

Experiment No. 1A

Static Hosting

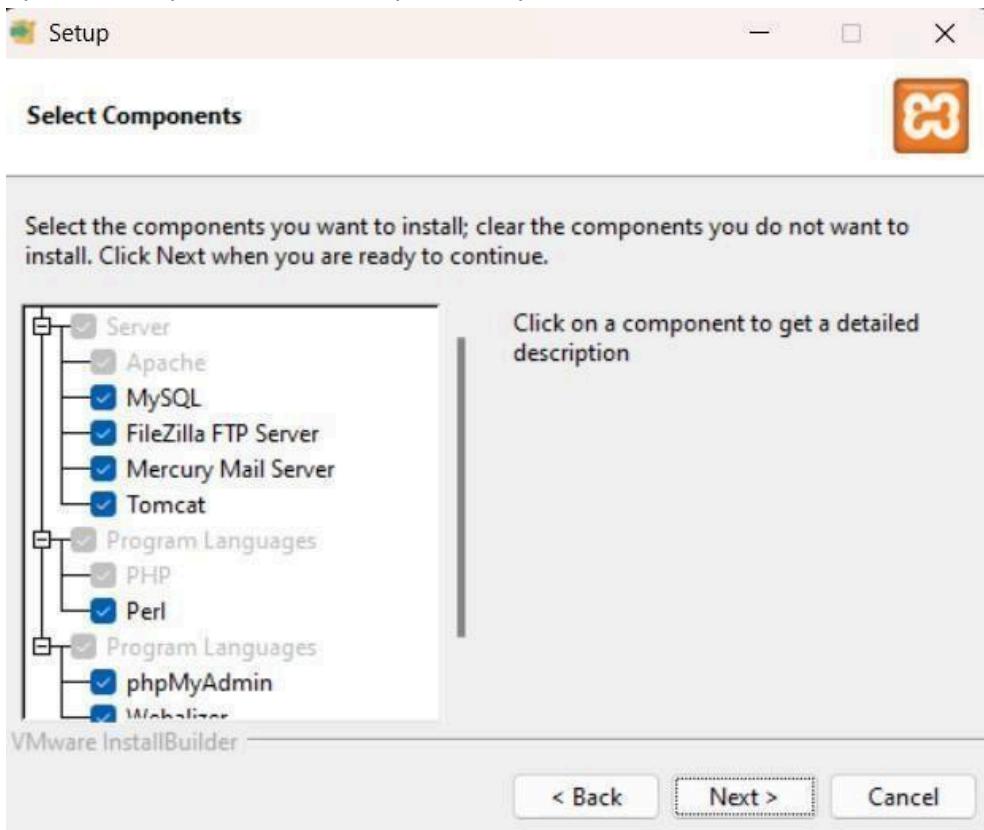
On local server (XAMPP)

Step 1: Install XAMPP from <https://www.apachefriends.org/>

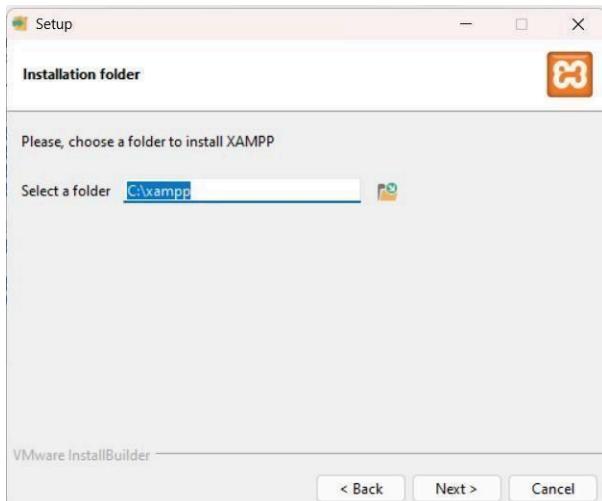
- 1) Pick your OS, and the download will start automatically.



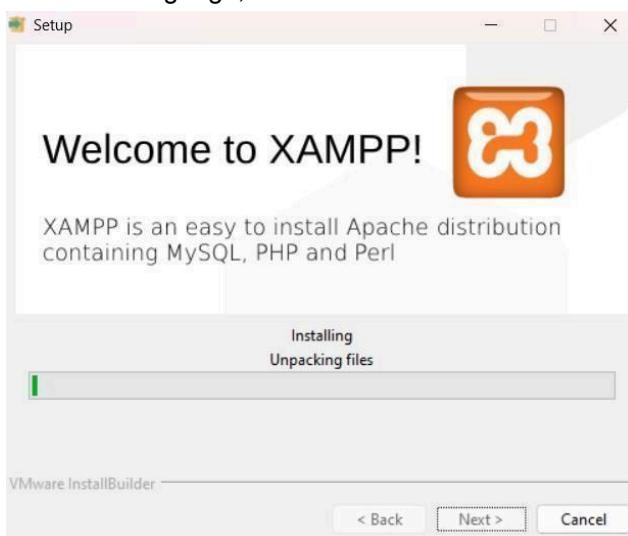
- 2) Open the setup file, select the required components, and click 'Next'.



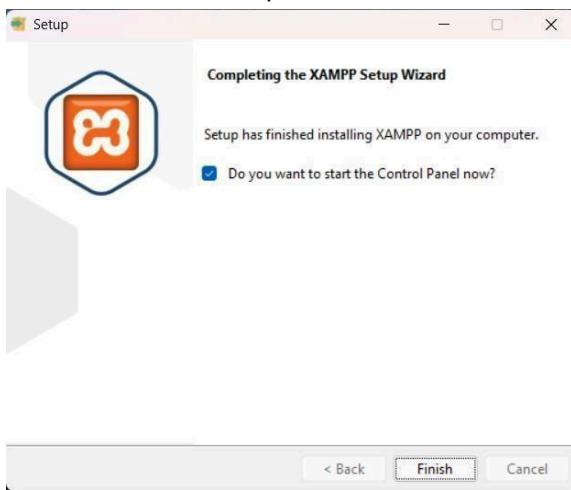
- 3) Choose an empty folder for XAMPP installation and click 'Next'.



- 4) Select the language, click next. XAMPP starts to install



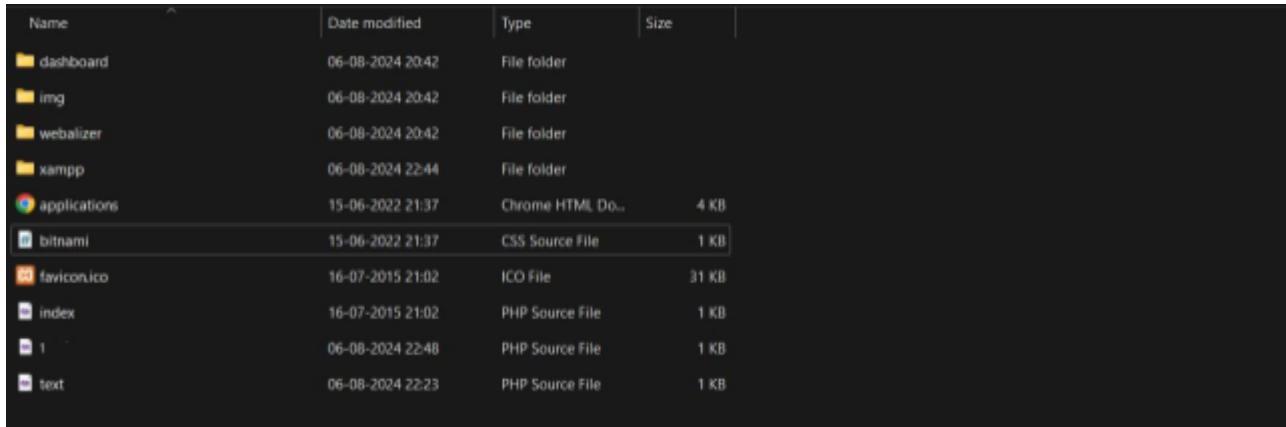
- 5) The installation is complete; click 'Finish'.



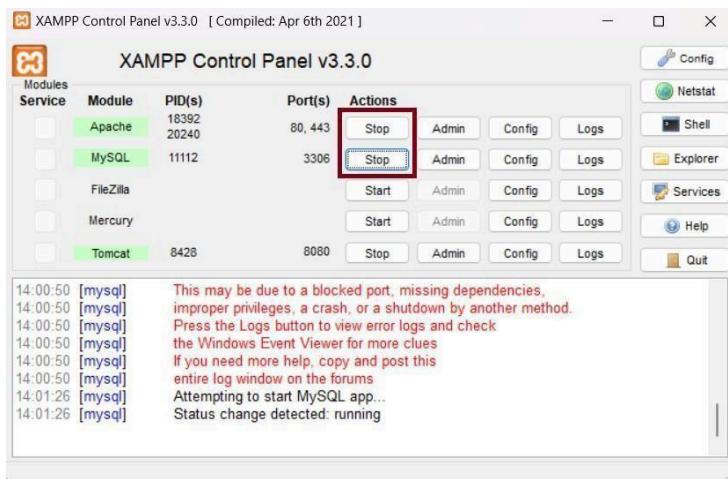
Step 2: Create a file to be hosted on the server, ensuring it has a .php extension.



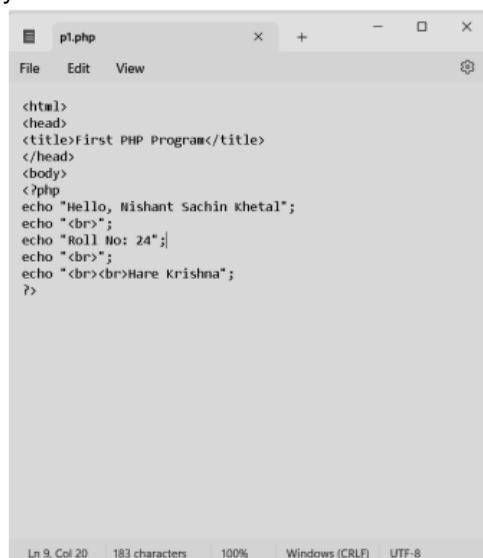
Step 3: Navigate to the directory where XAMPP is installed and open the `htdocs` folder. Place your project folder within this directory.



Step 4: Open the XAMPP Control Panel and start the Apache service (required) and the MySQL service (if needed)



Step 5: Open your web browser. Type `localhost/YOUR_FILENAME.php`. This will open your website on your browser.



AWS EC2 Instances

The screenshot shows the AWS Management Console Services menu. The menu is organized into several categories:

- Lambda**
- Batch**
- Elastic Beanstalk**
- Serverless Application Repository**
- AWS Outposts**
- EC2 Image Builder**
- AWS App Runner**
- AWS SimSpace Weaver**
- Containers**
 - Elastic Container Service
 - Elastic Kubernetes Service
 - Red Hat OpenShift Service on AWS
 - Elastic Container Registry
- Storage**
 - S3
 - EFS
 - FSx
 - S3 Glacier
 - Storage Gateway
 - AWS Backup
 - AWS Elastic Disaster Recovery
- Database**
- AWS Organizations**
- CloudWatch**
- AWS Auto Scaling**
- CloudFormation**
- AWS Config**
- OpsWorks**
- Service Catalog**
- Systems Manager**
- Trusted Advisor**
- Control Tower**
- AWS Well-Architected Tool**
- AWS Chatbot**
- Launch Wizard**
- AWS Compute Optimizer**
- Resource Groups & Tag Editor**
- Amazon Grafana**
- Amazon Prometheus**
- AWS Resilience Hub**
- Incident Manager**
- AWS License Manager**
- Service Quotas**
- AWS Proton**
- CloudTrail**
- AWS Resource Explorer**
- AWS User Notifications**
- Secrets Manager**
- GuardDuty**
- Amazon Inspector**
- Amazon Macie**
- IAM Identity Center**
- Certificate Manager**
- Key Management Service**
- CloudHSM**
- Directory Service**
- WAF & Shield**
- AWS Firewall Manager**
- AWS Artifact**
- Detective**
- AWS Signer**
- AWS Private Certificate Authority**
- Security Hub**
- AWS Audit Manager**
- Security Lake**
- Amazon Verified Permissions**
- AWS Payment Cryptography**
- IAM**
- Cloud Financial Management**
 - AWS Marketplace
 - AWS Billing Conductor
 - Billing and Cost Management

Step 1: Log in to your AWS account. Go to services and open EC2.

The screenshot shows the AWS EC2 Global View. The top navigation bar includes "EC2 Global View" and other icons. Below the navigation bar, a message states: "You are using the following Amazon EC2 resources in the US East (N. Virginia) Region:". A table displays the following resource counts:

Instances (running)	2	Auto Scaling Groups	0
Capacity Reservations	0	Dedicated Hosts	0
Elastic IPs	0	Instances	2
Key pairs	1	Load balancers	0
Placement groups	0	Security groups	3
Snapshots	0	Volumes	2

Step 2: Click on Launch an Instance:

The screenshot shows the 'Name and tags' section of the 'Launch an instance' wizard. It includes a 'Name' input field containing 'e.g. My Web Server' and a 'Add additional tags' link.

Step 3: Assign a name to your EC2 instance, keeping other options default. Scroll down and click on 'Launch Instance'.

The screenshot shows the 'Name and tags' section with the 'Name' input field now containing 'nishant'. The 'Add additional tags' link is also visible.

Step 4: Click on the name of your EC2 instance and go to the 'Details' section

The screenshot shows the 'Application and OS Images (Amazon Machine Image)' section. It features a search bar with placeholder text 'Search our full catalog including 1000s of application and OS images', a 'Recent' tab, a 'Quick Start' tab, and a grid of OS icons including Amazon Linux, macOS, Ubuntu, Windows, Red Hat, and SUSE. A search icon and a 'Browse more AMIs' link are also present.

Name	Instance ID	State	Type	Status Check	Alarm Status	Zone	Public IP
Nishant Khetal	i-0341f2e761db28aec	Running	t2.micro	2/2 checks passed	View alarms +	us-east-1c	ec2-52-
Nishant	i-0689f78090ca559a9	Running	t2.micro	2/2 checks passed	View alarms +	us-east-1c	ec2-34-

Step 5: Create a Connection.

Establishing Connection ...

```
Welcome to Ubuntu 24.04 LTS (GNU/Linux 6.8.0-1009-aws x86_64)

 * Documentation: https://help.ubuntu.com
 * Management: https://landscape.canonical.com
 * Support: https://ubuntu.com/pro

System information as of Sun Aug 4 05:22:27 UTC 2024

System load: 0.08      Processes: 103
Usage of /: 22.9% of 6.71GB  Users logged in: 0
Memory usage: 19%          IPv4 address for enx0: 172.31.32.106
Swap usage: 0%          

Expanded Security Maintenance for Applications is not enabled.

0 updates can be applied immediately.

Enable ESM Apps to receive additional future security updates.
See https://ubuntu.com/esm or run: sudo pro status

The list of available updates is more than a week old.
To check for new updates run: sudo apt update

The programs included with the Ubuntu system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*copyright.
```

Step 6: Run the commands:

```
root@ip-172-31-32-106:~/temp# wget https://www.free-css.com/assets/files/free-css-templates/download/page296/neogym.zip
--2024-08-04 05:39:23-- https://www.free-css.com/assets/files/free-css-templates/download/page296/neogym.zip
Resolving www.free-css.com (www.free-css.com)... 217.160.0.242, 2001:8d0:100f:f000::20f
Connecting to www.free-css.com (www.free-css.com)|217.160.0.242|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 724954 [application/zip]
Saving to: 'neogym.zip'

neogym.zip          0%[=====]   1      0  --.-KB/s  ne
ogym.zip          9%[=====]  1  63.73K  294KB/s  neog
ym.zip          33%[=====]  1 239.73K  563KB/s  neogym
.zip          100%[=====]  1 707.96K  1.17MB/s  in 0.6s

2024-08-04 05:39:24 (1.17 MB/s) - 'neogym.zip' saved [724954/724954]

root@ip-172-31-32-106:~/temp#
```

Step 7: Upload the zip file of website:

```
root@ip-172-31-32-106:~/temp# wget https://www.free-css.com/assets/files/free-css-templates/download/page296/neogym.zip
--2024-08-04 05:39:23-- https://www.free-css.com/assets/files/free-css-templates/download/page296/neogym.zip
Resolving www.free-css.com (www.free-css.com)... 217.160.0.242, 2001:8d0:100f:f000::20f
Connecting to www.free-css.com (www.free-css.com)|217.160.0.242|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 724954 [application/zip]
Saving to: 'neogym.zip'

neogym.zip          0%[=====]   1      0  --.-KB/s  ne
ogym.zip          9%[=====]  1  63.73K  294KB/s  neog
ym.zip          33%[=====]  1 239.73K  563KB/s  neogym
.zip          100%[=====]  1 707.96K  1.17MB/s  in 0.6s

2024-08-04 05:39:24 (1.17 MB/s) - 'neogym.zip' saved [724954/724954]

root@ip-172-31-32-106:~/temp#
```

```
root@ip-172-31-32-106:~/temp# cd neogym-html;
root@ip-172-31-32-106:~/temp/neogym-html# ls -lrt
total 56
-rw-r--r-- 1 root root 6738 Sep 15 2020 why.html
-rw-r--r-- 1 root root 6842 Sep 15 2020 trainer.html
-rw-r--r-- 1 root root 17875 Sep 15 2020 index.html
-rw-r--r-- 1 root root 5649 Sep 15 2020 contact.html
drwxr-xr-x 2 root root 4096 Sep 15 2020 js
drwxr-xr-x 2 root root 4096 Sep 15 2020 images
drwxr-xr-x 2 root root 4096 Sep 15 2020 css
root@ip-172-31-32-106:~/temp/neogym-html# mv * /var/www/html/
root@ip-172-31-32-106:~/temp/neogym-html# cd /var/www/html/
root@ip-172-31-32-106:/var/www/html# ls -lrt
total 56
-rw-r--r-- 1 root root 6738 Sep 15 2020 why.html
-rw-r--r-- 1 root root 6842 Sep 15 2020 trainer.html
-rw-r--r-- 1 root root 17875 Sep 15 2020 index.html
-rw-r--r-- 1 root root 5649 Sep 15 2020 contact.html
drwxr-xr-x 2 root root 4096 Sep 15 2020 js
drwxr-xr-x 2 root root 4096 Sep 15 2020 images
drwxr-xr-x 2 root root 4096 Sep 15 2020 css
root@ip-172-31-32-106:/var/www/html#
```

Step 8 : Provide all the required Inbound rules:

The screenshot shows the AWS CloudFormation Inbound Rules configuration page. It lists three security group rules:

- Rule 1: Type: SSH, Protocol: TCP, Port range: 22, Source: 0.0.0.0/0, Description: firsthosting
- Rule 2: Type: HTTP, Protocol: TCP, Port range: 80, Source: 0.0.0.0/0, Description: firsthosting
- Rule 3: Type: HTTPS, Protocol: TCP, Port range: 443, Source: 0.0.0.0/0, Description: firsthosting

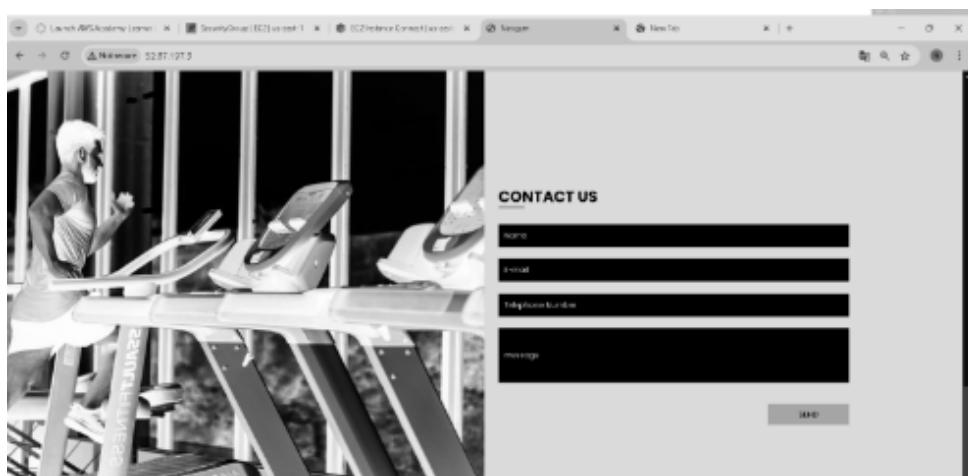
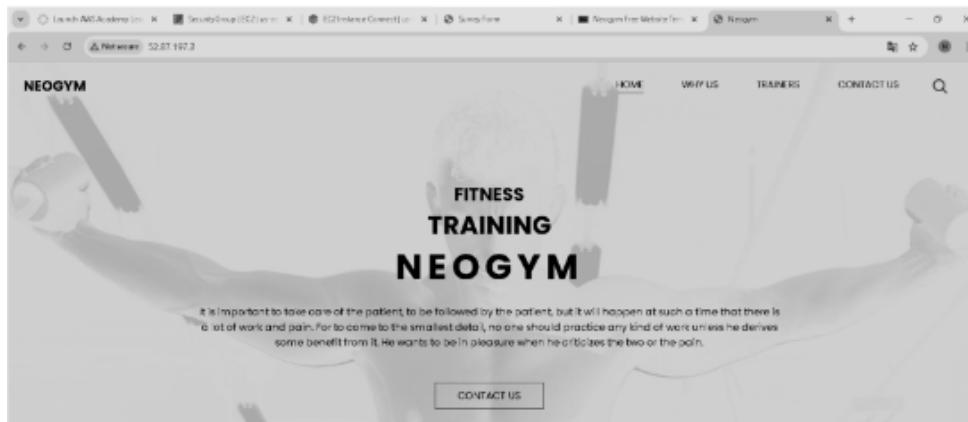
An "Add rule" button is visible at the bottom left.

Step 9: Start the Apache Server:

```
root@ip-172-31-32-106:/var/www/html# sudo systemctl status apache2
● apache2.service - The Apache HTTP Server
  Loaded: loaded (/usr/lib/systemd/system/apache2.service; enabled; preset: enabled)
  Active: active (running) since Sun 2024-08-04 05:30:27 UTC; 23min ago
    Docs: https://httpd.apache.org/docs/2.4/
 Main PID: 14033 (apache2)
   Tasks: 55 (limit: 1130)
     Memory: 7.2M (peak: 7.4M)
       CPU: 137ms
      CGroup: /system.slice/apache2.service
          ├─14033 /usr/sbin/apache2 -k start
          ├─14036 /usr/sbin/apache2 -k start
          └─14037 /usr/sbin/apache2 -k start
```

```
Aug 04 05:30:27 ip-172-31-32-106 systemd[1]: Starting apache2.service - The Apache HTTP Server...
Aug 04 05:30:27 ip-172-31-32-106 systemd[1]: Started apache2.service - The Apache HTTP Server.
```

Step 10 : Now reload the website. You can see your website with public IP:



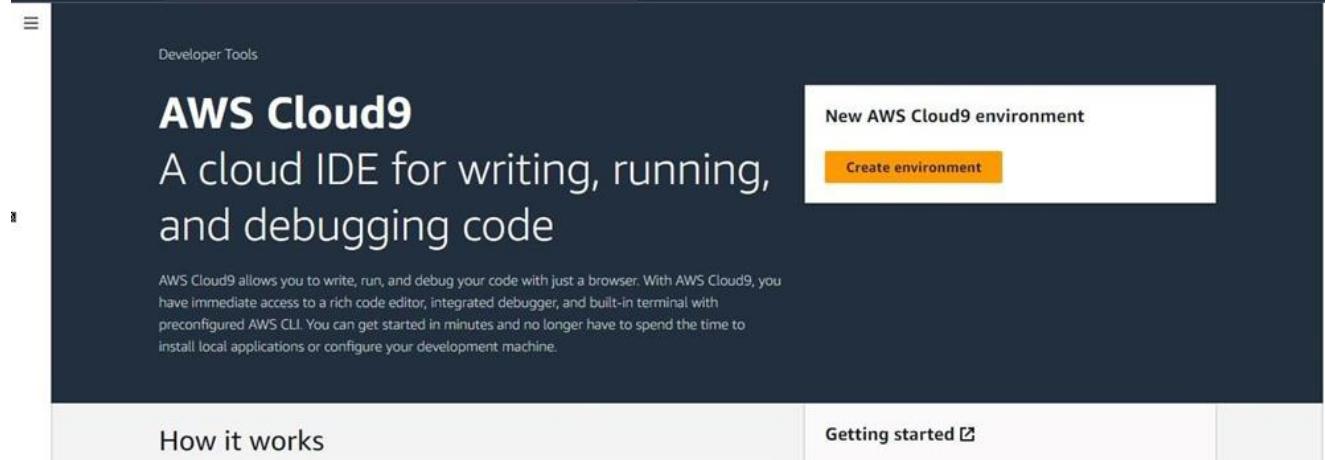
Name: Nishant S Khetal

D15C

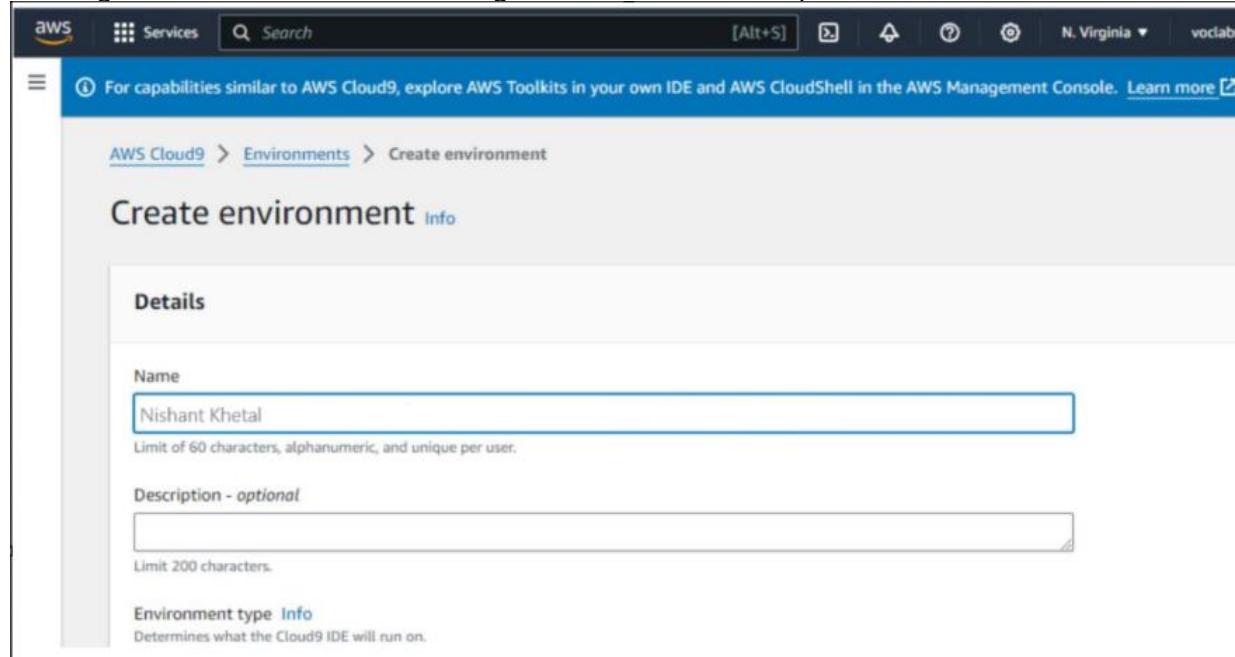
Roll No:24

Experiment No. 1B

Log in to your AWS account and search for Cloud9. Select the option to create a new environment.



Provide the name and other necessary configurations to create the environment. When configuring network settings, attempting to use the AWS Systems Manager may result in an error. The error message indicates a failure in creating the IAM resources required for SSM.



Use the Secure Shell option in Network settings

Network settings [Info](#)

Connection
How your environment is accessed.

AWS Systems Manager (SSM)
Accesses environment via SSM without opening inbound ports (no ingress).

Secure Shell (SSH)
Accesses environment directly via SSH, opens inbound ports.

▶ [VPC settings](#) [Info](#)

▶ [Tags - optional](#) [Info](#)
A tag is a label that you assign to an AWS resource. Each tag consists of a key and an optional value. You can use tags to search and filter your resources or track your AWS costs.

The following IAM resources will be created in your account

After completing the configuration, click on 'Create Environment' to set up the Cloud9 environment.

AWS Cloud9 [X](#)

Successfully created Shiven Bansal. To get the most out of your environment, see [Best practices for using AWS Cloud9](#) [Learn more](#) [X](#)

For capabilities similar to AWS Cloud9, explore AWS Toolkits in your own IDE and AWS CloudShell in the AWS Management Console. Learn more [X](#)

AWS Cloud9 > Environments

Environments (1)

Delete View details Open in Cloud9 [Create environment](#)

My environments

Name	Cloud9 IDE	Environment type	Connection	Permission	Owner ARN
Nishant_Khetal	Open	EC2 instance	Secure Shell (SSH)	Owner	arn:aws:sts::583784900342:assumed-role/voclabs/user3397511...

Click on the environment name to open the created Cloud9 Environment.

File Edit Find View Go Run Tools Window support PREVIEW HELP Share Settings

Go to Anything (Ctrl-P)

NishantCloud9 c9 README.md

Welcome Developer Tools

AWS Cloud9 Welcome to your development environment

AWS Cloud9 allows you to write, run, and debug your code with just a browser. You can run the IDE, write code for AWS Lambda and Amazon API Gateway, share your IDE with others in real time, and much more.

Getting started

Create File Upload Files...

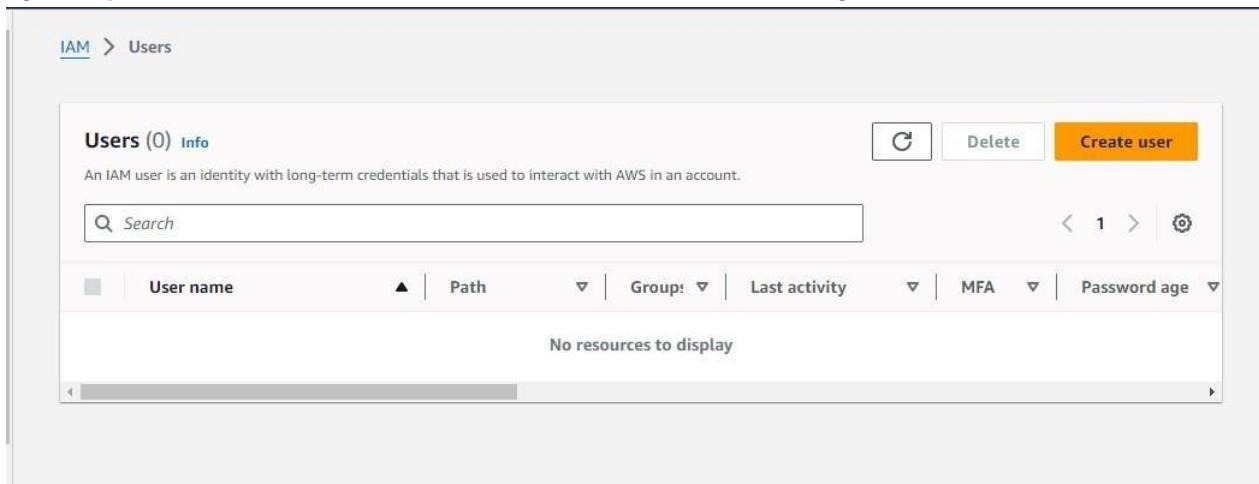
Toolkit for AWS Cloud9

This AWS Toolkit for Cloud9 is your IDE, and enables built-in identifier completion for AWS Lambda and Amazon API Gateway, share your IDE with others in real time, and much more.

bash -lp-172-31-33-21.ecx Immediate

voclabs~/environment \$

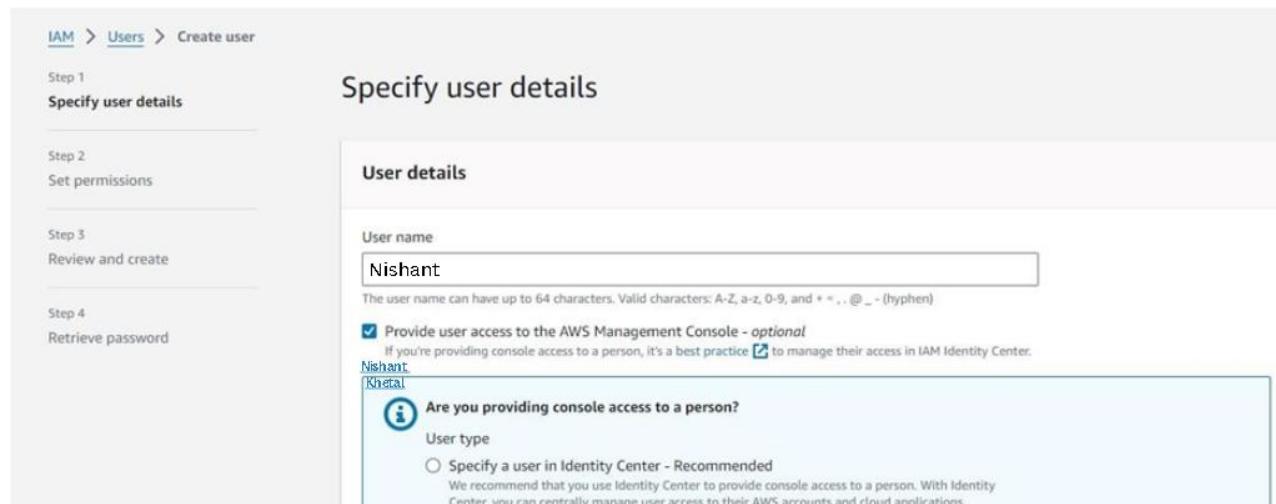
Log in to your AWS account and search for the IAM service. Navigate to the 'Users' tab and click on



The screenshot shows the AWS IAM 'Users' page. At the top, there is a breadcrumb navigation: IAM > Users. Below the header, it says 'Users (0) Info'. A note states: 'An IAM user is an identity with long-term credentials that is used to interact with AWS in an account.' There is a 'Search' input field and a pagination control showing page 1 of 1. The main table has columns: User name, Path, Group, Last activity, MFA, and Password age. A message at the bottom of the table area says 'No resources to display'.

