

# Assignment-1

Aim- To make a EC2 Machine in AWS

## THEORY-

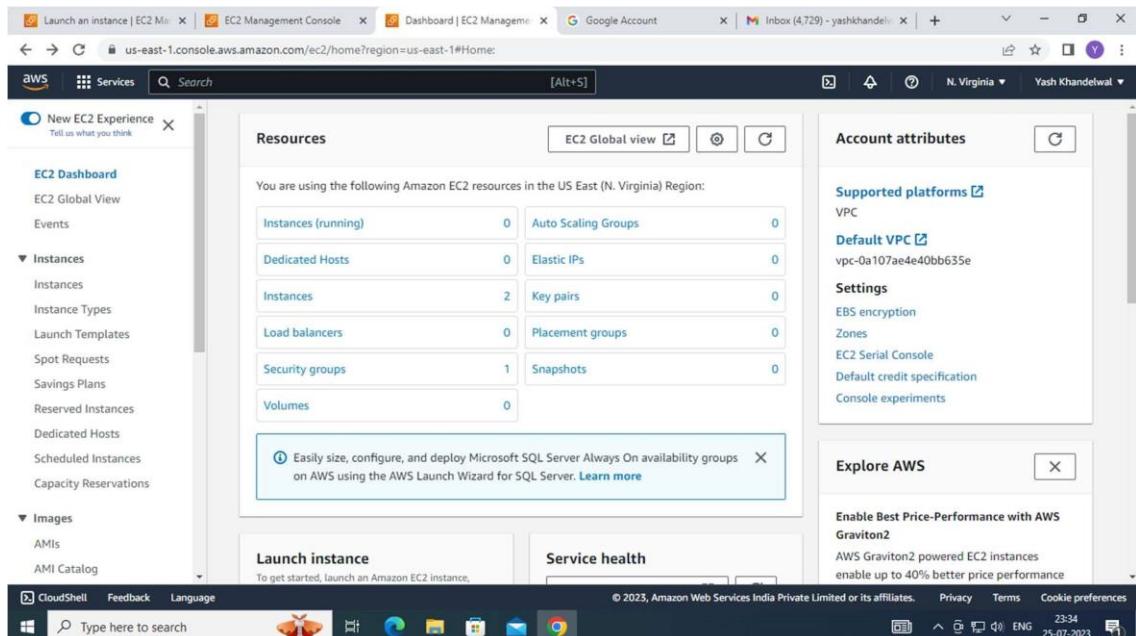
Amazon Elastic Compute Cloud (Amazon EC2) is a web service that provides secure, resizable compute capacity in the cloud. It is designed to make webscale cloud computing easier for developers.

## STEPS-

LOGIN TO AWS

ACCOUNT, THEN

SEARCH EC2 .



NOW CLICK ON LAUNCH / CREATE NEW INSTANCES.

The screenshot shows the AWS EC2 Management Console Home page. The left sidebar includes the EC2 Dashboard, Instances (with sub-options like Instances, Instance Types, Launch Templates, etc.), and Images (with sub-options like AMIs, AMI Catalog). The main content area has two main sections: 'Resources' (listing Instances (running), Auto Scaling Groups, Dedicated Hosts, Elastic IPs, Instances, Key pairs, Load balancers, Placement groups, Security groups, Snapshots, and Volumes) and 'Account attributes' (listing Supported platforms, Default VPC, Settings, and Explore AWS). A central callout box contains the text: 'Easily size, configure, and deploy Microsoft SQL Server Always On availability groups on AWS using the AWS Launch Wizard for SQL Server. Learn more'.

Select whether you want to go with or without key pair.

The screenshot shows the 'Launch instance' wizard at the 'Create key pair' step. It asks the user to select a key pair name. A note says: 'We noticed that you didn't select a key pair. If you want to be able to connect to your instance it is recommended that you create one.' Below are two radio buttons: 'Create new key pair' (selected) and 'Proceed without key pair'. A note below the second button says: 'Note that you will not be able to connect to this instance unless you already know the password built into this AMI.' At the bottom are 'Cancel' and 'Proceed without key pair' buttons.

Wait for the success conforma on.

Instance summary for i-0272635ce49afcae0 (Yash K) [Info](#)

Instance ID	i-0272635ce49afcae0 (Yash K)	Public IPv4 address	54.87.128.255   <a href="#">open address</a>	Private IP4 addresses	172.31.87.102
IPv6 address	-	Instance state	<span>Running</span>	Public IPv4 DNS	ec2-54-87-128-255.compute-1.amazonaws.com   <a href="#">open address</a>
Hostname type	IP name: ip-172-31-87-102.ec2.internal	Private IP DNS name (IPv4 only)	ip-172-31-87-102.ec2.internal	Instance type	t2.micro
IP name:	ip-172-31-87-102.ec2.internal	VPC ID	vpc-0a107ae4e40bb635e	Elastic IP addresses	-
Answer private resource DNS name	IPv4 (A)	Subnet ID	subnet-0c620e0f39f7d94b1	AWS Compute Optimizer finding	<a href="#">Opt-in to AWS Compute Optimizer for recommendations.</a>
Auto-assigned IP address	54.87.128.255 [Public IP]			Learn more	-
IAM Role	-			Auto Scaling Group name	-
AMI Catalog	IMDSv2				

Choose any machine you want to create here I am creating UBUNTU(free tier).

Click on T2 micro (free tier one)

Step 2: Choose an Instance Type

Currently selected: t2.micro (- ECUs, 1 vCPUs, 2.5 GHz, -, 1 GiB memory, EBS only)

	Family	Type	vCPUs	Memory (GiB)	Instance Storage (GB)	EBS-Optimized Available	Network Performance	IPv6 Support
<input type="checkbox"/>	t2	t2.nano	1	0.5	EBS only	-	Low to Moderate	Yes
<input checked="" type="checkbox"/>	t2	t2.micro <span>Free tier eligible</span>	1	1	EBS only	-	Low to Moderate	Yes
<input type="checkbox"/>	t2	t2.small	1	2	EBS only	-	Low to Moderate	Yes
<input type="checkbox"/>	t2	t2.medium	2	4	EBS only	-	Low to Moderate	Yes
<input type="checkbox"/>	t2	t2.large	2	8	EBS only	-	Low to Moderate	Yes
<input type="checkbox"/>	t2	t2.xlarge	4	16	EBS only	-	Moderate	Yes
<input type="checkbox"/>	t2	t2.2xlarge	8	32	EBS only	-	Moderate	Yes
<input type="checkbox"/>	t3	t3.nano	2	0.5	EBS only	Yes	Up to 5 Gigabit	Yes
<input type="checkbox"/>	t3	t3.micro	2	1	EBS only	Yes	Up to 5 Gigabit	Yes
<input type="checkbox"/>	t3	t3.small	2	2	EBS only	Yes	Up to 5 Gigabit	Yes
<input type="checkbox"/>	t3	t3.medium	2	4	EBS only	Yes	Up to 5 Gigabit	Yes

Cancel Previous [Review and Launch](#) Next: Configure Instance Details

Click on NEXT, then Again Click Next.

**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 manager, and more.

Number of instances: 1      Launch into Auto Scaling Group

Purchasing option: Request Spot instances

Network: vpc-eb864d80 (default)      Create new VPC

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

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

Placement group: Add instance to placement group

Capacity Reservation: Open

Domain join directory: No directory      Create new directory

IAM role: None      Create new IAM role

Shutdown behavior: Stop

Stop - Hibernate behavior: Enable hibernation as an additional stop behavior

THEN CREATE A KEY PAIR BY ANY NAME AND DOWNLOAD IT. THEN CLICK NEXT

Now add security group ALL TRAFFIC ,  
PROTOCOL – ALL,  
SOURCE- ANYWHERE.

**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, 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: DevServer

Description: DevServer created 2021-07-20T16:24:29.775+05:30

Type	Protocol	Port Range	Source	Description
SSH	TCP	22	Custom 0.0.0.0/0	e.g. 0.0.0.0/0
All traffic	All	0 - 65535	Anywhere 0.0.0.0/0	e.g. 0.0.0.0/0

Add Rule

**Warning**  
 Rules with source of 0.0.0.0/0 allow all IP addresses to access your instance. We recommend setting security group rules to allow access from known IP addresses only.

NOW WAIT TILL THE STATUS CHECK IS 2/2 and Instance is running.

Once Check is complete click on launch instances.

**Instance summary for i-0272635ce49afcae0 (Yash K) [Info](#)**

Instance ID	i-0272635ce49afcae0 (Yash K)	Public IPv4 address	54.87.128.255   <a href="#">open address</a>
IPv6 address	-	Instance state	Running
Hostname type	IP name: ip-172-31-87-102.ec2.internal	Private IP DNS name (IPv4 only)	ip-172-31-87-102.ec2.internal
Answer private resource DNS name	IPv4 (A)	Instance type	t2.micro
Auto-assigned IP address	54.87.128.255 [Public IP]	VPC ID	vpc-0a107ae4e40bb635e
IAM Role	-	Subnet ID	subnet-0c620e0f39f7d94b1
IMDSv2			

## FOLLOW SOME BASIC LINUX COMMANDS AS SHOWN BELOW-

```

The SIZE argument is an integer and optional unit (example: 10K is 10*1024).
Units are K,M,G,T,P,E,Z,Y (powers of 1024) or KB,MB,... (powers of 1000).
Binary prefixes can be used, too: KiB=K, MiB=M, and so on.

The TIME_STYLE argument can be full-iso, long-iso, iso, locale, or +FORMAT.
FORMAT is interpreted like in date(1). If FORMAT is FORMAT1<newline>FORMAT2,
then FORMAT1 applies to non-recent files and FORMAT2 to recent files.
TIME_STYLE prefixed with 'posix-' takes effect only outside the POSIX locale.
Also the TIME_STYLE environment variable sets the default style to use.

Using color to distinguish file types is disabled both by default and
with --color=never. With --color=auto, ls emits color codes only when
standard output is connected to a terminal. The LS_COLORS environment
variable can change the settings. Use the dircolors command to set it.

Exit status:
0 if OK,
1 if minor problems (e.g., cannot access subdirectory),
2 if serious trouble (e.g., cannot access command-line argument).

GNU coreutils online help: <https://www.gnu.org/software/coreutils/>
Report any translation bugs to <https://translationproject.org/team/>
Full documentation <https://www.gnu.org/software/coreutils/ls>
or available locally via: info '(coreutils) ls invocation'.
```

i-0272635ce49afcae0 (Yash K)  
 Public IPs: 54.87.128.255 Private IPs: 172.31.87.102

```

ls --help
The SIZE argument is an integer and optional unit (example: 10K is 10*1024).
Units are K,M,G,T,P,E,Z,Y (powers of 1024) or KB,MB,... (powers of 1000).
Binary prefixes can be used, too: KiB=K, MiB=M, and so on.

The TIME_STYLE argument can be full-iso, long-iso, iso, locale, or +FORMAT.
FORMAT is interpreted like in date(1). If FORMAT1 is <newline>FORMAT2,
then FORMAT1 applies to non-recent files and FORMAT2 to recent files.
TIME_STYLE prefixed with 'posix-' takes effect only outside the POSIX locale.
Also the TIME_STYLE environment variable sets the default style to use.

Using color to distinguish file types is disabled both by default and
with --color=never. With --color=auto, ls emits color codes only when
standard output is connected to a terminal. The LS_COLORS environment
variable can change the settings. Use the dircolors command to set it.

Exit status:
0 if OK,
1 if minor problems (e.g., cannot access subdirectory),
2 if serious trouble (e.g., cannot access command-line argument).

GNU coreutils online help: <https://www.gnu.org/software/coreutils/>
Report any translation bugs to <https://translationproject.org/team/>
Full documentation <https://www.gnu.org/software/coreutils/ls>
or available locally via: info '(coreutils) ls invocation'
[ec2-user@ip-172-31-87-102 ~]$ 

```

i-0272635ce49afcae0 (Yash K)  
PublicIPs: 54.87.128.255 PrivateIPs: 172.31.87.102



Name	Security group ID	Security group name	VPC ID	Description
-	sg-0dacb8c013807716f	default	vpc-0a107ae4e40bb635e	default VPC security gr...
<input checked="" type="checkbox"/>	sg-0ef419a432a52824	launch-wizard-1	vpc-0a107ae4e40bb635e	launch-wizard-1 create...

sg-0ef419a432a52824 - launch-wizard-1

Details    Inbound rules    Outbound rules    Tags

You can now check network connectivity with Reachability Analyzer    Run Reachability Analyzer

At last terminate the instance.

**CONCLUSION – HENCE LEARNED AND IMPLEMENTED THE STEPS TO CREATE AN EC2 MACHINE.**

## ASSIGNMENT-2

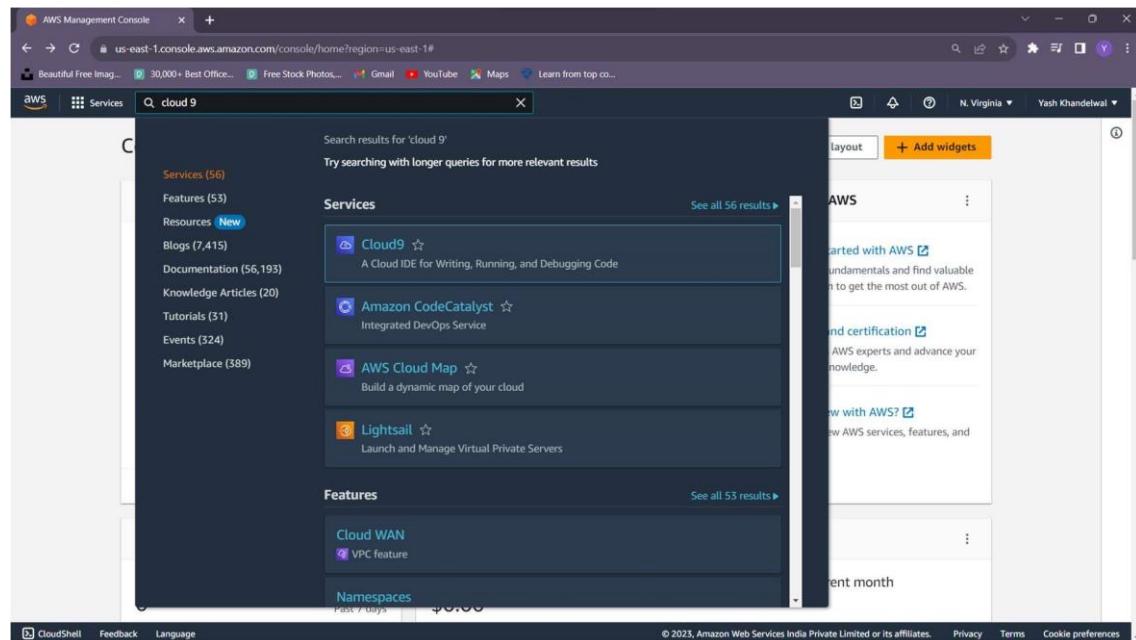
AIM- : To understand the benefits of Cloud Infrastructure and Setup AWS Cloud9 IDE, Launch AWS Cloud9 IDE and Perform Collaboration Demonstration.

### Theory-

Cloud9 IDE is an Online IDE, published as open source from version 2.0, until version 3.0. It supports multiple programming languages, including C, C++, PHP, Ruby, Perl, Python, JavaScript with Node.js, and Go. It is written almost entirely in JavaScript, and uses Node.js on the back-end.

### STEPS-

On the Create environment page enter a name for your environment. To add a description to your environment, enter it in the Description field. For Environment type, choose New EC2 instance to create an Amazon EC2 environment.



The screenshot shows the AWS Cloud9 homepage. At the top right, there is a call-to-action button labeled "Create environment". Below it, a sidebar titled "Getting started" lists several documentation links. The main content area features a section titled "How it works" which describes the process of creating a development environment on an Amazon EC2 instance or connecting to an existing Linux server via SSH.

On the Create environment page enter a name for your environment. To add a description to your environment, enter it in the Description field. For Environment type, choose New EC2 instance to create an Amazon EC2 environment.

The screenshot shows the "Create environment" configuration page. In the "Details" section, the "Name" field is filled with "Yash K". The "Description" field is empty. Under "Environment type", the "New EC2 instance" option is selected. This option is described as creating a new EC2 instance in the user's account. The "Existing compute" option is also available, allowing the user to reuse an existing instance or server.

Now click on Create Environment.

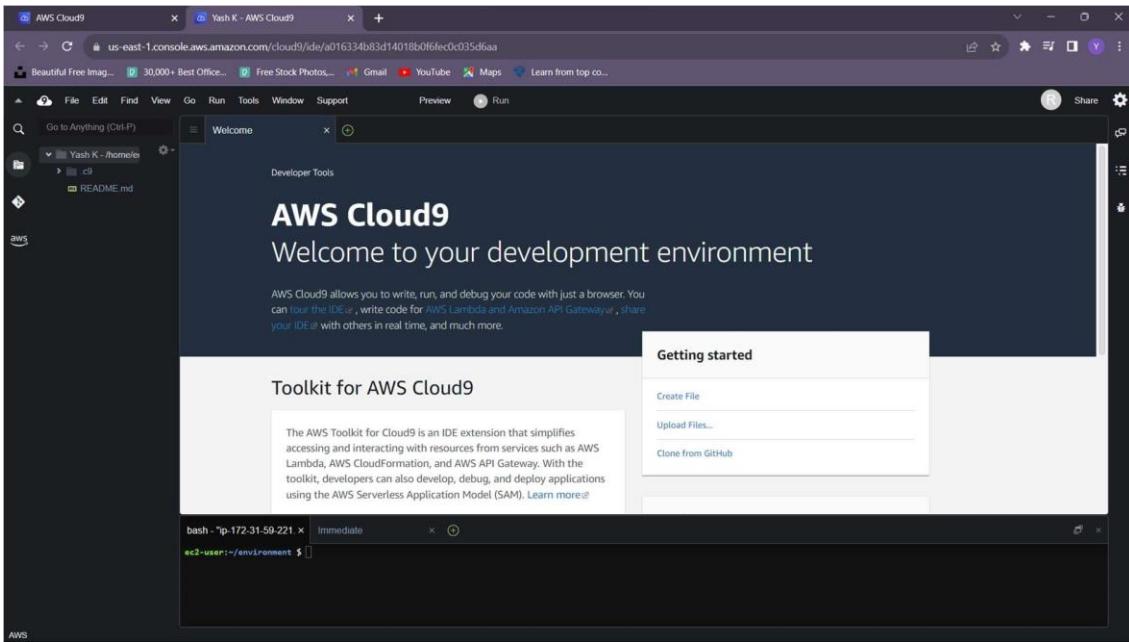
The screenshot shows the AWS Cloud9 environment management interface. On the left, there's a sidebar with 'AWS Cloud9' logo, 'Services' button, and a search bar. Below that are 'Environments' and 'Documentation' links. The main content area has a header 'Creating Yash K. This can take several minutes. While you wait, see Best practices for using AWS Cloud9'. It shows a table titled 'Environments (1)' with one entry: Name (Yash K), Cloud9 IDE (Open), Environment type (EC2 instance), Connection (AWS Systems Manager (SSM)), Permission (Owner), and Owner ARN (arn:awsiam::925116894536:root). There are buttons for 'Delete', 'View details', 'Open in Cloud9', and 'Create environment'.

Check cloud9 instance that you have recently created

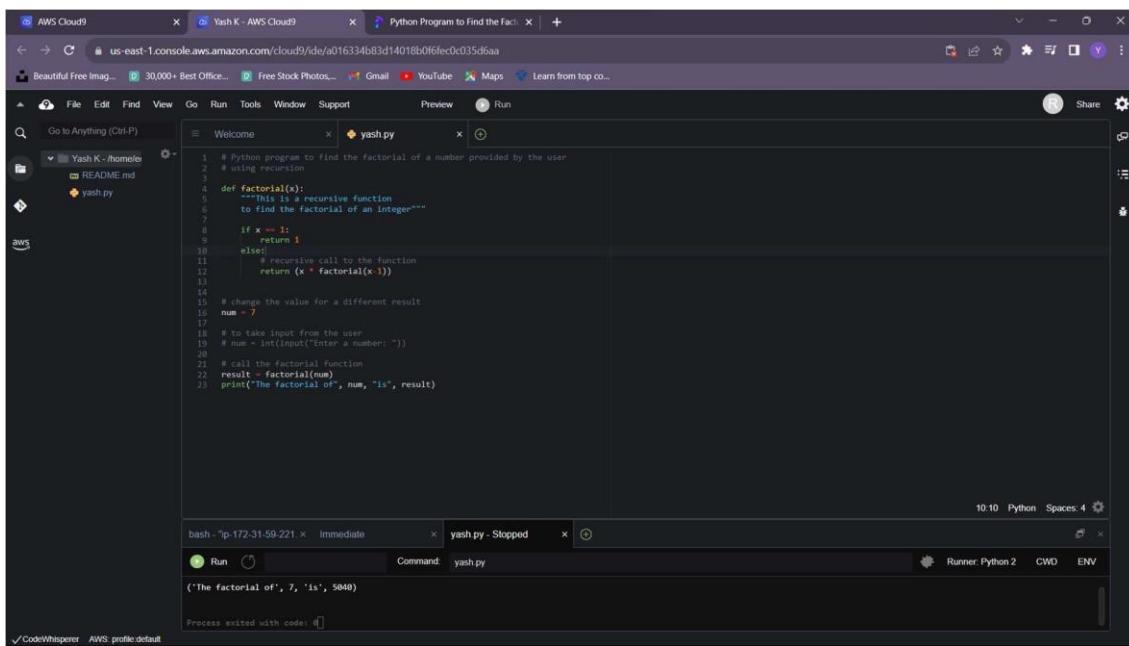
The screenshot shows the AWS Cloud9 IDE interface. The title bar says 'AWS Cloud9 IDE'. The main area features a large blue cloud icon with a white '9' in the center. Below the icon, the text 'Press Ctrl-Alt-Right to select the next instance of a word.' is displayed. At the bottom right, there's a link 'Status Page'.

---

Now select any coding language and perform any opera on via a code. Shown below



You can select any language that you want to execute



## LO Mapped: LO1

Conclusion: Understood the fundamentals of Cloud Computing and are fully proficient with Cloud based DevOps solutions deployment options to meet our business requirements.

## **ASSIGNMENT No. – 3**

**Aim:** To build your application using AWS CodeBuild and Deploy on S3/SEBS using AWS CodePipeline.

**LO Mapped:** LO1, LO2

### **Theory:**

Amazon Simple Storage Service (Amazon S3) is an object storage service offering industry-leading scalability, data availability, security, and performance. Customers of all sizes and industries can store and protect any amount of data for virtually any use case, such as data lakes, cloud-native applications, and mobile apps. With cost-effective storage classes and easy-to-use management features, you can optimize costs, organize data, and configure fine-tuned access controls to meet specific business, organizational, and compliance requirements.

Some of the benefits of AWS S3 are:

- Durability: S3 provides 99.99999999 percent durability.
- Low cost: S3 lets you store data in a range of "storage classes." These classes are based on the frequency and immediacy you require in accessing files.
- Scalability: S3 charges you only for what resources you actually use, and there are no hidden fees or overage charges. You can scale your storage resources to easily meet your organization's ever-changing demands.
- Availability: S3 offers 99.99 percent availability of objects
- Security: S3 offers an impressive range of access management tools and encryption features that provide top-notch security.
- Flexibility: S3 is ideal for a wide range of uses like data storage, data backup, software delivery, data archiving, disaster recovery, website hosting, mobile applications, IoT devices, and much more.
- Simple data transfer: You don't have to be an IT genius to execute data transfers on S3. The service revolves around simplicity and ease of use.

