

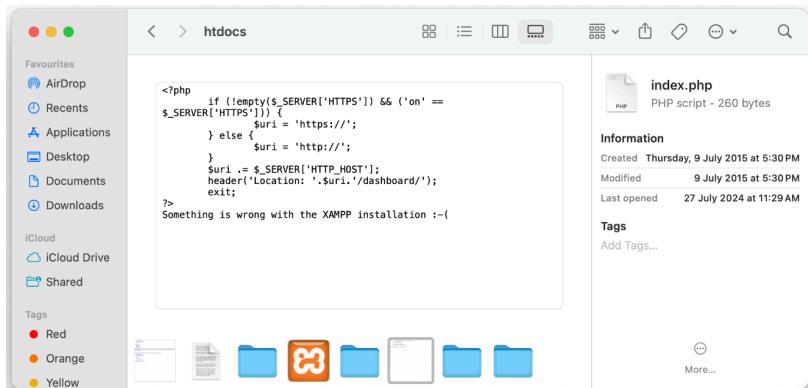
Experiment 1A

Hosting a static site using xampp

1. Download and open the xampp application



2. Go to the files “htdocs” and upload the php code there



3. Go to xampp, enter the folder name in the top link and run the file
You will see the website



Hosting a static website using AWS S3

1. Create a bucket in AWS S3 interface

The screenshot shows the AWS S3 console under the 'General purpose buckets' tab. It displays a list of buckets with columns for Name, AWS Region, IAM Access Analyzer, and Creation date. Two buckets are listed:

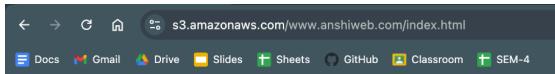
Name	AWS Region	IAM Access Analyzer	Creation date
www.anshiweb.com	US East (N. Virginia) us-east-1	View analyzer for us-east-1	August 1, 2024, 19:21:38 (UTC+05:30)
www.firstwebsiteofme.com	US East (N. Virginia) us-east-1	View analyzer for us-east-1	July 24, 2024, 21:35:15 (UTC+05:30)

2. Upload all your files in the buckets inventory

The screenshot shows the AWS S3 console for the bucket 'www.anshiweb.com'. It displays a list of objects with columns for Name, Type, Last modified, Size, and Storage class. Three objects are listed:

Name	Type	Last modified	Size	Storage class
error.html	html	August 1, 2024, 19:32:47 (UTC+05:30)	353.0 B	Standard
index.html	html	August 1, 2024, 19:32:47 (UTC+05:30)	275.0 B	Standard
style.css	css	August 1, 2024, 19:32:46 (UTC+05:30)	208.0 B	Standard

3. Click on the link of the bucket to go to the hosted site



Create instance using EC2

1. Open the aws console and choose EC2

The screenshot shows the AWS Console Home page. At the top, there's a navigation bar with a search bar and various icons. Below it, the main area has two sections: 'Recently visited' (with EC2 and S3 listed) and 'Applications (0)' (with a 'Create application' button). A sidebar on the right includes 'Reset to default layout' and '+ Add widgets' buttons.

2. Launch instance

This screenshot shows the 'Launch instance' page. It features a 'Launch instance' button in an orange box, a 'Migrate a server' button, and a note stating 'Note: Your instances will launch in the US East (N. Virginia) Region'. To the right, there's a 'Service health' section showing 'AWS Health Dashboard' and a green status message: 'This service is operating normally.'

3. Choose ubuntu

This screenshot shows the 'Browse more AMIs' page. It displays several AMI options: Amazon Linux (aws logo), macOS (Mac logo), Ubuntu (ubuntu logo), Windows (Microsoft logo), Red Hat (Red Hat logo), and SUSE Linux (SUSE logo). On the right, there's a search icon and a link to 'Browse more AMIs', with a sub-note: 'Including AMIs from AWS, Marketplace and the Community'.

4. Choose instance and key

▼ Instance type [Info](#) | [Get advice](#)

Instance type

t2.micro	Free tier eligible
Family: t2 1 vCPU 1 GiB Memory Current generation: true	
On-Demand Windows base pricing: 0.0162 USD per Hour	
On-Demand SUSE base pricing: 0.0116 USD per Hour	
On-Demand RHEL base pricing: 0.026 USD per Hour	
On-Demand Linux base pricing: 0.0116 USD per Hour	

All generations

[Compare instance types](#)

Additional costs apply for AMIs with pre-installed software

▼ 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*

vockey

[Create new key pair](#)

5. Configure network settings

▼ Network settings [Info](#) [Edit](#)

Network [Info](#)
vpc-0c4a6482e2565a490

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-2**' with the following rules:

Allow SSH traffic from Anywhere
Helps you connect to your instance

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 [X](#)

6. Configure storage settings

▼ **Configure storage** [Info](#) [Advanced](#)

1x GiB [▼](#) Root volume (Not encrypted)

(i) Free tier eligible customers can get up to 30 GB of EBS General Purpose (SSD) or Magnetic storage [X](#)

[Add new volume](#)

The selected AMI contains more instance store volumes than the instance allows. Only the first 0 instance store volumes from the AMI will be accessible from the instance

(i) Click refresh to view backup information [C](#)
The tags that you assign determine whether the instance will be backed up by any Data Lifecycle Manager policies.

0 x File systems [Edit](#)

7. This is the instance summary

▼ **Summary**

Number of instances [Info](#)

Software Image (AMI)
Canonical, Ubuntu, 24.04 LTS, ...[read more](#)
ami-04a81a99f5ec58529

Virtual server type (instance type)
t2.micro

Firewall (security group)
default

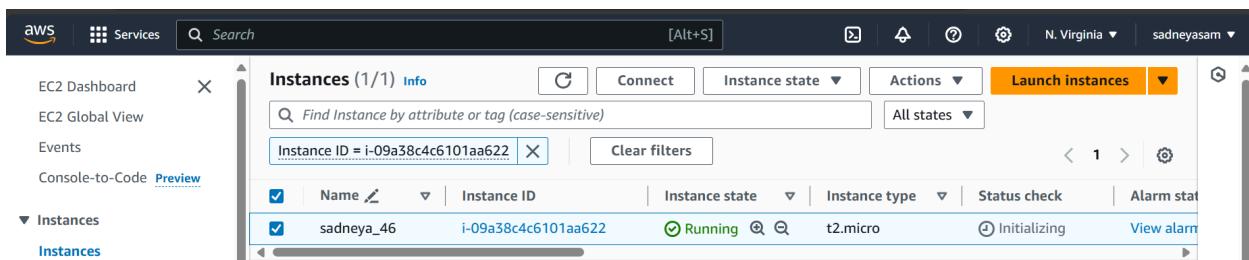
Storage (volumes)
1 volume(s) - 8 GiB

(i) **Free tier:** In your first year includes 750 hours of t2.micro (or t3.micro in the Regions in which t2.micro is unavailable) instance usage on free tier AMIs per month, 750 hours of public IPv4 [X](#)

8. The process was successful



9. Select instance



10. Instance summary

Instance summary for i-0b27a6f21d460ffe4 (sadneya_46) Info		
C Connect Instance state Actions		
Updated less than a minute ago		
Instance ID i-0b27a6f21d460ffe4 (sadneya_46)	Public IPv4 address 3.82.223.21 open address	Private IPv4 addresses 172.31.80.107
IPv6 address -	Instance state Running	Public IPv4 DNS ec2-3-82-223-21.compute-1.amazonaws.com open address
Hostname type IP name: ip-172-31-80-107.ec2.internal	Private IP DNS name (IPv4 only) ip-172-31-80-107.ec2.internal	Elastic IP addresses -
Answer private resource DNS name IPv4 (A)	Instance type t2.micro	AWS Compute Optimizer finding Opt-in to AWS Compute Optimizer for recommendations.
Auto-assigned IP address 3.82.223.21 [Public IP]	VPC ID vpc-051bba342b3626898	Learn more
IAM Role -	Subnet ID subnet-058dd8c2c4d107cb2	Auto Scaling Group name -

11. Connect to instance

The screenshot shows the 'Connect to instance' page in the AWS Management Console. The navigation path is EC2 > Instances > i-00f3bcf72585e5973 > Connect to instance. The main title is 'Connect to instance' with an 'Info' link. Below it, a message says 'Connect to your instance i-00f3bcf72585e5973 (sadneya_46) using any of these options'. There are four tabs at the top: 'EC2 Instance Connect' (selected), 'Session Manager', 'SSH client', and 'EC2 serial console'. A yellow warning box contains the text: 'Port 22 (SSH) is open to all IPv4 addresses' followed by a detailed description about inbound rules and security groups. Below the tabs, there are two sections: 'Instance ID' showing 'i-00f3bcf72585e5973 (sadneya_46)' and 'Connection Type' with two options: 'Connect using EC2 Instance Connect' (selected) and 'Connect using EC2 Instance Connect Endpoint'.

12. Run sudo apt update in the ssh terminal

```
ubuntu@ip-172-31-86-40:~$ sudo apt update
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 [265 kB]
Get:7 http://security.ubuntu.com/ubuntu noble-security/main Translation-en [63.1 kB]
Get:8 http://security.ubuntu.com/ubuntu noble-security/main amd64 c-n-f Metadata [3632 B]
Get:9 http://security.ubuntu.com/ubuntu noble-security/universe amd64 Packages [246 kB]
Get:10 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble/universe Translation-en [5982 kB]
Get:11 http://security.ubuntu.com/ubuntu noble-security/universe Translation-en [106 kB]
Get:12 http://security.ubuntu.com/ubuntu noble-security/universe amd64 Components [8632 B]
Get:13 http://security.ubuntu.com/ubuntu noble-security/universe amd64 c-n-f Metadata [9164 B]
Get:14 http://security.ubuntu.com/ubuntu noble-security/restricted amd64 Packages [208 kB]
Get:15 http://security.ubuntu.com/ubuntu noble-security/restricted Translation-en [40.7 kB]
Get:16 http://security.ubuntu.com/ubuntu noble-security/restricted amd64 c-n-f Metadata [420 B]
Get:17 http://security.ubuntu.com/ubuntu noble-security/multiverse amd64 Packages [10.6 kB]
Get:18 http://security.ubuntu.com/ubuntu noble-security/multiverse Translation-en [2808 B]
Get:19 http://security.ubuntu.com/ubuntu noble-security/multiverse amd64 Components [208 B]
Get:20 http://security.ubuntu.com/ubuntu noble-security/multiverse amd64 c-n-f Metadata [344 B]
Get:21 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble/universe amd64 Components [3871 kB]
Get:22 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble/universe amd64 c-n-f Metadata [301 kB]
Get:23 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble/multiverse amd64 Packages [269 kB]
Get:24 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble/multiverse Translation-en [118 kB]
```

13. Run top

```
ubuntu@ip-172-31-86-40:~$ top
```

i-00f3bcf72585e5973 (sadneya_46)

Public IPs: 44.203.74.158 Private IPs: 172.31.86.40

top - 04:37:13 up 10 min, 1 user, load average: 0.17, 0.09, 0.07										
Tasks: 105 total, 1 running, 104 sleeping, 0 stopped, 0 zombie										
%Cpu(s): 0.0 us, 0.0 sy, 0.0 ni, 100.0 id, 0.0 wa, 0.0 hi, 0.0 si, 0.0 st										
MiB Mem : 957.4 total, 227.6 free, 354.7 used, 530.5 buff/cache										
MiB Swap: 0.0 total, 0.0 free, 0.0 used. 602.8 avail Mem										
PID	USER	PR	NI	VIRT	RES	SHR	S	%CPU	%MEM	TIME+ COMMAND
1	root	20	0	22520	13536	9568	S	0.0	1.4	0:03.80 systemd
2	root	20	0	0	0	0	S	0.0	0.0	0:00.00 kthreadd
3	root	20	0	0	0	0	S	0.0	0.0	0:00.00 pool_workqueue_release
4	root	0	-20	0	0	0	I	0.0	0.0	0:00.00 kworker/R-rcu_g
5	root	0	-20	0	0	0	I	0.0	0.0	0:00.00 kworker/R-rcu_p
6	root	0	-20	0	0	0	I	0.0	0.0	0:00.00 kworker/R-slub_

14. Run history

```
ubuntu@ip-172-31-86-40:~$ history
1 sudo -l
2 apt update
3 sudo apt update
4 top
5 history
```

15. run vmstat

```
ubuntu@ip-172-31-86-40:~$ vmstat
procs -----memory----- ---swap-- ----io---- -system-- -----cpu-----
 r b    swpd   free   buff   cache   si   so    bi    bo   in    cs us sy id wa st gu
 2 0      0 233040 18712 524596     0     0    379   806  189    1  3  1 94  1  1  0
```

16. Run df

```
ubuntu@ip-172-31-86-40:~$ df
Filesystem      1K-blocks      Used Available Use% Mounted on
/dev/root        7034376  1814472    5203520  26% /
tmpfs            490208        0    490208  0% /dev/shm
tmpfs            196084       872    195212  1% /run
tmpfs             5120        0      5120  0% /run/lock
/dev/xvda16      901520    76972    761420  10% /boot
/dev/xvda15     106832     6246    100586  6% /boot/efi
tmpfs            98040       12     98028  1% /run/user/1000
```

17. Run df -kh, whatis df, df –help, uname -a

```
ubuntu@ip-172-31-86-40:~$ df -kh
Filesystem      Size  Used Avail Use% Mounted on
/dev/root       6.8G  1.8G  5.0G  26% /
tmpfs           479M    0   479M   0% /dev/shm
tmpfs           192M  872K  191M   1% /run
tmpfs           5.0M    0   5.0M   0% /run/lock
/dev/xvda16     881M   76M  744M  10% /boot
/dev/xvda15     105M   6.1M   99M   6% /boot/efi
tmpfs            96M   12K   96M   1% /run/user/1000
```

```
ubuntu@ip-172-31-86-40:~$ whatis df
df (1)          - report file system space usage
```

```
ubuntu@ip-172-31-86-40:~$ df --help
Usage: df [OPTION]... [FILE]...
Show information about the file system on which each FILE resides,
or all file systems by default.

Mandatory arguments to long options are mandatory for short options too.
  -a, --all            include pseudo, duplicate, inaccessible file systems
  -B, --block-size=SIZE scale sizes by SIZE before printing them; e.g.,
                        '-BM' prints sizes in units of 1,048,576 bytes;
                        see SIZE format below
  -h, --human-readable print sizes in powers of 1024 (e.g., 1023M)
  -H, --si              print sizes in powers of 1000 (e.g., 1.1G)
  -i, --inodes          list inode information instead of block usage
  -k                  like --block-size=1K
  -l, --local           limit listing to local file systems
  --no-sync            do not invoke sync before getting usage info (default)
  --output[=FIELD_LIST] use the output format defined by FIELD_LIST,
                        or print all fields if FIELD_LIST is omitted.
  -P, --portability    use the POSIX output format
  --sync               invoke sync before getting usage info
  --total              elide all entries insignificant to available space,
                        and produce a grand total
  -t, --type=TYPE      limit listing to file systems of type TYPE
  -T, --print-type     print file system type
  -x, --exclude-type=TYPE limit listing to file systems not of type TYPE
  -v                  (ignored)
```



18. Create and do file operations

```
ubuntu@ip-172-31-80-107:~$ mkdir test
ubuntu@ip-172-31-80-107:~$ ls
test
ubuntu@ip-172-31-80-107:~$ cd test
ubuntu@ip-172-31-80-107:~/test$ touch file1
ubuntu@ip-172-31-80-107:~/test$ ls
file1
ubuntu@ip-172-31-80-107:~/test$ touch file2 file3
ubuntu@ip-172-31-80-107:~/test$ ls
file1 file2 file3
ubuntu@ip-172-31-80-107:~/test$ rm file*
ubuntu@ip-172-31-80-107:~/test$ ls
ubuntu@ip-172-31-80-107:~/test$ cd ..
ubuntu@ip-172-31-80-107:~$ rmdir test
ubuntu@ip-172-31-80-107:~$ cd ..
ubuntu@ip-172-31-80-107:/home$ ls
ubuntu
ubuntu@ip-172-31-80-107:/home$ cd ubuntu
ubuntu@ip-172-31-80-107:~$ mkdir test1 test2 test3
ubuntu@ip-172-31-80-107:~$ ls
test1 test2 test3
```

EXPERIMENT 1B

Abcd

The screenshot shows two browser windows. The top window is the AWS Cloud9 search results page, displaying a list of services and features related to 'cloud 9'. The bottom window is the AWS Cloud9 control home page, showing the Cloud9 IDE interface.

AWS Cloud9 Search Results (Top Window):

- Services (58):**
 - Cloud9
 - Amazon CodeCatalyst
 - AWS Cloud Map
 - AWS Deadline Cloud
 - Cloud WAN
 - Namespaces
- Features (89):**
 - Cloud WAN
 - VPC feature
 - Namespaces
 - AWS Cloud Map feature

AWS Cloud9 Control Home Page (Bottom Window):

The page title is "AWS Cloud9" and it describes it as "A cloud IDE for writing, running, and debugging code". A prominent button says "Create environment". Below the main heading, there's a section titled "How it works" which states: "Create an AWS Cloud9 development environment on a new Amazon EC2 instance or connect it to your own Linux server through SSH. Once you've created an AWS Cloud9 environment, you will have immediate access to a rich code editor". There are also "Getting started" links for "Before you start" and "Create an environment".