'Create User' to add a new user.

Give the user name, select the “provide user access” checkbox. Also select the option “I want to create an IAM user”. Otherwise we will have to enable the Identity center and specify a user there.



The screenshot shows the 'Specify user details' step of the IAM User creation wizard. On the left, a sidebar lists steps: Step 1 (Specify user details), Step 2 (Set permissions), Step 3 (Review and create), and Step 4 (Retrieve password). The main panel is titled 'Specify user details' and contains a 'User details' section. It shows a 'User name' input field with 'Nishant' typed in. Below it is a note: 'The user name can have up to 64 characters. Valid characters: A-Z, a-z, 0-9, and + = , . @ _ - (hyphen)'. A checked checkbox says 'Provide user access to the AWS Management Console - optional'. A tooltip explains: 'If you're providing console access to a person, it's a best practice [to manage their access in IAM Identity Center](#)'. Under 'User type', there are two options: 'Specify a user in Identity Center - Recommended' (radio button selected) and 'Specify a user in IAM - Not recommended' (radio button unselected). A note below the first option says: 'We recommend that you use Identity Center to provide console access to a person. With Identity Center, you can centrally manage user access to their AWS accounts and cloud applications.'

- Provide user access to the AWS Management Console - *optional*
If you're providing console access to a person, it's a best practice [to manage their access in IAM Identity Center](#).



Are you providing console access to a person?

User type

Specify a user in Identity Center - Recommended

We recommend that you use Identity Center to provide console access to a person. With Identity Center, you can centrally manage user access to their AWS accounts and cloud applications.

I want to create an IAM user

We recommend that you create IAM users only if you need to enable programmatic access through access keys, service-specific credentials for AWS CodeCommit or Amazon Keypairs, or a backup credential for emergency account access.

Console password

Autogenerated password

You can view the password after you create the user.

Custom password

Enter a custom password for the user.

- Must be at least 8 characters long



IAM > Users > Create user

Step 1

[Specify user details](#)

Step 2

Set permissions

Step 3

[Review and create](#)

Step 4

[Retrieve password](#)

Set permissions

Add user to an existing group or create a new one. Using groups is a best-practice way to manage user's permissions by job functions. [Learn more](#)

Permissions options

Add user to group

Add user to an existing group, or create a new group. We recommend using groups to manage user permissions by job function.

Copy permissions

Copy all group memberships, attached managed policies, and inline policies from an existing user.

Attach policies directly

Attach a managed policy directly to a user. As a best practice, we recommend attaching policies to a group instead. Then, add the user to the appropriate group.



Get started with groups

Create a group and select policies to attach to the group. We recommend using groups to manage user

[Create group](#)

Next, click on 'Add User to Group.' If no existing group is available, choose 'Create Group.' Assign a name to the group, select any necessary policies, and then create the group.

Create user group

Create a user group and select policies to attach to the group. We recommend using groups to manage user permissions by job function, AWS service access, or custom permissions. [Learn more](#)

User group name
Enter a meaningful name to identify this group.

Maximum 128 characters. Use alphanumeric and '+,-,@,-,' characters.

Permissions policies (947)

Filter by Type
 All ty... ▾

<input type="checkbox"/>	Policy name	Type	Use...	Description
<input type="checkbox"/>	AdministratorAccess	AWS managed ...	None	Provides full access to AWS services.
<input type="checkbox"/>	AdministratorAcc...	AWS managed	None	Grants account administrative perm
<input type="checkbox"/>	AdministratorAcc...	AWS managed	None	Grants account administrative perm

Create user group

Once the group is created, select the group in which the user should be added.

Review and create

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

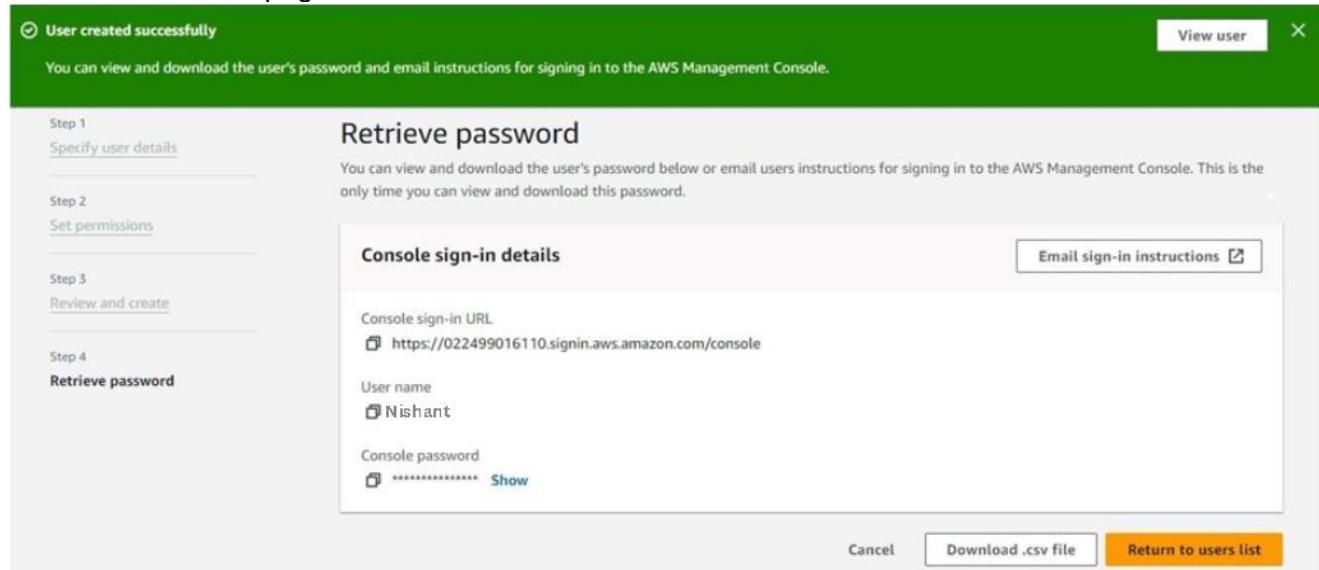
User details

User name Nishant	Console password type Custom password	Require password reset No
----------------------	--	------------------------------

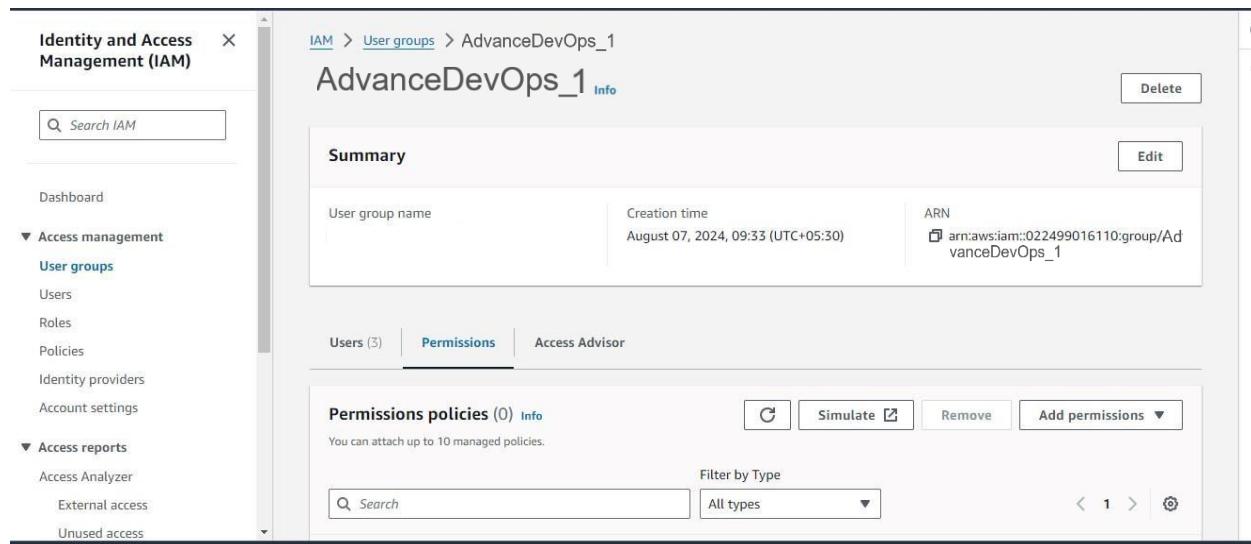
Permissions summary

Name	Type	Used as
AdvanceDevOps_1	Group	Permissions group
AdvanceDevOps_2	Group	Permissions group
AdvDevOpsLab_3	Group	Permissions group

Review all the configurations and user details, then click on 'Create User.' Afterward, you will be directed to the next page.



After creation of the user, go to the user groups tab and select the group in which the user has been added. Navigate to the permissions tab. Click on add permissions and attach policy.



Search for the 'AWSCloud9EnvironmentMember' policy and attach it.

IAM > User groups > AdvanceDevOps_3_21_9 > Add permissions

Attach permission policies to AdvanceDevOps_1

▶ Current permissions policies (0)

Other permission policies (945) Nishant

You can attach up to 10 managed policies to this user group. All of the users in this group inherit the attached permissions.

Policy name		Type	Used as	Description
<input type="checkbox"/>	AdministratorAccess	AWS managed - job function	None	Provides full access to AWS services an...
<input type="checkbox"/>	AdministratorAccess-Amp...	AWS managed	None	Grants account administrative permis...
<input type="checkbox"/>	AdministratorAccess-AWS...	AWS managed	None	Grants account administrative permis...
<input type="checkbox"/>	AlexaForBusinessDeviceS...	AWS managed	None	Provide device setup access to AlexaFo...

▶ Current permissions policies (0)

Other permission policies (1/945)

You can attach up to 10 managed policies to this user group. All of the users in this group inherit the attached permissions.

Policy name		Type	Used as	Description
<input type="checkbox"/>	AWSCloud9Administrator	AWS managed	None	Provides administrator access to AWS ...
<input checked="" type="checkbox"/>	AWSCloud9Environment...	AWS managed	None	Provides the ability to be invited into ...
<input type="checkbox"/>	AWSCloud9SSMInstanceP...	AWS managed	None	This policy will be used to attach a rol...
<input type="checkbox"/>	AWSCloud9User	AWS managed	None	Provides permission to create AWS Clo...

[Cancel](#) [Attach policies](#)

Identity and Access Management (IAM)

Search IAM

Dashboard

Access management

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

Access reports

- Access Analyzer
- External access
- Unused access

Policies attached to this user group.

Summary

User group name	Creation time	ARN
AdvanceDevOps_3_21_9	August 07, 2024, 09:33 (UTC+05:30)	arn:aws:iam::022499016110:group/AdvanceDevOps_3_21_9

Users (3) Permissions Access Advisor

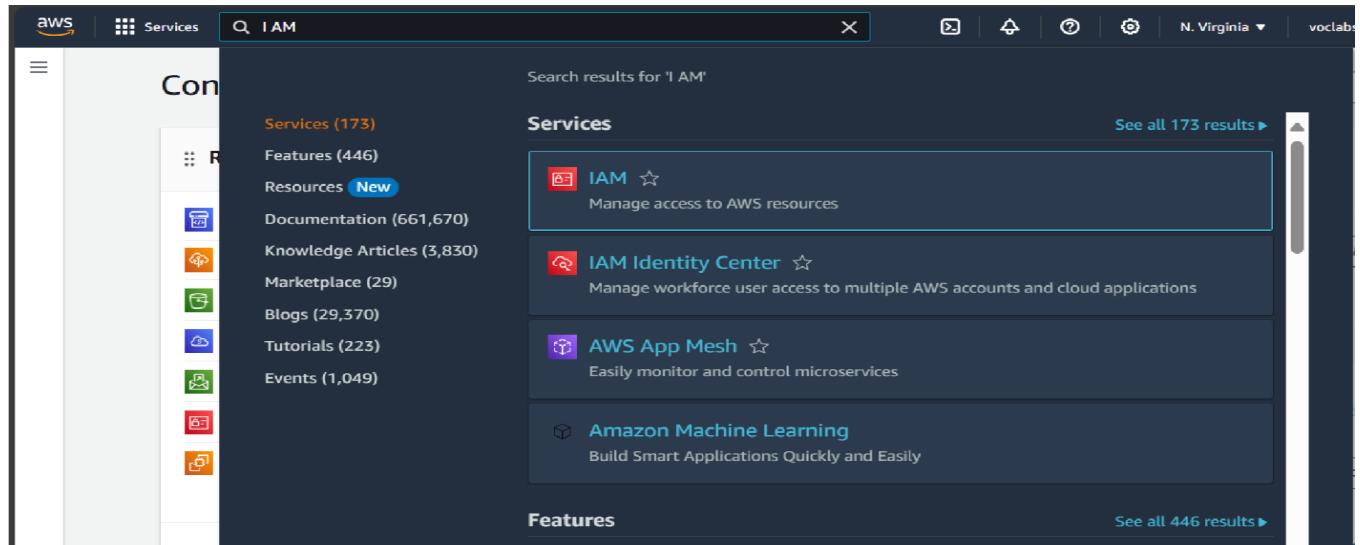
Permissions policies (1) Info

You can attach up to 10 managed policies.

Policy name		Type	Attached entities
<input type="checkbox"/>	AWSCloud9EnvironmentMe...	AWS managed	3

Experiment No: 2**Step1 :- Creation of role:**

1. Login to your AWS account and search for IAM



2. Then go into the role section and click on create role.

The screenshot shows the 'Roles' page under the 'Identity and Access Management (IAM)' service. It displays a list of 14 existing roles, each with a checkbox, the role name, the trusted entity (e.g., AWS Service: ec2), and the last activity time. A 'Create role' button is visible at the top right of the table.

Role name	Trusted entities	Last activity
aws-elasticbeanstalk-ec2-role	AWS Service: ec2	17 minutes ago
aws-elasticbeanstalk-service-role	AWS Service: elasticbeanstalk	20 minutes ago
AWSCodePipelineServiceRole-us-east-1-pipeline	AWS Service: codepipeline	Yesterday
AWSCodePipelineServiceRole-us-east-1-sadneya	AWS Service: codepipeline	2 days ago
AWSCodePipelineServiceRole-us-east-1-sadneya_46	AWS Service: codepipeline	Yesterday
AWSCodePipelineServiceRole-us-east-1-sadneya46	AWS Service: codepipeline	-
AWSServiceRoleForAutoScaling	AWS Service: autoscaling (Service-Linker)	23 minutes ago
AWSServiceRoleForSupport	AWS Service: support (Service-Linker)	-
AWSServiceRoleForTrustedAdvisor	AWS Service: trustedadvisor (Service-Linker)	-
codebuild-sadneya46-service-role	AWS Service: codebuild	-
codebuild-sampleweb-service-role	AWS Service: codebuild	-

3. Then select a trusted entity as AWS service.

The screenshot shows the 'Select trusted entity' step of the IAM Role creation wizard. On the left, a sidebar lists 'Step 1 Select trusted entity', 'Step 2 Add permissions', and 'Step 3 Name, review, and create'. The main area is titled 'Trusted entity type' and contains four options:

- 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**
Allows users federated with SAML 2.0 from a corporate directory to perform actions in this account.
- Custom trust policy**
Create a custom trust policy to enable others to perform actions in this account.

4. Select use case as EC2.

The screenshot shows the 'Use case' step of the IAM Role creation wizard. It displays a single option under 'Service or use case': 'EC2'. Below it, a list of use cases is shown, with 'EC2' selected:

- EC2**
Allows EC2 instances to call AWS services on your behalf.
- EC2 Role for AWS Systems Manager**
Allows EC2 instances to call AWS services like CloudWatch and Systems Manager on your behalf.
- EC2 Spot Fleet Role**
Allows EC2 Spot Fleet to request and terminate Spot Instances on your behalf.
- EC2 - Spot Fleet Auto Scaling**
Allows Auto Scaling to access and update EC2 spot fleets on your behalf.
- EC2 - Spot Fleet Tagging**
Allows EC2 to launch spot instances and attach tags to the launched instances on your behalf.
- EC2 - Spot Instances**
Allows EC2 Spot Instances to launch and manage spot instances on your behalf.
- EC2 - Spot Fleet**
Allows EC2 Spot Fleet to launch and manage spot fleet instances on your behalf.

5. Select permissions as AWS Elastic Beanstalk Web Tier and AWS elastic Beanstalk worker tier.

The screenshot shows the 'Permissions policy summary' and 'Step 3: Add tags' steps of the IAM Role creation wizard.

Permissions policy summary:

Policy name	Type	Attached as
AWS-ElasticBeanstalk-WebTier	AWS managed	Permissions policy
AWS-ElasticBeanstalk-WorkerTier	AWS managed	Permissions policy

Step 3: Add tags:

Add tags - optional Info
Tags are key-value pairs that you can add to AWS resources to help identify, organize, or search for resources.

No tags associated with the resource.

Add new tag
You can add up to 50 more tags.

Cancel Previous Create role

6. Give a name to Role.

The screenshot shows the 'Create role' wizard in the AWS IAM console. The current step is 'Step 1: Select trusted entities'. The 'Role details' section contains the following information:

- Role name:** elasticbeanstalk-nishant
- Description:** Allows EC2 instances to call AWS services on your behalf.

7. Then the role gets created

The screenshot shows the 'Roles' page in the AWS IAM console. A new role named 'elasticbeanstalk-nishant' has been created. The 'Summary' section displays the following details:

Creation date	ARN	Instance profile ARN
August 09, 2024, 09:33 (UTC+05:30)	arn:aws:iam::851725480355:role/aws-elasticbeanstalk-nishant	arn:aws:iam::851725480355:instance-profile/aws-elasticbeanstalk-nishant
Last activity	Maximum session duration	
-	1 hour	

The 'Permissions' tab is selected, showing 'Permissions policies (2)'. There are buttons for 'Edit', 'Delete', 'Simulate', 'Remove', and 'Add permissions'.

Step 2 :- Creation Elastic Beanstalk Environment

1. search for Elastic Beanstalk in the search box.

The screenshot shows the AWS search interface with the query 'elastiCache'. The results are filtered under the 'Services' category. The first result is 'ElastiCache' with a subtitle 'In-Memory Cache'. Below it is 'Elastic Transcoder' with a subtitle 'Easy-to-Use Scalable Media Transcoding'. The third result is 'Elastic Beanstalk' with a subtitle 'Run and Manage Web Apps'. The fourth result is 'Elastic Container Service' with a subtitle 'Highly secure, reliable, and scalable way to run containers'. There are also sections for 'Features' and 'Documentation'.

2. Open up Elastic Beanstalk and name your web app.

The screenshot shows the 'Create New Environment' wizard. In the 'Application information' section, the 'Application name' is set to 'nishant'. In the 'Environment information' section, the 'Environment name' is set to 'nishant-env'. Both sections include descriptive text and validation messages.

Application information Info

Application name

Maximum length of 100 characters.

► Application tags (optional)

Environment information Info

Choose the name, subdomain and description for your environment. These cannot be changed later.

Environment name

Must be from 4 to 40 characters in length. The name can contain only letters, numbers, and hyphens. It can't start or end with a hyphen. This name must be unique within a region in your account.

3. Select platform as PHP.

Platform Info

Platform type

Managed platform
Platforms published and maintained by Amazon Elastic Beanstalk. [Learn more](#)

Custom platform
Platforms created and owned by you. This option is unavailable if you have no platforms.

Platform

PHP

Platform branch

PHP 8.3 running on 64bit Amazon Linux 2023

Platform version

4.3.1 (Recommended)

4. After clicking on next you need to select the use existing role. Then you will see the existing role select it like here it is aws-elasticbeanstalk-service-role. Which we created in 1st part. Select role, then select key you have created then profile will be automatically selected according to role. then click on create application by keeping all the remaining settings as it is.

Step 3 - optional

[Set up networking, database, and tags](#)

Step 4 - optional

[Configure instance traffic and scaling](#)

Step 5 - optional

[Configure updates, monitoring, and logging](#)

Step 6

[Review](#)

Service role

Create and use new service role
 Use an existing service role
Existing service roles
Choose an existing IAM role for Elastic Beanstalk to assume as a service role. The existing IAM role must have the required IAM managed policies.

aws-elasticbeanstalk-service-role

EC2 key pair

Select an EC2 key pair to securely log in to your EC2 instances. [Learn more](#)

key-linux

EC2 instance profile

Choose an IAM instance profile with managed policies that allow your EC2 instances to perform required operations.

aws-elasticbeanstalk-nishant

[View permission details](#)

[Cancel](#) [Skip to review](#) [Previous](#) **Next**

Keep Set up networking, database and tags, Configure instance traffic and scaling, Configure updates, monitoring and logging all these default.

5. Beanstalk creates a sample environment for you to deploy your application. By default, it creates an EC2 instance, a security group, an Auto Scaling group, an Amazon S3 Bucket, Amazon CloudWatch alarms and a domain name for your Application.

The screenshot shows the AWS Elastic Beanstalk console interface. On the left, a sidebar lists applications and environments. Under 'Environment: Sadneya123-env', there are links for 'Go to environment' and other configuration options. The main area displays the 'nishant-env' environment details. A green banner at the top indicates 'Environment successfully launched.' The 'Environment overview' section shows 'Health' as 'Ok', 'Domain' as 'abhinav215-env.eba-6w7emmur.us-east-1.elasticbeanstalk.com', 'Environment ID' as 'e-vw23gecggs', and 'Application name' as 'Abhinav'. To the right, the 'Platform' section shows 'Node.js 20 running on 64bit Amazon Linux 2023/6.1.8', 'Running version' as '—', and 'Platform state' as 'Supported'. At the bottom, navigation tabs include Events, Health, Logs, Monitoring, Alarms, Managed updates, and Tags.

Step 3: Get a copy of your sample code

In this step, we will get the sample code from [this](#) GitHub Repository to later host it. The pipeline takes code from the source and then performs actions on it.

For this experiment, as a source, we will use this forked GitHub repository. We can alternatively also use Amazon S3 and AWS CodeCommit.

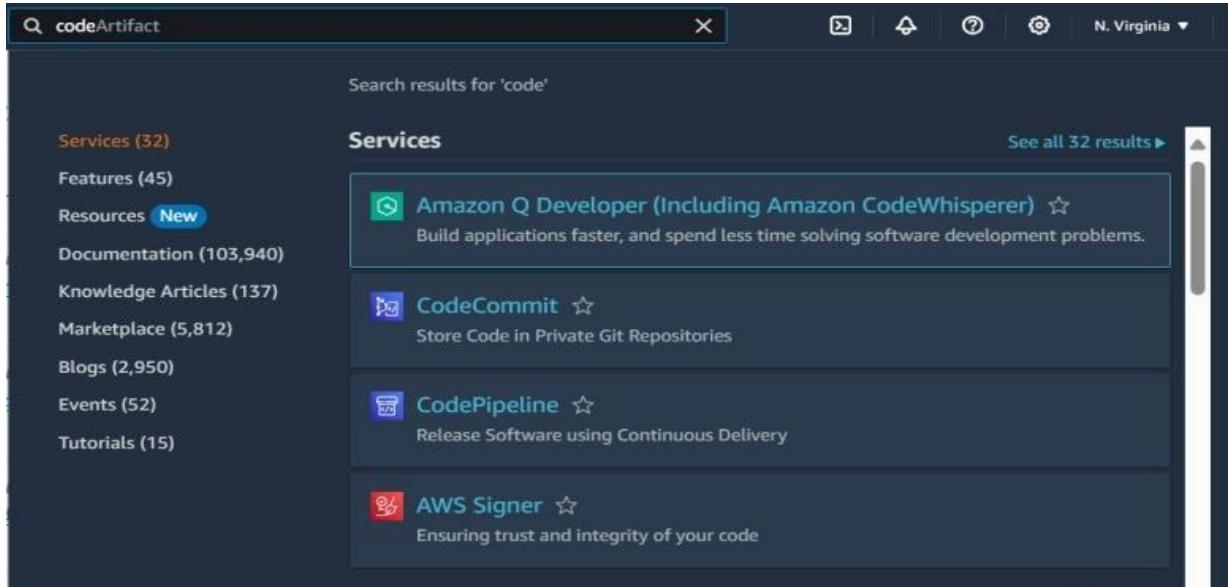
Go to the repository shared above and simply fork it.

The screenshot shows a GitHub forked repository page for 'aws-codepipeline-s3-codedeploy-linux-2.0'. At the top, there's a header with the repository name, a 'Public' link, and standard GitHub navigation buttons like Watch, Fork, and Star. Below the header, the main repository page is displayed. It includes tabs for Code, Pull requests, Actions, Projects, Wiki, Security, Insights, and Settings. The 'Code' tab is selected. The repository was forked from 'mosharm/aws-codepipeline-s3-codedeploy-linux-2.0'. The 'About' section contains a brief description: 'Use this sample when creating a simple pipeline in AWS CodePipeline while following the Simple Pipeline Walkthrough tutorial.' It lists files like README.md, CONTRIBUTING.md, LICENSE, and app-specification.yml, along with their commit history. The 'Releases' and 'Packages' sections are currently empty. The 'Languages' section shows no data.

Step 4: Creating a CodePipeline

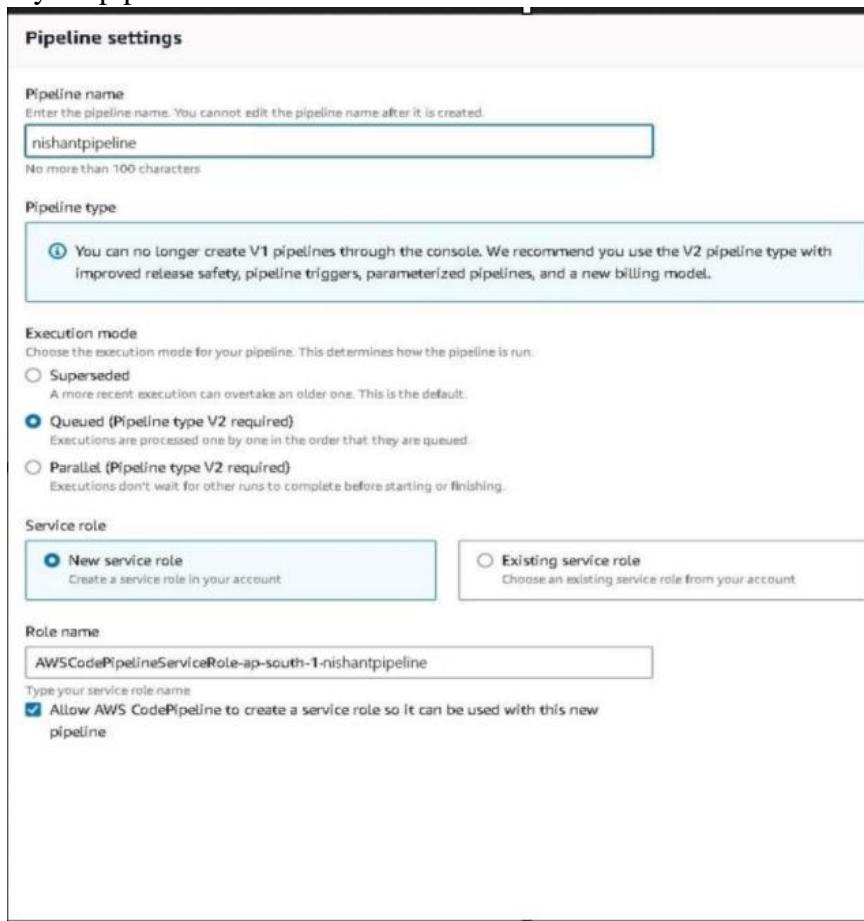
In this step, we'll create a simple pipeline that has its source and deployment information. In this case, however, we will skip the build stage where you get to plug in our preferred build provider.

1. Search CodePipeline in the search bar and click on create a new Pipeline.



The screenshot shows the AWS search interface with the query 'code' entered in the search bar. The results are filtered under the 'Services' category, which contains 32 items. The 'CodePipeline' service is listed third in the results, featuring a blue icon, the service name, a star rating, and a brief description: 'Release Software using Continuous Delivery'.

2. Give a name to your pipeline.



The screenshot displays the 'Pipeline settings' configuration page for creating a new pipeline. The pipeline is named 'nishantpipeline'. The 'Pipeline type' section notes that V1 pipelines are deprecated and recommends using V2. The 'Execution mode' section shows 'Queued (Pipeline type V2 required)' selected. The 'Service role' section offers options for creating a new service role ('New service role') or choosing an existing one ('Existing service role'). A 'Role name' field is filled with 'AWSCodePipelineServiceRole-ap-south-1-nishantpipeline'. A checkbox at the bottom allows AWS to create a service role if it doesn't exist.

Pipeline settings

Pipeline name
Enter the pipeline name. You cannot edit the pipeline name after it is created.
 No more than 100 characters

Pipeline type
You can no longer create V1 pipelines through the console. We recommend you use the V2 pipeline type with improved release safety, pipeline triggers, parameterized pipelines, and a new billing model.

Execution mode
Choose the execution mode for your pipeline. This determines how the pipeline is run.

Superseded
A more recent execution can overtake an older one. This is the default.

Queued (Pipeline type V2 required)
Executions are processed one by one in the order that they are queued.

Parallel (Pipeline type V2 required)
Executions don't wait for other runs to complete before starting or finishing.

Service role

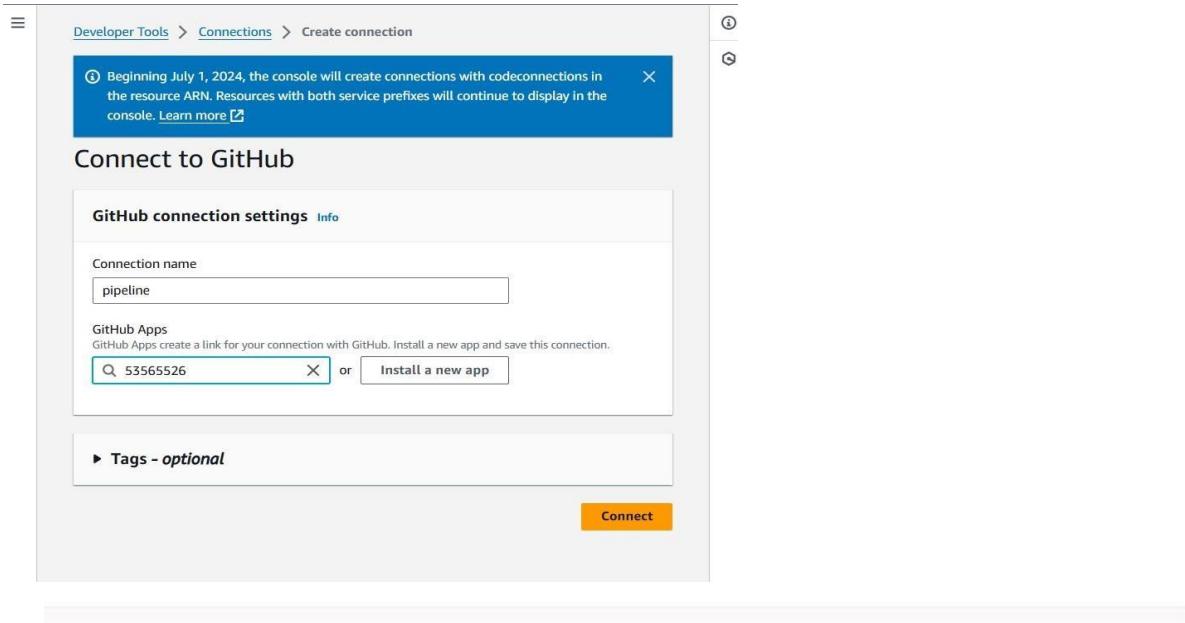
New service role
Create a service role in your account

Existing service role
Choose an existing service role from your account

Role name

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

3. In the source stage, choose GitHub v2 as the provider, then connect your GitHub account to AWS by creating a connection. You'd need your GitHub credentials and then you'd need to authorize and install AWS on the forked GitHub Repository.



