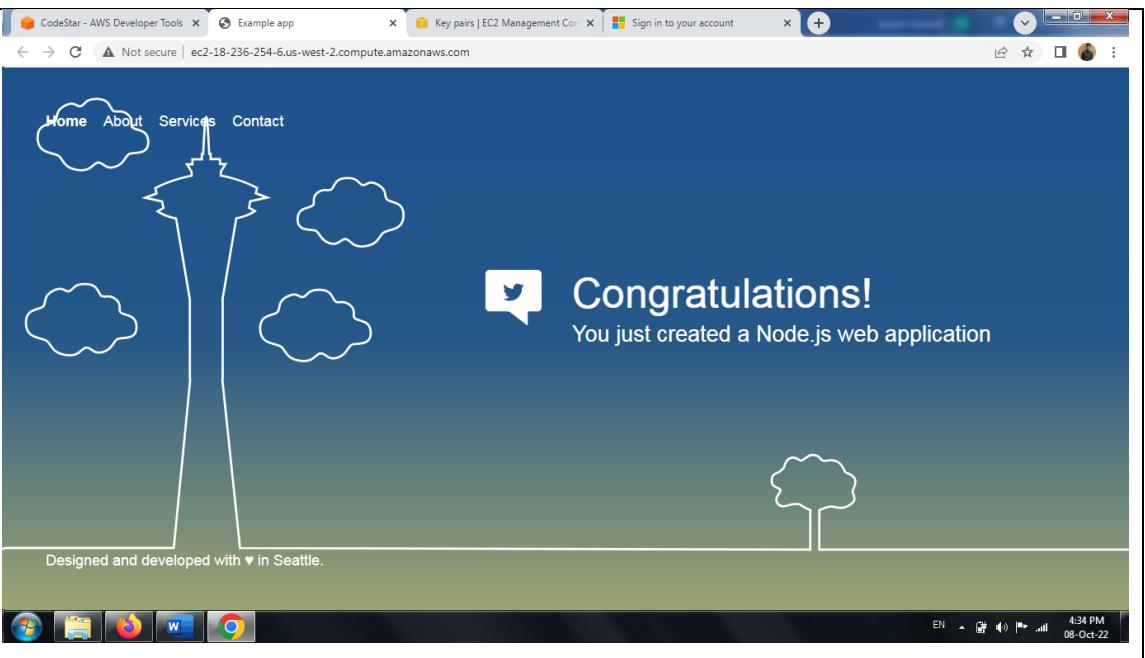
Go to Pipeline and wait till Status to become Succeeded.

The screenshot shows the AWS CodeStar console with the "Pipeline" tab selected in the "nodejs website" project. An orange arrow points to the "Pipeline" tab. The Pipeline details show a pipeline named "nodejs-website-Pipeline" with a most recent action of "Build: PackageExport using CodeBuild" from 2 minutes ago, and a status of "In progress". The Pipeline tab is highlighted with an orange border. The status bar at the bottom shows the URL <https://us-west-2.console.aws.amazon.com/codesuite/codestar/projects/nodejs-website/pipeline?region=us-west-2>.



Click on View Application

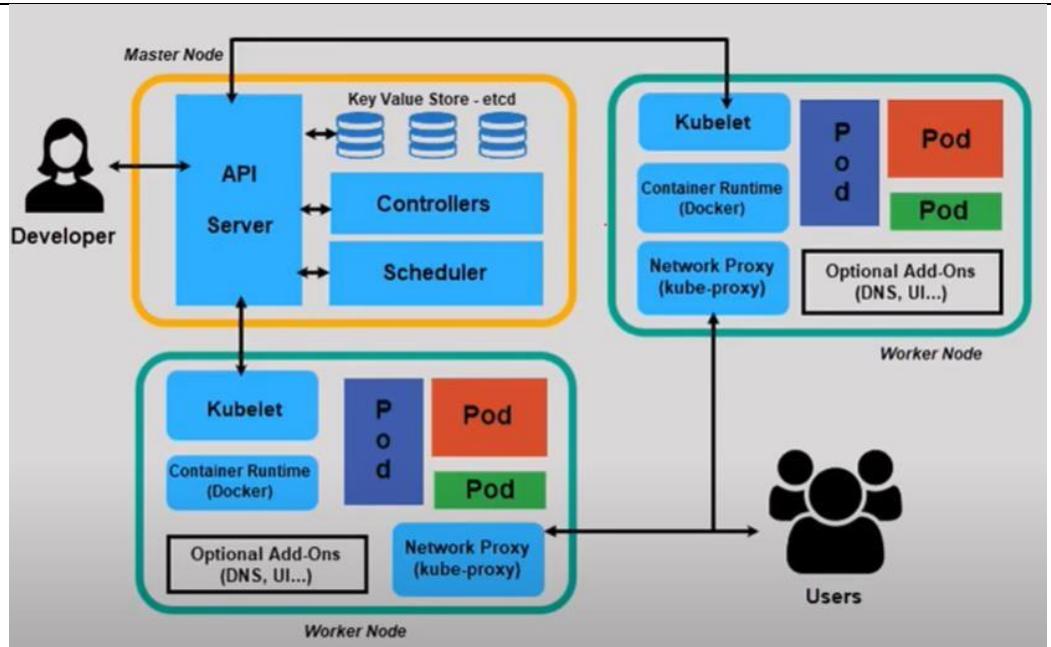


	
Conclusion	Hence we learnt to implement AWS Codestar and deployed Node.js Project.

Semester	T.E. Semester V – Information Technology
Subject	Advance DevOps Lab
Subject Professor In-charge	Prof. Indu Anoop
Laboratory	

Student Name	Soham Anant Sawant	
Roll Number	21101B2002	
Grade and Subject Teacher's Signature		

Experiment	3	
Problem Statement	To understand Kubernetes Cluster Architecture, install and Spin up a Kubernetes Cluster on Linux Machines/Cloud Platforms	
Resources / Apparatus Required	Hardware: Computer System (Internet Connectivity)	Software: Web Browser
Details	<p>Theory: Kubernetes led by google is an open-source platform for managing container technologies such as Docker.</p> <p>Docker lets you create containers for a pre-configured image and application. <i>Kubernetes [Greek for "Pilot"] provides the next step, allowing you to balance loads between containers and run multiple containers across multiple systems.</i></p>	



Container: Provides an isolated context in which an app together with its environment (supporting structure eg: web server) can run.

Pods: Represents a runnable unit usually consisting of a single container. [May contain more containers if containers are tightly coupled] Kubernetes connects the pod to the n/w and rest of the Kubernetes eco-system.

Code	<p>Prerequisite: 2 AWS instance (virtual servers-ubuntu 20) one acting as Master Node and Other as Worker Node https://mobaxterm.mobatek.net/download.html</p> <p><u>SECTION A: Docker Installation</u></p> <p><u>Step 1: Install Docker [on both master and worker]</u></p> <p>Kubernetes requires an existing Docker installation. If you already have Docker installed, skip ahead to Step 2. If you do not have Docker , install it by following these steps on both master and worker nodes after having acquired root privilege using command [sudo su]:</p> <pre><code>apt-get update</code></pre> <pre><code>apt-get install docker.io -y</code></pre> <pre><code>docker --version</code></pre>
------	---

Step 2: Start and Enable Docker [on both master and worker]

```
systemctl enable docker
```

```
systemctl status docker
```

```
systemctl start docker
```

SECTION B: Kubernetes Installation

Step 3: Add Kubernetes Signing Key [on both master and worker]

Since you are downloading Kubernetes from a non-standard repository, it is essential to ensure that the software is authentic. This is done by adding a signing key.

1. Enter the following to add a signing key:

```
curl -s https://packages.cloud.google.com/apt/doc/apt-key.gpg | apt-key add
```

If you get an error that curl is not installed, install it with:

```
apt-get install curl
```

2. Then repeat the previous command to install the signing keys. Repeat for each server node.

Step 4: Add Software Repositories [on both master and worker]

Kubernetes is not included in the default repositories. To add them, enter the following:

```
apt-add-repository "deb http://apt.kubernetes.io/ kubernetes-xenial main"
```

Repeat on each server node.

Step 5: Kubernetes Installation Tools [on both master and worker]

Kubeadm (Kubernetes Admin) is a tool that helps initialize a cluster. It fast-tracks setup by using community-sourced best practices. Kubelet is the work package, which runs on every node and starts containers. The tool gives you command-line access to clusters.

1. Install Kubernetes tools with the command:

```
apt-get install kubeadm kubelet kubectl -y
```

```
apt-mark hold kubeadm kubelet kubectl
```

Allow the process to complete.

2. Verify the installation with:

```
kubeadm version
```

3. Repeat for each server node.

SECTION C: Kubernetes Deployment

Step 6: Begin Kubernetes Deployment

Start by disabling the swap memory on each server **[on both master and worker]**:

```
swapoff --a
```

Step 7: Assign Unique Hostname for Each Server Node [on master only]

Decide which server to set as the master node. Then enter the command:

```
hostnamectl set-hostname master-node
```

Next, set a worker node hostname by entering the following on the worker server **[on worker only]**:

```
hostnamectl set-hostname worker
```

*If you have additional worker nodes, use this process to set a unique hostname on each.

Step 8: Initialize Kubernetes on Master Node [on master only]

Switch to the master server node, and enter the following:

```
kubeadm init --pod-network-cidr=10.244.0.0/16 --ignore-preflight-errors=all
```

Once this command finishes, it will display a kubeadm join message at the end. Make a note of the whole entry. This will be used to join the worker nodes to the cluster.

Next, enter the following to create a directory for the cluster:

```
mkdir -p $HOME/.kube
```

```
cp -i /etc/kubernetes/admin.conf $HOME/.kube/config
```

```
chown $(id -u):$(id -g) $HOME/.kube/config
```

Step 9: Deploy Pod Network to Cluster [on master only]

A Pod Network is a way to allow communication between different nodes in the cluster. We will use the flannel virtual network. Enter the following:

```
kubectl apply -f  
https://raw.githubusercontent.com/coreos/flannel/master/Documentation/  
kube-flannel.yml
```

Allow the process to complete.

Verify that everything is running and communicating:

```
kubectl get pods --all-namespaces
```

Step 10: Join Worker Node to Cluster [on worker only]

As indicated in Step 7, you can enter the kubeadm join command on each worker node to connect it to the cluster. Switch to the worker system and enter the command you noted from Step 7:

```
kubeadm join 172.31.30.132:6443 --token td0tas.u34zdwnvwxgh0nke \  
--discovery-token-ca-cert-hash  
sha256:47adb5a895cae9be531fa3219201bf2db921105dedd86f9a248f77c7  
4edb3eac
```

ON EC2 make sure you open the port in security group ADVERTISED HERE: Replace the alphanumeric codes with those from your master server. Repeat for each worker node on the cluster. Wait a few minutes; then you can check the status of the nodes.

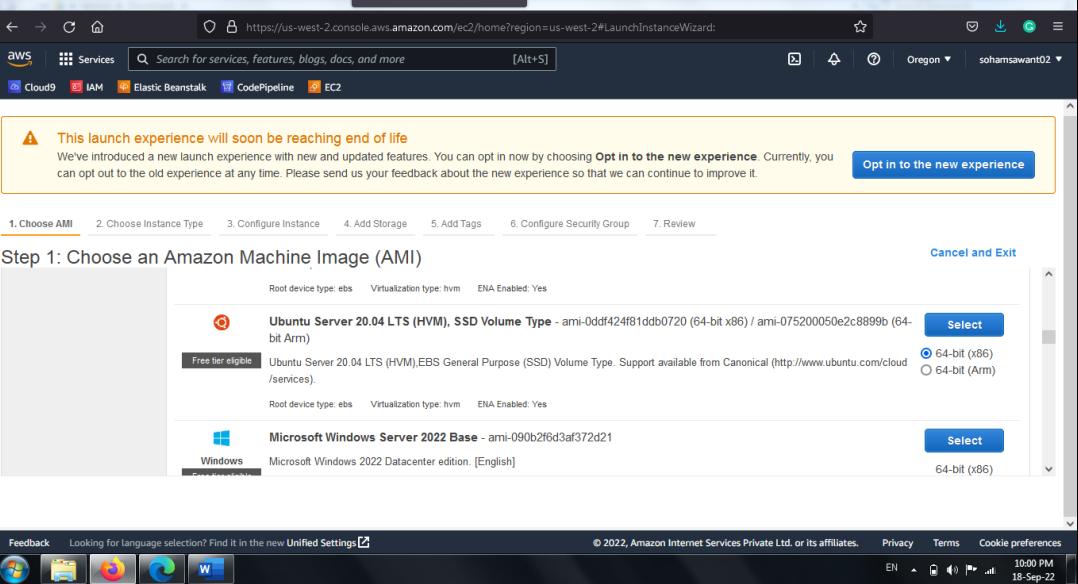
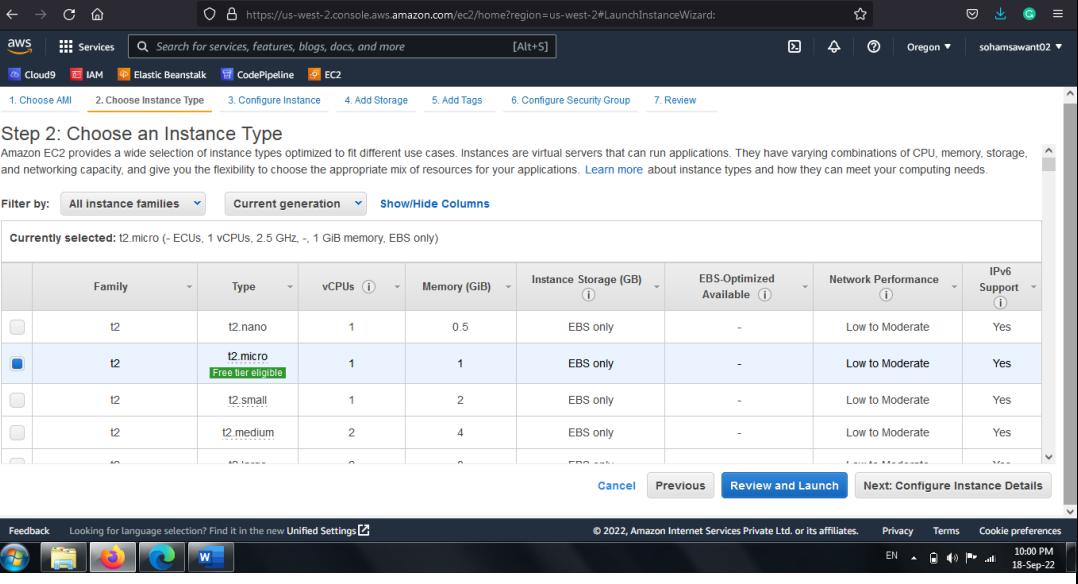
Switch to the master server, and enter:

```
kubectl get nodes
```

The system should display the worker nodes that you joined to the cluster.

NAME	STATUS	ROLES	AGE	VERSION
master	Ready	master	1d	v1.14.0
worker	Ready	<none>	1d	v1.14.0

If all of your nodes have the value Ready for STATUS, it means that they're part of the cluster and ready to run workloads. If, however, a few of the nodes have NotReady as the STATUS, it could mean that the worker nodes haven't finished their setup yet. Wait for around five to ten minutes before re-running kubectl get node and inspecting the new output. If a few nodes

	<p>still have NotReady as the status, you might have to verify and re-run the commands in the previous steps.</p> <p>(*Kubernetes to be continued in Exp4: use of kubectl command and to manage cluster and deploy application)</p> <p>[Note: In case you want to stop and continue later, Remember to stop the running instances on AWS , also detach the volume to not incur costs just in case a free-tier resource is not chosen]</p>
Output	 

Step 3: Configure Instance Details

Configure the instance to suit your requirements. You can launch multiple instances from the same AMI, request Spot instances to take advantage of the lower pricing, assign an access management role to the instance, and more.

Number of instances	<input type="text" value="2"/>	Launch into Auto Scaling Group
Purchasing option	<input type="checkbox"/> Request Spot instances	
Network	vpc-0e76d9fc9d10c151a (default)	<input type="button" value="Create new VPC"/>
Subnet	No preference (default subnet in any Availability Zone)	<input type="button" value="Create new subnet"/>
Auto-assign Public IP	<input type="checkbox"/> Use subnet setting (Enable)	
Hostname type	<input type="checkbox"/> Use subnet setting (IP name)	
DNS Hostname	<input checked="" type="checkbox"/> Enable IP name IPv4 (A record) DNS requests <input checked="" type="checkbox"/> Enable resource-based IPv4 (A record) DNS requests <input type="checkbox"/> Enable resource-based IPv6 (AAAA record) DNS requests	

Step 4: Add Storage

Your instance will be launched with the following storage device settings. You can attach additional EBS volumes and instance store volumes to your instance, or edit the settings of the root volume. You can also attach additional EBS volumes after launching an instance, but not instance store volumes. [Learn more](#) about storage options in Amazon EC2.

Volume Type	Device	Snapshot	Size (GiB)	Volume Type	IOPS	Throughput (MB/s)	Delete on Termination	Encryption
Root	/dev/sda1	snap-023fa25b5b51cc51a	<input type="text" value="8"/>	General Purpose SSD (gp2)	100 / 3000	N/A	<input checked="" type="checkbox"/>	<input type="button" value="Not Encrypt"/>

Step 5: Add Tags

A tag consists of a case-sensitive key-value pair. For example, you could define a tag with key = Name and value = Webserver. A copy of a tag can be applied to volumes, instances or both. Tags will be applied to all instances and volumes. [Learn more](#) about tagging your Amazon EC2 resources.