AWS Services Search [Alt+S] N. Virginia v ocblabs/user3404102=SAMANT_SADNEYA_SADANAND @ 4250-0137-5268 ▾

AWS Cloud9 > Environments > Create environment

Create environment Info

Details

Name Limit of 60 characters, alphanumeric, and unique per user.

Description - *optional* Limit 200 characters.

Environment type Info
Determines what the Cloud9 IDE will run on.

New EC2 instance
Cloud9 creates an EC2 instance in your account. The configuration of your EC2 instance cannot be changed by Cloud9 after creation.

Existing compute
You have an existing instance or server that you'd like to use.

New EC2 instance

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

New EC2 instance

Instance type Info
The memory and CPU of the EC2 instance that will be created for Cloud9 to run on.

t2.micro (1 GiB RAM + 1 vCPU)
Free-tier eligible. Ideal for educational users and exploration.

t3.small (2 GiB RAM + 2 vCPU)
Recommended for small web projects.

m5.large (8 GiB RAM + 2 vCPU)
Recommended for production and most general-purpose development.

Additional instance types
Explore additional instances to fit your need.

Platform Info
This will be installed on your EC2 instance. We recommend Amazon Linux 2023.

Amazon Linux 2023

Timeout
How long Cloud9 can be inactive (no user input) before auto-hibernating. This helps prevent unnecessary charges.

30 minutes

aws Services Search [Alt+S] N. Virginia v vclabs/user3404102=SAMANT_SADNEYA_SADANAND @ 4250-0137-5268 ▾

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.

No tags associated with the resource.

Add new tag

You can add up to 50 more tags.

i The following IAM resources will be created in your account

▼ VPC settings Info

Amazon Virtual Private Cloud (VPC)
The VPC that your environment will access. To allow the AWS Cloud9 environment to connect to its EC2 instance, attach an internet gateway (IGW) to your VPC. [Create new VPC](#)

vpc-051bba342b3626898

Name -

Subnet
Used to setup your VPC configuration. To use a private subnet, select AWS Systems Manager (SSM) as the connection type. [Create new subnet](#)

No preference

Uses default subnet in any Availability Zone

▼ 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.

No tags associated with the resource.

Add new tag

You can add up to 50 more tags.

i The following IAM resources will be created in your account

- **AWSServiceRoleForAWSCloud9** - AWS Cloud9 creates a service-linked role for you. This allows AWS Cloud9 to call other AWS services on your behalf. You can delete the role from the AWS IAM console once you no longer have any AWS Cloud9 environments. [Learn more](#)

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

Amazon Virtual Private Cloud (VPC)
The VPC that your environment will access. To allow the AWS Cloud9 environment to connect to its EC2 instance, attach an internet gateway (IGW) to your VPC. [Create new VPC](#)

vpc-051bba342b3626898
Name -

Subnet
Used to setup your VPC configuration. To use a private subnet, select AWS Systems Manager (SSM) as the connection type. [Create new subnet](#)

No preference
Uses default subnet in any Availability Zone

AWS Services Search [Alt+S] N. Virginia voclabs/user3404102=SAMANT_SADNEYA_SADANAND @ 4250-0137-5268 ▾

AWS Cloud9 Creating sadneya_46. This can take several minutes. While you wait, see [Best practices for using AWS Cloud9](#)

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

AWS Cloud9 > Environments

Environments (1)						
		Delete	View details	Open in Cloud9	Create environment	
My environments						
Name	Cloud9 IDE	Environment type	Connection	Permission	Owner ARN	
<input type="radio"/> sadneya_46	Open	EC2 instance	Secure Shell (SSH)	Owner	arn:aws:sts::42500137526:role/voclabs/user3404102=Sadneya_Sadanand	

<https://us-east-1.console.aws.amazon.com/iam/home?region=us-east-1#/home>

Identity and Access Management (IAM)

IAM Dashboard

IAM resources

User groups	Users	Roles	Policies	Identity providers
0	0	20	4	0

What's new

Updates for features in IAM

- AWS IAM Access Analyzer now offers policy checks for public and critical resource access. 1 month ago
- AWS IAM Access Analyzer now offers recommendations to refine unused access. 1 month ago
- AWS Launches Console-based Bulk Policy Migration for Billing and Cost Management Console Access. 2 months ago
- IAM Roles Anywhere now supports modifying the mapping of certificate attributes. 4 months ago

[View all](#)

[more](#)

Dashboard

Access management

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

Access reports

- Access Analyzer
- External access
- Unused access
- Analyzer settings

<https://us-east-1.console.aws.amazon.com/iam/home?region=us-east-1#/users>

Identity and Access Management (IAM)

IAM > Users

Users (0) Info

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

User name	Path	Group:	Last activity	MFA
No resources to display				

[Create user](#)

Dashboard

Access management

Users

- Roles
- Policies
- Identity providers
- Account settings

Access reports

- Access Analyzer
- External access
- Unused access
- Analyzer settings

The screenshot shows the AWS IAM search results page. The search term 'IAM' has been entered into the search bar. The results are categorized into 'Services' and 'Features'.

Services

- IAM: Manage access to AWS resources
- IAM Identity Center: Manage workforce user access to multiple AWS accounts and cloud applications
- Resource Access Manager: Share AWS resources with other accounts or AWS Organizations
- AWS App Mesh: Easily monitor and control microservices

Features

- Groups: IAM feature

On the right side, there is a sidebar titled 'Storage Lens dashboard' with a creation date of August 1, 2024, at 9:21:38 (UTC+05:30). Below it, another entry has a creation date of July 24, 2024, at 21:35:15 (UTC+05:30).

The screenshot shows the 'Specify user details' step of the 'Create user' wizard. The user name 'sadneya_46' has been entered into the 'User name' field. A note below the field states: 'The user name can have up to 64 characters. Valid characters: A-Z, a-z, 0-9, and + = , . @ _ - (hyphen)'.

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.

Info If you are creating programmatic access through access keys or service-specific credentials for AWS CodeCommit or Amazon Keyspaces, you can generate them after you create this IAM user. [Learn more](#)

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

Screenshot of the AWS IAM User Creation Step 2: Set permissions page.

User details

User name: sadneya_46

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 Keyspaces, 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
- Must include at least three of the following mix of character types: uppercase letters (A-Z), lowercase letters (a-z), numbers (0-9), and symbols ! @ # \$ % ^ & () _ * - (hyphen) = { } { }

Screenshot of the AWS IAM User Creation Step 3: Set permissions page.

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 permissions by job function, AWS service access, or custom permissions. [Learn more](#)

Set permissions boundary - optional

Cancel Previous Next

Screenshot of the AWS IAM User Creation Step 4: Create user group page.

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.
WebAppUser

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

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.

Specify user details

User name: sadneya_46

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.

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
 - Must include at least three of the following mix of character types: uppercase letters (A-Z), lowercase letters (a-z), numbers (0-9), and symbols ! @ # \$ % ^ & * () _ + - (hyphen) = [{ }] !

Show password

Users must create a new password at next sign-in - Recommended
Users automatically get the [IAMUserChangePassword](#) policy to allow them to change their own password.

If you are creating programmatic access through access keys or service-specific credentials for AWS CodeCommit or Amazon Keyspaces, you can generate them after you create this IAM user. [Learn more](#)

Cancel **Next**

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](#)

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.

Permissions policies (1/1227)

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

Policy name	Type	Attached entities
AWSCloud9Administrator	AWS managed	0
AWSCloud9EnvironmentMember	AWS managed	0
AWScloud9ServiceRolePolicy	AWS managed	1
AWScloud9SSMInstanceProfile	AWS managed	0
AWSCloud9User	AWS managed	0

https://us-east-1.console.aws.amazon.com/iam/home?region=us-east-1#/users/create

WebAppUsers user group created.

Step 2 Set permissions

Step 3 Review and create

Step 4 Retrieve password

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.

User groups (1)

Group name	Users	Attached policies	Created
WebAppUsers	0	-	2024-08-02 (Now)

▶ Set permissions boundary - optional

Cancel Previous Next

https://us-east-1.console.aws.amazon.com/iam/home?region=us-east-1#/users/create

WebAppUsers user group created.

Step 2 Set permissions

Step 3 Review and create

Step 4 Retrieve password

User details

User name sadnya_46	Console password type Custom password	Require password reset Yes
--	--	-------------------------------

Permissions summary

Name	Type	Used as
IAMUserChangePassword	AWS managed	Permissions policy

Tags - optional

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

No tags associated with the resource.

Add new tag

You can add up to 50 more tags.

Cancel Previous Create user

User created successfully

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

IAM > Users > Create user

Step 1 Specify user details

Step 2 Set permissions

Step 3 Review and create

Step 4 Retrieve password

Retrieve password

You can view and download the user's password below or email users instructions for signing in to the AWS Management Console. This is the only time you can view and download this password.

Console sign-in details

Email sign-in instructions

Console sign-in URL
https://851725480355.signin.aws.amazon.com/console

User name
sadneya_46

Console password
***** Show

Cancel Download .csv file Return to users list

Identity and Access Management (IAM)

IAM > Users > sadneya_46

sadneya_46 Info Delete

Search IAM

Dashboard

Access management

User groups

Users

Roles

Policies

Identity providers

Account settings

Access reports

Access Analyzer

External access

Unused access

Analyzer settings

Credential report

Organization activity

Service control policies

Summary

ARN
arn:aws:iam::851725480355:user/sadneya_46

Console access
Enabled without MFA

Access key 1
Create access key

Created
August 02, 2024, 19:28 (UTC+05:30)

Last console sign-in
Never

Permissions Groups Tags Security credentials Access Advisor

Permissions policies (1)

Permissions are defined by policies attached to the user directly or through groups.

Filter by Type

Search All types

Policy name IAMUserChangePassword Type AWS managed Attached via Directly

Screenshot of the AWS IAM User Groups page:

The URL is <https://us-east-1.console.aws.amazon.com/iam/home?region=us-east-1#/groups>.

The left sidebar shows the IAM navigation menu with "User groups" selected.

The main content area displays "User groups (1) Info". It shows one group named "WebAppUsers" with the following details:

Group name	Users	Permissions	Creation time
WebAppUsers	0	Not defined	4 minutes ago

Screenshot of the AWS IAM User Group Details page for "WebAppUsers":

The URL is <https://us-east-1.console.aws.amazon.com/iam/home?region=us-east-1#/groups/details/WebAppUser...>.

The left sidebar shows the IAM navigation menu with "User groups" selected.

The main content area displays the "WebAppUsers" info page under "User groups".

Summary section:

User group name	Creation time	ARN
WebAppUsers	August 02, 2024, 19:25 (UTC+05:30)	arn:aws:iam::851725480355:group/WebAppUsers

Users tab (selected):

Users in this group (0)

User name	Groups	Last activity	Creation time
No resources to display			

Experiment 2

1. Go to the amazon console page

The screenshot shows the AWS Console Home page. On the left, there's a sidebar with 'Recently visited' services: Billing and Cost Management, S3, IAM, and EC2. Below this is a 'View all services' link. On the right, there's a section titled 'Applications (0)' with a 'Create application' button. A message says 'No applications' and 'Get started by creating an application.' There's also a 'Go to myApplications' link.

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

The screenshot shows the AWS Services menu. It lists several services: ElastiCache (In-Memory Cache), Elastic Transcoder (Easy-to-Use Scalable Media Transcoding), Elastic Beanstalk (Run and Manage Web Apps), and Elastic Container Service (Highly secure, reliable, and scalable way to run containers). The 'Elastic Beanstalk' service is highlighted.

3. Enter your website and its information

The screenshot shows the 'Application information' form. The 'Application name' field is filled with 'website123'. Below it, a note says 'Maximum length of 100 characters.' There's also a section for 'Application tags (optional)' which is currently empty.

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

Platform branch

Platform version

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-ec2-role

View permission details

Cancel Skip to review Previous Next

4. Your environment is created

Environment overview

Health

Ok

Environment ID

e-vw23gecgs

Domain

Sadneya123-env.eba-gw7emmur.us-east-1.elasticbeanstalk.com

Application name

sadneya123

Step 2: Get a copy of your sample code

Create a new fork

A *fork* is a copy of a repository. Forking a repository allows you to freely experiment with changes without affecting the original project. [View existing forks](#)

Required fields are marked with an asterisk (*).

Owner *

sadneya145

Repository name *

aws-codepipeline-s3-code

aws-codepipeline-s3-codedeploy-linux-2.0 is available.

By default, forks are named the same as their upstream repository. You can customize the name to distinguish it further.

Description (optional)

Use this sample when creating a simple pipeline in AWS CodePipeline while following the Simple Pipeline Walkthrough

Copy the `master` branch only

Contribute back to [imosharma/aws-codepipeline-s3-codedeploy-linux-2.0](#) by adding your own branch. [Learn more](#).

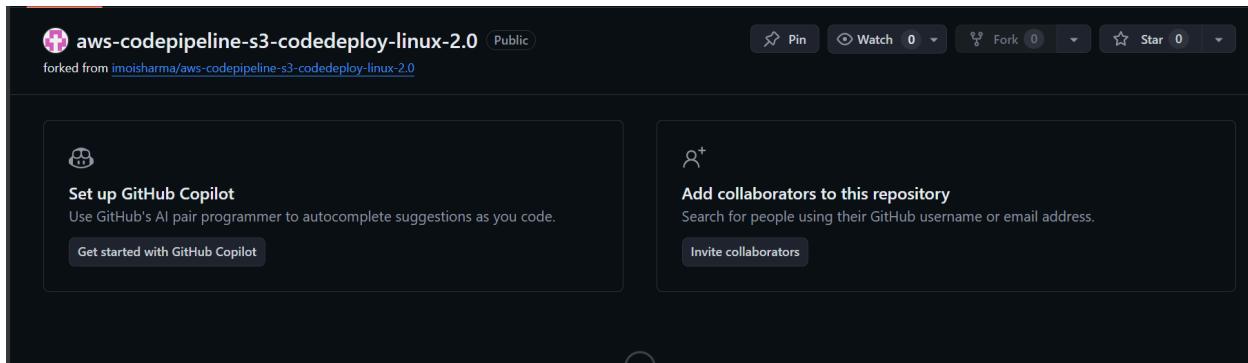
You are creating a fork in your personal account.

Create fork

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

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

Go to the repository shared above and simply fork it.



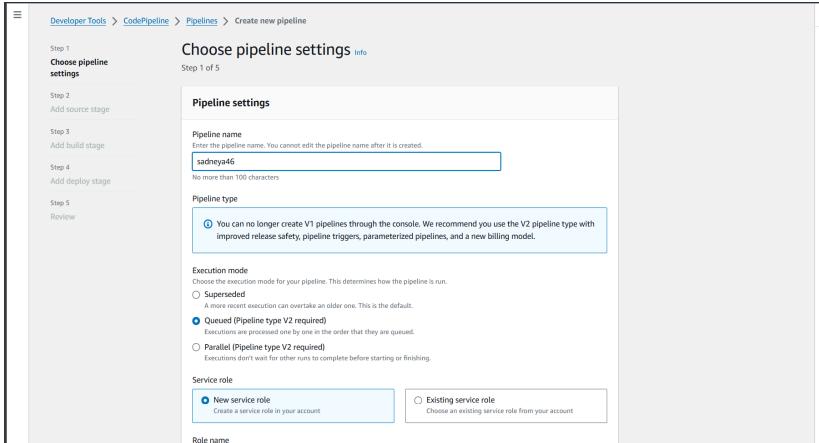
Step 3: Creating a CodePipeline

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.

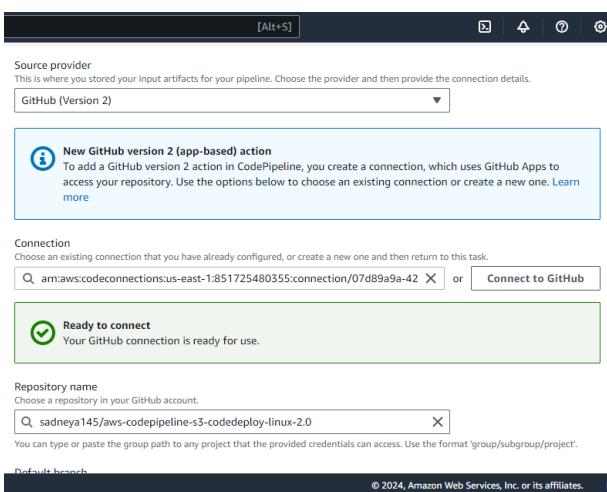
1. Go to AWS Developer Tools -> CodePipeline and create a new Pipeline. Fill in the initial settings first.

The screenshot shows the AWS Lambda console. A search bar at the top left contains the text 'codeArtifact'. The search results for 'code' are displayed under the 'Services' section, with 'Amazon Q Developer (Including Amazon CodeWhisperer)' at the top, followed by 'CodeCommit', 'CodePipeline', and others. On the right side of the screen, the 'CodePipeline' service details are shown, including its ARN, description ('Release Software using Continuous Delivery'), and various configuration tabs like 'General', 'Triggers', 'Actions', 'Metrics', and 'Logs'.