Source

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

GitHub (Version 2)

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

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

or [Connect to GitHub](#)

Ready to connect
Your GitHub connection is ready for use.

Repository name
Choose a repository in your GitHub account.

You can type or paste the group path to any project that the provided credentials can access. Use the format 'group/subgroup/project'.

Default branch
Default branch will be used only when pipeline execution starts from a different source or manually started.

Output artifact format
Choose the output artifact format.

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

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

4. Then select trigger type none.

Trigger

Trigger type
Choose the trigger type that starts your pipeline.

No filter
Starts your pipeline on any push and clones the HEAD.

Specify filter
Starts your pipeline on a specific filter and clones the exact commit. Pipeline type V2 is required.

Do not detect changes
Don't automatically trigger the pipeline.

After that, click Continue and skip the build stage. Proceed to the Deployment stage.

Step 5: Deployment

1. Choose Beanstalk as the Deploy Provider, same region as the Bucket and Beanstalk, name and environment name.

Deploy

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

AWS Elastic Beanstalk ▾

Region
Asia Pacific (Mumbai) ▾

Input artifacts
Choose an input artifact for this action. [Learn more](#)

SourceArtifact ▾
No more than 100 characters

Application name
Choose an application that you have already created in the AWS Elastic Beanstalk console. Or create an application in the AWS Elastic Beanstalk console and then return to this task.

nishant

Environment name
Choose an environment that you have already created in the AWS Elastic Beanstalk console. Or create an environment in the AWS Elastic Beanstalk console and then return to this task.

nishant-env

Configure automatic rollback on stage failure

2. Then it will give you this result on screen. i.e. deployed successfully.

The screenshot shows the AWS CodePipeline interface. It consists of two main sections: 'Source' and 'Deploy'.
The 'Source' section shows a GitHub step that has succeeded. It includes details like the pipeline execution ID (a23b429a-7039-444c-8386-b03ba43a425f), the commit hash (da19c44a), and the message 'Source: Update README.md'. A large downward arrow points from the Source section to the Deploy section.
The 'Deploy' section shows an AWS Elastic Beanstalk step that has also succeeded. It includes the same pipeline execution ID and commit hash, along with the message 'Source: Update README.md'.
Below the pipeline interface is a browser window displaying a success message: 'Congratulations! You have successfully created a pipeline that retrieved this source application from an Amazon S3 bucket and deployed it to three Amazon EC2 instances using AWS CodeDeploy.' The URL in the browser bar is nishant-248-env.eba-8p3k0byus.us-east-1.elasticbeanstalk.com.

If you can see this, that means that you successfully created an automated software using CodePipeline.

Step 6: Committing changes to update app

1. In this we make some changes in the file. Open github.com then open the forked repository. Then update the changes in the index.html file and finally commit those changes.

The screenshot shows a GitHub repository page for a sample AWS CodePipeline pipeline. The repository has 1 branch and 0 tags. The master branch is up-to-date with the remote. The commit history shows 20 commits from imoisharma, starting with adding a template and dist folder, followed by s3 setup scripts, CONTRIBUTING.md, LICENSE, and finally index.html. The repository has an Apache-2.0 license, 0 stars, 0 watching, and 1 fork.

Commit changes

Commit message
Update index.html

Extended description
Add an optional extended description..

Commit directly to the `master` branch
 Create a new branch for this commit and start a pull request [Learn more about pull requests](#)

Cancel **Commit changes**

2. Then again start the deployment of the pipeline. And check the changes in the website

Hiii Nishant Khetal

You have successfully created a pipeline that retrieved this source application from an Amazon S3 bucket and deployed it to three Amazon EC2 instances using AWS CodeDeploy.

For next steps, read the AWS CodePipeline Documentation.

Experiment No. 3

Aim: To understand the Kubernetes Cluster Architecture, install and Spin Up a Kubernetes Cluster on Linux Machines/Cloud.

Steps:

1. Create 3 EC2 Ubuntu Instances on AWS. (Name 1 as Master, the other 2 as worker-1 and worker-2)

The screenshot shows the AWS EC2 'Launch an instance' wizard. The first step, 'Name and tags', has 'master' entered in the 'Name' field. The second step, 'Application and OS Images (Amazon Machine Image)', lists several AMI options: Amazon Linux, macOS, Ubuntu, Windows, Red Hat, and SUSE Linux. A search bar and a 'Browse more AMIs' link are also present. The third step, 'Key pair (login)', shows 'serverkey-01' selected in the key pair dropdown. A 'Create new key pair' button is available.

EC2 > Instances > Launch an instance

Launch an instance Info

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

Name and tags Info

Name
master

▼ Application and OS Images (Amazon Machine Image) Info

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

Recents Quick Start

Amazon Linux	macOS	Ubuntu	Windows	Red Hat	SUSE Linux
--------------	-------	--------	---------	---------	------------

Including AMIs from AWS, Marketplace and

▼ Key pair (login) Info

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

Key pair name - *required*
serverkey-01

▼ Network settings [Info](#)

[Edit](#)

Network [Info](#)
vpc-0404d393731afabc3

Subnet [Info](#)
No preference (Default subnet in any availability zone)

Auto-assign public IP [Info](#)
Enable

Additional charges apply when outside of **free tier allowance**

Firewall (security groups) [Info](#)
A security group is a set of firewall rules that control the traffic for your instance. Add rules to allow specific traffic to reach your instance.

Create security group Select existing security group

We'll create a new security group called '**launch-wizard-4**' with the following rules:

<input checked="" type="checkbox"/> Allow SSH traffic from Helps you connect to your instance	Anywhere 0.0.0.0/0
<input checked="" type="checkbox"/> Allow HTTPS traffic from the internet To set up an endpoint, for example when creating a web server	
<input checked="" type="checkbox"/> Allow HTTP traffic from the internet To set up an endpoint, for example when creating a web server	

⚠ 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. X

Instances (3) Info										
		Name ↴		Instance ID	Instance state	Status check	Alarm status	Availability Zone	Public IPv4 DNS	Public IPv4 ...
<input type="checkbox"/>	master	i-0a57da166d1f22307	Running	Running	t2.micro	Initializing	View alarms +	us-east-1c	ec2-54-165-196-241.co...	54.165.196.24
<input type="checkbox"/>	worker-2	i-09ef6bb0a0bdefc3c	Running	Running	t2.micro	2/2 checks passed	View alarms +	us-east-1c	ec2-184-72-206-70.co...	184.72.206.70
<input type="checkbox"/>	worker-1	i-04e9c3add3858c21c	Running	Running	t2.micro	Initializing	View alarms +	us-east-1c	ec2-54-211-147-10.co...	54.211.147.10

2. SSH into all 3 machines

- Give permissions to the current user to the downloaded pem file using -
chmod 400 <security_filename.pem>

```
ADMIN@Khetal MINGW64 ~ (master)
$ cd Downloads/

ADMIN@Khetal MINGW64 ~ (master)
$ chmod 400 "serverkey-01.pem"

ADMIN@Khetal MINGW64 ~/Downloads (master)
$ |
```

- ssh into all the three machines using –

ssh -i (keyname).pem (username)@(public ipv4 dns address)

where keyname is name of the key you created. (server-01.pem). Other details can be found on the Instance dashboard.

```
ADMIN@Khetal MINGW64 ~ (master)
$ ssh -i "serverkey-01.pem" ec2-user@ec2-54-165-196-241.compute-1.amazonaws.com
,
~\_\_ #####      Amazon Linux 2023
~~ \####\_
~~ \###|
~~ \#/ .--> https://aws.amazon.com/linux/amazon-linux-2023
~~ V~' .-
~~ / .
~~ .-. / .
~/m/ '
[ec2-user@ip-172-31-90-103 ~]$
```

3. Installation Of Docker on three machines

```
[ec2-user@ip-172-31-90-103 ~]$ sudo yum install docker -y
Last metadata expiration check: 0:13:41 ago on Sat Sep 14 03:42:27 2024.
Dependencies resolved.
=====
Package          Arch    Version        Repository      Size
=====
Installing:
docker           x86_64  25.0.6-1.amzn2023.0.2  amazonlinux   44 M
Installing dependencies:
containerd       x86_64  1.7.20-1.amzn2023.0.1   amazonlinux   35 M
iptables-libs   x86_64  1.8.8-3.amzn2023.0.2   amazonlinux   401 k
iptables-nft    x86_64  1.8.8-3.amzn2023.0.2   amazonlinux   183 k
libcgroup        x86_64  3.0-1.amzn2023.0.1   amazonlinux   75 k
libnetfilter_conntrack x86_64  1.0.8-2.amzn2023.0.2  amazonlinux   58 k
libnftnetlink    x86_64  1.0.1-19.amzn2023.0.2  amazonlinux   30 k
libnftnl         x86_64  1.2.2-2.amzn2023.0.2  amazonlinux   84 k
pigz             x86_64  2.5-1.amzn2023.0.3   amazonlinux   83 k
runc             x86_64  1.1.13-1.amzn2023.0.1   amazonlinux   3.2 M
=====
Transaction summary
=====
Install 10 Packages

Total download size: 84 M
Installed size: 317 M
Downloading Packages:
(1/10): iptables-libs-1.8.8-3.amzn2023.0.2.x86_2.2 MB/s | 401 kB    00:00
(2/10): iptables-nft-1.8.8-3.amzn2023.0.2.x86_6 2.5 MB/s | 183 kB    00:00
(3/10): libcgroup-3.0-1.amzn2023.0.1.x86_64.rpm 1.3 MB/s | 75 kB    00:00
(4/10): libnetfilter_conntrack-1.0.8-2.amzn2023 1.3 MB/s | 58 kB    00:00
(5/10): libnftnetlink-1.0.1-19.amzn2023.0.2.x86_938 kB/s | 30 kB    00:00
(6/10): libnftnl-1.2.2-2.amzn2023.0.2.x86_64.rpm 1.6 MB/s | 84 kB    00:00
(7/10): pigz-2.5-1.amzn2023.0.3.x86_64.rpm     1.7 MB/s | 83 kB    00:00
(8/10): runc-1.1.13-1.amzn2023.0.1.x86_64.rpm   21 MB/s | 3.2 MB   00:00
(9/10): containerd-1.7.20-1.amzn2023.0.1.x86_64 31 MB/s | 35 MB   00:01
(10/10): docker-25.0.6-1.amzn2023.0.2.x86_64.rpm 28 MB/s | 44 MB   00:01
=====
Total                                         52 MB/s | 84 MB   00:01
Running transaction check
Transaction check succeeded.
Running transaction test
Transaction test succeeded.
Running transaction
  Preparing      :
  Installing    : runc-1.1.13-1.amzn2023.0.1.x86_64
  Installing    : containerd-1.7.20-1.amzn2023.0.1.x86_64
  Running scriptlet: containerd-1.7.20-1.amzn2023.0.1.x86_64
  Installing    : pigz-2.5-1.amzn2023.0.3.x86_64
  Installing    : libnftnl-1.2.2-2.amzn2023.0.2.x86_64
  Installing    : libnftnetlink-1.0.1-19.amzn2023.0.2.x86_64
  Installing    : libnetfilter_conntrack-1.0.8-2.amzn2023.0.2.x86_64
  Installing    : iptables-libs-1.8.8-3.amzn2023.0.2.x86_64
  Installing    : iptables-nft-1.8.8-3.amzn2023.0.2.x86_64
```

- **sudo yum install docker -y**

```
Installed:
  containerd-1.7.20-1.amzn2023.0.1.x86_64
  libnetfilter_conntrack-1.0.8-2.amzn2023.0.2.x86_64
  docker-25.0.6-1.amzn2023.0.2.x86_64
  libnfnetlink-1.0.1-19.amzn2023.0.2.x86_64

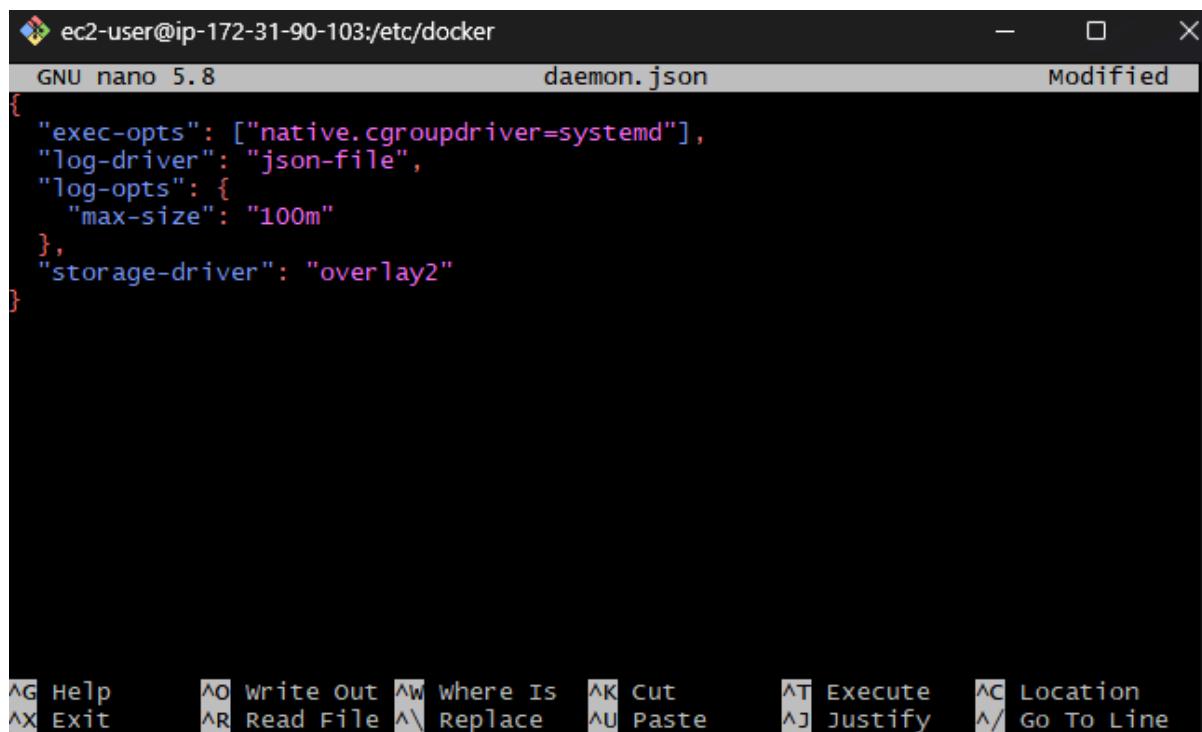
Complete!
[ec2-user@ip-172-31-90-103 ~]$
```

- Configure cgroup in a daemon.json

(this can be done by creating the file and using **nano** text editor)

```
{
  "exec-opts": ["native.cgroupdriver=systemd"],
  "log-driver": "json-file",
  "log-opt": {
    "max-size": "100m"
  },
  "storage-driver": "overlay2"
}
```

```
[ec2-user@ip-172-31-90-103 docker]$ sudo nano daemon.json
[ec2-user@ip-172-31-90-103 docker]$ |
```



- Enable and start docker and also load the daemon.json

```
sudo systemctl enable docker
```

```
sudo systemctl daemon-reload
```

```
sudo systemctl restart docker
```

```
[ec2-user@ip-172-31-90-103 docker]$ sudo systemctl enable docker
Created symlink /etc/systemd/system/multi-user.target.wants/docker.service → /usr/lib/systemd/system/docker.service.
[ec2-user@ip-172-31-90-103 docker]$ sudo systemctl daemon-reload
[ec2-user@ip-172-31-90-103 docker]$ 
sudo systemctl restart docker
[ec2-user@ip-172-31-90-103 docker]$ |
```

- Check if docker is installed

```
[ec2-user@ip-172-31-90-103 docker]$ docker --version
Docker version 25.0.5, build 5dc9bcc
[ec2-user@ip-172-31-90-103 docker]$ |
```

4. Install Kubernetes on all 3 machines

- SELinux needs to be disabled before configuring kubelet

```
sudo setenforce 0
```

```
sudo sed -i 's/^SELINUX=enforcing$/SELINUX=permissive/' /etc/selinux/config
```

```
[ec2-user@ip-172-31-90-103 docker]$ sudo setenforce 0
[sudo] password for ec2-user:
sudo sed -i 's/^SELINUX=enforcing$/SELINUX=permissive/' /etc/selinux/config
[ec2-user@ip-172-31-90-103 docker]$ |
```

- Add Kubernetes using the repo

(this is done by creating **kubernetes.repo** file in **/etc/yum.repos.d** and configuring it using **nano** editor)
[**kubernetes**]

```
name=Kubernetes
```

```
baseurl=https://pkgs.k8s.io/core:/stable:/v1.30/rpm/
```

```
enabled=1
```

```
gpgcheck=1
```

```
gpgkey=https://pkgs.k8s.io/core:/stable:/v1.30/rpm/repo/repodata/repomd.xml.key
```

```
exclude=kubelet kubeadm kubectl cri-tools kubernetes-cni
```

```
[ec2-user@ip-172-31-90-103 docker]$ cd /etc/yum.repos.d/
[ec2-user@ip-172-31-90-103 yum.repos.d]$ ls
amazonlinux.repo  kernel-livepatch.repo
[ec2-user@ip-172-31-90-103 yum.repos.d]$ sudo nano kubernetes.repo
[ec2-user@ip-172-31-90-103 yum.repos.d]$ ls
amazonlinux.repo  kernel-livepatch.repo  kubernetes.repo
[ec2-user@ip-172-31-90-103 yum.repos.d]$ |
```

The screenshot shows a terminal window with the following content:

```
ec2-user@ip-172-31-90-103:/etc/yum.repos.d
GNU nano 5.8                               kubernetes.repo                         Modified
[kubernetes]
name=Kubernetes
baseurl=https://pkgs.k8s.io/core:/stable:/v1.30/rpm/
enabled=1
gpgcheck=1
gpgkey=https://pkgs.k8s.io/core:/stable:/v1.30/rpm/repo/repodata/repomd.xml.key
exclude=kubelet kubeadm kubectl cri-tools kubernetes-cni
```

At the bottom of the terminal window, there is a menu bar with the following options:

- AG Help
- AO Write Out
- AW Where Is
- AK Cut
- AT Execute
- AC Location
- AX Exit
- AR Read File
- AV Replace
- AU Paste
- AJ Justify
- ^/ Go To Line

- Update packages list using **sudo yum update**

```
[ec2-user@ip-172-31-90-103 yum.repos.d]$ sudo yum update
Kubernetes                                         125 kB/s | 17 kB     00:00
Dependencies resolved.
Nothing to do.
Complete!
[ec2-user@ip-172-31-90-103 yum.repos.d]$ |
```

- Install kubelet kubeadm kubectl

```
sudo yum install -y kubelet kubeadm kubectl --disableexcludes=kubernetes
```

```
[ec2-user@ip-172-31-90-103 yum.repos.d]$ sudo yum install -y kubelet kubeadm kubectl --disableexcludes=kubernetes
Last metadata expiration check: 0:00:42 ago on Sat Sep 14 04:08:20 2024.
Dependencies resolved.
```

Package	Arch	Version	Repository	size
<hr/>				
Installing:				
kubeadm	x86_64	1.30.5-150500.1.1	kubernetes	10 M
kubectl	x86_64	1.30.5-150500.1.1	kubernetes	10 M
kubelet	x86_64	1.30.5-150500.1.1	kubernetes	17 M
<hr/>				
Installing dependencies:				
conntrack-tools	x86_64	1.4.6-2.amzn2023.0.2	amazonlinux	208 k
cri-tools	x86_64	1.30.1-150500.1.1	kubernetes	8.6 M
kubernetes-cni	x86_64	1.4.0-150500.1.1	kubernetes	6.7 M
libnetfilter_cthelper	x86_64	1.0.0-21.amzn2023.0.2	amazonlinux	24 k
libnetfilter_cttimeout	x86_64	1.0.0-19.amzn2023.0.2	amazonlinux	24 k
libnetfilter_queue	x86_64	1.0.5-2.amzn2023.0.2	amazonlinux	30 k
<hr/>				
Transaction summary				
Install	9 Packages			
<hr/>				
Total download size: 53 M				
Installed size: 292 M				
Downloading Packages:				
(1/9): libnetfilter_cttimeout-1.0.0-19.amzn2023	448 kB/s 24 kB	00:00		
(2/9): libnetfilter_cthelper-1.0.0-21.amzn2023.	409 kB/s 24 kB	00:00		
(3/9): libnetfilter_queue-1.0.5-2.amzn2023.0.2.	1.5 MB/s 30 kB	00:00		
(4/9): conntrack-tools-1.4.6-2.amzn2023.0.2.x86	1.8 MB/s 208 kB	00:00		
(5/9): cri-tools-1.30.1-150500.1.1.x86_64.rpm	28 MB/s 8.6 MB	00:00		
(6/9): kubectl-1.30.5-150500.1.1.x86_64.rpm	23 MB/s 10 MB	00:00		
(7/9): kubeadm-1.30.5-150500.1.1.x86_64.rpm	18 MB/s 10 MB	00:00		
(8/9): kubelet-1.30.5-150500.1.1.x86_64.rpm	37 MB/s 17 MB	00:00		
(9/9): kubernetes-cni-1.4.0-150500.1.1.x86_64.r	20 MB/s 6.7 MB	00:00		
<hr/>				
Total	56 MB/s 53 MB	00:00		

```
Installed:
  conntrack-tools-1.4.6-2.amzn2023.0.2.x86_64
  cri-tools-1.30.1-150500.1.1.x86_64
  kubeadm-1.30.5-150500.1.1.x86_64
  kubectl-1.30.5-150500.1.1.x86_64
  kubelet-1.30.5-150500.1.1.x86_64
  kubernetes-cni-1.4.0-150500.1.1.x86_64
  libnetfilter_cthelper-1.0.0-21.amzn2023.0.2.x86_64
  libnetfilter_cttimeout-1.0.0-19.amzn2023.0.2.x86_64
  libnetfilter_queue-1.0.5-2.amzn2023.0.2.x86_64

Complete!
[ec2-user@ip-172-31-90-103 yum.repos.d]$
```

- After installing Kubernetes, we need to configure internet options to allow bridging.

```
sudo swapoff -a
echo "net.bridge.bridge-nf-call-iptables=1" | sudo tee -a /etc/sysctl.conf
sudo sysctl -p
```

```
[ec2-user@ip-172-31-90-103 yum.repos.d]$ . sudo swapoff -a
-bash: sudo: No such file or directory
[ec2-user@ip-172-31-90-103 yum.repos.d]$ sudo swapoff -a
[ec2-user@ip-172-31-90-103 yum.repos.d]$ echo "net.bridge.bridge-nf-call-iptables=1" | sudo tee -a /etc/sysctl.conf
net.bridge.bridge-nf-call-iptables=1
[ec2-user@ip-172-31-90-103 yum.repos.d]$ sudo sysctl -p
net.bridge.bridge-nf-call-iptables = 1
[ec2-user@ip-172-31-90-103 yum.repos.d]$ |
```

5. Perform this ONLY on the Master machine Initialize the Kubecluster

```
sudo kubeadm init --podnetwork-cidr=10.244.0.0/16
```

```
[ec2-user@ip-172-31-90-103 yum.repos.d]$ sudo kubeadm init --pod-network-cidr=10.244.0.0/16 --ignore-preflight-errors=all
I0914 04:12:17.448521 27990 version.go:256] remote version is much newer: v1.3.1.0; falling back to: stable-1.30
[init] using Kubernetes version: v1.30.4
[preflight] Running pre-flight checks
    [WARNING NumCPU]: the number of available CPUs 1 is less than the required 2
    [WARNING Mem]: the system RAM (949 MB) is less than the minimum 1700 MB
    [WARNING FileExisting-socat]: socat not found in system path
    [WARNING FileExisting-tc]: tc not found in system path
    [WARNING Service-Kubelet]: kubelet service is not enabled, please run 'systemctl enable kubelet.service'
[preflight] Pulling images required for setting up a Kubernetes cluster
[preflight] This might take a minute or two, depending on the speed of your internet connection
[preflight] You can also perform this action in beforehand using 'kubeadm config images pull'
w0914 04:12:17.711154 27990 checks.go:844] detected that the sandbox image "registry.k8s.io/pause:3.8" of the container runtime is inconsistent with that used by kubeadm. It is recommended to use "registry.k8s.io/pause:3.9" as the CRI sandbox image.
[certs] Using certificateDir folder "/etc/kubernetes/pki"
[certs] Generating "ca" certificate and key
[certs] Generating "apiserver" certificate and key
```

```
Your Kubernetes control-plane has initialized successfully!
```

```
To start using your cluster, you need to run the following as a regular user:
```

```
mkdir -p $HOME/.kube
sudo cp -i /etc/kubernetes/admin.conf $HOME/.kube/config
sudo chown $(id -u):$(id -g) $HOME/.kube/config
```

```
Alternatively, if you are the root user, you can run:
```

```
export KUBECONFIG=/etc/kubernetes/admin.conf
```

```
You should now deploy a pod network to the cluster.
```

```
Run "kubectl apply -f [podnetwork].yaml" with one of the options listed at:
https://kubernetes.io/docs/concepts/cluster-administration/addons/
```

```
Then you can join any number of worker nodes by running the following on each as root:
```

```
kubeadm join 172.31.90.103:6443 --token 0zk8w3.xyegkydsy42vfscm \
--discovery-token-ca-cert-hash sha256:31c672892b19dcb869fc46362d189234128f5bfc302bd41ae8c6078c56173f00
[ec2-user@ip-172-31-90-103 yum.repos.d]$
```

□ Save the token

```
kubeadm join 172.31.90.103:6443 --token 0zk8w3.xyegkydsy42vfscm \
--discovery-token-ca-cert-hash
sha256:31c672892b19dcb869fc46362d189234128f5bfc302bd41ae8c6078c56173f00
```

□ Copy the mkdir and chown commands from the top and execute them

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

```
[ec2-user@ip-172-31-90-103 yum.repos.d]$ mkdir -p $HOME/.kube
[ec2-user@ip-172-31-90-103 yum.repos.d]$ |
```

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

```
[ec2-user@ip-172-31-90-103 yum.repos.d]$ sudo cp -i /etc/kubernetes/admin.conf $HOME/.kube/config
[ec2-user@ip-172-31-90-103 yum.repos.d]$ |
sudo chown $(id -u):$(id -g) $HOME/.kube/config
```

```
[ec2-user@ip-172-31-90-103 yum.repos.d]$ |
sudo chown $(id -u):$(id -g) $HOME/.kube/config
[ec2-user@ip-172-31-90-103 yum.repos.d]$ |
```

- Then, add a common networking plugin called flammel file as mentioned in the code.

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

```
[ec2-user@ip-172-31-90-103 yum.repos.d]$ kubectl apply -f https://raw.githubusercontent.com/coreos/flannel/master/Documentation/kube-flannel.yml
namespace/kube-flannel created
clusterrole.rbac.authorization.k8s.io/flannel created
clusterrolebinding.rbac.authorization.k8s.io/flannel created
serviceaccount/flannel created
configmap/kube-flannel-cfg created
daemonset.apps/kube-flannel-ds created
[ec2-user@ip-172-31-90-103 yum.repos.d]$
```

6. Perform this ONLY on the worker machines

- Paste the below command on all 2 worker machines

```
sudo yum install iproute-tc -y
sudo systemctl enable kubelet
sudo systemctl restart kubelet
```

- Now use the token from earlier to join into worker instances

```
kubeadm join 172.31.90.103:6443 --token 0zk8w3.xyegkydsy42vfscm \
--discovery-token-ca-cert-hash
sha256:31c672892b19dc869fc46362d189234128f5bfc302bd41ae8c6078c56173f00
```

- **kubectl get nodes** to check whether master and worker nodes are connected successfully

```
[ec2-user@ip-54-211-147-10 docker]$ kubectl get nodes
NAME           STATUS   ROLES      AGE     VERSION
ip-172-31-90-103.ec2.internal   Ready    control-plane   3m21s   v1.30.5
```

Conclusion:

An EC2 instance was created on AWS Linux, and Docker, Kubernetes, Kubelet, Kubeadm, and Kubectl were installed. Kubernetes was initialized on the master node, which provided a token for connecting the master and worker nodes. On the slave node, `iproute` was installed, and Kubelet was enabled and restarted. However, there was an issue with joining the slave node to the cluster, resulting in only the master node being listed when running `kubectl get nodes`

Experiment No. 4

Aim: To install Kubectl and execute Kubectl commands to manage the Kubernetes cluster and deploy Your First Kubernetes Application.

Steps:

1. Create an EC2 Linux Instances on AWS.

The screenshot shows the 'Launch an instance' step in the AWS EC2 wizard. It includes sections for 'Name and tags', 'Application and OS Images (Amazon Machine Image)', and 'Key pair (login)'. The 'Name and tags' section has 'nishant' entered. The 'Application and OS Images' section shows various AMI options like Amazon Linux, macOS, Ubuntu, Windows, Red Hat, and SUSE. The 'Key pair (login)' section shows 'serverkey-01' selected.

Name and tags Info

Name
nishant Add additional tags

▼ Application and OS Images (Amazon Machine Image) Info

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

Search our full catalog including 1000s of application and OS images

Recents Quick Start

Amazon Linux macOS Ubuntu Windows Red Hat SUSE Li Search Browse more AMIs Including AMIs from

▼ Key pair (login) Info

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

Key pair name - *required*
serverkey-01 ▼ Create new key pair

▼ Network settings [Info](#)

[Edit](#)[Network](#) | [Info](#)

vpc-0404d393731afabc3

[Subnet](#) | [Info](#)

No preference (Default subnet in any availability zone)

[Auto-assign public IP](#) | [Info](#)

Enable

[Additional charges apply](#) when outside of [free tier allowance](#)[Firewall \(security groups\)](#) | [Info](#)

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

 [Create security group](#) [Select existing security group](#)

We'll create a new security group called '**launch-wizard-4**' with the following rules:

 [Allow SSH traffic from](#)

Helps you connect to your instance

Anywhere

▼

0.0.0.0/0

 [Allow HTTPS traffic from the internet](#)

To set up an endpoint, for example when creating a web server

 [Allow HTTP traffic from the internet](#)

To set up an endpoint, for example when creating a web server



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.



Instances (2) [Info](#)

[Connect](#)

Instance state ▾

Actions ▾

[Launch instances](#) ▾

<input type="text"/> Find instance by attribute or tag (case-sensitive)		All states ▾	< 1 >	①			
Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public IP
Nishant Khetal	i-0341f2e761db28aec	Running ⓘ ⓘ	t2.micro	2/2 checks passed View alarms +	View alarms +	us-east-1c	ec2-52-
Nishant	i-0689f78090ca339a9	Running ⓘ ⓘ	t2.micro	2/2 checks passed View alarms +	View alarms +	us-east-1c	ec2-34-

2. Then click on Id of that instance then click on connect

The screenshot shows the 'Connect to instance' dialog box. At the top, it says 'EC2 > Instances > i-0a57da166d1f22307 > Connect to instance'. Below that, it says 'Connect to instance i-0a57da166d1f22307 (nishant) using any of these options'. There are four tabs: 'EC2 Instance Connect' (selected), 'Session Manager', 'SSH client', and 'EC2 serial console'. A yellow warning box states: 'Port 22 (SSH) is open to all IPv4 addresses. Port 22 (SSH) is currently open to all IPv4 addresses, indicated by 0.0.0.0/0 in the inbound rule in [your security group](#). For increased security, consider restricting access to only the EC2 Instance Connect service IP addresses for your Region: 18.206.107.24/29. [Learn more.](#)'

Instance ID: i-0a57da166d1f22307 (abhinav)

Connection Type:

- Connect using EC2 Instance Connect: Connect using the EC2 Instance Connect browser-based client, with a public IPv4 address.
- Connect using EC2 Instance Connect Endpoint: Connect using the EC2 Instance Connect browser-based client, with a private IPv4 address and a VPC endpoint.

Public IPv4 address: 54.165.196.241

Username: ec2-user

Note: In most cases, the default username, ec2-user, is correct. However, read your AMI usage instructions to check if the AMI owner has changed the default AMI username.