Key	(128 characters maximum)	Value	(256 characters maximum)	Instances	Volumes	Network Interfaces
-----	--------------------------	-------	--------------------------	-----------	---------	--------------------

This resource currently has no tags

Choose the Add tag button or [click to add a Name tag](#). Make sure your [IAM policy](#) includes permissions to create tags.

(Up to 50 tags maximum)

Step 6: Configure Security Group

Configure the security group for the instance. You can choose to allow specific traffic or restrict access to the instance. You can also choose to allow specific traffic or restrict access to the instance.

Step 7: Review

Review the configuration details for the instance. You can make changes or proceed to launch the instance.

Step 6: Configure Security Group

about Amazon EC2 security groups.

Assign a security group: Create a new security group
 Select an existing security group

Security group name: launch-wizard-1

Description: launch-wizard-1 created 2022-09-18T22:02:41.743+05:30

Type	Protocol	Port Range	Source	Description
SSH	TCP	22	Custom 0.0.0.0	e.g. SSH for Admin Desktop
HTTP	TCP	80	Anywhere 0.0.0.0, ::/0	e.g. SSH for Admin Desktop
HTTPS	TCP	443	Anywhere 0.0.0.0, ::/0	e.g. SSH for Admin Desktop
Custom TCP	TCP	6443	Anywhere 0.0.0.0, ::/0	e.g. SSH for Admin Desktop

Add Rule

Cancel **Previous** **Review and Launch**

Feedback Looking for language selection? Find it in the new [Unified Settings](#). © 2022, Amazon Internet Services Private Ltd. or its affiliates. Privacy Terms Cookie preferences EN 10:04 PM 18-Sep-22

Step 7: Review Instance Launch

Security group name: launch-wizard-1

Description: launch-wizard-1 created 2022-09-18T22:02:41.743+05:30

Type	Protocol	Port Range	Source	Description
SSH	TCP	22	0.0.0.0/0	
HTTP	TCP	80	0.0.0.0/0	
HTTP	TCP	80	::/0	
HTTPS	TCP	443	0.0.0.0/0	
HTTPS	TCP	443	::/0	
Custom TCP Rule	TCP	6443	0.0.0.0/0	
Custom TCP Rule	TCP	6443	::/0	

Instance Details

Edit instance details

Cancel **Previous** **Launch**

Feedback Looking for language selection? Find it in the new [Unified Settings](#). © 2022, Amazon Internet Services Private Ltd. or its affiliates. Privacy Terms Cookie preferences EN 10:05 PM 18-Sep-22

Select an existing key pair or create a new key pair

A key pair consists of a **public key** that AWS stores, and a **private key file** that you store. Together, they allow you to connect to your instance securely. For Windows AMIs, the private key file is required to obtain the password used to log into your instance. For Linux AMIs, the private key file allows you to securely SSH into your instance. Amazon EC2 supports ED25519 and RSA key pair types.

Note: The selected key pair will be added to the set of keys authorized for this instance. Learn more about [removing existing key pairs from a public AMI](#).

Create a new key pair
Key pair type: RSA ED25519
Key pair name: Kubernetes

Download Key Pair

You have to download the **private key file** (*.pem file) before you can continue. **Store it in a secure and accessible location.** You will not be able to download the file again after it's created.

Cancel **Launch Instances**

Feedback Looking for language selection? Find it in the new [Unified Settings](#). © 2022, Amazon Internet Services Private Ltd. or its affiliates. Privacy Terms Cookie preferences EN 10:06 PM 18-Sep-22

Launch Status

Your instances are now launching
The following instance launches have been initiated: [i-0a465ddc3d177d712](#), [i-05f90019811c78950](#) [View launch log](#)

Get notified of estimated charges
Create billing alerts to get an email notification when estimated charges on your AWS bill exceed an amount you define (for example, if you exceed the free usage tier).

How to connect to your instances

Your instances are launching, and it may take a few minutes until they are in the **running** state, when they will be ready for you to use. Usage hours on your new instances will start immediately and continue to accrue until you stop or terminate your instances.

Click [View Instances](#) to monitor your instances' status. Once your instances are in the **running** state, you can [connect](#) to them from the Instances screen. [Find out](#) how to connect to your instances.

Here are some helpful resources to get you started

- How to connect to your Linux instance
- Learn about AWS Free Usage Tier
- Amazon EC2 User Guide
- Amazon EC2 Discussion Forum

Instances (1 / 2) Info

Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone
K8sMaster	i-0a465ddc3d177d712	Running	t2.micro	Initializing	No alarms	us-west-2c
K8sWorker	i-05f90019811c78950	Running	t2.micro	Initializing	No alarms	us-west-2c

Instance: i-0a465ddc3d177d712 (K8sMaster)

Details | Security | Networking | Storage | Status checks | Monitoring | Tags

Instance summary

Instance ID i-0a465ddc3d177d712 (K8sMaster)	Public IPv4 address 54.189.93.192 open address	Private IPv4 addresses 172.31.10.9
IPv6 address -	Instance state Running	Public IPv4 DNS ec2-54-189-93-192.us-west-2.compute.amazonaws.com open address

Connect to instance

Connect to your instance i-0a465ddc3d177d712 using any of these options

EC2 Instance Connect | Session Manager | SSH client | EC2 serial console

Instance ID
[i-0a465ddc3d177d712](#)

Public IP address
[54.189.93.192](#)

User name

Note: In most cases, the guessed user name is correct. However, read your AMI usage instructions to check if the AMI owner has changed the default AMI user name.

Cancel | **Connect**

https://us-west-2.console.aws.amazon.com/ec2/home?region=us-west-2#ConnectToInstance\$instanceId=i-05f90019811c78950

EC2 Instance Connect

Instance ID: i-05f90019811c78950 (K8sWorker)

Public IP address: 35.164.2.189

User name: ubuntu

Note: In most cases, the guessed user name is correct. However, read your AMI usage instructions to check if the AMI owner has changed the default AMI user name.

Cancel Connect

Feedback Looking for language selection? Find it in the new Unified Settings

© 2022, Amazon Internet Services Private Ltd. or its affiliates. Privacy Terms Cookie preferences

EN 10:12 PM 18-Sep-22

https://us-west-2.console.aws.amazon.com/ec2-instance-connect/ssh?connType=standard&instanceId=i-0a465ddc3d177d712

```
ot@ip-172-31-10-9:/home/ubuntu$ hostnamectl set-hostname master-node
ot@ip-172-31-10-9:/home/ubuntu$ exit
it
ubuntu@ip-172-31-10-9:~$ sudo su
ot@master-node:/home/ubuntu$ tput setaf 02
ot@master-node:/home/ubuntu$ apt-get update
t:1 http://us-west-2.ec2.archive.ubuntu.com/ubuntu focal InRelease
t:2 http://us-west-2.ec2.archive.ubuntu.com/ubuntu focal-updates InRelease [114 kB]
t:3 http://us-west-2.ec2.archive.ubuntu.com/ubuntu focal-backports InRelease [108 kB]
t:4 http://us-west-2.ec2.archive.ubuntu.com/ubuntu focal/universe amd64 Packages [8628 kB]
t:5 http://security.ubuntu.com/ubuntu focal-security InRelease [114 kB]
t:6 http://us-west-2.ec2.archive.ubuntu.com/ubuntu focal/universe Translation-en [5124 kB]
t:7 http://us-west-2.ec2.archive.ubuntu.com/ubuntu focal/universe amd64 c-n-f Metadata [265 kB]
t:8 http://us-west-2.ec2.archive.ubuntu.com/ubuntu focal/multiverse amd64 Packages [114 kB]
t:9 http://us-west-2.ec2.archive.ubuntu.com/ubuntu focal/multiverse Translation-en [104 kB]
t:10 http://us-west-2.ec2.archive.ubuntu.com/ubuntu focal/multiverse amd64 c-n-f Metadata [9136 B]
t:11 http://us-west-2.ec2.archive.ubuntu.com/ubuntu focal-updates/main amd64 Packages [2086 kB]
t:12 http://us-west-2.ec2.archive.ubuntu.com/ubuntu focal-updates/main Translation-en [371 kB]
t:13 http://us-west-2.ec2.archive.ubuntu.com/ubuntu focal-updates/universe amd64 c-n-f Metadata [15.9 kB]
t:14 http://us-west-2.ec2.archive.ubuntu.com/ubuntu focal-updates/restricted amd64 Packages [1262 kB]
t:15 http://us-west-2.ec2.archive.ubuntu.com/ubuntu focal-updates/restricted Translation-en [179 kB]
t:16 http://us-west-2.ec2.archive.ubuntu.com/ubuntu focal-updates/restricted amd64 c-n-f Metadata [584 B]
t:17 http://us-west-2.ec2.archive.ubuntu.com/ubuntu focal-updates/universe amd64 Packages [953 kB]
t:18 http://us-west-2.ec2.archive.ubuntu.com/ubuntu focal-updates/universe Translation-en [218 kB]
```

i-0a465ddc3d177d712 (K8sMaster)

PublicIPs: 54.189.93.192 PrivateIPs: 172.31.10.9

Feedback Looking for language selection? Find it in the new Unified Settings

© 2022, Amazon Internet Services Private Ltd. or its affiliates. Privacy Terms Cookie preferences

EN 10:35 PM 18-Sep-22

https://us-west-2.console.aws.amazon.com/ec2-instance-connect/ssh?region=us-west-2&connType=standard&instanceId=i-05f90019811c78950

```
root@ip-172-31-5-21:/home/ubuntu$ hostnamectl set-hostname worker01
root@ip-172-31-5-21:/home/ubuntu$ exit
exit
ubuntu@ip-172-31-5-21:~$ sudo su
root@worker01:/home/ubuntu$ tput setaf 01
root@worker01:/home/ubuntu$ apt-get update
Get:1 http://us-west-2.ec2.archive.ubuntu.com/ubuntu focal InRelease
Get:2 http://us-west-2.ec2.archive.ubuntu.com/ubuntu focal-updates InRelease [114 kB]
Get:3 http://us-west-2.ec2.archive.ubuntu.com/ubuntu focal-backports InRelease [108 kB]
Get:4 http://us-west-2.ec2.archive.ubuntu.com/ubuntu focal/universe amd64 Packages [8628 kB]
Get:5 http://security.ubuntu.com/ubuntu focal-security InRelease [114 kB]
Get:6 http://us-west-2.ec2.archive.ubuntu.com/ubuntu focal/universe Translation-en [5124 kB]
Get:7 http://us-west-2.ec2.archive.ubuntu.com/ubuntu focal/universe amd64 c-n-f Metadata [265 kB]
Get:8 http://us-west-2.ec2.archive.ubuntu.com/ubuntu focal/multiverse amd64 Packages [114 kB]
Get:9 http://us-west-2.ec2.archive.ubuntu.com/ubuntu focal/multiverse Translation-en [104 kB]
Get:10 http://us-west-2.ec2.archive.ubuntu.com/ubuntu focal/multiverse amd64 c-n-f Metadata [9136 B]
Get:11 http://us-west-2.ec2.archive.ubuntu.com/ubuntu focal-updates/main amd64 Packages [2086 kB]
Get:12 http://us-west-2.ec2.archive.ubuntu.com/ubuntu focal-updates/main Translation-en [371 kB]
Get:13 http://us-west-2.ec2.archive.ubuntu.com/ubuntu focal-updates/universe amd64 c-n-f Metadata [15.9 kB]
Get:14 http://us-west-2.ec2.archive.ubuntu.com/ubuntu focal-updates/restricted amd64 Packages [1262 kB]
Get:15 http://us-west-2.ec2.archive.ubuntu.com/ubuntu focal-updates/restricted Translation-en [179 kB]
Get:16 http://us-west-2.ec2.archive.ubuntu.com/ubuntu focal-updates/restricted amd64 c-n-f Metadata [584 B]
Get:17 http://us-west-2.ec2.archive.ubuntu.com/ubuntu focal-updates/universe amd64 Packages [953 kB]
Get:18 http://us-west-2.ec2.archive.ubuntu.com/ubuntu focal-updates/universe Translation-en [218 kB]
```

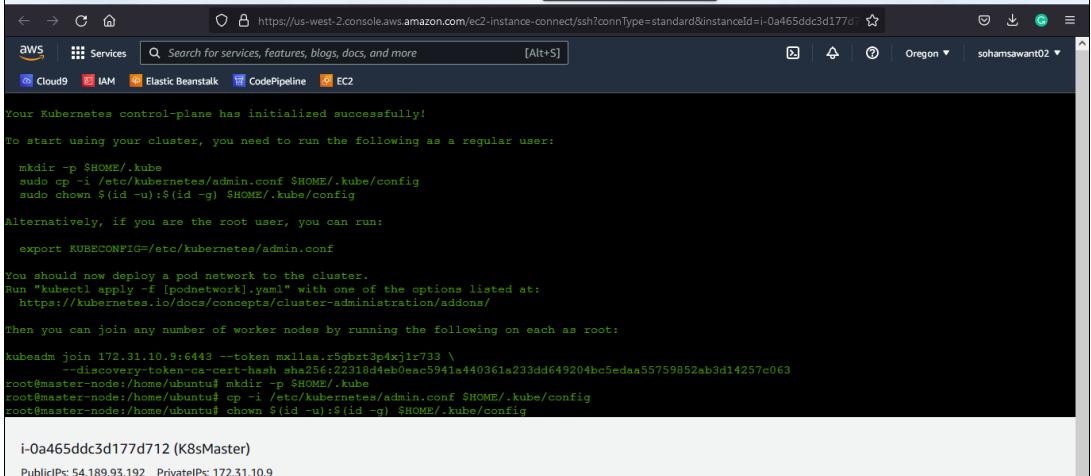
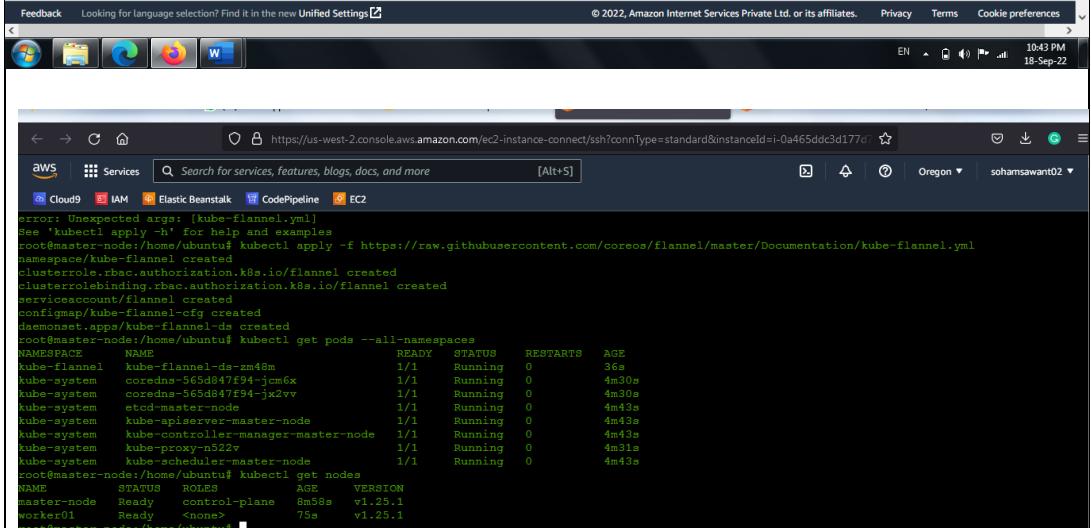
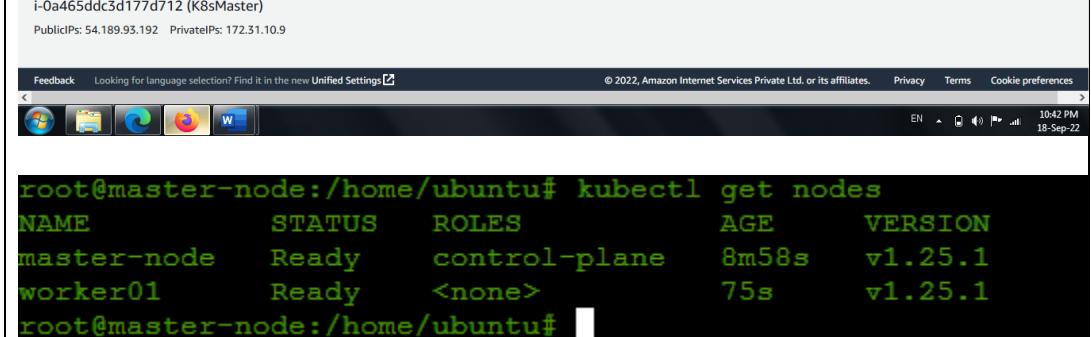
i-05f90019811c78950 (K8sWorker)

PublicIPs: 35.164.2.189 PrivateIPs: 172.31.5.21

Feedback Looking for language selection? Find it in the new Unified Settings

© 2022, Amazon Internet Services Private Ltd. or its affiliates. Privacy Terms Cookie preferences