The screenshot shows the AWS CodePipeline console. The left sidebar navigation menu includes 'Developer Tools', 'CodePipeline', 'Source', 'Artifacts', 'Build', 'Deploy', 'Pipeline', 'Getting started', 'Pipelines', and 'Settings'. The main content area shows the 'Pipelines' list with a single entry: 'Pipelines' (Info). A prominent orange 'Create pipeline' button is located at the top right of the list table. The table columns include 'Name', 'Latest execution status', 'Latest source revisions', 'Latest execution started', and 'Most recent executions'. A message at the bottom of the table states 'No results' and 'There are no results to display.'



2. In the source stage, choose GitHub v1 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.



3. Then, simply choose this forked repository and the branch which you will be able to find in the search box. After that, click Continue and skip the build stage. Proceed to the Deployment stage.

Step 4: Deployment

1. Choose Beanstalk as the Deploy Provider, same region as the Bucket and Beanstalk, name and environment name. Click Next, Review and create the pipeline.

You cannot skip this stage
Pipelines must have at least two stages. Your second stage must be either a build or deployment stage. Choose a provider for either the build stage or deployment stage.

Deploy

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

Region
US East (N. Virginia)

Input artifacts
Choose an input artifact for this action. Learn more

Application name
sadneya123

Environment name
Sadneya123-env

Configure automatic rollback on stage failure

Cancel Previous Next

Developer Tools > CodePipeline > Pipelines > Create new pipeline

Step 1 Choose pipeline settings

Step 2 Add source stage

Step 3 Add build stage

Step 4 Add deploy stage

Step 5 Review

Step 1: Choose pipeline settings

Pipeline settings

Pipeline name: sadneya_46
Pipeline type: V2
Execution mode: QUEUED
Artifact location: codepipeline-us-east-1-204862929919
Service role name: AWSCodePipelineServiceRole-us-east-1-sadneya_46

Variables

Name	Default value	Description
No variables		

No variables defined at the pipeline level in this pipeline.

Step 2: Add source stage

Source action provider

Source action provider: GitHub (Version 2)
OutputArtifactFormat: CODE_ZIP
DetectChanges: false
ConnectionArn:

aws Services Search [Alt+S] N. Virginia sadneyasam

Step 3: Add build stage

Build action provider

Build stage

No build

Step 4: Add deploy stage

Deploy action provider

Deploy action provider

AWS Elastic Beanstalk

ApplicationName

sadneya123

EnvironmentName

Sadneya123-env

Configure automatic rollback on stage failure

Disabled

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.

ⓘ You can add additional sources and triggers by editing the pipeline after it is created.

aws Services Search [Alt+S] N. Virginia sadneyasam

Choose pipeline settings Step 1 of 5

Step 2 Add source stage

Step 3 Add build stage

Step 4 Add deploy stage

Step 5 Review

Pipeline settings

Pipeline name

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

pipeline

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.

Source code

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

2. Review all the settings and click on create pipeline

The screenshot shows the AWS CodePipeline 'Create new pipeline' process at Step 5 of 5, labeled 'Review'. On the left, a sidebar lists steps: Step 1 (Choose pipeline settings), Step 2 (Add source stage), Step 3 (Add build stage), Step 4 (Add deploy stage), and Step 5 (Review). The main area displays 'Step 1: Choose pipeline settings' with the following details:

- Pipeline name: sadneya_46
- Pipeline type: V2
- Execution mode: QUEUED
- Artifact location: A new Amazon S3 bucket will be created as the default artifact store for your pipeline.
- Service role name: AWSCodePipelineServiceRole-us-east-1-sadneya_46

Below this is a 'Variables' section with a table:

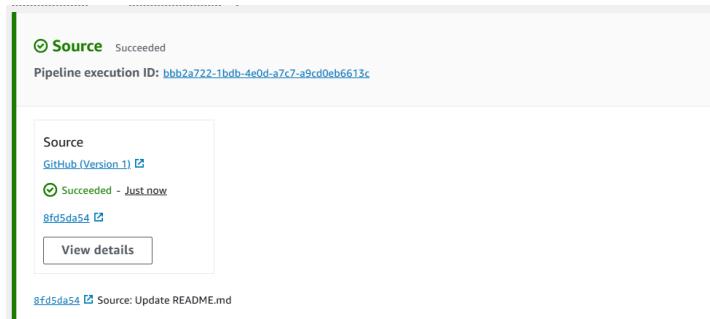
Name	Default value	Description
No variables		

The screenshot shows the continuation of the AWS CodePipeline pipeline creation process through Step 4:

- Step 2: Add source stage**: Shows a 'Source action provider' configuration with GitHub (Version 1) selected. Details include: PollForSourceChanges: false, Repo: aws-codepipeline-s3-codedeploy-linux-2.0, Owner: sadneya145, Branch: master.
- Step 3: Add build stage**: Shows a 'Build action provider' configuration with No build selected.
- Step 4: Add deploy stage**: Shows a 'Deploy action provider' configuration with AWS Elastic Beanstalk selected. Details include: ApplicationName: sadneya_46, EnvironmentName: Sadneya46-env, Configure automatic rollback on stage failure: Disabled.

At the bottom, there are 'Cancel', 'Previous', and a prominent orange 'Create pipeline' button.

3.in the end, you can see that the pipeline has been deployed successfully



Experiment No. 3

Objective: To understand the Kubernetes Cluster Architecture and to set up a Kubernetes Cluster on Linux Machines/Cloud.

Procedure:

1. Instance Setup:

- o Create three EC2 instances using Amazon Linux as the operating system.
- o Ensure that SSH traffic is allowed from any source.
- o For optimal performance, choose an instance type of at least t2.medium, as Kubernetes recommends a minimum of 2 vCPUs.

	Name	Instance ID	Instance state	Instance type
	kube-master	i-00aa79ac09d7462c0	Running	t2.medium
	kube-worker1	i-0bab86cd3fbfcba0a	Running	t2.medium
	kube-worker2	i-00dcfd302ffd80dda	Running	t2.medium

2. SSH Access:

- o SSH into each of the three machines using separate terminal windows: `ssh -i <keyname>.pem ubuntu@<public_ip_address>`

```
quantum@machine ~ ~/Downloads ssh -i "ec-2-ubuntu.pem" ec2-user@ec2-3-88-111-183.compute-1.amazonaws.com
The authenticity of host 'ec2-3-88-111-183.compute-1.amazonaws.com (3.88.111.183)' can't be established.
ED25519 key fingerprint is SHA256:pQu+xS9foYbY3de1twjZcVVA0zmGwGv6PHmVruF/Q1s.
This key is not known by any other names.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added 'ec2-3-88-111-183.compute-1.amazonaws.com' (ED25519) to the list of known hosts.
      #_
      ~\_\#\#\#_          Amazon Linux 2023
      ~~ \#\#\#\#\\
      ~~   \#\#\#
      ~~     \#/ ___  https://aws.amazon.com/linux/amazon-linux-2023
      ~~     V~' '-'>
      ~~     /
      ~~...- .-
      ~~ / _/
      ~~ /m/'
```

3. Docker Installation and Configuration:

- o On all three machines, install Docker with the command: `sudo yum install docker -y`
- o Configure Docker to use `systemd` as the cgroup driver by creating and editing the `daemon.json` file:
 - Change directory to `/etc/docker`

- Use the command `cat <<EOF | sudo tee /etc/docker/daemon.json` followed by the JSON configuration details and end with EOF
- Enable and restart Docker: `sudo systemctl enable docker sudo systemctl daemon-reload sudo systemctl restart docker docker -v`

```
[ec2-user@ip-172-31-92-18 ~]$ sudo yum install docker -y
Last metadata expiration check: 0:09:56 ago on Wed Sep 11 15:19:39 2024.
Dependencies resolved.
```

Package	Architecture
Installing:	
<code>docker</code>	<code>x86_64</code>
Installing dependencies:	
<code>containerd</code>	<code>x86_64</code>
<code>iptables-libs</code>	<code>x86_64</code>
<code>iptables-nft</code>	<code>x86_64</code>
<code>libcgroup</code>	<code>x86_64</code>
<code>libnetfilter_conntrack</code>	<code>x86_64</code>
<code>libnftnetlink</code>	<code>x86_64</code>
<code>libnftnl</code>	<code>x86_64</code>
<code>pigz</code>	<code>x86_64</code>
<code>runc</code>	<code>x86_64</code>

Transaction Summary

4. Kubernetes Installation:

- Disable SELinux before configuring kubelet: `sudo setenforce 0 sudo sed -i 's/^SELINUX=enforcing$/SELINUX=permissive/' /etc/selinux/config`
- Add the Kubernetes repository and install Kubernetes components:
 - Use the command `cat <<EOF | sudo tee /etc/yum.repos.d/kubernetes.repo` followed by the repository configuration details and end with EOF `sudo yum update sudo yum install -y kubelet kubeadm kubectl --disableexcludes=kubernetes`

- o Configure networking for bridging: `sudo swapoff -a echo "net.bridge.bridge-nf-call-iptables=1" | sudo tee -a /etc/sysctl.conf sudo sysctl -p`

```
[ec2-user@ip-172-31-81-63 docker]$ sudo systemctl enable docker
sudo systemctl daemon-reload
sudo systemctl restart docker
docker -v

Created symlink /etc/systemd/system/multi-user.target.wants/docker.service → /usr/lib/systemd/system/docker.service.
Docker version 25.0.5, build 5dc9bcc
[ec2-user@ip-172-31-81-63 docker]$
```

```
[ec2-user@ip-172-31-81-63 docker]$ sudo setenforce 0
[ec2-user@ip-172-31-81-63 docker]$ sudo sed -i 's/^SELINUX=enforcing$/SELINUX=permissive/' /etc/selinux/config
```

```
[ec2-user@ip-172-31-81-63 docker]$ sudo yum install -y kubelet kubeadm kubectl --disableexcludes=kubernetes
Last metadata expiration check: 0:01:34 ago on Wed Sep 11 15:39:05 2024.
Dependencies resolved.
=====
Package           Architecture      Version
=====
Installing:
kubeadm          x86_64          1.30.4-150500.1.1
kubectl          x86_64          1.30.4-150500.1.1
kubelet          x86_64          1.30.4-150500.1.1
Installing dependencies:
conntrack-tools   x86_64          1.4.6-2.amzn2023.0.2
cri-tools         x86_64          1.30.1-150500.1.1
kubernetes-cni    x86_64          1.4.0-150500.1.1
libnetfilter_cthelper x86_64        1.0.0-21.amzn2023.0.2
libnetfilter_cttimeout x86_64        1.0.0-19.amzn2023.0.2
libnetfilter_queue x86_64          1.0.5-2.amzn2023.0.2
socat             x86_64          1.7.4.2-1.amzn2023.0.2
Transaction Summary
=====
Install 10 Packages
```

5. Master Node Setup:

- o Initialize the Kubernetes master node (perform only on the master machine): `sudo kubeadm init --pod-network-cidr=10.244.0.0/16 --ignore-preflight-errors=all`
- o After initialization, set up the Kubernetes configuration on the master node: `mkdir -p $HOME/.kube sudo cp -i /etc/kubernetes/admin.conf $HOME/.kube/config sudo chown $(id -u):$(id -g) $HOME/.kube/config`

- Save the generated join command from the output for worker nodes. This command is unique and specific to your cluster setup: `kubeadm join 172.31.91.120:6443 --token r8j60r.n1j6h0klbewvoka5 --discovery-token-ca-cert-hashsha256 :dd8426260174d673303aef17717f740772fcf7ee782245bc653eefcf4a13 05da7`

```
ubuntu@ip-172-31-28-117:~$ sudo kubeadm join 172.31.27.176:6443 --token ttay2x.n0squeukjai8sgfg3 \
--discovery-token-ca-cert-hash sha256:d6fc5fb7e984c83e2807780047fec6c4f2acfe9da9184ecc028d77157608fbb6

[preflight] Running pre-flight checks
[preflight] Reading configuration from the cluster...
[preflight] FYI: You can look at this config file with 'kubectl -n kube-system get cm kubeadm-config -o yaml'
[kubelet-start] Writing kubelet configuration to file "/var/lib/kubelet/config.yaml"
[kubelet-start] Writing kubelet environment file with flags to file "/var/lib/kubelet/kubeadm-flags.env"
[kubelet-start] Starting the kubelet
[kubelet-check] Waiting for a healthy kubelet at http://127.0.0.1:10248/healthz. This can take up to 4m0s
[kubelet-check] The kubelet is healthy after 501.396793ms
[kubelet-start] Waiting for the kubelet to perform the TLS Bootstrap

This node has joined the cluster:
* Certificate signing request was sent to apiserver and a response was received.
* The Kubelet was informed of the new secure connection details.

Run 'kubectl get nodes' on the control-plane to see this node join the cluster.
```

- Deploy the Flannel networking plugin to enable pod communication: `kubectl apply -f https://raw.githubusercontent.com/coreos/flannel/master/Documentation/kube-flannel.yml`
- Check the status of the pods to ensure they are running: `kubectl get pods --all-namespaces`

```
ubuntu@ip-172-31-27-176:~$ kubectl get nodes
NAME           STATUS    ROLES      AGE     VERSION
ip-172-31-18-135  NotReady <none>    88s    v1.31.1
ip-172-31-27-176  NotReady control-plane 10m    v1.31.1
ip-172-31-28-117  NotReady <none>    2m58s   v1.31.1
```

```
ubuntu@ip-172-31-27-176:~$ kubectl apply -f https://docs.projectcalico.org/manifests/calico.yaml
poddisruptionbudget.policy/calico-kube-controllers created
serviceaccount/calico-kube-controllers created
serviceaccount/calico-node created
configmap/calico-config created
customresourcedefinition.apiextensions.k8s.io/bgpconfigurations.crd.projectcalico.org created
customresourcedefinition.apiextensions.k8s.io/bgppeers.crd.projectcalico.org created
customresourcedefinition.apiextensions.k8s.io/blockaffinities.crd.projectcalico.org created
customresourcedefinition.apiextensions.k8s.io/caliconodestatuses.crd.projectcalico.org created
customresourcedefinition.apiextensions.k8s.io/clusterinformations.crd.projectcalico.org created
customresourcedefinition.apiextensions.k8s.io/felixconfigurations.crd.projectcalico.org created
customresourcedefinition.apiextensions.k8s.io/globalnetworkpolicies.crd.projectcalico.org created
customresourcedefinition.apiextensions.k8s.io/globalnetworksets.crd.projectcalico.org created
customresourcedefinition.apiextensions.k8s.io/hostendpoints.crd.projectcalico.org created
customresourcedefinition.apiextensions.k8s.io/ipamblocks.crd.projectcalico.org created
customresourcedefinition.apiextensions.k8s.io/ipamconfigs.crd.projectcalico.org created
customresourcedefinition.apiextensions.k8s.io/ipamhandles.crd.projectcalico.org created
customresourcedefinition.apiextensions.k8s.io/ippools.crd.projectcalico.org created
customresourcedefinition.apiextensions.k8s.io/ipservations.crd.projectcalico.org created
customresourcedefinition.apiextensions.k8s.io/kubecontrollersconfigurations.crd.projectcalico.org created
customresourcedefinition.apiextensions.k8s.io/networkpolicies.crd.projectcalico.org created
customresourcedefinition.apiextensions.k8s.io/networksets.crd.projectcalico.org created
clusterrole.rbac.authorization.k8s.io/calico-kube-controllers created
clusterrole.rbac.authorization.k8s.io/calico-node created
clusterrolebinding.rbac.authorization.k8s.io/calico-node created
daemonset.apps/calico-node created
deployment.apps/calico-kube-controllers created
```

sudo systemctl status kubelet

```
ubuntu@ip-172-31-27-176:~$ sudo systemctl status kubelet
● kubelet.service - kubelet: The Kubernetes Node Agent
   Loaded: loaded (/usr/lib/systemd/system/kubelet.service; enabled; preset: enabled)
   Drop-In: /usr/lib/systemd/system/kubelet.service.d
     └─10-kubeadm.conf
     Active: active (running) since Mon 2024-09-16 15:40:01 UTC; 11min ago
       Docs: https://kubernetes.io/docs/
 Main PID: 5989 (kubelet)
   Tasks: 10 (limit: 4676)
     Memory: 32.6M (peak: 33.2M)
        CPU: 10.785s
      CGroup: /system.slice/kubelet.service
             └─5989 /usr/bin/kubelet --bootstrap-kubeconfig=/etc/kubernetes/bootstrap-kubelet.conf --kubeconfig=/etc/kubernetes/kubelet.conf --config=/var/
```

Sep 16 15:51:29 ip-172-31-27-176 kubelet[5989]: I0916 15:51:29.497458 5989 reconciler_common.go:245] "operationExecutor.VerifyControllerAttachedVolume >
Sep 16 15:51:29 ip-172-31-27-176 kubelet[5989]: I0916 15:51:29.497516 5989 reconciler_common.go:245] "operationExecutor.VerifyControllerAttachedVolume >
Sep 16 15:51:29 ip-172-31-27-176 kubelet[5989]: I0916 15:51:29.497569 5989 reconciler_common.go:245] "operationExecutor.VerifyControllerAttachedVolume >
Sep 16 15:51:29 ip-172-31-27-176 kubelet[5989]: I0916 15:51:29.497620 5989 reconciler_common.go:245] "operationExecutor.VerifyControllerAttachedVolume >
Sep 16 15:51:29 ip-172-31-27-176 kubelet[5989]: I0916 15:51:29.497669 5989 reconciler_common.go:245] "operationExecutor.VerifyControllerAttachedVolume >
Sep 16 15:51:29 ip-172-31-27-176 kubelet[5989]: I0916 15:51:29.497711 5989 reconciler_common.go:245] "operationExecutor.VerifyControllerAttachedVolume >
Sep 16 15:51:31 ip-172-31-27-176 kubelet[5989]: E0916 15:51:31.605091 5989 kubelet.go:2902] "Container runtime network not ready" networkReady="NetworkR>
Sep 16 15:51:32 ip-172-31-27-176 kubelet[5989]: E0916 15:51:32.366237 5989 scope.go:117] "RemoveContainer" containerID="f44f06967c5b3e567e07841a7b4352ae>
Sep 16 15:51:36 ip-172-31-27-176 kubelet[5989]: E0916 15:51:36.686675 5989 kubelet.go:2902] "Container runtime network not ready" networkReady="NetworkR>
Sep 16 15:51:41 ip-172-31-27-176 kubelet[5989]: E0916 15:51:41.608404 5989 kubelet.go:2902] "Container runtime network not ready" networkReady="NetworkR>

Now Run command kubectl get nodes -o wide we can see Status is ready.