Buttons: Cancel, Connect

3. SSH into the created machine instance

- Give permissions to the current user to the downloaded pem file using -
chmod 400 <security_filename.pem>

```
ADMIN@Khetal MINGW64 ~ (master)
$ cd Downloads/

ADMIN@Khetal MINGW64 ~ (master)
$ chmod 400 "serverkey-01.pem"

ADMIN@Khetal MINGW64 ~/Downloads (master)
$ |
```

- Ssh using –

ssh -i (keyname).pem (username)@(public ipv4 dns address)

where keyname is name of the key you created. (server-01.pem). Other details can be found on the Instance dashboard.

```
ADMIN@Khetal MINGW64 ~ (master)
$ ssh -i "serverkey-01.pem" ec2-user@ec2-54-165-196-241.compute-1.amazonaws.com
,
~\_\_ ##### Amazon Linux 2023
~~ \_\_\#\#\#
~~ \#\#\#
~~ \#/ , __ https://aws.amazon.com/linux/amazon-linux-2023
~~ V~, .->
~~ /
~~ ._. /
~/ /-
/_m/
[ec2-user@ip-172-31-90-103 ~]$
```

4. Installation Of Docker007

- sudo yum install docker -y

```
[ec2-user@ip-172-31-90-103 ~]$ sudo yum install docker -y
Last metadata expiration check: 0:13:41 ago on Sat Sep 14 03:42:27 2024.
Dependencies resolved.
=====
Package          Arch    Version           Repository      Size
=====
Installing:
  docker        x86_64  25.0.6-1.amzn2023.0.2   amazonlinux   44 M
Installing dependencies:
  containerd    x86_64  1.7.20-1.amzn2023.0.1   amazonlinux   35 M
  iptables-libs x86_64  1.8.8-3.amzn2023.0.2   amazonlinux   401 k
  iptables-nft  x86_64  1.8.8-3.amzn2023.0.2   amazonlinux   183 k
  libcgroup     x86_64  3.0-1.amzn2023.0.1   amazonlinux   75 k
  libnetfilter_conntrack x86_64  1.0.8-2.amzn2023.0.2   amazonlinux   58 k
  libnftnlink   x86_64  1.0.1-19.amzn2023.0.2   amazonlinux   30 k
  libnftnl     x86_64  1.2.2-2.amzn2023.0.2   amazonlinux   84 k
  pigz         x86_64  2.5-1.amzn2023.0.3   amazonlinux   83 k
  runc         x86_64  1.1.13-1.amzn2023.0.1   amazonlinux   3.2 M

Transaction Summary
=====
Install 10 Packages

Total download size: 84 M
Installed size: 317 M
Downloading Packages:
(1/10): iptables-libs-1.8.8-3.amzn2023.0.2.x86_2.2 MB/s | 401 kB    00:00
(2/10): iptables-nft-1.8.8-3.amzn2023.0.2.x86_6 2.5 MB/s | 183 kB    00:00
(3/10): libcgroup-3.0-1.amzn2023.0.1.x86_64.rpm 1.3 MB/s | 75 kB    00:00
(4/10): libnetfilter_conntrack-1.0.8-2.amzn2023 1.3 MB/s | 58 kB    00:00
(5/10): libnftnlink-1.0.1-19.amzn2023.0.2.x86_938 kB/s | 30 kB    00:00
(6/10): libnftnl-1.2.2-2.amzn2023.0.2.x86_64.rp 1.6 MB/s | 84 kB    00:00
(7/10): pigz-2.5-1.amzn2023.0.3.x86_64.rpm    1.7 MB/s | 83 kB    00:00
(8/10): runc-1.1.13-1.amzn2023.0.1.x86_64.rpm   21 MB/s | 3.2 MB   00:00
(9/10): containerd-1.7.20-1.amzn2023.0.1.x86_64 31 MB/s | 35 MB   00:01
(10/10): docker-25.0.6-1.amzn2023.0.2.x86_64.rp 28 MB/s | 44 MB   00:01
-----
Total                                         52 MB/s | 84 MB   00:01

Running transaction check
Transaction check succeeded.
Running transaction test
Transaction test succeeded.
Running transaction
  Preparing   :
  Installing  : runc-1.1.13-1.amzn2023.0.1.x86_64
  Installing  : containerd-1.7.20-1.amzn2023.0.1.x86_64
  Running scriptlet: containerd-1.7.20-1.amzn2023.0.1.x86_64
  Installing  : pigz-2.5-1.amzn2023.0.3.x86_64
  Installing  : libnftnl-1.2.2-2.amzn2023.0.2.x86_64
  Installing  : libnftnlink-1.0.1-19.amzn2023.0.2.x86_64
  Installing  : libnetfilter_conntrack-1.0.8-2.amzn2023.0.2.x86_64
  Installing  : iptables-libs-1.8.8-3.amzn2023.0.2.x86_64
  Installing  : iptables-nft-1.8.8-3.amzn2023.0.2.x86_64
```

```
Installed:
  containerd-1.7.20-1.amzn2023.0.1.x86_64
  libnetfilter_conntrack-1.0.8-2.amzn2023.0.2.x86_64
  docker-25.0.6-1.amzn2023.0.2.x86_64
  libnfnetlink-1.0.1-19.amzn2023.0.2.x86_64

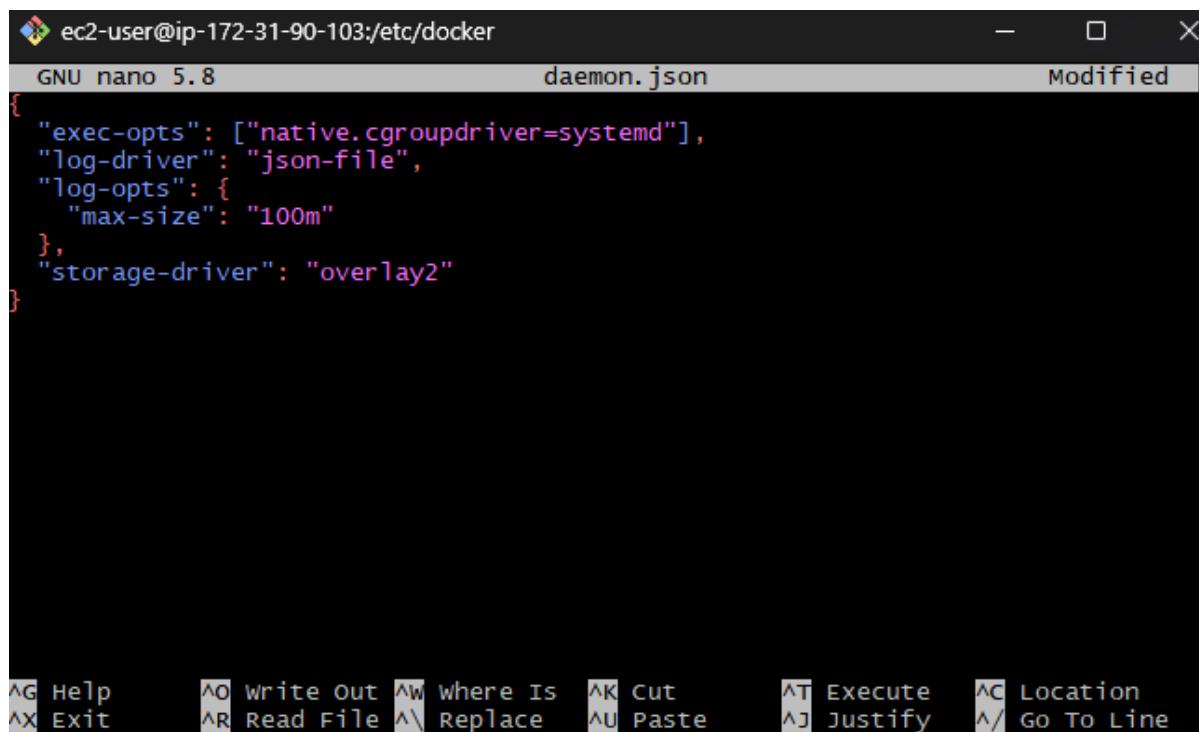
Complete!
[ec2-user@ip-172-31-90-103 ~]$
```

- Configure cgroup in a daemon.json

(this can be done by creating the file and using **nano** text editor)

```
{
  "exec-opts": ["native.cgroupdriver=systemd"],
  "log-driver": "json-file",
  "log-opts": {
    "max-size": "100m"
  },
  "storage-driver": "overlay2"
}
```

```
[ec2-user@ip-172-31-90-103 docker]$ sudo nano daemon.json
[ec2-user@ip-172-31-90-103 docker]$ |
```



- Enable and start docker and also load the daemon.json

```
sudo systemctl enable docker
```

```
sudo systemctl daemon-reload
```

```
sudo systemctl restart docker
```

```
[ec2-user@ip-172-31-90-103 docker]$ sudo systemctl enable docker
created symlink /etc/systemd/system/multi-user.target.wants/docker.service → /usr/lib/systemd/system/docker.service.
[ec2-user@ip-172-31-90-103 docker]$ sudo systemctl daemon-reload
[ec2-user@ip-172-31-90-103 docker]$
sudo systemctl restart docker
[ec2-user@ip-172-31-90-103 docker]$ |
```

- Check if docker is installed

```
[ec2-user@ip-172-31-90-103 docker]$ docker --version
Docker version 25.0.5, build 5dc9bcc
[ec2-user@ip-172-31-90-103 docker]$ |
```

5. Install Kubernetes

- SELinux needs to be disabled before configuring kubelet

```
sudo setenforce 0
```

```
sudo sed -i 's/^SELINUX=enforcing$/SELINUX=permissive/' /etc/selinux/config
```

```
[ec2-user@ip-172-31-90-103 docker]$ sudo setenforce 0
[sudo] password for ec2-user:
sudo sed -i 's/^SELINUX=enforcing$/SELINUX=permissive/' /etc/selinux/config
[ec2-user@ip-172-31-90-103 docker]$ |
```

- Add Kubernetes using the repo

(this is done by creating **kubernetes.repo** file in **/etc/yum.repos.d** and configuring it using **nano** editor)

```
[kubernetes]
```

```
name=Kubernetes
```

```
baseurl=https://pkgs.k8s.io/core:/stable:/v1.30/rpm/
```

```
enabled=1
```

```
gpgcheck=1
```

```
gpgkey=https://pkgs.k8s.io/core:/stable:/v1.30/rpm/repo/repodata/repomd.xml.key
```

```
exclude=kubelet kubeadm kubectl cri-tools kubernetes-cni
```

```
[ec2-user@ip-172-31-90-103 docker]$ cd /etc/yum.repos.d/
[ec2-user@ip-172-31-90-103 yum.repos.d]$ ls
amazonlinux.repo  kernel-livepatch.repo
[ec2-user@ip-172-31-90-103 yum.repos.d]$ sudo nano kubernetes.repo
[ec2-user@ip-172-31-90-103 yum.repos.d]$ ls
amazonlinux.repo  kernel-livepatch.repo  kubernetes.repo
[ec2-user@ip-172-31-90-103 yum.repos.d]$ |
```

```
ec2-user@ip-172-31-90-103:/etc/yum.repos.d
GNU nano 5.8          kubernetes.repo          Modified
[kubernetes]
name=Kubernetes
baseurl=https://pkgs.k8s.io/core:/stable:/v1.30/rpm/
enabled=1
gpgcheck=1
gpgkey=https://pkgs.k8s.io/core:/stable:/v1.30/rpm/repo/repodata/repomd.xml.key
exclude=kubelet kubeadm kubectl cri-tools kubernetes-cni
```

AG Help AO Write Out AW Where Is AK Cut AT Execute AC Location
AX Exit AR Read File AV Replace AU Paste AJ Justify ^/ Go To Line

- Update packages list using **sudo yum update**

```
[ec2-user@ip-172-31-90-103 yum.repos.d]$ sudo yum update
Kubernetes                               125 kB/s | 17 kB     00:00
Dependencies resolved.
Nothing to do.
Complete!
[ec2-user@ip-172-31-90-103 yum.repos.d]$
```

- Install kubelet kubeadm kubectl

```
sudo yum install -y kubelet kubeadm kubectl --disableexcludes=kubernetes
```

```
[ec2-user@ip-172-31-90-103 yum.repos.d]$ sudo yum install -y kubelet kubeadm kubectl --disableexcludes=kubernetes
Last metadata expiration check: 0:00:42 ago on Sat Sep 14 04:08:20 2024.
Dependencies resolved.
```

Package	Arch	Version	Repository	size
<hr/>				
Installing:				
kubeadm	x86_64	1.30.5-150500.1.1	kubernetes	10 M
kubectl	x86_64	1.30.5-150500.1.1	kubernetes	10 M
kubelet	x86_64	1.30.5-150500.1.1	kubernetes	17 M
<hr/>				
Installing dependencies:				
conntrack-tools	x86_64	1.4.6-2.amzn2023.0.2	amazonlinux	208 k
cri-tools	x86_64	1.30.1-150500.1.1	kubernetes	8.6 M
kubernetes-cni	x86_64	1.4.0-150500.1.1	kubernetes	6.7 M
libnetfilter_cthelper	x86_64	1.0.0-21.amzn2023.0.2	amazonlinux	24 k
libnetfilter_cttimeout	x86_64	1.0.0-19.amzn2023.0.2	amazonlinux	24 k
libnetfilter_queue	x86_64	1.0.5-2.amzn2023.0.2	amazonlinux	30 k
<hr/>				
Transaction summary				
Install	9 Packages			
<hr/>				
Total download size: 53 M				
Installed size: 292 M				
Downloading Packages:				
(1/9): libnetfilter_cttimeout-1.0.0-19.amzn2023	448 kB/s 24 kB	00:00		
(2/9): libnetfilter_cthelper-1.0.0-21.amzn2023.	409 kB/s 24 kB	00:00		
(3/9): libnetfilter_queue-1.0.5-2.amzn2023.0.2.	1.5 MB/s 30 kB	00:00		
(4/9): conntrack-tools-1.4.6-2.amzn2023.0.2.x86	1.8 MB/s 208 kB	00:00		
(5/9): cri-tools-1.30.1-150500.1.1.x86_64.rpm	28 MB/s 8.6 MB	00:00		
(6/9): kubectl-1.30.5-150500.1.1.x86_64.rpm	23 MB/s 10 MB	00:00		
(7/9): kubeadm-1.30.5-150500.1.1.x86_64.rpm	18 MB/s 10 MB	00:00		
(8/9): kubelet-1.30.5-150500.1.1.x86_64.rpm	37 MB/s 17 MB	00:00		
(9/9): kubernetes-cni-1.4.0-150500.1.1.x86_64.r	20 MB/s 6.7 MB	00:00		
<hr/>				
Total	56 MB/s 53 MB	00:00		

```
Installed:
  conntrack-tools-1.4.6-2.amzn2023.0.2.x86_64
  cri-tools-1.30.1-150500.1.1.x86_64
  kubeadm-1.30.5-150500.1.1.x86_64
  kubectl-1.30.5-150500.1.1.x86_64
  kubelet-1.30.5-150500.1.1.x86_64
  kubernetes-cni-1.4.0-150500.1.1.x86_64
  libnetfilter_cthelper-1.0.0-21.amzn2023.0.2.x86_64
  libnetfilter_cttimeout-1.0.0-19.amzn2023.0.2.x86_64
  libnetfilter_queue-1.0.5-2.amzn2023.0.2.x86_64

Complete!
[ec2-user@ip-172-31-90-103 yum.repos.d]$
```

- After installing Kubernetes, we need to configure internet options to allow bridging.

```
sudo swapoff -a
echo "net.bridge.bridge-nf-call-iptables=1" | sudo tee -a /etc/sysctl.conf
sudo sysctl -p
```

```
[ec2-user@ip-172-31-90-103 yum.repos.d]$ . sudo swapoff -a
-bash: sudo: No such file or directory
[ec2-user@ip-172-31-90-103 yum.repos.d]$ sudo swapoff -a
[ec2-user@ip-172-31-90-103 yum.repos.d]$ echo "net.bridge.bridge-nf-call-iptables=1" | sudo tee -a /etc/sysctl.conf
net.bridge.bridge-nf-call-iptables=1
[ec2-user@ip-172-31-90-103 yum.repos.d]$ sudo sysctl -p
net.bridge.bridge-nf-call-iptables = 1
[ec2-user@ip-172-31-90-103 yum.repos.d]$ |
```

6. Initialize the Kubecluster

```
sudo kubeadm init --podnetwork-cidr=10.244.0.0/16
```

```
[ec2-user@ip-172-31-90-103 yum.repos.d]$ sudo kubeadm init --pod-network-cidr=10.244.0.0/16 --ignore-preflight-errors=all
I0914 04:12:17.448521 27990 version.go:256] remote version is much newer: v1.3.1.0; falling back to: stable-1.30
[init] using Kubernetes version: v1.30.4
[preflight] Running pre-flight checks
    [WARNING NumCPU]: the number of available CPUs 1 is less than the required 2
    [WARNING Mem]: the system RAM (949 MB) is less than the minimum 1700 MB
    [WARNING FileExisting-socat]: socat not found in system path
    [WARNING FileExisting-tc]: tc not found in system path
    [WARNING Service-Kubelet]: kubelet service is not enabled, please run 'systemctl enable kubelet.service'
[preflight] Pulling images required for setting up a Kubernetes cluster
[preflight] This might take a minute or two, depending on the speed of your internet connection
[preflight] You can also perform this action in beforehand using 'kubeadm config images pull'
w0914 04:12:17.711154 27990 checks.go:844] detected that the sandbox image "registry.k8s.io/pause:3.8" of the container runtime is inconsistent with that used by kubeadm. It is recommended to use "registry.k8s.io/pause:3.9" as the CRI sandbox image.
[certs] Using certificateDir folder "/etc/kubernetes/pki"
[certs] Generating "ca" certificate and key
[certs] Generating "apiserver" certificate and key
```

```
Your Kubernetes control-plane has initialized successfully!
```

```
To start using your cluster, you need to run the following as a regular user:
```

```
mkdir -p $HOME/.kube
sudo cp -i /etc/kubernetes/admin.conf $HOME/.kube/config
sudo chown $(id -u):$(id -g) $HOME/.kube/config
```

```
Alternatively, if you are the root user, you can run:
```

```
export KUBECONFIG=/etc/kubernetes/admin.conf
```

```
You should now deploy a pod network to the cluster.
```

```
Run "kubectl apply -f [podnetwork].yaml" with one of the options listed at:
https://kubernetes.io/docs/concepts/cluster-administration/addons/
```

```
Then you can join any number of worker nodes by running the following on each as root:
```

```
kubeadm join 172.31.90.103:6443 --token 0zk8w3.xyegkydsy42vfscm \
--discovery-token-ca-cert-hash sha256:31c672892b19dcb869fc46362d189234128f5bfc302bd41ae8c6078c56173f00
[ec2-user@ip-172-31-90-103 yum.repos.d]$
```

□ Save the token

```
kubeadm join 172.31.90.103:6443 --token 0zk8w3.xyegkydsy42vfscm \
--discovery-token-ca-cert-hash
sha256:31c672892b19dcb869fc46362d189234128f5bfc302bd41ae8c6078c56173f00
```

□ Copy the mkdir and chown commands from the top and execute them

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

```
[ec2-user@ip-172-31-90-103 yum.repos.d]$ mkdir -p $HOME/.kube
[ec2-user@ip-172-31-90-103 yum.repos.d]$ |
```

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

```
[ec2-user@ip-172-31-90-103 yum.repos.d]$ sudo cp -i /etc/kubernetes/admin.conf $HOME/.kube/config
[ec2-user@ip-172-31-90-103 yum.repos.d]$ 
sudo chown $(id -u):$(id -g) $HOME/.kube/config
```

```
[ec2-user@ip-172-31-90-103 yum.repos.d]$ 
sudo chown $(id -u):$(id -g) $HOME/.kube/config
[ec2-user@ip-172-31-90-103 yum.repos.d]$
```

- Then, add a common networking plugin called flammel file as mentioned in the code.

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

```
[ec2-user@ip-172-31-90-103 yum.repos.d]$ kubectl apply -f https://raw.githubusercontent.com/coreos/flannel/master/Documentation/kube-flannel.yml
namespace/kube-flannel created
clusterrole.rbac.authorization.k8s.io/flannel created
clusterrolebinding.rbac.authorization.k8s.io/flannel created
serviceaccount/flannel created
configmap/kube-flannel-cfg created
daemonset.apps/kube-flannel-ds created
[ec2-user@ip-172-31-90-103 yum.repos.d]$
```

7. Deploy nginx server

- Apply deployment using this following command:

```
kubectl apply -f https://k8s.io/examples/pods/simple-pod.yaml
```

```
[ec2-user@ip-172-31-90-103 docker]$ kubectl apply -f https://k8s.io/examples/pods/simple-pod.yaml
pod/nginx created
```

- use `kubectl get nodes` to check whether the pod gets created or not

```
[ec2-user@ip-172-31-90-103 docker]$ kubectl get pods
NAME      READY   STATUS    RESTARTS   AGE
nginx    0/1     Pending   0          12s
```

`kubectl describe pod nginx` (This command will help to describe the pods it gives reason for failure as it shows the untolerated taints which need to be untainted.)

```
[ec2-user@ip-172-31-90-103 docker]$ kubectl describe pod nginx
Name:           nginx
Namespace:      default
Priority:       0
Service Account: default
Node:           <none>
Labels:          <none>
Annotations:    <none>
Status:         Pending
IP:
IPs:            <none>
Containers:
  nginx:
    Image:        nginx:1.14.2
    Port:         80/TCP
    Host Port:   0/TCP
    Environment: <none>
    Mounts:
      /var/run/secrets/kubernetes.io/serviceaccount from kube-api-access-k4lj6 (ro)
      Type:          Projected (a volume that contains injected data from m
      ultiple sources)
      TokenExpirationSeconds: 3607
      ConfigMapName:  kube-root-ca.crt
      ConfigMapOptional: <nil>
      DownwardAPI:   true
QoS Class:      BestEffort
Node-Selectors: <none>
Tolerations:    node.kubernetes.io/not-ready:NoExecute op=Exists for 3
                 0s
                           node.kubernetes.io/unreachable:NoExecute op=Exists for
                           300s
Events:
  Type      Reason     Age   From           Message
  ----      ----     --   --           --
  Warning   FailedScheduling 7s   default-scheduler 0/1 nodes are available: 1 no
de(s) had untolerated taint {node-role.kubernetes.io/control-plane: }. preemption:
0/1 nodes are available: 1 Preemption is not helpful for scheduling.
```

- check pod status

```
[ec2-user@ip-172-31-90-103 ~]$ kubectl get pods
NAME      READY   STATUS    RESTARTS   AGE
nginx     1/1     Running   1 (6s ago)  90s
```

```
[ec2-user@ip-172-31-90-103 ~]$ kubectl port-forward nginx 8081:80
Forwarding from 127.0.0.1:8081 -> 80
Forwarding from [::1]:8081 -> 80
```

8. Verify your deployment

- Open up a new terminal and ssh to your EC2 instance. Then, use this curl command to check if the Nginx server is running. `curl --head http://127.0.0.1:8080` If the response is 200 OK and you can see the Nginx server name, your deployment was successful. We have successfully deployed our Nginx server on our EC2 instance.

```
[ec2-user@172-31-90-103 ~]$ curl --head http://127.0.0.1:8080
HTTP/1.1 200 OK
Server: nginx/1.18.0 (Linux)
Date: Sat, 14 Sep 2024 12:31:53 GMT
Content-Type: text/html
Content-Length: 612
Connection: keep-alive
```

Conclusion:

An AWS EC2 Linux instance was set up, and Docker and Kubernetes were installed. Kubernetes was initialized successfully, and the required commands were executed. Flannel was installed as a networking plugin. Although there was an initial error with the Nginx deployment, it was eventually deployed successfully using the `simple-pod.yml` file and accessed via localhost on port 8080.

Name: Nishant S Khetal

D15C

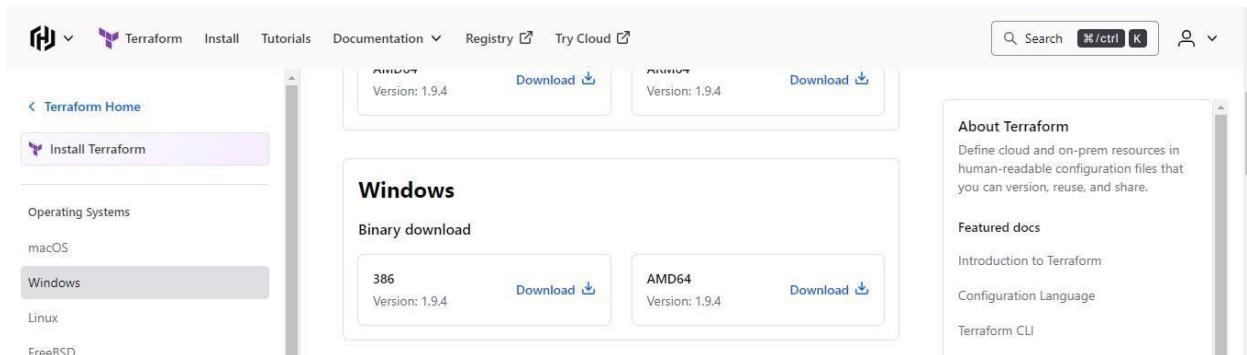
Roll No: 24

Experiment No. 5

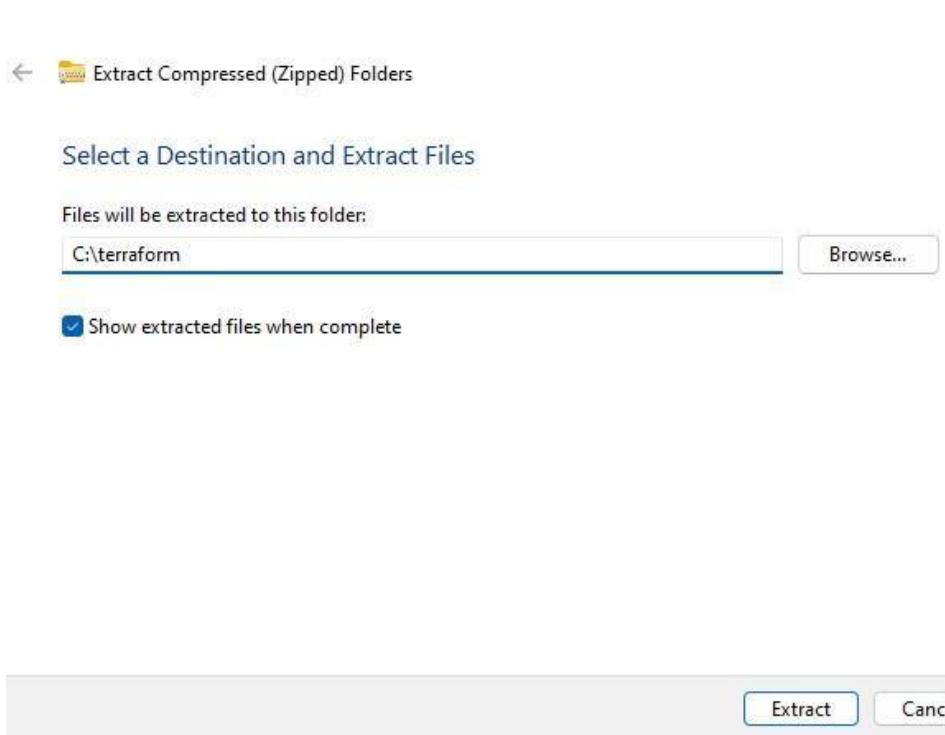
Installation of Terraform

Step 1: To install Terraform, go to the official Terraform website linked below. Navigate to the Downloads section, choose Windows, and download the 64-bit version for your system.

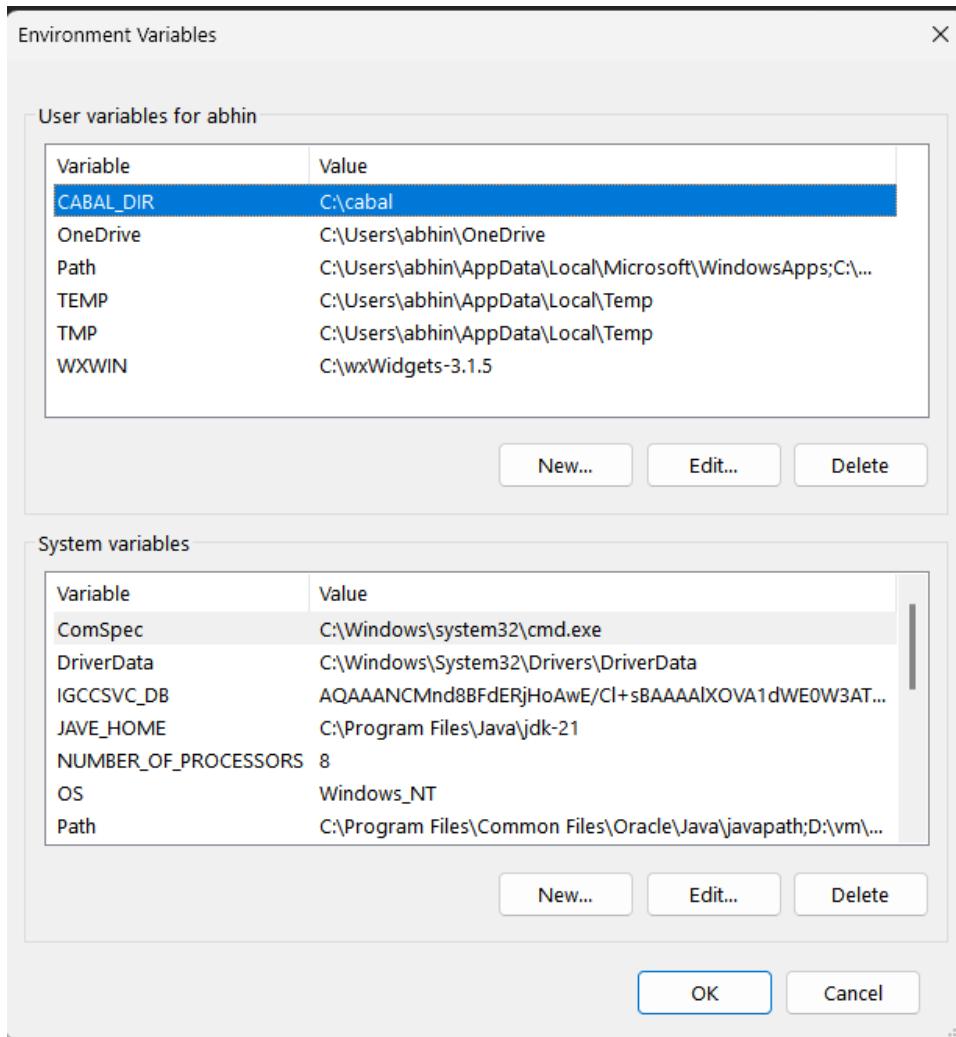
Website: <https://www.terraform.io/downloads.html>



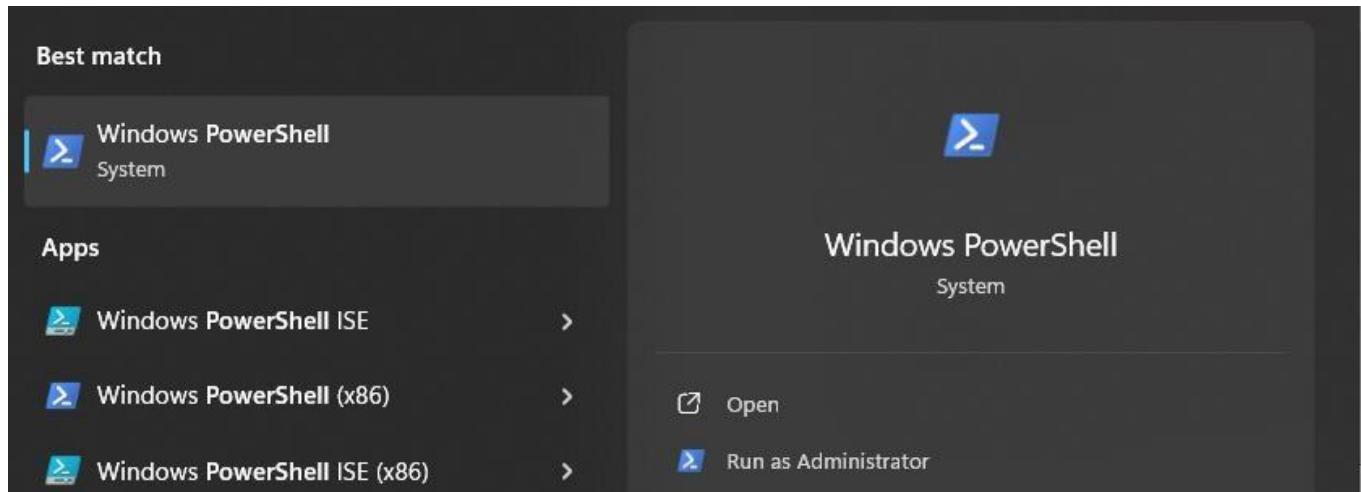
Step 2: Extract the downloaded `Terraform.exe` file to the `C:\Terraform` directory on your computer.



Step 3: Configure the system path for Terraform in the Environment Variables.



Step 4: Launch Windows PowerShell with administrator privileges.