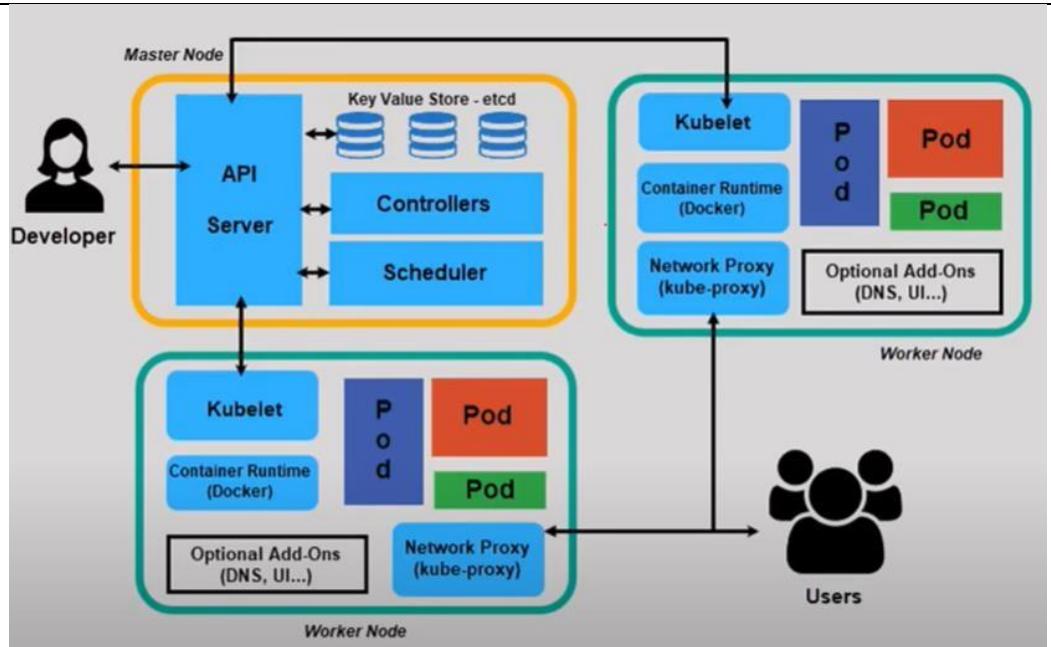
EN 11:05 PM 18-Sep-22

	 <p>Your Kubernetes control-plane has initialized successfully!</p> <p>To start using your cluster, you need to run the following as a regular user:</p> <pre>mkdir -p \$HOME/.kube sudo cp -i /etc/kubernetes/admin.conf \$HOME/.kube/config sudo chown \$(id -u):\$(id -g) \$HOME/.kube/config</pre> <p>Alternatively, if you are the root user, you can run:</p> <pre>export KUBECONFIG=/etc/kubernetes/admin.conf</pre> <p>You should now deploy a pod network to the cluster.</p> <p>Run "kubectl apply -f [podnetwork].yaml" with one of the options listed at: https://kubernetes.io/docs/concepts/cluster-administration/addons/</p> <p>Then you can join any number of worker nodes by running the following on each as root:</p> <pre>kubeadm join 172.31.10.9:6443 --token mxllaa.r5gbzt3p4xjlr733 \ --discovery-token-ca-cert-hash sha256:22318d4eb0eac5941a440361a233dd649204bc5edaa55759852ab3d14257c063 root@master-node:/home/ubuntu\$ mkdiz -p \$HOME/.kube root@master-node:/home/ubuntu\$ cp -i /etc/kubernetes/admin.conf \$HOME/.kube/config root@master-node:/home/ubuntu\$ chown \$(id -u):\$(id -g) \$HOME/.kube/config</pre> <p>i-0a465ddc3d177d712 (K8sMaster)</p> <p>PublicIP: 54.189.93.192 PrivateIP: 172.31.10.9</p>																																																																					
	 <p>Feedback Looking for language selection? Find it in the new Unified Settings [?]</p> <p>© 2022, Amazon Internet Services Private Ltd. or its affiliates. Privacy Terms Cookie preferences</p> <p>EN 10:43 PM 18-Sep-22</p> <p>root@master-node:/home/ubuntu\$ kubectl apply -f https://raw.githubusercontent.com/coreos/flannel/master/Documentation/kube-flannel.yml</p> <p>namespace/kube-flannel created</p> <p>clusterrole/rbac.authorization.k8s.io/flannel created</p> <p>clusterrolebinding/rbac.authorization.k8s.io/flannel created</p> <p>serviceaccount/flannel created</p> <p>configmap/kube-flannel-cfg created</p> <p>daemonset.apps/kube-flannel-ds created</p> <p>root@master-node:/home/ubuntu\$ kubectl get pods --all-namespaces</p> <table border="1"> <thead> <tr> <th>NAME/SPACE</th> <th>NAME</th> <th>READY</th> <th>STATUS</th> <th>RESTARTS</th> <th>AGE</th> </tr> </thead> <tbody> <tr> <td>kube-flannel</td> <td>kube-flannel-da-zm48m</td> <td>1/1</td> <td>Running</td> <td>0</td> <td>36s</td> </tr> <tr> <td>kube-system</td> <td>coredns-565d847ff94-jcm6x</td> <td>1/1</td> <td>Running</td> <td>0</td> <td>4m30s</td> </tr> <tr> <td>kube-system</td> <td>coredns-565d847ff94-jx2vv</td> <td>1/1</td> <td>Running</td> <td>0</td> <td>4m30s</td> </tr> <tr> <td>kube-system</td> <td>etcd-master-node</td> <td>1/1</td> <td>Running</td> <td>0</td> <td>4m43s</td> </tr> <tr> <td>kube-system</td> <td>kube-apiserver-master-node</td> <td>1/1</td> <td>Running</td> <td>0</td> <td>4m43s</td> </tr> <tr> <td>kube-system</td> <td>kube-controller-manager-master-node</td> <td>1/1</td> <td>Running</td> <td>0</td> <td>4m43s</td> </tr> <tr> <td>kube-system</td> <td>kube-proxy-n522v</td> <td>1/1</td> <td>Running</td> <td>0</td> <td>4m1s</td> </tr> <tr> <td>kube-system</td> <td>kube-scheduler-master-node</td> <td>1/1</td> <td>Running</td> <td>0</td> <td>4m43s</td> </tr> </tbody> </table> <p>root@master-node:/home/ubuntu\$ kubectl get nodes</p> <table border="1"> <thead> <tr> <th>NAME</th> <th>STATUS</th> <th>ROLES</th> <th>AGE</th> <th>VERSION</th> </tr> </thead> <tbody> <tr> <td>master-node</td> <td>Ready</td> <td>control-plane</td> <td>8m58s</td> <td>v1.25.1</td> </tr> <tr> <td>worker01</td> <td>Ready</td> <td><none></td> <td>75s</td> <td>v1.25.1</td> </tr> </tbody> </table> <p>i-0a465ddc3d177d712 (K8sMaster)</p> <p>PublicIPs: 54.189.93.192 PrivateIPs: 172.31.10.9</p>	NAME/SPACE	NAME	READY	STATUS	RESTARTS	AGE	kube-flannel	kube-flannel-da-zm48m	1/1	Running	0	36s	kube-system	coredns-565d847ff94-jcm6x	1/1	Running	0	4m30s	kube-system	coredns-565d847ff94-jx2vv	1/1	Running	0	4m30s	kube-system	etcd-master-node	1/1	Running	0	4m43s	kube-system	kube-apiserver-master-node	1/1	Running	0	4m43s	kube-system	kube-controller-manager-master-node	1/1	Running	0	4m43s	kube-system	kube-proxy-n522v	1/1	Running	0	4m1s	kube-system	kube-scheduler-master-node	1/1	Running	0	4m43s	NAME	STATUS	ROLES	AGE	VERSION	master-node	Ready	control-plane	8m58s	v1.25.1	worker01	Ready	<none>	75s	v1.25.1
NAME/SPACE	NAME	READY	STATUS	RESTARTS	AGE																																																																	
kube-flannel	kube-flannel-da-zm48m	1/1	Running	0	36s																																																																	
kube-system	coredns-565d847ff94-jcm6x	1/1	Running	0	4m30s																																																																	
kube-system	coredns-565d847ff94-jx2vv	1/1	Running	0	4m30s																																																																	
kube-system	etcd-master-node	1/1	Running	0	4m43s																																																																	
kube-system	kube-apiserver-master-node	1/1	Running	0	4m43s																																																																	
kube-system	kube-controller-manager-master-node	1/1	Running	0	4m43s																																																																	
kube-system	kube-proxy-n522v	1/1	Running	0	4m1s																																																																	
kube-system	kube-scheduler-master-node	1/1	Running	0	4m43s																																																																	
NAME	STATUS	ROLES	AGE	VERSION																																																																		
master-node	Ready	control-plane	8m58s	v1.25.1																																																																		
worker01	Ready	<none>	75s	v1.25.1																																																																		
	 <p>Feedback Looking for language selection? Find it in the new Unified Settings [?]</p> <p>© 2022, Amazon Internet Services Private Ltd. or its affiliates. Privacy Terms Cookie preferences</p> <p>EN 10:42 PM 18-Sep-22</p> <pre>root@master-node:/home/ubuntu# kubectl get nodes NAME STATUS ROLES AGE VERSION master-node Ready control-plane 8m58s v1.25.1 worker01 Ready <none> 75s v1.25.1 root@master-node:/home/ubuntu#</pre>																																																																					
Conclusion	Kubernetes was installed and a Kubernetes Cluster was spun up on a ubuntu machine using AWS Cloud Platform.																																																																					

Semester	T.E. Semester V – Information Technology
Subject	Advance DevOps Lab
Subject Professor In-charge	Prof. Indu Anoop
Laboratory	

Student Name	Soham Anant Sawant	
Roll Number	21101B2002	
Grade and Subject Teacher's Signature		

Experiment	4	
Problem Statement	To install kubectl and execute kubectl commands to manage the Kubernetes cluster and deploy your first Kubernetes Application.	
Resources / Apparatus Required	Hardware: Computer System (Internet Connectivity)	Software: Web Browser
Details	<p>Theory: Kubernetes led by google is an open-source platform for managing container technologies such as Docker.</p> <p>Docker lets you create containers for a pre-configured image and application. <i>Kubernetes [Greek for "Pilot"] provides the next step, allowing you to balance loads between containers and run multiple containers across multiple systems.</i></p>	



Container: Provides an isolated context in which an app together with its environment (supporting structure eg: web server) can run.

Pods: Represents a runnable unit usually consisting of a single container. [May contain more containers if containers are tightly coupled] Kubernetes connects the pod to the n/w and rest of the Kubernetes eco-system.

Code	<p>Prerequisite: 2 AWS instance (virtual servers-ubuntu 20) one acting as Master Node and Other as Worker Node. Docker and Kubernetes installation done on both nodes. https://mobaxterm.mobatek.net/download.html</p> <p>Now that your cluster is verified successfully, let's schedule an example Nginx application on the cluster.</p> <p><u>SECTION D: Running An Application on the Cluster</u></p> <p>You can now deploy any containerized application to your cluster. To keep things familiar, let's deploy Nginx using Deployments and Services to see how this application can be deployed to the cluster. You can use the commands below for other containerized applications as well, provided you change the Docker image name and any relevant flags (such as ports and volumes).</p>
------	---

Step 1: Create deployment named nginx [on master]

Still within the master node, execute the following command to create a deployment named nginx:

```
kubectl create deployment nginx --image=nginx
```

A deployment is a type of Kubernetes object that ensures there's always a specified number of pods running based on a defined template, even if the pod crashes during the cluster's lifetime.

The above deployment will create a pod with one container from the Docker registry's Nginx Docker Image.

Next, run the following command to create a service named nginx that will expose the app publicly. It will do so through a NodePort, a scheme that will make the pod accessible through an arbitrary port opened on each node of the cluster:

```
kubectl expose deploy nginx --port 80 --target-port 80 --type NodePort
```

Services are another type of Kubernetes object that expose cluster internal services to clients, both internal and external. They are also capable of load balancing requests to multiple pods, and are an integral component in Kubernetes, frequently interacting with other components.

Run the following command:

```
kubectl get services
```

This will output text like the following:

Output

NAME	TYPE	CLUSTER-IP	EXTERNAL-IP	PORT(S)	AGE
kubernetes	ClusterIP	10.96.0.1	<none>	443/TCP	1d
nginx	NodePort	10.109.228.209	<none>	80: nginx_port /TCP	40m

From the third line of the above output, you can retrieve the port that Nginx is running on. Kubernetes will assign a random port that is **greater than 30000** automatically, while ensuring that the port is not already bound by another service.

Note: if you're running your setup on ec2 ensure the nginx_port is open under the inbound rules in the security groups.

To test that everything is working, visit
`http://worker_1_ip:nginx_port`
or
`http://worker_2_ip:nginx_port`
through a browser on your local machine. You will see Nginx's familiar welcome page.

To see the deployed container on worker node switch to worker
`docker ps`

Output: you will see the container for nginx image running.

SECTION E: Scale up replicas for a deployment

If you want to scale up the replicas for a deployment (nginx in our case) the use the following command:

```
kubectl scale --current-replicas=1 --replicas=2 deployment/nginx  
kubectl get pods
```

Output: you will see 2/2 as output in nginx deployment.

```
kubectl describe deployment/nginx
```

Output: give details about the service deployed

If you would like to remove the Nginx application, first delete the nginx service from the master node:

```
kubectl delete service nginx
```

Run the following to ensure that the service has been deleted:

```
kubectl get services
```

You will see the following output:

Output

NAME	TYPE	CLUSTER-IP	EXTERNAL-IP	PORT(S)	AGE
kubernetes	ClusterIP	10.96.0.1	<none>	443/TCP	1d

Then delete the deployment:

```
kubectl delete deployment nginx
```

Run the following to confirm that this worked:

```
kubectl get deployments
```

Output

No resources found.

Output

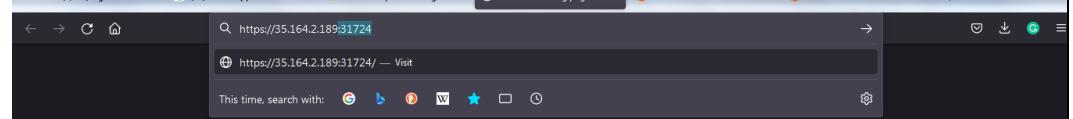
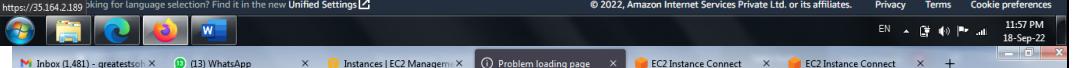
On master node:

```
root@master-node:/home/ubuntu# kubectl create deployment nginx --image=nginx
deployment.apps/nginx created
root@master-node:/home/ubuntu# kubectl expose deploy nginx --port 80 --target-port 80 --type NodePort
service/nginx exposed
root@master-node:/home/ubuntu# kubectl get services
NAME      TYPE      CLUSTER-IP   EXTERNAL-IP   PORT(S)      AGE
kubernetes  ClusterIP  10.96.0.1    <none>        443/TCP     82m
nginx      NodePort  10.109.157.40  <none>        80:31724/TCP 33s
root@master-node:/home/ubuntu#
```

i-0a465ddc3d177d712 (K8sMaster)
Public IPs: 54.189.93.192 Private IPs: 172.31.10.9



This screenshot shows the AWS EC2 Instances page. It lists two instances: 'K8sMaster' (Instance ID: i-0a465ddc3d177d712) and 'K8sWorker' (Instance ID: i-05f90019811c78950). Both instances are in a 'Running' state and are t2.micro type. The 'K8sWorker' instance is highlighted with a blue circle. The page also shows filters for Instance state (set to 'running'), a search bar, and various navigation and action buttons.



Unable to connect

An error occurred during a connection to 35.164.2.189.

- The site could be temporarily unavailable or too busy. Try again in a few moments.
- If you are unable to load any pages, check your computer's network connection.
- If your computer or network is protected by a firewall or proxy, make sure that Firefox is permitted to access the Web.

[Try Again](#)



The following screenshots demonstrate the configuration of an AWS Security Group (SG) for an EC2 instance.

Screenshot 1: EC2 Instances Overview

Screenshot 2: Security Group Details

Name	Protocol	Port Range	Type	Description
sgr-0bd7dec30deb845...	HTTPS	443	Custom TCP	Custom TCP
sgr-0e5e8cadcd1e24e2e	Custom TCP	6443	Custom TCP	Custom TCP
sqr-082d1acb35f0266a8	HTTP	80	Custom TCP	HTTP
-	TCP	12345	Custom TCP	Custom TCP
-	TCP	12345	Custom TCP	Custom TCP
-	TCP	12345	Custom TCP	Custom TCP

Screenshot 3: Modify Inbound Security Group Rules

Protocol	Port Range	Type	Description
HTTP	80	Custom TCP	Custom TCP
HTTPS	443	Custom TCP	Custom TCP
Custom TCP	6443	Custom TCP	Custom TCP
SSH	22	Custom TCP	Custom TCP
HTTP	80	Custom TCP	Custom TCP
Custom TCP	31724	Custom TCP	Custom TCP

The connection has timed out

An error occurred during a connection to 35.164.2.189:31724.

- The site could be temporarily unavailable or too busy. Try again in a few moments.
- If you are unable to load any pages, check your computer's network connection.
- If your computer or network is protected by a firewall or proxy, make sure that Firefox is permitted to access the Web.