First login to your AWS account.

The screenshot shows the AWS S3 console interface. At the top, a green header bar indicates "Upload succeeded" with a link to "View details below". Below this, a table summarizes the upload results:

Destination	Succeeded	Failed
s3://mynews3webbucket	2 files, 13.4 KB [100.00%]	0 files, 0 B [0%]

Below the summary, a "Files and folders" section displays two items:

Name	Folder	Type	Size	Status
welcome.jpg	img/	image/jpeg	13.1 KB	Succeeded
index.html	-	text/html	279.0 B	Succeeded

At the bottom of the page, there are links for "CloudShell", "Feedback", and "Language", along with a copyright notice: "© 2023, Amazon Web Services India Private Limited or its affiliates." and links for "Privacy", "Terms", and "Cookie preferences".

Then create a new S3 bucket.

The screenshot shows the AWS S3 console interface. A green header bar indicates "Successfully edited public access" with a link to "View details below". Below this, a section titled "Make public: status" shows the following information:

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

Summary	Source	Successfully edited public access	Failed to edit public access
	s3://mynews3webbucket	2 objects, 13.4 KB	0 objects

Below the summary, a "Failed to edit public access" tab is selected, showing "0" entries. At the bottom of the page, there are links for "CloudShell", "Feedback", and "Language", along with a copyright notice: "© 2023, Amazon Web Services India Private Limited or its affiliates." and links for "Privacy", "Terms", and "Cookie preferences".

After that host a static web page using the S3 bucket.

The screenshot shows a web browser window displaying a static website from an S3 bucket. The URL in the address bar is "mynews3webbucket.s3.ap-south-1.amazonaws.com/index.html". The page content is a colorful "WELCOME" banner.

### Website for AWS

**Conclusion:** In this assignment we learnt about AWS S3 bucket and hosted a static web page using the S3 bucket.

# Assignment Number: 4

Aim: To study AWS Code Pipeline and deploy web application using Code Pipeline.

LO mapped: LO1, LO2

Theory:

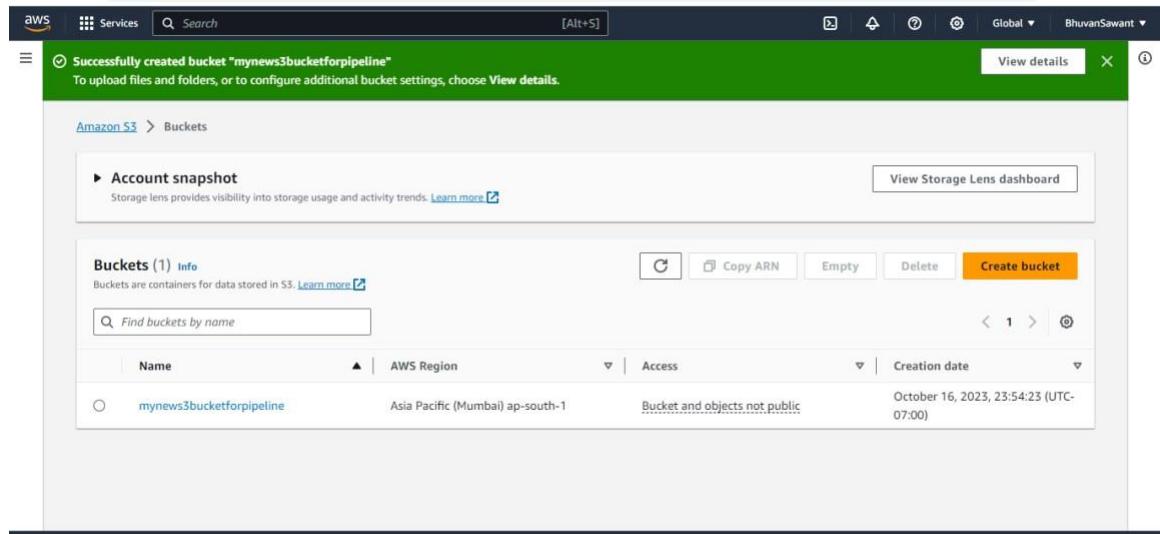
AWS CodePipeline is a continuous integration and continuous delivery (CI/CD) service provided by Amazon Web Services (AWS).

The key aspects of AWS CodePipeline:

**Overview:**

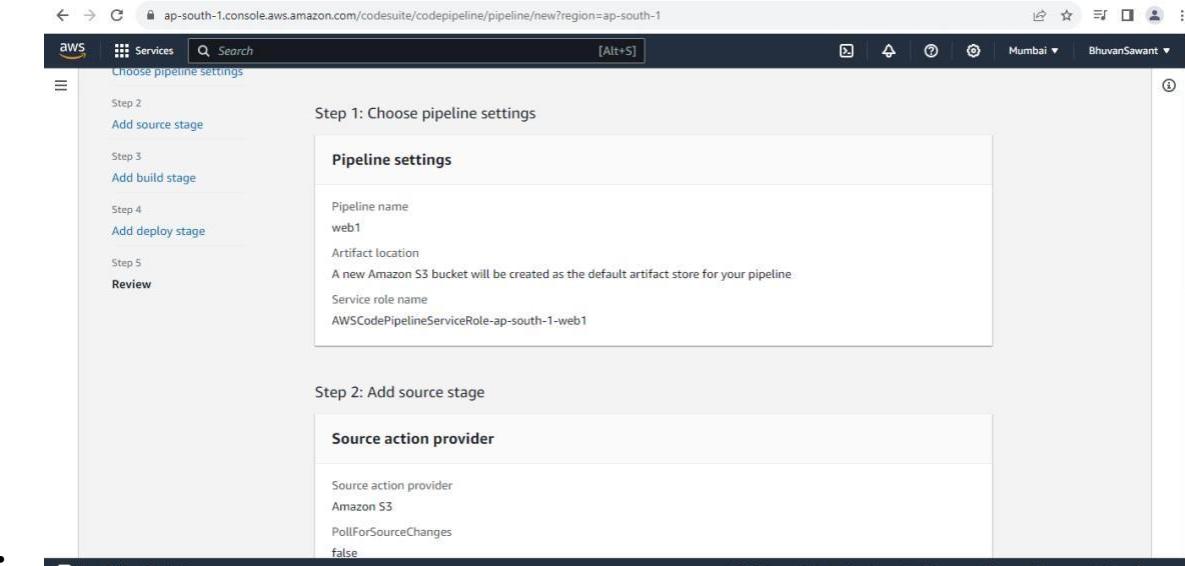
## 1. CI/CD Workflow:

- AWS CodePipeline facilitates the automation of the build, test, and deployment phases of the release process. It allows you to define a series of stages, each of which can represent a phase in your release pipeline.



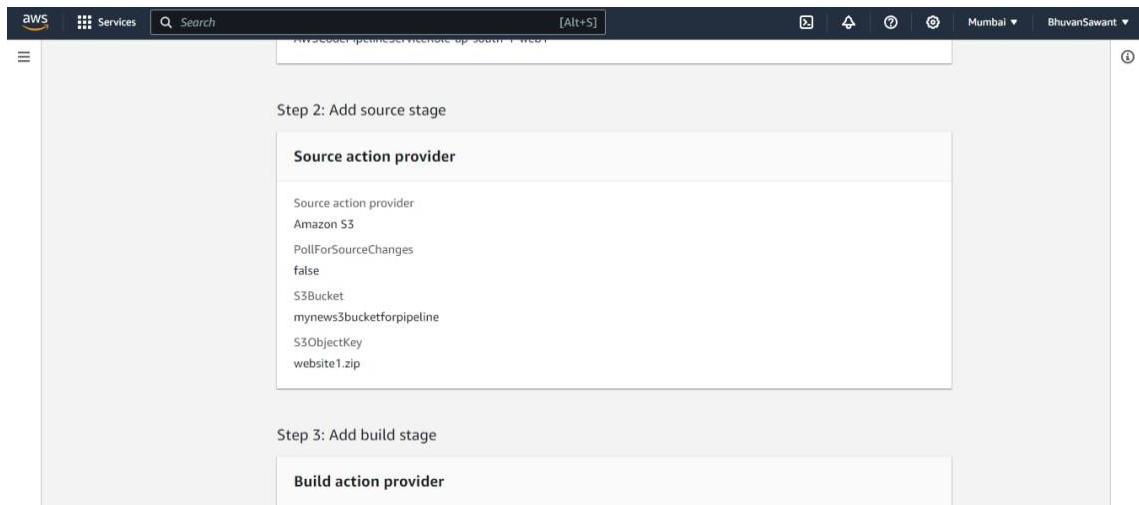
## 2. Integration with Other AWS Services:

- CodePipeline integrates with various AWS services, such as AWS CodeBuild for building applications, AWS CodeDeploy for automating deployments, and AWS Lambda for running custom actions.



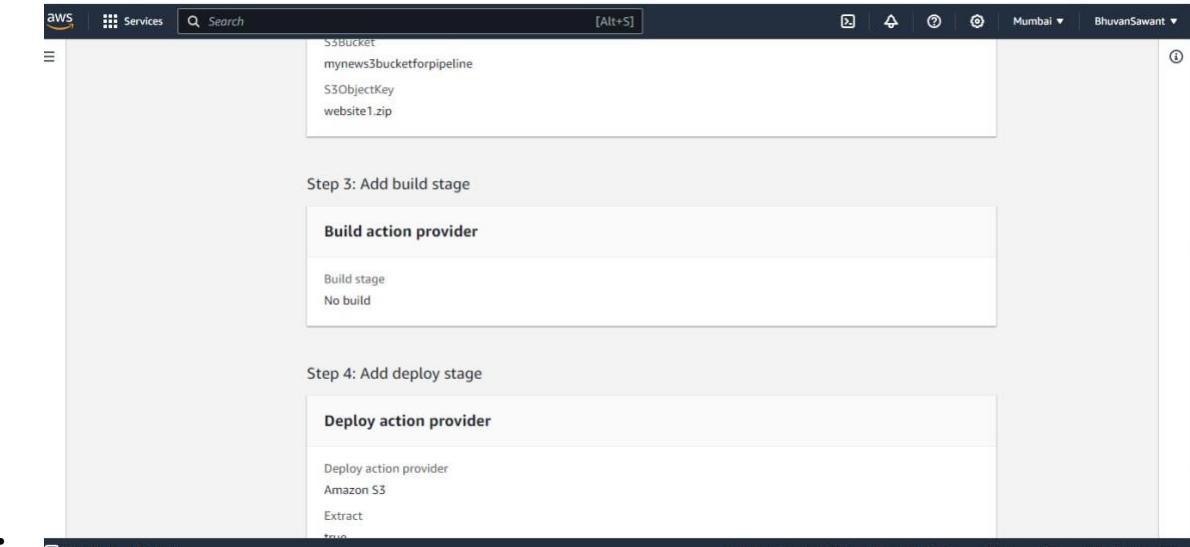
### 3. Pipeline Execution:

- Pipelines consist of a series of stages, and each stage can have one or more actions. Actions represent a task, such as source code retrieval or deployment to a specific environment.



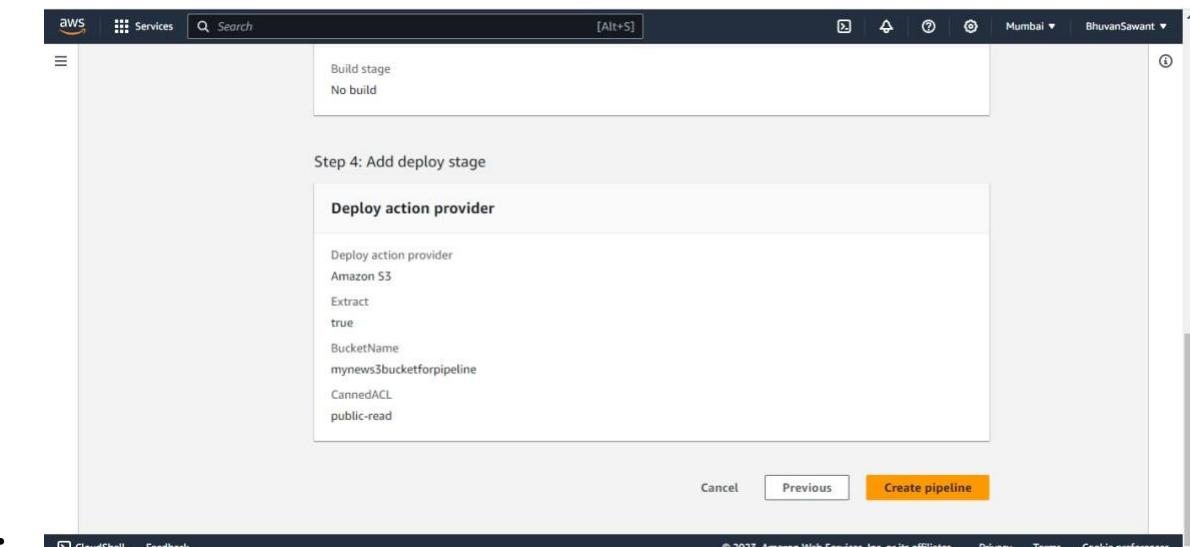
### 4. Source Providers:

- CodePipeline supports integration with various source code repositories, including AWS CodeCommit, GitHub, and Amazon S3.



## 5. Artifact Management:

- CodePipeline uses artifacts to store the files and data needed for each action in a pipeline. Artifacts can be passed between stages to ensure consistency in the deployment process.



## 6. Integration with Third-Party Tools:

- Besides AWS services, CodePipeline supports integration with third-party tools. This is achieved through custom actions, which allow you to use external tools and scripts in your pipeline.

The screenshot shows the AWS S3 console with a file upload in progress. A single file, 'website1.zip', is listed in the 'Files and folders' section. The destination is set to 'mynews3bucketforpipeline'. The 'Destination details' section shows bucket settings for new objects. The 'Permissions' section indicates public access is granted. The bottom navigation bar includes CloudShell and Feedback links.

## 7. Pipeline Visualizations:

- CodePipeline provides a visual representation of your release process, making it easy to understand and monitor the status of each stage and action.

The screenshot shows the AWS S3 console displaying the upload status. It confirms that the upload was successful ('Upload succeeded') with one file ('1 file, 285.0 B (100.00%)'). The 'Summary' table provides a quick overview of the upload results. The 'Files and folders' tab is selected, showing the uploaded file.

## Key Concepts:

### 1. Pipeline:

- A pipeline is a series of stages that represents your release process. Each stage can contain one or more actions.

### 2. Stage:

- A stage is a logical unit in a pipeline, representing a phase in the release process. Stages are executed sequentially.

### 3. Action:

- An action represents a task within a stage. Actions can include tasks such as building code, deploying to a test environment, or running tests.

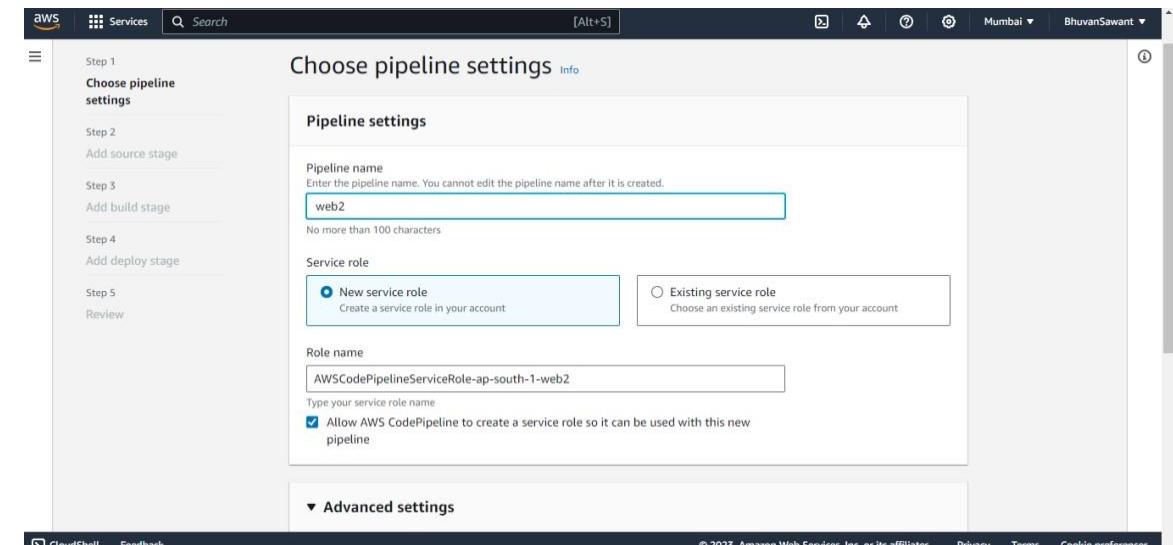
### 4. Artifact:

- Artifacts are the files and data that are produced as a result of an action. They are used to pass information between stages in a pipeline.

To deploy web application using CodePipeline here are the following steps to be followed:

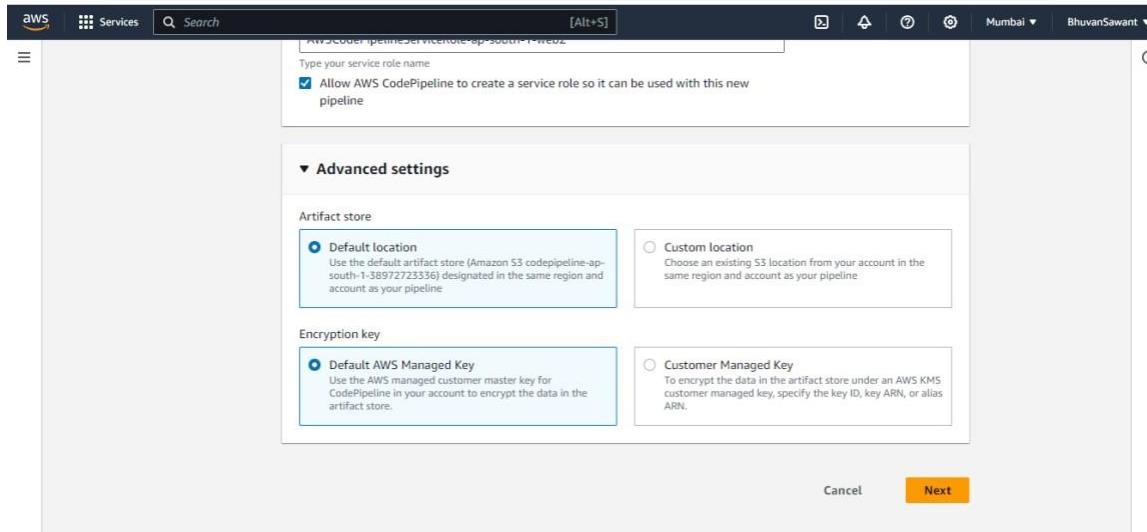
#### 1. Set Up Source Stage:

- Configure a source stage in AWS CodePipeline, linking to your version control system (e.g., CodeCommit, GitHub).



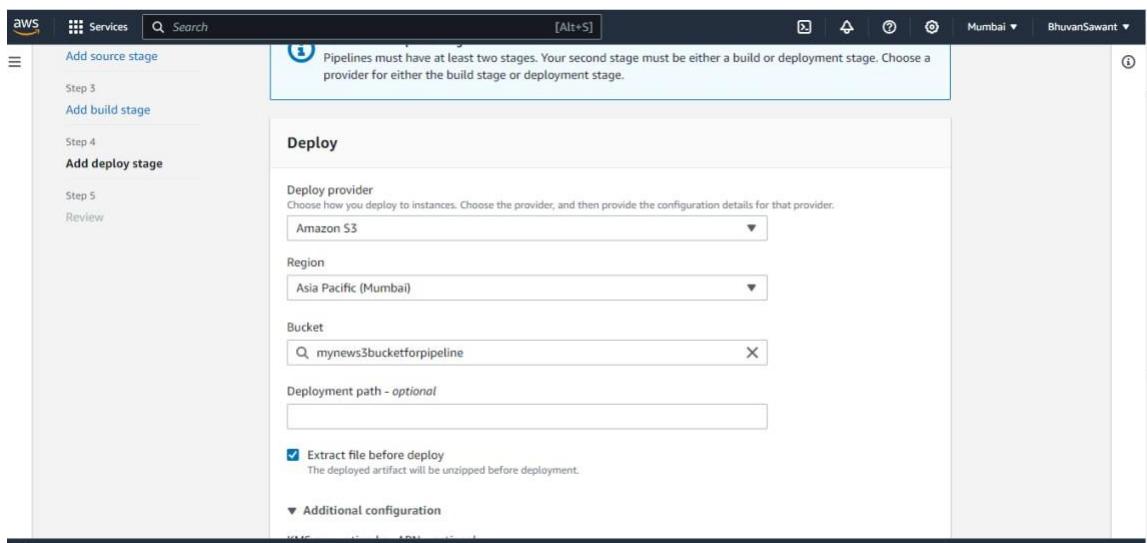
#### 2. Configure Build Stage:

- Set up a build stage using AWS CodeBuild to compile, test, and package your web application.



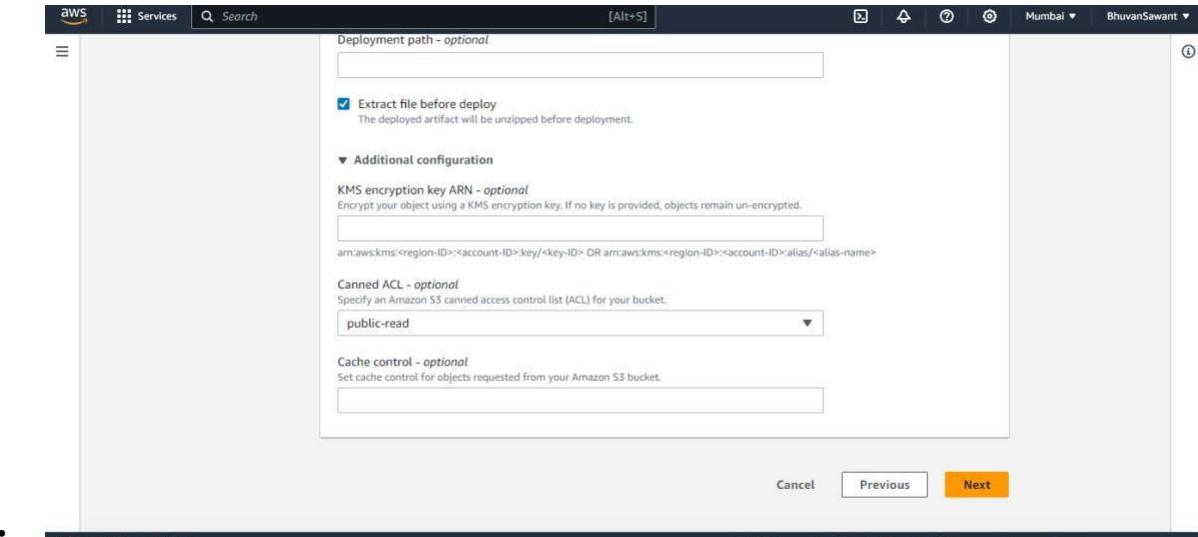
### 3. Define Deployment Stage:

- Create a deployment stage using AWS CodeDeploy or another deployment provider to deploy your application to target environments.