Step 5: Run the command `terraform` to confirm its functionality. If any errors occur, review and update the Terraform path in your environment variables as needed.

```
Administrator: Windows Powr X + | v
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Install the latest PowerShell for new features and improvements! https://aka.ms/PSWindows

PS C:\Users\abhin> terraform
Usage: terraform [global options] <subcommand> [args]

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

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

All other commands:
  console        Try Terraform expressions at an interactive command prompt
  fmt            Reformat your configuration in the standard style
  force-unlock  Release a stuck lock on the current workspace
  get            Install or upgrade remote Terraform modules
  graph          Generate a Graphviz graph of the steps in an operation
  import         Associate existing infrastructure with a Terraform resource
  login          Obtain and save credentials for a remote host
  logout         Remove locally-stored credentials for a remote host
  metadata       Metadata related commands
  output         Show output values from your root module
  providers     Show the providers required for this configuration
  refresh        Update the state to match remote systems
  show           Show the current state or a saved plan
  state          Advanced state management
  taint          Mark a resource instance as not fully functional
```

Experiment No.6

- Creating a docker image using terraform

```
C:\Users\nishant>docker --version
Docker version 26.1.3, build b72abbb

C:\Users\nishant>docker

Usage: docker [OPTIONS] COMMAND
      A self-sufficient runtime for containers

Common Commands:
  run      Create and run a new container from an image
  exec    Execute a command in a running container
  ps      List containers
  build   Build an image from a Dockerfile
  pull    Download an image from a registry
  push    Upload an image to a registry
  images  List images
  login   Log in to a registry
  logout  Log out from a registry
  search  Search Docker Hub for images
  version Show the Docker version information
  info    Display system-wide information

Management Commands:
  builder  Manage builds
  buildx* Docker Buildx
  compose* Docker Compose
  container Manage containers
  context   Manage contexts
```

- Create a folder named ‘Terraform Scripts’ in which we save our different types of scripts which will be further used in this experiment.



- Create a new folder named ‘Docker’ in the ‘TerraformScripts’ folder. Then create a new docker.tf file using a text editor and write the following contents into it to create a Ubuntu Linux container.

This PC > Temporary Storage (D:) > Terraform_Scripts >				
	Name	Date modified	Type	Size
ss	Docker	8/22/2024 4:10 PM	File folder	
This PC > Temporary Storage (D:) > Terraform_Scripts > Docker				
	Name	Date modified	Type	Size
fr	docker.tf	8/22/2024 4:12 PM	TF File	1 KB

 docker.tf - Notepad

```
File Edit Format View Help
terraform {
    required_providers {
        docker = {
            source = "kreuzwerker/docker"
            version = "2.21.0"
        }
    }
}
provider "docker" {
host = "npipe://./pipe/docker_engine"
}

# Pulls the image
resource "docker_image" "ubuntu" {
    name = "ubuntu:latest"
}

# Create a container
resource "docker_container" "foo" {
    image = docker_image.ubuntu.image_id
    name = "foo"
}
```

4. Execute Terraform Init command to initialize the resources

```
D:\Terraform_Scripts\Docker>terraform init
Initializing the backend...
Initializing provider plugins...
- Finding kreuzwerker/docker versions matching "2.21.0"...
- Installing kreuzwerker/docker v2.21.0...
- Installed kreuzwerker/docker v2.21.0 (self-signed, key ID BD080C4571C6104C)
  Partner and community providers are signed by their developers.
  If you'd like to know more about provider signing, you can read about it here:
  https://www.terraform.io/docs/cli/plugins/signing.html
  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.

D:\Terraform_Scripts\Docker>
```

5. Execute Terraform plan to see the available resources

```
D:\Terraform_Scripts\Docker>terraform plan

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

Terraform will perform the following actions:

# docker_container.foo will be created
+ resource "docker_container" "foo" {
  + attach          = false
  + bridge          = (known after apply)
  + command         = (known after apply)
  + container_logs = (known after apply)
  + entrypoint      = (known after apply)
  + env             = (known after apply)
  + exit_code       = (known after apply)
  + gateway         = (known after apply)
  + hostname        = (known after apply)
  + id              = (known after apply)
  + image           = (known after apply)
  + init            = (known after apply)
  + ip_address      = (known after apply)
  + ip_prefix_length = (known after apply)
  + ipc_mode        = (known after apply)
  + log_driver      = (known after apply)
  + logs            = false
  + must_run        = true
  + name            = "foo"
  + network_data    = (known after apply)
  + read_only       = false
  + remove_volumes = true
  + restart         = "no"
  + rm              = false
  + runtime          = (known after apply)
  + security_opts   = (known after apply)
  + shm_size         = (known after apply)
  + start            = true
  + stdin_open      = false
  + stop_signal      = (known after apply)
  + stop_timeout     = (known after apply)
  + tty              = false

  + healthcheck (known after apply)

  + labels (known after apply)
}

# docker_image.ubuntu will be created
+ resource "docker_image" "ubuntu" {
  + id          = (known after apply)
  + image_id    = (known after apply)
  + latest      = (known after apply)
  + name        = "ubuntu:latest"
  + output      = (known after apply)
  + repo_digest = (known after apply)
}

Plan: 2 to add, 0 to change, 0 to destroy.
```

6. Execute Terraform apply to apply the configuration, which will automatically create and run the Ubuntu Linux container based on our configuration. Using command : “terraform apply”

```
D:\Terraform_Scripts\Dockers>terraform apply

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

Terraform will perform the following actions:

# docker_container.foo will be created
+ resource "docker_container" "foo" {
  + attach           = false
  + bridge           = (known after apply)
  + command          = (known after apply)
  + container_logs   = (known after apply)
  + entrypoint        = (known after apply)
  + env               = (known after apply)
  + exit_code         = (known after apply)
  + gateway           = (known after apply)
  + hostname          = (known after apply)
  + id                = (known after apply)
  + image              = (known after apply)
  + init               = (known after apply)
  + ip_address         = (known after apply)
  + ip_prefix_length  = (known after apply)
  + ipc_mode           = (known after apply)
  + log_driver          = (known after apply)
  + logs               = false
  + must_run           = true
  + name               = "foo"
  + network_data       = (known after apply)
  + read_only           = false
  + remove_volumes     = true
  + restart             = "no"
  + rm                 = false
  + runtime             = (known after apply)
  + security_opts       = (known after apply)
  + shm_size            = (known after apply)
  + start               = true
  + stdio_open          = false
  + stop_signal          = (known after apply)
  + stop_timeout         = (known after apply)
  + tty                 = false

  + healthcheck (known after apply)

  + labels (known after apply)
}

# docker_image.ubuntu will be created
+ resource "docker_image" "ubuntu" {
  + id                = (known after apply)
  + image_id          = (known after apply)
  + latest             = (known after apply)
  + name               = "ubuntu:latest"
  + output              = (known after apply)
  + repo_digest         = (known after apply)
}

Plan: 2 to add, 0 to change, 0 to destroy.

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

Enter a value: yes

docker image.ubuntu: Creating...
docker image.ubuntu: Still creating... [19s elapsed] docker image.ubuntu: Still creating... (20s elapsed) docker image.ubuntu: Still creating... [30s elapsed]
docker image.ubuntu: Creation complete after 30s [id=sha256:263966596d42ad38ae9914716692777ba9ff8779a62ad93a74fe82e3e1f
ubuntu:latest] docker_container.foo: Creating...
```

7. Check Docker images, Before and After Executing Apply step

REPOSITORY	TAG	IMAGE ID	CREATED	SIZE
mcr.microsoft.com/dotnet/framework/aspnet	4.8-windowsservercore-ltsc2022	0b1ef1176a57	6 weeks ago	5.43GB
mcr.microsoft.com/dotnet/framework/sdk	4.8-windowsservercore-ltsc2022	c3f8c2735565	6 weeks ago	9.04GB
mcr.microsoft.com/dotnet/framework/runtime	4.8-windowsservercore-ltsc2022	e69ea8a5ec1b	6 weeks ago	5.1GB
mcr.microsoft.com/windows/servercore	ltsc2022	e60f47e635b7	7 weeks ago	4.84GB
mcr.microsoft.com/windows/nanoserver	ltsc2022	f0ca29645006	7 weeks ago	292MB

REPOSITORY	TAG	IMAGE ID	CREATED	SIZE
mcr.microsoft.com/dotnet/framework/aspnet	4.8-windowsservercore-ltsc2022	0b1ef1176a57	6 weeks ago	5.43GB
mcr.microsoft.com/dotnet/framework/sdk	4.8-windowsservercore-ltsc2022	c3f8c2735565	6 weeks ago	9.04GB
mcr.microsoft.com/dotnet/framework/runtime	4.8-windowsservercore-ltsc2022	e69ea8a5ec1b	6 weeks ago	5.1GB
mcr.microsoft.com/windows/servercore	ltsc2022	e60f47e635b7	7 weeks ago	4.84GB
mcr.microsoft.com/windows/nanoserver	ltsc2022	f0ca29645006	7 weeks ago	292MB
ubuntu	Latest	2dc39ba859dc	2 minutes ago	77.8MB

- Execute Terraform destroy to delete the configuration, which will automatically delete the Ubuntu Container.

```
D:\Terraform_Scripts\ Docker>terraform destroy
docker_image.ubuntu: Refreshing state... [id=sha256:2dc29b50dc2d30101475692777ba087762d92de0221fubuntu:latest]
Terraform used the selected providers to generate the following execution plan. Resource actions are indicated with the following symbols:
  destroy
Terraform will perform the following actions:

# docker_image.ubuntu will be destroyed
resource "docker_image" "ubuntu" {
  - id      = "sha256:2dc39b859dc2ade8ae30147b5692777ba9ff9779a62ad93a78de82e3elfubuntu:latest" -> null
  - image_id = "sha256:2dc39b859dc2ade8ae30147b5692777ba9ff9779a62ad93a78de82e3elf" -> null
  - Latest   = "sha256:2dc39b859dc2ade8ae30147b5692777ba9ff9779a62ad93a78de82e3elf" -> null
  - name     = "ubuntu:latest" -> null
  - repo digest = "ubuntu@sha256:204a3d7bb4d7723452be3923b06cd7043704030041c83c#7856c1" -> null
}

Plan: to add, 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

docker image.ubuntu: Destroying... [id=sha256:2de99b59cd42ade83814765692777ba5ff8779a62ad93ad62e3elfubuntu:latest]
docker image.ubuntu: Destruction complete after 0s

Destroy complete! Resources: 1 destroyed.
```

- Check Docker images, After Executing Destroy step

REPOSITORY	TAG	IMAGE ID	CREATED	SIZE
mcr.microsoft.com/dotnet/framework/aspnet	4.8-windowsservercore-ltsc2022	0b1ef1176a57	6 weeks ago	5.43GB
mcr.microsoft.com/dotnet/framework/sdk	4.8-windowsservercore-ltsc2022	c3f8c2735565	6 weeks ago	9.04GB
mcr.microsoft.com/dotnet/framework/runtime	4.8-windowsservercore-ltsc2022	e69ea8a5ec1b	6 weeks ago	5.1GB
mcr.microsoft.com/windows/servercore	ltsc2022	e60f47e635b7	7 weeks ago	4.84GB
mcr.microsoft.com/windows/nanoserver	ltsc2022	f0ca29645006	7 weeks ago	292MB

Experiment No: 07

Aim: To investigate Static Analysis (SAST) techniques and demonstrate the integration of Jenkins with SonarQube for assessing code quality..

Theory:

Overview of SAST:

Static application security testing (SAST), or static analysis, is a testing methodology that analyzes source code to find security vulnerabilities that make your organization's applications susceptible to attack. SAST scans an application before the code is compiled. It's also known as white box testing.

Challenges Addressed by SAST:

SAST takes place very early in the software development life cycle (SDLC) as it does not require a working application and can take place without code being executed. It helps developers identify vulnerabilities in the initial stages of development and quickly resolve issues without breaking builds or passing on vulnerabilities to the final release of the application.

SAST tools give developers real-time feedback as they code, helping them fix issues before they pass the code to the next phase of the SDLC. This prevents security-related issues from being considered an afterthought. SAST tools also provide graphical representations of the issues found, from source to sink. These help you navigate the code easier. Some tools point out the exact location of vulnerabilities and highlight the risky code. Tools can also provide in-depth guidance on how to fix issues and the best place in the code to fix them, without requiring deep security domain expertise.

It's important to note that SAST tools must be run on the application on a regular basis, such as during daily/monthly builds, every time code is checked in, or during a code release.

Importance of SAST:

Developers dramatically outnumber security staff. It can be challenging for an organization to find the resources to perform code reviews on even a fraction of its applications. A key strength of SAST tools is the ability to analyze 100% of the codebase. Additionally, they are much faster than manual secure code reviews performed by humans. These tools can scan millions of lines of code in a matter of minutes. SAST tools automatically identify critical vulnerabilities—such as buffer overflows, SQL injection, cross-site scripting, and others—with high confidence. Thus, integrating static analysis into the SDLC can yield dramatic results in the overall quality of the code developed.

Key Steps for Effective SAST:

There are six simple steps needed to perform SAST efficiently in organizations that have a very large number of applications built with different languages, frameworks, and platforms.

1. **Finalize the tool.** Select a static analysis tool that can perform code reviews of applications written in the programming languages you use. The tool should also be able to comprehend the underlying framework used by your software.
2. **Setup Infrastructure:** This step involves handling the licensing requirements, setting up access control and authorization, and procuring the resources required (e.g., servers and databases) to deploy the tool.
3. **Customize the tool.** Fine-tune the tool to suit the needs of the organization. For example, you might configure it to reduce false positives or find additional security vulnerabilities by writing new rules or updating existing ones. Integrate the tool into the build environment, create dashboards for tracking scan results, and build custom reports.
4. **Prioritize and onboard applications.** Once the tool is ready, onboard your applications. If you have a large number of applications, prioritize the high-risk applications to scan first. Eventually, all your applications should be onboarded and scanned regularly, with application scans synced with release cycles, daily or monthly builds, or code check-ins.
5. **Analyze scan results.** This step involves triaging the results of the scan to remove false positives. Once the set of issues is finalized, they should be tracked and provided to the deployment teams for proper and timely remediation.
6. **Educate developers.** Proper governance ensures that your development teams are employing the scanning tools properly. The software security touchpoints should be present within the SDLC. SAST should be incorporated as part of your application development and deployment process.

Integrating Jenkins with SonarQube:

Prerequisites:

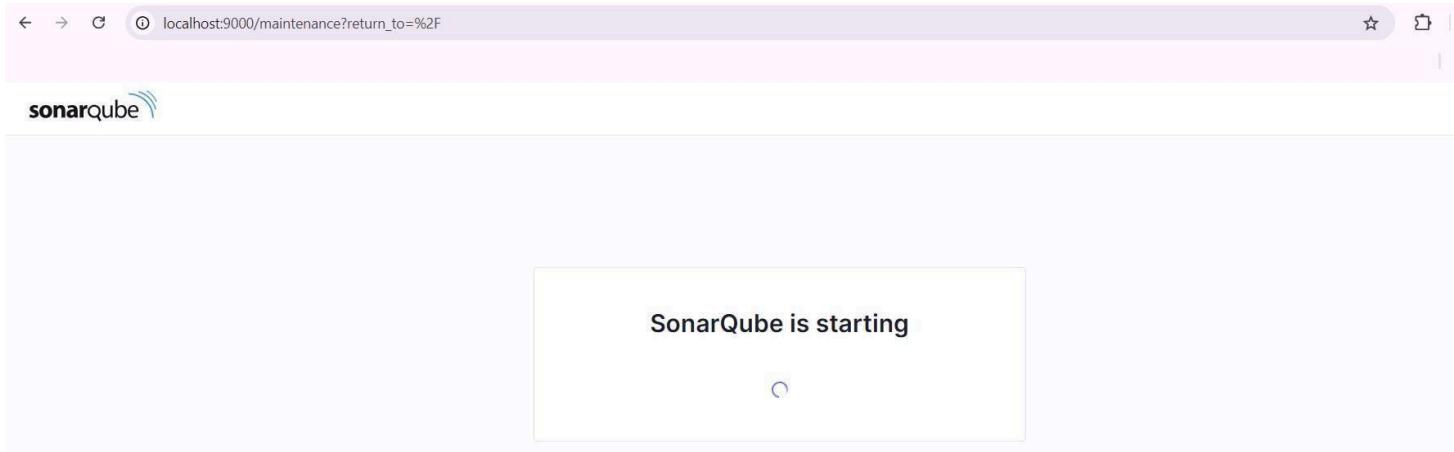
- Installed Jenkins
- Docker setup for SonarQube
- Access to the SonarQube Docker image

Steps to integrate Jenkins with SonarQube

1. Navigate to Jenkins at localhost:8000.
2. Launch SonarQube using Docker and confirm accessibility at port 9000.

```
C:\Users\ADMIN>docker run -d --name sonarqube -e SONAR_ES_BOOTSTRAP_CHECKS_DISABLE=true -p 9000:9000 sonarqube:latest
Unable to find image 'sonarqube:latest' locally
latest: Pulling from library/sonarqube
7478e0ac0f23: Pull complete
90a925ab929a: Pull complete
7d9a34308537: Pull complete
80338217a4ab: Pull complete
1a5fd5c7e184: Pull complete
7b87d6fa783d: Pull complete
bd819c9b5ead: Pull complete
4f4fb700ef54: Pull complete
Digest: sha256:72e9fec71242af83faf65f95a40d5e3bb2822a6c3b2cda8568790f3d31aecde
Status: Downloaded newer image for sonarqube:latest
de76efbeef2054aeb442b86ba54c2916039b8757b388482d9780ffc69f5d8bbe
```

3. Once the container is up and running, you can check the status of SonarQube at localhost port 9000.



4. Login to SonarQube using username **admin** and password **admin**.
5. In Jenkins, install the SonarQube Scanner plugin to facilitate the analysis.

Setup the project and come back to Jenkins Dashboard.

6. Configure the connection settings for SonarQube within Jenkins, entering credentials

A screenshot of the Jenkins Marketplace search results. A search bar at the top contains the text "sonar". To the right of the search bar is a blue "Install" button. Below the search bar, the results are listed. The first result is "SonarQube Scanner 2.17.2", which is marked as "Released". It has two tabs: "External Site/Tool Integrations" and "Build Reports". A description below the tabs states: "This plugin allows an easy integration of SonarQube, the open source platform for Continuous Inspection of code quality." To the right of the description is the text "7 mo 9 days ago".

7. Create a new freestyle project in Jenkins and link it to a test repository hosted on GitHub.

A screenshot of the Jenkins System configuration page under "Manage Jenkins > System > SonarQube installations". The page title is "SonarQube installations" and the subtitle is "List of SonarQube installations". There is a form with fields for "Name" (containing "sonarqube"), "Server URL" (containing "http://localhost:9000"), and "Server authentication token" (containing "- none -"). A "+ Add" button is also present. An "Advanced" dropdown menu is visible at the bottom right.

8. Search for SonarQube Scanner under Global Tool Configuration. Choose the latest configuration and choose Install automatically.

A screenshot of the Jenkins Global Tool Configuration page under "Manage Jenkins > Global Tool Configuration > SonarQube Scanner installations". The page title is "SonarQube Scanner installations". It shows a single entry for "sonarqube" with the "Install automatically" checkbox checked. A detailed view of the "sonarqube" entry is shown, titled "Install from Maven Central". It includes a "Version" field containing "SonarQube Scanner 6.1.0.4477" and an "Add Installer" dropdown menu.

9. After the configuration, create a New Item in Jenkins, choose a freestyle project.

New Item

Enter an item name

SonarQube

Select an item type



Freestyle project

Classic, general-purpose job type that checks out from up to one SCM, executes build steps serially, followed by post-build steps like archiving artifacts and sending email notifications.

10. Choose this GitHub repository in Source Code Management.

Source Code Management

None

Git ?

Repositories ?

Repository URL ?

https://github.com/shazforiot/MSBuild_firstproject.git

Credentials ?

- none -

+ Add

Advanced

https://github.com/shazforiot/MSBuild_firstproject.git

It is a sample hello-world project with no vulnerabilities and issues, just to test the integration.

11. Under Build-> Execute SonarQube Scanner, enter these Analysis properties. Mention the SonarQube Project Key, Login, Password, Source path and Host URL.

The screenshot shows the Jenkins configuration interface for a build step. The left sidebar has tabs: General, Source Code Management, Build Triggers, Build Environment, Build Steps (which is selected), and Post-build Actions. The main area is titled 'Build Steps' and contains a section for 'Execute SonarQube Scanner'. It includes fields for 'SonarQube Installation' (set to 'sonarqube'), 'JDK' (set to '(Inherit From Job)'), 'Path to project properties' (empty), 'Analysis properties' (containing the following text: sonar.projectKey=sonarqube, sonar.login=admin, sonar.password=admin123, sonar.sources=C:\\ProgramData\\Jenkins\\jenkins\\workspace\\SonarQube, sonar.host.url=http://127.0.0.1:9000), 'Additional arguments' (empty), and 'JVM Options' (containing -Dsonar.ws.timeout=300). At the bottom are 'Save' and 'Apply' buttons.

12. Go to http://localhost:9000/<user_name>/permissions and allow Execute Permissions to the Admin user.

The screenshot shows the Jenkins User Management page. At the top, there are links for 'Administer System', 'Administer', 'Execute Analysis', and 'Create'. Below that, a table lists users with their roles: 'A Administrator admin'. To the right of the table are checkboxes for 'Quality Gates' (unchecked), 'Quality Profiles' (unchecked), and 'Projects' (checked). The 'Projects' checkbox is highlighted with a blue checkmark.

13. Run The Build.

The screenshot shows the Jenkins Project Summary page for a project. The left sidebar has buttons: Status (selected), Changes, Workspace, Build Now (selected), Configure, Delete Project, SonarQube, and Rename.

Check the console output.

Console Output

[Download](#) [Copy](#) [View as plain text](#)

```
Started by user Nishant Khetal
Running as SYSTEM
Building on the built-in node in workspace C:\ProgramData\Jenkins\.jenkins\workspace\SonarQube
The recommended git tool is: NONE
No credentials specified
> git.exe rev-parse --resolve-git-dir C:\ProgramData\Jenkins\.jenkins\workspace\SonarQube\.git # timeout=10
Fetching changes from the remote Git repository
> git.exe config remote.origin.url https://github.com/shazforiot/MSBuild_firstproject.git # timeout=10
Fetching upstream changes from https://github.com/shazforiot/MSBuild_firstproject.git
> git.exe -version # timeout=10
> git --version # 'git version 2.45.2.windows.1'
> git.exe fetch --tags --force --progress -- https://github.com/shazforiot/MSBuild_firstproject.git +refs/heads/*:refs/remotes/origin/* # timeout=10
> git.exe rev-parse "refs/remotes/origin/master^{commit}" # timeout=10
Checking out Revision f2bc042c04c6e72427c380bcaee6d6fee7b49adf (refs/remotes/origin/master)
> git.exe config core.sparsecheckout # timeout=10
> git.exe checkout -f f2bc042c04c6e72427c380bcaee6d6fee7b49adf # timeout=10
Commit message: "updated"
> git.exe rev-list --no-walk f2bc042c04c6e72427c380bcaee6d6fee7b49adf # timeout=10
[SonarQube] $ C:\ProgramData\Jenkins\.jenkins\tools\hudson.plugins.sonar.SonarRunnerInstallation\sonarqube\bin\sonar-scanner.bat -Dsonar.host.url=http://localhost:9000 -Dsonar.projectKey=sonarqube -Dsonar.login=admin -Dsonar.host.url=http://127.0.0.1:9000 -Dsonar.sources=C:\ProgramData\Jenkins\.jenkins\workspace\SonarQube -Dsonar.password=admin123 -Dsonar.projectBaseDir=C:\ProgramData\Jenkins\.jenkins\workspace\SonarQube
16:16:39.198 WARN Property 'sonar.host.url' with value 'http://localhost:9000' is overridden with value 'http://127.0.0.1:9000'
16:16:39.206 INFO Scanner configuration file: C:\ProgramData\Jenkins\.jenkins\tools\hudson.plugins.sonar.SonarRunnerInstallation\sonarqube\bin..\conf\sonar-scanner.properties
16:16:39.206 INFO Project root configuration file: NONE
16:16:39.230 INFO SonarScanner CLI 6.1.0.4477
16:16:39.230 INFO Java 21.0.4 Eclipse Adoptium (64-bit)
16:16:39.230 INFO Windows 11 10.0 amd64
16:16:39.230 INFO SONAR_SCANNER_OPTS=-Dsonar.ws.timeout=300
16:16:39.254 INFO User cache: C:\Windows\system32\config\systemprofile\.sonar\cache
```

```
16:16:58.734 INFO Using git CLI to retrieve untracked files
16:16:58.791 INFO Analyzing language associated files and files included via "sonar.text.inclusions" that are tracked by git
16:16:58.856 INFO 14 source files to be analyzed
16:16:59.154 INFO 14/14 source files have been analyzed
16:16:59.154 INFO Sensor TextAndSecretsSensor [text] (done) | time=1306ms
16:16:59.163 INFO -----
16:16:59.373 INFO Sensor C# [csharp]
16:16:59.373 WARN Your project contains C# files which cannot be analyzed with the scanner you are using. To analyze C# or VB.NET, you must use the SonarScanner for .NET 5.x or higher, see https://redirect.sonarsource.com/doc/install-configure-scanner-msbuild.html
16:16:59.373 INFO Sensor C# [csharp] (done) | time=0ms
16:16:59.373 INFO Sensor Analysis Warnings import [csharp]
16:16:59.379 INFO Sensor Analysis Warnings import [csharp] (done) | time=0ms
16:16:59.379 INFO Sensor C# File Caching Sensor [csharp]
16:16:59.379 WARN Incremental PR analysis: Could not determine common base path, cache will not be computed. Consider setting 'sonar.projectBaseDir' property.
16:16:59.379 INFO Sensor C# File Caching Sensor [csharp] (done) | time=6ms
16:16:59.379 INFO Sensor Zero Coverage Sensor
16:16:59.389 INFO Sensor Zero Coverage Sensor (done) | time=10ms
16:16:59.389 INFO SCM Publisher SCM provider for this project is: git
16:16:59.389 INFO SCM Publisher 4 source files to be analyzed
16:16:59.388 INFO SCM Publisher 4/4 source files have been analyzed (done) | time=449ms
16:16:59.846 INFO CPD Executor Calculating CPD for 0 files
16:16:59.846 INFO CPD Executor CPD calculation finished (done) | time=0ms
16:16:59.854 INFO SCM revision ID 'f2bc042c04c6e72427c380bcaee6d6fee7b49adf'
16:17:00.121 INFO Analysis report generated in 120ms, dir size=201.1 kB
16:17:00.195 INFO Analysis report compressed in 57ms, zip size=22.4 kB
16:17:00.393 INFO Analysis report uploaded in 195ms
16:17:00.394 INFO ANALYSIS SUCCESSFUL, you can find the results at: http://127.0.0.1:9000/dashboard?id=sonarqube
16:17:00.395 INFO Note that you will be able to access the updated dashboard once the server has processed the submitted analysis report
16:17:00.395 INFO More about the report processing at http://127.0.0.1:9000/api/ce/task?id=acd819f5-9e70-42ab-bff7-3cc893e2cae4
16:17:00.405 INFO Analysis total time: 18.743 s
16:17:00.408 INFO SonarScanner Engine completed successfully
16:17:00.494 INFO EXECUTION SUCCESS
16:17:00.494 INFO Total time: 21.288s
Finished: SUCCESS
```

14. Once the build is complete, check the project in SonarQube.

The screenshot shows the SonarQube dashboard for the 'main' project. At the top, there's a green 'Passed' status with a small warning icon and the text 'The last analysis has warnings. See details'. Below this, there are two tabs: 'New Code' and 'Overall Code', with 'Overall Code' being the active tab. The dashboard is divided into several sections: Security (0 Open issues), Reliability (0 Open issues), Maintainability (0 Open issues), Accepted issues (0), Coverage (0.0%), and Duplications (0.0%). Each section includes a progress bar and some descriptive text. The overall status is 'Passed'.

In this way, we have integrated Jenkins with SonarQube for SAST.

Conclusion:

This experiment effectively showcased the integration of Jenkins with SonarQube for conducting static code analysis. Through the configuration of both tools, we were able to assess a sample project, gaining valuable insights into the SAST process and the role of Jenkins in automating the identification of code vulnerabilities.

Experiment No. 08

Aim: To establish a Jenkins CI/CD Pipeline integrated with SonarQube/GitLab for conducting static analysis on a sample web application (Java/Python) to identify bugs, code smells, and security vulnerabilities.

Theory:

Understanding SAST:

Static application security testing (SAST), or static analysis, is a testing methodology that analyzes source code to find security vulnerabilities that make your organization's applications susceptible to attack. SAST scans an application before the code is compiled. It's also known as white box testing.

Challenges Addressed by SAST:

SAST takes place very early in the software development life cycle (SDLC) as it does not require a working application and can take place without code being executed. It helps developers identify vulnerabilities in the initial stages of development and quickly resolve issues without breaking builds or passing on vulnerabilities to the final release of the application.

SAST tools give developers real-time feedback as they code, helping them fix issues before they pass the code to the next phase of the SDLC. This prevents security-related issues from being considered an afterthought. SAST tools also provide graphical representations of the issues found, from source to sink. These help you navigate the code easier. Some tools point out the exact location of vulnerabilities and highlight the risky code. Tools can also provide in-depth guidance on how to fix issues and the best place in the code to fix them, without requiring deep security domain expertise.

It's important to note that SAST tools must be run on the application on a regular basis, such as during daily/monthly builds, every time code is checked in, or during a code release.

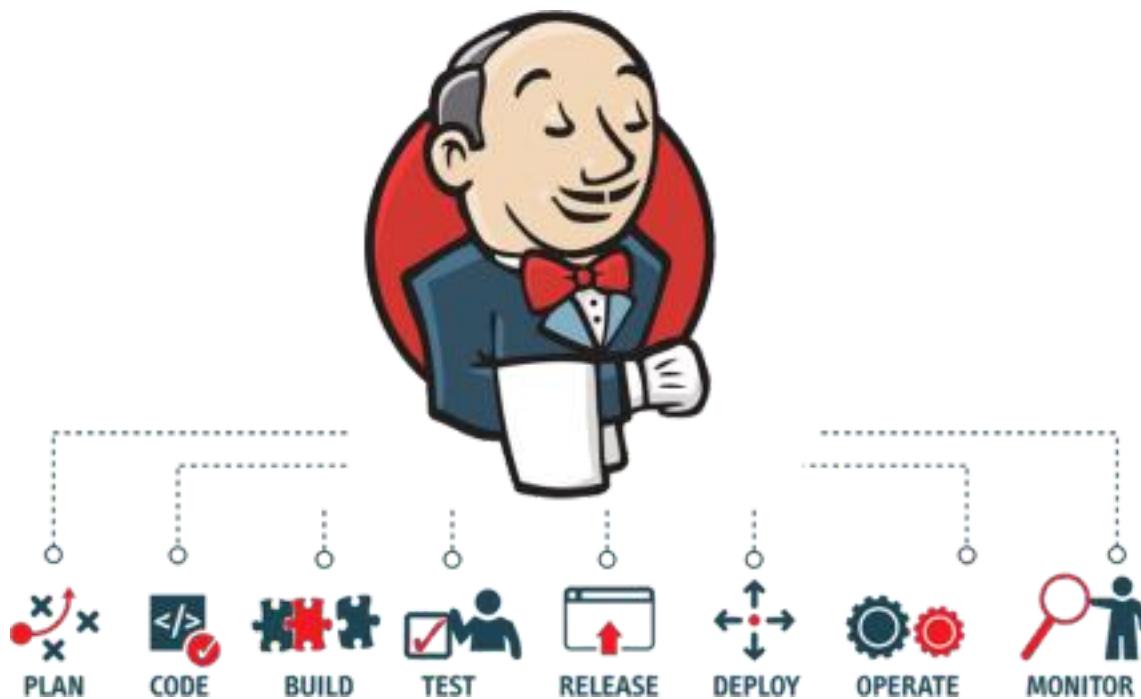
Importance of SAST:

Developers dramatically outnumber security staff. It can be challenging for an organization to find the resources to perform code reviews on even a fraction of its applications. A key strength of SAST tools is the ability to analyze 100% of the codebase. Additionally, they are much faster than manual secure code reviews performed by humans. These tools can scan millions of lines of code in a matter of minutes. SAST tools automatically identify critical vulnerabilities—such as buffer overflows, SQL injection, cross-site scripting, and others—with high confidence.

What is a CI/CD Pipeline?

CI/CD pipeline refers to the Continuous Integration/Continuous Delivery pipeline. Before we dive deep into this segment, let's first understand what is meant by the term 'pipeline'?

A pipeline is a concept that introduces a series of events or tasks that are connected in a sequence to make quick software releases. For example, there is a task, that task has got five different stages, and each stage has got some steps. All the steps in phase one have to be completed, to mark the latter stage to be complete.



Now, consider the CI/CD pipeline as the backbone of the DevOps approach. This Pipeline is responsible for building codes, running tests, and deploying new software versions. The CI/CD pipeline serves as the foundation of the DevOps methodology, managing code builds, tests, and deployment of new software versions in a structured manner.

Introduction to SonarQube:

SonarQube is an open-source platform developed by SonarSource for continuous inspection of code quality. Sonar does static code analysis, which provides a detailed report of bugs, code smells, vulnerabilities, code duplications.

It supports 25+ major programming languages through built-in rulesets and can also be extended with various plugins.