NAME	STATUS	ROLES	AGE	VERSION	INTERNAL-IP	EXTERNAL-IP	OS-IMAGE	KERNEL-VERSION	CONTAINER-RUNTIME
ip-172-31-18-135	Ready	<none>	6m19s	v1.31.1	172.31.18.135	<none>	Ubuntu 24.04 LTS	6.8.0-1012-aws	containerd://1.7.12
ip-172-31-27-176	Ready	control-plane	15m	v1.31.1	172.31.27.176	<none>	Ubuntu 24.04 LTS	6.8.0-1012-aws	containerd://1.7.12
ip-172-31-28-117	Ready	<none>	7m49s	v1.31.1	172.31.28.117	<none>	Ubuntu 24.04 LTS	6.8.0-1012-aws	containerd://1.7.12

6. Worker Node Setup:

- On each worker node, install the required package and configure kubelet: sudo yum install iproute-tc -y sudo systemctl enable kubelet sudo systemctl restart kubelet

- Join the worker nodes to the Kubernetes cluster using the join command from the master node: kubeadm join 172.31.91.120:6443 --token r8j60r.n1j6h0klbewvoka5\--discovery-token-ca-cert-hashsha256 :dd8426260174d673303aef17717f740772fcf7ee782245bc653eecf4a13 05da7

Node 1

```
ubuntu@ip-172-31-28-117:~$ sudo kubeadm join 172.31.27.176:6443 --token ttay2x.n0squeukjai8sgfg3 \
--discovery-token-ca-cert-hash sha256:d6fc5fb7e984c83e2807780047fec6c4f2acfe9da9184ecc028d77157608fbb6

[preflight] Running pre-flight checks
[preflight] Reading configuration from the cluster...
[preflight] FYI: You can look at this config file with 'kubectl -n kube-system get cm kubeadm-config -o yaml'
[kubelet-start] Writing kubelet configuration to file "/var/lib/kubelet/config.yaml"
[kubelet-start] Writing kubelet environment file with flags to file "/var/lib/kubelet/kubeadm-flags.env"
[kubelet-start] Starting the kubelet
[kubelet-check] Waiting for a healthy kubelet at http://127.0.0.1:10248/healthz. This can take up to 4m0s
[kubelet-check] The kubelet is healthy after 501.396793ms
[kubelet-start] Waiting for the kubelet to perform the TLS Bootstrap

This node has joined the cluster:
* Certificate signing request was sent to apiserver and a response was received.
* The Kubelet was informed of the new secure connection details.

Run 'kubectl get nodes' on the control-plane to see this node join the cluster.

ubuntu@ip-172-31-28-117:~$
```

Node 2

```
ubuntu@ip-172-31-18-135:~$ sudo kubeadm join 172.31.27.176:6443 --token ttay2x.n0squeukjai8sgfg3 \
--discovery-token-ca-cert-hash sha256:d6fc5fb7e984c83e2807780047fec6c4f2acfe9da9184ecc028d77157608fbb6

[preflight] Running pre-flight checks
[preflight] Reading configuration from the cluster...
[preflight] FYI: You can look at this config file with 'kubectl -n kube-system get cm kubeadm-config -o yaml'
[kubelet-start] Writing kubelet configuration to file "/var/lib/kubelet/config.yaml"
[kubelet-start] Writing kubelet environment file with flags to file "/var/lib/kubelet/kubeadm-flags.env"
[kubelet-start] Starting the kubelet
[kubelet-check] Waiting for a healthy kubelet at http://127.0.0.1:10248/healthz. This can take up to 4m0s
[kubelet-check] The kubelet is healthy after 1.001003808s
[kubelet-start] Waiting for the kubelet to perform the TLS Bootstrap

This node has joined the cluster:
* Certificate signing request was sent to apiserver and a response was received.
* The Kubelet was informed of the new secure connection details.

Run 'kubectl get nodes' on the control-plane to see this node join the cluster.

ubuntu@ip-172-31-18-135:~$
```

7. Verify Node Status:

- o On the master node, verify that the worker nodes have successfully joined the cluster by running: `watch kubectl get nodes`

Or run `kubectl get nodes`

```
ubuntu@ip-172-31-27-176:~$ kubectl get nodes
NAME           STATUS   ROLES      AGE     VERSION
ip-172-31-18-135 Ready    Node2      24m    v1.31.1
ip-172-31-27-176 Ready    control-plane 33m    v1.31.1
ip-172-31-28-117 Ready    Node1      25m    v1.31.1
ubuntu@ip-172-31-27-176:~$
```

Conclusion:

Setting up the Kubernetes cluster involved several challenges. Network configuration issues initially hindered the deployment of the Flannel plugin, requiring open ports and a functional Kubernetes API server. Disabling SELinux and adjusting firewall rules were essential for proper communication between

components. Worker nodes experienced difficulties with the kubelet service, which needed to be correctly configured and restarted. Additionally, accurate copying of the join command, including the token and discovery-token-ca-cert-hash, was crucial for integrating worker nodes into the cluster. These issues underscored the need for precise configuration and troubleshooting to achieve a stable Kubernetes setup.

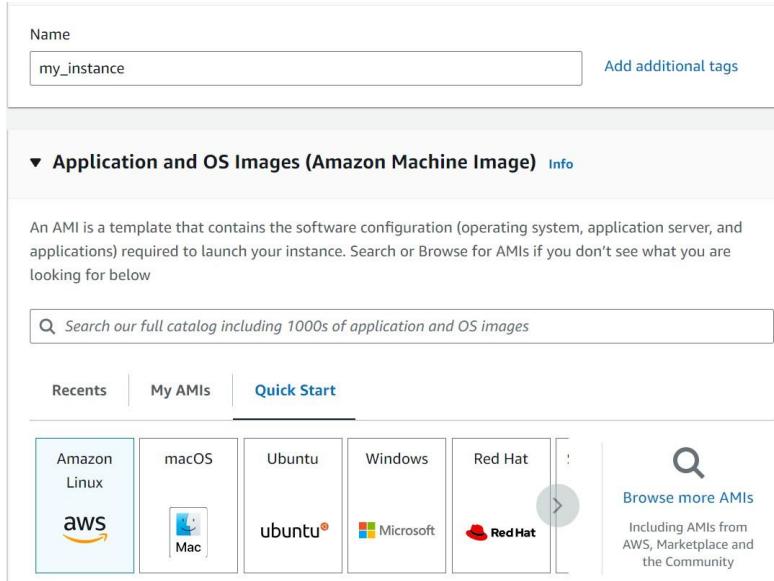
Experiment 4

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

Procedure:

1. Creation Of EC-2 instance

Create an EC2 AWS Linux instance on AWS .also edit the Security Group Inbound Rules to allow SSH. then select the t2.micro instance type



- Allow SSH traffic from Anywhere
Helps you connect to your instance
- 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

- Thus Kuber named -instance gets created.Then click on Id of that instance then click on connect button you will se this.

Connect to instance Info

Connect to your instance i-0f68279e506401ef2 (insty) using any of these options

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

Instance ID
i-0f68279e506401ef2 (insty)

1. Open an SSH client.
2. Locate your private key file. The key used to launch this instance is key2.pem
3. Run this command, if necessary, to ensure your key is not publicly viewable.
chmod 400 "key2.pem"
4. Connect to your instance using its Public DNS:
ec2-54-82-44-168.compute-1.amazonaws.com

Example:
ssh -i "key2.pem" ec2-user@ec2-54-82-44-168.compute-1.amazonaws.com

- Then go into SSH client where you will get this command

Chmod 400 “keyname.pem”

ssh -i <keyname>.pem ubuntu@<public_ip_address> copy it and then connect it and run the following command for establishing connection.(I have entered this command on git bash where i entered in downloads where server.pem is stored then as the key is not accessible hence we need to change its mode using chmod 400 “key name.pem”. Then use the given command for making connections).

```
Anshi@anshi MINGW64 ~
$ cd Downloads

Anshi@anshi MINGW64 ~/Downloads
$ chmod 400 "key2.pem"

Anshi@anshi MINGW64 ~/Downloads
$ ec2-3-85-239-227.compute-1.amazonaws.com
bash: ec2-3-85-239-227.compute-1.amazonaws.com: command not found

Anshi@anshi MINGW64 ~/Downloads
$ ssh -i "key2.pem" ec2-user@ec2-3-85-239-227.compute-1.amazonaws.com
The authenticity of host 'ec2-3-85-239-227.compute-1.amazonaws.com (3.85.239.227)' can't be established.
ED25519 key fingerprint is SHA256:3ytsjVzbzSc5N7KSAwq0IAh/LRz+zWwqk1lf4gWKjfY.
This key is not known by any other names.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added 'ec2-3-85-239-227.compute-1.amazonaws.com' (ED25519)
to the list of known hosts.

      #
      #####
      ##### \
      \###|   Amazon Linux 2023
      \|/ \
      V~' '-->
      ~~
      ~~-.-
      /--/
      /m/ ,-
```

2. Installation of Docker

- For installation of Docker into the machines run the following command: sudo yum install docker -y

```
[[email protected] ~]$ sudo yum install docker -y
Last metadata expiration check: 0:05:13 ago on Fri Sep 13 13:17:25 2024.
Dependencies resolved.

=====
Package          Architecture Version      Repository
=====
Installing:
  docker          x86_64       25.0.6-1.amzn2023.0.2  amazonlinux
Installing dependencies:
  containerd      x86_64       1.7.20-1.amzn2023.0.1   amazonlinux
  iptables-libs   x86_64       1.8.8-3.amzn2023.0.2   amazonlinux
  iptables-nft    x86_64       1.8.8-3.amzn2023.0.2   amazonlinux
  libcgroup        x86_64       3.0-1.amzn2023.0.1    amazonlinux
  libnetfilter_conntrack x86_64       1.0.8-2.amzn2023.0.2   amazonlinux
  libnftnl         x86_64       1.0.1-19.amzn2023.0.2  amazonlinux
  libnftnl1        x86_64       1.2.2-2.amzn2023.0.2  amazonlinux
  pigz            x86_64       2.5-1.amzn2023.0.3    amazonlinux
  runc            x86_64       1.1.13-1.amzn2023.0.1  amazonlinux

Transaction Summary
=====
```

- Then, configure cgroup in a daemon.json file by using following commands cd /etc/docker

```
cat <<EOF | sudo tee /etc/docker/daemon.json
{
  "exec-opts": [
    "native.cgroupdriver=systemd"
  ],
  "log-driver": "json-file",
  "log-opt": {
    "max-size": "100m"
  },
  "storage-driver": "overlay2"
}
EOF
```

```
[ec2-user@ip-172-31-26-174 ~]$ cd /etc/docker
[ec2-user@ip-172-31-26-174 docker]$ cat <<EOF | sudo tee /etc/docker/daemon.json
{
  "exec-opts": ["native.cgroupdriver=systemd"],
  "log-driver": "json-file",
  "log-opts": {
    "max-size": "100m"
  },
  "storage-driver": "overlay2"
}
EOF
{
  "exec-opts": ["native.cgroupdriver=systemd"],
  "log-driver": "json-file",
  "log-opts": {
    "max-size": "100m"
  },
  "storage-driver": "overlay2"
}

```

- Then after this run the following command to enable and start docker and also to load the daemon.json file.

```
sudo systemctl enable docker
sudo systemctl daemon-reload
sudo systemctl restart docker
```

```
[ec2-user@ip-172-31-26-174 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-172-31-26-174 docker]$ sudo systemctl daemon-reload
[ec2-user@ip-172-31-26-174 docker]$ sudo systemctl restart docker
[ec2-user@ip-172-31-26-174 docker]$ docker -v
Docker version 25.0.5, build 5dc9bcc
```

- docker -v

```
[ec2-user@ip-172-31-80-126 docker]$ docker -v
Docker version 25.0.5, build 5dc9bcc
```

3. Then Install Kubernetes with the following command.

- SELinux needs to be disable before configuring kubelet thus run the following command

```
sudo setenforce 0
```

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

```
[ec2-user@ip-172-31-26-174 docker]$ sudo setenforce 0
sudo sed -i 's/^SELINUX=enforcing$/SELINUX=permissive/' /etc/selinux/config
```

- Here We are adding kubernetes using the repository whose command is given below. cat <<EOF | sudo tee /etc/yum.repos.d/kubernetes.repo

```
[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
EOF
```

```
[ec2-user@ip-172-31-26-174 docker]$ sudo setenforce 0
sudo sed -i 's/^SELINUX=enforcing$/SELINUX=permissive/' /etc/selinux/config
[ec2-user@ip-172-31-26-174 docker]$ cat <<EOF | sudo tee /etc/yum.repos.d/kubernetes.repo
[kubernetes]
name=Kubernetes
baseurl=https://pkgs.k8s.io/core:/stable:/v1.30/rpm/
enabled=1
pgpcheck=1
pgpkey=https://pkgs.k8s.io/core:/stable:/v1.30/rpm/repo/repomd.xml.key
exclude=kubelet kubeadm kubectl cri-tools kubernetes-cni
EOF
[kubernetes]
name=Kubernetes
baseurl=https://pkgs.k8s.io/core:/stable:/v1.30/rpm/
enabled=1
pgpcheck=1
pgpkey=https://pkgs.k8s.io/core:/stable:/v1.30/rpm/repo/repomd.xml.key
exclude=kubelet kubeadm kubectl cri-tools kubernetes-cni
```

- After that Run following command to make the updation and also to install kubelet ,kubeadm, kubectl:

`sudo yum update`

```
[ec2-user@ip-172-31-80-126 docker]$ sudo yum update
Kubernetes
Dependencies resolved.
Nothing to do.
Complete!
```

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

```
[ec2-user@ip-172-31-80-126 docker]$ sudo yum install -y kubelet kubeadm kubectl --disableexcludes=kubernetes
Last metadata expiration check: 0:00:10 ago on Fri Sep 13 10:31:17 2024.
Dependencies resolved.
```

Package	Architecture	Version	Repository	Size
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
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

`Transaction Summary`

`Install 9 Packages`

```
Total
Kubernetes
Importing GPG key 0x9A296436:
  Userid : "isv:kubernetes OBS Project <isv:kubernetes@build.opensuse.org>"
  Fingerprint: DE15 B144 86CD 377B 9E87 6E1A 2346 54DA 9A29 6436
  From : https://pkgs.k8s.io/core:/stable:/v1.30/rpm/repo/repomd.xml.key
Key imported successfully
Running transaction check
Transaction check succeeded.
Running transaction test
Transaction test succeeded.
Running transaction
  Preparing :
  Installing : kubernetes-cni-1.4.0-150500.1.1.x86_64
  Installing : cri-tools-1.30.1-150500.1.1.x86_64
  Installing : libnetfilter_queue-1.0.5-2.amzn2023.0.2.x86_64
  Installing : libnetfilter_cttimeout-1.0.0-19.amzn2023.0.2.x86_64
  Installing : libnetfilter_cthelper-1.0.0-21.amzn2023.0.2.x86_64
  Installing : conntrack-tools-1.4.6-2.amzn2023.0.2.x86_64
  Running scriptlet: conntrack-tools-1.4.6-2.amzn2023.0.2.x86_64
  Installing : kubelet-1.30.5-150500.1.1.x86_64
  Running scriptlet: kubelet-1.30.5-150500.1.1.x86_64
  Installing : kubeadm-1.30.5-150500.1.1.x86_64
  Installing : kubectl-1.30.5-150500.1.1.x86_64
  Running scriptlet: kubectl-1.30.5-150500.1.1.x86_64
  verifying   : conntrack-tools-1.4.6-2.amzn2023.0.2.x86_64
  Verifying   : libnetfilter_cthelper-1.0.0-21.amzn2023.0.2.x86_64
  Verifying   : libnetfilter_cttimeout-1.0.0-19.amzn2023.0.2.x86_64
  verifying   : libnetfilter_queue-1.0.5-2.amzn2023.0.2.x86_64
  verifying   : cri-tools-1.30.1-150500.1.1.x86_64
  Verifying   : kubeadm-1.30.5-150500.1.1.x86_64
  Verifying   : kubelet-1.30.5-150500.1.1.x86_64
  Verifying   : kubectl-1.30.5-150500.1.1.x86_64
  Verifying   : kubernetes-cni-1.4.0-150500.1.1.x86_64
  Complete!
```

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

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

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

```
[ec2-user@ip-172-31-26-174 docker]$ sudo swapoff -a
echo "net.bridge.bridge-nf-call-iptables=1" | sudo tee -a /etc/sysctl.conf
sudo sysctl -p
net.bridge.bridge-nf-call-iptables=1
net.bridge.bridge-nf-call-iptables = 1
```

4. Initialize the Kubecluster

```
sudo kubeadm init --pod-network-cidr=10.244.0.0/16
```

```
[ec2-user@ip-172-31-80-126 docker]$ sudo kubeadm init --pod-network-cidr=10.244.0.0/16
[INFO] 10:32:44.629146 26680 version.go:256] remote version is much newer: v1.31.0; falling back to: stable-1.30
[init] Using Kubernetes version: v1.30.4
[preflight] Running pre-flight checks
```

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 172.31.26.174:6443 --token pv0yyi.xhllqhclfjr50pt8 \
--discovery-token-ca-cert-hash sha256:8293b2f6d29de466bd859007f5adbcdb3a
ecb0c446ba09033d32a5846b3d434f
```

- copy the token and save for future use .