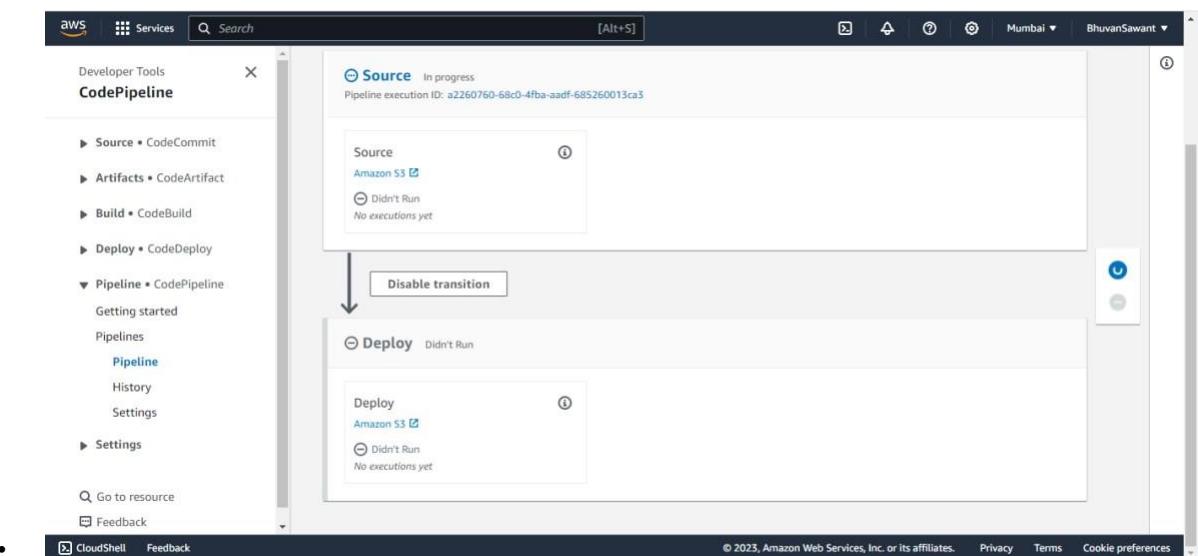
### 4. Configure Approval (Optional):

- Optionally, add a manual approval stage to review and approve deployments before proceeding to the next stage.



## 5. Artifact Passing:

- Ensure proper passing of artifacts between stages to maintain consistency in the deployment process.



## 6. Add Monitoring (Optional):

- Integrate monitoring tools (e.g., AWS CloudWatch) to track the performance and health of your application during and after deployment.

## 7. Configure Notifications (Optional):

- Set up notifications using AWS SNS or other services to receive alerts about pipeline events and status changes.

**8. Test and Validate:**

- Test the pipeline by triggering a build, ensuring that each stage executes successfully, and the application deploys as expected.

**9. Modify Pipeline as Needed:**

- Make adjustments to the pipeline configuration based on the specific requirements of your web application and deployment process.

**10. Continuous Improvement:**

- Implement continuous improvement practices, such as monitoring feedback, optimizing build and deployment scripts, and iterating on the pipeline structure.

Conclusion: By this assignment we learned how to host static web page through codepipeline.

.

## Assignment No. 5

### **Q.1 What are the various Kubernetes services running on nodes? Describe the role of each service.**

Kubernetes nodes run several services to ensure the proper operation of containerized applications. Here are the key services and their roles:

#### **1. Kubelet:**

- Role: The Kubelet is an agent running on each node, responsible for managing and maintaining the containers running on the node. It communicates with the Kubernetes API server to ensure that the containers are in the desired state.

#### **2. Container Runtime:**

- Role: This is the software responsible for running containers, such as Docker or containerd. It ensures that containers are isolated and that the processes inside them have limited access to resources.

#### **3. Kube Proxy:**

- Role: Kube Proxy maintains network rules on nodes. It enables communication between different pods and services, implementing network policies defined in Kubernetes.

#### **4. CRI (Container Runtime Interface):**

- Role: CRI is an interface between Kubernetes and the container runtime. It standardizes the way Kubernetes interacts with different container runtimes, ensuring compatibility.

#### **5. Node Problem Detector:**

- Role: Node Problem Detector is responsible for detecting and reporting node-level issues, such as hardware or operating system problems. It helps with node recovery and maintenance.

#### **6. Container Storage Interface (CSI):**

Role: CSI allows storage systems to be seamlessly integrated with Kubernetes. It enables dynamic provisioning and management of persistent storage for pods.

## 7. **Sysctld:**

- Role: Sysctld enforces system-level policies by managing kernel parameters and ensuring they align with Kubernetes security and performance requirements.

## 8. **Time Synchronization:**

- Role: Accurate time synchronization is essential for various Kubernetes features, like certificate management and coordination among nodes. NTP (Network Time Protocol) or similar services ensure time synchronization.

## 9. **OS-Level Monitoring and Logging Agents:**

- Role: These agents collect data on the node's performance, resource usage, and logs. Popular agents include Prometheus for monitoring and Fluentd or Logstash for log collection.

## 10. **OS Update Agents:**

- Role: OS update agents ensure that the underlying operating system remains up to date with the latest security patches and updates.

## **Q.2 What is Pod Disruption Budget (PDB)?**

A Pod Disruption Budget (PDB) is a resource in Kubernetes used to define policies that control the disruption of pods during maintenance operations, such as node draining or scaling down a deployment. PDBs help ensure high availability of applications by setting constraints on how many pods of a certain type can be disrupted at a given time.

The key elements of a PDB include:

- **minAvailable:** This field specifies the minimum number of pods that must be available at all times. It prevents excessive disruption by ensuring that a minimum number of healthy pods are maintained.

- **maxUnavailable:** This field allows a certain number or percentage of pods to be unavailable during disruptions. It provides flexibility while ensuring some level of redundancy.
- **Selector:** PDBs are associated with pods using labels and selectors, specifying which pods are subject to the budget.

By using PDBs, you can control the impact of maintenance operations on applications, avoiding scenarios where too many pods are disrupted simultaneously, which could lead to service downtime or instability. **Q.3**

### **What is the role of Load Balance in Kubernetes?**

Load balancing in Kubernetes is a critical component for ensuring the availability, scalability, and reliability of applications. Here are the key roles of load balancing in Kubernetes:

#### **1. Service Discovery:**

- Load balancers provide a stable endpoint (IP or DNS name) for services. Clients can connect to the service without needing to know the specific pod's IP address, allowing for dynamic scaling and failover.

#### **2. Traffic Distribution:**

- Load balancers distribute incoming traffic to multiple pods or replicas of a service. This ensures even distribution of workloads, preventing overloading of specific pods and enhancing application performance.

#### **3. High Availability:**

- Load balancers can detect unhealthy pods or nodes and redirect traffic away from them. This ensures that even if some pods fail, the service remains available and operational.

#### **4. Scalability:**

- As traffic increases, load balancers can route requests to new pods or nodes. This elasticity allows applications to handle increased load without manual intervention.

## **5. Session Persistence:**

Some load balancers support session affinity, ensuring that requests from the same client are consistently directed to the same pod. This is important for stateful applications.

## **6. Security:**

- Load balancers can enforce security policies, such as rate limiting or access control, to protect services from malicious traffic.

## **7. External Connectivity:**

- In cloud environments, load balancers often provide external IP addresses, allowing applications to be exposed to the internet securely. This is particularly important for web services.

## **8. Global Load Balancing:**

- Some load balancers support global load balancing, distributing traffic across multiple clusters in different geographical regions for improved performance and resilience.

Load balancing in Kubernetes can be implemented using various technologies, including Ingress controllers, service types like NodePort or LoadBalancer, and cloud-managed load balancers from providers like AWS, Google Cloud, and Azure. It plays a crucial role in managing the complexity of containerized applications and ensuring their reliability and scalability.



# Assignment Number: 6

**Aim:** To build, change, and destroy AWS / GCP / Microsoft Azure / Digital Ocean infrastructure

Using Terraform. **LO mapped:** LO1 and LO3

## Theory:

To build change, and destroy AWS infrastructure Using Terraform we would use following tools:

1. IAM (AWS SERVICE)
2. EC2(only for AMI id)
3. VS code
4. Terraform installation in Local Machine via Environment variables

## IAM Service:

AWS Identity and Access Management (IAM) acts like a guardian for your AWS resources. It lets you create personalized "passes" for users and roles, specifying what they're allowed to do. You can group users together for easier management. IAM also keeps a record of actions taken, ensuring security and accountability. Essentially, IAM safeguards your AWS "club" by controlling who can access what.

### STEPS TO CREATE AN IAM USER: I.

In AWS search for IAM

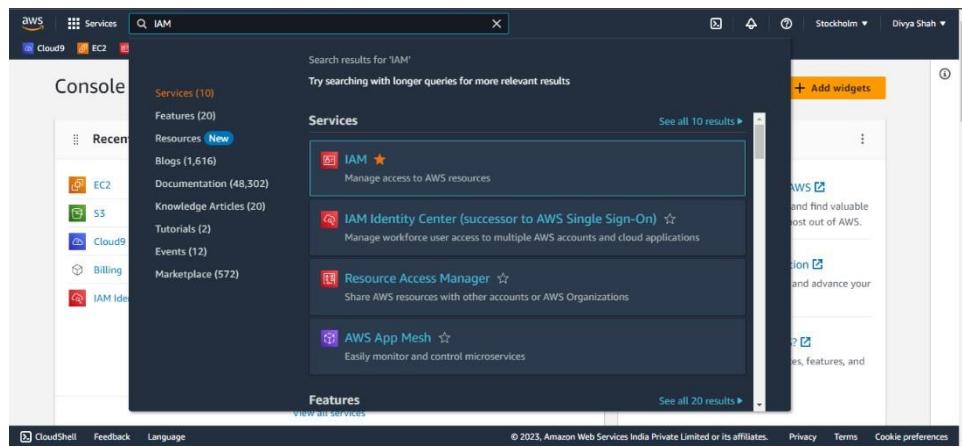


Fig 1

Diptanshu Mishra

Roll No:78

## II. The dashboard of IAM will look like this (Fig. 2)

Now click on “Users” present in the left panel to create an IAM User:

The screenshot shows the AWS IAM Dashboard. On the left sidebar, under 'Access management', 'Users' is selected. The main area displays 'Security recommendations' with two items: 'Add MFA for root user' and 'Root user has no active access keys'. Below this is the 'IAM resources' section, which includes tabs for User groups, Users, Roles, Policies, and Identity providers. The 'Users' tab shows 0 users. To the right, there are sections for 'AWS Account' (Account ID: 317209544424, Account Alias: Create, Sign-in URL: https://317209544424.signin.aws.amazon.com/console), 'Quick Links' (My security credentials, Policy simulator, Web identity federation playground), and 'Tools' (CloudShell, Feedback, Language). A 'What's new' section at the bottom lists recent changes.

Fig. 2

The screenshot shows the 'Users' page in the AWS IAM service. The left sidebar shows 'Access management' selected. The main area displays a table titled 'Users (0) Info' with one row: 'User name' (t22\_115). The table includes columns for User name, Path, Group, Last activity, MFA, Password age, Console last sign-in, and Access key ID. Buttons for 'Create user' and 'Manage workforce users' are visible at the top right. A message at the top encourages using Identity Center for workforce users.

Fig.3

## III. Provide a “Username” and click next:

The screenshot shows the 'Specify user details' step of the IAM User creation wizard. The left sidebar shows 'Step 1: Specify user details' selected. The main area has a 'User details' section with a 'User name' field containing 't22\_115'. Below it is a note about character restrictions and a checkbox for 'Provide access to the AWS Management Console - optional'. At the bottom are 'Cancel' and 'Next Step' buttons.

4

Fig. 4

Diptanshu Mishra

Roll No:78

- IV. Here for permissions click on “Attach policies directly” and select “Administrator Access”. Then click on next as follows:

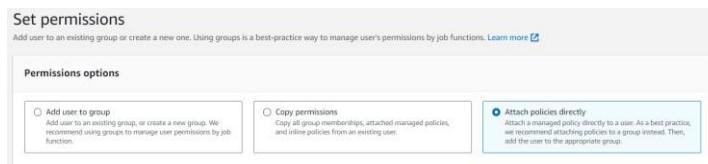


Fig. 5

Permissions policies (1/1122)		
Choose one or more policies to attach to your new user.		
<input type="button" value="Create policy"/> <input type="button" value="Edit"/>		
Policy name	Type	Attached entities
<input type="checkbox"/> AccessAnalyzerServiceRolePolicy	AWS managed	0
<input checked="" type="checkbox"/> AdministratorAccess	AWS managed - job function	0
<input type="checkbox"/> AdministratorAccess-Amplify	AWS managed	0
<input type="checkbox"/> AdministratorAccess-AWSElasticBea...	AWS managed	0
<input type="checkbox"/> AlexaForBusinessDeviceSetup	AWS managed	0
<input type="checkbox"/> AlexaForBusinessFullAccess	AWS managed	0

Fig. 6

6

- V. Now click on “Create user”:

User details		
User name: T22_115	Console password type: None	Require password reset: No
<b>Permissions summary</b> Name:  AdministratorAccess Type: AWS managed - job function Used as: Permissions policy		
<b>Tags - optional</b> <small>Tags are key-value pairs you can add to AWS resources to help identify, organize, or search for resources. Choose any tags you want to associate with this user.</small> <small>No tags associated with the resource.</small> <input type="button" value="Add new tag"/> <small>You can add up to 50 more tags.</small>		

Fig. 7

- VI. After successful IAM user creation go to user and click on “Username” and then create an access key for it as follows:

Users (1) <small>Info</small>										
An IAM user is an identity with long-term credentials that is used to interact with AWS in an account.										
<input type="button" value="Create user"/> <input type="button" value="Delete"/>										
<input type="button" value="Search"/>										
	User name	Path	Group	Last activity	MFA	Password age	Console last sign-in	Access key ID		
<input type="checkbox"/>	T22_115	/	0	-	-	-	-	-		

Fig. 8

Diptanshu Mishra

Roll No:78



Fig.9

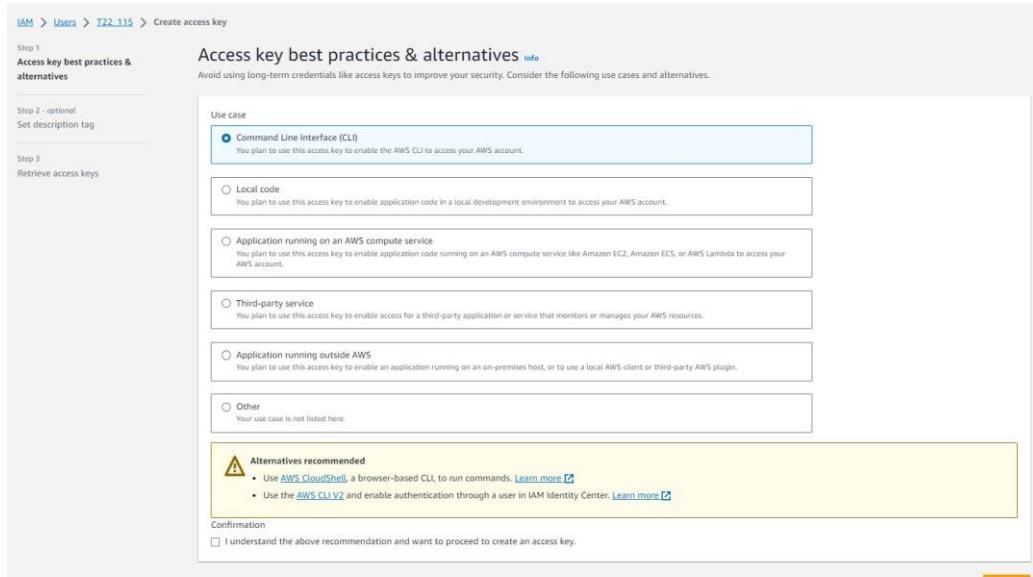


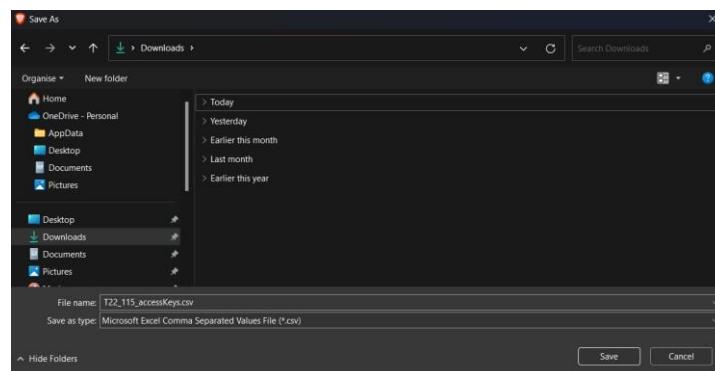
Fig.10

The below description is recommended to be kept empty. After this click on “Generate a key”:



Fig.11

VII. Now we need to Download the .csv file for copying the Access key as well as Secret access key:



Diptanshu Mishra

Roll No:78

Fig. 12

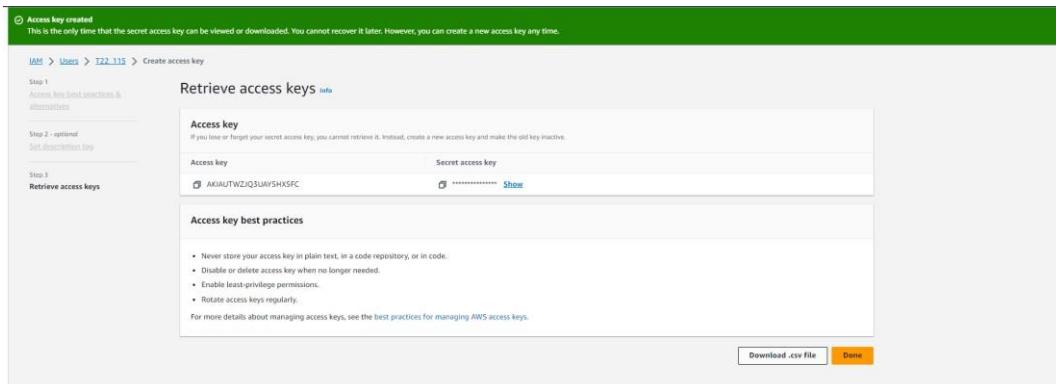


Fig.13

- VIII. Now we need to open vs code an create a folder for .tf file operations. You can follow procedure as follows for creating .tf file with following code:

```

provider "aws" {
  region="us-east-2" // recommended to use us-east-2 for bug free working
  access_key="AKIAUTWZJQ3UAY5HX5FC" // Your access key
  secret_key = "PLYZXQpTHvx$OW7J5YLCQU+wBxOEpEmojnVN2yDI" //Your secret key
}

resource "aws_instance" "myT22_115"{
  ami="ami-0d3183af565a0a95d"
  instance_type="t2.micro"
}

```

Fig.14

- IX. Now we need to install terraform from chrome to set it in our environment variables:

*Note: You need to download “386” for local windows machine.*

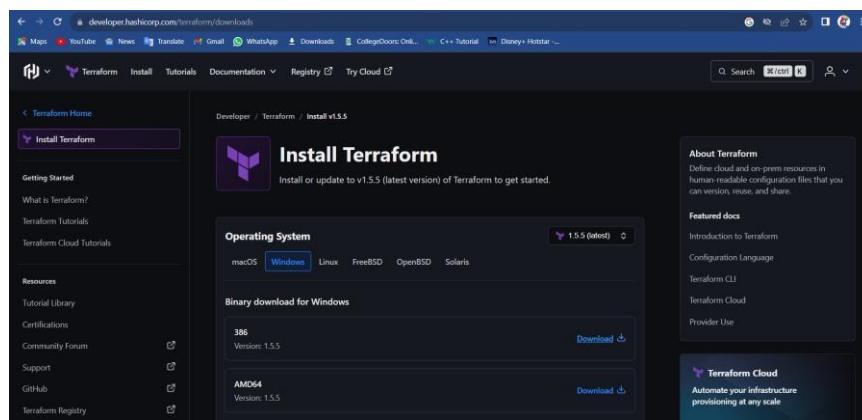


Fig.15

Copy path of downloaded Terraform and set it in environment variables as follow:

Diptanshu Mishra

Roll No:78

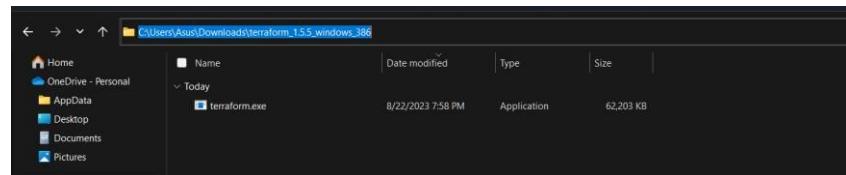


Fig.16

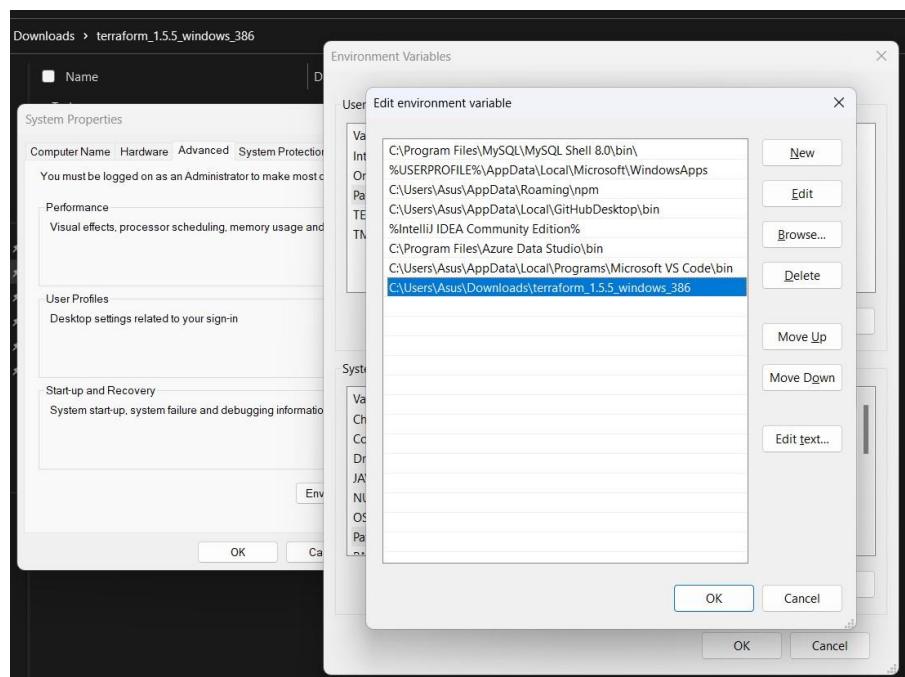


Fig.17

X. Now open cmd to location where you have saved you .tf file:

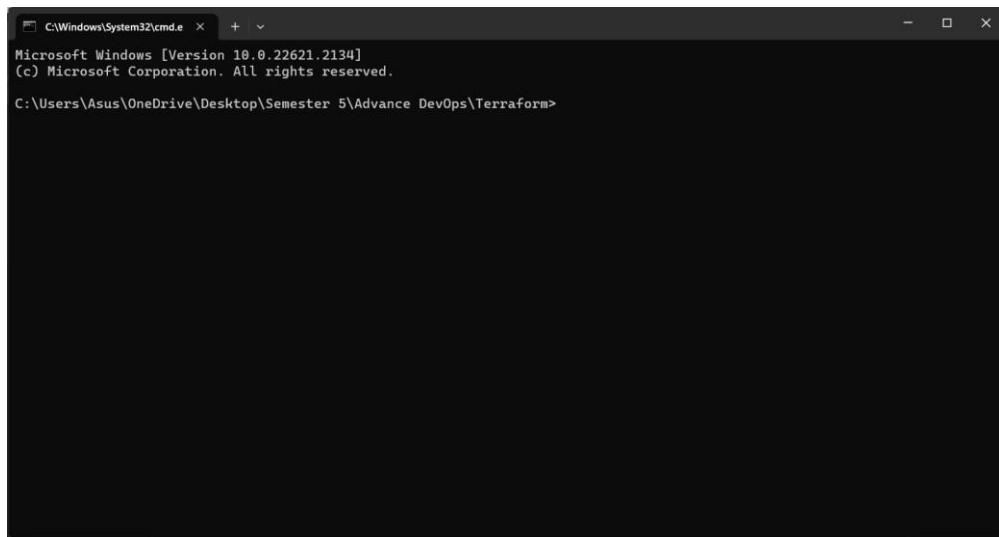


Fig. 18

XI. Now run command:

*command: terraform init*

```

Microsoft Windows [Version 10.0.22621.2134]
(c) Microsoft Corporation. All rights reserved.

C:\Users\Asus\OneDrive\Desktop\Semester 5\Advance DevOps\Terraform>terraform init

Initializing the backend...

Initializing provider plugins...
- Finding latest version of hashicorp/aws...
- Installing hashicorp/aws v5.13.1...
- Installed hashicorp/aws v5.13.1 (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.

C:\Users\Asus\OneDrive\Desktop\Semester 5\Advance DevOps\Terraform>

```

Fig. 19

## XII. Now run command:

*command: terraform plan*

```

+ outpost_arn          = (known after apply)
+ password_data        = (known after apply)
+ placement_group      = (known after apply)
+ placement_partition_number = (known after apply)
+ primary_network_interface_id = (known after apply)
+ private_dns          = (known after apply)
+ private_ip           = (known after apply)
+ public_dns           = (known after apply)
+ public_ip            = (known after apply)
+ secondary_private_ips = (known after apply)
+ security_groups      = (known after apply)
+ source_dest_check    = true
+ spot_instance_request_id = (known after apply)
+ subnet_id            = (known after apply)
+ tags_all             = (known after apply)
+ tenancy              = (known after apply)
+ user_data             = (known after apply)
+ user_data_base64     = (known after apply)
+ user_data_replace_on_change = false
+ vpc_security_group_ids = (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.

C:\Users\Asus\OneDrive\Desktop\Semester 5\Advance DevOps\Terraform>

```

Fig. 20

The new files which are created after above command.