Advantages of SonarQube

- **Sustainability** - Reduces complexity, possible vulnerabilities, and code duplications, optimizing the life of applications.
- **Productivity boost** - Reduces the scale, cost of maintenance, and risk of the application; as such, it removes the need to spend more time changing the code
- **Quality Assurance** - Code quality control is an inseparable part of the process of software development.
- **Error Detection** - Detects errors in the code and alerts developers to fix them automatically before submitting them for output.
- **Increase consistency** - Determines where the code criteria are breached and enhances the quality
- **Scalability** - No restriction on the number of projects to be evaluated
- **Enhance developer skills** - Regular feedback on quality problems helps developers to improve their coding skills

Integrating Jenkins with SonarQube:

Prerequisites:

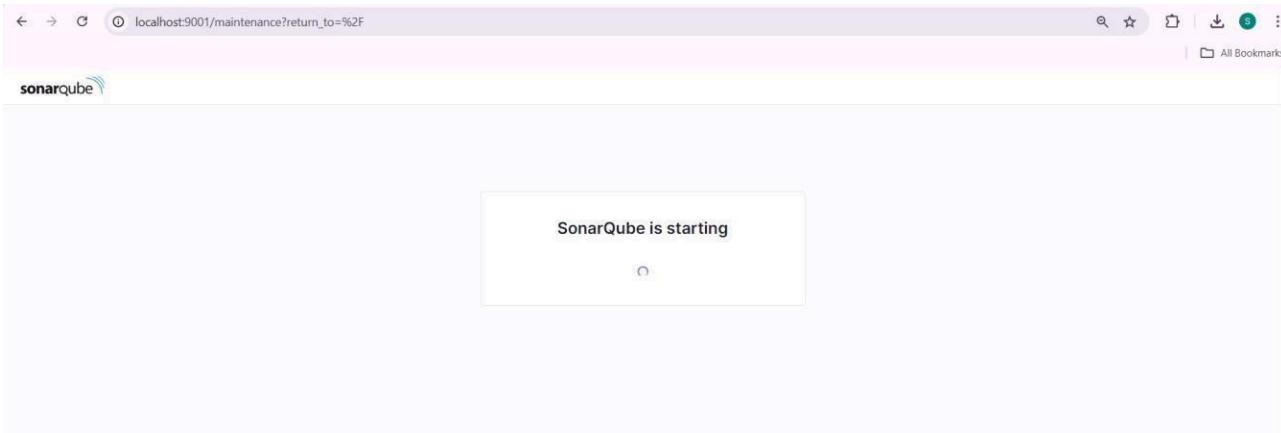
- Jenkins installed
- Docker Installed (for SonarQube)
- SonarQube Docker Image

Steps to Create a Jenkins CI/CD Pipeline and Utilize SonarQube for SAST:

1. Open up Jenkins Dashboard on localhost, port 8080 or whichever port it is at for you.
2. Run SonarQube in a Docker container using this command –

```
C:\Users\ADMIN>docker run -d --name sonarqube2 -e SONAR_ES_BOOTSTRAP_CHECKS_DISABLE=true -p 9001:9000 sonarqube:latest  
fda86b00e3989f3eb5aca8396b29b2a0adc95bcfe0fc5d85cf1237491e7678b9
```

3. Once the container is up and running, you can check the status of SonarQube at localhost port 9000.



4. Login to SonarQube using username *admin* and password *admin*.
5. Create a manual project in SonarQube with the name **sonarqube-test**

1 of 2

Create a local project

Project display name *

sonarqube-test

Project key *

sonarqube-test

Main branch name *

main

The name of your project's default branch [Learn More](#)

[Cancel](#)

[Next](#)

Setup the project and come back to Jenkins Dashboard.

6. Create a New Item in Jenkins, choose **Pipeline**.

New Item

Enter an item name

SonarQube-8

Select an item type



Freestyle project

Classic, general-purpose job type that checks out from up to one SCM, executes build steps serially, followed by post-build steps like archiving artifacts and sending email notifications.



Maven project

Build a maven project. Jenkins takes advantage of your POM files and drastically reduces the configuration.



Pipeline

Orchestrates long-running activities that can span multiple build agents. Suitable for building pipelines (formerly known as workflows) and/or organizing complex activities that do not easily fit in free-style job type.



Multi-configuration project

Suitable for projects that need a large number of different configurations, such as testing on multiple environments, platform-specific builds, etc.



[Folder](#)

7. Input the following script under Pipeline Script:

```
node {  
    stage('Cloning the GitHub Repo') {  
        git 'https://github.com/shazforiot/GOL.git'  
    } stage('SonarQube  
analysis') {  
        withSonarQubeEnv('sonarqube') {  
            sh "<PATH_TO SONARQUBE FOLDER>/bin//sonar-scanner \  
-D sonar.login=<SonarQube_USERNAME> \  
-D sonar.password=<SonarQube_PASSWORD> \  
-D sonar.projectKey=<Project_KEY> \  
-D sonar.exclusions=vendor/**,resources/**,**/*.java \  
-D sonar.host.url=http://127.0.0.1:9000/"  
        }  
    }  
}
```

Pipeline

Definition

Pipeline script

Script ?

```
2 * stage('Cloning the GitHub Repo') {  
3     git 'https://github.com/shazforiot/GOL.git'  
4 }  
5  
6 * stage('SonarQube analysis') {  
7     withSonarQubeEnv('sonarqube') {  
8         bat """  
C:\\ProgramData\\Jenkins\\.jenkins\\tools\\hudson.plugins.sonar.SonarRunnerInstallation\\sonarqube\\bin\\sonar-scanner ^  
-D sonar.login=admin ^  
-D sonar.password=admin123 ^  
-D sonar.projectKey=sonarqube-test ^  
-D sonar.exclusions=vendor/**,resources/**,**/*.java ^  
-D sonar.host.url=http://127.0.0.1:9001/  
"""  
18 }  
19 }
```

It is a java sample project which has a lot of repetitions and issues that will be detected by SonarQube.

8. Execute The Build.

9. Review the console output upon build completion.

SonarQube-8

Stage View

Cloning the GitHub Repo	SonarQube analysis
5s	31min 25s
1s	12min 7s
8s	50min 43s aborted
#4 Sep 26 18:04 No Changes	
#3 Sep 26 17:13 No Changes	
#2 Sep 26 17:11 No Changes	
#1 Sep 26 17:11 No Changes	

Build History

- #4 Sep 26, 2024, 6:04 PM
- #3 Sep 26, 2024, 5:13 PM

Console Output

```

Skipping 4,248 KB... Full Log
18:13:34.790 WARN Too many duplication references on file gameoflife-web/tools/jmeter/docs/api/org/apache/jmeter/protocol/http/gui/HTTPArgumentsPanel.html for block at line 798. Keep only the first 100 references.
18:13:34.790 WARN Too many duplication references on file gameoflife-web/tools/jmeter/docs/api/org/apache/jmeter/protocol/http/gui/HTTPArgumentsPanel.html for block at line 810. Keep only the first 100 references.
18:13:34.790 WARN Too many duplication references on file gameoflife-web/tools/jmeter/docs/api/org/apache/jmeter/protocol/http/gui/HTTPArgumentsPanel.html for block at line 823. Keep only the first 100 references.
18:13:34.790 WARN Too many duplication references on file gameoflife-web/tools/jmeter/docs/api/org/apache/jmeter/protocol/http/gui/HTTPArgumentsPanel.html for block at line 844. Keep only the first 100 references.
18:13:34.790 WARN Too many duplication references on file gameoflife-web/tools/jmeter/docs/api/org/apache/jmeter/protocol/http/gui/HTTPArgumentsPanel.html for block at line 509. Keep only the first 100 references.
18:13:34.790 WARN Too many duplication references on file gameoflife-web/tools/jmeter/docs/api/org/apache/jmeter/protocol/http/gui/HTTPArgumentsPanel.html for block at line 1065. Keep only the first 100 references.
18:13:34.790 WARN Too many duplication references on file gameoflife-web/tools/jmeter/docs/api/org/apache/jmeter/protocol/http/gui/HTTPArgumentsPanel.html for block at line 776. Keep only the first 100 references.
18:13:34.790 WARN Too many duplication references on file gameoflife-web/tools/jmeter/docs/api/org/apache/jmeter/protocol/http/gui/HTTPArgumentsPanel.html for block at line 778. Keep only the first 100 references.
18:13:34.790 WARN Too many duplication references on file gameoflife-web/tools/jmeter/docs/api/org/apache/jmeter/protocol/http/gui/HTTPArgumentsPanel.html for block at line 530. Keep only the first 100 references.
18:13:34.790 WARN Too many duplication references on file gameoflife-web/tools/jmeter/docs/api/org/apache/jmeter/protocol/http/gui/HTTPArgumentsPanel.html for block at line 648. Keep only the first 100 references.
18:13:34.790 WARN Too many duplication references on file gameoflife-web/tools/jmeter/docs/api/org/apache/jmeter/protocol/http/gui/HTTPArgumentsPanel.html for block at line 798. Keep only the first 100 references.
18:13:34.790 WARN Too many duplication references on file gameoflife-web/tools/jmeter/docs/api/org/apache/jmeter/protocol/http/gui/HTTPArgumentsPanel.html for block at line 546. Keep only the first 100 references.
18:13:34.790 WARN Too many duplication references on file gameoflife-web/tools/jmeter/docs/api/org/apache/jmeter/protocol/http/gui/HTTPArgumentsPanel.html for block at line 546. Keep only the first 100 references.
18:13:34.790 WARN Too many duplication references on file gameoflife-web/tools/jmeter/docs/api/org/apache/jmeter/protocol/http/gui/HTTPArgumentsPanel.html for block at line 546. Keep only the first 100 references.
18:13:34.790 WARN Too many duplication references on file gameoflife-web/tools/jmeter/docs/api/org/apache/jmeter/protocol/http/gui/HTTPArgumentsPanel.html for block at line 634. Keep only the first 100 references.
18:13:34.790 WARN Too many duplication references on file gameoflife-web/tools/jmeter/docs/api/org/apache/jmeter/protocol/http/gui/HTTPArgumentsPanel.html for block at line 778. Keep only the first 100 references.

references.
18:13:39.657 WARN Too many duplication references on file gameoflife-web/tools/jmeter/docs/api/org/apache/jmeter/gui/util/TextAreaCellRenderer.html for block at line 75. Keep only the first 100 references.
18:13:39.657 WARN Too many duplication references on file gameoflife-web/tools/jmeter/docs/api/org/apache/jmeter/gui/util/TextAreaCellRenderer.html for block at line 41. Keep only the first 100 references.
18:13:39.657 WARN Too many duplication references on file gameoflife-web/tools/jmeter/docs/api/org/apache/jmeter/gui/util/TextAreaCellRenderer.html for block at line 17. Keep only the first 100 references.
18:13:39.657 WARN Too many duplication references on file gameoflife-web/tools/jmeter/docs/api/org/apache/jmeter/gui/util/TextAreaCellRenderer.html for block at line 296. Keep only the first 100 references.
18:13:39.657 WARN Too many duplication references on file gameoflife-web/tools/jmeter/docs/api/org/apache/jmeter/gui/util/TextAreaCellRenderer.html for block at line 75. Keep only the first 100 references.
18:13:39.657 INFO CPD Executor CPD calculation finished (done) | time=15897ms
18:13:39.674 INFO SCM revision ID 'ba799ba7eb576f04a4612322b0412c5e6e1e5e4'
18:15:49.696 INFO Analysis report generated in 5022ms, dir size=127.2 MB
18:16:08.759 INFO Analysis report compressed in 19048ms, zip size=29.6 MB
18:16:09.884 INFO Analysis report uploaded in 1129ms
18:16:09.887 INFO ANALYSIS SUCCESSFUL, you can find the results at: http://127.0.0.1:9001/dashboard?id=sonarqube-test
18:16:09.887 INFO Note that you will be able to access the updated dashboard once the server has processed the submitted analysis report
18:16:09.887 INFO More about the report processing at http://127.0.0.1:9001/api/ce/task?id=6f22c333-3777-4a21-b058-0ab4c049625c
18:16:22.970 INFO Analysis total time: 12:02.242 s
18:16:22.975 INFO SonarScanner Engine completed successfully
18:16:23.699 INFO EXECUTION SUCCESS
18:16:23.706 INFO Total time: 12:05.758s
[Pipeline] }
[Pipeline] // withSonarQubeEnv
[Pipeline] }
[Pipeline] // stage
[Pipeline] }
[Pipeline] // node
[Pipeline] End of Pipeline
Finished: SUCCESS

```

10. After that, check the project in SonarQube.

The screenshot shows the SonarQube main dashboard for the 'main' branch of the 'sonarqube-test' project. The overall status is 'Passed'. Key metrics displayed include:

- Security:** 0 Open issues (A grade)
- Reliability:** 68K Open issues (C grade)
- Maintainability:** 164k Open issues (A grade)
- Accepted issues:** 0
- Coverage:** On 0 lines to cover.
- Duplications:** 50.6% (On 750k lines)
- Security Hotspots:** 3 (E grade)

Other tabs visible in the header include Projects, Issues, Rules, Quality Profiles, Quality Gates, Administration, More, and a search bar.

Under different tabs, check all different issues with the code.

11. Categories of Code Problems:

The screenshot shows the SonarQube 'Issues' tab for the 'main' branch of the 'sonarqube-test' project. The left sidebar includes filters for Software Quality, Severity, Type (Bug selected), and Scope. The main area displays three specific code problems:

- Add "lang" and/or "xml:lang" attributes to this "<html>" element.** (Intentionality: accessibility wcag2-a) - Reliability: C, Status: Open, Not assigned.
- Insert a <!DOCTYPE> declaration to before this <html> tag.** (Consistency: user-experience) - Reliability: C, Status: Open, Not assigned.
- Add "<th>" headers to this "<table>".** (Intentionality: accessibility wcag2-a) - Reliability: C, Status: Open, Not assigned.

A yellow warning box at the bottom left 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."

Code Smells

The screenshot shows the SonarQube Issues page for the project 'sonarqube-test'. The left sidebar has 'Issues' selected. The main area displays a list of code smells under the 'gameoflife-acceptance-tests/Dockerfile' file. The first item is 'Use a specific version tag for the image.' with an 'Intentionality' status of 'No tags'. Below it are three more items, each with a 'Maintainability' status of 'Open' and 'Not assigned'. A warning banner 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.'

Intentional Issues

The screenshot shows the SonarQube Issues page for the project 'sonarqube-test'. The left sidebar has 'Issues' selected. The main area displays a list of intentional issues under the 'gameoflife-acceptance-tests/Dockerfile' file. The first item is 'Use a specific version tag for the image.' with an 'Intentionality' status of 'No tags'. Below it are three more items, each with a 'Maintainability' status of 'Open' and 'Not assigned'. A warning banner 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.'

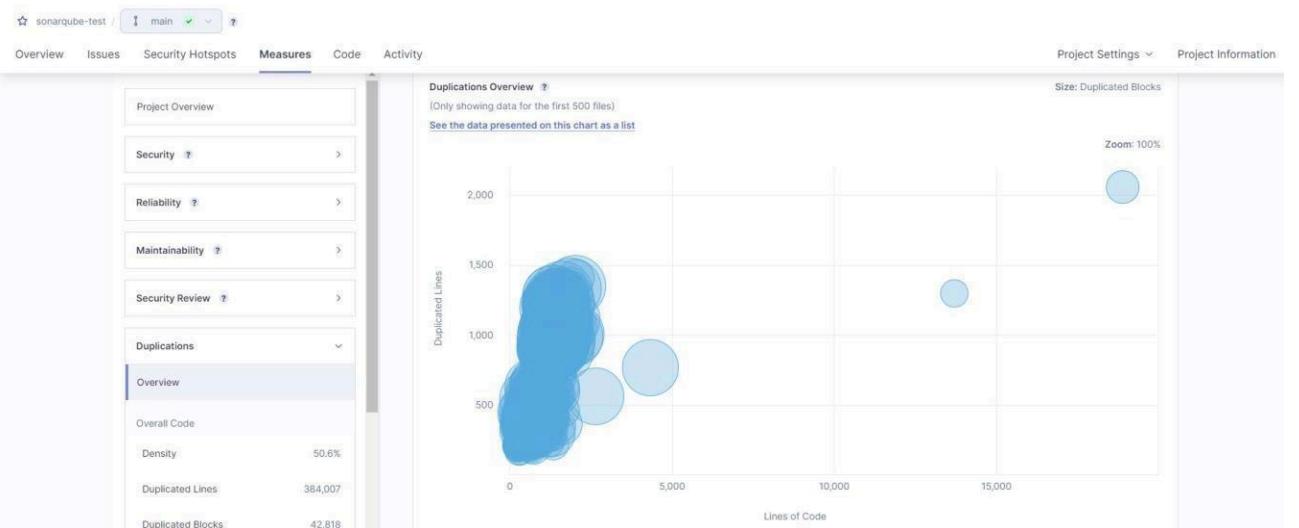
Reliability Issue

The screenshot shows the SonarQube Issues page for the project 'sonarqube-test'. The left sidebar has 'Issues' selected. The main area displays a list of reliability issues under the 'gameoflife-core/build/reports/tests/all-tests.html' file. The first item is 'Anchors must have content and the content must be accessible by a screen reader.' with a 'Consistency' status of '20,856 issues' and an 'Accessibility' status of '217d effort'. Below it are four more items, each with a 'Maintainability' and 'Reliability' status of 'Open' and 'Not assigned'. A warning banner at the bottom states: 'Embedded database should be used for evaluation purposes only.'

Maintainability Issue

The screenshot shows the SonarQube interface for the project 'sonarqube-test'. The 'Issues' tab is selected. On the left, there's a sidebar with filters for 'Clean Code Attribute' (Consistency: 164k, Intentionality: 15, Adaptability: 0, Responsibility: 0) and 'Software Quality' (Security: 0, Reliability: 21k, Maintainability: 164k). The main area displays a list of issues under the file 'gameoflife-core/build/reports/tests/all-tests.html'. The first issue is 'Remove this deprecated "width" attribute.' It has a 'Consistency' priority, is labeled 'html5 obsolete', and is marked as a 'Major' code smell. It was created 4 years ago with 5min effort. There are three more similar issues listed below it.

Duplicates



In this way, we have created a CI/CD Pipeline with Jenkins and integrated it with SonarQube to find issues in the code like bugs, code smells, duplicates, cyclomatic complexities, etc.

Conclusion:

I successfully established a CI/CD pipeline using Jenkins integrated with SonarQube for static code analysis of a sample Java application. I configured SonarQube within a Docker container and set up Jenkins to clone the GitHub repository for analysis. The pipeline effectively identified various issues, including bugs and security vulnerabilities, which I reviewed in SonarQube. This experience improved my skills in CI/CD tool configuration and underscored the significance of automated code quality maintenance. Overall, I gained valuable insights into the integration of tools for effective software development practices.

Aim: To Understand Continuous monitoring and Installation and configuration of Nagios Core, Nagios Plugins and NRPE (Nagios Remote Plugin Executor) on Linux Machine.

Theory:

What is Nagios?

Nagios is an open-source software for continuous monitoring of systems, networks, and infrastructures. It runs plugins stored on a server that is connected with a host or another server on your network or the Internet. In case of any failure, Nagios alerts about the issues so that the technical team can perform the recovery process immediately.

Nagios is used for continuous monitoring of systems, applications, service and business processes in a DevOps culture.

Why We Need Nagios tool?

Here are the important reasons to use Nagios monitoring tool:

- Detects all types of network or server issues
- Helps you to find the root cause of the problem which allows you to get the permanent solution to the problem
- Active monitoring of your entire infrastructure and business processes
- Allows you to monitor and troubleshoot server performance issues
- Helps you to plan for infrastructure upgrades before outdated systems create failures
- You can maintain the security and availability of the service
- Automatically fix problems in a panic situation

Features of Nagios

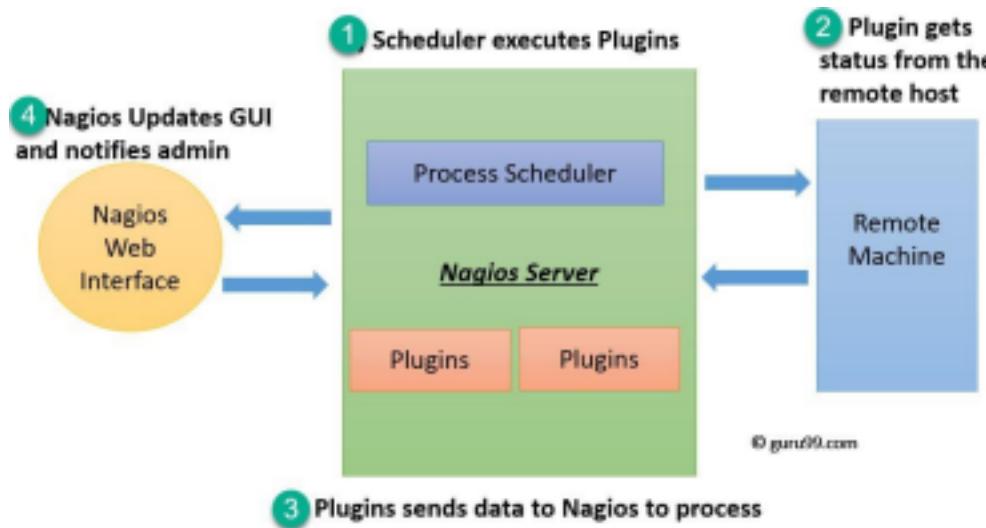
Following are the important features of Nagios monitoring tool:

- Relatively scalable, Manageable, and Secure
- Good log and database system
- Informative and attractive web interfaces
- Automatically send alerts if condition changes
- If the services are running fine, then there is no need to do check that host is alive
- Helps you to detect network errors or server crashes
- You can troubleshoot the performance issues of the server.
- The issues, if any, can be fixed automatically as they are identified during the monitoring process
- You can monitor the entire business process and IT infrastructure with a single pass
- The product's architecture is easy to write new plugins in the language of your choice
- Nagios allows you to read its configuration from an entire directory which helps you to decide how to define individual files

- Utilizes topology to determine dependencies
- Monitor network services like HTTP, SMTP, HTTP, SNMP, FTP, SSH, POP, etc.
- Helps you to define network host hierarchy using parent hosts
- Ability to define event handlers that runs during service or host events for proactive problem resolution
- Support for implementing redundant monitoring hosts

Nagios Architecture

Nagios is a client-server architecture. Usually, on a network, a Nagios server is running on a host, and plugins are running on all the remote hosts which should be monitored.



1. The scheduler is a component of the server part of Nagios. It sends a signal to execute the plugins at the remote host.
2. The plugin gets the status from the remote host
3. The plugin sends the data to the process scheduler
4. The process scheduler updates the GUI and notifications are sent to admins.

Installation of Nagios

Prerequisites: AWS Free Tier

Steps:

1. Create an Amazon Linux EC2 Instance in AWS and name it - nagios-host

The screenshot shows the AWS EC2 Instances page. On the left, there's a sidebar with links like EC2 Dashboard, EC2 Global View, Events, Console-to-Code Preview, Instances, and Instances. The main area is titled 'Instances (1) Info' and shows a single instance named 'nagios-host' with the ID 'i-028182fbe9c070820'. The instance is listed as 'Running' with a status check of 'Initializing'. There are buttons for 'Launch instances', 'Actions', and 'All states'.

2. Under Security Group, make sure HTTP, HTTPS, SSH, ICMP are open from everywhere.
You have to edit the inbound rules of the specified Security Group for this.

Security group rule ID	Type	Protocol	Port range	Source	Description - optional
sgr-04093d1a208295e38	SSH	TCP	22	C...	
-	HTTP	TCP	80	A...	
-	HTTPS	TCP	443	A...	
-	All ICMP - IPv4	ICMP	All	A...	

Add rule

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

Cancel **Preview changes** **Save rules**

3. SSH into Your EC2 instance or simply use EC2 Instance Connect from the browser.



4. Update the package indices and install the following packages using yum

```
sudo yum update
sudo yum install httpd php
sudo yum install gcc glibc glibc-common
sudo yum install gd gd-devel
yum install: error: unrecognized arguments: -Oy
[ec2-user@ip-172-31-38-150 ~]$ sudo yum install gd gd-devel -y
Last metadata expiration check: 0:05:12 ago on Sun Oct 6 11:15:04 2024.
Dependencies resolved.
```

Package	Architecture	Version	Repository	Size
Installing:				
gd	x86_64	2.3.3-5.amzn2023.0.3	amazonlinux	139
gd-devel	x86_64	2.3.3-5.amzn2023.0.3	amazonlinux	38
Installing dependencies:				
brotli	x86_64	1.0.9-4.amzn2023.0.2	amazonlinux	314
brotli-devel	x86_64	1.0.9-4.amzn2023.0.2	amazonlinux	31
bzip2-devel	x86_64	1.0.8-6.amzn2023.0.2	amazonlinux	214
cairo	x86_64	1.17.6-2.amzn2023.0.1	amazonlinux	684
cmake-filesystem	x86_64	3.22.2-1.amzn2023.0.4	amazonlinux	16
fontconfig	x86_64	2.13.94-2.amzn2023.0.2	amazonlinux	273
fontconfig-devel	x86_64	2.13.94-2.amzn2023.0.2	amazonlinux	128
fnts-filesystem	noarch	1:2.0.5-12.amzn2023.0.2	amazonlinux	9.5
freetype	x86_64	2.13.2-5.amzn2023.0.1	amazonlinux	423
freetype-devel	x86_64	2.13.2-5.amzn2023.0.1	amazonlinux	912
glib2-devel	x86_64	2.74.7-689.amzn2023.0.2	amazonlinux	486
google-noto-fonts-common	noarch	20201206-2.amzn2023.0.2	amazonlinux	15
google-noto-sans-vf-fonts	noarch	20201206-2.amzn2023.0.2	amazonlinux	492
graphite2	x86_64	1.3.14-7.amzn2023.0.2	amazonlinux	97
graphite2-devel	x86_64	1.3.14-7.amzn2023.0.2	amazonlinux	21
harfbuzz	x86_64	7.0.0-2.amzn2023.0.1	amazonlinux	868
harfbuzz-devel	x86_64	7.0.0-2.amzn2023.0.1	amazonlinux	404
harfbuzz-icu	x86_64	7.0.0-2.amzn2023.0.1	amazonlinux	18
jbigkit-libs	x86_64	2.1-21.amzn2023.0.2	amazonlinux	54
langpacks-core-font-en	noarch	3.0-21.amzn2023.0.4	amazonlinux	10
libICE	x86_64	1.0.10-6.amzn2023.0.2	amazonlinux	71
libSM	x86_64	1.2.3-8.amzn2023.0.2	amazonlinux	42

5. Create a new Nagios User with its password. You'll have to enter the password twice for confirmation.

```
sudo adduser -m nagios
sudo passwd nagios
```

```
Complete!
[ec2-user@ip-172-31-38-150 ~]$ sudo adduser -m nagios
[ec2-user@ip-172-31-38-150 ~]$ sudo passwd nagios
Changing password for user nagios.
New password:
Retype new password:
passwd: all authentication tokens updated successfully.
[ec2-user@ip-172-31-38-150 ~]$
```

6. Create a new user group

```
sudo groupadd nagcmd
```

7. Use these commands so that you don't have to use sudo for Apache and Nagios

```
sudo usermod -a -G nagcmd nagios
sudo usermod -a -G nagcmd apache
```

8. Create a new directory for Nagios downloads

```
mkdir ~/downloads
cd ~/downloads
passwd: all authentication tokens updated successfully.
[ec2-user@ip-172-31-38-150 ~]$ sudo groupadd nagcmd
[ec2-user@ip-172-31-38-150 ~]$ sudo usermod -a -G nagcmd nagios
[ec2-user@ip-172-31-38-150 ~]$ sudo usermod -a -G nagcmd apache
[ec2-user@ip-172-31-38-150 ~]$ mkdir ~/downloads
[ec2-user@ip-172-31-38-150 ~]$ cd ~/downloads
[ec2-user@ip-172-31-38-150 downloads]$ wget https://go.nagios.org/24-09-17/6kqcx
```

9. Use wget to download the source zip files.

```
wget https://go.nagios.org/1/975333/2024-09-17/6kqcx
wget https://nagios-plugins.org/download/nagios-plugins-2.4.11.tar.gz
```

```
[ec2-user@ip-172-31-38-150 downloads]$ wget https://go.nagios.org/24-09-17/6kqcx
--2024-10-06 11:23:50-- https://go.nagios.org/1/975333/2024-09-17/6kqcx
Resolving go.nagios.org (go.nagios.org) ... 3.92.127.219, ...
Connecting to go.nagios.org (go.nagios.org) |3.92.1| HTTP request sent, awaiting response... 302 Found
Location: http://assets.nagios.com/downloads/nagios.tar.gz?utm_source=Nagios.org&utm_content=Download+5+Download+&pi_content=1e9662c93afb2ed6bd2e3f3cc3
```

10. Use tar to unzip and change to that directory.

```
tar zxvf nagios-4.5.5.tar.gz
cd nagios-4.5.5
--2024-10-06 11:23:50-- https://go.nagios.org/1/975333/2024-09-17/6kqcx
Resolving go.nagios.org (go.nagios.org) ... 3.92.120.28, 52.54.96.194, 3.21572.219, ...
Connecting to go.nagios.org (go.nagios.org) |3.92.120.28|:443... connected.
HTTP request sent, awaiting response... 302 Found
Location: http://assets.nagios.com/downloads/nagioscore/releases/nagios-4.5.tar.gz?utm_source=Nagios.org&utm_content=Download+Form&utm_campaign=Core+4.5+Download+&pi_content=1e9662c93afb2ed6bd2e3f3cc38771a7f01125e969f2a75b0e24439d4a81d8 [following]
--2024-10-06 11:23:50-- http://assets.nagios.com/downloads/nagioscore/releases/nagios-4.5.5.tar.gz?utm_source=Nagios.org&utm_content=Download+Form&utm_campaign=Core+4.5.5+Download+&pi_content=1e9662c93afb2ed6bd2e3f3cc38771a7f011e969f2a75b0e2254439d4a81d8
Resolving assets.nagios.com (assets.nagios.com) ... 45.79.49.120, 2600:3c00:03c:92ff:feff:45ce
Connecting to assets.nagios.com (assets.nagios.com) |45.79.49.120|:80... con
```

11. Run the configuration script with the same group name you previously created.

```
./configure --with-command-group=nagcmd
```

12. Compile the source code.

```
make all
```

The screenshot shows a Windows Command Prompt window titled "Command Prompt". It displays the configuration summary for Nagios 4.5.5, followed by the compilation process. The configuration summary includes details like Nagios executable, user/group, command broker, and web interface URLs. The compilation process shows the execution of "make all" in the "/home/ec2-user/downloads/nagios-4.5.5/base" directory, with gcc commands for various source files like nagios.c, broker.c, nebmod.c, common/shared.c, query-handler.c, workers.c, checks.c, config.c, commands.c, and events.c. A warning message is visible regarding a null pointer argument to the log_debug_info function.

```
*** Configuration summary for nagios 4.5.5 2024-09-17 ***

General Options:
-----
    Nagios executable: nagios
    Nagios user/group: nagios,nagios
    Command user/group: nagios,nagcmd
        Event Broker: yes
        Install ${prefix}: /usr/local/nagios
    Install ${includedir}: /usr/local/nagios/include/nagios
        Lock file: /run/nagios.lock
    Check result directory: /usr/local/nagios/var/spool/checkresults
        Init directory: /lib/systemd/system
    Apache conf.d directory: /etc/httpd/conf.d
        Mail program: /bin/mail
    Host OS: linux-gnu
    IOBroker Method: epoll

Web Interface Options:
-----
    HTML URL: http://localhost/nagios/
    CGI URL: http://localhost/nagios/cgi-bin/
Traceroute (used by WAP): /usr/bin/traceroute

Review the options above for accuracy. If they look okay,
type 'make all' to compile the main program and CGIs.

[ec2-user@ip-172-31-80-195 nagios-4.5.5]$ make all
cd ./base && make
make[1]: Entering directory '/home/ec2-user/downloads/nagios-4.5.5/base'
gcc -Wall -I.. -I. -I..../lib -I..../include -I..../include -I.. -g -O2 -DHAVE_CONFIG_H -DNSCORE -c -o nagios.o ./nagios.c
gcc -Wall -I.. -I. -I..../lib -I..../include -I..../include -I.. -g -O2 -DHAVE_CONFIG_H -DNSCORE -c -o broker.o broker.c
gcc -Wall -I.. -I. -I..../lib -I..../include -I..../include -I.. -g -O2 -DHAVE_CONFIG_H -DNSCORE -c -o nebmod.o nebmod.c
gcc -Wall -I.. -I. -I..../lib -I..../include -I..../include -I.. -g -O2 -DHAVE_CONFIG_H -DNSCORE -c -o ../common/shared.o ../common/shared.c
gcc -Wall -I.. -I. -I..../lib -I..../include -I..../include -I..../include -I.. -g -O2 -DHAVE_CONFIG_H -DNSCORE -c -o query-handler.o query-handler.c
gcc -Wall -I.. -I. -I..../lib -I..../include -I..../include -I.. -g -O2 -DHAVE_CONFIG_H -DNSCORE -c -o workers.o workers.c
In function 'get_wproc_list',
  inlined from 'get_worker' at workers.c:277:12:
workers.c:253:17: warning: '%s' directive argument is null [-Wformat-overflow]
  253 |         log_debug_info(DEBUGL_CHECKS, 1, "Found specialized worker(s) for '%s'", (slash && *slash != '/') ? slash : cmd_name);
  |
gcc -Wall -I.. -I. -I..../lib -I..../include -I..../include -I.. -g -O2 -DHAVE_CONFIG_H -DNSCORE -c -o checks.o checks.c
gcc -Wall -I.. -I. -I..../lib -I..../include -I..../include -I.. -g -O2 -DHAVE_CONFIG_H -DNSCORE -c -o config.o config.c
gcc -Wall -I.. -I. -I..../lib -I..../include -I..../include -I.. -g -O2 -DHAVE_CONFIG_H -DNSCORE -c -o commands.o commands.c
gcc -Wall -I.. -I. -I..../lib -I..../include -I..../include -I.. -g -O2 -DHAVE_CONFIG_H -DNSCORE -c -o events.o events.c
```