[Try Again](#)

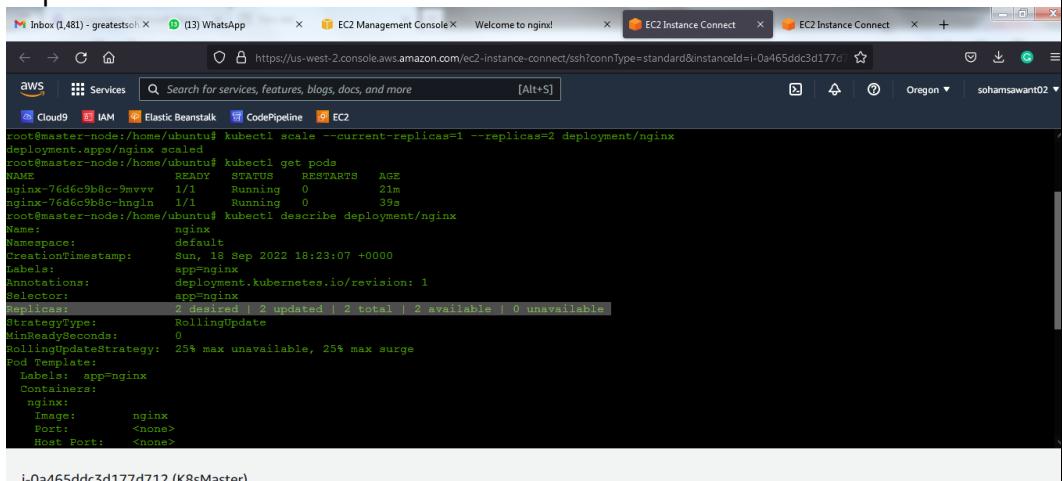
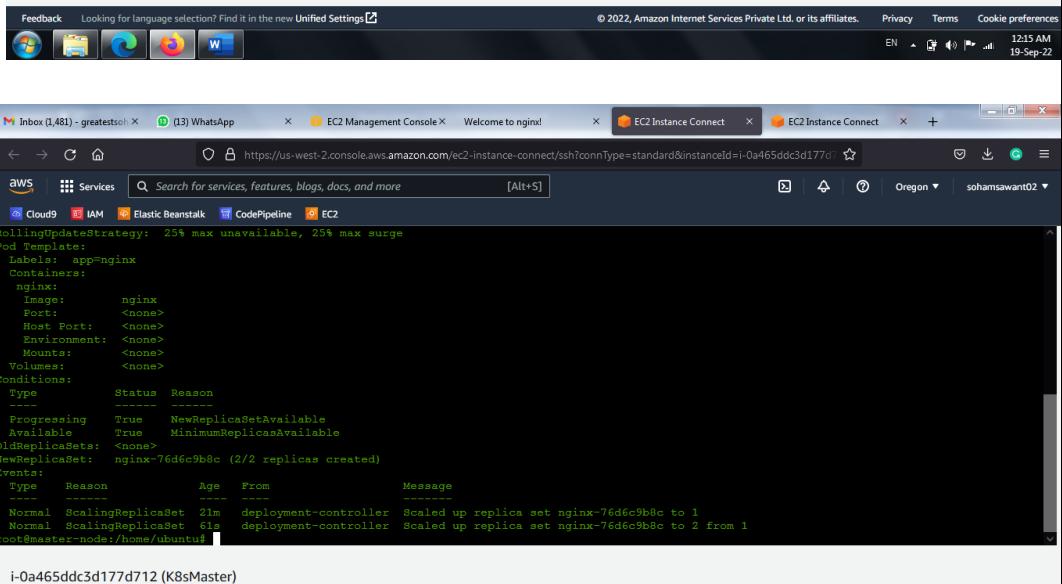
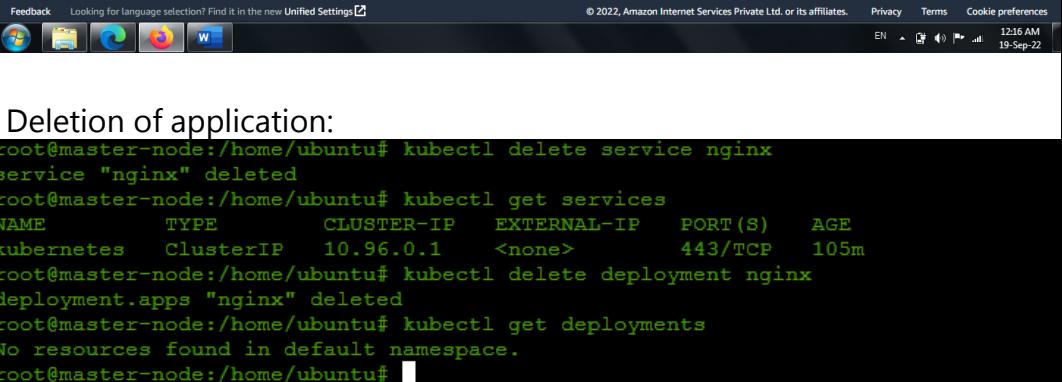
Access of worker node ip via browser to see successfully deployed application:

Welcome to nginx!

If you see this page, the nginx web server is successfully installed and working. Further configuration is required.

For online documentation and support please refer to nginx.org. Commercial support is available at nginx.com.

Thank you for using nginx.

	<h3>Replication of Pods:</h3>  <pre> root@master-node:/home/ubuntu# kubectl scale --current-replicas=1 --replicas=2 deployment/nginx deployment.apps/nginx scaled root@master-node:/home/ubuntu# kubectl get pods NAME READY STATUS RESTARTS AGE nginx-76d6c9b8c-9mvr 1/1 Running 0 21m nginx-76d6c9b8c-hgln 1/1 Running 0 39s root@master-node:/home/ubuntu# kubectl describe deployment/nginx Name: nginx Namespace: default CreationTimestamp: Sun, 18 Sep 2022 18:23:07 +0000 Labels: app=nginx Annotations: deployment.kubernetes.io/revision: 1 Selector: app=nginx Status: 2 desired 2 updated 2 total 2 available 0 unavailable StrategyType: RollingUpdate MinAvailable: 0 MaxUnavailable: 25% max unavailable, 25% max surge Pod Template: Labels: app=nginx Containers: nginx: Image: nginx Port: <none> Host Port: <none> </pre> <p>i-0a465ddc3d177d712 (K8sMaster) PublicIPs: 54.189.93.192 PrivateIPs: 172.31.10.9</p>
	 <pre> Feedback Looking for language selection? Find it in the new Unified Settings © 2022, Amazon Internet Services Private Ltd. or its affiliates. Privacy Terms Cookie preferences EN 12:15 AM 19-Sep-22 root@master-node:/home/ubuntu# kubectl get events PodTemplate: 25% max unavailable, 25% max surge Pod Template: Labels: app=nginx Containers: nginx: Image: nginx Port: <none> Host Port: <none> Environment: <none> Mounts: <none> Volumes: <none> Conditions: Type Status Reason Progressing True NewReplicaSetAvailable Available True MinimumReplicasAvailable OldReplicaSets: <none> NewReplicaSet: nginx-76d6c9b8c (2/2 replicas created) Events: Type Reason Age From Message ---- ----- -- --- ----- Normal ScalingReplicaSet 21m deployment-controller Scaled up replica set nginx-76d6c9b8c to 1 Normal ScalingReplicaSet 61s deployment-controller Scaled up replica set nginx-76d6c9b8c to 2 from 1 root@master-node:/home/ubuntu# </pre> <p>i-0a465ddc3d177d712 (K8sMaster) PublicIPs: 54.189.93.192 PrivateIPs: 172.31.10.9</p>
	 <pre> Feedback Looking for language selection? Find it in the new Unified Settings © 2022, Amazon Internet Services Private Ltd. or its affiliates. Privacy Terms Cookie preferences EN 12:16 AM 19-Sep-22 root@master-node:/home/ubuntu# kubectl delete service nginx service "nginx" deleted root@master-node:/home/ubuntu# kubectl get services NAME TYPE CLUSTER-IP EXTERNAL-IP PORT(S) AGE kubernetes ClusterIP 10.96.0.1 <none> 443/TCP 105m root@master-node:/home/ubuntu# kubectl delete deployment nginx deployment.apps "nginx" deleted root@master-node:/home/ubuntu# kubectl get deployments No resources found in default namespace. root@master-node:/home/ubuntu# </pre> <p>i-0a465ddc3d177d712 (K8sMaster) PublicIPs: 54.189.93.192 PrivateIPs: 172.31.10.9</p>
Conclusion	executed kubectl commands to manage the Kubernetes cluster and deploy a nginx Application.

Semester	T.E. Semester V – Information Technology
Subject	Advance DevOps Lab
Subject Professor In-charge	Prof. Indu Anoop
Laboratory	

Student Name	Soham Anant Sawant	
Roll Number	21101B2002	
Grade and Subject Teacher's Signature		

Experiment	9 and 10	
Problem Statement	Installation of Nagios and NPREG Plugin	
Resources / Apparatus Required	Hardware: Computer System	Software: Web Browser
Details	<p>Nagios is an open source monitoring system for computer systems. It was designed to run on the Linux operating system and can monitor devices running Linux, Windows and Unix operating systems (OSes).</p> <p>Nagios software runs periodic checks on critical parameters of application, network and server resources. For example, Nagios can monitor memory usage, disk usage, microprocessor load, the number of currently running processes and log files. Nagios also can monitor services, such as Simple Mail Transfer Protocol (SMTP), Post Office Protocol 3 (POP3), Hypertext Transfer Protocol (HTTP) and other common network protocols. Active checks are initiated by Nagios, while passive checks come from external applications connected to the monitoring tool.</p>	
Code	<u>Demo of Nagios</u> https://nagiosxi.demos.nagios.com/ <u>PART A: Steps: Actual Installation of Nagios on Host PC</u> Download Nagios .ova file and Install VMWare Player 16 to import the .ova file https://www.nagios.com/downloads/nagios-xi/vmware/	

The Nagios website displays a pre-installed OVA for Nagios XI. A red arrow points to the "Download Now" button for the "Workstation Pro/Player" section. Below the main content, a screenshot of the VMware website shows the download progress for the "nagiosxi-5.9.1-64.ova" file.

- Right click on nagios in vmware startup and check setting in hardware , go to network adapter
- in network connecton, change it to NAT : used to share the host's IP address
- Click ok and start you vm, ip address should appear

The image shows three sequential screenshots of the Nagios XI installation process:

- Terminal Session:** A terminal window titled "Nagios - VMware Workstation 16 Player (Non-commercial use only)" showing the Nagios XI logo and a system log message: "localhost login: [21:564695] systemd-journal[1637]: Time jumped backwards, rotating. [- 29.108115] hettimer: interrupt took 497698 us". A red arrow points to the "Default root Password: nagi0ns1" prompt.
- Welcome Screen:** A web browser window titled "Nagios XI" showing the "Welcome" page. It includes a "Access Nagios XI" button, which is highlighted with a red arrow. Below it, there is a note about the Nagios Library and support forums.
- Install Screen:** A web browser window titled "Install - Nagios XI" showing the "Nagios XI Installation" page. It displays "General System Settings" with fields for Program URL (http://192.168.152.128/nagiosxi/), Timezone (UTC+05:30 Mumbai), Language (English (English)), and User Interface Theme (Modern). A checkbox for "Use HTTPS only (all HTTP requests will be redirected to HTTPS)" is present. Below this is the "License Settings" section, which shows "License Type" with options Trial, Licensed, and Free (Limited) selected. The "Free" option is highlighted with a red arrow. A note at the bottom states: "Free license is limited to 7 nodes and up to a total of 100 host/service checks".

Nagios XI Install

Nagios XI Installation

Finalize your Nagios XI installation and step the initial configuration. These settings can be changed later.

Admin Account Settings

Username	nagiosadmin
Password	dL%TdWsfEucuosFL2og#
Full Name	Soham Sawant
Email Address	sohamseawant2000@gmail.com

Admin Notification Settings

Send this account email notifications [?](#) [Advanced email notification settings](#)

Nagios XI Login

Login

nagiosadmin

Login
[Forgot your password?](#)

Select Language:

PART A Output : Can view localhost PC for status monitoring [Installation of Nagios Done]

Monitoring of localhost

Nagios XI

Home Views Dashboards Reports Configure Tools Help Admin

Upgrade to a licensed version of Nagios XI and get support and upgrade benefits.

Home Dashboard

Getting Started Guide

- Change your account settings
- Change your notifications settings
- Configure your monitoring setup

Host Status Summary

Up	Down	Unreachable	Pending
1	0	0	0
Unhandled	Problems	All	
0	0	1	

Last Updated: 2022-10-31 08:53:33

Service Status Summary

Ok	Warning	Unknown	Critical	Pending
12	0	0	0	0
Unhandled	Problems	All		
0	0	12		

Last Updated: 2022-10-31 08:53:33

Nagios XI 5.9.1 • Check for Updates

PART B: [Setup] Monitor Remote Host/PC using NPRE (Nagios Remote Plugin Executor)

Install Ubuntu20.iso on a virtual box which will be our Nagios Client to monitor remotely. Type the following commands on the terminal of the Nagios Client.

The screenshot shows the Nagios XI configuration interface. A red arrow points from the top navigation bar's 'Configuration' dropdown to the 'Configuration Wizards' option in the sub-menu. Another red arrow points to the 'Amazon EC2' section, which is described as 'Monitor an Amazon EC2 Instance.'

The screenshot shows the AWS EC2 dashboard. A red arrow points to the 'Launch instance' button in the center of the main content area.

The screenshot shows the AWS Quick Start page for selecting an Amazon Machine Image (AMI). A red arrow points to the 'Ubuntu' AMI card, which is highlighted. Below it, the 'Ubuntu Server 22.04 LTS (HVM), SSD Volume Type' AMI details are shown.

Create key pair

Key pairs allow you to connect to your instance securely.

Enter the name of the key pair below. When prompted, store the private key in a secure and accessible location on your computer. You will need it later to connect to your instance. [Learn more](#)

Key pair name

The name can include up to 255 ASCII characters. It can't include leading or trailing spaces.

Key pair type

RSA RSA encrypted private and public key pair

ED25519 ED25519 encrypted private and public key pair (Not supported for Windows instances)

Private key file format

.pem For use with OpenSSH

Feedback Looking for language selection? Find it in the new [Unified Settings](#)

© 2022, Amazon Internet Services Private Ltd. or its affiliates. Privacy Terms Cookie preferences

Key pair (login) Info

You can use a key pair to securely connect to your instance. Ensure that you have access to the selected key pair before you launch the instance.

Key pair name - required

[Create new key pair](#)

Summary

Number of instances [Info](#)

Software Image (AMI)

Canonical, Ubuntu, 22.04 LTS, ...[read more](#)
ami-08c40ec9ead489470

Virtual server type (instance type)

t2.micro

Firewall (security group)

.. ..

[Cancel](#) [Launch instance](#)

Success

Successfully initiated launch of instance (i-01d6e5d4128f7cb66)

[Launch log](#)

Next Steps

Create billing and free tier usage alerts
To manage costs and avoid surprise bills, set up email notifications for billing and free tier usage thresholds

Connect to your instance
Once your instance is running, log into it from your local computer.
[Connect to instance](#)

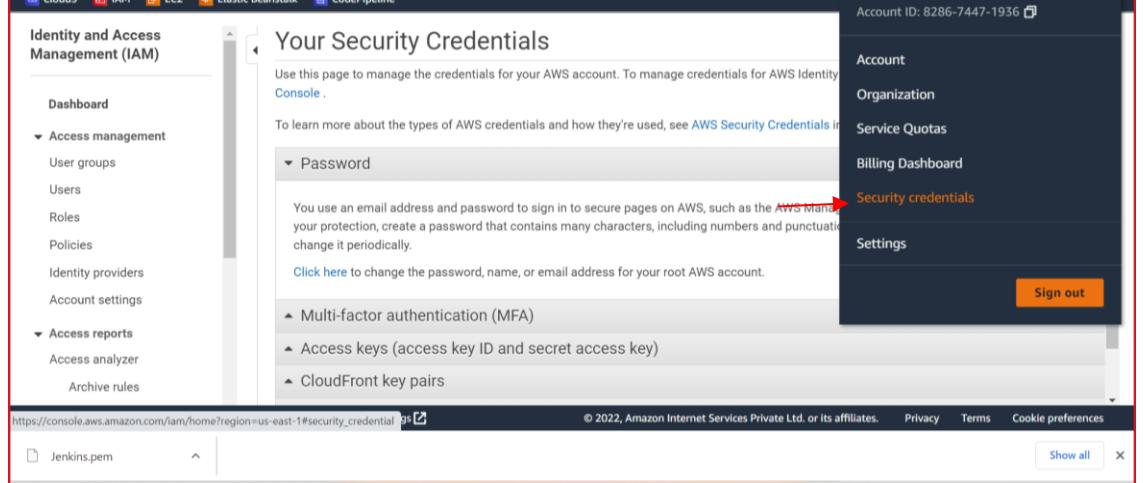
Connect an RDS database [New](#)
Configure the connection between an EC2 instance and a database to allow traffic flow between them.