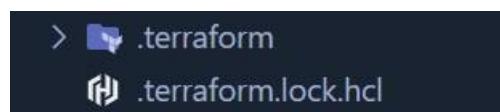


Fig. 21

## XIII. Now we need to run the command:

*command: terraform apply*

```

+ monitoring           = (known after apply)
+ outpost_arn          = (known after apply)
+ password_data        = (known after apply)
+ placement_group      = (known after apply)
+ placement_partition_number = (known after apply)
+ primary_network_interface_id = (known after apply)
+ private_dns          = (known after apply)
+ private_ip           = (known after apply)
+ public_dns           = (known after apply)
+ public_ip            = (known after apply)
+ secondary_private_ips = (known after apply)
+ security_groups      = (known after apply)
+ source_dest_check    = true
+ spot_instance_request_id = (known after apply)
+ subnet_id            = (known after apply)
+ tags_all             = (known after apply)
+ tenancy              = (known after apply)
+ user_data             = (known after apply)
+ user_data_base64     = (known after apply)
+ user_data_replace_on_change = false
+ vpc_security_group_ids = (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:

```

Fig. 22

```

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.myT22_115: Creating...
aws_instance.myT22_115: Still creating... [10s elapsed]
aws_instance.myT22_115: Still creating... [20s elapsed]
aws_instance.myT22_115: Creation complete after 25s [id=i-0882605e5f3bc53fb]

Apply complete! Resources: 1 added, 0 changed, 0 destroyed.

C:\Users\Asus\OneDrive\Desktop\Semester 5\Advance DevOps\Terraform>

```

Fig. 23

XIV. Now after the above command you can see an instance is created in AWS:

	Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public IPv4 DNS
	-	i-0882605e5f3bc53fb	Running	t2.micro	2/2 checks passed	No alarms	+ us-east-2c	ec2-18-221-129-

Fig. 24

XV. Now to destroy the created ec2 instance we need to run command:

*command: terraform destroy*

```

- root_block_device {
  - delete_on_termination = true -> null
  - device_name           = "/dev/xvda" -> null
  - encrypted             = false -> null
  - iops                  = 100 -> null
  - kms_key_id            = {} -> null
  - throughput            = 0 -> null
  - volume_id              = "vol-065015f5e8beb670c" -> null
  - volume_size            = 8 -> null
  - volume_type            = "gp2" -> null
}
}

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

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.

Enter a value: yes

aws_instance.myT22_115: Destroying... [id=i-0882605e5f3bc53fb]
aws_instance.myT22_115: Still destroying... [id=i-0882605e5f3bc53fb, 10s elapsed]
aws_instance.myT22_115: Still destroying... [id=i-0882605e5f3bc53fb, 20s elapsed]
aws_instance.myT22_115: Still destroying... [id=i-0882605e5f3bc53fb, 30s elapsed]
aws_instance.myT22_115: Destruction complete after 32s

Destroy complete! Resources: 1 destroyed.

```

Fig. 25

Resources		EC2 Global view	
You are using the following Amazon EC2 resources in the US East (Ohio) Region:			
Instances (running)	0	Auto Scaling Groups	0
Elastic IPs	0	Instances	1
Load balancers	0	Placement groups	0
Snapshots	0	Volumes	0
<span style="color: #0072bc;">(i)</span> Easily size, configure, and deploy Microsoft SQL Server Always On availability groups on AWS using the <a href="#">AWS Launch Wizard for SQL Server</a> . <a href="#">Learn more</a>			

Fig. 26

**Note: Do Delete your IAM User before logging off from your AWS account and any other sort of instance or program running.**

Diptanshu Mishra

Roll No:78

**Conclusion:** By this assignment we created, changed and destroyed AWS infrastructure using terraform.

## Assignment 7

Aim: TO perform static analysis on python programs using SonarCube SAST processes

LO Mapped: L04 Theory:

Download and Sonar Scanner

The screenshot shows the SonarQube download page. At the top, there's a banner for SonarQube 9.1. Below it, the main heading is "Download SonarQube" with the subtitle "The leading product for Code Quality and Security". A sub-subtitle "HELPING DEVS SINCE 2008" is present. The page features four main sections representing different editions:

- Community EDITION**: Described as "Used and loved by 200,000+ companies" and "FREE & OPEN SOURCE". It has a "Download for free" button.
- Developer EDITION**: Described as "Built for developers by developers". It has a "Download" button.
- Enterprise EDITION**: Described as "Designed to meet Enterprise Requirements". It has a "Download" button.
- Data Center EDITION**: Described as "Designed for High Availability". It has a "Download" button.

Each section lists specific features or requirements. For example, the Developer edition supports C, C++, Obj-C, Swift, ABAP, T-SQL, PL/SQL, and detection of injection flaws.

The screenshot shows the SonarScanner documentation page under the "Docs 9.1" section. The left sidebar contains navigation links for various SonarQube components like Requirements, Setup and Upgrade, Analyzing Source Code, Scanners, Analysis Parameters, Languages, Test Coverage & Execution, Importing External Issues, and Background Tasks. The main content area is titled "SonarScanner". It includes a "4.6.2" version section with a "Show more versions" link, a "Configuring your project" section with instructions on creating a configuration file, and a code snippet for "sonar-project.properties". The code snippet includes comments for project key, name, version, and source path.

```

# must be unique in a given SonarQube instance
sonar.projectKey=my:project

# --- optional properties ---

# defaults to project key
sonar.projectName=My project
# defaults to 'not provided'
sonar.projectVersion=1.0

# Path is relative to the sonar-project.properties file. Defaults to .
#sonar.sources=.

```

## After downloading, set Environment Variables. Add "sonarqube-9.1.0.47736\bin" to Path.

Open command prompt. Run commands:

- cd "sonarqube-9.1.0.47736\bin\windows-x86-64"
- StartSonar.bat

```
Administrator: SonarQube
Microsoft Windows [Version 10.0.19043.1237]
(c) Microsoft Corporation. All rights reserved.

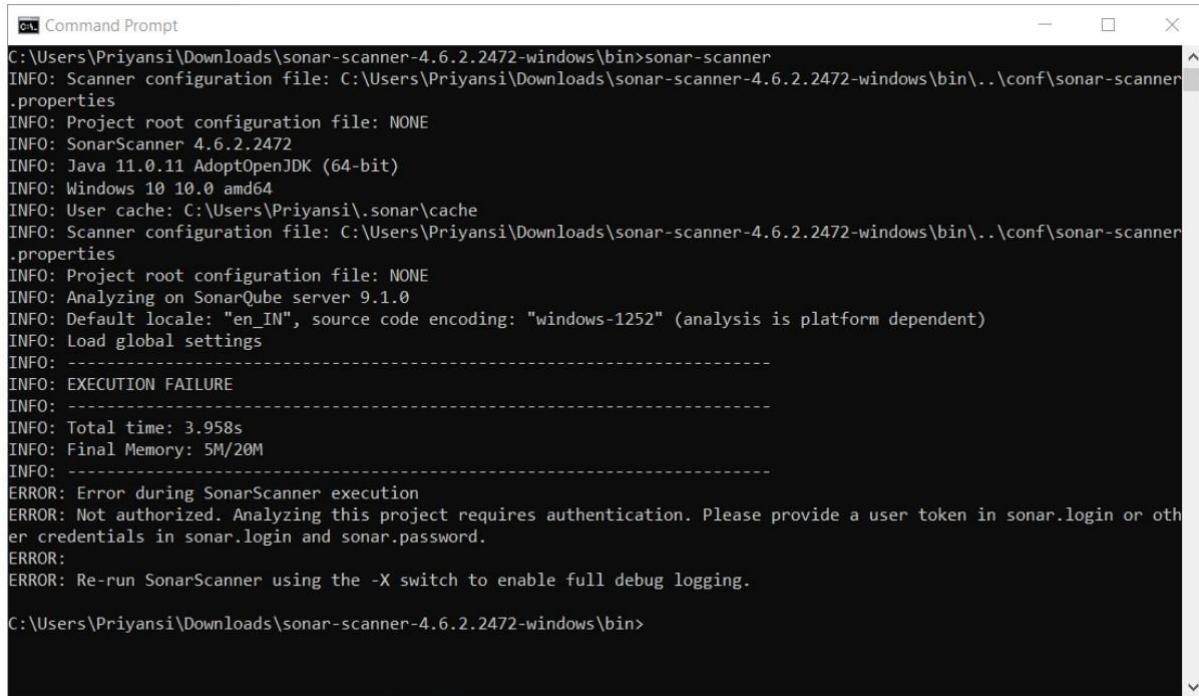
C:\Users\Priyansi\Downloads\sonarqube-9.1.0.47736\bin\windows-x86-64>StartSonar.bat
wrapper | Wrapper Started as Console
wrapper | Launching a JVM...
Wrapper (Version 3.2.3) http://wrapper.tanukisoftware.org
jvm 1 | Copyright 1999-2006 Tanuki Software, Inc. All Rights Reserved.
jvm 1 |
jvm 1 | 2021-09-29 13:50:37 INFO app[[o.s.a.AppFileSystem] Cleaning or creating temp directory C:\Users\Priyansi\Downloads\sonarqube-9.1.0.47736\temp
jvm 1 | 2021-09-29 13:50:37 INFO app[[o.s.a.es.Settings] Elasticsearch listening on [HTTP: 127.0.0.1:9001, TCP: 127.0.0.1:53055]
jvm 1 | 2021-09-29 13:50:37 INFO app[[o.s.a.ProcessLauncherImpl] Launch process[{"key='es', ipcIndex=1, logFilenamePrefix='es'}] from [C:\Users\Priyansi\Downloads\sonarqube-9.1.0.47736\elasticsearch]: C:\Program Files\Java\jdk-11.0.12\bin\java -Djava.awt.headless=true -Dfile.encoding=UTF-8 -Djava.io.tmpdir=C:\Users\Priyansi\Downloads\sonarqube-9.1.0.47736\tmp -Xms512m -Xmx512m -XX:+OmitStackTraceInFastThrow -Dio.netty.noKeySetOptimization=true -Dio.netty.recycler.maxCapacityPerThread=9 -Dio.netty.allocator.numDirectArenas=0 -Dlog4j2.disable.jmc=true -Djava.locale.providers=COMPAT -Xmx12m -Xms12m -XX:+MaxDirectMemorySize=256m -XX:HeapDumpOnOutOfMemoryError -Delasticsearch -Des.path.home=C:\Users\Priyansi\Downloads\sonarqube-9.1.0.47736\elasticsearch -Des.path.conf=C:\Users\Priyansi\Downloads\sonarqube-9.1.0.47736\tmp\conf\*
`-cp lib/* org.elasticsearch.bootstrap.Elasticsearch
jvm 1 | 2021-09-29 13:50:37 INFO app[[o.s.a.SchedulerImpl] Waiting for Elasticsearch to be up and running
jvm 1 | 2021-09-29 13:50:39 ERROR app[[o.s.a.esManagedProcess] Failed to check status
jvm 1 | org.elasticsearch.common.ElasticsearchException: java.util.concurrent.ExecutionException: java.net.ConnectException: Timeout connecting to [/127.0.0.1:9001]
jvm 1 | at org.elasticsearch.client.RestHighLevelClient.performClientRequest(RestHighLevelClient.java:2078)
jvm 1 | at org.elasticsearch.client.RestHighLevelClient.get(ClusterHealthStatus$ClusterStateResponse$RestHighLevelClient.java:173)
jvm 1 | at org.elasticsearch.client.RestHighLevelClient$ClusterHealthFuture.get(ClusterHealthStatus$ClusterStateResponse$RestHighLevelClient.java:170)
jvm 1 | at org.elasticsearch.client.ClusterClient.get(ClusterHealthClient.java:119)
jvm 1 | at org.sonar.application.es.EsConnectorImpl.getClusterHealthStatus(EsConnectorImpl.java:64)
jvm 1 | at org.sonar.application.process.EsManagedProcess.checkStatus(EsManagedProcess.java:90)
jvm 1 | at org.sonar.application.process.EsManagedProcess.checkOperational(EsManagedProcess.java:75)
jvm 1 | at org.sonar.application.process.EsManagedProcess.isOperational(EsManagedProcess.java:66)
jvm 1 | at org.sonar.application.process.ManagedProcessHandler.refreshState(ManagedProcessHandler.java:220)
jvm 1 | at org.sonar.application.process.ManagedProcessHandler.refresh(ManagedProcessHandler.java:209)
jvm 1 | Caused by: java.util.concurrent.ExecutionException: java.net.ConnectException: Timed out connecting to [/127.0.0.1:9001]
jvm 1 | at org.apache.http.nio.pool.RouteSpecificPool$timeout(RouteSpecificPool.java:169)
jvm 1 | at org.apache.http.nio.pool.AbstractNIOConnPool.requestTimedOut(AbstractNIOConnPool.java:628)
jvm 1 | at org.apache.http.nio.reactor.SessionRequestingImpl.timeout(SessionRequestingImpl.java:184)
jvm 1 | at org.apache.http.impl.nio.reactor.DefaultConnectingIOReactor.processTimeouts(DefaultConnectingIOReactor.java:214)
jvm 1 | at org.apache.http.impl.nio.reactor.DefaultConnectingIOReactor.processEvents(DefaultConnectingIOReactor.java:158)
jvm 1 | at org.apache.http.impl.nio.reactor.AbstractMultiworkerIOReactor.execute(AbstractMultiworkerIOReactor.java:351)
jvm 1 | at org.apache.http.impl.nio.conn.PoolingNHttpClientConnectionManager.execute(PoolingNHttpClientConnectionManager.java:221)
jvm 1 | at org.apache.http.impl.nio.client.CloseableHttpAsyncClientBase$1.run(CloseableHttpAsyncClientBase.java:64)
jvm 1 | at java.base/java.lang.Thread.run(Thread.java:34)
jvm 1 |
... 10 common frames omitted
jvm 1 | Caused by: java.net.ConnectException: Timeout connecting to [/127.0.0.1:9001]
jvm 1 | at org.apache.http.nio.pool.RouteSpecificPool$timeout(RouteSpecificPool.java:169)
jvm 1 | at org.apache.http.nio.pool.AbstractNIOConnPool.requestTimedOut(AbstractNIOConnPool.java:628)
jvm 1 | at org.apache.http.nio.reactor.SessionRequestingImpl.timeout(SessionRequestingImpl.java:184)
jvm 1 | at org.apache.http.impl.nio.reactor.DefaultConnectingIOReactor.processTimeouts(DefaultConnectingIOReactor.java:214)
jvm 1 | at org.apache.http.impl.nio.reactor.DefaultConnectingIOReactor.processEvents(DefaultConnectingIOReactor.java:158)
jvm 1 | at org.apache.http.impl.nio.reactor.AbstractMultiworkerIOReactor.execute(AbstractMultiworkerIOReactor.java:351)
jvm 1 | at org.apache.http.impl.nio.conn.PoolingNHttpClientConnectionManager.execute(PoolingNHttpClientConnectionManager.java:221)
jvm 1 | at org.apache.http.impl.nio.client.CloseableHttpAsyncClientBase$1.run(CloseableHttpAsyncClientBase.java:64)
jvm 1 | at java.base/java.lang.Thread.run(Thread.java:34)
```

```
Administrator: SonarQube
java 1 | at org.elasticsearch.client.RestHighLevelClient.performRequest(RestHighLevelClient.java:1702)
java 1 | at org.elasticsearch.client.RestHighLevelClient.performRequestAndParseEntity(RestHighLevelClient.java:1672)
java 1 | at org.elasticsearch.client.ClusterClient.health(ClusterClient.java:119)
java 1 | at org.sonar.application.es.EsConnectorImpl.getClusterHealthStatus(EsConnectorImpl.java:64)
java 1 | at org.sonar.application.process.EsManagedProcess.checkStatus(EsManagedProcess.java:90)
java 1 | at org.sonar.application.process.EsManagedProcess.checkOperational(EsManagedProcess.java:75)
java 1 | at org.sonar.application.process.EsManagedProcess.isOperational(EsManagedProcess.java:66)
java 1 | at org.sonar.application.process.ManagedProcessHandler.refreshState(ManagedProcessHandler.java:220)
java 1 | at org.sonar.application.process.ManagedProcessHandler.refresh(ManagedProcessHandler.java:209)
java 1 | Caused by: java.util.concurrent.ExecutionException: java.net.ConnectException: Timed out connecting to [/127.0.0.1:9001]
java 1 | at org.elasticsearch.common.util.concurrent.BaseFuture$Sync.getValue(BaseFuture.java:262)
java 1 | at org.elasticsearch.common.util.concurrent.BaseFuture$Sync.get(BaseFuture.java:249)
java 1 | at org.elasticsearch.common.util.concurrent.BaseFuture.get(BaseFuture.java:76)
java 1 | at org.elasticsearch.client.RestHighLevelClient.performClientRequest(RestHighLevelClient.java:2075)
java 1 | ... 10 common frames omitted
java 1 | Caused by: java.net.ConnectException: Timeout connecting to [/127.0.0.1:9001]
java 1 | at org.apache.http.nio.pool.RouteSpecificPool$timeout(RouteSpecificPool.java:169)
java 1 | at org.apache.http.nio.pool.AbstractNIOConnPool.requestTimedOut(AbstractNIOConnPool.java:628)
java 1 | at org.apache.http.nio.reactor.SessionRequestingImpl.timeout(SessionRequestingImpl.java:184)
java 1 | at org.apache.http.impl.nio.reactor.DefaultConnectingIOReactor.processTimeouts(DefaultConnectingIOReactor.java:214)
java 1 | at org.apache.http.impl.nio.reactor.DefaultConnectingIOReactor.processEvents(DefaultConnectingIOReactor.java:158)
java 1 | at org.apache.http.impl.nio.reactor.AbstractMultiworkerIOReactor.execute(AbstractMultiworkerIOReactor.java:351)
java 1 | at org.apache.http.impl.nio.conn.PoolingNHttpClientConnectionManager.execute(PoolingNHttpClientConnectionManager.java:221)
java 1 | at org.apache.http.impl.nio.client.CloseableHttpAsyncClientBase$1.run(CloseableHttpAsyncClientBase.java:64)
java 1 | at java.base/java.lang.Thread.run(Thread.java:834)
java 1 | 2021-09-29 13:51:50 INFO app[[s.a.SchedulerImpl] Processes is up
java 1 | 2021-09-29 13:51:50 INFO app[[s.a.SchedulerImpl] Starting search process[{"key='web', ipcIndex=2, logFilenamePrefix='web'}] from [C:\Users\Priyansi\Downloads\sonarqube-9.1.0.47736]: C:\Program Files\Java\jdk-11.0.12\bin\java -Djava.awt.headless=true -Dfile.encoding=UTF-8 -Djava.io.tmpdir=C:\Users\Priyansi\Downloads\sonarqube-9.1.0.47736\tmp -Xms512m -Xmx512m -XX:+HeapDumpOnOutOfMemoryError -Dhttp.nonProxyHosts=localhost|127.7.1::1 -Djava.net.http.nonProxyHosts=localhost|127.7.1::1 -Djava.net.http.nonProxyHosts=localhost|127.7.1::1 -Djava.net.http.nonProxyHosts=localhost|127.7.1::1 -Djava.net.http.nonProxyHosts=localhost|127.7.1::1 -cp .\lib\sonar-application-9.1.0.47736.jar;C:\Users\Priyansi\Downloads\sonarqube-9.1.0.47736\lib\jdbc\h2-1.4.199.jar org.sonar.server.web.WebServer C:\Users\Priyansi\Downloads\sonarqube-9.1.0.47736\tmp\sq-process177945169724401819properties
java 1 | [ 2021-09-29 13:51:42 INFO app[[o.s.a.SchedulerImpl] Process[web] is up
java 1 | [ 2021-09-29 13:51:42 INFO app[[o.s.a.ProcessLauncherImpl] Launch process[{"key='es', ipcIndex=3, logFilenamePrefix='es'}] from [C:\Users\Priyansi\Downloads\sonarqube-9.1.0.47736]: C:\Program Files\Java\jdk-11.0.12\bin\java -Djava.awt.headless=true -Dfile.encoding=UTF-8 -Djava.io.tmpdir=C:\Users\Priyansi\Downloads\sonarqube-9.1.0.47736\tmp -Xms512m -Xmx512m -XX:+HeapDumpOnOutOfMemoryError -Dhttp.nonProxyHosts=localhost|127.7.1::1 -Djava.net.http.nonProxyHosts=localhost|127.7.1::1 -Djava.net.http.nonProxyHosts=localhost|127.7.1::1 -Djava.net.http.nonProxyHosts=localhost|127.7.1::1 -Djava.net.http.nonProxyHosts=localhost|127.7.1::1 -cp .\lib\sonar-application-9.1.0.47736.jar;C:\Users\Priyansi\Downloads\sonarqube-9.1.0.47736\lib\jdbc\h2-1.4.199.jar org.sonar.ce.app.CeServer C:\Users\Priyansi\Downloads\sonarqube-9.1.0.47736\tmp\sq-process3544487414319509.properties
java 1 | [ 2021-09-29 13:51:42 WARN app[[start] #####]
java 1 | [ 2021-09-29 13:51:42 WARN app[[start] Default Administrator credentials are still being used. Make sure to change the password or deactivate the account.
java 1 | [ 2021-09-29 13:51:42 WARN app[[start] #####]
java 1 | [ 2021-09-29 13:51:42 INFO app[[o.s.a.SchedulerImpl] Process[ce] is up
java 1 | [ 2021-09-29 13:51:46 INFO app[[o.s.a.SchedulerImpl] SonarQube is up
```