13. Install binaries, init script and sample config files. Lastly, set permissions on the external command directory.

```
sudo make install
sudo make install-init
sudo make install-config
sudo make install-commandmode
```

```
[ec2-user@ip-172-31-38-150 nagios-4.4.6]$ sudo make install-webconf
/usr/bin/install -c -m 644 sample-config/httpd.conf /etc/httpd/conf.d/nagio
s.conf
if [ 0 -eq 1 ]; then \
    ln -s /etc/httpd/conf.d/nagios.conf /etc/apache2/sites-enabled/nagi
os.conf; \
fi

*** Nagios/Apache conf file installed ***

[ec2-user@ip-172-31-38-150 nagios-4.4.6]$
```

```
sudo nano /usr/local/nagios/etc/objects/contacts.cfg
=====
[ec2-user@ip-172-31-38-150 nagios-plugins-2.4.11]$ sudo yum install openssl
evel
Last metadata expiration check: 0:13:31 ago on Sun Oct  6 11:15:04 2024.
Dependencies resolved.
=====
=====
Package           Architecture      Version
Repository        Size
=====
Installing:
openssl-devel      x86_64          1:3.0.8-1.amzn2023.0.14
amazonlinux        3.0 M
=====
Transaction Summary
```

14. Configure the web interface.

```
sudo make install-webconf
```

```
[ec2-user@ip-172-31-38-150 nagios-4.4.6]$ sudo make install-webconf
/usr/bin/install -c -m 644 sample-config/httpd.conf /etc/httpd/conf.d/nagio
s.conf
if [ 0 -eq 1 ]; then \
    ln -s /etc/httpd/conf.d/nagios.conf /etc/apache2/sites-enabled/nagi
os.conf; \
fi

*** Nagios/Apache conf file installed ***

[ec2-user@ip-172-31-38-150 nagios-4.4.6]$
```

15. Create a nagiosadmin account for nagios login along with password. You'll have to specify the password twice.

```
sudo htpasswd -c /usr/local/nagios/etc/htpasswd.users nagiosadmin
```

```
[ec2-user@ip-172-31-38-150 nagios-4.4.6]$ sudo htpasswd -c /usr/local/nagios/etc/htpasswd.users nagiosadmin
[sudo] password for nagiosadmin:
[sudo] password for nagiosadmin:
New password:
Re-type new password:
Adding password for user nagiosadmin
[ec2-user@ip-172-31-38-150 nagios-4.4.6]$
```

16. Restart Apache

```
sudo service httpd restart
```

```
[ec2-user@ip-172-31-38-150 nagios-4.4.6]$ sudo service httpd restart
Redirecting to /bin/systemctl restart httpd.service
[ec2-user@ip-172-31-38-150 nagios-4.4.6]$
```

17. Go back to the downloads folder and unzip the plugins zip file.

```
cd ~/downloads
tar zxvf nagios-plugins-2.4.11.tar.gz
[ec2-user@ip-172-31-38-150 downloads]$ tar zxvf nagios-plugins-2.4.11.tar.gz
nagios-plugins-2.4.11/
nagios-plugins-2.4.11/build-aux/
nagios-plugins-2.4.11/build-aux/compile
nagios-plugins-2.4.11/build-aux/config.guess
nagios-plugins-2.4.11/build-aux/config.rpath
nagios-plugins-2.4.11/build-aux/config.sub
nagios-plugins-2.4.11/build-aux/install-sh
nagios-plugins-2.4.11/build-aux/ltdlmain.sh
nagios-plugins-2.4.11/build-aux/missing
nagios-plugins-2.4.11/build-aux/mkinstalldirs
```

18. Compile and install plugins

```
cd nagios-plugins-2.4.11
./configure --with-nagios-user=nagios --with-nagios-group=nagios
make
sudo make install
[ec2-user@ip-172-31-38-150 downloads]$ cd nagios-plugins-2.4.11
[ec2-user@ip-172-31-38-150 nagios-plugins-2.4.11]$ ./configure --with-nagios-user=nagios
--with-nagios-group=nagios
[ec2-user@ip-172-31-38-150 nagios-plugins-2.4.11]$ ./configure --with-nagios-user=nagios
--with-nagios-group=nagios
checking for a BSD-compatible install... /usr/bin/install -c
checking whether build environment is sane... yes
checking for a thread-safe mkdir -p... /usr/bin/mkdir -p
checking for gawk... gawk
checking whether make sets $(MAKE)... yes
checking whether make supports nested variables... yes
checking whether to enable maintainer-specific portions of Makefiles... yes
```

19. Start Nagios

Add Nagios to the list of system services

```
sudo chkconfig --add nagios
```

```
sudo chkconfig nagios on
```

```
[ec2-user@ip-172-31-80-195 nagios-plugins-2.4.11]$ sudo chkconfig --add nagios
sudo chkconfig nagios on
error reading information on service nagios: No such file or directory
Note: Forwarding request to 'systemctl enable nagios.service'.
Created symlink /etc/systemd/system/multi-user.target.wants/nagios.service → /usr/lib/systemd/system/nagios.service.
```

Verify the sample configuration files

```
sudo /usr/local/nagios/bin/nagios -v /usr/local/nagios/etc/nagios.cfg
```

```
[ec2-user@ip-172-31-38-150 nagios-plugins-2.4.11]$ sudo /usr/local/nagios/bin/nagios -v
/usr/local/nagios/etc/nagios.cfg
/usr/local/nagios/etc/nagios.cfg
```

```
Nagios Core 4.4.6
Copyright (c) 2009-present Nagios Core Development Team and Community Contributors
Copyright (c) 1999-2009 Ethan Galstad
Last Modified: 2020-04-28
License: GPL
```

```
Website: https://www.nagios.org
Reading configuration data...
  Read main config file okay...
  Read object config files okay...
```

```
Running pre-flight check on configuration data...
```

```
Checking objects...
  Checked 8 services.
  Checked 1 hosts.
  Checked 1 host groups.
  Checked 0 service groups.
  Checked 1 contacts.
  Checked 1 contact groups.
  Checked 24 commands.
  Checked 5 time periods.
```

If there are no errors, you can go ahead and start Nagios.

```
sudo service nagios start
```

```
Things look okay - No serious problems were detected during the pre-flight check
[ec2-user@ip-172-31-38-150 nagios-plugins-2.4.11]$ sudo service nagios start
Redirecting to /bin/systemctl start nagios.service
[ec2-user@ip-172-31-38-150 nagios-plugins-2.4.11]$
```

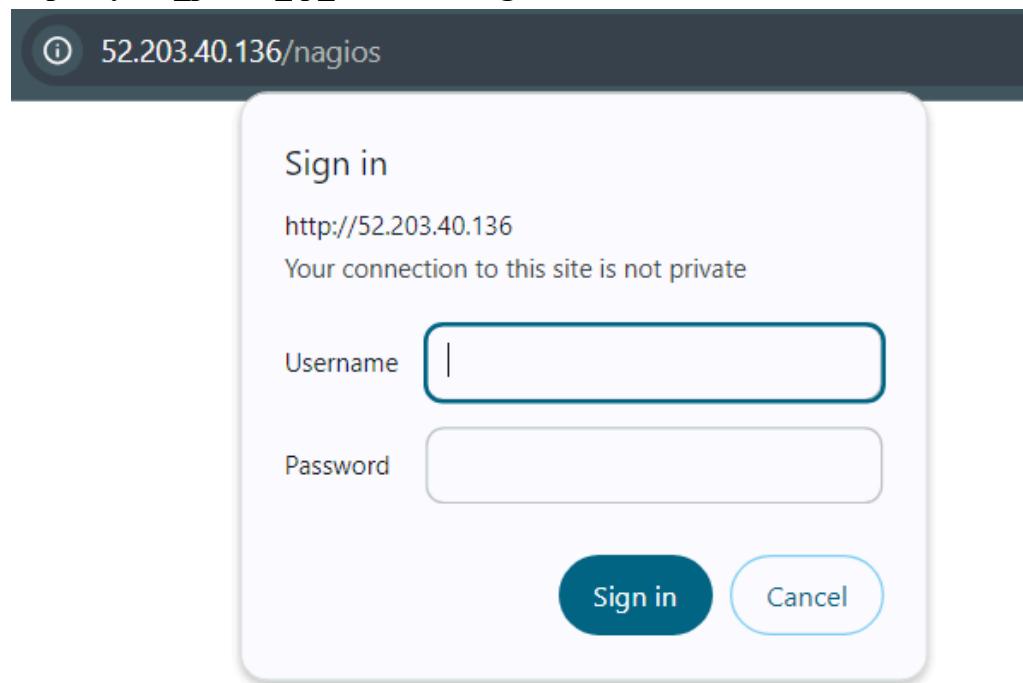
20. Check the status of Nagios

```
sudo systemctl status nagios
Redirecting to /bin/systemctl start nagios.service
[ec2-user@ip-172-31-38-150 nagios-plugins-2.4.11]$ sudo systemctl status nagios
● nagios.service - Nagios Core 4.4.6
    Loaded: loaded (/usr/lib/systemd/system/nagios.service; enabled; preset: disabled)
    Active: active (running) since Sun 2024-10-06 11:51:46 UTC; 1min 33s ago
      Docs: https://www.nagios.org/documentation
      Main PID: 89956 (nagios)
        Tasks: 6 (limit: 1112)
       Memory: 2.4M
          CPU: 36ms
        CGroup: /system.slice/nagios.service
                ├─89956 /usr/local/nagios/bin/nagios -d /usr/local/nagios/etc/nagios.cfg
                ├─89957 /usr/local/nagios/bin/nagios --worker /usr/local/nagios/var/rw/nagios.qh
                ├─89958 /usr/local/nagios/bin/nagios --worker /usr/local/nagios/var/rw/nagios.qh
                ├─89959 /usr/local/nagios/bin/nagios --worker /usr/local/nagios/var/rw/nagios.qh
                ├─89960 /usr/local/nagios/bin/nagios --worker /usr/local/nagios/var/rw/nagios.qh
                └─89961 /usr/local/nagios/bin/nagios -d /usr/local/nagios/etc/nagios.cfg

Oct 06 11:51:46 ip-172-31-38-150.ec2.internal nagios[89956]: qh: Socket '/usr/local/nagios/var/rw/nagios.qh' successfull
Oct 06 11:51:46 ip-172-31-38-150.ec2.internal nagios[89956]: qh: core query handler ready
```

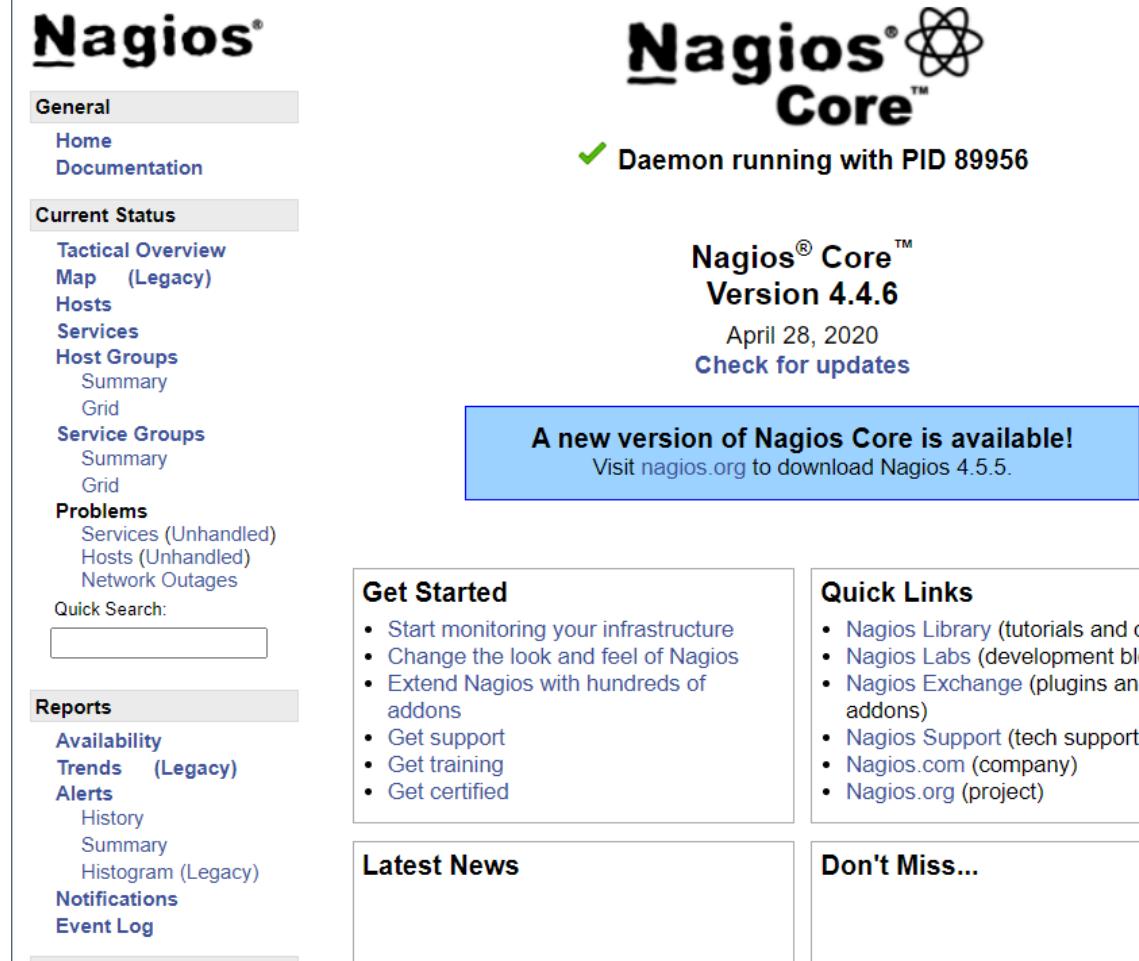
21. Go back to EC2 Console and copy the Public IP address of this instance

22. Open up your browser and look for
http://<your_public_ip_address>/nagios



Enter username as nagiosadmin and password which you set in Step 16.

23. After entering the correct credentials, you will see this page.



The screenshot shows the Nagios Core 4.4.6 monitoring interface. At the top right, it displays "Nagios® Core™ Version 4.4.6" and the date "April 28, 2020". A prominent green checkmark indicates "Daemon running with PID 89956". Below this, a blue banner announces "A new version of Nagios Core is available! Visit nagios.org to download Nagios 4.5.5." On the left side, there is a navigation menu with sections like General, Current Status, Problems, Reports, and Notifications. The "Current Status" section is expanded, showing links for Tactical Overview, Map (Legacy), Hosts, Services, Host Groups, Service Groups, and Problems. The "Problems" section lists Services (Unhandled), Hosts (Unhandled), and Network Outages. The "Reports" section lists Availability, Trends (Legacy), Alerts, and Notifications. The "Notifications" section lists Event Log. On the right side, there are two boxes: "Get Started" with a list of five items: Start monitoring your infrastructure, Change the look and feel of Nagios, Extend Nagios with hundreds of addons, Get support, Get training, and Get certified; and "Quick Links" with a list of six items: Nagios Library (tutorials and docs), Nagios Labs (development blog), Nagios Exchange (plugins and addons), Nagios Support (tech support), Nagios.com (company), and Nagios.org (project). Below these boxes are "Latest News" and "Don't Miss..." sections, which are currently empty.

This means that Nagios was correctly installed and configured with its plugins so far.

Conclusion: We have successfully installed and configured Nagios Core, Nagios Plugins, and NRPE on a Linux machine. This enables us to effectively manage system performance and proactively address potential issues.

Advanced DevOps

Lab Experiment 10

Aim: To perform Port, Service monitoring, Windows/Linux server monitoring using

Nagios. **Steps:**

Prerequisites: AWS Free Tier, Nagios Server running on Amazon Linux Machine.

1. To Confirm that Nagios is running **on the server side**, run this *sudo systemctl status nagios* on the “NAGIOS HOST”.

```
ec2-user@ip-172-31-81-4:~ % 
Amazon Linux 2023
https://aws.amazon.com/linux/amazon-linux-2023

Last login: Mon Sep 23 15:36:29 2024 from 152.58.4.81
[ec2-user@ip-172-31-81-4 ~]$ sudo systemctl status nagios
● nagios.service - Nagios Core 4.5.5
    Loaded: loaded (/usr/lib/systemd/system/nagios.service; enabled; preset: disabled)
      Active: active (running) since Mon 2024-09-23 16:32:36 UTC; 18min ago
        Docs: https://www.nagios.org/documentation
    Process: 1969 ExecStartPre=/usr/local/nagios/bin/nagios -v /usr/local/nagios/etc/nagios.cfg (code=exited, status=0/SUCCESS)
   Process: 1971 ExecStart=/usr/local/nagios/bin/nagios -d /usr/local/nagios/etc/nagios.cfg (code=exited, status=0/SUCCESS)
 Main PID: 1972 (nagios)
    Tasks: 6 (limit: 1112)
   Memory: 6.8M
      CPU: 320ms
     CGroup: /system.slice/nagios.service
             └─1972 /usr/local/nagios/bin/nagios -d /usr/local/nagios/etc/nagios.cfg
                 ├─1974 /usr/local/nagios/bin/nagios --worker /usr/local/nagios/var/rw/nagios.qh
                 ├─1975 /usr/local/nagios/bin/nagios --worker /usr/local/nagios/var/rw/nagios.qh
                 ├─1976 /usr/local/nagios/bin/nagios --worker /usr/local/nagios/var/rw/nagios.qh
                 ├─1977 /usr/local/nagios/bin/nagios --worker /usr/local/nagios/var/rw/nagios.qh
                 └─1983 /usr/local/nagios/bin/nagios -d /usr/local/nagios/etc/nagios.cfg

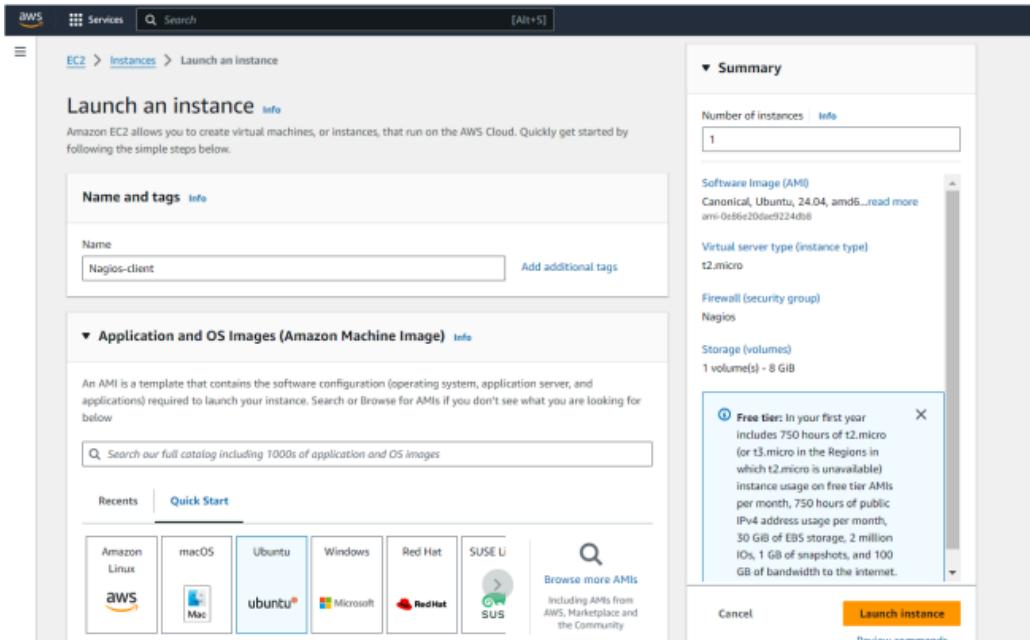
Sep 23 16:32:36 ip-172-31-81-4.ec2.internal systemd[1]: Started nagios.service - Nagios Core 4.5.5.
```

You can proceed if you get this message.

2. Before we begin,

To monitor a Linux machine, create an Ubuntu 20.04 server EC2 Instance in AWS.

Provide it with the same security group as the Nagios Host and name it ‘linux-client’ alongside the host.



For now, leave this machine as is, and go back to your nagios HOST machine.

3. On the server, run this command

```
ps -ef | grep nagios
Last login: Sat Oct  5 16:58:17 2024 from 42.111.112.18
[ec2-user@ip-172-31-43-65 ~]$ ps -ef | grep nagios
nagios    97412      1  0 17:34 ?          00:00:00 /usr/local/nagios/bin/nagios -d /usr/local/nagios
nagios    97413    97412  0 17:34 ?          00:00:00 /usr/local/nagios/bin/nagios --worker /
s.qh
nagios    97414    97412  0 17:34 ?          00:00:00 /usr/local/nagios/bin/nagios --worker /
s.qh
nagios    97415    97412  0 17:34 ?          00:00:00 /usr/local/nagios/bin/nagios --worker /
s.qh
nagios    97416    97412  0 17:34 ?          00:00:00 /usr/local/nagios/bin/nagios --worker /
s.qh
nagios    97417    97412  0 17:34 ?          00:00:00 /usr/local/nagios/bin/nagios -d /usr/local/nagios
ec2-user   98423    98399  0 17:51 pts/2        00:00:00 grep --color=auto nagios
```

4. Become a root user and create 2 folders

```
sudo su
mkdir /usr/local/nagios/etc/objects/monitorhosts
mkdir /usr/local/nagios/etc/objects/monitorhosts/linuxhosts
```

```
[ec2-user@ip-172-31-43-65 ~]$ sudo su
mkdir /usr/local/nagios/etc/objects/monitorhosts
mkdir /usr/local/nagios/etc/objects/monitorhosts/linuxhosts
[root@ip-172-31-43-65 ec2-user]# |
```

5. Copy the sample localhost.cfg file to linuxhost folder

```
cp /usr/local/nagios/etc/objects/localhost.cfg
/usr/local/nagios/etc/objects/monitorhosts/linuxhosts/linuxserver.cfg
```

```
[root@ip-172-31-81-4 ec2-user]# cp /usr/local/nagios/etc/objects/localhost.cfg /usr/local/nagios/etc/objects/monitorhosts/linuxhosts/linuxserver.cfg
[root@ip-172-31-81-4 ec2-user]# |
```

6. Open linuxserver.cfg using nano and make the following changes

```
nano
/usr/local/nagios/etc/objects/monitorhosts/linuxhosts/linuxserver.cfg
```

Change the hostname to linuxserver (EVERYWHERE ON THE FILE)
 Change address to the public IP address of your **LINUX CLIENT.**

```
GNU nano 5.8                               /usr/local/nagios/etc/objects/monitorhosts/linuxhosts/linuxserver.cfg
#####
# LOCALHOST.CFG - SAMPLE OBJECT CONFIG FILE FOR MONITORING THIS MACHINE
#
##
## NOTE: This config file is intended to serve as an *extremely* simple
##       example of how you can create configuration entries to monitor
##       the local (Linux) machine.
##
#####

#####
# HOST DEFINITION
#
#####
# Define a host for the local machine
define host {
    use          linux-server           ; Name of host template to use
                           ; This host definition will inherit all variables that are defined
                           ; in (or inherited by) the linux-server host template definition.
    host_name    localhost
    alias        localhost
    address      127.0.0.1
}

#####
# HOST GROUP DEFINITION
#
```

Change hostgroup_name under hostgroup to linux-servers1

Everywhere else on the file, change the hostname to linuxserver instead of localhost.

7. Open the Nagios Config file and add the following line

```
nano /usr/local/nagios/etc/nagios.cfg
```

```
##Add this line
cfg_dir=/usr/local/nagios/etc/objects/monitorhosts/
GNU nano 5.8                               /usr/local/nagios/etc/nagios.cfg
#####
# NAGIOS.CFG - Sample Main Config File for Nagios 4.5.5
#
# Read the documentation for more information on this configuration
# file. I've provided some comments here, but things may not be so
# clear without further explanation.
#
#
#####
# LOG FILE
# This is the main log file where service and host events are logged
# for historical purposes. This should be the first option specified
# in the config file!!!
log_file=/usr/local/nagios/var/nagios.log

# OBJECT CONFIGURATION FILE(S)
# These are the object configuration files in which you define hosts,
# host groups, contacts, contact groups, services, etc.
# You can split your object definitions across several config files
# if you wish (as shown below), or keep them all in a single config file.

# You can specify individual object config files as shown below:
cfg_file=/usr/local/nagios/etc/objects/commands.cfg
cfg_file=/usr/local/nagios/etc/objects/contacts.cfg
cfg_file=/usr/local/nagios/etc/objects/timeperiods.cfg
cfg_file=/usr/local/nagios/etc/objects/templates.cfg
cfg_dir=/usr/local/nagios/etc/objects/monitorhosts/
# Definitions for monitoring the local (Linux) host
cfg_file=/usr/local/nagios/etc/objects/localhost.cfg

# Definitions for monitoring a Windows machine

[G Help   ^W Write Out   ^W Where Is   ^W Cut   ^W Paste   T Execute Justify   L Location Go To Line   U Undo   R Redo   M-A Set Mark   M-G Copy   M-J To Bracket   M-O Where Was
^X Exit   ^R Read File   ^R Replace
```

8. Verify the configuration files

```
/usr/local/nagios/bin/nagios -v /usr/local/nagios/etc/nagios.cfg
```

```
Running pre-flight check on configuration data...

Checking objects...
    Checked 8 services.
    Checked 2 hosts.
    Checked 2 host groups.
    Checked 0 service groups.
    Checked 1 contacts.
    Checked 1 contact groups.
    Checked 24 commands.
    Checked 5 time periods.
    Checked 0 host escalations.
    Checked 0 service escalations.

Checking for circular paths...
    Checked 2 hosts
    Checked 0 service dependencies
    Checked 0 host dependencies
    Checked 5 timeperiods

Checking global event handlers...
Checking obsessive compulsive processor commands...
Checking misc settings...

Total Warnings: 0
Total Errors: 0

Things look okay - No serious problems were detected during the pre-flight check
[root@ip-172-31-43-65 ec2-user]# |
```

You are good to go if there are no errors.

9. Restart the nagios service

```
service nagios restart
[root@ip-172-31-81-4 ec2-user]# service nagios restart
Redirecting to /bin/systemctl restart nagios.service
[root@ip-172-31-81-4 ec2-user]# sudo systemctl status nagios
```

Now it is time to switch to the client machine.

10. SSH into the machine or simply use the EC2 Instance Connect feature.

11. Make a package index update and install gcc, nagios-nrpe-server and the plugins.

```
sudo apt update -y
sudo apt install gcc -y
sudo apt install -y nagios-nrpe-server nagios-plugins
ubuntu@ip-172-31-33-76:~$ sudo apt update -y
sudo apt install gcc -y
sudo apt install -y nagios-nrpe-server nagios-plugins
Hit:1 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble InRelease
Get:2 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates InRelease [126 kB]
Get:3 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-backports InRelease [126 kB]
Get:4 http://security.ubuntu.com/ubuntu noble-security InRelease [126 kB]
Get:5 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble/universe amd64 Packages [15.0 MB]
Get:6 http://security.ubuntu.com/ubuntu noble-security/main amd64 Packages [382 kB]
Get:7 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble/universe Translation-en [5982 kB]
Get:8 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble/universe amd64 Components [3871 kB]
Get:9 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble/universe amd64 c-n-f Metadata [301 kB]
Get:10 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble/multiverse amd64 Packages [269 kB]
Get:11 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble/multiverse Translation-en [118 kB]
Get:12 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble/multiverse amd64 Components [35.0 kB]
Get:13 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble/multiverse amd64 c-n-f Metadata [8328 B]
Get:14 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates/main amd64 Packages [537 kB]
Get:15 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates/main Translation-en [132 kB]
Get:16 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates/main amd64 c-n-f Metadata [8860 B]
Get:17 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates/universe amd64 Packages [384 kB]
Get:18 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates/universe Translation-en [159 kB]
Get:19 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates/universe amd64 Components [45.0 kB]
Get:20 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates/universe amd64 c-n-f Metadata [14.9 kB]
Get:21 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates/multiverse amd64 Packages [14.4 kB]
Get:22 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates/multiverse Translation-en [3608 B]
Get:23 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates/multiverse amd64 Components [212 kB]
```

12. Open nrpe.cfg file to make changes.

```
sudo nano /etc/nagios/nrpe.cfg
```

Under allowed_hosts, add your nagios host IP address like so

```

ubuntu@ip-172-31-83-152:~$ nano /etc/nagios/nrpe.cfg
GNU nano 7.2                               /etc/nagios/nrpe.cfg *
# This determines the effective user that the NRPE daemon should run as.
# You can either supply a username or a UID.
#
# NOTE: This option is ignored if NRPE is running under either inetd or xinetd
nrpe_user=nagios

# NRPE GROUP
# This determines the effective group that the NRPE daemon should run as.
# You can either supply a group name or a GID.
#
# NOTE: This option is ignored if NRPE is running under either inetd or xinetd
nrpe_group=nagios

# ALLOWED HOST ADDRESSES
# This is an optional comma-delimited list of IP address or hostnames
# that are allowed to talk to the NRPE daemon. Network addresses with a bit mask
# (i.e. 192.168.1.0/24) are also supported. Hostname wildcards are not currently
# supported.
#
# Note: The daemon only does rudimentary checking of the client's IP
# address. I would highly recommend adding entries in your /etc/hosts.allow
# file to allow only the specified host to connect to the port
# you are running this daemon on.
#
# NOTE: This option is ignored if NRPE is running under either inetd or xinetd
allowed_hosts=127.0.0.1,:1,3.86.12.126

# COMMAND ARGUMENT PROCESSING

```

File menu: Help, Exit, Write Out, Read File, Where Is, Replace, Cut, Paste, Execute Justify, Location Go To Line, Undo, Redo, Set Mark, Copy, To Bracket, Where Was.

13. Restart the NRPE server

```

sudo systemctl restart nagios-nrpe-server
ubuntu@ip-172-31-83-152:~$ sudo nano /etc/nagios/nrpe.cfg

ubuntu@ip-172-31-83-152:~$ sudo systemctl restart nagios-nrpe-server
ubuntu@ip-172-31-83-152:~$ 

```

14. Now, check your nagios dashboard and you'll see a new host being added.

Click on Hosts.

Nagios® Core

Daemon running with PID 4560

Nagios® Core™ Version 4.5.5
September 17, 2024
Check for updates

Get Started

- Start monitoring your infrastructure
- Change the look and feel of Nagios
- Extend Nagios with hundreds of addons
- Get support
- Get training
- Get certified

Latest News

Don't Miss...

Quick Links

- Nagios Library (tutorials and docs)
- Nagios Labs (development blog)
- Nagios Exchange (plugins and addons)
- Nagios Support (tech support)
- Nagios.com (company)
- Nagios.org (project)

Copyright © 2010-2024 Nagios Core Development Team and Community Contributors. Copyright © 1999-2009 Ethan Galstad. See the THANKS file for more information on contributors.

Nagios Core is licensed under the GNU General Public License and is provided AS IS WITH NO WARRANTY OF ANY KIND, INCLUDING THE WARRANTY OF DESIGN, MERCHANTABILITY, AND FITNESS FOR A PARTICULAR PURPOSE. Nagios, Nagios Core and the Nagios logo are trademarks, service marks, registered trademarks or registered service marks owned by Nagios Enterprises, LLC. Use of the Nagios mark is governed by the trademark use restrictions.