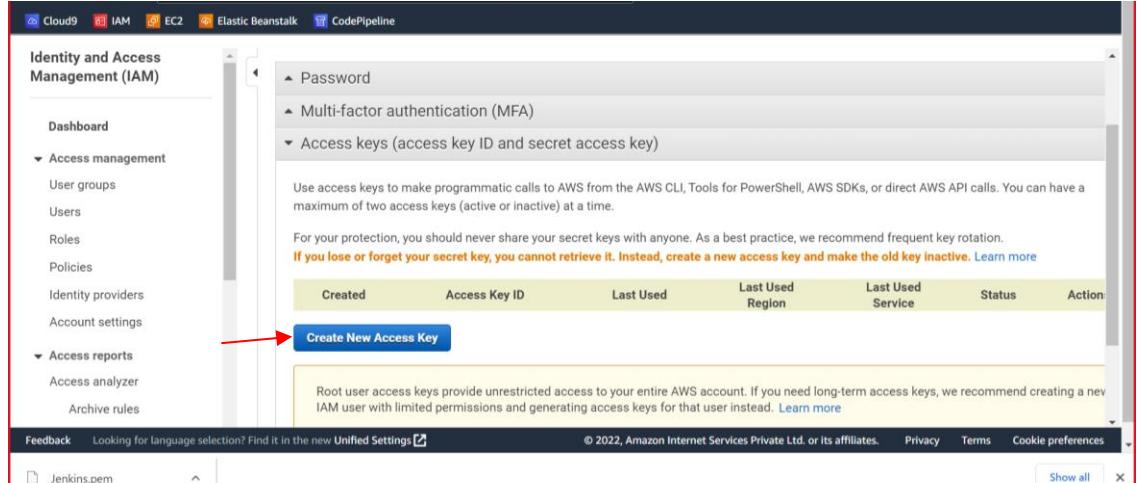
Feedback Looking for language selection? Find it in the new [Unified Settings](#)

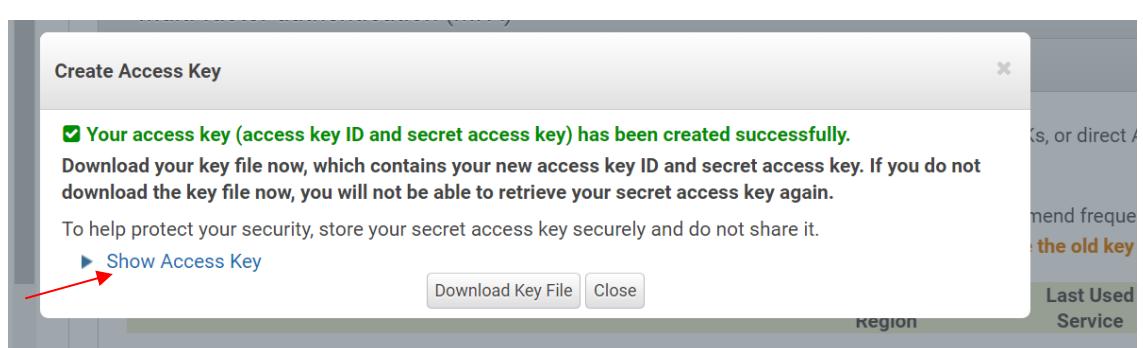
© 2022, Amazon Internet Services Private Ltd. or its affiliates. Privacy Terms Cookie preferences

Jenkins.pem

The screenshot shows three sequential steps for creating AWS IAM access keys:

- Step 1: Your Security Credentials**


This page allows managing AWS credentials. It includes sections for changing the password, enabling MFA, and managing access keys.
- Step 2: Create New Access Key**


This step shows the creation of a new access key. The dialog box contains instructions and a prominent blue 'Create New Access Key' button.
- Step 3: Access Key Created**


The final step confirms the successful creation of the access key. It provides a link to view the key details and includes download and close buttons.

Configuration Wizard: Amazon EC2 - Step 1

AWS Credentials

Specify the credentials for the EC2 instance you would like to monitor.

Access Key ID: AKIA4B4GY7AAOZNLXER
 The Access Key ID of the instance to be monitored.

Secret Access Key: VvYFT1LV1/0uYatWJaz0Tdst1vh82o66KRQe/50Z
 The Secret Access Key of the instance to be monitored.

EC2 Instances

We don't know which Instances are available yet! Fill out the EC2 Account Information section, then click "Get Available Instances".

Get Available Instances

EC2 Instances

Select the EC2 instances you would like to monitor.

us-east-1

Instance ID	IP Address
i-01d6e5d4128f7cb66	3.83.131.123

Next >

Nagios XI

Home Views Dashboards Reports Configure Tools Help Admin

Upgrade to a licensed version of Nagios XI and get support and upgrade benefits.

Configure Configuration Options

Configuration Tools Configuration Wizards Auto-Discovery Manage Templates

Auto Deployment Deploy Agent Manage Deployed Agents Deployment Settings

Advanced Configuration Core Config Manager

More Options My Account Settings System Configuration User Management Unconfigured Objects Deadpool Settings

Configuration Wizard: Amazon EC2 - Step 2

EC2 Instance Metrics

Instance ID	IP Address	Region
i-01d6e5d4128f7cb66	3.83.131.123	us-east-1

Configuration Wizard: Amazon EC2 - Step 3

Monitoring Settings

Under normal circumstances: Monitor the host and service(s) every minutes.

When a potential problem is first detected: Re-check the host and service(s) every minutes up to times before sending a notification.

Amazon EC2 Monitoring Wizard

Configuration applied successfully.

Your configuration changes have been successfully applied and the monitoring engine was restarted.

Configuration Request Successful

Run this monitoring wizard again Run another monitoring wizard

Other Options:

- View status details for 3.83.131.123
- View the latest configuration snapshots

The screenshot shows the Nagios XI monitoring wizard interface. On the left, a sidebar lists various configuration options like Configuration Wizards, Auto-Discovery, and Advanced Configuration. The main area displays several monitoring tools: DigitalOcean, Folder Watch, Google Cloud, Linode, Linux SNMP, Linux Server, and Microsoft Azure Cloud. A red arrow points to the 'Linux Server (legacy)' option, which is described as monitoring a remote Linux server with NRPE.

This screenshot shows a virtual machine configuration interface. It lists two VMs: 'ubuntu 22.04' (Powered Off) and 'vm' (Powered Off). The 'ubuntu 22.04' VM is selected. The configuration details for this VM include:

- General**: Name: ubuntu 22.04, Operating System: Ubuntu (64-bit)
- System**: Base Memory: 4352 MB, Boot Order: Floppy, Optical, Hard Disk, Acceleration: VT-x/AMD-V, Nested Paging, KVM Paravirtualization
- Display**: Video Memory: 16 MB, Graphics Controller: VMSVGA, Remote Desktop Server: Disabled, Recording: Disabled
- Storage**: Controller: IDE, IDE Secondary Device 0: [Optical Drive] Empty, Controller: SATA, SATA Port 0: ubuntu 22.04.vdi (Normal, 15.00 GB)
- Audio**: Host Driver: Windows DirectSound, Controller: ICH AC97
- Network**: Adapter 1: Intel PRO/1000 MT Desktop (NAT)
- USB**: USB Controller: OHCI

ifconfig

Note down your Nagios Client ipaddress: 192.168.29.13

This screenshot shows the 'Configuration Wizard: Linux Server (legacy) - Step 1' page. The sidebar on the left is identical to the previous screenshot. The main form is titled 'Linux Server Information' and contains the following fields:

- IP Address:** 172.17.0.1 (with a note: 'The IP address or FQDN name of the Linux server you'd like to monitor')
- Linux Distribution:** Ubuntu (with a note: 'The Linux distribution running on the server you'd like to monitor')

At the bottom are 'Back' and 'Next >' buttons.

Installing The Agent Download the Linux NRPE agent to the /tmp directory on the Linux server you wish to monitor.

```
cd /tmp
```

```
wget https://assets.nagios.com/downloads/nagiosxi/agents/linux-nrpe-agent.tar.gz
```

Unpack the installation archive you just downloaded:

```
sudo tar xzf linux-nrpe-agent.tar.gz
```

Enter the newly created agent subdirectory:

```
cd linux-nrpe-agent
```

```
sudo -i
```

```
cd /tmp
```

```
cd linux-nrpe-agent
```

```
./fullinstall
```

The script will stop to prompt you for the IP address(es) for your monitoring server(s). You will need to type in either a) a single address, b) a list of addresses separated only by spaces, c) a subnet using CIDR notation like 10.25.0.0/16. That will configure the xinetd daemon to allow connections from those locations to the agent process.

```
ifconfig
```

```
docker0: flags=4099<UP,BROADCAST,MULTICAST> mtu 1500
      inet 172.17.0.1 netmask 255.255.0.0 broadcast 172.17.255.255
        ether 02:42:24:03:2a:66 txqueuelen 0 (Ethernet)
          RX packets 0 bytes 0 (0.0 B)
          RX errors 0 dropped 0 overruns 0 frame 0
          TX packets 0 bytes 0 (0.0 B)
          TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

enp0s3: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
      inet 10.0.2.15 netmask 255.255.255.0 broadcast 10.0.2.255
        inet6 fe80::b66e:5751:23b6:c0ca prefixlen 64 scopeid 0x20<link>
          ether 08:00:27:c7:31:9e txqueuelen 1000 (Ethernet)
            RX packets 382 bytes 464081 (464.0 KB)
            RX errors 0 dropped 0 overruns 0 frame 0
            TX packets 240 bytes 23161 (23.1 KB)
            TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING> mtu 65536
      inet 127.0.0.1 netmask 255.0.0.0
        inet6 ::1 prefixlen 128 scopeid 0x10<host>
          loop txqueuelen 1000 (Local Loopback)
            RX packets 175 bytes 16128 (16.1 KB)
            RX errors 0 dropped 0 overruns 0 frame 0
            TX packets 175 bytes 16128 (16.1 KB)
            TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0
```

```

root@manasidayete:/home/manasidayete# cd /temp
bash: cd: /temp: No such file or directory
root@manasidayete:/home/manasidayete# cd /tmp
root@manasidayete:/tmp# wget https://assets.nagios.com/downloads/nagiosxi/agents/linux-nrpe-agent.tar.gz
--2022-10-31 19:59:52-- https://assets.nagios.com/downloads/nagiosxi/agents/linux-nrpe-agent.tar.gz
Resolving assets.nagios.com (assets.nagios.com)... 45.79.49.120, 2600:3c00::f03c:92ff:fe:7:45ce
Connecting to assets.nagios.com (assets.nagios.com)|45.79.49.120|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 3588667 (3.4M) [application/x-gzip]
Saving to: 'linux-nrpe-agent.tar.gz'

linux-nrpe-agent.tar 100%[=====] 3.42M 583KB/s in 7.0s

2022-10-31 20:00:00 (502 KB/s) - 'linux-nrpe-agent.tar.gz' saved [3588667/358867]

root@manasidayete:/tmp# tar xzf linux-nrpe-agent.tar.gz
root@manasidayete:/tmp# cd
root@manasidayete:~# tar xzf linux-nrpe-agent.tar.gz
tar (child): linux-nrpe-agent.tar.gz: Cannot open: No such file or directory
tar (child): Error is not recoverable: exiting now
tar: Child returned status 2

```

```

root@manasidayete:/tmp# tar xzf linux-nrpe-agent.tar.gz
root@manasidayete:/tmp# cd
root@manasidayete:~# tar xzf linux-nrpe-agent.tar.gz
tar (child): linux-nrpe-agent.tar.gz: Cannot open: No such file or directory
tar (child): Error is not recoverable: exiting now
tar: Child returned status 2
tar: Error is not recoverable: exiting now
root@manasidayete:~# cd /tmp
root@manasidayete:/tmp# tar xzf linux-nrpe-agent.tar.gz
root@manasidayete:/tmp# cd linux-nrpe-agent
root@manasidayete:/tmp/linux-nrpe-agent# ./fullinstall
=====
Nagios Linux Agent Installer
=====

This script will install the Nagios Linux Agent by executing all necessary
sub-scripts.

IMPORTANT: This script should only be used on a clean installed system:

RedHat Enterprise, CentOS, Fedora, Cloud Linux or Oracle
openSUSE or SUSE Enterprise
Ubuntu or Debian

Do NOT use this on a system running any other distro or that
does not allow additional package installation.

```

```

#
###                                     ##
#
### If you would like to change this list, enter all IP addresses to allow, ##
#
### separated by SPACES only, and then press Enter.                         ##
### (Put the address(es) of your Nagios XI servers(s) here.)                  ##
#
###                                     ##
#
#####
# Allow from: 192.168.29.13
xinetd.service is not a native service, redirecting to systemd-sysv-install.
Executing: /lib/systemd/systemd-sysv-install enable xinetd
Subcomponents installed OK
RESULT=0

#####
###                                     ##
###      Nagios XI Linux Agent Installation Complete!    ##
###                                     ##
#####

If you experience any problems, please attach the file install.log that was just
created to any support requests.

root@nagiosclient-VirtualBox:/tmp/linux-nrpe-agent#

```

Now, go to Nagios Server and check the status of Nagios Client

PART B: [Output] : Monitor Remote Host/PC using NPRE (Nagios Remote Plugin Executor)

The screenshot displays two main sections of the Nagios XI interface:

- Host Status**: Shows the status of all hosts. There are two hosts listed: "localhost" (Up) and "nagiosclient" (Up). A summary table shows 0 Up, 0 Down, 0 Unreachable, and 0 Pending hosts. The "Service Status Summary" table shows 12 OK, 0 Warning, 0 Unknown, 3 Critical, and 25 Pending services.
- Network Status Map**: A map showing the network topology. A node labeled "nagiosclient" is highlighted with a red border. A tooltip provides detailed information about this host, including its name, alias, address, state, state duration, last status check, last state change, parent hosts (none), immediate children (0), and services (critical: 12, ok: 1).

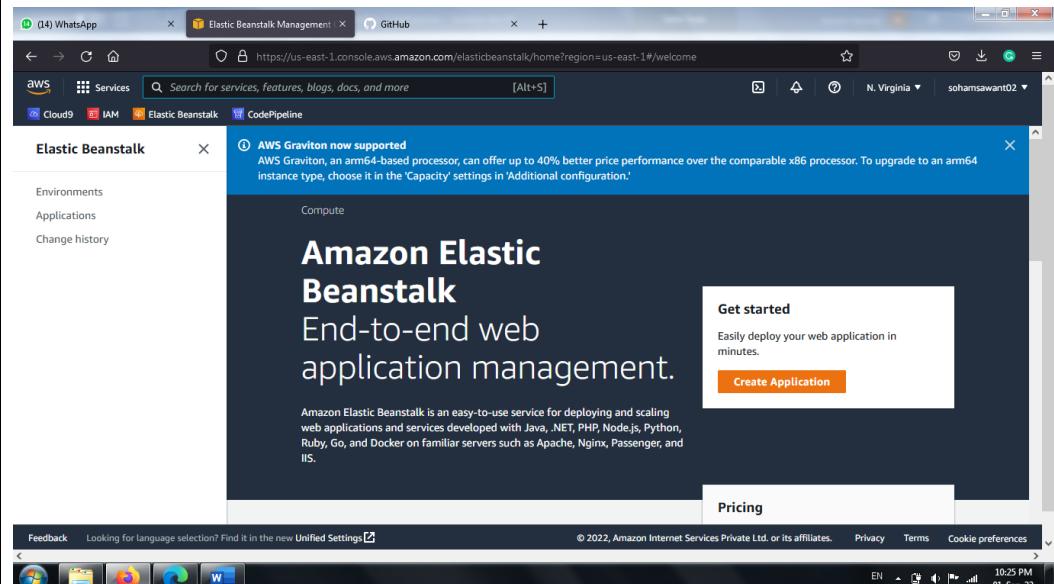
Conclusion	Successfully implemented through code how to build, change, destroy AWS infrastructure using terraform
------------	--

Semester	T.E. Semester V – Information Technology
Subject	Advance DevOps Lab
Subject Professor In-charge	Prof. Indu Anoop
Laboratory	

Student Name	Soham Anant Sawant	
Roll Number	21101B2002	
Grade and Subject Teacher's Signature		

Experiment	2	
Problem Statement	To Build Your Application using AWS CodeBuild and Deploy on S3 / SEBS using AWS CodePipeline, deploy Sample Application on EC2 instance using AWS CodeDeploy.	
Resources / Apparatus Required	Hardware: Computer System	Software: Web Browser
Details	<p>Theory: Continuous deployment allows you to deploy revisions to a production environment automatically without explicit approval from a developer, making the entire software release process automated. You will create the pipeline using AWS CodePipeline, a service that builds, tests, and deploys your code every time there is a code change. You will use your GitHub account, an Amazon Simple Storage Service (S3) bucket, or an AWS CodeCommit repository as the source location for the sample app's code. You will also use AWS Elastic Beanstalk as the deployment target for the sample app. Your completed pipeline will be able to detect changes made to the source repository containing the sample app and then automatically update your live sample app</p>	
Code	<p>Steps:</p> <p>Step1: Create a deployment environment</p> <p>Your continuous deployment pipeline will need a target environment containing virtual servers, or Amazon EC2 instances, where it will deploy sample code. You will prepare this environment before creating the pipeline. To simplify the process of setting up and configuring EC2</p>	

instances for this experiment, you will spin up a sample environment using AWS Elastic Beanstalk. Elastic Beanstalk lets you easily host web applications without needing to launch, configure, or operate virtual servers on your own. It automatically provisions and operates the infrastructure (e.g., virtual servers, load balancers, etc.) and provides the application stack (e.g., OS, language and framework, web, and application server, etc.) for you.



The screenshot shows the AWS Elastic Beanstalk Management console. The main page features a banner about AWS Graviton support and the title "Amazon Elastic Beanstalk: End-to-end web application management." Below the title, a brief description of the service is provided. A "Get started" button is visible. The left sidebar includes links for "Environments", "Applications", and "Change history".

Create Application Dialog:

The screenshot also shows a modal dialog titled "Create a web app". It contains fields for "Application name" (set to "SohamDemoWebEnv") and "Application tags". A note at the bottom states: "Apply up to 50 tags. You can use tags to group and filter your resources. A tag is a key-value pair. The key must be unique within the resource and is case-sensitive." The dialog has a "Next Step" button at the bottom right.

The screenshot displays two consecutive steps of the AWS Elastic Beanstalk 'Getting Started' wizard.

Step 1: Application tags

This step allows you to apply tags to your resources. A key-value pair is shown:

- Key: [empty]
- Value: [empty]

Buttons include "Add tag", "Remove tag", and "50 remaining".

Step 2: Platform

Platform settings are configured as follows:

- Platform: Tomcat
- Platform branch: Tomcat 8.5 with Corretto 11 running on 64bit Amazon Linux 2
- Platform version: 4.2.18 (Recommended)

Application code

Code selection options are:

- Sample application: Get started right away with sample code.
- Upload your code: Upload a source bundle from your computer or copy one from Amazon S3.

Buttons at the bottom include "Cancel", "Configure more options", and "Create application".

Elastic Beanstalk will begin creating a sample environment for you to deploy your application to. It will create an Amazon EC2 instance, a security group, an Auto Scaling group, an Amazon S3 bucket, Amazon CloudWatch alarms, and a domain name for your application. Note: This will take several minutes to complete.