Open another command prompt. Run command: e

```
cd "sonar-scanner-4.6.2.2472-windows\bin"
```

```
sonar-scanner
```

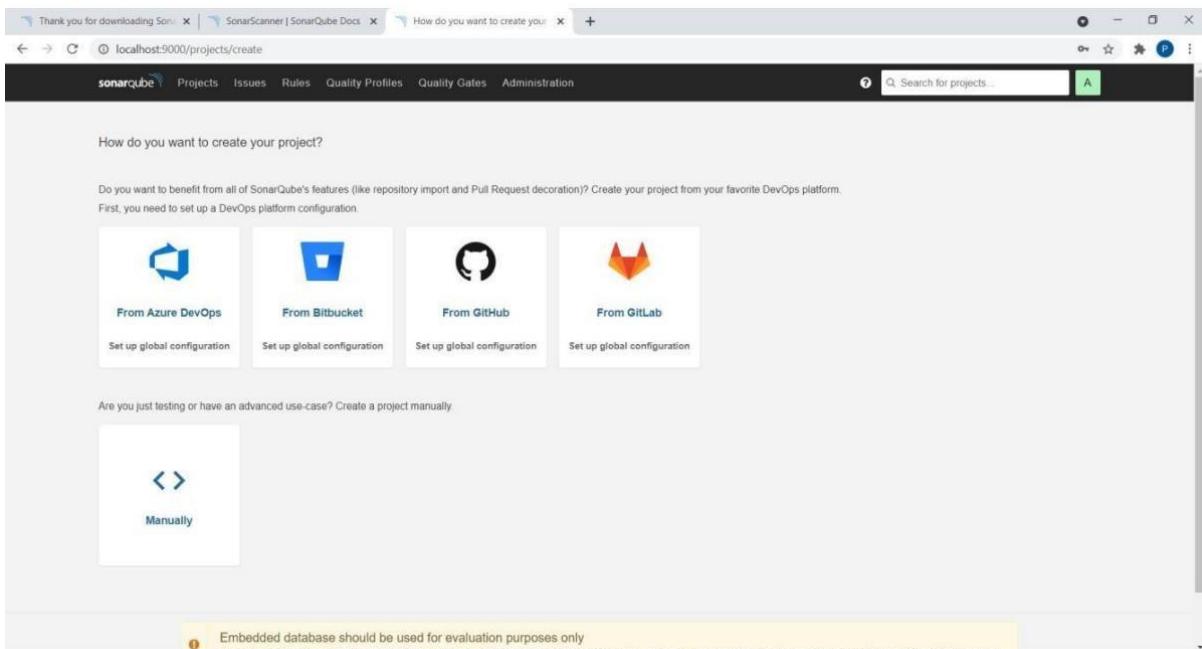
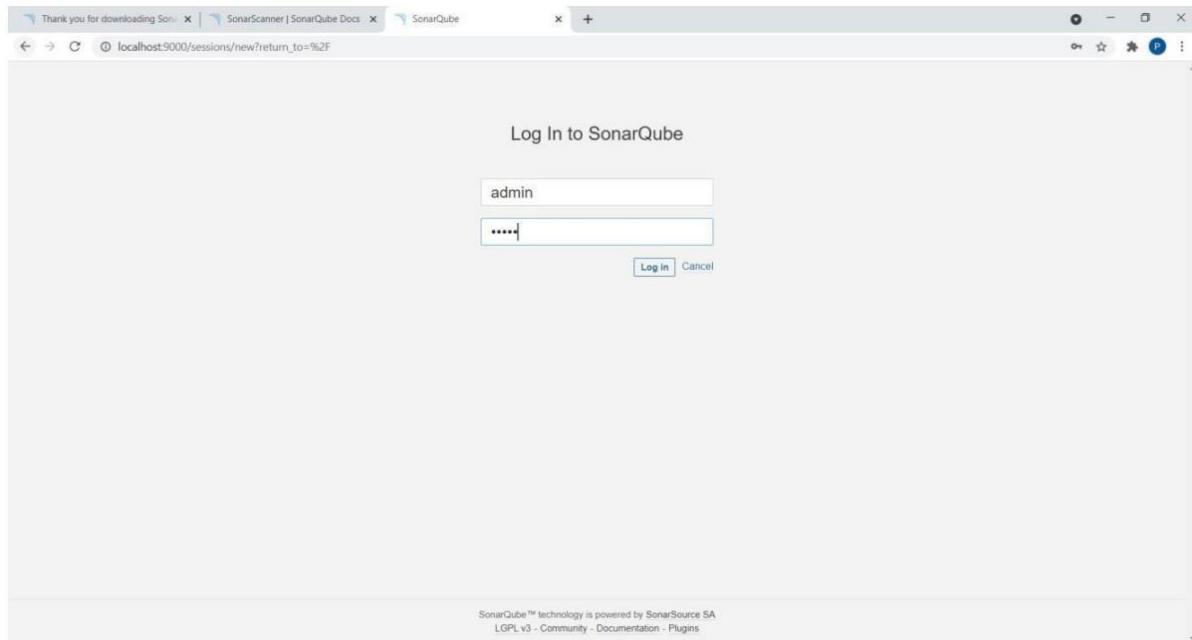


```
C:\Users\Priyansi\Downloads\sonar-scanner-4.6.2.2472-windows\bin>sonar-scanner
INFO: Scanner configuration file: C:\Users\Priyansi\Downloads\sonar-scanner-4.6.2.2472-windows\bin\..\conf\sonar-scanner.properties
INFO: Project root configuration file: NONE
INFO: SonarScanner 4.6.2.2472
INFO: Java 11.0.11 AdoptOpenJDK (64-bit)
INFO: Windows 10 10.0 amd64
INFO: User cache: C:\Users\Priyansi\.sonar\cache
INFO: Scanner configuration file: C:\Users\Priyansi\Downloads\sonar-scanner-4.6.2.2472-windows\bin\..\conf\sonar-scanner.properties
INFO: Project root configuration file: NONE
INFO: Analyzing on SonarQube server 9.1.0
INFO: Default locale: "en_IN", source code encoding: "windows-1252" (analysis is platform dependent)
INFO: Load global settings
INFO: -----
INFO: EXECUTION FAILURE
INFO: -----
INFO: Total time: 3.958s
INFO: Final Memory: 5M/20M
INFO: -----
ERROR: Error during SonarScanner execution
ERROR: Not authorized. Analyzing this project requires authentication. Please provide a user token in sonar.login or other credentials in sonar.login and sonar.password.
ERROR:
ERROR: Re-run SonarScanner using the -X switch to enable full debug logging.

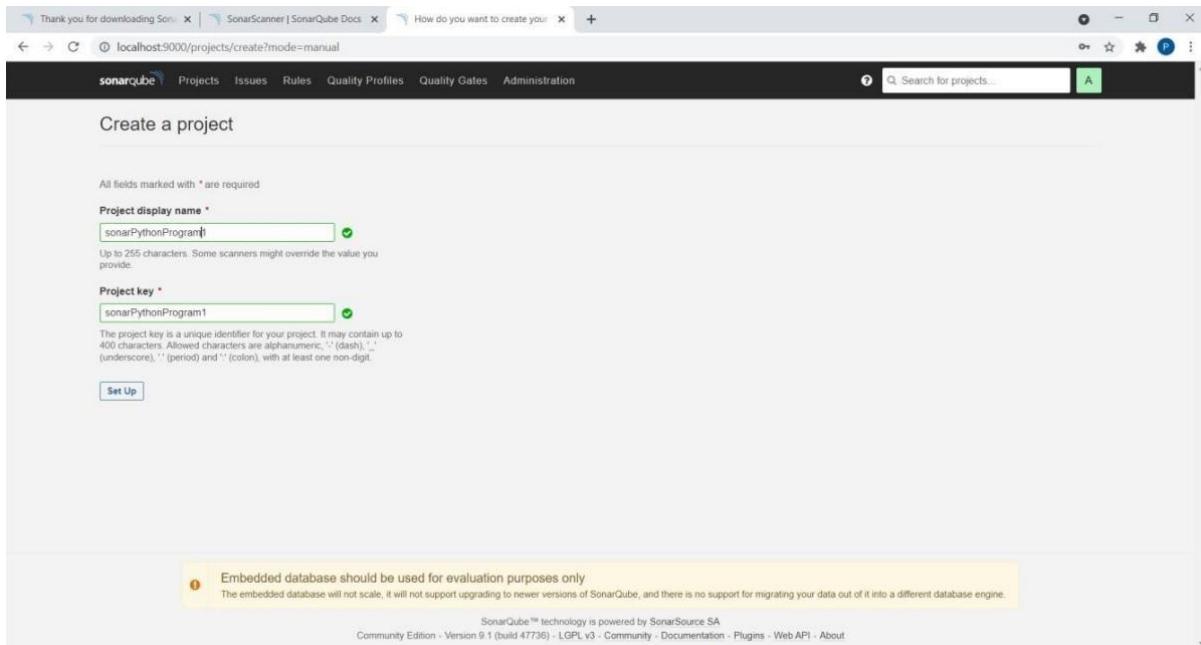
C:\Users\Priyansi\Downloads\sonar-scanner-4.6.2.2472-windows\bin>
```

Server up and running on localhost:9000

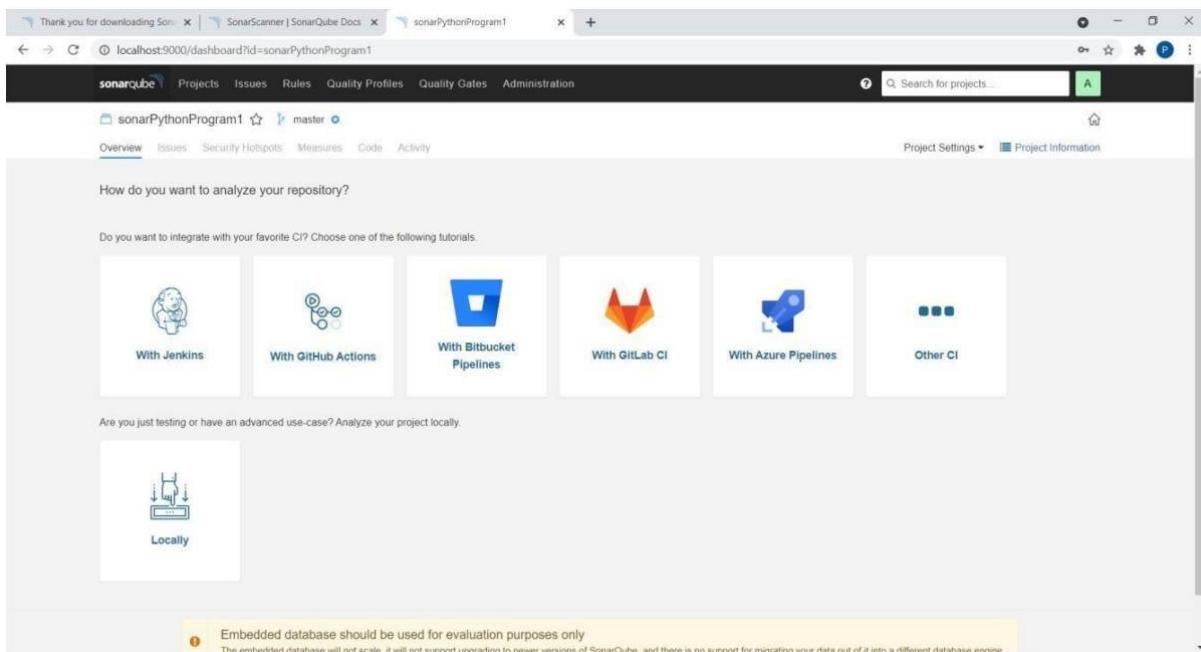
Login using credentials as User: admin and Password: admin  
and Set a new password



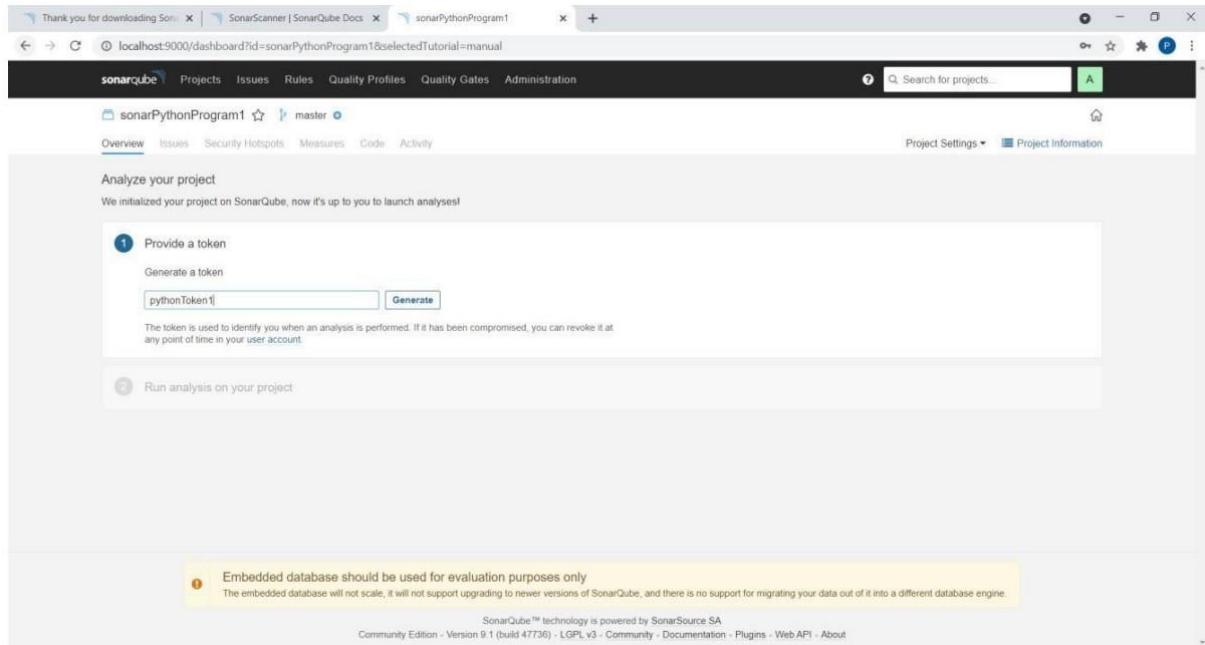
Click on Create a project Manually.



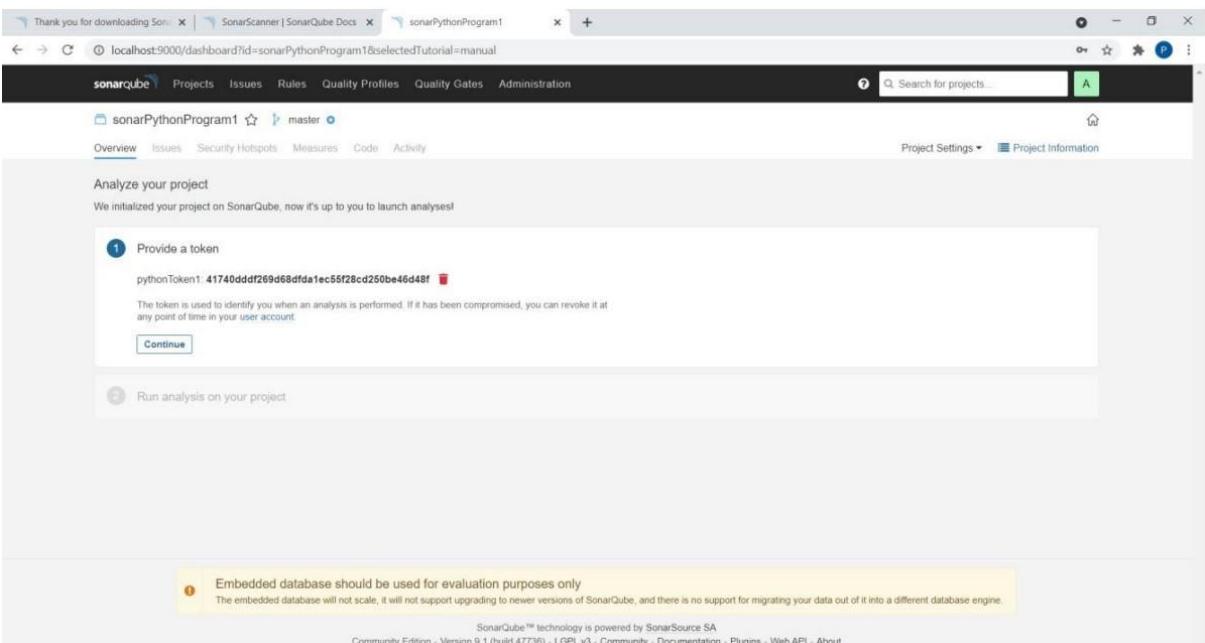
Give any Project display name.



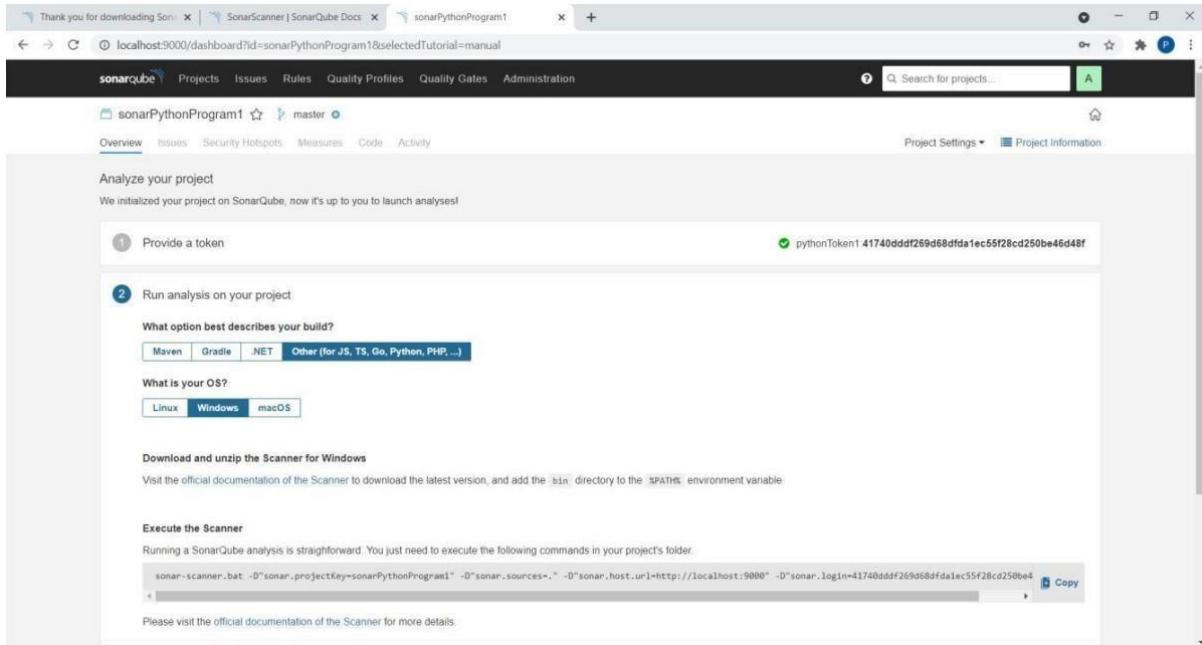
Click on Locally.



Give any name to token and click on Generate.



Click on Continue.



Save a Python program in a folder.

```
class Solution(object):
    def romanToInt(self, s):
        roman = num = 0
        while i < len(s):
            if i+1 < len(s) and s[i:i+2] in roman:
                num += roman[s[i:i+2]]
            else:
                #print(i)
                num += roman[s[i]]
        return num
obl = Solution()
print(obl.I")
print(obl . I")
```

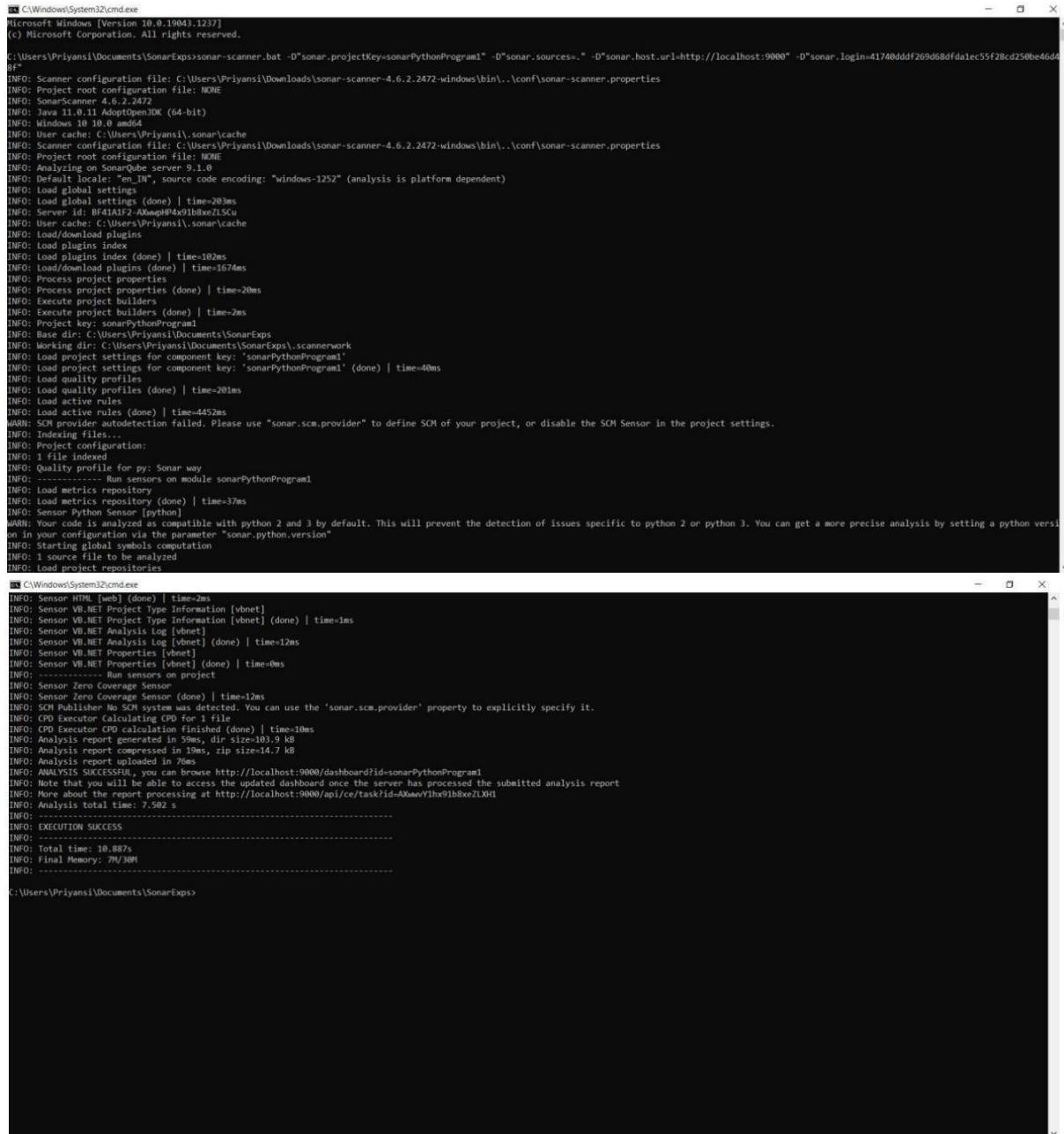
Open command prompt in this folder and Run program using copied command.