Click on linuxserver to see the host details

Current Network Status

Last Updated: Sun Oct 6 17:55:03 UTC 2024
Updated every 90 seconds
Nagios® Core™ 4.5.5 - www.nagios.org
Logged in as nagiosadmin

[View Service Status Detail For All Host Groups](#)
[View Status Overview For All Host Groups](#)
[View Status Summary For All Host Groups](#)
[View Status Grid For All Host Groups](#)

Host Status Totals				Service Status Totals			
Up	Down	Unreachable	Pending	Ok	Warning	Unknown	Critical
2	0	0	0	6	1	0	1
All Problems				All Types			
0	2			2	8		

Host Status Details For All Host Groups

Host	Status	Last Check	Duration	Status Information
linuxserver	UP	10-06-2024 17:50:12	0d 0h 24m 51s	PING OK - Packet loss = 0%, RTA = 0.77 ms
localhost	UP	10-06-2024 17:53:57	1d 0h 21m 53s	PING OK - Packet loss = 0%, RTA = 0.03 ms

Results 1 - 2 of 2 Matching Hosts

You can click Services to see all services and ports being monitored.

Host Information

Last Updated: Sun Oct 6 17:43:35 UTC 2024
Updated every 90 seconds
Nagios® Core™ 4.5.5 - www.nagios.org
Logged in as nagiosadmin

[View Status Detail For This Host](#)
[View Alert History For This Host](#)
[View Alert Histogram For This Host](#)
[View Availability Report For This Host](#)
[View Notifications For This Host](#)

Host
localhost
(localhost)

Member of
linux-servers, lnx-servers1

Host State Information

Host Status: UP (for 1d 0h 10m 25s)
Status Information: PING OK - Packet loss = 0%, RTA = 0.03 ms
Performance Data: rta=0.03000ms;3000.000000;5000.000000;0.000000 pl=0%:80;100.0
Current Attempt: 1/1 (HARD state)
Last Check Time: 10-06-2024 17:38:57
Check Type: ACTIVE
Check Latency / Duration: 0.000 / 140 seconds
Next Scheduled Active Check: 10-06-2024 17:43:57
Last State Change: 10-05-2024 17:33:10
Last Notification: N/A (notification 0)
Is This Host Flapping? NO (0.00% state change)
In Scheduled Downtime? NO
Last Update: 10-06-2024 17:43:34 (0d 0h 0m 1s ago)

Active Checks: ENABLED
Passive Checks: ENABLED
Obsessing: ENABLED
Notifications: ENABLED
Event Handler: ENABLED
Flap Detection: ENABLED

Host Commands

- Locate host on map
- Disable active checks of this host
- Re-schedule the next check of this host
- Submit passive check result for this host
- Stop accepting passive checks for this host
- Stop obsessing over this host
- Disable notifications for this host
- Send custom host notification
- Schedule downtime for this host
- Schedule downtime for all services on this host
- Disable notifications for all services on this host
- Enable notifications for all services on this host
- Schedule a check of all services on this host
- Disable checks of all services on this host
- Enable checks of all services on this host
- Disable event handler for this host
- Disable flap detection for this host
- Clear flapping state for this host

Host Comments

Add a new comment Delete all comments

Entry Time Author Comment Comment ID Persistent Type Expires Actions

This host has no comments associated with it.

Current Network Status

Last Updated: Sun Oct 6 17:58:02 UTC 2024
Updated every 90 seconds
Nagios® Core™ 4.5.5 - www.nagios.org
Logged in as nagiosadmin

[View History For All hosts](#)
[View Notifications For All Hosts](#)
[View Host Status Detail For All Hosts](#)

Host Status Totals				Service Status Totals			
Up	Down	Unreachable	Pending	Ok	Warning	Unknown	Critical
2	0	0	0	6	1	0	1
All Problems				All Types			
0	2			2	8		

Service Status Details For All Hosts

Host	Service	Status	Last Check	Duration	Attempt	Status Information
localhost	Current Load	OK	10-06-2024 17:56:27	1d 0h 24m 52s	1/4	OK - load average 0.00, 0.00, 0.00
localhost	Current Users	OK	10-06-2024 17:57:42	1d 0h 24m 14s	1/4	USERS OK - 6 users currently logged in
localhost	HTTP	WARNING	10-06-2024 17:53:57	0d 0h 19m 5s	4/4	HTTP WARNING: HTTP/1.1 403 Forbidden - 319 bytes in 0.001 second response time
localhost	PING	OK	10-06-2024 17:55:12	1d 0h 22m 59s	1/4	PING OK - Packet loss = 0%, RTA = 0.03 ms
localhost	Root Partition	OK	10-06-2024 17:57:04	1d 0h 22m 22s	1/4	DISK OK - free space / 5567 MB (68.59% inode=98%):
localhost	SSH	OK	10-06-2024 17:53:19	1d 0h 21m 44s	1/4	SSH OK - OpenSSH_8.7 (protocol 2.0)
localhost	Swap Usage	CRITICAL	10-06-2024 17:54:34	1d 0h 31m 7s	4/4	SWAP CRITICAL - 0% free (0 MB out of 0 MB) - Swap is either disabled, not present, or of zero size.
localhost	Total Processes	OK	10-06-2024 17:56:22	1d 0h 20m 29s	1/4	PROCS OK, 39 processes with STATE = RSDOT

Results 1 - 8 of 8 Matching Services

As you can see, we have our linuxserver up and running. It is showing critical status on HTTP due to permission errors and swap because there is no partition created.

In this case, we have monitored -

Servers: 1 linux server

Services: swap

Ports: 22, 80 (ssh, http)

Processes: User status, Current load, total processes, root partition, etc.

Recommended Cleanup

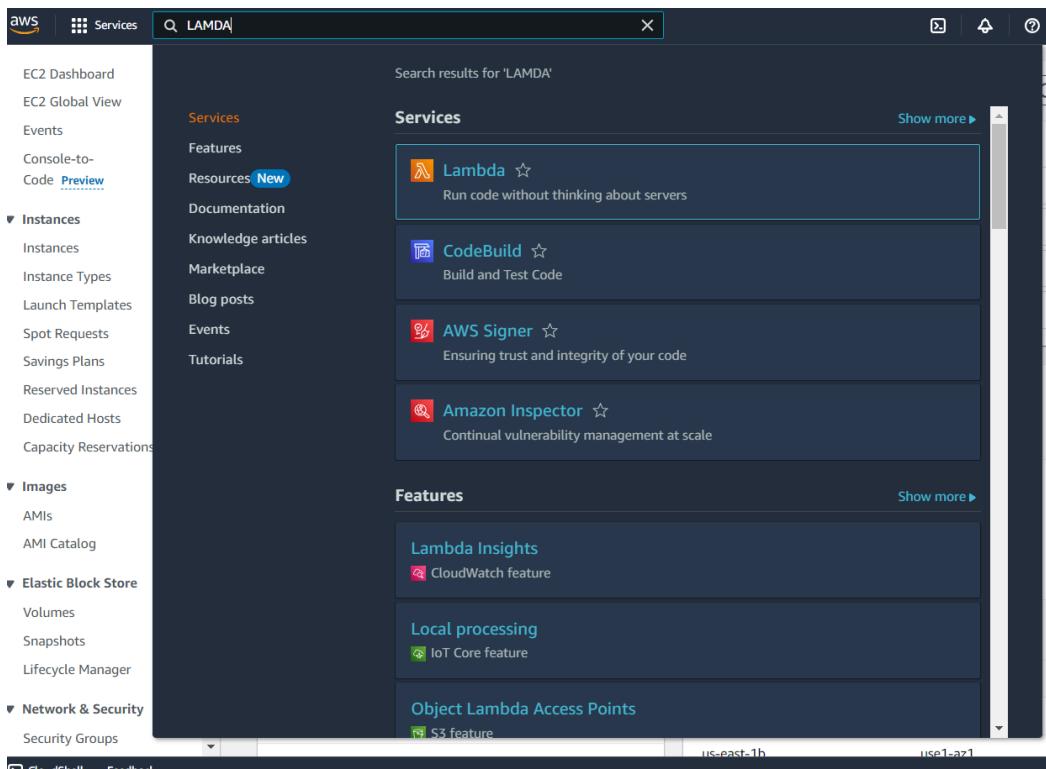
- Terminate both of your EC-2 instances to avoid charges.
- Delete the security group if you created a new one (it won't affect your bill, you may avoid it)

Conclusion: In conclusion, the experiment focused on monitoring ports, services, and a Linux server using Nagios. Through the step-by-step process, we successfully configured Nagios to monitor essential network services on the Linux server. By setting up both the Nagios host and client, we were able to track system performance, ensure service availability, and monitor key metrics like CPU and memory usage.

Aim: To understand AWS Lambda, its workflow, various functions and create your first Lambda functions using Python / Java / Nodejs.

Step 1: Accessing AWS

Log in to your AWS Personal/Academy account. Navigate to the Lambda service by searching for "Lambda" in the AWS Management Console.



Step 2: Creating a New Lambda Function

Click on the "Create function" button. Provide a name for your Lambda function and select the language you wish to use, such as Python 3.12. For architecture, choose x86, and for execution role, opt to create a new role with basic Lambda g permissions.

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

```
1 * exports.handler = async (event) => {
2   console.log(event);
3   return 'Hello from Lambda!';
4 };
5
```

Step 3: Configuring Basic Settings

To modify the basic settings, navigate to the "Configuration" tab and click on "Edit" under General Settings. Here, you can add a description and adjust the memory and timeout settings. For this experiment, I set the timeout to 1 second, which is sufficient for testing.

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 Info
Choose the language to use to write your function. Note that the console code editor supports only Node.js, Python, and Ruby.

Architecture Info
Choose the instruction set architecture you want for your function code.
 x86_64
 arm64

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

▶ Change default execution role

Permissions [Info](#)

By default, Lambda will create an execution role with permissions to upload logs to 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

ⓘ Role creation might take a few minutes. Please do not delete the role or edit the trust or permissions policies in this role.

Lambda will create an execution role named ATHARV_LAMDA-role-0u7c9ooi, with permission to upload logs to Amazon CloudWatch Logs.

▶ Additional Configurations

Use additional configurations to set up code signing, function URL, tags, and Amazon VPC access for your function.

[Cancel](#) [Create function](#)

aws Services Search [Alt+S]

Successfully created the function lamda_demo. You can now change its code and configuration. To invoke your function with a test event, choose "Test".

Lambda > Functions > lamda_demo

lamda_demo

▼ Function overview [Info](#)

Throttle [Copy ARN](#) Actions ▾

Diagram Template

lambda_demo

+ Add trigger + Add destination

Description -

Last modified 26 seconds ago

Function ARN [arn:aws:lambda:eu-north-1:010928207735:function:lambda_demo](#)

Function URL [Info](#) -

Export to Application Composer Download ▾

Learn how to implement common use cases in AWS Lambda.

Create a simple web app

In this tutorial you will learn how to:

- Build a simple web app consisting of a Lambda function with a function URL that outputs a webpage
- Invoke your function through its function URL

[Learn more](#) [Start tutorial](#)

Code Test Monitor Configuration Aliases Versions

CloudShell Feedback © 2024, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences

Successfully created the function **lambda_demo**. You can now change its code and configuration. To invoke your function with a test event, choose "Test".

The screenshot shows the AWS Lambda console interface. At the top, there's a green banner with the message: "Successfully created the function lambda_demo. 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 currently selected. In the main area, there's a "Code source" section with an "Info" link and an "Upload from" button. A toolbar above the code editor includes "File", "Edit", "Find", "View", "Go", "Tools", "Window", "Test" (which is highlighted in blue), and "Deploy". The code editor itself shows a file named "lambda_function.py" with the following content:

```
1 import json
2
3 def lambda_handler(event, context):
4     # TODO implement
5     return {
6         'statusCode': 200,
7         'body': json.dumps('Hello from Lambda!')
8     }
9
```

Step 4: Testing the Function

Click on the "Test" tab and select "Create a new event." Name your event, set the event sharing to private, and choose the "hello-world" template.

The screenshot shows the "Test event" configuration page. At the top, there's a "Test event" link and an "Info" link, followed by "Save" and "Test" buttons. The main area contains the following fields:

- Test event action:** A radio button group with "Create new event" selected (indicated by a blue border) and "Edit saved event" (indicated by a grey border).
- Event name:** An input field containing "MyEventName". Below it, a note says: "Maximum of 25 characters consisting of letters, numbers, dots, hyphens and underscores."
- Event sharing settings:** A radio button group with "Private" selected (indicated by a blue border) and "Shareable" (indicated by a grey border). Below "Private", a note says: "This event is only available in the Lambda console and to the event creator. You can configure a total of 10. [Learn more](#)". Below "Shareable", a note says: "This event is available to IAM users within the same account who have permissions to access and use shareable events. [Learn more](#)".
- Template - optional:** A dropdown menu showing "hello-world".
- Event JSON:** A code editor showing a sample JSON object:

```
1 {  
2     "key1": "value1",  
3     "key2": "value2",  
4     "key3": "value3"  
5 }
```

A "Format JSON" button is located to the right of the code editor.

Code source **Info**

File Edit Find View Go Tools Window **Test** Deploy

Upload from ▾

Go to Anything (Ctrl-P)

Environment

lambda_demo /

lambda_function.py

lambda_function

Configure test event Ctrl-Shift-C

Private saved events

● demo1

```
1 import json
2
3 def lambda_handler(event, context):
4     # TODO implement
5     return {
6         'statusCode': 200,
7         'body': json.dumps('Hello from Lambda!')
8     }
```

Code **Test** **Monitor** **Configuration** **Aliases** **Versions**

Code source **Info**

File Edit Find View Go Tools Window **Test** Deploy

Upload from ▾

Go to Anything (Ctrl-P)

Environment

lambda_demo /

lambda_function.py

lambda_function

Execution result:

Execution results

Test Event Name: demo1

Response:

```
{ "statusCode": 200,
  "body": "\\"Hello from Lambda\\\""
}
```

Function Logs

```
START RequestId: 86829fa5-e154-46b3-8ff5-6f12ba6c3efa Version: $LATEST
END RequestId: 86829fa5-e154-46b3-8ff5-6f12ba6c3efa
REPORT RequestId: 86829fa5-e154-46b3-8ff5-6f12ba6c3efa Duration: 1.35 ms Billed Duration: 2 ms Memory Size: 128 MB Max Memory Used: 32 MB
```

Request ID

```
86829fa5-e154-46b3-8ff5-6f12ba6c3efa
```

The screenshot shows the AWS Lambda function editor interface. The top navigation bar includes 'File', 'Edit', 'Find', 'View', 'Go', 'Tools', 'Window', 'Test' (which is currently selected), 'Deploy', and a status message 'Changes not deployed'. On the left, there's a sidebar with 'Environment' and a search bar 'Go to Anything (Ctrl-P)'. The main area displays the code for 'lambda_function.py':

```
1 import json
2
3 def lambda_handler(event, context):
4     # TODO implement
5     my_string="Hello this is Exp 11"
6     return {
7         'statusCode': 200,
8         'body': json.dumps(my_string);
9     }
10
```

Step 5: Running the Test

In the Code section, select the newly created event from the dropdown menu and click on "Test." You should see the output displayed below.

The screenshot shows the AWS Lambda function editor interface after running a test. The 'Test' button is now greyed out. The main area displays the 'Execution result' tab, which shows the following details:

- Test Event Name:** demo1
- Response:**

```
{
    "statusCode": 200,
    "body": "\\"Hello this is Exp 11\\\""
}
```
- Function Logs:**

```
START RequestId: 298ba9d9-f73b-4abc-927b-6d5e5c8ce4ec Version: $LATEST
END RequestId: 298ba9d9-f73b-4abc-927b-6d5e5c8ce4ec
REPORT RequestId: 298ba9d9-f73b-4abc-927b-6d5e5c8ce4ec Duration: 1.62 ms Billed Duration: 2 ms Memory Size: 128 MB Max Memory Used: 32 MB Init Du
```
- Request ID:** 298ba9d9-f73b-4abc-927b-6d5e5c8ce4ec

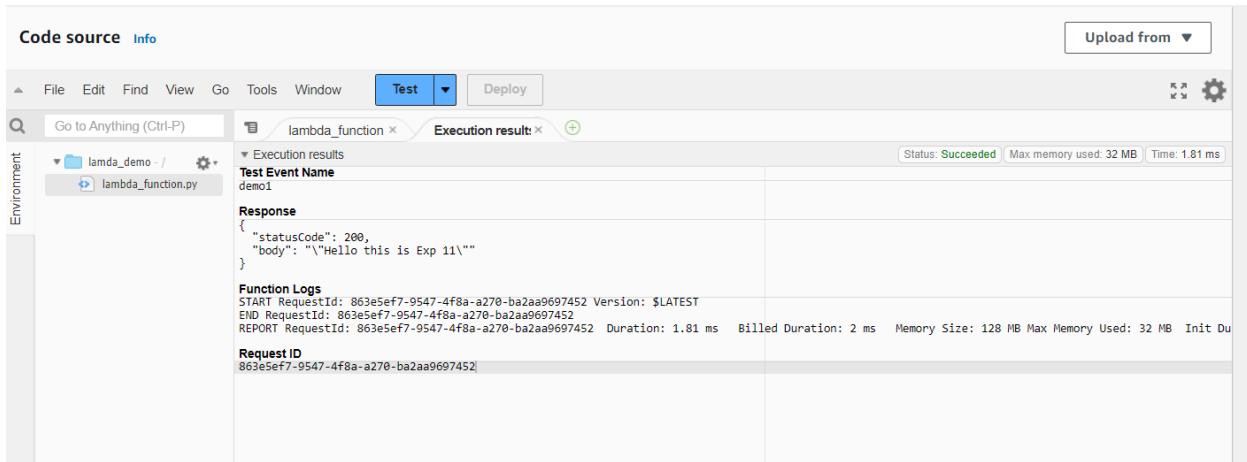
Step 6: Editing and Deploying the Code

You can modify your Lambda function's code as needed. I updated the code to display a new string. After making changes, press 'Ctrl + S' to save and then click on "Deploy" to apply the updates.



Step 7: Final Testing

Return to the "Test" tab and execute the test again to observe the output. You should see a status code of 200 along with your string output and function logs confirming a successful deployment.



The screenshot shows the AWS Lambda Test interface. At the top, there are tabs for 'Code source' and 'Info'. Below the tabs is a menu bar with File, Edit, Find, View, Go, Tools, Window, a 'Test' button, and a 'Deploy' button. To the right of the 'Test' button is an 'Upload from' dropdown. On the left, there's a sidebar labeled 'Environment' with a tree view showing 'lambda_demo /' and 'lambda_function.py'. The main area has tabs for 'Execution results' and 'Execution result:' (which is currently selected). Under 'Execution results', it shows a 'Test Event Name' of 'demo1'. The 'Response' section displays a JSON object: { "statusCode": 200, "body": "Hello this is Exp 11\\\""}'. The 'Function Logs' section contains several log entries, including 'START RequestId: 863e5ef7-9547-4f8a-a270-ba2aa9697452 Version: \$LATEST', 'END RequestId: 863e5ef7-9547-4f8a-a270-ba2aa9697452', and 'REPORT RequestId: 863e5ef7-9547-4f8a-a270-ba2aa9697452 Duration: 1.81 ms Billed Duration: 2 ms Memory Size: 128 MB Max Memory Used: 32 MB Init Duration: 0 ms'. The 'Request ID' is also listed as 863e5ef7-9547-4f8a-a270-ba2aa9697452.

Conclusion:

In this experiment, you created and tested your first AWS Lambda function using Python. You learned to navigate the AWS Management Console, configure basic settings, and modify your function's code. This experience highlights the ease of deploying serverless applications with Lambda, allowing you to focus on coding rather than infrastructure management. You now have a foundational understanding to explore more complex serverless solutions and integrate AWS services for greater functionality.

Aim: To create a Lambda function which will log “An Image has been added” once you add an object to a specific bucket in S3.

Step 1: Create a s3 bucket. 1) Search for S3 bucket in the services search. Then click on create bucket.

2) Keep the bucket as a general purpose bucket. Give a name to your bucket.

General configuration

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

Bucket type [Info](#)

General purpose
Recommended for most use cases and access patterns. General purpose buckets are the original S3 bucket type. They allow a mix of storage classes that redundantly store objects across multiple Availability Zones.

Directory
Recommended for low-latency use cases. These buckets use only the S3 Express One Zone storage class, which provides faster processing of data within a single Availability Zone.

Bucket name [Info](#)
nishantbucket24

Bucket name must be unique within the global namespace and follow the bucket naming rules. [See rules for bucket naming](#)

Copy settings from existing bucket - optional
Only the bucket settings in the following configuration are copied:
[Choose bucket](#)

Format: s3://bucket/prefix

Object Ownership [Info](#)

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

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

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

Object Ownership
Bucket owner enforced

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.

4) Keeping all other options the same, click on create. This would create your bucket. Now click on the name of the bucket.

Successfully created bucket "s3lamdaexp11". To upload files and folders, or to configure additional bucket settings, choose [View details](#).

Amazon S3 > Buckets

Account snapshot - updated every 24 hours [All AWS Regions](#)
Storage lens provides visibility into storage usage and activity trends. [Learn more](#)

[View Storage Lens dashboard](#)

General purpose buckets [Info](#) [All AWS Regions](#)

Buckets are containers for data stored in S3.

Name	AWS Region	IAM Access Analyzer	Creation date
elasticbeanstalk-eu-north-1-010928207735	Europe (Stockholm) eu-north-1	View analyzer for eu-north-1	August 14, 2024, 22:12:26 (UTC+05:30)
s3lamdaexp11	US East (N. Virginia) us-east-1	View analyzer for us-east-1	October 7, 2024, 09:40:50 (UTC+05:30)

5) Here, click on upload, then add files. Select any image that you want to upload in the bucket and click on upload.

Amazon S3 > Buckets > s3lamdaexp11

s3lamdaexp11 [Info](#)

Objects (0) [Info](#)

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

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

Find objects by prefix

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

[Upload](#)

Amazon S3 > Buckets > s3lamdaexp11 > Upload

Upload [Info](#)

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

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

Files and folders (1 Total, 990.9 KB)

All files and folders in this table will be uploaded.

Name	Folder
football.jpg	-

Remove [Add files](#) [Add folder](#)

Find by name

Destination [Info](#)

Destination

<s3://s3lamdaexp11>

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

► Permissions
Grant public access and access to other AWS accounts.

► Properties
Specify storage class, encryption settings, tags, and more.

6) The image has been uploaded to the bucket.

The screenshot shows the AWS Lambda 'Upload: status' page. At the top, a green bar indicates 'upload succeeded' with a link to 'View details below.' Below this, a message says 'The information below will no longer be available after you navigate away from this page.' The main section is titled 'Summary' and shows the destination as 's3://s3lamdaexp11'. It lists one file, 'football.jpg', which is 990.9 KB and has a status of 'Succeeded'. There are also sections for 'Files and folders' and 'Configuration'. Under 'Files and folders', there is a table showing the uploaded file.

Name	Folder	Type	Size	Status	Error
football.jpg	-	image/jpeg	990.9 KB	Succeeded	-

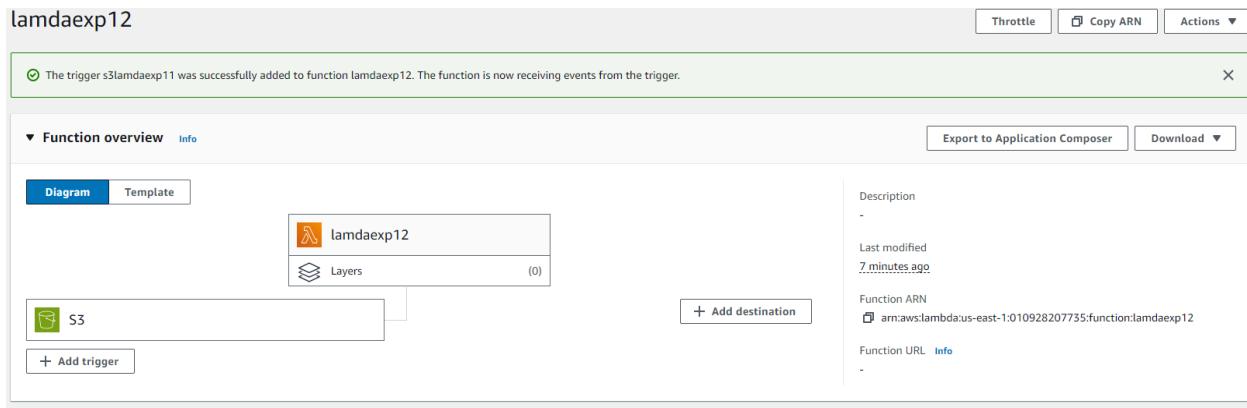
Step 2: Configure Lambda function

1) Go to the lambda function you had created before. (Services → Lambda → Click on name of function). Here, click on add trigger

The screenshot shows the 'Create function' wizard. It starts with a choice between 'Author from scratch', 'Use a blueprint', and 'Container image'. The 'Author from scratch' option is selected. The next step is 'Basic information', where the function name is set to 'lamdaexp12'. The 'Runtime' is chosen as 'Node.js 20.x' and the 'Architecture' as 'x86_64'. In the 'Permissions' section, there is a link to 'Change default execution role'. The wizard is currently at the 'Basic information' step.

2) Under trigger configuration, search for S3 and select it.

3) Here, select the S3 bucket you created for this experiment. Acknowledge the condition given by AWS. then click on Add. This will add the S3 bucket trigger to your function



- 4) Scroll down to the code section of the function. Add the following javascript code to the code area by replacing the existing code

```
export const handler = async (event) => {
if (!event.Records || event.Records.length === 0) {
console.error("No records found in the event.");
return {
statusCode: 400,
body: JSON.stringify('No records found in the event')
};
}
// Extract bucket name and object key from the event
const record = event.Records[0];
const bucketName = record.s3.bucket.name;
const objectKey = decodeURIComponent(record.s3.object.key.replace(/\+/g, ' ')); // Handle encoded keys
console.log(`An image has been added to the bucket ${bucketName}: ${objectKey}`);
console.log(`Event Source: ${record.eventSource}`);
console.log(`Event Source: ${record.eventSource}`);
console.log(`Event Source: ${record.eventSource}`);
console.log(`Event Source: ${record.eventSource}`);
return {
statusCode: 200,
body: JSON.stringify('Log entry created successfully!')
};
};
```

This JSON structure represents an S3 event notification triggered when an object is uploaded to an S3 bucket. It contains details about the event, including the bucket name (example-bucket), the object key (test/key), and metadata like the object's size, the event source (aws:s3), and the event time.

The screenshot shows a code editor interface with the following details:

- Title Bar:** Code source Info
- Toolbar:** File, Edit, Find, View, Go, Tools, Window, Test, Deploy, Changes not deployed
- Environment:** lamdaexp12 / index.mjs
- Code Area:** The code is an asynchronous handler for an S3 event. It logs the event if no records are found, extracts the bucket name and object key, decodes the object key, and logs the event source multiple times. It returns a 200 status with a success message.

```
1 exports.handler = async (event) => {
2   if (!event.Records || event.Records.length === 0) {
3     console.error('No records found in the event.');
4     return {
5       statusCode: 400,
6       body: JSON.stringify('No records found in the event')
7     };
8   }
9   // Extract bucket name and object key from the event
10  const record = event.Records[0];
11  const objectKey = decodeURIComponent(record.s3.object.key.replace(/\+/g, ' ')); // Handle encoded keys
12  const bucketName = record.s3.bucket.name;
13  console.log(`An image has been added to the bucket ${bucketName}: ${objectKey}`);
14  console.log(`Event Source: ${record.eventSource}`);
15  console.log(`Event Source: ${record.eventSource}`);
16  console.log(`Event Source: ${record.eventSource}`);
17  console.log(`Event Source: ${record.eventSource}`);
18  return {
19    statusCode: 200,
20    body: JSON.stringify('Log entry created successfully!')
21  };
22};
```

The screenshot shows a code editor interface with the following details:

- Title Bar:** Code source Info
- Toolbar:** File, Edit, Find, View, Go, Tools, Window, Test, Deploy, Changes not deployed
- Environment:** lamdaexp12 / index.mjs
- Code Area:** The code is identical to the one in the previous screenshot, representing an S3 event handler with logging logic.

```
1 exports.handler = async (event) => {
2   if (!event.Records || event.Records.length === 0) {
3     console.error('No records found in the event.');
4     return {
5       statusCode: 400,
6       body: JSON.stringify('No records found in the event')
7     };
8   }
9   // Extract bucket name and object key from the event
10  const record = event.Records[0];
11  const objectKey = decodeURIComponent(record.s3.object.key.replace(/\+/g, ' ')); // Handle encoded keys
12  const bucketName = record.s3.bucket.name;
13  console.log(`An image has been added to the bucket ${bucketName}: ${objectKey}`);
14  console.log(`Event Source: ${record.eventSource}`);
15  console.log(`Event Source: ${record.eventSource}`);
16  console.log(`Event Source: ${record.eventSource}`);
17  console.log(`Event Source: ${record.eventSource}`);
18  return {
19    statusCode: 200,
20    body: JSON.stringify('Log entry created successfully!')
21  };
22};
```

Private

This event is only available in the Lambda console and to the event creator. You can configure a total of 10. [Learn more](#)

 Shareable

This event is available to IAM users within the same account who have permissions to access and use shareable events. [Learn more](#)

Template - optional

hello-world

Event JSON

[Format JSON](#)

```
1 [ ]  
2 "Records": [  
3 {  
4     "eventVersion": "2.0",  
5     "eventSource": "aws:s3",  
6     "awsRegion": "us-east-1",  
7     "eventTime": "1970-01-01T00:00:00.000Z",  
8     "eventName": "ObjectCreated:Put",  
9     "userIdentity": {  
10         "principalId": "EXAMPLE"  
11     },  
12     "requestParameters": {  
13         "sourceIPAddress": "127.0.0.1"  
14     },  
15     "responseElements": {  
16         "x-amz-request-id": "EXAMPLE123456789",  
17         "x-amz-id-2": "EXAMPLE123/5678abcdefghijklambdaisawesome/mnopqrstuvwxyzABCDEFGHI"  
18     },  
19     "s3": {  
20         "s3SchemaVersion": "1.0",  
21         "configurationId": "testConfigRule",  
22         "bucket": {  
23             "name": "example-bucket",  
24             "ownerIdentity": {  
25                 "principalId": "EXAMPLE"  
26             },  
27             "arn": "arn:aws:s3:::example-bucket"  
28         },  
29         "object": {  
30             "key": "test%2Fkey",  
31         }  
32     }  
33 }
```

1:1 JSON Spaces: 2

[Cancel](#)

[Invoke](#)

[Save](#)

The test event **myevent1** was successfully saved.

Function URL [Info](#)

Step 3: Check the logs

1) To check the logs explicitly, search for CloudWatch on services and open it in a new tab

The screenshot shows the AWS search interface with the query 'cloud watch' entered in the search bar. The results are categorized under 'Services' and 'Features'.

Services

- CloudWatch (Monitor Resources and Applications)
- Athena (Serverless interactive analytics service)
- Amazon EventBridge (Serverless service for building event-driven applications)
- S3 (Scalable Storage in the Cloud)

Features

- CloudWatch dashboard (Systems Manager feature)
- Data sources (Athena feature)
- Create a SFTP server (AWS Transfer Family feature)
- Event buses

2) Here, Click on Logs → Log Groups. Select the log that has the lambda function name you just ran.

The screenshot shows the CloudWatch Log groups interface. At the top, there's a search bar labeled "Filter log groups or try prefix search" and a checkbox for "Exact match". Below the search bar is a table with one row. The first column contains a checkbox and the log group name "/aws/lambda/lamdaexp12". The second column shows the log class as "Standard". The third column has a "Configure" link. The fourth column shows retention settings: "Never expire". There are also "Metric filters" and "Contributor Insights" columns.

3) Here, under Log streams, select the log stream you want to check.

This screenshot shows the "Log group details" page for the log group "/aws/lambda/lamdaexp12". The "Log streams" tab is selected. It displays a table with one row for a log stream named "2024/10/07[\$LATEST]0bfd52dd5b8a444ab1e15bfe46be5f00". The table includes columns for ARN, Last event time (2024-10-07 04:34:00 UTC), and Duration (1.48 ms). Below the table is a "Logs" section with a "View logs" button.

4) Here again, we can see that 'An image has been added to the bucket'.

This screenshot shows the "Log events" page for the log stream "2024/10/07[\$LATEST]0bfd52dd5b8a444ab1e15bfe46be5f00". The log entries are listed in a table. One entry from October 7, 2024, at 04:34:00 UTC, with a duration of 1.48 ms, contains the message: "2024-09-30T09:24:40,324Z 01723939-720B-421b-a558-4321055754b INFO An image has been added to the bucket example-bucket: test/key".

Conclusion:

In this experiment, In addition to demonstrating the integration of AWS Lambda with S3, this experiment showcases the scalability and flexibility of serverless architectures. By leveraging these services, we can build applications that respond in real-time to changes in data, such as the addition of files to S3 buckets, without the need for managing underlying server infrastructure. This not only enhances efficiency but also reduces operational costs, allowing developers to focus on building features rather than maintaining systems. Furthermore, the ability to log and monitor events through CloudWatch opens opportunities for further automation and analytics, paving the way for more complex workflows and data processing solutions.