The screenshot shows the AWS Elastic Beanstalk console. On the left, there's a sidebar with 'Environments', 'Applications', 'Change history', and sections for 'SohamDemoWebEnv' and 'Sohamdemowebenv-env'. The 'Sohamdemowebenv-env' section is expanded, showing 'Go to environment', 'Configuration', 'Logs', 'Health', 'Monitoring', and 'Alarms'. The main area displays the 'Sohamdemowebenv-env' environment details. It includes a 'Health' section with a green checkmark and 'Ok' status, a 'Running version' section with a 'Sample Application' button and 'Upload and deploy' button, and a 'Platform' section showing 'Tomcat 8.5 with Corretto 11 running on 64bit Amazon Linux 2/4.2.18'. A blue banner at the top says 'AWS Graviton now supported'.

The screenshot shows the 'Welcome' page for the 'Sohamdemowebenv-env' environment. The page has a large blue header with the word 'Congratulations'. Below it, a message says 'Your first AWS Elastic Beanstalk Application is now running on your own dedicated environment in the AWS Cloud'. To the right, there's a 'What's Next?' section with links to learn about building, deploying, and managing applications using AWS Elastic Beanstalk, AWS Elastic Beanstalk concepts, creating new application versions, and managing environments. There's also a 'Download the AWS Reference Application' section with a link to explore a fully-featured reference application using the AWS SDK for Java. At the bottom, there's information about the AWS Toolkit for Eclipse, including links to build and deploy applications directly from Eclipse, get started with Eclipse and AWS Elastic Beanstalk, and view all AWS Elastic Beanstalk documentation.

Step2: Get a copy of the sample code

In this step, you will retrieve a copy of the sample app's code and choose a source to host the code. The pipeline takes code from the source and then performs actions on it. You can use one of three options as your source:

- **a GitHub repository,**
- an Amazon S3 bucket, or
- an AWS CodeCommit repository.

A sample project's GitHub repository can be created in your GitHub account, or you can fork(clone) the following GitHub repository. (Ensure there is a `buildspec.yml` file in your repository for the Build Phase)

<https://github.com/sohamsawant02/CodePipeline.git>

```
MINGW64:/c/Users/lenovo/Desktop/ADEV Exp2
lenovo@lenovo-PC MINGW64 ~/Desktop/ADEV Exp2
$ git init
Initialized empty Git repository in C:/Users/lenovo/Desktop/ADEV Exp2/.git/
lenovo@lenovo-PC MINGW64 ~/Desktop/ADEV Exp2 (master)
$ git add -A

lenovo@lenovo-PC MINGW64 ~/Desktop/ADEV Exp2 (master)
$ git commit -m "Initial Commit"
[master (root-commit) 24ee0a7] Initial Commit
 2 files changed, 24 insertions(+)
 create mode 100644 buildspec.yml
 create mode 100644 index.html

lenovo@lenovo-PC MINGW64 ~/Desktop/ADEV Exp2 (master)
$ |

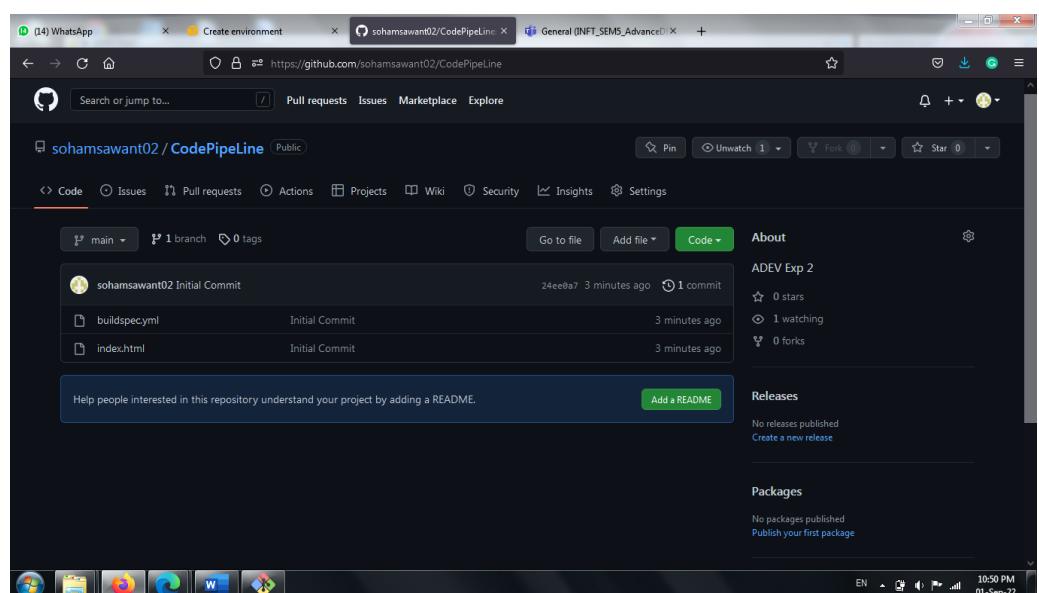
lenovo@lenovo-PC MINGW64 ~/Desktop/ADEV Exp2 (master)
$

lenovo@lenovo-PC MINGW64 ~/Desktop/ADEV Exp2 (master)
$ git branch -M main

lenovo@lenovo-PC MINGW64 ~/Desktop/ADEV Exp2 (main)
$ git remote add origin https://github.com/sohamsawant02/CodePipeline.git

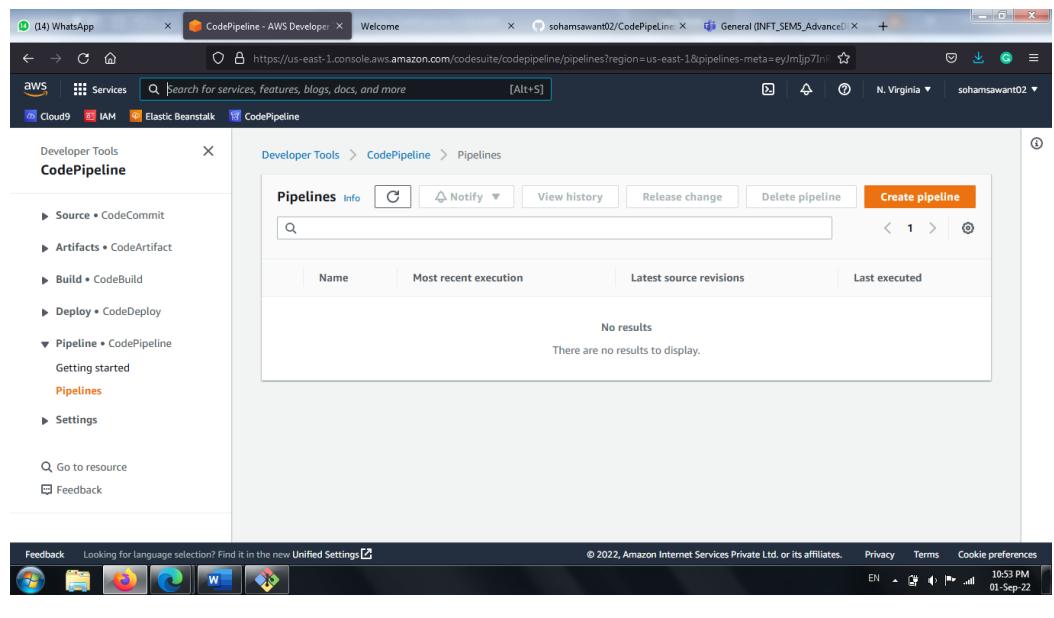
lenovo@lenovo-PC MINGW64 ~/Desktop/ADEV Exp2 (main)
$ git push -u origin main
Enumerating objects: 4, done.
Counting objects: 100% (4/4), done.
Delta compression using up to 4 threads
Compressing objects: 100% (4/4), done.
Writing objects: 100% (4/4), 568 bytes | 142.00 KiB/s, done.
Total 4 (delta 0), reused 0 (delta 0), pack-reused 0
To https://github.com/sohamsawant02/CodePipeline.git
 * [new branch]      main -> main
branch 'main' set up to track 'origin/main'.

lenovo@lenovo-PC MINGW64 ~/Desktop/ADEV Exp2 (main)
$
```



Step3: Create your Pipeline

In this step, you will create and configure a CICD pipeline. You will provide CodePipeline with the locations of your source repository [In this case GitHub repository] and deployment environment [AWS Elastic Beanstalk environment created in Step 1]. A true continuous deployment pipeline requires a build stage before deployment, where code is compiled, and unit tested. CodePipeline lets you plug your preferred build provider .[although optional, we will be using AWS CodeDeploy] into your pipeline



Choose pipeline settings Info

Pipeline settings

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

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

Type your service role name

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

Advanced settings

Artifact store
 Default location
Create a default S3 bucket in your account.

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

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

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

Cancel **Next**

Connect to Github

The screenshot shows the AWS CodePipeline interface for creating a new pipeline. The left sidebar lists various stages: Source (CodeCommit), Artifacts (CodeArtifact), Build (CodeBuild), Deploy (CodeDeploy), Pipeline (CodePipeline), and Settings. The main area is titled 'Step 2 Add source stage'. It shows the 'Source' provider dropdown set to 'GitHub (Version 2)'. Below it, a callout box provides information about the new GitHub version 2 action. At the bottom, there's a search bar for connections and a 'Connect to GitHub' button.

The screenshot shows the AWS Developer Tools interface with the 'CodePipeline' section selected. A modal window titled 'Create a connection' is open, specifically for 'Create GitHub App connection'. The 'Connection name' field contains 'greatestsoham@gmail.com'. Below it is a section for 'Tags - optional'. At the bottom right of the modal is a large orange 'Connect to GitHub' button.

Create a connection

Create GitHub App connection

Connection name: greatestsoham@gmail.com

Tags - optional

Connect to GitHub

Authorize AWS Connector for GitHub by Amazon Web Services

AWS Connector for GitHub by Amazon Web Services would like permission to:

- Verify your GitHub identity (sohamsawant02)
- Know which resources you can access
- Act on behalf of your GitHub account

Learn more about AWS Connector for GitHub

Cancel Authorize AWS Connector for GitHub

Authorizing will redirect to <https://redirect.codestar.aws>

Not owned or operated by GitHub

Created 2 years ago

Connect to GitHub

GitHub connection settings

Connection name: greatestsoham@gmail.com

GitHub Apps

Install a new app

Tags - optional

Connect

Add source stage Info

Source

Choose pipeline settings

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

New GitHub version 2 (app-based) action
To add a GitHub version 2 action in CodePipeline, you create a connection, which uses GitHub Apps to access your repository. Use the options below to choose an existing connection or create a new one. [Learn more](#)

Connection
Choose an existing connection that you have already configured, or create a new one and then return to this task.

or

Ready to connect
Your GitHub connection is ready for use. It can be used with this stage.

Repository name
Choose a repository in your GitHub account.

Branch name
Choose a branch of the repository.

Change detection options

Start the pipeline on source code change
Automatically starts your pipeline when a change occurs in the source code. If turned off, your pipeline only runs if you start it manually or on a schedule.

Output artifact format
Choose the output artifact format.

CodePipeline default
AWS CodePipeline uses the default zip format for artifacts in the pipeline. Does not include git metadata about the repository.

Full clone
AWS CodePipeline passes metadata about the repository that allows subsequent actions to do a full git clone. Only supported for AWS CodeBuild actions.

Create build project

Project configuration

Project name

DemoCodeBuild

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

Description - optional

DemoCodeBuild



Enable concurrent build limit - optional

Limit the number of allowed concurrent builds for this project.

Restrict number of concurrent builds this project can start

► Additional configuration tags

Environment

Environment image

Managed image

Use an image managed by AWS CodeBuild

Custom image

Specify a Docker image

Operating system

Ubuntu

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

Runtime(s)

Standard

Image

aws/codebuild/standard:4.0

Image version

Always use the latest image for this runtime version

Environment type

Linux

Privileged

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

Service role

New service role

Create a service role in your account

Existing service role

Choose an existing service role from your account

Role name

codebuild-DemoCodeBuild-service-role

codebuild-DemoCodeBuild-service-role

Type your service role name

Additional configuration

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

Buildspec

Build specifications

- Use a buildspec file
Store build commands in a YAML-formatted buildspec file
- Insert build commands
Store build commands as build project configuration

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

Batch configuration

You can run a group of builds as a single execution. Batch configuration is also available in advanced option when starting build.

Define batch configuration - optional
You can also define or override batch configuration when starting a build batch.

Logs

WhatsApp X CodePipeline - AWS Developer X Welcome X sohamswant02/CodePipeline X General (INFT_SEMS_Advanced) X + N. Virginia sohamswant02

Developer Tools X

CodePipeline

- Source • CodeCommit
- Artifacts • CodeArtifact
- Build • CodeBuild
- Deploy • CodeDeploy
- Pipeline • CodePipeline
 - Getting started
 - Pipelines
 - Settings
- Q. Go to resource
- Feedback

Add build stage Info

Build - optional

Build provider
This is the tool of your build project. Provide build artifact details like operating system, build spec file, and output file names.
AWS CodeBuild

Region
US East (N. Virginia)

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

Successfully created DemoCodeBuild in CodeBuild.

Environment variables - optional
Choose the key value, and type for your CodeBuild environment variables. In the value field, you can reference variables generated by CodePipeline. Learn more

Add environment variable

Single build
Triggers a single build.

Batch build
Triggers multiple builds as a single execution.

Feedback Looking for language selection? Find it in the new Unified Settings Info

© 2022, Amazon Internet Services Private Ltd. or its affiliates. Privacy Terms Cookie preferences EN 11:11 PM 01-Sep-22

The screenshot shows the AWS CodePipeline console interface. On the left, a sidebar menu lists various pipeline stages: Source, Artifacts, Build, Deploy, Pipeline, Getting started, Pipelines, and Settings. The 'Pipeline' section is expanded, showing 'CodePipeline' as the selected stage.

The main content area displays the 'Add deploy stage' step. It includes a 'Deploy - optional' section where the 'Deploy provider' is set to 'AWS Elastic Beanstalk' and the 'Region' is set to 'US East (N. Virginia)'. Below this, the 'Application name' field contains 'SohamDemoWebEnv'. A note indicates that this application must already exist in the AWS Elastic Beanstalk console.

At the bottom right of the step, there are 'Cancel', 'Previous', 'Skip deploy stage', and 'Next' buttons. The 'Next' button is highlighted in orange.

Step 4: Add deploy stage

Deploy action provider

Deploy action provider: AWS Elastic Beanstalk
ApplicationName: SohamDemoWebEnv
EnvironmentName: Sohamdemowebenv-env

Cancel Previous Create pipeline

The screenshot shows the AWS CodePipeline console. A green banner at the top says "Success" and "Congratulations! The pipeline SohamCodePipeline has been created." Below it, a "Details" section shows a step named "Source: Initial Commit" with a status of "24ee8a77". An arrow points down to a "Deploy" step, which is also labeled "Succeeded" and has a Pipeline execution ID of "69856b43-c253-4c25-86ee-ba237ac90fac". This step is associated with "AWS Elastic Beanstalk" and has a status of "Succeeded - Just now". On the right side, there are three green checkmarks indicating successful stages.

The screenshot shows the AWS Elastic Beanstalk console. It displays the "Application 'SohamDemoWebEnv' environments" page. A table lists one environment: "Sohamdemowebenv-env". The URL for this environment is highlighted with a red box and is "sohamdemowebenv-env.eba-mebb8gmv.us-east-1.elasticbeanstalk.com".

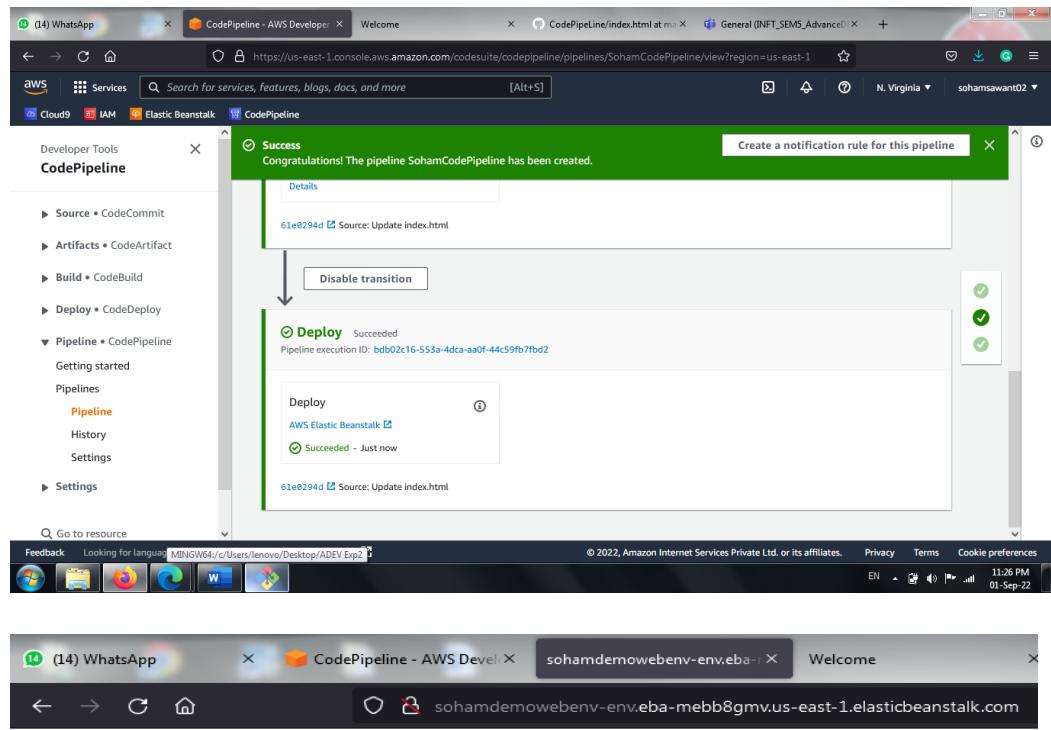
Environment name	Health	Date created	Last modified	URL	Running versions
Sohamdemowebenv-env	Ok	2022-09-01 22:36:48 UTC+0530	2022-09-01 23:21:16 UTC+0530	sohamdemowebenv-env.eba-mebb8gmv.us-east-1.elasticbeanstalk.com	code-pipeline-1662054656686-BuildArtifact-9a2ad35-6557-4cd4-ac5f-125dbff3ee

You have successfully created an automated software release pipeline using AWS CodePipeline! Using CodePipeline, you created a pipeline that uses GitHub as the source location for application code and then deploys the code to an Amazon EC2 instance managed by AWS Elastic Beanstalk. Click on the Environment URL to view application.