`sonar-scanner.bat -D"sonar.projectKey=<YourDisplayName>"`

-

`D"sonar.sources=." -D"sonar.host.url=http://localhost:9000" -`

`D"sonar.login=<YourTokenGeneratedID>"`

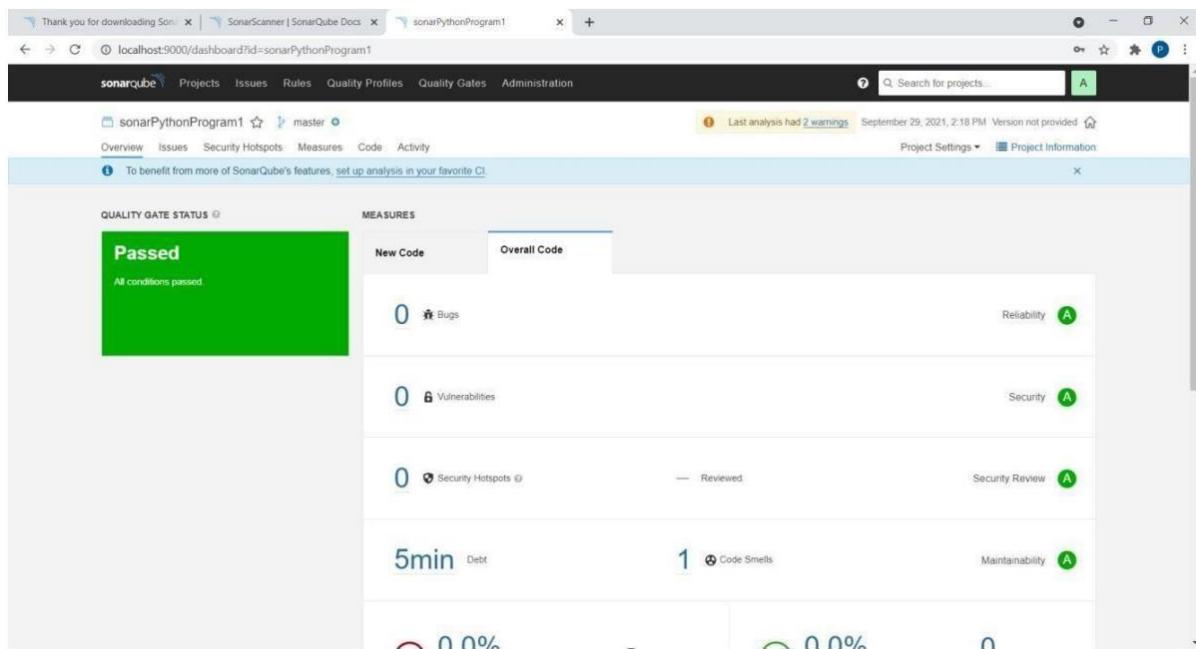


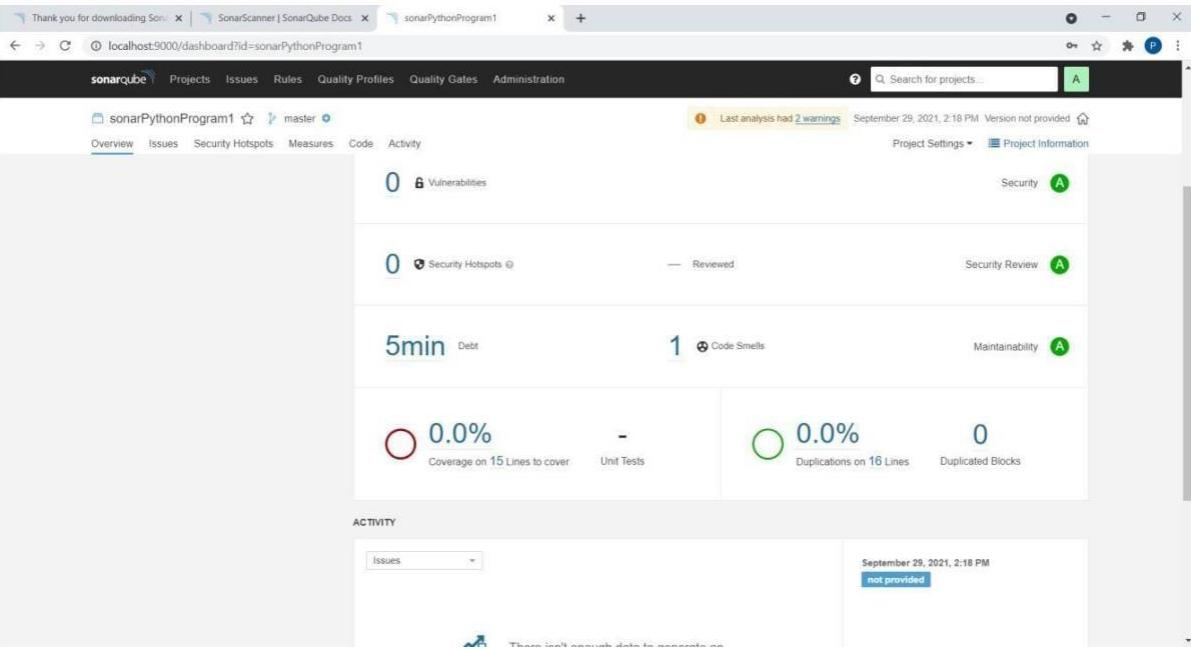
```
C:\Windows\System32\cmd.exe
Microsoft Windows [Version 10.0.19043.1237]
(c) Microsoft Corporation. All rights reserved.

C:\Users\Priyansi\Documents\SonarExps>sonar-scanner.bat -D"sonar.projectKey=sonarPythonProgram1" -D"sonar.sources=." -D"sonar.host.url=http://localhost:9000" -D"sonar.login=41740dddf269d680dfda1ec55f28cd250be46d4
INFO: Scanner configuration file: C:\Users\Priyansi\Downloads\sonar-scanner-4.6.2.2472-windows\bin..\conf\sonar-scanner.properties
INFO: Project root configuration file: NONE
INFO: SonarScanner 4.6.2.2472
INFO: Java 11.0.11 AdoptOpenJDK (64-bit)
INFO: Windows 10 10.0 amd64
INFO: User cache: C:\Users\Priyansi\.sonar\cache
INFO: Scanner configuration file: C:\Users\Priyansi\Downloads\sonar-scanner-4.6.2.2472-windows\bin..\conf\sonar-scanner.properties
INFO: Project root configuration file: NONE
INFO: Default locale: "en_US", source code encoding: "windows-1252" (analysis is platform dependent)
INFO: load global settings
INFO: load global settings (done) | time=203ms
INFO: Server id: BF41A1F2-AXwqPH4x91b8xeZLScu
INFO: User cache: C:\Users\Priyansi\.sonar\cache
INFO: load/download plugins
INFO: Load plugins index
INFO: Load plugins index (done) | time=102ms
INFO: Load/download plugins (done) | time=1674ms
INFO: Process project properties
INFO: Process project properties (done) | time=20ms
INFO: Execute project builders
INFO: Execute project builders (done) | time=2ms
INFO: Project key: sonarPythonProgram1
INFO: Base dir: C:\Users\Priyansi\Documents\SonarExps
INFO: Working dir: C:\Users\Priyansi\Documents\SonarExps\scannework
INFO: Load project settings for component key: 'sonarPythonProgram1'
INFO: Load project settings for component key: 'sonarPythonProgram1' (done) | time=40ms
INFO: Load quality profiles
INFO: Load quality profiles (done) | time=201ms
INFO: Load active rules
INFO: Load active rules (done) | time=452ms
WARN: SCM provider autodetection failed. Please use "sonar.scm.provider" to define SCM of your project, or disable the SCM Sensor in the project settings.
INFO: Indexing files...
INFO: Project configuration:
INFO: 1 file indexed
INFO: Quality profile for py: Sonar way
INFO: Quality profile for py: Run sensors on module sonarPythonProgram1
INFO: Load metrics repository
INFO: Load metrics repository (done) | time=37ms
INFO: Sensor Python Sensor [python]
WARN: Your code is analyzed as compatible with python 2 and 3 by default. This will prevent the detection of issues specific to python 2 or python 3. You can get a more precise analysis by setting a python version in your configuration via the parameter "sonar.python.version"
INFO: Starting global symbols computation
INFO: 1 source file to be analyzed
INFO: Load project repositories

C:\Windows\System32\cmd.exe
INFO: Sensor HTML [web] (done) | time=2ms
INFO: Sensor VB.NET Project Type Information [vbnet]
INFO: Sensor VB.NET Project Type Information [vbnet] (done) | time=1ms
INFO: Sensor VB.NET Analysis Log [vbnet]
INFO: Sensor VB.NET Analysis Log [vbnet] (done) | time=12ms
INFO: Sensor VB.NET Properties [vbnet]
INFO: Sensor VB.NET Properties [vbnet] (done) | time=0ms
INFO: Sensor Zero Coverage Sensor
INFO: Sensor Zero Coverage Sensor (done) | time=12ms
INFO: SCM Publisher No SCM system was detected. You can use the 'sonar.scm.provider' property to explicitly specify it.
INFO: CPD Executor Calculating CPD for 1 file
INFO: CPD Executor CPD calculation finished (done) | time=10ms
INFO: Analysis report generated in 59ms, dir size=103.9 kB
INFO: Analysis report compressed in 7ms, zip size=14.7 kB
INFO: Analysis report uploaded in 7ms
INFO: ANALYSIS SUCCESSFUL, you can browse http://localhost:9000/dashboard?id=sonarPythonProgram1
INFO: Note that you will be able to access the updated dashboard once the server has processed the submitted analysis report
INFO: More about the report processing at http://localhost:9000/api/ce/task?id=AXwvYlh9ibBxeZLXH1
INFO: Analysis total time: 7.502 s
INFO: -----
INFO: EXECUTION SUCCESS
INFO: -----
INFO: Total time: 10.007s
INFO: Final Memory: 79/50M
INFO: -----
```

Given below is the inspection of code quality to perform automatic reviews with static analysis of code to detect bugs, code smells, and security vulnerabilities.

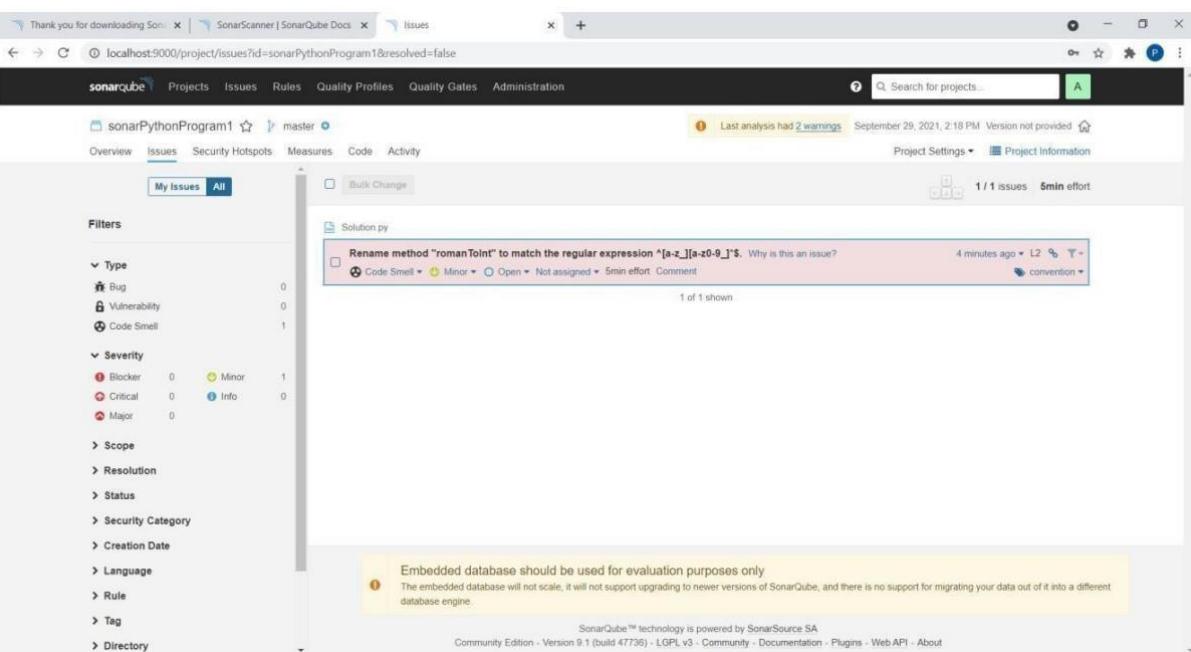




The screenshot shows the SonarQube dashboard for the project 'sonarPythonProgram1'. The dashboard provides an overview of the code quality with various metrics:

- Vulnerabilities:** 0
- Security Hotspots:** 0
- Debt:** 5min
- Code Smells:** 1
- Coverage:** 0.0% (Coverage on 15 Lines to cover)
- Duplications:** 0.0% (Duplications on 16 Lines)
- Maintainability:** 0

The 'ACTIVITY' section shows the last analysis was on September 29, 2021, at 2:18 PM, with 'not provided' details.

The screenshot shows the 'Issues' page for the same project. It displays a single issue found in 'Solution.py':

- Title:** Rename method "romanToInt" to match the regular expression ^[a-z\_][a-zA-Z0-9\_]\*\$.
- Type:** Code Smell
- Severity:** Minor
- Status:** Open
- Assignee:** Not assigned
- Effort:** 5min effort
- Comment:** Why is this an issue?

The page also includes a sidebar with filters for Type, Severity, Scope, Resolution, Status, Security Category, Creation Date, Language, Rule, Tag, and Directory. A note at the bottom states: "Embedded database should be used for evaluation purposes only. The embedded database will not scale, it will not support upgrading to newer versions of SonarQube, and there is no support for migrating your data out of it into a different database engine."

Press "Ctrl + C" to stop the server.

```

[1] Command Prompt
jvm 1 |      at org.sonar.application.process.EsManagedProcess.isOperational(EsManagedProcess.java:60)
jvm 1 |      at org.sonar.application.process.ManagedProcessHandler.refreshState(ManagedProcessHandler.java:220)
jvm 1 |      at org.sonar.application.process.ManagedProcessHandler$EventWatcher.run(ManagedProcessHandler.java:285)
jvm 1 | Caused by: java.util.concurrent.ExecutionException: java.util.concurrent.TimeoutException: connecting to [/127.0.0.1:9001]
jvm 1 |      at org.elasticsearch.common.util.concurrent.BaseFuture$Sync.getValue(BaseFuture.java:202)
jvm 1 |      at org.elasticsearch.common.util.concurrent.BaseFuture$Sync.get(BaseFuture.java:249)
jvm 1 |      at org.elasticsearch.common.util.concurrent.BaseFuture.get(BaseFuture.java:76)
jvm 1 |      at org.elasticsearch.client.RestHighLevelClient.performClientRequest(RestHighLevelClient.java:2075)
jvm 1 |      ... 10 common frames omitted
jvm 1 | Caused by: java.net.ConnectException: Timeout connecting to [/127.0.0.1:9001]
jvm 1 |      at org.apache.http.nio.pool.RouteSpecificPool.timeout(RouteSpecificPool.java:169)
jvm 1 |      at org.apache.http.nio.pool.AbstractMIOConnPool.requestTimeout(AbstractMIOConnPool.java:626)
jvm 1 |      at org.apache.http.nio.pool.AbstractMIOConnPool$RequestTimeout$1.run(AbstractMIOConnPool.java:894)
jvm 1 |      at org.apache.http.impl.nio.reactor.SessionRequestingImpl.timeout(SessionRequestingImpl.java:184)
jvm 1 |      at org.apache.http.impl.nio.reactor.DefaultConnectingIOReactor.processTimeouts(DefaultConnectingIOReactor.java:214)
jvm 1 |      at org.apache.http.impl.nio.reactor.DefaultConnectingIOReactor.processEvents(DefaultConnectingIOReactor.java:158)
jvm 1 |      at org.apache.http.impl.nio.reactor.AbstractMultiWorkerIOReactor.execute(AbstractMultiWorkerIOReactor.java:351)
jvm 1 |      at org.apache.http.impl.nio.conn.PoolingHttpClientConnectionManager.execute(PoolingHttpClientConnectionManager.java:221)
jvm 1 |      at org.apache.http.impl.nio.client.CloseableHttpAsyncClientBase$1.run(CloseableHttpAsyncClientBase.java:64)
jvm 1 |      at java.base/java.lang.Thread.run(Thread.java:834)
jvm 1 | 2021-09-29 13:50:50 INFO app[]|o.s.a.SchedulerImpl| Process[ce] is up
jvm 1 | 2021-09-29 13:50:50 INFO app[]|o.s.a.ProcessLauncherImpl| Launch process[[key='web', ipcIndex=2, logfilenamePrefix=web]] from [C:\Users\Priyansi\Downloads\sonarqube-9.1.0.47736]: C:\Program Files\Java\jdk-11.0.12\bin\java -Djava.awt.headless=true -Dfile.encoding=UTF-8 -Djava.io.tmpdir=C:\Users\Priyansi\Downloads\sonarqube-9.1.0.47736\tmp -XX:OmitStackTraceInFastThrow --add-opens=java.base/java.util=ALL-UNNAMED --add-opens=java.base/java.lang=ALL-UNNAMED --add-opens=java.base/java.io=ALL-UNNAMED --add-opens=java.rmi=sun.rmi.transport=ALL-UNNAMED --add-exports=java.base/jdk.internal.ref=ALL-UNNAMED --add-opens=java.base/sun.management=ALL-UNNAMED --add-opens=jdk.management/com.sun.management.internal=ALL-UNNAMED -Xms512m -Xmx128m -XX:+HeapDumpOnOutOfMemoryError -Dhttp.nonProxyHosts=localhost[127.*[:1]] -cp ./lib/sonar-application-9.1.0.47736.jar;C:\Users\Priyansi\Downloads\sonarqube-9.1.0.47736\lib\jdbch2\h2-1.4.199.jar org.sonar.server.app.webServer C:\Users\Priyansi\Downloads\sonarqube-9.1.0.47736\tmp\sq-process1779451691724418110\properties
jvm 1 | 2021-09-29 13:51:42 INFO app[]|o.s.a.SchedulerImpl| Process[web] is up
jvm 1 | 2021-09-29 13:51:42 INFO app[]|o.s.a.ProcessLauncherImpl| Launch process[[key='ce', ipcIndex=3, logfilenamePrefix=ce]] from [C:\Users\Priyansi\Downloads\sonarqube-9.1.0.47736]: C:\Program Files\Java\jdk-11.0.12\bin\java -Djava.awt.headless=true -Dfile.encoding=UTF-8 -Djava.io.tmpdir=C:\Users\Priyansi\Downloads\sonarqube-9.1.0.47736\tmp -XX:OmitStackTraceInFastThrow --add-opens=java.base/java.util=ALL-UNNAMED --add-exports=java.base/jdk.internal.ref=ALL-UNNAMED --add-opens=java.base/java.nio=ALL-UNNAMED --add-opens=java.base/sun.nio.ch=ALL-UNNAMED --add-opens=java.management/com.sun.management.internal=ALL-UNNAMED --add-opens=jdk.management/com.sun.management.internal=ALL-UNNAMED -Xms512m -Xmx128m -XX:+HeapDumpOnOutOfMemoryError -Dhttp.nonProxyHosts=localhost[127.*[:1]] -cp ./lib/sonar-application-9.1.0.47736.jar;C:\Users\Priyansi\Downloads\sonarqube-9.1.0.47736\lib\jdbch2\h2-1.4.199.jar org.sonar.ce.app.CeServer C:\Users\Priyansi\Downloads\sonarqube-9.1.0.47736\tmp\sq-process39444487414319983\properties
jvm 1 | 2021-09-29 13:51:42 WARN app[]|[startup] ######
jvm 1 | 2021-09-29 13:51:42 WARN app[]|[startup] Default Administrator credentials are still being used. Make sure to change the password or deactivate the account.
jvm 1 | 2021-09-29 13:51:42 WARN app[]|[startup] #####
jvm 1 | 2021-09-29 13:51:46 INFO app[]|o.s.a.SchedulerImpl| Process[ce] is up
jvm 1 | 2021-09-29 13:51:46 INFO app[]|o.s.a.SchedulerImpl| SonarQube is up
wrapper | ^CRL-C trapped. Shutting down...
jvm 1 | 2021-09-29 14:38:58 INFO app[]|o.s.a.SchedulerImpl| Stopping SonarQube
jvm 1 | 2021-09-29 14:38:58 INFO app[]|o.s.a.SchedulerImpl| Process[ce] is stopped
jvm 1 | 2021-09-29 14:38:58 INFO app[]|o.s.a.SchedulerImpl| Process[web] is stopped
jvm 1 | 2021-09-29 14:38:58 INFO app[]|o.s.a.SchedulerImpl| Process[es] is stopped
jvm 1 | 2021-09-29 14:38:58 INFO app[]|o.s.a.SchedulerImpl| SonarQube is stopped
wrapper | <-- Wrapper Stopped
Terminate batch job (Y/N)? y
C:\Users\Priyansi\Downloads\sonarqube-9.1.0.47736\bin\windows-x86-64>

```

## Conclusion:

In this assignment we implemented static analysis on python programs using SonarCube SAST processes.

# Assignment 8

Aim: To understand Continuous Monitoring using Nagios.

LO Mapped: LO5

Theory:

Login to Aws.