```
kubeadm join 172.31.26.174:6443 --token pv0yyi.xhllqhclfjr50pt8
--discovery-token-ca-cert-hash
sha256:8293b2f6d29de466bd859007f5adbcdb3aecb0c446ba09033d32a5846b3d434f
```
- Copy the mkdir and chown commands from the top and execute them

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

```
[ec2-user@ip-172-31-80-126 docker]$ mkdir -p $HOME/.kube
sudo cp -i /etc/kubernetes/admin.conf $HOME/.kube/config
sudo chown $(id -u):$(id -g) $HOME/.kube/config
```

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

```
kubectl apply -f
```

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

```
[ec2-user@ip-172-31-26-174 docker]$ 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
```

5. Now that the cluster is up and running, we can deploy our nginx server on this cluster. Apply deployment using this following command:

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

```
[ec2-user@ip-172-31-26-174 docker]$ kubectl apply -f https://k8s.io/examples/pods/simple-pod.yaml  
pod/nginx created
```

Then use **kubectl get pods** to check whether the pod gets created or not.

```
[ec2-user@ip-172-31-26-174 docker]$ kubectl get pods  
NAME      READY   STATUS    RESTARTS   AGE  
nginx    0/1     Pending   0          12s
```

To convert state from pending to running use following command:

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.

- kubectl describe pod nginx

```
[ec2-user@ip-172-31-26-174 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)
```

```

Conditions:
  Type          Status
  PodScheduled  False
Volumes:
  kube-api-access-k4lj6:
    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
                        00s
                                         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.

```

- kubectl taint nodes --all node-role.kubernetes.io/control-plane-
- ```
[ec2-user@ip-172-31-26-174 ~]$ kubectl taint nodes --all node-role.kubernetes.io
/control-plane-
node/ip-172-31-26-174.ec2.internal untainted
```
6. Now check pod status is is running perform **kubectl get pods** this command.

```
[ec2-user@ip-172-31-28-70 docker]$ kubectl get pods
NAME READY STATUS RESTARTS AGE
nginx 0/1 ContainerCreating 0 39s
[ec2-user@ip-172-31-28-70 docker]$ kubectl get pods
NAME READY STATUS RESTARTS AGE
nginx 1/1 Running 1 (45s ago) 70s
```

7. Lastly, mention the port you want to host. Here i have used localhost 8081 then check it.

kubectl port-forward nginx 8081:80

```
[ec2-user@ip-172-31-26-174 ~]$ 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://127.0.0.1:8081`

```

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\bhush\OneDrive\Desktop\New folder (4)> ssh -i "Master_Ec2_Key.pem" ubuntu@ec2-54-196-129-215.compute-1.amazonaws.com
Welcome to Ubuntu 24.04 LTS (GNU/Linux 6.8.0-1012-aws x86_64)

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

System information as of Sun Sep 15 07:58:53 UTC 2024

System load: 0.15 Processes: 152
Usage of /: 55.3% of 6.71GB Users logged in: 1
Memory usage: 20% IPv4 address for enX0: 172.31.20.171
Swap usage: 0%

* Ubuntu Pro delivers the most comprehensive open source security and
 compliance features.

https://ubuntu.com/aws/pro

Expanded Security Maintenance for Applications is not enabled.

132 updates can be applied immediately.
38 of these updates are standard security updates.
To see these additional updates run: apt list --upgradable

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

Last login: Sun Sep 15 07:27:22 2024 from 152.58.7.117

```

## HTTP/1.1 200 OK

```

ubuntu@ip-172-31-20-171:~$ curl --head http://127.0.0.1:8080
HTTP/1.1 200 OK
Server: nginx/1.14.2
Date: Sun, 15 Sep 2024 07:59:03 GMT
Content-Type: text/html
Content-Length: 612
Last-Modified: Tue, 04 Dec 2018 14:44:49 GMT
Connection: keep-alive
ETag: "5c0692e1-264"
Accept-Ranges: bytes

```

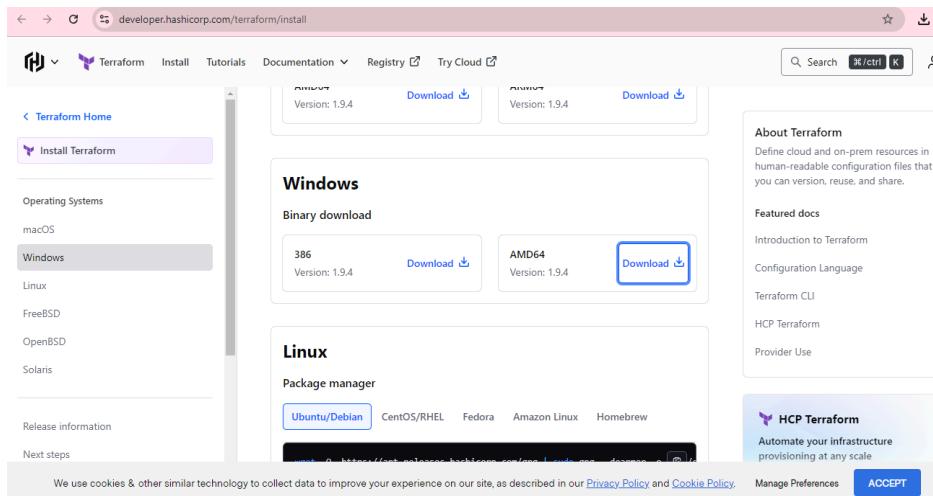
If the response is 200 OK and you can see the Nginx server name, your deployment was successful. We have successfully deployed our Nginx server on our EC2 instance.

**Conclusion:** Firstly I created an EC2 AWS Linux instance successfully.then installed docker and kubernetes successfully.then initialized kubernetes which given me token and chown and mkdir command. Then I execute mkdir and chown the command successfully. Then I installed a networking plugin called flannel successfully. Then I tried to deploy nginx which initially gave an error. Then I deployed (simple-pod.yml ) nginx successfully and also checked by using the get pods command.then hosted it on localhost 8081 ie http://localhost:8081 successfully.

## 1. Download terraform

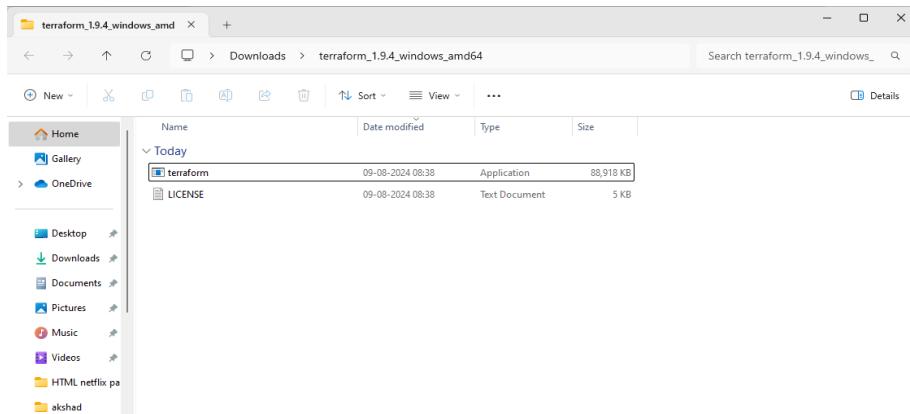
To install Terraform, First Download the Terraform Cli Utility for windows from terraforms official website

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

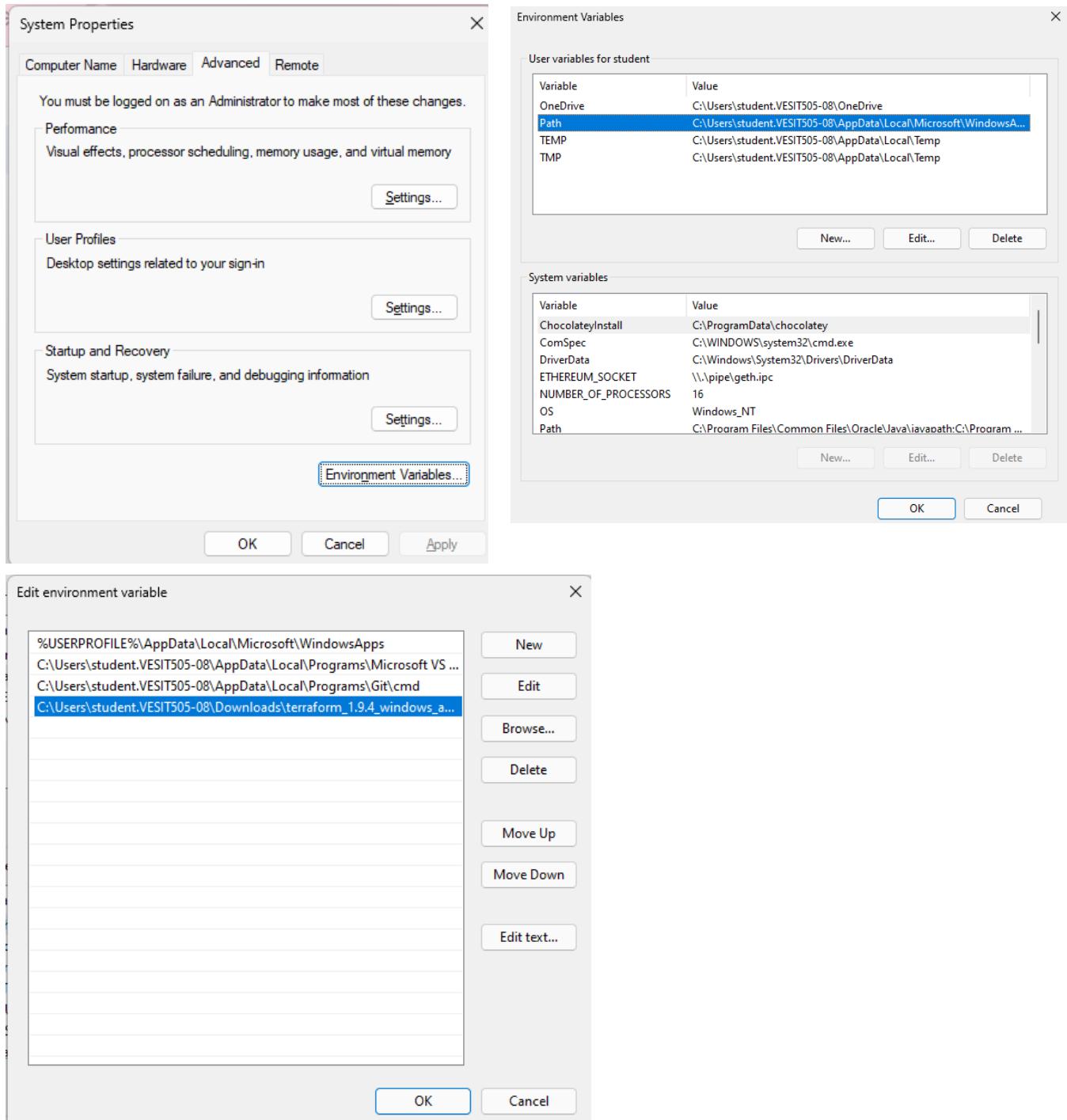


The screenshot shows the Terraform download page on developer.hashicorp.com. The left sidebar has a 'Windows' tab selected under 'Operating Systems'. The main content area shows the 'Windows' section with 'Binary download' options for '386' and 'AMD64' architectures, both at version 1.9.4. Below this is the 'Linux' section with 'Package manager' options for 'Ubuntu/Debian', 'CentOS/RHEL', 'Fedora', 'Amazon Linux', and 'Homebrew'. To the right, there's an 'About Terraform' summary, 'Featured docs' links, and an 'HCP Terraform' section.

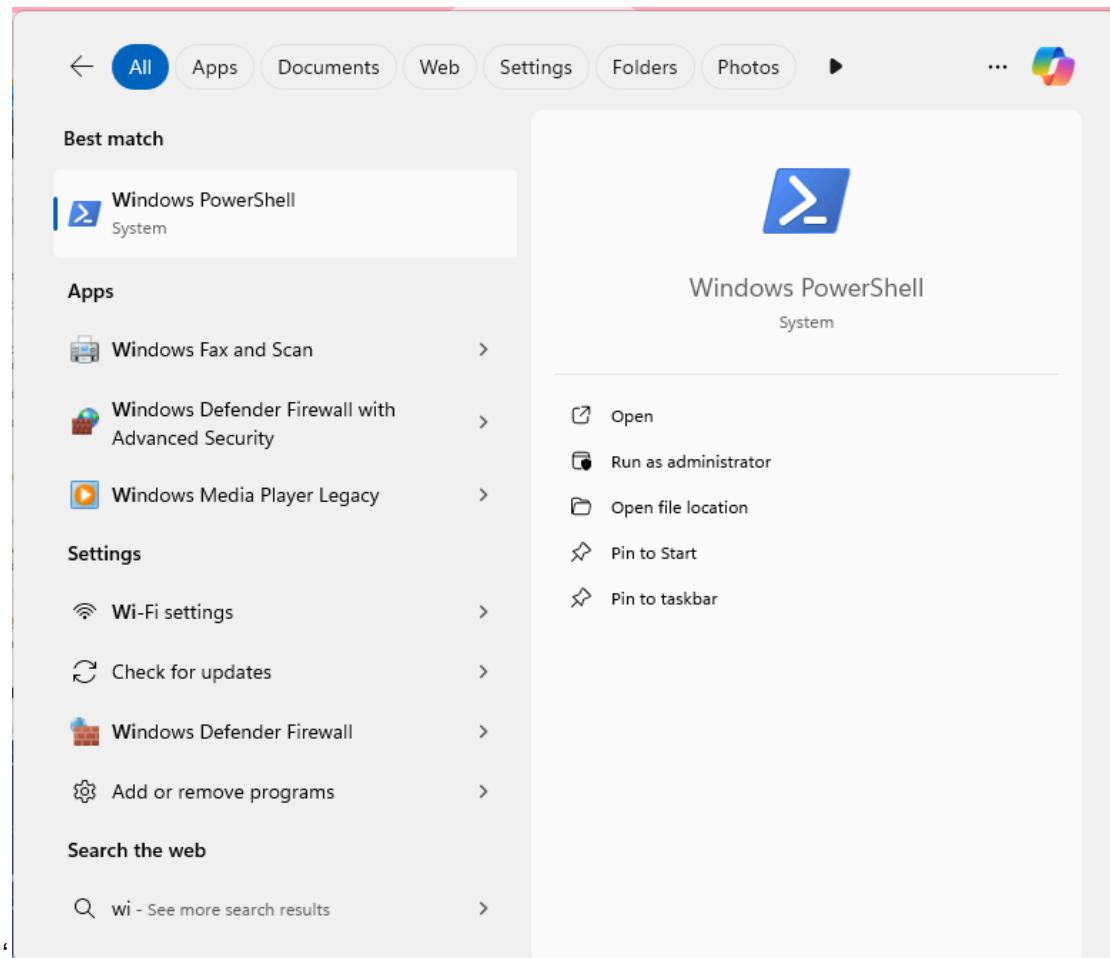
## 2. Extract the downloaded setup file Terraform.exe in C:\Terraform directory



### 3. Set the System path for Terraform in Environment Variables



#### 4. Open PowerShell with Admin Access



## 5. Open Terraform in PowerShell and check its functionality

```
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\student.VESIT505-08> 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

 untaint Remove the 'tainted' state from a resource instance
 version Show the current Terraform version
 workspace Workspace management

Global options (use these before the subcommand, if any):
 -chdir=DIR Switch to a different working directory before executing the
 given subcommand.
 -help Show this help output, or the help for a specified subcommand.
 -version An alias for the "version" subcommand.
PS C:\Users\student.VESIT505-08> |
```

## Experiment 6:

### 1. Run some basic commands to check the version and confirm installation of docker

```
C:\Users\INFT505-17>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
```

```
C:\Users\INFT505-17>docker --version
Docker version 27.1.1, build 6312585
```

### 2. Initialise terraform

```
C:\Users\INFT505-17>terraform init
Terraform initialized in an empty directory!

The directory has no Terraform configuration files. You may begin working
with Terraform immediately by creating Terraform configuration files.

C:\Users\INFT505-17>cd Desktop
C:\Users\INFT505-17\Desktop>
C:\Users\INFT505-17\Desktop>cd Terraform Scripts
C:\Users\INFT505-17\Desktop\Terraform Scripts>cd Docker
C:\Users\INFT505-17\Desktop\Terraform Scripts\Docke>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.
```

### 3. Terraform plan

```
C:\Users\INFT505-17\Desktop\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)
```

### 4. Terraform apply

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

Do you want to perform these actions?