The screenshot shows a web browser displaying the application's home page. The URL in the address bar is "sohamdemowebenv-env.eba-mebb8gmv.us-east-1.elasticbeanstalk.com". The page content reads: "Welcome to demo version2 of the home page Our application to show Soham's code pipeline is working (Integration of Exp1 and Exp2: Prof Indu Anoop)".

Step 5: Commit a change to repository and view update of webpage

In this step, you will revise the sample code and commit the change to your GitHub repository. CodePipeline will detect your updated sample code and then automatically initiate deploying it to your EC2 instance via Elastic Beanstalk



Step 6: Clean up your resources

To avoid future charges, you will delete all the resources you launched throughout this tutorial, which includes the pipeline, the Elastic Beanstalk application, and the source you set up to host the code.

- First, you will delete your pipeline:
 - In the pipeline view, click Edit.
 - Click Delete.
 - Type in the name of your pipeline and click Delete.
- Second, delete your Elastic Beanstalk application:
 - Visit the Elastic Beanstalk console.
 - Click Actions.
 - Then click Terminate Environment

Conclusion	Successfully created an automated software release pipeline using AWS CodePipeline. Using CodePipeline, you created a pipeline that uses GitHub as the source location for application code and then deployed the code to an Amazon EC2 instance managed by AWS Elastic Beanstalk. The pipeline will automatically deploy the code every time there is a code change that is committed to the repository.
------------	---

Semester	T.E. Semester V – Information Technology
Subject	Advance DevOps Lab
Subject Professor In-charge	Prof. Indu Anoop
Laboratory	

Student Name	Soham Anant Sawant	
Roll Number	21101B2002	
Grade and Subject Teacher's Signature		

Experiment	11 and 12	
Problem Statement	AWS Lambda	
Resources / Apparatus Required	Hardware: Computer System	Software: Web Browser
Details	<p>Reference: https://docs.aws.amazon.com/lambda/latest/dg/with-s3-example.html</p> <p>Step 1: Create IAM Role: Create role with following attached permissions:</p> <ul style="list-style-type: none"> • AWSLambdaFullAccess • AmazonS3FullAccess • CloudWatchFullAccess <p>Step 2: Create a bucket in AWS S3 to upload image. Ensure region of bucket is same as that of lambda function.</p> <p>Step 3: Create a Lambda Function using node.js Blueprint Template for uploading image to s3 Bucket.</p> <p>Step 4: Create a trigger to invoke creation of logs in Cloud Watch when an image is uploaded in the specified bucket. Test the code and view the logs in Cloud Watch.</p>	

Screenshot of the AWS IAM Dashboard:

Identity and Access Management (IAM)

IAM dashboard

Security recommendations

- Add MFA for root user**: Add MFA for root user - Enable multi-factor authentication (MFA) for the root user to improve security for this account.
- Root user has no active access keys**: Using access keys attached to an IAM user instead of the root user improves security.

IAM resources

User groups	Users	Roles	Policies	Identity providers
0	0	3	0	0

What's new

AWS Account

Account ID: 828674471936 **Create**
Sign-in URL for IAM user account: https://828674471936.ws.amazon.com/console

Quick Links

My security credentials: Manage your access key factor authentication (AWS KMS)

© 2022, Amazon Internet Services Private Ltd. or its affiliates. Privacy Terms Cookie preferences

Screenshot of the AWS IAM Roles page:

Identity and Access Management (IAM)

IAM > Roles

Roles (3) Info

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.

<input type="checkbox"/>	Role name	Trusted entities	Last activi...
<input type="checkbox"/>	AWSServiceRoleForAWSCloud9 ✖ Deletion failed.	AWS Service: cloud9 (Service-Linked Role)	24 days ago
<input type="checkbox"/>	AWSServiceRoleForSupport	AWS Service: support (Service-Linked Role)	-
<input type="checkbox"/>	AWSServiceRoleForTrustedAdvisor	AWS Service: trustedadvisor (Service-Linked Role)	-

Roles Anywhere Info

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

Create role

Feedback Looking for language selection? Find it in the new [Unified Settings](#)

© 2022, Amazon Internet Services Private Ltd. or its affiliates. Privacy Terms Cookie preferences

Cloud9 IAM EC2 Elastic Beanstalk CodePipeline

Add permissions

Step 3 Name, review, and create

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.

Feedback Looking for language selection? Find it in the new Unified Settings © 2022, Amazon Internet Services Private Ltd. or its affiliates. Privacy Terms Cookie preferences

Cloud9 IAM EC2 Elastic Beanstalk CodePipeline

Add permissions

Step 3 Name, review, and create

Permissions policies (Selected 1/777) Info

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

Filter policies by property or policy name and press enter. 1 match < 1 >

"lambda" X "full" X Clear filters

Policy name	Type	Description
AWSLambda_FullAc...	AWS ma...	Grants full access to AWS Lambda service, AWS Lambda con...

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

Feedback Looking for language selection? Find it in the new Unified Settings © 2022, Amazon Internet Services Private Ltd. or its affiliates. Privacy Terms Cookie preferences

Screenshot of the AWS IAM Role creation process, Step 2: Add permissions.

Permissions policies (Selected 2/777) Info

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

Filter policies by property or policy name and press enter.

2 matches | 1 | Clear filters

Policy name	Type	Description
<input checked="" type="checkbox"/> AmazonS3FullAccess	AWS ma...	Provides full access to all buckets via the AWS Management ...
<input type="checkbox"/> AmazonS3Outposts...	AWS ma...	Provides full access to Amazon S3 on Outposts via the AWS ...

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.

Feedback Looking for language selection? Find it in the new Unified Settings

© 2022, Amazon Internet Services Private Ltd. or its affiliates. Privacy Terms Cookie preferences

Screenshot of the AWS IAM Role creation process, 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. {

Feedback Looking for language selection? Find it in the new Unified Settings

27°C Smoke © 2022, Amazon Internet Services Private Ltd. or its affiliates. Privacy Terms Cookie preferences

ENG IN 23:47 01-11-2022

Cloud9 IAM EC2 Elastic Beanstalk CodePipeline

Identity and Access Management (IAM)

Role AWSLambdaDemoRole created.

IAM > Roles

Roles (4) Info

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 activi...
AWSLambdaDemoRole	AWS Service: lambda	
AWSServiceRoleForAWSCloud9	AWS Service: cloud9 (Service-Linked Role)	24 days ago
AWSServiceRoleForSupport	AWS Service: support (Service-Linked Role)	
AWSServiceRoleForTrustedAdvisor	AWS Service: trustedadvisor (Service-Linked Role)	

Search IAM

Dashboard

Access management

- User groups
- Users
- Roles**
- Policies
- Identity providers
- Account settings

Access reports

Create role

Cloud9 IAM EC2 Elastic Beanstalk S3

Create bucket Info

Buckets are containers for data stored in S3. Learn more Link

General configuration

Bucket name

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

AWS Region

US East (N. Virginia) us-east-1

Copy settings from existing bucket - *optional*
Only the bucket settings in the following configuration are copied.

Choose bucket

Object Ownership Info

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

Feedback Looking for language selection? Find it in the new Unified Settings Link

© 2022, Amazon Internet Services Private Ltd. or its affiliates. Privacy Terms Cookie preferences

S Cloud9 IAM EC2 Elastic Beanstalk CodePipeline S3

☰

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.

Feedback Looking for language selection? Find it in the new Unified Settings © 2022, Amazon Internet Services Private Ltd. or its affiliates. Privacy Terms Cookie preferences

(28) WhatsApp Exp11_12_AWSLambda_Refenc... S3 Management Console

s3.console.aws.amazon.com/s3/buckets?region=us-east-1

aws Services Search for services, features, blogs, docs, and more [Alt+S]

Cloud9 IAM EC2 Elastic Beanstalk CodePipeline S3

Amazon S3

Buckets

- Access Points
- Object Lambda Access Points
- Multi-Region Access Points
- Batch Operations
- Access analyzer for S3

Block Public Access settings for this account

▼ Storage Lens

- Dashboards
- AWS Organizations settings

Successfully created bucket "demobucket". 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 (7) Info

Buckets are containers for data stored in S3. Learn more

Copy ARN Empty Delete

Find buckets by name

Name	AWS Region	Access	Creation
aws-rondestar-us-east-1-R2R674471936	US East (N. Virginia) us-east-1	C	October 2022

Feedback Looking for language selection? Find it in the new Unified Settings © 2022, Amazon Internet Services Private Ltd. or its affiliates. Privacy Terms Cookie preferences

S Cloud9 IAM EC2 Elastic Beanstalk CodePipeline S3 Lambda

AWS Lambda

Lambda > Functions

Functions (0) Last fetched now Actions Create function

Filter by tags and attributes or search by keyword

Function name	Description	Package type	Runtime	Last modified
There is no data to display.				

Dashboard Applications Functions Additional resources Code signing configurations Layers Replicas Related AWS resources Step Functions state machines

© 2022, Amazon Internet Services Private Ltd. or its affiliates. Privacy Terms Cookie preferences

This screenshot shows the AWS Lambda Functions page. The left sidebar contains links for Dashboard, Applications, Functions (which is selected), Additional resources, and Related AWS resources. The main content area shows a table titled 'Functions (0)' with columns for Function name, Description, Package type, Runtime, and Last modified. A message at the bottom of the table says 'There is no data to display.' The top navigation bar includes links for Cloud9, IAM, EC2, Elastic Beanstalk, CodePipeline, S3, and Lambda.

Cloud9 IAM EC2 Elastic Beanstalk CodePipeline S3 Lambda

Create function

AWS Serverless Application Repository applications have moved to Create application.

Author from scratch Start with a simple Hello World example.

Use a blueprint Build a Lambda application from sample code and configuration presets for common use cases.

Container image Select a container image to deploy for your function.

Blueprints (1/49) Info Last fetched 29 seconds ago Export

Filter blueprints Blueprint attributes = s3 Clear filters

Blueprints (1/49) Info Last fetched 29 seconds ago Export

Blueprint attributes = s3 Clear filters

Get S3 object An Amazon S3 trigger that retrieves metadata for the object that has been updated. python3.7 · s3

Use Rekognition to detect faces An Amazon S3 trigger that usesrekognition APIs to detect faces python · rekognition · s3

Get S3 object An Amazon S3 trigger that retrieves metadata for the object that has been updated. nodejs · s3

Feedback Looking for language selection? Find it in the new Unified Settings © 2022, Amazon Internet Services Private Ltd. or its affiliates. Privacy Terms Cookie preferences

This screenshot shows the 'Create function' page. It features three main options: 'Author from scratch', 'Use a blueprint' (which is selected and highlighted in blue), and 'Container image'. Below these are sections for 'Blueprints (1/49)' and a list of three available blueprints: 'Get S3 object', 'Use Rekognition to detect faces', and 'Get S3 object'. Each blueprint entry includes a description, a language icon, and a file name. At the bottom, there's a feedback link and copyright information.

Cloud9 IAM EC2 Elastic Beanstalk CodePipeline S3 Lambda

Lambda > Functions > Create function > Configure blueprint s3-get-object

Basic information [Info](#)

Function name: demolambdafunctions3

Execution role:

Choose a role that defines the permissions of your function. To create a custom role, go to the IAM console.

Create a new role with basic Lambda permissions
 Use an existing role
 Create a new role from AWS policy templates

Existing role:

Choose an existing role that you've created to be used with this Lambda function. The role must have permission to upload logs to Amazon CloudWatch Logs.

AWSLambdaDemoRole [View the AWSLambdaDemoRoleManasi role on the IAM console.](#)

Feedback Looking for language selection? Find it in the new [Unified Settings](#).

© 2022, Amazon Internet Services Private Ltd. or its affiliates. [Privacy](#) [Terms](#) [Cookie preferences](#)

Cloud9 IAM EC2 Elastic Beanstalk CodePipeline S3 Lambda

Bucket

Please select the S3 bucket that serves as the event source. The bucket must be in the same region as the function.

s3/demobucket [X](#) [C](#)

Bucket region: us-east-1

Event type

Select the events that you want to trigger the Lambda function. You can optionally set up a prefix or suffix for an event. However, for each bucket, individual events cannot have multiple configurations with overlapping prefixes or suffixes that could match the same object key.

All object create events [▼](#)

Prefix - optional

Enter a single optional prefix to limit the notifications to objects with keys that start with matching characters.

e.g. images/

Suffix - optional

Enter a single optional suffix to limit the notifications to objects with keys that end with matching characters.

e.g. .jpg

Recursive invocation

If your function writes objects to an S3 bucket, ensure that you are using different S3 buckets for input and output. Writing to the same bucket increases the risk of creating a recursive invocation, which can result in increased Lambda usage and increased costs. [Learn more](#)

I acknowledge that using the same S3 bucket for both input and output is not recommended.

Feedback Looking for language selection? Find it in the new [Unified Settings](#).

© 2022, Amazon Internet Services Private Ltd. or its affiliates. [Privacy](#) [Terms](#) [Cookie preferences](#)

Cloud9 IAM EC2 Elastic Beanstalk CodePipeline S3 Lambda

```
7
8 exports.handler = async (event, context) => {
9     //console.log('Received event:', JSON.stringify(event, null, 2));
10
11    // Get the object from the event and show its content type
12    const bucket = event.Records[0].s3.bucket.name;
13    const key = decodeURIComponent(event.Records[0].s3.object.key.replace(/\+/g, ' '));
14    const params = {
15        Bucket: bucket,
16        Key: key,
17    };
18    try {
19        const { ContentType } = await s3.getObject(params).promise();
20        console.log('CONTENT TYPE:', ContentType);
21        return ContentType;
22    } catch (err) {
23        console.log(err);
24        const message = `Error getting object ${key} from bucket ${bucket}. Make sure they e
25        console.log(message);
26        throw new Error(message);
27    }
28};
29
```

Cancel Create function

Feedback Looking for language selection? Find it in the new Unified Settings [?](#)

© 2022, Amazon Internet Services Private Ltd. or its affiliates. Privacy Terms Cookie preferences

Cloud9 IAM EC2 Elastic Beanstalk CodePipeline S3 Lambda

Lambda > Functions > manasidemolambdafunctions3

idemolambdafunctions3

Throttle Copy ARN Actions ▾

Congratulations! Your Lambda function "manasidemolambdafunctions3" has been successfully created and configured with manasidemobucket as a trigger. Choose Test to input a test event and test your function.

Function overview Info

 idemolambdafunctions3

 S3 Layers (0)

+ Add destination

Description
An Amazon S3 trigger that retrieves metadata for the object that has been updated.

Last modified
13 seconds ago

Function ARN

The screenshot shows the AWS Lambda console's General configuration page. On the left is a sidebar with links: Code, Test, Monitor, Configuration (which is selected), Aliases, and Versions. The main area has a title "General configuration" with an "Edit" button. It contains the following fields:

Description	Memory	Ephemeral storage
An Amazon S3 trigger that retrieves metadata for the object that has been updated.	128 MB	512 MB
Timeout	0 min 3 sec	

A modal window titled "AWS Compute Optimizer" is open, prompting to "Opt in to see memory recommendations for your Lambda functions. View details".

At the bottom, there are footer links: Feedback, Unified Settings, © 2022, Privacy, Terms, and Cookie preferences.

The screenshot shows the AWS Lambda console's General configuration page. The sidebar shows the same navigation as the previous screenshot. The main area has a title "General configuration" with an "Edit" button. It contains the following fields:

Set memory to between 128 MB and 10240 MB

Ephemeral storage Info
You can configure up to 10 GB of ephemeral storage (/tmp) for your function. View pricing

512 MB

Set ephemeral storage (/tmp) to between 512 MB and 10240 MB.

Timeout
0 min 30 sec

Execution role
Choose a role that defines the permissions of your function. To create a custom role, go to the IAM console.

Use an existing role
 Create a new role from AWS policy templates

Existing role
Choose an existing role that you've created to be used with this Lambda function. The role must have permission to upload logs to Amazon CloudWatch Logs.

AWSLambdaDemoRoleManasi

View the AWSLambdaDemoRoleManasi role on the IAM console.

Cancel Save

Cloud9 IAM EC2 Elastic Beanstalk CodePipeline S3 Lambda

Successfully updated the function demolambdafunctions3.