Create Ec2 instances on Aws account of any linux os

Then Run the following command in SS

```
ubuntu@ip-172-31-13-219:~$ sudo apt update
Hit:1 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu focal InRelease
Get:2 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu focal-updates InRelease [114 kB]
Get:3 http://security.ubuntu.com/ubuntu focal-security InRelease [114 kB]
Get:4 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu focal-backports InRelease [101 kB]
Get:5 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu focal/universe amd64 Packages [8628 kB]
Get:6 http://security.ubuntu.com/ubuntu focal-security/main amd64 Packages [909 kB]
Get:7 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu focal/universe Translation-en [5124 kB]
```

```
cd /usr/src/
```

```
sudo wget https://github.com/NagiosEnterprises/nagioscore/archive/nagios-4.4.2.tar.gz
```

```
ubuntu@ip-172-31-13-219:/usr/src$ sudo wget https://github.com/NagiosEnterprises/nagioscore/archive/nagios-4.4.2.tar.gz
--2021-10-05 10:24:05-- https://github.com/NagiosEnterprises/nagioscore/archive/nagios-4.4.2.tar.gz
Resolving github.com... 13.234.176.102
Connecting to github.com (github.com)|13.234.176.102|:443... connected.
HTTP request sent, awaiting response... 302 Found
Location: https://codeload.github.com/NagiosEnterprises/nagioscore/tar.gz/nagios-4.4.2 [following]
--2021-10-05 10:24:05-- https://codeload.github.com/NagiosEnterprises/nagioscore/tar.gz/nagios-4.4.2
Resolving codeload.github.com (codeload.github.com)|13.233.43.20|
Connecting to codeload.github.com (codeload.github.com)|13.233.43.20|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: unspecified [application/x-gzip]
Saving to: 'nagios-4.4.2.tar.gz'

nagios-4.4.2.tar.gz                                              [ <=>                               ] 10.78M 16.9MB/s   in 0.6s

2021-10-05 10:24:06 (16.9 MB/s) - 'nagios-4.4.2.tar.gz' saved [11301457]
```

```
sudo tar zxf nagios-* .tar.gz
```

```
cd nagioscore-nagios-*/
```

```
ubuntu@ip-172-31-13-219:/usr/src$ cd nagioscore-nagios-*/
ubuntu@ip-172-31-13-219:/usr/src/nagioscore-nagios-4.4.2$ █
```

Now finally run the following command

```
sudo ./configure --with-httpd-conf=/etc/apache2/sites-enabled
```

if error comes install c compiler on the linux by following this link

<https://linuxize.com/post/how-to-install-gcc-compiler-on-ubuntu-18->

[04/ Finally-](#)

```
4. Run the configure script.  
NOTE: If you can't get the configure script to recognize the GD libs  
on your system, get over it and move on to other things. The  
CGIs that use the GD libs are just a small part of the entire  
Nagios package. Get everything else working first and then  
revisit the problem. Make sure to check the nagios-users  
mailing list archives for possible solutions to GD library  
problems when you resume your troubleshooting.  
*****  
checking ltdl.h usability... no  
checking ltdl.h presence... no  
checking for ltdl.h... no  
checking dlfcn.h usability... yes  
checking dlfcn.h presence... yes  
checking for dlfcn.h... yes  
checking for dlopen in -ldl... yes  
checking for extra flags needed to export symbols... -Wl,-export-dynamic  
checking for linker flags for loadable modules... -shared  
checking for traceroute... no  
checking for type va_list... yes  
checking for perl... /usr/bin/perl  
checking for unzip... no
```

Now let us install plugins by

```
sudo wget -O nagios-plugins.tar.gz https://github.com/nagios-plugins/nagiosplugins/archive/release-2.2.1.tar.g
```

then

```
sudo tar zxf nagios-plugins.tar.gz
```

then

```
cd nagios-plugins-release-2.2.1 finally start
```

and then check status of nagi os

```
sudo systemctl start nagios
```

```
sudo systemctl status nagios
```

```
* nagios.service - Nagios Core 4.4.2
   Loaded: loaded (/lib/systemd/system/nagios.service; enabled; vendor preset: enabled)
   Active: active (running) since Fri 2018-11-16 14:54:21 PST; 1s ago
     Docs: https://www.nagios.org/documentation
  Process: 18294 ExecStopPost=/bin/rm -f /usr/local/nagios/var/rw/nagios.cmd (code=exited, status=0/SUCCESS)
  Process: 18293 ExecStop=/bin/kill -s TERM ${MAINPID} (code=exited, status=0/SUCCESS)
  Process: 18315 ExecStart=/usr/local/nagios/bin/nagios -d /usr/local/nagios/etc/nagios.cfg (code=exited, status=0/SUCCESS)
  Process: 18313 ExecStartPre=/usr/local/nagios/bin/nagios -v /usr/local/nagios/etc/nagios.cfg (code=exited, status=0/SUCCESS)
 Main PID: 18325 (nagios)
    Tasks: 6 (limit: 2319)
   CGroup: /system.slice/nagios.service
```

Steps-

Go to google.com, Search Nagios Demo

Click on the first link shown below

The screenshot shows a Google search results page. The search query "nagios demo" is entered in the search bar. Below the search bar, there are navigation links for All, Videos, Images, News, Shopping, More, and Tools. The "All" link is underlined, indicating it is the selected category. The search results section displays the following information:

About 3,87,000 results (0.44 seconds)

<https://exchange.nagios.org> › directory › Demos › details

**Nagios XI Online Demo**

An online **demo** of Nagios XI. The **demo** allows you to test configuration wizards, dashlets, dashboards, views, and more. Reviews (0).

<https://exchange.nagios.org> › directory › Demos

**Demos - Nagios Exchange**

An online **demo** of Nagios Log Server. The **demo** allows you to view system logs and event logs, giving some examples on how you can visualize data sent into Nagios ...

<https://exchange.nagios.org> › directory › Demos › details

Now click on the website-

**Nagios®**

Network: Enterprise | Support | Library | Project | Exchange

Home    Directory    About

Home | Directory | Demos | Nagios XI Online Demo

### Directory Tree

### Search Exchange


[Advanced Search](#)

## Nagios XI Online Demo

[Submit review](#) | [Recommend](#) | [Print](#) | [Visit](#) | [Claim](#)

Rating		Favoured: 0
0 votes		
Owner <a href="#">egalstad</a>		
Website <a href="http://nagiosxi.demos.nagios.com">nagiosxi.demos.nagios.com</a>		
Hits 141800		

Now click on login as administrator

The screenshot shows the Nagios XI login interface. On the left is the login form with fields for Username and Password, and a 'Forgot your password?' link. Below the form is a 'Select Language:' dropdown with various flags. On the right, there's a banner for 'Nagios XI Demo System'. Underneath it, the 'Demo Account Options' section lists five types of user access with their respective log-in buttons and credentials:

- Administrator Access**: Username: nagiosadmin, Password: nagiosadmin
- Read-Only User Access**: Username: readonly, Password: readonly
- Advanced User Access**: Username: advanced, Password: advanced
- Normal User Access**: Username: jdoe, Password: jdoe
- Administrator Access - showing the dark theme.**: Username: darktheme, Password: darktheme

It will have interface like this

**Nagios XI**

- Home Dashboard
- Tactical Overview
- Birdseye
- Operations Center
- Operations Screen
- Open Service Problems
- Open Host Problems
- All Service Problems
- All Host Problems
- Network Outages

- Service Status
- Host Status
- Hostgroup Summary
- Hostgroup Overview
- Hostgroup Grid
- Servicegroup Summary
- Servicegroup Overview
- Servicegroup Grid
- BPI
- Metrics

- Performance Graphs
- Graph Explorer

- World Map
- BMap
- Hypermap
- Mnemmap
- NagVis
- Network Status Map

- Initial Setup Tasks:
- Configure mail settings

**Home Dashboard**

**Getting Started Guide**

**Common Tasks:**

- Change your account settings
- Change your account password and general preferences
- Change your notifications settings
- Change how and when you receive alert notifications.
- Configure your Monitoring setup
- Add or modify items to be monitored with easy-to-use wizards.

**Host Status Summary**

Up	Down	Unreachable	Pending
50	0	0	6
3	3	3	59

Last Updated: 2021-10-05 05:06:48

**Service Status Summary**

OK	Warning	Unknown	Critical	Pending
764	100	5	2	7
236	236	3	1007	

Last Updated: 2021-10-05 05:06:48

**We're Here To Help!**

Our knowledgeable techs are happy to help you with any questions or problems you may have getting Nagios up and running.

Support Forum / Customer Support Forum  
Help Resources  
Customer Ticket Support Center  
Customer Phone Support: +1 651-204-9102 Ext. 4

**Start Monitoring**

Run a Config Wizard

Run Auto-Discovery

Advanced Config

**About | Legal | Copyright © 2008-2021 Nagios Enterprises, LLC**

Now click on Host status-

**Nagios XI**

- Home Dashboard
- Tactical Overview
- Birdseye
- Operations Center
- Operations Screen
- Open Service Problems
- Open Host Problems
- All Service Problems
- All Host Problems
- Network Outages

- Service Status
- Host Status
- Hostgroup Summary
- Hostgroup Overview
- Hostgroup Grid
- Servicegroup Summary
- Servicegroup Overview
- Servicegroup Grid
- BPI
- Metrics

- Performance Graphs
- Graph Explorer

- World Map
- BMap
- Hypermap
- Mnemmap
- NagVis
- Network Status Map

**Host Status**

All hosts

Showing 1-59 of 59 total records

Host	Status	Duration	Attempt	Last Check	Status Information
europa nagios.local	Down	426d 19h 2m 42s	5/5	2021-10-05 05:04:53	CRITICAL - 192.168.4.54: Host unreachable @ 192.168.5.66. rta nan, lost 100%
www.acme.com	Down	1190d 17h 28m 49s	5/5	2021-10-05 05:05:20	CRITICAL - 216.27.178.28. rta nan, lost 100%
www.chaoticmoon.com	Down	851d 16h 42m 45s	5/5	2021-10-05 05:05:50	check_http: Invalid hostname/address - www.chaoticmoon.com
Firewall	Up	1190d 17h 28m 11s	1/10	2021-10-05 05:02:49	OK - 127.0.0.1 rta 0.020ms lost 0%
Log-Server.nagios.local	Up	2275d 8h 1m 2s	1/5	2021-10-05 05:05:22	OK - localhost rta 0.022ms lost 0%
Log-Server2.nagios.local	Up	1180d 14h 8m 21s	1/5	2021-10-05 05:06:53	OK - localhost rta 0.026ms lost 0%
NOAA	Pending	3763d 12h 58m 36s	1/3	2012-01-02 09:43:01	HTTP OK HTTP/1.1 200 OK - 99753 bytes in 0.478 seconds
Netw	Pending	N/A	1/5	N/A	No check results for host yet...
Network-Analyzer.nagios	Pending	N/A	1/5	N/A	No check results for host yet...
Network-Analyzer.nagios.local	Up	2275d 7h 58m 0s	1/5	2021-10-05 05:07:12	OK - localhost rta 0.021ms lost 0%
Network-Analyzer2.	Pending	N/A	1/5	N/A	No check results for host yet...
Network-Analyzer2.nagios.local	Up	2275d 7h 57m 50s	1/5	2021-10-05 05:06:42	OK - localhost rta 0.021ms lost 0%
Router	Up	1190d 17h 31m 9s	1/10	2021-10-05 05:03:25	OK - 127.0.0.1 rta 0.020ms lost 0%

**Host Status Summary**

Up	Down	Unreachable	Pending
50	0	0	6
3	3	3	59

Last Updated: 2021-10-05 05:07:25

**Service Status Summary**

OK	Warning	Unknown	Critical	Pending
764	100	5	2	7
236	236	3	1007	

Last Updated: 2021-10-05 05:07:25

**About | Legal | Copyright © 2008-2021 Nagios Enterprises, LLC**

In the above image one can see Host Status Summary and Service Status Summary also how many host are up, down and also errors in detail Now click on Host Group Status.

**Host Status Summary**

Up	Down	Unreachable	Pending
30	0	0	6
Unhandled	Problems	All	
4	3	59	

Last Updated: 2021-10-05 05:09:55

**Service Status Summary**

Ok	Warning	Unknown	Critical	Pending
761	99	5	7	
Unhandled	Problems	All		
239	239	1007		

Last Updated: 2021-10-05 05:09:55

**Status Summary For All Host Groups**

Host Group	Hosts	Services
Monitoring Servers (Monitoring Servers)	5 Up	93 Ok 14 Warning 2 Critical
Hostgroup Two (hg2)	1 Up	11 Ok 4 Warning 2 Unknown 8 Critical
Some Other Hostgroup (hg3)	2 Up	11 Ok 1 Warning 2 Critical
Linux Servers (linux-servers)	11 Up	210 Ok 27 Warning 2 Unknown 13 Critical
Network Devices (network-devices)	7 Up	215 Ok 35 Warning 62 Critical

Here we can see Status Summary For All Host Groups

Now we click on BBMap

In this we can see status of following stuff in each host-



Now we have Network status map which is graphical representation of the network status

The screenshot displays the Nagios XI interface with the 'Network Status Map' selected in the left sidebar. The main area shows a complex network topology with numerous nodes representing hosts and services. A legend on the right side of the map indicates node colors based on status: green for healthy, red for critical, yellow for warning, and grey for unknown or unavailable. The map is densely populated with nodes, including several 'Nagios' hosts, a 'Nagios Process' central node, and many external services like 'www.google.com', 'www.facebook.com', and 'www.twitter.com'. The left sidebar contains a navigation menu with sections like 'Details', 'Metrics', 'Graphs', 'Maps', and 'Incident Management'.

Conclusion: In this assignment we learned about how to continuously monitor Nagios commands

## ASSIGNMENT 9

AIM: To understand Lambda Function and create a Lambda function which will log "An Image has been added" once you add an object to a specific bucket in S3.

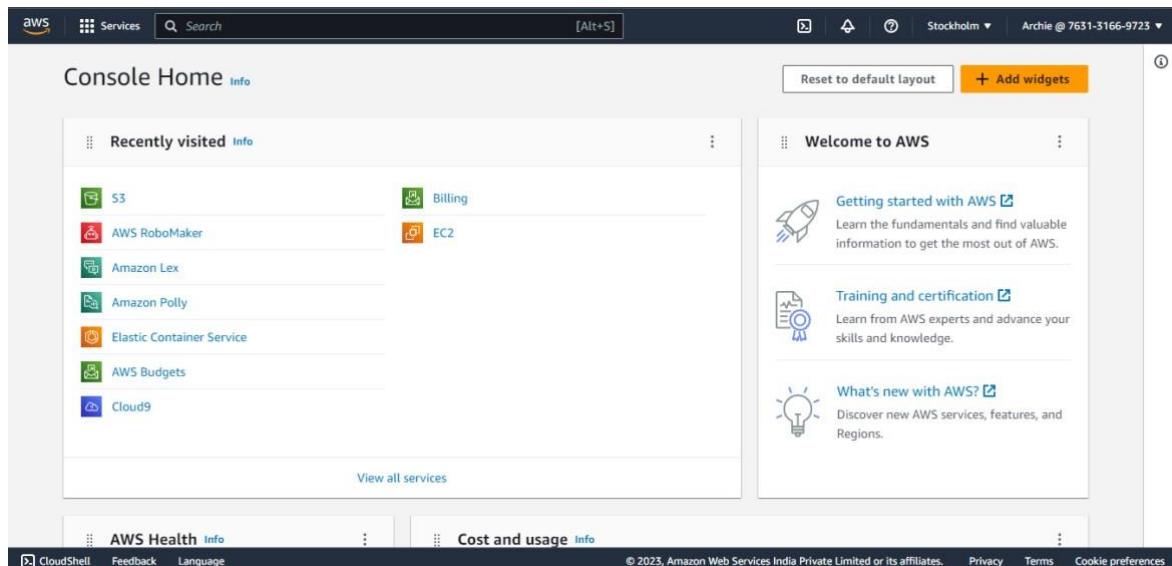
LO MAPPED: LO1, LO5

### THEORY:

You can use Lambda to process event notifications from Amazon Simple Storage Service. Amazon S3 can send an event to a Lambda function when an object is created or deleted. You configure notification settings on a bucket, and grant Amazon S3 permission to invoke a function on the function's resource-based permissions policy.

### STEPS TO FOLLOW:

1. Log in as IAM User



2. Create a S3 bucket and Enable the "Block all public access"

**Block Public Access settings for this bucket**

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

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

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

**Successfully created bucket "labdabucket117"**  
To upload files and folders, or to configure additional bucket settings choose [View details](#).

View details (1)

Amazon S3 > Buckets

**Account snapshot**  
Storage lens provides visibility into storage usage and activity trends. [Learn more](#)

**Buckets (1) Info**  
Buckets are containers for data stored in S3. [Learn more](#)

Name	AWS Region	Access	Creation date
labdabucket117	Europe (Stockholm) eu-north-1	Objects can be public	August 30, 2023, 22:56:19 (UTC-07:00)

CloudShell Feedback Language © 2023, Amazon Web Services India Private Limited or its affiliates. Privacy Terms Cookie preferences

### 3. Search IAM on the console and go to "Roles". Click on Create Roles

**Identity and Access Management (IAM)**

Search IAM

**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 activity
AWSCloud9SSMAccessRole	AWS Service: cloud9, and 1 more. <a href="#">Edit</a>	29 days ago
AWSServiceRoleForAWSCloud9	AWS Service: cloud9 (Service-Linked Role) <a href="#">Edit</a>	29 days ago
AWSServiceRoleForSupport	AWS Service: support (Service-Linked Role) <a href="#">Edit</a>	-
AWSServiceRoleForTrustedAdvisor	AWS Service: trustedadvisor (Service-Linked Role) <a href="#">Edit</a>	-

**Roles Anywhere** [Info](#)  
Authenticate your non AWS workloads and securely provide access to AWS services.

Manage

CloudShell Feedback Language © 2023, Amazon Web Services India Private Limited or its affiliates. Privacy Terms Cookie preferences

### 4. Select the options of "AWS service" and "lambda"

**Step 2**  
Add permissions

**Step 3**  
Name, review, and create

**Trusted entity type**

- AWS service Allow AWS services like EC2, Lambda, or others to perform actions in this account.
- AWS account Allow entities in other AWS accounts belonging to you or a 3rd party to perform actions in this account.
- Web identity Allows users federated by the specified external web identity provider to assume this role to perform actions in this account.
- SAML 2.0 federation Allow users federated with SAML 2.0 from a corporate directory to perform actions in this account.
- Custom trust policy Create a custom trust policy to enable others to perform actions in this account.

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

Common use cases

- EC2 Allows EC2 instances to call AWS services on your behalf.
- Lambda Allows Lambda functions to call AWS services on your behalf.

Use cases for other AWS services:

Choose a service to view use case

## 5. Enable the "CloudWatchFullAccess" and "AmazonS3FullAccess"

**Step 2**  
Add permissions

**Step 3**  
Name, review, and create

**Permissions policies (Selected 2/874)** Info  
Choose one or more policies to attach to your new role.

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

"cloud" X Clear filters

Policy name	Type	Description
AWSCloudFormationFullAccess	AWS managed	Provides access to AWS CloudFormation via the AWS Management Console.
CloudFrontFullAccess	AWS managed	Provides full access to the CloudFront console plus the ability to list Amazon CloudFront distributions.
AWSCloudHSMFullAccess	AWS managed	Provides full access to all CloudHSM resources.
AWSCloudHSMReadOnlyAccess	AWS managed	Provides read only access to all CloudHSM resources.
CloudFrontReadonlyAccess	AWS managed	Provides access to CloudFront distribution configuration information and lists.
CloudSearchFullAccess	AWS managed	Provides full access to the Amazon CloudSearch configuration service.
CloudSearchReadOnlyAccess	AWS managed	Provides read only access to the Amazon CloudSearch configuration service.
<input checked="" type="checkbox"/> CloudWatchFullAccess	AWS managed	Provides full access to CloudWatch.

**Step 1**  
Select trusted entity

**Step 2**  
Add permissions

**Step 3**  
Name, review, and create

**Add permissions** Info

**Permissions policies (Selected 2/874)** Info  
Choose one or more policies to attach to your new role.

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

"s3" X Clear filters

Policy name	Type	Description
<input checked="" type="checkbox"/> AmazonS3FullAccess	AWS managed	Provides full access to all buckets via the AWS Management Console.
AmazonS3ReadOnlyAccess	AWS managed	Provides read only access to all buckets via the AWS Management Console.
AmazonDMSRedshiftFullAccess	AWS managed	Provides access to manage S3 settings for Redshift endpoints for DMS.
QuickSightAccessForS3	AWS managed	Policy used by QuickSight team to access customer data produced by S3 ...
AmazonS3OutpostsFullAccess	AWS managed	Provides full access to Amazon S3 on Outposts via the AWS Management ...
AmazonS3OutpostsReadOnlyAccess	AWS managed	Provides read only access to Amazon S3 on Outposts via the AWS Manag ...

## 6. Click on Create Role and you will be redirected to this dashboard. Give the Role a Name and Click on Done.

Step 1  
Select trusted entity

Step 2  
Add permissions

Step 3  
Name, review, and create

**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  "Version": "2012-10-17",  
2

© 2023, Amazon Web Services India Private Limited or its affiliates. Privacy Terms Cookie preferences

## 7. Search for Lambda in the console and click on Create Function.

Compute

# AWS Lambda

lets you run code without thinking about servers.

You pay only for the compute time that you consume — there is no charge when your code is not running. With Lambda, you can run code for virtually any type of application or backend service, all with zero administration.

**Get started**

Author a Lambda function from scratch, or choose from one of many preconfigured examples.

**Create a function**

**How it works**

Run Next: Lambda responds to events

.NET | Go | Java | **Node.js** | Python | Ruby | Custom runtime

https://eu-north-1.console.aws.amazon.com/lambda/home?region=eu-north-1#/create/function?firstrun=true

© 2023, Amazon Web Services India Private Limited or its affiliates. Privacy Terms Cookie preferences

## 8. Change the settings of the Lambda Function.

Lambda > Function > Create function

**Create function**

Choose one of the following options to create your function.

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.

Browse serverless app repository  
Deploy a sample Lambda application from the AWS Serverless Application Repository.

**Basic information**

**Function name**  
Enter a name that describes the purpose of your function.  
  
Use only letters, numbers, hyphens, or underscores with no spaces.

**Runtime**  
Choose the language to use for writing your function. Note that the console code editor supports only Node.js, Python, and Ruby.

**Architecture**  
Choose the instruction set architecture you want for your function code.  
 x86\_64  
 arm64

**Permissions**

By default, Lambda will create an execution role with permissions to invoke logs in Amazon CloudWatch Logs. You can customize this default role later when adding triggers.

**Change default execution role**

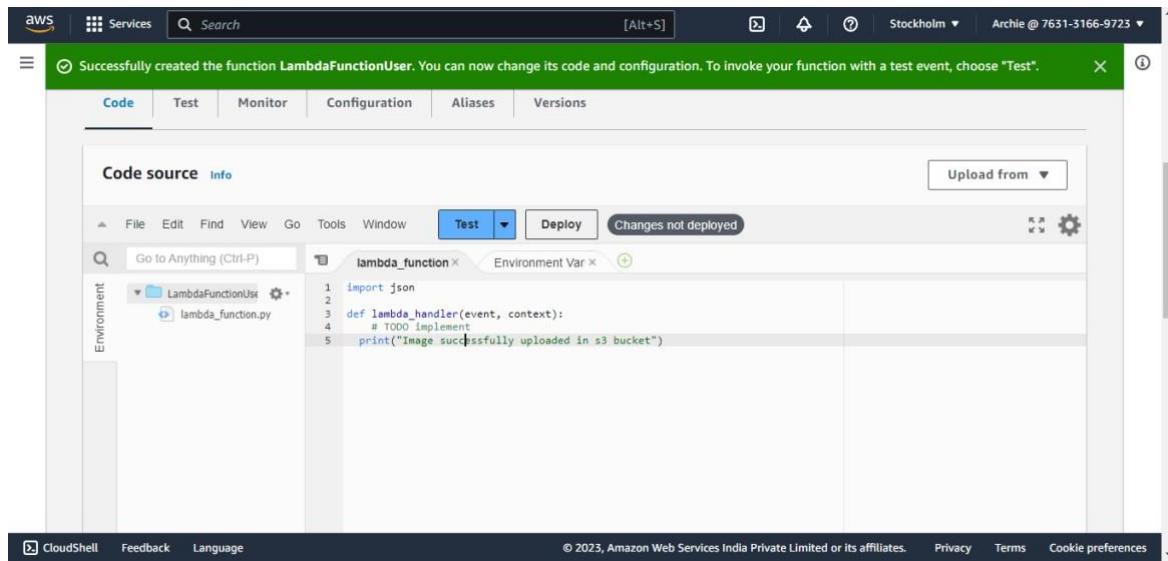
**Execution role**  
Choose a role that defines the permissions of your function. To create a custom role, go to the IAM console.