Terraform will perform the actions described above.  
Only 'yes' will be accepted to approve.

Enter a value: yes

```
docker_container.foo: Creating...
docker_container.foo: Creation complete after 3s [id=1adc9dfc498825398e014939d7966749d843181417953d0b9f462a55ae7c1492]
```

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

```
C:\Users\INFT505-17\Desktop\Terraform Scripts\Docker>docker images
REPOSITORY TAG IMAGE ID CREATED SIZE
ubuntu latest edbfe74c41f8 3 weeks ago 78.1MB
```

### 5. Terraform destroy

```
C:\Users\INFT505-17\Desktop\Terraform Scripts\Docker>terraform destroy
docker_image.ubuntu: Refreshing state... [id=sha256:edbfe74c41f8a3501ce542e137cf28ea04dd03e6df8c9d66519b6ad761c2598aubuntu:latest]
docker_container.foo: Refreshing state... [id=ad11c2cdc9ccae208b906f7e6be260805512a66568a65f6b3bffb3682cf30cb4]
```

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_container.foo will be destroyed
- resource "docker_container" "foo" {
 - attach = false -> null
 - command = [
 - "/bin/bash",
 - "c",
 - "while true; do sleep 3600; done",
] -> null
 - cpu_shares = 0 -> null
 - dns = [] -> null
 - dns_opts = [] -> null
 - dns_search = [] -> null
 - entrypoint = [] -> null
```

```

docker_image.ubuntu will be destroyed
resource "docker_image" "ubuntu" {
 - id = "sha256:edbfe74c41f8a3501ce542e137cf28ea04dd03e6df8c9d66519b6ad761c2598aubuntu:latest" -> null
 - image_id = "sha256:edbfe74c41f8a3501ce542e137cf28ea04dd03e6df8c9d66519b6ad761c2598a" -> null
 - latest = "sha256:edbfe74c41f8a3501ce542e137cf28ea04dd03e6df8c9d66519b6ad761c2598a" -> null
 - name = "ubuntu:latest" -> null
 - repo_digest = "ubuntu@sha256:8a37d68f4f73ebf3d4efafbcf66379bf3728902a8038616808f04e34a9ab63ee" -> null
}

Plan: 0 to add, 0 to change, 2 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_container.foo: Destroying... [id=ad11c2cdc9ccae208b906f7e6be260805512a66568a65f6b3bffb3682cf30cb4]
docker_container.foo: Destruction complete after 2s
docker_image.ubuntu: Destroying... [id=sha256:edbfe74c41f8a3501ce542e137cf28ea04dd03e6df8c9d66519b6ad761c2598aubuntu:latest]
docker_image.ubuntu: Destruction complete after 0s

Destroy complete! Resources: 2 destroyed.

```

C:\Users\INFT505-17\Desktop\Terraform Scripts\Docker>`docker images`

| REPOSITORY | TAG | IMAGE ID | CREATED | SIZE |
|------------|-----|----------|---------|------|
|------------|-----|----------|---------|------|

C:\Users\INFT505-17\Desktop\Terraform Scripts\Docker>

## 6. Terraform Validate

C:\Users\INFT505-17\Desktop\Terraform Scripts\Docker>`terraform validate`  
Success! The configuration is valid.

## 7. Terraform refresh

C:\Users\INFT505-17\Desktop\Terraform Scripts\Docker>`terraform refresh`  
`docker_image.ubuntu: Refreshing state... [id=sha256:edbfe74c41f8a3501ce542e137cf28ea04dd03e6df8c9d66519b6ad761c2598aubuntu:latest]`  
`docker_container.foo: Refreshing state... [id=4adda4f9a5c585809eebe921da757560d333f9a0101a8cd25784bd238355aba5]`

## 8. Terraform state list

C:\Users\INFT505-17\Desktop\Terraform Scripts\Docker>`terraform state list`  
`docker_container.foo`  
`docker_image.ubuntu`

**Aim:** To understand Static Analysis SAST process and learn to integrate Jenkins SAST to SonarQube/GitLab.

**Theory:** 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.

## What problems does SAST solve?

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.

## Why is SAST important?

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.

## What are the key steps to run SAST effectively?

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. **Create the scanning infrastructure, and deploy the tool.** 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. **Provide governance and training.** 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:

Windows installation

Step 1 Install JDK 1.8

Step 2 download and install jenkins

<https://www.blazemeter.com/blog/how-to-install-jenkins-on-windows>

### Ubuntu installation

<https://www.digitalocean.com/community/tutorials/how-to-install-java-with-apt-on-ubuntu-20-04#installing-the-default-jre-jdk>

Step 1 Install JDK 1.8

sudo apt-get install openjdk-8-jre

sudo apt install default-jre

<https://www.digitalocean.com/community/tutorials/how-to-install-jenkins-on-ubuntu-20-04>

[Open SSH](#)

## Prerequisites:

- [Jenkins installed](#)
- [Docker Installed](#) (for SonarQube)

(sudo apt-get install docker-ce=5:20.10.15~3-0~ubuntu-jammy  
docker-ce-cli=5:20.10.15~3-0~ubuntu-jammy containerd.io docker-compose-plugin)

- SonarQube Docker Image

## Steps to integrate Jenkins with SonarQube

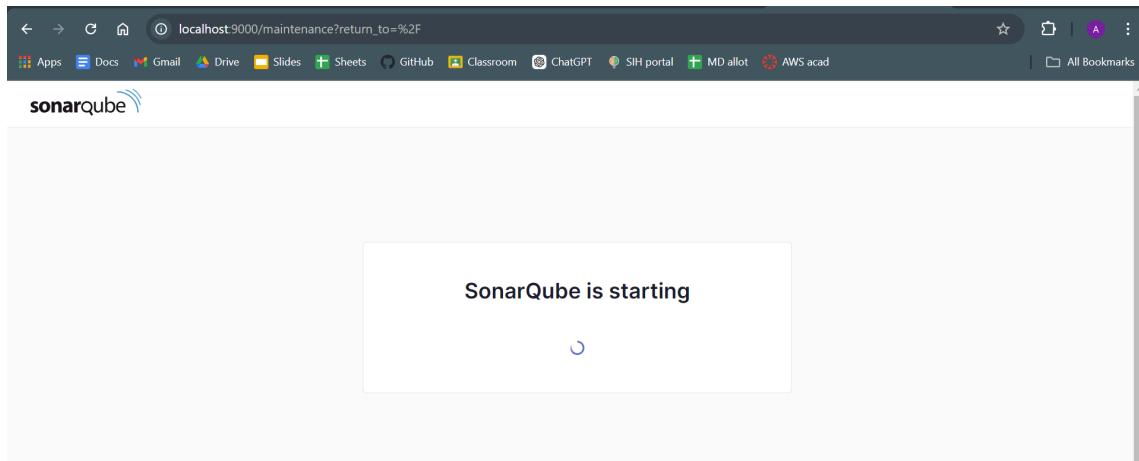
1. Open up Jenkins Dashboard on localhost, port 8080 or whichever port it is at for you.
2. Run SonarQube in a Docker container using this command -

```
C:\Windows\System32>docker run -d --name sonarqube -e SONAR_ES_BOOTSTRAP_CHECKS_DISABLE=true -p 9000:9000 sonarqube:latest
st
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: sha256:72e9feec71242af83faf65f95a40d5e3bb2822a6c3b2cda8568790f3d31aecde
Status: Downloaded newer image for sonarqube:latest
b37288b0b410d9fab6bebf8d6b87a0ef7f75387b6070308da2f24c8548481cc
```

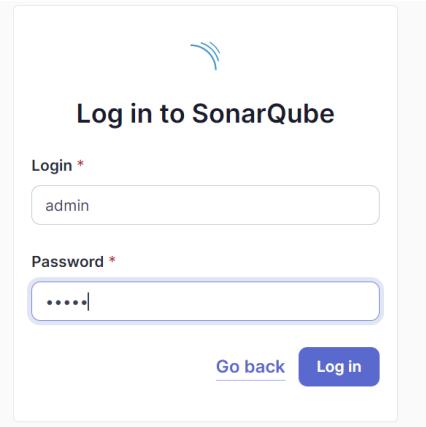
### Warning: run below command only once

```
docker run -d --name sonarqube -e SONAR_ES_BOOTSTRAP_CHECKS_DISABLE=true -p 9000:9000 sonarqube:latest
```

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**

### Create a local project

Project display name \*

Project key \*

Main branch name \*

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

[Cancel](#) [Next](#)

Setup the project and come back to Jenkins Dashboard.

Go to Manage Jenkins and search for SonarQube Scanner for Jenkins and install it.

### SonarQube Scanner for Jenkins 2.17.2

This plugin allows an easy integration of [SonarQube](#), the open source platform for Continuous Inspection of code quality.

[Report an issue with this plugin](#)



## 6. Under Jenkins ‘Configure System’, look for SonarQube Servers and enter the details.

Enter the Server Authentication token if needed.

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

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

## 9. Choose this GitHub repository in Source Code Management.

[https://github.com/shazforiot/MSBuild\\_firstproject.git](https://github.com/shazforiot/MSBuild_firstproject.git)

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

Git

Repositories

Repository URL

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

Credentials

- none -

+ Add

Advanced

the integration.

10. Under Build-> Execute SonarQube Scanner, enter these Analysis properties. Mention the SonarQube Project Key, Login, Password, Source path and Host URL. 11. Go to [http://localhost:9000/<user\\_name>/permissions](http://localhost:9000/<user_name>/permissions) and allow Execute Permissions to the Admin user.

Build Steps

Execute SonarQube Scanner

JDK

JDK to be used for this SonarQube analysis

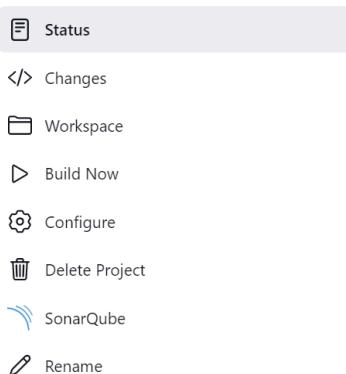
(Inherit From Job)

Path to project properties

Analysis properties

```
sonar.projectKey=sonarqube_test
sonar.projectName=sonarqube_test
sonar.projectVersion=1.0
sonar.sources=
sonar.language=java
sonar.host.url=http://localhost:9000
```

12. Run The Build.



Check the console output.

Console Output

Started by user Anshi Tiwari  
Running as SYSTEM  
Building in workspace C:\ProgramData\Jenkins\.jenkins\workspace\anshi\_item  
The recommended git tool is: NONE  
No credentials specified  
> git.exe rev-parse --resolve-git-dir C:\ProgramData\Jenkins\.jenkins\workspace\anshi\_item\.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 2.46.0.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"

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

The screenshot shows the SonarQube interface for the 'main' branch. At the top, there's a navigation bar with a star icon, the project name 'sonarqube-anshi /', and dropdown menus for 'main' and 'Activity'. Below the navigation, there are tabs for 'Overview', 'Issues', 'Security Hotspots', 'Measures', 'Code', and 'Activity'. The 'Overview' tab is selected. In the main content area, the word 'main' is displayed above a green box containing a white checkmark icon and the word 'Passed'. To the left of this box is a yellow warning icon with a triangle. Below the box, a tooltip says 'The last analysis has warnings. See details'.

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

## Conclusion of the Experiment

This experiment focused on integrating Jenkins with SonarQube for Static Application Security Testing (SAST). Throughout the process, several challenges emerged:

1. **Token Authentication Issues:** Encountering a 401 Unauthorized error emphasized the importance of correctly managing authentication tokens and user permissions within SonarQube.

2. **Project Configuration:** Initial confusion regarding the configuration parameters in Jenkins highlighted the need for attention to detail. Ensuring the `sonar.login` property was set correctly was crucial for successful integration.
3. **Maven Misunderstanding:** Misunderstanding the role of Maven led to unnecessary command-line attempts. Recognizing that Jenkins handles the build process streamlined the workflow.
4. **Error Navigation:** Reading console logs proved vital in troubleshooting and identifying issues, enhancing my problem-solving skills.

**Aim:** Create a Jenkins CICD Pipeline with SonarQube / GitLab Integration to perform a static analysis of the code to detect bugs, code smells, and security vulnerabilities on a sample Web / Java / Python application.

### **Theory:**

#### **What is 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.

#### **What problems does SAST solve?**

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.

#### **Why is SAST important?**

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 Pipeline executes the job in a defined manner by first coding it and then structuring it inside several blocks that may include several steps or tasks.

## What is 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.

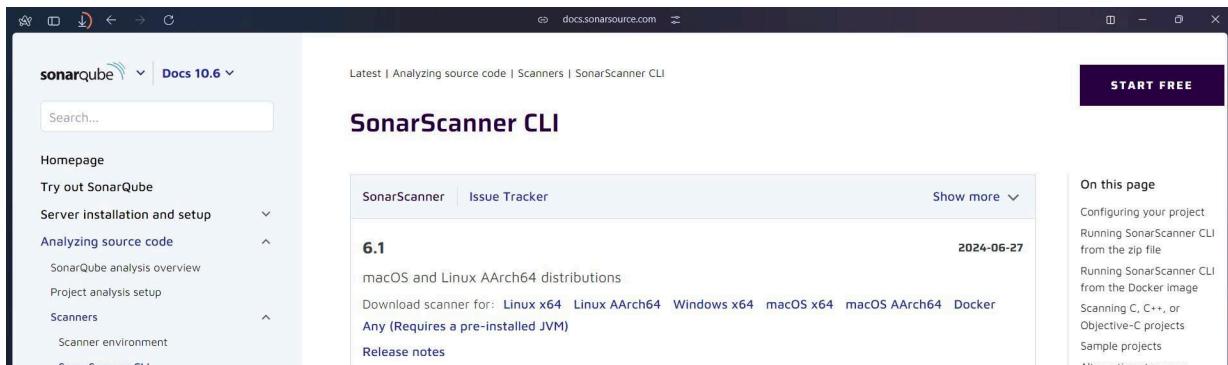
## Benefits of SonarQube

- **Sustainability** - Reduces complexity, possible vulnerabilities, and code duplications, optimising the life of applications.
- **Increase productivity** - 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 code** - Code quality control is an inseparable part of the process of software development.
- **Detect Errors** - 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
- **Business scaling** - 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

## Steps to create a Jenkins CI/CD Pipeline and use SonarQube to perform SAST

### Step 1: Download sonar scanner

<https://docs.sonarsource.com/sonarqube/latest/analyzing-source-code/scanners/sonarscanner/> Visit this link and download the sonarqube scanner CLI.



Extract the downloaded zip file in a folder.

1) Docker

Run docker -v command.