Code Test Monitor Configuration Aliases Versions

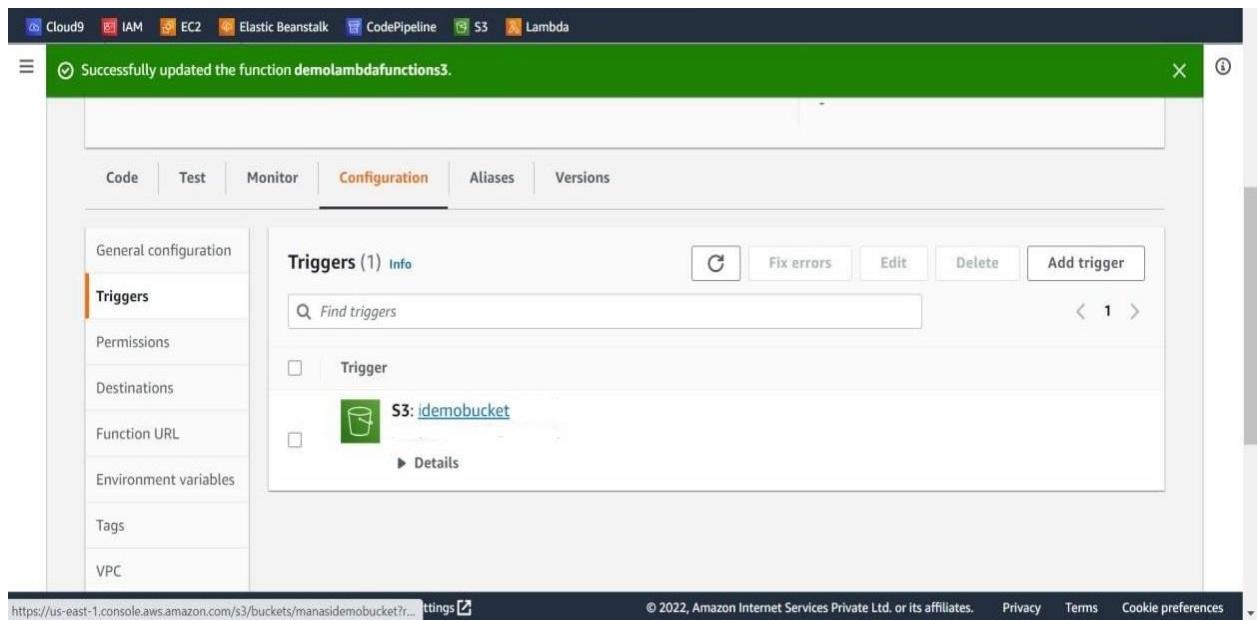
General configuration Triggers (1) Info

Find triggers Trigger S3: demobucket Details

Permissions Destinations Function URL Environment variables Tags VPC

https://us-east-1.console.aws.amazon.com/s3/buckets/manasidemobucket?...ttings [Edit](#)

© 2022, Amazon Internet Services Private Ltd. or its affiliates. Privacy Terms Cookie preferences



Cloud9 IAM EC2 Elastic Beanstalk CodePipeline S3 Lambda

Amazon S3 Buckets Amazon S3 > Buckets > manasidemobucket

demobucket 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 [here](#).

Copy S3 URI Copy URL Download Open Actions

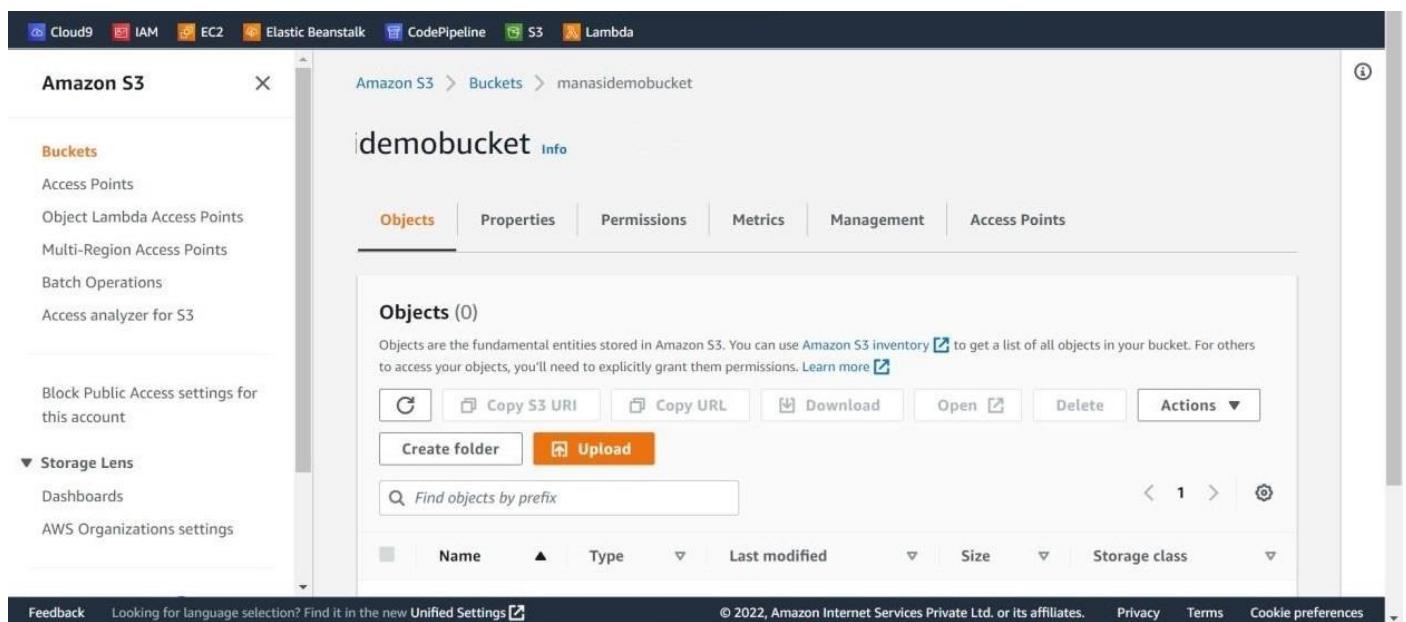
Create folder Upload

Find objects by prefix

Name	Type	Last modified	Size	Storage class
------	------	---------------	------	---------------

Feedback Looking for language selection? Find it in the new Unified Settings [Edit](#)

© 2022, Amazon Internet Services Private Ltd. or its affiliates. Privacy Terms Cookie preferences



Cloud9 IAM EC2 Elastic Beanstalk CodePipeline S3 Lambda

Amazon S3 > Buckets > manasidemobucket > Upload

Upload Info

Add the files and folders you want to upload to S3. To upload a file larger than 160GB, use the AWS CLI, AWS SDK or Amazon S3 REST API. Learn more [↗](#)

Drag and drop files and folders you want to upload here, or choose **Add files**, or **Add folder**.

Files and folders (0)	Remove	Add files	Add folder								
All files and folders in this table will be uploaded.											
<input type="text"/> Find by name 🔍											
<table border="1"><thead><tr><th>Name</th><th>Folder</th><th>Type</th><th>Size</th></tr></thead><tbody><tr><td colspan="4">No files or folders</td></tr></tbody></table>				Name	Folder	Type	Size	No files or folders			
Name	Folder	Type	Size								
No files or folders											

Feedback Looking for language selection? Find it in the new [Unified Settings](#) [↗](#) © 2022, Amazon Internet Services Private Ltd. or its affiliates. Privacy Terms Cookie preferences

Cloud9 IAM EC2 Elastic Beanstalk CodePipeline S3 Lambda

Upload succeeded
View details below.

Upload: status

The information below will no longer be available after you navigate away from this page.

Summary

Destination	Succeeded	Failed
s3://demobucket	1 file, 42.8 KB (100.00%)	0 files, 0 B (0%)

[Files and folders](#) [Configuration](#)

Feedback Looking for language selection? Find it in the new [Unified Settings](#) [↗](#) © 2022, Amazon Internet Services Private Ltd. or its affiliates. Privacy Terms Cookie preferences

The screenshot shows the AWS CloudWatch Logs interface. On the left, the navigation pane is open with the 'Logs' section selected. In the main area, under 'Log streams', there is one entry:

Log stream	Last event time
2022/11/01/[\$LATEST]765294ecf6c74841ab467925f2931...	2022-11-02 00:07:24 (UTC+05:30)

Below this, the 'Log events' section displays the log entries from the selected stream. One specific entry is highlighted with a blue arrow pointing to its 'Message' field:

Timestamp	Message
2022-11-02T00:07:23.481+05:30	2022-11-01T18:37:23.481Z undefined INFO Loading function
2022-11-02T00:07:23.766+05:30	START RequestId: 47561fdf-fe2f-4beb-9cbf-912cc10d8ba2 Version: \$LATEST
2022-11-02T00:07:24.392+05:30	2022-11-01T18:37:24.391Z 47561fdf-fe2f-4beb-9cbf-912cc10d8ba2 INFO CONTENT TYPE: image...
2022-11-01T18:37:24.391Z	47561fdf-fe2f-4beb-9cbf-912cc10d8ba2 INFO CONTENT TYPE: image/jpeg
2022-11-02T00:07:24.430+05:30	END RequestId: 47561fdf-fe2f-4beb-9cbf-912cc10d8ba2
2022-11-02T00:07:24.430+05:30	REPORT RequestId: 47561fdf-fe2f-4beb-9cbf-912cc10d8ba2 Duration: 664.03 ms Billed Dura...

The screenshot shows the AWS CloudWatch Logs interface, specifically the 'Log events' section. The same log stream and timestamp range as the previous screenshot are shown. A blue arrow points to a 'Copy' button located next to the last log entry's message:

Timestamp	Message
2022-11-02T00:07:23.481+05:30	2022-11-01T18:37:23.481Z undefined INFO Loading function
2022-11-02T00:07:23.766+05:30	START RequestId: 47561fdf-fe2f-4beb-9cbf-912cc10d8ba2 Version: \$LATEST
2022-11-02T00:07:24.392+05:30	2022-11-01T18:37:24.391Z 47561fdf-fe2f-4beb-9cbf-912cc10d8ba2 INFO CONTENT TYPE: image...
2022-11-01T18:37:24.391Z	47561fdf-fe2f-4beb-9cbf-912cc10d8ba2 INFO CONTENT TYPE: image/jpeg
2022-11-02T00:07:24.430+05:30	END RequestId: 47561fdf-fe2f-4beb-9cbf-912cc10d8ba2
2022-11-02T00:07:24.430+05:30	REPORT RequestId: 47561fdf-fe2f-4beb-9cbf-912cc10d8ba2 Duration: 664.03 ms Billed Dura...

To change the message :

```
10     "principalId": "EXAMPLE",
11 },
12 },
13 "requestParameters": {
14     "sourceIPAddress": "127.0.0.1"
15 },
16 "responseElements": {
17     "x-amz-request-id": "EXAMPLE123456789",
18     "x-amz-id-2": "EXAMPLE123/5678abcdefghijklmabdaisawesome/mnopqrstuvwxyzABCDEFGH"
19 },
20 "s3": {
21     "s3SchemaVersion": "1.0",
22     "configurationId": "testConfigRule",
23     "bucket": {
24         "name": "demobucket",
25         "ownerIdentity": {
26             "principalId": "EXAMPLE"
27         },
28         "arn": "arn:aws:s3:::demobucket"
29     },
30     "object": {
31         "key": "bhole.jpg",
32         "size": 1024,
33         "etag": "5c4e2900899516135f6ae842c5e77a4f",
34         "sequencer": "0A1B2C3D4E5F678901"
35     }
36 }
```

Feedback Looking for language selection? Find it in the new Unified Settings [?](#) © 2022, Amazon Internet Services Private Ltd. or its affiliates. Privacy Terms Cookie preferences

Cloud9 IAM EC2 Elastic Beanstalk CodePipeline S3 Lambda

The test event **idemoevent** was successfully saved.

Execution result: succeeded (logs)

Test event [Info](#)

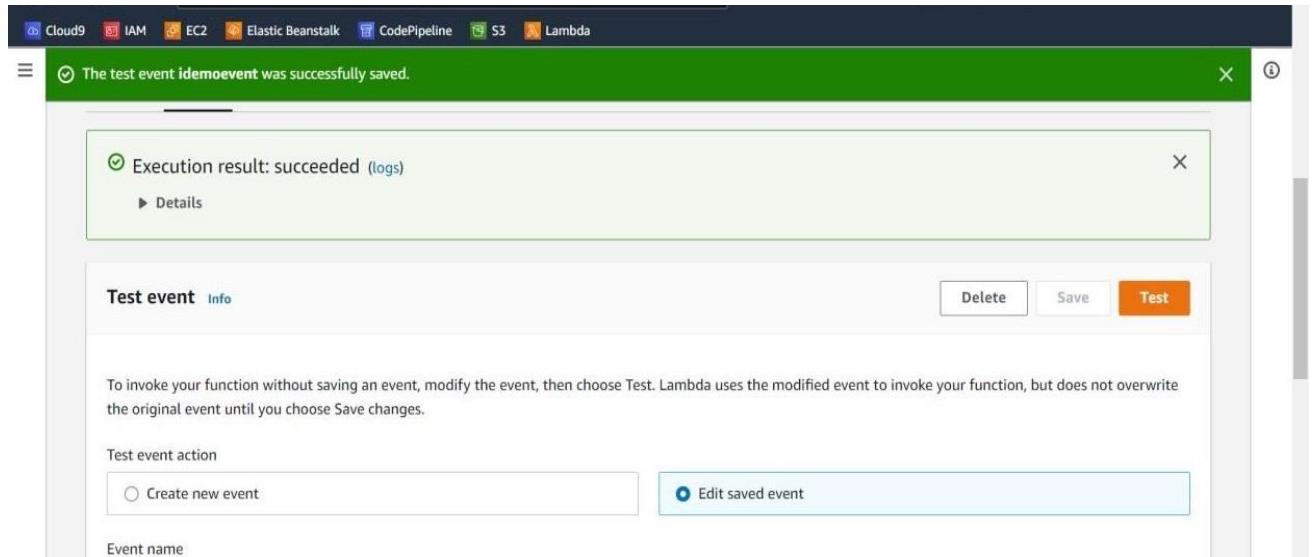
To invoke your function without saving an event, modify the event, then choose Test. Lambda uses the modified event to invoke your function, but does not overwrite the original event until you choose Save changes.

Test event action

Create new event Edit saved event

Event name

Delete Save Test



Cloud9 IAM EC2 Elastic Beanstalk CodePipeline S3 Lambda

The test event **idemoevent** was successfully saved.

+ Add trigger

on:manasidemolambdafunctions3

Function URL [Info](#)

Code Test Monitor Configuration Aliases Versions

Code source [Info](#)

Upload from

File Edit Find View Go Tools Window Test Deploy

Go to Anything (Ctrl-P)

index.js Execution result

Environment manasidemolambda

Index.js

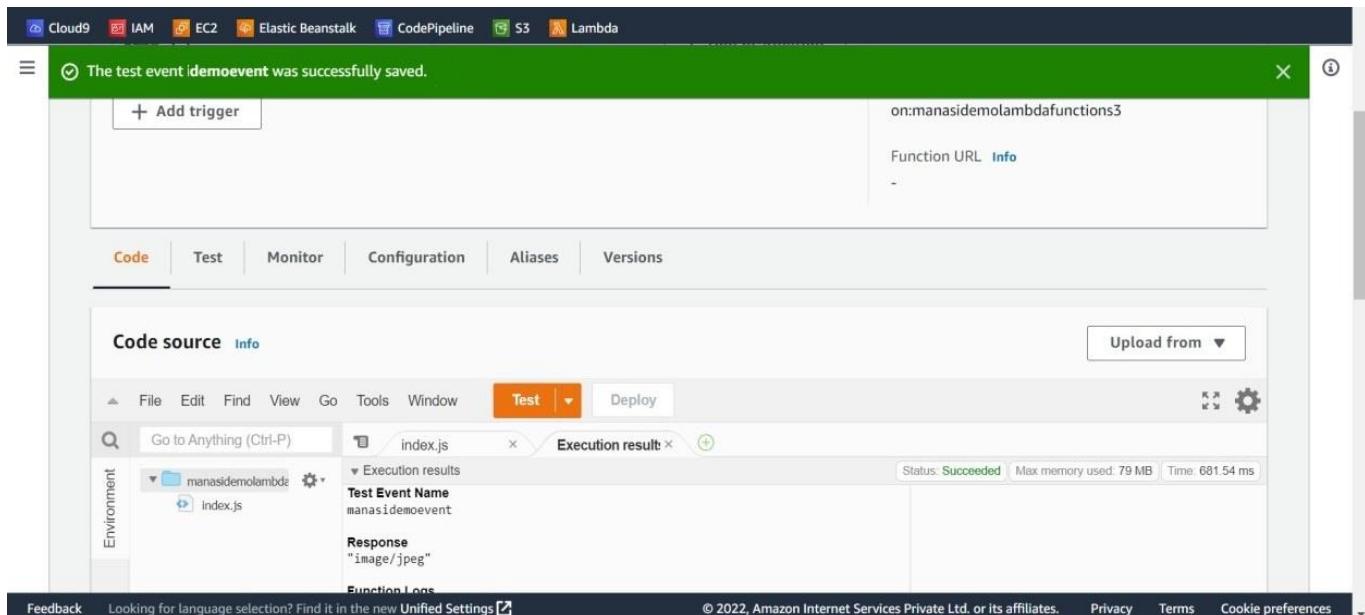
Test Event Name manasidemoevent

Response "image/jpeg"

Function Logs

Feedback Looking for language selection? Find it in the new Unified Settings [?](#)

© 2022, Amazon Internet Services Private Ltd. or its affiliates. Privacy Terms Cookie preferences



The screenshot shows the AWS Lambda function editor interface. At the top, there's a navigation bar with 'Go', 'Tools', 'Window', 'Test' (which is highlighted in orange), 'Deploy', and a status message 'Changes not deployed'. Below the navigation bar, there are tabs for 'index.js' and 'Execution results'. The 'index.js' tab is active, displaying the following code:

```
const bucket = event.Records[0].s3.bucket.name;
const key = decodeURIComponent(event.Records[0].s3.object.key.replace(/\+/g, ' '));
const params = {
  Bucket: bucket,
  Key: key,
};
try {
  const { ContentType } = await s3.getObject(params).promise();
  console.log('CONTENT TYPE:', ContentType);
  console.log('An image has been added');
  return ContentType;
} catch (err) {
  console.log(err);
  const message = `Error getting object ${key} from bucket ${bucket}. Make sure they exist`;
  console.log(message);
  throw new Error(message);
}
```

Conclusion

Successfully studied AWS Lambda.