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

CloudShell Feedback Language

© 2023, Amazon Web Services India Private Limited or its affiliates. Privacy Terms Cookie preferences

9. Add the python code and click on deploy to save the changes.

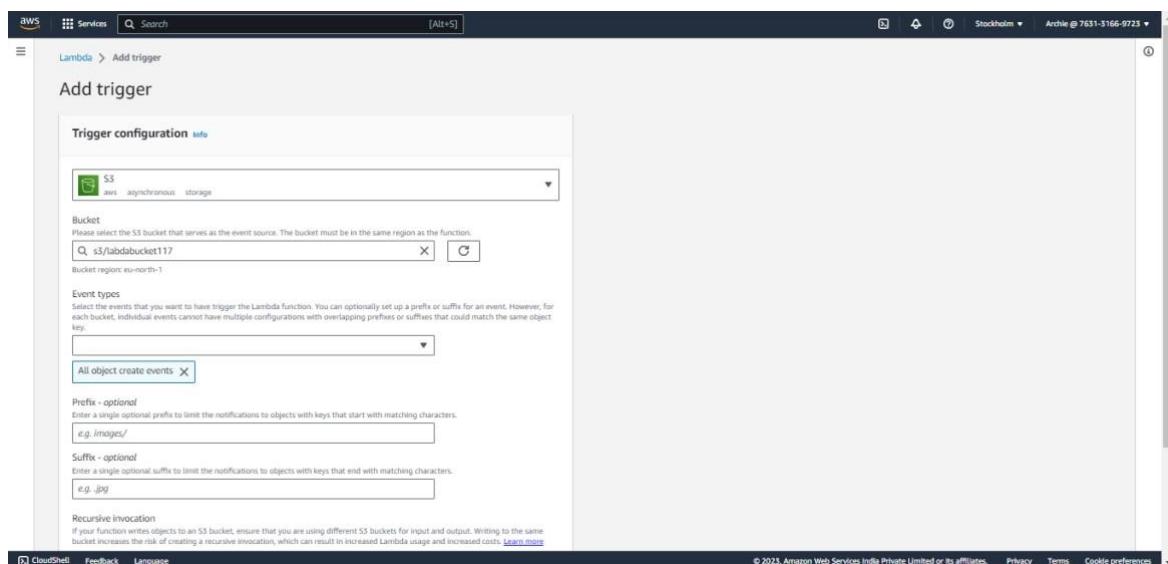


The screenshot shows the AWS Lambda function configuration interface. At the top, a green banner says "Successfully created the function LambdaFunctionUser. You can now change its code and configuration. To invoke your function with a test event, choose 'Test'." Below the banner, there are tabs for Code, Test, Monitor, Configuration, Aliases, and Versions. The Code tab is selected. The main area shows a code editor with the following Python code:

```
1 import json
2
3 def lambda_handler(event, context):
4     # TODO implement
5     print("Image successfully uploaded in s3 bucket")
```

Below the code editor, there are buttons for Upload from, Test, Deploy, and Changes not deployed. The Deploy button is highlighted in blue. On the left, there's a sidebar for Environment variables and a CloudWatch Metrics icon. At the bottom, there are links for CloudShell, Feedback, Language, and a footer with copyright information and privacy terms.

10. Scroll above and click on ADD TRIGGER. Select the following options and click on Done.



The screenshot shows the AWS Lambda trigger configuration interface. At the top, it says "Lambda > Add trigger". The main section is titled "Add trigger" and "Trigger configuration". It shows a dropdown menu set to "S3" and a search bar containing "s3://abdbucket117". Below this, there are sections for "Event types", "Prefix - optional", "Suffix - optional", and "Recursive invocation". The "Event types" section has a dropdown menu with "All object create events" selected. The "Prefix" field contains "e.g. images/" and the "Suffix" field contains "e.g. .jpg". The "Recursive invocation" note at the bottom states: "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." At the bottom, there are links for CloudShell, Feedback, Language, and a footer with copyright information and privacy terms.

11. Go to S3 bucket and click on Add files. Select a image and click on Upload.

Screenshot of the AWS S3 console showing the 'labdabucket117' bucket. The 'Objects' tab is selected, showing 0 objects. The 'Upload' button is highlighted.

**Buckets**

- Access Points
- Object Lambda Access Points
- Multi-Region Access Points
- Batch Operations
- IAM Access Analyzer for S3

Block Public Access settings for this account

**Storage Lens**

- Dashboards
- AWS Organizations settings

Feature spotlight 7

CloudShell Feedback Language © 2023, Amazon Web Services India Private Limited or its affiliates. Privacy Terms Cookie preferences

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 [\[Link\]](#)

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

**Files and folders (1 Total, 266.2 KB)**

<input type="checkbox"/>	Name	Folder	Type	Size
<input type="checkbox"/>	Screenshot (3).png	-	image/png	266.2 KB

**Destination**

Destination  
s3://labdabucket117

▶ Destination details  
Bucket settings that impact new objects stored in the specified destination.

CloudShell Feedback Language © 2023, Amazon Web Services India Private Limited or its affiliates. Privacy Terms Cookie preferences

**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://labdabucket117	<span style="color: green;">Succeeded</span> 1 file, 266.2 KB (100.00%)	<span style="color: gray;">Failed</span> 0 files, 0 B (0%)

**Files and folders** Configuration

**Files and folders (1 Total, 266.2 KB)**

CloudShell Feedback Language © 2023, Amazon Web Services India Private Limited or its affiliates. Privacy Terms Cookie preferences

12. Search CloudWatch and go to Log groups. Select the existing Log Group.

The screenshot shows the AWS CloudWatch interface. The left sidebar is titled "CloudWatch" and includes sections for Favorites and recent items, Dashboards, Alarms, Logs (selected), Log groups (selected), Metrics, X-Ray traces, Events, Application monitoring, and Insights. The main content area is titled "Log groups (1)" and shows a single log group named "/aws/lambda/LambdaFunctionUser". Below the log group name are dropdown menus for Data protection, Sensitivity, Retention, and Metric filters, all set to their default values. A search bar at the top allows filtering by prefix search or exact match. Buttons for Actions, View in Logs Insights, Start tailing, and Create log group are available. The bottom of the screen includes standard AWS navigation links for CloudShell, Feedback, Language, and copyright information.

13. Click on the link provided and then you will see the message displayed.

The image contains three screenshots of the AWS CloudWatch interface, illustrating the process of viewing log streams and events for a specific log group.

- Screenshot 1:** Shows the "Log groups" page with one entry: "Log group: /aws/lambda/LambdaFunctionUser". It displays metrics like creation time (1 minute ago), retention (Never expire), and subscription filters (0).
- Screenshot 2:** Shows the "Log streams" page for the selected log group. It lists one log stream: "2023/08/31/[\$LATEST]97bf6ca2f8dd4cf2b383b55caf775fe...". The timestamp for the last event is shown as "2023-08-30 23:08:48 (UTC-07:00)".
- Screenshot 3:** Shows the "Log events" page for the selected log stream. It lists several log entries with timestamps and messages. Key messages include:
  - "INIT\_START Runtime Version: python3.11.v10 Runtime Version ARN: arn:aws:lambda:..."
  - "START RequestId: d87be3bc-640d-4d9a-864d-2d10c71bc4a9 Version: \$LATEST"
  - "Image successfully uploaded in s3 bucket"
  - "END RequestId: d87be3bc-640d-4d9a-864d-2d10c71bc4a9 Duration: 1.51 ms Billed D..."

**CONCLUSION:** In this assignment, we learnt how to create a Lambda function which will log "An Image has been added" once you add an object to a specific bucket in S3.

## Assignment No. 10

Aim : Students are expected to study AWS Lambda function and create the following Lambda functions using Python:

To add data to Dynamo DB database.

Theory :

Creating an AWS Lambda function to add data to a DynamoDB database using Python is a common use case for serverless applications. You can use AWS Lambda and the AWS SDK for Python (Boto3) to achieve this. Here's a step-by-step guide:

Set up Your Environment:

Ensure you have an AWS account and access to the AWS Management Console.

Install the AWS Command Line Interface (CLI) and configure it with your AWS credentials.

Create a DynamoDB Table:

In the AWS DynamoDB Management Console, create a table with the desired schema. Note the table name and primary key (usually the partition key).

Create a Lambda Function:

Go to the AWS Lambda Management Console.

Click the "Create function" button.

Choose a blueprint or create a custom function.

Configure the function with a name, runtime (Python 3.8 or later is recommended), and execution role.

Write the Lambda Function Code:

In the "Function code" section of the Lambda function, you can write Python code to add data to the DynamoDB table. Here's a simple example using Boto3:

```
python Copy
code import
os import
boto3

# Initialize the AWS SDK
dynamodb = boto3.resource('dynamodb')

def lambda_handler(event, context):
    # Retrieve data to be added to DynamoDB from the event
    data = {
        'id': 'unique_id',
        'name': 'John Doe',
        'email': 'john@example.com'
    }

    # Specify the DynamoDB table name
    table_name = 'your-dynamodb-table-name'

    # Get the DynamoDB table
    table =
        dynamodb.Table(table_name)

    # Add the data to DynamoDB
```

```
table.put_item(Item=data)
```

This code adds a new item to the specified DynamoDB table.

#### Set Up Required Permissions:

Ensure that the Lambda function's execution role has the necessary permissions to interact with DynamoDB. You can create a role with appropriate policies using AWS IAM.

#### Test the Function:

You can test the Lambda function by invoking it manually with sample data.

#### Deploy the Function:

Save and deploy your Lambda function.

#### Monitor and Troubleshoot:

You can monitor the function's execution and review logs in the AWS CloudWatch Logs to ensure it's working as expected.

#### Additional Configuration:

Depending on your use case, you can extend the Lambda function to add more complex data to DynamoDB or trigger it using AWS services like S3, API Gateway, or other event sources.

Make sure to replace the example code with your specific DynamoDB table name and data. This Lambda function will add the specified data to the DynamoDB table whenever it's invoked.

#### Conclusion :

In the following we learnt how to use lambda functions to create mongodb table.

## Adv. DevOps Written Assignment : 01

### **1. what security measures can be taken while using Kubernetes?**

1. Role-Based Access Control (RBAC): RBAC restricts who can perform actions within a Kubernetes cluster. It defines roles and role bindings to specify what resources and operations users or service accounts can access. This prevents unauthorized access and actions within the cluster.
  
2. Regular Updates: Keeping Kubernetes and its components up to date is crucial. New releases often include security patches. Regular updates help mitigate known vulnerabilities and ensure your cluster remains secure.
  
3. Network Policies: Network policies allow you to define rules for communication between pods. By specifying which pods can communicate with each other, you can limit the attack surface and prevent unauthorized access.
  
4. Container Security Tools: Employ container security tools like vulnerability scanners to assess the security of container images. These tools can identify and remediate vulnerabilities in the containerized applications before they are deployed.
  
5. Monitoring and Audit: Implement monitoring and auditing solutions to track cluster activity. This helps detect and respond to suspicious or unauthorized behavior. Tools like Prometheus and Grafana can be used for monitoring, while audit logs provide insights into cluster activity.
  
6. Secrets Management: Sensitive data like API keys, passwords, and certificates should be stored securely using Kubernetes secrets or external vaults. This prevents sensitive information from being exposed within containers or configuration files.

7. PodSecurityPolicies (PSP): PSP is a Kubernetes feature that enforces security policies at the pod level. It allows you to define restrictions on privilege escalation, host access, and other security-sensitive configurations for pods.

8. Namespaces: Use Kubernetes namespaces to logically isolate workloads. This provides a level of separation between different applications or teams, reducing the risk of unauthorized access or interference between them.

9. Admission Controllers: Admission controllers are webhook plugins that intercept and validate requests to the Kubernetes API server. You can use them to enforce custom policies and ensure that only compliant resources are admitted to the cluster.

10. Container Runtime Security: Implement container runtime security solutions like Docker Security Scanning or container runtime protection tools. These tools monitor containers at runtime for abnormal behavior, helping to detect and respond to potential threats.

Combining these measures into a comprehensive security strategy is essential for safeguarding your Kubernetes cluster and the applications running within it. It's important to stay informed about best practices and evolving security threats in the Kubernetes ecosystem.

## **2. What are the three security techniques that can be used to protect data?**

Three security techniques commonly used to protect data are:

**1. Encryption:** Encryption is the process of converting data into a secure format that can only be read by someone with the decryption key. It ensures that even if unauthorized parties access the data, they cannot understand it without the correct key. Two common types of encryption are:

- Data-at-rest Encryption: Protects data when it's stored on disk or in a database.
- Data-in-transit Encryption: Secures data as it's transmitted between systems over networks.

**2. Access Control:** Access control mechanisms regulate who can access data and what actions they can perform on it. This involves setting permissions, roles, and policies to ensure that only authorized users or applications can access and manipulate data. Role-Based Access

Control (RBAC) and Attribute-Based Access Control (ABAC) are commonly used access control models.

**3. Data Masking/Redaction:** Data masking or redaction involves obscuring or replacing sensitive data with fictitious or scrambled values. This is often used in non-production environments or when sharing data with third parties. It ensures that even if someone gains access to the data, they cannot see the actual sensitive information.

These techniques are often used in combination to create a layered approach to data security, providing multiple levels of protection to safeguard sensitive information from unauthorized access and disclosure.

### **3. How do you expose a service using ingress in Kubernetes?**

To expose a service using Ingress in Kubernetes, you need to follow these steps:

**1. Set up Kubernetes:** Ensure you have a Kubernetes cluster up and running, and you have the `kubectl` command-line tool configured to communicate with the cluster.

**2. Deploy Your Application:** Deploy your application as a Kubernetes Deployment or a Pod, and create a Kubernetes Service to expose it internally within the cluster. This Service will be the target for the Ingress.

**3. Install an Ingress Controller:** You need to have an Ingress controller installed in your cluster. Some popular options include Nginx Ingress Controller, Traefik, or HAProxy Ingress. The controller will manage the Ingress resources and configure the load balancer.

For example, to install the Nginx Ingress Controller, you can use:

```
```bash
```

```
kubectl apply -f https://raw.githubusercontent.com/kubernetes/ingress-nginx/controllerv1.0.0/deploy/static/provider/cloud/deploy.yaml
```

```
```
```

**4. Create an Ingress Resource:** Define an Ingress resource that specifies the rules for routing traffic to your service. Here's an example Ingress resource manifest:

```
```yaml          apiVersion:  
networking.k8s.io/v1      kind:  
  
Ingress  metadata:  
  name: my-ingress  
  
spec:  rules:  
  - host: example.com  
    http:      paths:  
      - path: /path  
    pathType: Prefix  
  
  backend:  
    service:  
      name: your-service  
      port:  
        number: 80  
    ...
```

In this example, traffic for `example.com/path` will be routed to `your-service`.

**5. Apply the Ingress Resource:** Use `kubectl apply` to create the Ingress resource in your cluster:

```
```bash  kubectl apply -f your-ingress.yaml
```

```

**6. Configure DNS:** Ensure that the DNS records for the specified hostname (e.g., `example.com`) point to the external IP address of your Ingress controller.

**7. Access Your Service:** After DNS propagation, you should be able to access your service externally via the hostname and path you defined in the Ingress resource.

#### **4. Which service protocols does Kubernetes ingress expose?**

Kubernetes Ingress is primarily designed to expose HTTP and HTTPS services, making it suitable for routing and load balancing web traffic. However, with the evolution of Kubernetes and Ingress controllers, it has expanded to support additional protocols and features:

- 1. HTTP:** Ingress is commonly used to expose HTTP services. You can define routing rules based on URL paths, hostnames, and other HTTP attributes.
- 2. HTTPS:** Secure HTTP services can be exposed through Ingress by configuring TLS certificates. This allows you to terminate SSL/TLS encryption at the Ingress controller and route decrypted traffic to your services.
- 3. TCP:** Some Ingress controllers, like Nginx Ingress, support TCP services. This enables you to expose non-HTTP services such as databases or custom protocols. TCP-based routing typically relies on port numbers.
- 4. UDP:** While less common, some Ingress controllers support UDP services. UDP is a connectionless protocol used for various purposes, including DNS and VoIP. Exposing UDP services may require specific controller support.
- 5. gRPC:** If your services use the gRPC protocol, you can configure Ingress resources to handle gRPC traffic. gRPC is a high-performance RPC (Remote Procedure Call) framework often used for communication between microservices.
- 6. WebSocket:** Ingress controllers can be configured to support WebSocket connections. WebSocket is a protocol that enables full-duplex communication over a single TCP connection and is used for real-time applications.
- 7. Custom Protocols:** In some cases, you may need to expose services using custom or proprietary protocols. Depending on your Ingress controller and its capabilities, you might be able to configure it to handle these custom protocols.

Additionally, Ingress controllers often evolve, so it's essential to refer to the documentation and features of the specific controller you plan to use to ensure compatibility with your service protocols.

## WRITTEN ASSIGNMENT 2

### Q1. How to deploy lambda function on AWS?

Deploying a Lambda function on AWS involves several steps. Lambda is a serverless compute service that lets you run code without provisioning or managing servers. Here's a step-by-step guide on how to deploy a Lambda function:

#### Step 1: Create a Lambda Function

1. Log in to the AWS Management Console.
2. Navigate to the Lambda service by searching for it in the AWS Services section.
3. Click the "Create function" button.

#### Step 2: Configure Your Function

4. Choose "Author from scratch."
5. Fill in the basic information for your function, including the name, runtime, and execution role.
6. For "Execution role," you can create a new role from a template or use an existing role if you have one with the necessary permissions.
7. Click the "Create function" button.

#### Step 3: Upload Your Code

8. In the "Function code" section, you can upload your code either directly (ZIP file or folder) or by specifying a repository from AWS CodeCommit, Amazon S3, or other options.
9. Configure the handler if needed. The handler is the method that AWS Lambda invokes.

10. Set environment variables if your code requires them.
11. Adjust the runtime settings if necessary.
12. Click the "Deploy" button to upload your code.

#### **Step 4: Define the Trigger**

13. In the "Add triggers" section, you can specify event sources that will trigger your Lambda function. This can be an API Gateway, an S3 bucket, an SNS topic, etc.
14. Configure the trigger according to your needs and grant permissions as required.

#### **Step 5: Test Your Function**

15. You can test your Lambda function by creating a test event or using a sample test event provided by AWS.
16. Click the "Test" button to run your function and see the results.

#### **Step 6: Monitor and Troubleshoot (Optional)**

17. AWS Lambda provides various monitoring and logging options. You can configure CloudWatch alarms, inspect logs, and set up notifications to keep an eye on your function's performance.

#### **Step 7: Save and Deploy the Function**

18. After you've configured everything to your satisfaction, click the "Save" button to save your changes.
19. Click the "Deploy" button to make your function live and ready to respond to triggers.

## **Step 8: Invocation and Scaling**

Your Lambda function is now deployed and ready to be invoked by the trigger you configured. AWS Lambda handles the scaling of resources based on the incoming workload automatically.

## **Q2. What are the deployment options for AWS Lambda?**

AWS Lambda offers several deployment options:

- 1. Code Upload:** You can directly upload your function code as a ZIP file or a deployment package when creating or updating your Lambda function.
- 2. AWS Lambda Layers:** You can use Lambda Layers to separate your function code from its dependencies. This allows you to manage and version common libraries independently and share them across multiple functions.
- 3. AWS SAM (Serverless Application Model):** AWS SAM is a framework for building serverless applications. You can define your Lambda functions and their associated resources in a SAM template file, then deploy the entire application using AWS SAM CLI.
- 4. AWS CloudFormation:** You can use AWS CloudFormation templates to define and deploy Lambda functions along with other AWS resources. This enables infrastructure as code (IaC) for your serverless applications.

**5. AWS Serverless Application Repository:** You can publish and share serverless applications and Lambda functions using the AWS Serverless Application Repository. Users can deploy your applications directly from the repository.

**6. AWS CodePipeline:** You can integrate AWS Lambda deployment into a continuous integration/continuous deployment (CI/CD) pipeline using AWS CodePipeline. This automates the building, testing, and deployment of your Lambda functions.

**7. Container Image Support:** AWS Lambda now supports deploying functions as container images, allowing you to package your function and dependencies in a Docker container and deploy them as a Lambda function.

These deployment options provide flexibility and enable you to choose the one that best fits your application architecture and development workflow.

### **Q3 What are the 3 full deployment modes that can be used for AWS?**

In the context of AWS Lambda, there are three primary deployment modes:

**1. Automatic Deployment:** AWS Lambda provides automatic deployment when you directly upload your function code as a ZIP

file or specify a deployment package. This is the most common and straightforward deployment method, and it's suitable for most use cases.

## **2. Infrastructure as Code (IaC) with CloudFormation: AWS**

CloudFormation is a service that allows you to define and provision AWS infrastructure and resources in a template. You can use CloudFormation to define your Lambda functions, their associated resources, and any other AWS resources your application needs. When you create or update the CloudFormation stack, it deploys the Lambda functions and other resources defined in the template. This approach is more suitable for complex serverless applications and infrastructure orchestration.

## **3. Serverless Application Model (AWS SAM): AWS SAM is an**

opensource framework for building serverless applications. It extends AWS CloudFormation to provide a simplified way to define serverless resources, including Lambda functions. You can use AWS SAM to define your functions and related resources in a SAM template, and then deploy the entire application using AWS SAM CLI. This approach is a balance between the simplicity of direct deployment and the power of CloudFormation for more complex applications.

These deployment modes offer different levels of control and abstraction, allowing you to choose the one that best matches your deployment needs and development workflow.

## **Q4. What are the 3 components of AWS Lambda?**

AWS Lambda consists of three primary components:

**1. Function Code:** This is the core component of AWS Lambda. It's the code that you want to run in response to events. You can write your code in various supported programming languages (e.g., Node.js, Python, Java, C#, etc.). You can package your code along with its dependencies into a deployment package, which you upload to Lambda.

**2. Event Sources:** Event sources are triggers that invoke your Lambda function. AWS Lambda supports various event sources, such as Amazon S3, Amazon DynamoDB, Amazon SNS, AWS API Gateway, and more. When an event occurs in one of these services, it triggers the execution of your Lambda function.

**3. Execution Environment:** AWS Lambda manages the execution environment for your function. It automatically scales and provisions the necessary infrastructure to run your code. You don't need to worry about server provisioning or resource management. AWS Lambda also provides runtime support for various programming languages, so your code runs in an isolated environment with the necessary resources to execute.