```
C:\Users\Anshi>docker -v
Docker version 27.1.1, build 6312585
```

2) Install sonarqube image

Command: docker pull sonarqube

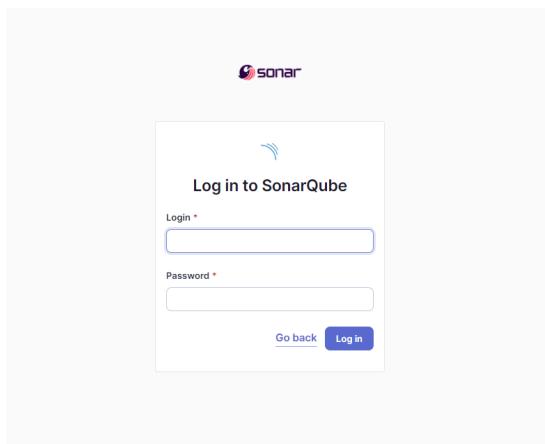
```
C:\Users\Anshi>docker pull sonarqube
Using default tag: latest
latest: Pulling from library/sonarqube
Digest: sha256:72e9feec71242af83faf65f95a40d5e3bb2822a6c3b2cda8568790f3d31aecde
Status: Image is up to date for sonarqube:latest
docker.io/library/sonarqube:latest
```

3) Run sonarqube image

```
docker run -d --name sonarqube -e SONAR_ES_BOOTSTRAP_CHECKS_DISABLE=true -p 9000:9000
sonarqube:latest
```

```
PS C:\Users\saira\OneDrive\Desktop\AdvDevOps\lab7> docker run -d --name sonarqube -e SONAR_ES_BOOTSTRAP_CHECKS_DISABLE=t
rue -p 9000:9000 sonarqube:latest
36ff8a656bd28857ba9a28bf7bb0174099ae3232a9fc9ba2766d46f0c14d08a6
```

4) Run localhost:9000



- 5) Login using username="admin", password="admin". It will prompt you to set a new password.

**Update your password**

⚠ This account should not use the default password.

**Enter a new password**  
All fields marked with \* are required

**Old Password \***

**New Password \***

**Confirm Password \***

**Update**

- 6) This is the interface. Create a local project with the name sonarqube.

1 of 2

**Create a local project**

Project display name \*

sonarqube

Project key \*

sonarqube

Main branch name \*

main

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

Cancel Next

- 7) Open jenkins dashboard using localhost on whichever port it is hosted.

| S | W | Name ↓     | Last Success  |
|---|---|------------|---------------|
| ✓ | ☀ | anshi_item | 1 day 8 hr #6 |

- 8) Go to manage jenkins → Search for Sonarqube Scanner for Jenkins and install it.

The screenshot shows the Jenkins plugin search interface. A search bar at the top contains the text "sonarqube". Below the search bar, there are two tabs: "Install" and "Name ↴". Under the "Install" tab, three plugins are listed:

- SonarQube Scanner** 2.17.2  
External Site/Tool Integrations Build Reports  
This plugin allows an easy integration of [SonarQube](#), the open source platform for Continuous Inspection of code quality.
- Sonar Gerrit** 388.v9b\_f1cb\_e42306  
External Site/Tool Integrations  
This plugin allows to submit issues from [SonarQube](#) to [Gerrit](#) as comments directly.
- SonarQube Generic Coverage** 1.0  
TODO

9) Now, go to Manage Jenkins → System. Under Sonarqube servers, add a server. Add server authentication token if needed.

The screenshot shows the "Add SonarQube" configuration form. The form fields are as follows:

- Name**: A mandatory field indicated by a red exclamation mark icon and the text "This property is mandatory".
- Server URL**: Default is `http://localhost:9000`. The URL input field is empty.
- Server authentication token**: A dropdown menu showing "- none -". Below it is a "+ Add" button and an "Advanced" dropdown menu.

At the bottom of the form are "Save" and "Apply" buttons.

- 10) Go to Manage Jenkins → Tools. Go to SonarQube scanner, choose the latest configuration and choose install automatically.

SonarQube Scanner

Name  
sonar-scan

Install automatically ?

Install from Maven Central

Version  
SonarQube Scanner 6.2.0.4584

Add Installer ▾

- 11) After configuration, create a New Item → choose a pipeline project.

#### New Item

Enter an item name  
anshi-sonar

Select an item type



Freestyle project

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



Pipeline

Orchestrates long-running activities that can span multiple build agents. Suitable for build workflows and/or organizing complex activities that do not easily fit in free-style job types.

- 12) Under Pipeline script, enter the following:

```
node {
stage('Cloning the GitHub Repo') {
git 'https://github.com/shazforiot/GOL.git'

stage('SonarQube analysis') {
withSonarQubeEnv('<Name of SonarQube environment on Jenkins>') { sh """"
<PATH_TO SONARQUBE_SCANNER_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=<Link to hosted SonarQube>(default: http://localhost:9000/) """
}
}
}
```

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

Script ?

```
1 ▼ node {
2 ▼ stage('Cloning the GitHub Repo') {
3 git 'https://github.com/shazforiot/GOL.git'
4 }
5 ▼ stage('SonarQube analysis') {
6 withSonarQubeEnv('<Name of SonarQube environment on Jenkins>') { sh """
7 <PATH_TO SONARQUBE_SCANNER_FOLDER>/bin/sonar-scanner \
8 -D sonar.login=<admin> \
9 -D sonar.password=<admin123> \
10 -D sonar.projectKey=<sonarqube-anshi-2> \
11 -D sonar.exclusions=vendor/**,resources/**,**/*.java \
12 -D sonar.host.url=<Link to hosted SonarQube>(default: http://localhost:9000) """
13 }
14 }
15 }
16 }
```

### 13) Check the console output once

#### ✓ Console Output

 Download

 Copy

[View as plain text](#)

```
Started by user Anshi Tiwari
[Pipeline] Start of Pipeline
[Pipeline] node
Running on Jenkins in C:\ProgramData\Jenkins\workspace\ansi-sonar
[Pipeline] {
[Pipeline] stage
[Pipeline] { (Cloning the GitHub Repo)
[Pipeline] git

21:10:28.890 INFO SCM revision ID 'ba799ba7e1b576f04a4612322b0412c5e6e1e5e4'
21:10:34.239 INFO Analysis report generated in 3621ms, dir size=127.2 MB
21:10:51.328 INFO Analysis report compressed in 17087ms, zip size=29.6 MB
21:10:51.873 INFO Analysis report uploaded in 546ms
21:10:51.874 INFO ANALYSIS SUCCESSFUL, you can find the results at: http://localhost:9000/dashboard?id=sonarqube-anshi-2
21:10:51.874 INFO Note that you will be able to access the updated dashboard once the server has processed the submitted analysis report
21:10:51.874 INFO More about the report processing at http://localhost:9000/api/ce/task?id=b1d670e7-bff2-41c8-8e0a-b5ab1d303aac
21:11:03.958 INFO Analysis total time: 7:46.716 s
21:11:03.960 INFO SonarScanner Engine completed successfully
21:11:04.665 INFO EXECUTION SUCCESS
21:11:04.666 INFO Total time: 7:51.358s
[Pipeline] }
[Pipeline] // withSonarQubeEnv
[Pipeline] }
[Pipeline] // stage
[Pipeline] }
[Pipeline] // node
[Pipeline] End of Pipeline
Finished: SUCCESS
```

## 14) Now, check the project in SonarQube

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

The screenshot shows the SonarQube interface for the 'main' project. At the top, there's a navigation bar with tabs for Overview, Issues, Security Hotspots, Measures, Code, and Activity. The 'Overview' tab is currently selected. Below the navigation bar, a large green box displays a checkmark and the word 'Passed' next to the text 'Quality Gate'. A note below says '(○) New analysis in progress'. There are two tabs at the bottom: 'New Code' and 'Overall Code', with 'Overall Code' being the active tab.

## 15) Code Problems

### • Consistency

The screenshot shows the 'Clean Code Attribute' report for the 'Intentionality' category. On the left, there's a sidebar with sections for 'Issues in new code' and 'Clean Code Attribute' (which is expanded). Under 'Clean Code Attribute', the 'Intentionality' section is highlighted, showing a value of 197k. Other sections include 'Adaptability' (0), 'Responsibility' (0), and a 'Software Quality' section. On the right, the main panel shows two code problems for the file 'gameoflife-core/build/reports/tests/all-tests.html'. The first problem is 'Insert a <!DOCTYPE> declaration to before this <html> tag.' (Consistency, Reliability, user-experience). The second problem is 'Remove this deprecated "width" attribute.' (Consistency, Maintainability, html5, obsolete).

### • Intentionality

The screenshot shows the 'Clean Code Attribute' report for the 'Intentionality' category. On the left, there's a sidebar with sections for 'Issues in new code' and 'Clean Code Attribute' (which is expanded). Under 'Clean Code Attribute', the 'Intentionality' section is highlighted, showing a value of 14k. Other sections include 'Adaptability' (0), 'Responsibility' (0), and a 'Software Quality' section. On the right, the main panel shows two code problems for the file 'gameoflife-core/build/reports/tests/all-tests.html'. The first problem is 'Add "lang" and/or "xml:lang" attributes to this "<html>" element' (Intentionality, Reliability, accessibility, wcag2-a). The second problem is 'Add "<th>" headers to this "<table>".' (Intentionality, Reliability, accessibility, wcag2-a).

## **Conclusion**

The experiment successfully integrated SonarQube analysis into a Jenkins pipeline for a GitHub project, allowing for automated code quality checks. While the cloning of the repository worked well, several challenges arose, particularly with configuring the SonarScanner path and handling command execution errors in the Windows environment. Issues with authentication and the need for a user token added complexity to the setup. Despite these hurdles, the integration improved the project's continuous integration process, and lessons learned can inform future enhancements, such as better error handling and notifications for analysis results.

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

## **Steps:**

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

The screenshot shows the AWS CloudFormation console interface for creating a new stack. The 'Name and tags' section has 'nagios' entered in the 'Name' field. The 'Application and OS Images (Amazon Machine Image)' section shows 'Amazon Linux 2023 AMI' selected, which is marked as 'Free tier eligible'. The 'Key pair (login)' section has 'server' selected as the key pair name. In the 'Network settings' section, the VPC 'vpc-051bba342b3626898' is chosen, and the 'Auto-assign public IP' option is enabled. A note at the bottom states: 'Security groups that you add or remove here will be added to or removed from all your network interfaces.'

2. Under Security Group, make sure HTTP, HTTPS, SSH, ICMP are open from everywhere.

The screenshot shows the AWS CloudFormation console under the Network & Security section. A security group named 'sg-0b1355e80625c05ee - Nagios' is selected. The 'Inbound rules' tab is active, displaying the following rules:

| Name                  | Security group rule... | IP version      | Type      | Protocol | Port range | Source | Description |
|-----------------------|------------------------|-----------------|-----------|----------|------------|--------|-------------|
| sgr-086d9781957f5d... | IPv4                   | SSH             | TCP       | 22       | 0.0.0.0/0  | -      |             |
| sgr-05372165ab2a49... | IPv6                   | All ICMP - IPv6 | IPv6 ICMP | All      | /0         | -      |             |
| sgr-0073fb1b89d0214e0 | IPv4                   | HTTPS           | TCP       | 443      | 0.0.0.0/0  | -      |             |
| sgr-0d667517b1040d... | IPv4                   | Custom TCP      | TCP       | 5666     | 0.0.0.0/0  | -      |             |
| sgr-0cd7176351b1f8596 | IPv4                   | All ICMP - IPv4 | ICMP      | All      | 0.0.0.0/0  | -      |             |
| sgr-0225770a0073a7... | IPv4                   | All traffic     | All       | All      | 0.0.0.0/0  | -      |             |
| sgr-0368947d47bb93... | IPv4                   | HTTP            | TCP       | 80       | 0.0.0.0/0  | -      |             |

You have to edit the inbound rules of the specified Security Group for this.

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

```
PS C:\Users\Sadneya\Downloads> ssh -i "server.pem" ec2-user@ec2-18-232-155-134.compute-1.amazonaws.com
The authenticity of host 'ec2-18-232-155-134.compute-1.amazonaws.com (18.232.155.134)' can't be established.
ED25519 key fingerprint is SHA256:6UVLjB6FbGB7A92sIEobs4886tozb5yML0ekn5Xzfrw.
This key is not known by any other names
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added 'ec2-18-232-155-134.compute-1.amazonaws.com' (ED25519) to the list of known hosts.

 _#
 /_ _###_ Amazon Linux 2023
 ~~ _#####\
 ~~ \###|
 ~~ \#/ ___ https://aws.amazon.com/linux/amazon-linux-2023
 ~~ .-. _/
 ~~ / _/
 /m/^
```

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
```

```
[ec2-user@ip-172-31-39-90 ~]$ sudo yum update
sudo yum install httpd php
sudo yum install gcc glibc glibc-common
sudo yum install gd gd-devel
Last metadata expiration check: 0:02:27 ago on Fri Oct 4 03:35:02 2024.
Dependencies resolved.
Nothing to do.
Complete!
```

```
Installed:
brotli-1.0.9-4.amzn2023.0.2.x86_64
bzip2-devel-1.0.8-6.amzn2023.0.2.x86_64
cmake-fsystesm-3.22.2-1.amzn2023.0.4.x86_64
fontconfig-devel-2.13.94-2.amzn2023.0.2.x86_64
freetype-2.13.2-5.amzn2023.0.1.x86_64
gd-2.3.3-5.amzn2023.0.3.x86_64
glib2-devel-2.74.7-689.amzn2023.0.2.x86_64
google-noto-sans-vf-fonts-20201206-2.amzn2023.0.2.noarch
graphite2-devel-1.3.14-7.amzn2023.0.2.x86_64
harfbuzz-devel-7.0.0-2.amzn2023.0.1.x86_64
jbigkit-libs-2.1-21.amzn2023.0.2.x86_64
libICE-1.0.10-6.amzn2023.0.2.x86_64
libX11-1.7.2-3.amzn2023.0.4.x86_64
libX11-devel-1.7.2-3.amzn2023.0.4.x86_64
libXau-1.0.9-6.amzn2023.0.2.x86_64
libXext-1.3.4-6.amzn2023.0.2.x86_64
libXpm-devel-3.5.15-2.amzn2023.0.3.x86_64
libXt-1.2.0-4.amzn2023.0.2.x86_64
libffi-devel-3.4.4-1.amzn2023.0.1.x86_64
libicu-devel-67.1-7.amzn2023.0.3.x86_64
libjpeg-turbo-devel-2.1.4-2.amzn2023.0.5.x86_64
libpng-2.1.6.37-10.amzn2023.0.6.x86_64
libselinux-devel-3.4-5.amzn2023.0.2.x86_64
libtiff-4.4.0-4.amzn2023.0.18.x86_64
libwebp-1.2.4-1.amzn2023.0.6.x86_64
libxcb-1.13.1-7.amzn2023.0.2.x86_64
libxml2-devel-2.10.4-1.amzn2023.0.6.x86_64
pcre2-utf16-10.40-1.amzn2023.0.3.x86_64
pixman-0.40.0-3.amzn2023.0.3.x86_64
xml-common-0.6.3-56.amzn2023.0.2.noarch
xz-devel-5.2.5-9.amzn2023.0.2.x86_64
```

```
brotli-devel-1.0.9-4.amzn2023.0.2.x86_64
cairo-1.17.6-2.amzn2023.0.1.x86_64
fontconfig-2.13.94-2.amzn2023.0.2.x86_64
fonts-fsystesm-1.2.0.5-12.amzn2023.0.2.noarch
freetype-devel-2.13.2-5.amzn2023.0.1.x86_64
gd-devel-2.3.3-5.amzn2023.0.3.x86_64
google-noto-fonts-common-20201206-2.amzn2023.0.2.noarch
graphite2-1.3.14-7.amzn2023.0.2.x86_64
harfbuzz-7.0.0-2.amzn2023.0.1.x86_64
harfbuzz-icu-7.0.0-2.amzn2023.0.1.x86_64
langpacks-core-font-en-3.0-21.amzn2023.0.4.noarch
libSM-1.2.3-8.amzn2023.0.2.x86_64
libX11-common-1.7.2-3.amzn2023.0.4.noarch
libX11-xcb-1.7.2-3.amzn2023.0.4.x86_64
libXau-devel-1.0.9-6.amzn2023.0.2.x86_64
libXpm-3.5.15-2.amzn2023.0.3.x86_64
libXrender-0.9.10-14.amzn2023.0.2.x86_64
libblkid-devel-2.37.4-1.amzn2023.0.4.x86_64
libicu-67.1-7.amzn2023.0.3.x86_64
libjpeg-turbo-2.1.4-2.amzn2023.0.5.x86_64
libmount-devel-2.37.4-1.amzn2023.0.4.x86_64
libpng-devel-2.1.6.37-10.amzn2023.0.6.x86_64
libsep0-devel-3.4-3.amzn2023.0.3.x86_64
libtiff-devel-4.4.0-4.amzn2023.0.18.x86_64
libwebp-devel-1.2.4-1.amzn2023.0.6.x86_64
libxcb-devel-1.13.1-7.amzn2023.0.2.x86_64
pcre2-devel-10.40-1.amzn2023.0.3.x86_64
pcre2-utf32-10.40-1.amzn2023.0.3.x86_64
sysprof-capture-devel-3.40.1-2.amzn2023.0.2.x86_64
xorg-x11proto-devel-2021.4-1.amzn2023.0.2.noarch
zlib-devel-1.2.11-33.amzn2023.0.5.x86_64
```

```
Complete!
```

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-172-31-39-90 ~]$ sudo adduser -m nagios
sudo passwd nagios
Changing password for user nagios.
New password:
```

```
Retype new password:
passwd: all authentication tokens updated successfully.
```

## 6. Create a new user group

```
sudo groupadd nagcmd
```

```
[ec2-user@ip-172-31-39-90 ~]$ 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
```

```
[ec2-user@ip-172-31-39-90 ~]$ sudo usermod -a -G nagcmd nagios
sudo usermod -a -G nagcmd apache
```

## 8. Create a new directory for Nagios downloads

```
mkdir ~/downloads
cd ~/downloads
```

```
[ec2-user@ip-172-31-39-90 ~]$ mkdir ~/downloads
cd ~/downloads
```

## 9. Use wget to download the source zip files.

```
wget https://go.nagios.org/l/975333/2024-09-17/6kqcx
```

```
[ec2-user@ip-172-31-39-90 downloads]$ wget https://go.nagios.org/l/975333/2024-09-17/6kqcx
--2024-10-04 03:45:50-- https://go.nagios.org/l/975333/2024-09-17/6kqcx
Resolving go.nagios.org (go.nagios.org)... 3.92.120.28, 34.237.219.119, 52.54.96.194, ...
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-4.5.5.tar.gz?utm_source=Nagios.org&utm_content=Download+Form&utm_campaign=Core+4.5.5+Download+&pi_content=1e9662c93afb2ed6bd2e3f3cc38771a7f01125e969f2a75b0e2254439d4a81d8 [following]
--2024-10-04 03:45:51-- 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=1e9662c93afb2ed6bd2e3f3cc38771a7f01125e969f2a75b0e2254439d4a81d8
Resolving assets.nagios.com (assets.nagios.com)... 45.79.49.120, 2600:3c00::f03c:92ff:fef7:45ce
Connecting to assets.nagios.com (assets.nagios.com)|45.79.49.120|:80... connected.
HTTP request sent, awaiting response... 301 Moved Permanently
Location: https://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=1e9662c93afb2ed6bd2e3f3cc38771a7f01125e969f2a75b0e2254439d4a81d8 [following]
--2024-10-04 03:45:51-- https://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=1e9662c93afb2ed6bd2e3f3cc38771a7f01125e969f2a75b0e2254439d4a81d8
Connecting to assets.nagios.com (assets.nagios.com)|45.79.49.120|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 2065473 (2.0M) [application/x-gzip]
Saving to: '6kqcx'

6kqcx 100%[=====] 1.97M 7.32MB/s i
2024-10-04 03:45:51 (7.32 MB/s) - '6kqcx' saved [2065473/2065473]
[ec2-user@ip-172-31-39-90 downloads]$ |
```

Wget <http://nagios-plugins.org/download/nagios-plugins-2.0.3.tar.gz>

```
[ec2-user@ip-172-31-39-90 downloads]$ wget http://nagios-plugins.org/download/nagios-plugins-2.4.11.tar.gz
--2024-10-04 03:47:43-- http://nagios-plugins.org/download/nagios-plugins-2.4.11.tar.gz
Resolving nagios-plugins.org (nagios-plugins.org)... 45.56.123.251
Connecting to nagios-plugins.org (nagios-plugins.org)|45.56.123.251|:80... connected.
HTTP request sent, awaiting response... 200 OK
Length: 2753049 (2.6M) [application/x-gzip]
Saving to: 'nagios-plugins-2.4.11.tar.gz'

nagios-plugins-2.4.11.tar.gz 100%[=====] 2.62M
2024-10-04 03:47:44 (6.85 MB/s) - 'nagios-plugins-2.4.11.tar.gz' saved [2753049/2753049]
```

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

```
tar zxvf 6kqcx
```

```
[ec2-user@ip-172-31-39-90 downloads]$ tar zxvf 6kqcx
nagios-4.5.5/
nagios-4.5.5/.github/
nagios-4.5.5/.github/workflows/
nagios-4.5.5/.github/workflows/test.yml
nagios-4.5.5/.gitignore
nagios-4.5.5/CONTRIBUTING.md
nagios-4.5.5/Changelog
nagios-4.5.5/INSTALLING
nagios-4.5.5/LEGAL
nagios-4.5.5/LICENSE
nagios-4.5.5/Makefile.in
nagios-4.5.5/README.md
nagios-4.5.5/THANKS
nagios-4.5.5/UPGRADING
nagios-4.5.5/aclocal.m4
nagios-4.5.5/autoconf-macros/
nagios-4.5.5/autoconf-macros/.gitignore
nagios-4.5.5/autoconf-macros/CHANGELOG.md
nagios-4.5.5/autoconf-macros/LICENSE
```

```
[ec2-user@ip-172-31-39-90 downloads]$ cd nagios-4.5.5
```

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

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

```
[ec2-user@ip-172-31-39-90 nagios-4.5.5]$./configure --with-command-group=nagcmd
checking for a BSD-compatible install... /usr/bin/install -c
checking build system type... x86_64-pc-linux-gnu
checking host system type... x86_64-pc-linux-gnu
checking for gcc... gcc
checking whether the C compiler works... yes
checking for C compiler default output file name... a.out
checking for suffix of executables...
checking whether we are cross compiling... no
checking for suffix of object files... o
checking whether the compiler supports GNU C... yes
checking whether gcc accepts -g... yes
checking for gcc option to enable C11 features... none needed
checking whether make sets $(MAKE)... yes
checking whether ln -s works... yes
checking for strip... /usr/bin/strip
checking for sys/wait.h that is POSIX.1 compatible... yes
checking for stdio.h... yes
```

error:

```
 checking for Kerberos include files... configure: WARNING: could not find include files
 checking for pkg-config... pkg-config
 checking for SSL headers... configure: error: Cannot find ssl headers
```

```
checking for SSL headers... configure: error: Cannot find ssl headers
[ec2-user@ip-172-31-39-90 nagios-4.5.5]$ sudo yum install openssl-devel
Last metadata expiration check: 0:16:59 ago on Fri Oct 4 03:35:02 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
=====
Install 1 Package

Total download size: 3.0 M
Installed size: 4.7 M
Is this ok [y/N]: y
Downloading Packages:
openssl-devel-3.0.8-1.amzn2023.0.14.x86_64.rpm 19 MB/s | 3.0 MB 00:00
=====
Total
Running transaction check
Transaction check succeeded.
Running transaction test
Transaction test succeeded.
Running transaction
 Preparing : 1/1
 Installing : openssl-devel-1:3.0.8-1.amzn2023.0.14.x86_64 1/1
 Running scriptlet: openssl-devel-1:3.0.8-1.amzn2023.0.14.x86_64 1/1
 Verifying : openssl-devel-1:3.0.8-1.amzn2023.0.14.x86_64 1/1
=====
Installed:
 openssl-devel-1:3.0.8-1.amzn2023.0.14.x86_64
=====
Complete!
```

```
[ec2-user@ip-172-31-39-90 nagios-4.5.5]$./configure --with-command-group=nagcmd
checking for a BSD-compatible install... /usr/bin/install -c
checking build system type... x86_64-pc-linux-gnu
checking host system type... x86_64-pc-linux-gnu
checking for gcc... gcc
checking whether the C compiler works... yes
checking for C compiler default output file name... a.out
checking for suffix of executables...
checking whether we are cross compiling... no
checking for suffix of object files... o
checking whether the compiler supports GNU C... yes
checking whether gcc accepts -g... yes
checking for gcc option to enable C11 features... none needed
checking whether make sets $(MAKE)... yes
checking whether ln -s works... yes
checking for strip... /usr/bin/strip
checking for sys/wait.h that is POSIX.1 compatible... yes
checking for stdio.h... yes
checking for stdlib.h... yes
checking for string.h... yes
checking for inttypes.h... yes
checking for stdint.h... yes
checking for strings.h... yes
checking for sys/stat.h... yes
checking for sys/types.h... yes
checking for unistd.h... yes
checking for arpa/inet.h... yes
checking for ctype.h... yes
checking for dirent.h... yes
checking for errno.h... yes
checking for fcntl.h... yes
checking for getopt.h... yes
checking for grp.h... yes
```

```
Creating sample config files in sample-config/ ...

*** 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.
```

12. Compile the source code.

```
make all
```

```
*** Compile finished ***

If the main program and CGIs compiled without any errors, you
can continue with testing or installing Nagios as follows (type
'make' without any arguments for a list of all possible options):

make test
 - This runs the test suite

make install
 - This installs the main program, CGIs, and HTML files

make install-init
 - This installs the init script in /lib/systemd/system

make install-daemoninit
 - This will initialize the init script
 in /lib/systemd/system
```

```
*** Support Notes ****
```

If you have questions about configuring or running Nagios,  
please make sure that you:

- Look at the sample config files
- Read the documentation on the Nagios Library at:  
<https://library.nagios.com>

before you post a question to one of the mailing lists.  
Also make sure to include pertinent information that could  
help others help you. This might include:

- What version of Nagios you are using
- What version of the plugins you are using
- Relevant snippets from your config files
- Relevant error messages from the Nagios log file

For more information on obtaining support for Nagios, visit:

<https://support.nagios.com>

```

```

Enjoy.

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
```

14. Edit the config file and change the email address.

```
sudo nano /usr/local/nagios/etc/objects
```

```
[ec2-user@ip-172-31-39-90 nagios-4.5.5]$ sudo make install
sudo make install-init
sudo make install-config
sudo make install-commandmode
cd ./base && make install
make[1]: Entering directory '/home/ec2-user/downloads/nagios-4.5.5/base'
/usr/bin/install -c -m 775 -o nagios -g nagios -d /usr/local/nagios/bin
/usr/bin/install -c -s -m 774 -o nagios -g nagios nagios /usr/local/nagios/bin
/usr/bin/install -c -s -m 774 -o nagios -g nagios nagios /usr/local/nagios/bin
make[1]: Leaving directory '/home/ec2-user/downloads/nagios-4.5.5/base'
cd ./cgi && make install
make[1]: Entering directory '/home/ec2-user/downloads/nagios-4.5.5/cgi'
make install-basic
make[2]: Entering directory '/home/ec2-user/downloads/nagios-4.5.5/cgi'
/usr/bin/install -c -m 775 -o nagios -g nagios -d /usr/local/nagios/sbin
for file in *.cgi; do \
 /usr/bin/install -c -s -m 775 -o nagios -g nagios $file /usr/local/nagios/sbin;
done
make[2]: Leaving directory '/home/ec2-user/downloads/nagios-4.5.5/cgi'
make[1]: Leaving directory '/home/ec2-user/downloads/nagios-4.5.5/cgi'
cd ./html && make install
make[1]: Entering directory '/home/ec2-user/downloads/nagios-4.5.5/html'
```

```
*** Main program, CGIs and HTML files installed ***
```

```
You can continue with installing Nagios as follows (type 'make' without any arguments for a list of all possible options):
```

```
make install-init
- This installs the init script in /lib/systemd/system

make install-commandmode
- This installs and configures permissions on the directory for holding the external command file

make install-config
- This installs sample config files in /usr/local/nagios/etc
```

```
*** Config files installed ***
```

```
Remember, these are *SAMPLE* config files. You'll need to read the documentation for more information on how to actually define services, hosts, etc. to fit your particular needs.
```

```
/usr/bin/install -c -m 775 -o nagios -g nagcmd -d /usr/local/nagios/var/rw
chmod g+s /usr/local/nagios/var/rw
```

```
*** External command directory configured ***
```

```
cd nagios-plugins-2.4.11
```

```
[ec2-user@ip-172-31-39-90 nagios-plugins-2.4.11]$./configure
--with-nagios-user=nagios --with-nagios-group=nagios
```

```
[ec2-user@ip-172-31-39-90 downloads]$ cd nagios-plugins-2.4.11
[ec2-user@ip-172-31-39-90 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
checking build system type... x86_64-pc-linux-gnu
checking host system type... x86_64-pc-linux-gnu
checking for gcc... gcc
checking whether the C compiler works... yes
checking for C compiler default output file name... a.out
checking for suffix of executables...
checking whether we are cross compiling... no
checking for suffix of object files... o
checking whether we are using the GNU C compiler... yes
checking whether gcc accepts -g... yes
checking for gcc option to accept ISO C89... none needed
checking whether gcc understands -c and -o together... yes
checking whether make supports the include directive... yes (GNU style)
checking dependency style of gcc... gcc3
checking how to run the C preprocessor... gcc -E
checking for grep that handles long lines and -e... /usr/bin/grep
checking for egrep... /usr/bin/grep -E
checking for Minix Amsterdam compiler... no
checking for ar... ar
```

## 15. Configure the web interface.

```
sudo make install-webconf
```

```
[ec2-user@ip-172-31-39-90 nagios-4.5.5]$ sudo make install-webconf
/usr/bin/install -c -m 644 sample-config/httpd.conf /etc/httpd/conf.d/nagios.conf
if [0 -eq 1]; then \
 ln -s /etc/httpd/conf.d/nagios.conf /etc/apache2/sites-enabled/nagios.conf; \
fi
*** Nagios/Apache conf file installed ***
```

## 16. 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-172-31-39-90 nagios-4.5.5]$ sudo htpasswd -c /usr/local/nagios/etc/htpasswd.users nagiosadmin
New password:
Re-type new password:
Adding password for user nagiosadmin
```

## 17. Restart Apache

```
sudo service httpd restart
```

```
[ec2-user@ip-172-31-39-90 nagios-4.5.5]$ sudo service httpd restart
Redirecting to /bin/systemctl restart httpd.service
```

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

```
cd ~/downloads
```

```
tar zxvf nagios-plugins-2.0.3.tar.gz
```

```
[ec2-user@ip-172-31-39-90 nagios-4.5.5]$ cd ~/downloads
[ec2-user@ip-172-31-39-90 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/lmain.sh
nagios-plugins-2.4.11/build-aux/missing
nagios-plugins-2.4.11/build-aux/mkinstalldirs
nagios-plugins-2.4.11/build-aux/depcomp
nagios-plugins-2.4.11/build-aux/snippet/
nagios-plugins-2.4.11/build-aux/snippet/_Noreturn.h
nagios-plugins-2.4.11/build-aux/snippet/arg-nonnull.h
nagios-plugins-2.4.11/build-aux/snippet/c++defs.h
nagios-plugins-2.4.11/build-aux/snippet/warn-on-use.h
nagios-plugins-2.4.11/build-aux/test-driver
nagios-plugins-2.4.11/config_test/
nagios-plugins-2.4.11/config_test/Makefile
nagios-plugins-2.4.11/config_test/run_tests
nagios-plugins-2.4.11/config_test/child_test.c
nagios-plugins-2.4.11/gl/
nagios-plugins-2.4.11/gl/m4/
```

20. Start Nagios

Add Nagios to the list of system services

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

```
(error)
```

```
[ec2-user@ip-172-31-39-90 nagios-plugins-2.4.11]$ sudo chkconfig --add nagios
error reading information on service nagios: No such file or directory
```

```
sudo chkconfig nagios on
```

Verify the sample configuration files

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

```
ec2-user@ip-172-31-39-90:~ $ sudo /usr/local/nagios/bin/nagios -v /usr/local/nagios/etc/nagios.cfg

Nagios Core 4.5.5
Copyright (c) 2009-present Nagios Core Development Team and Community Contributors
Copyright (c) 1999-2009 Ethan Galstad
Last Modified: 2024-09-17
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.
 Checked 0 host escalations.
 Checked 0 service escalations.
Checking for circular paths...
 Checked 1 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
```

```
ec2-user@ip-172-31-39-90:~/ $ sudo /usr/local/nagios/bin/nagios -v /usr/local/nagios/etc/nagios.cfg

Nagios Core 4.5.5
Copyright (c) 2009-present Nagios Core Development Team and Community Contributors
Copyright (c) 1999-2009 Ethan Galstad
Last Modified: 2024-09-17
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.
 Checked 0 host escalations.
 Checked 0 service escalations.
Checking for circular paths...
 Checked 1 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
```

```
sudo service nagios start
[ec2-user@ip-172-31-39-90 ~]$ sudo service nagios start
Redirecting to /bin/systemctl start nagios.service
```

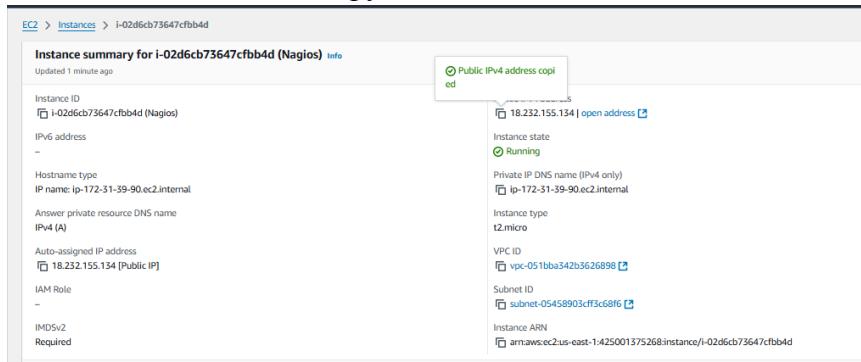
## 21. Check the status of Nagios

```
sudo systemctl status nagios
```

```
[ec2-user@ip-172-31-39-90 ~]$ sudo systemctl status nagios
● nagios.service - Nagios Core 4.5.5
 Loaded: loaded (/usr/lib/systemd/system/nagios.service; disabled; preset: disabled)
 Active: active (running) since Fri 2024-10-04 04:14:29 UTC; 1min 41s ago
 Docs: https://www.nagios.org/documentation
Process: 75298 ExecStartPre=/usr/local/nagios/bin/nagios -v /usr/local/nagios/etc/nagios.cfg (code=exited, status=0/SUCCESS)
Process: 75299 ExecStart=/usr/local/nagios/bin/nagios -d /usr/local/nagios/etc/nagios.cfg (code=exited, status=0/SUCCESS)
Main PID: 75300 (nagios)
 Tasks: 6 (limit: 1112)
 Memory: 5.6M
 CPU: 89ms
 CGroup: /system.slice/nagios.service
 ├─75300 /usr/local/nagios/bin/nagios -d /usr/local/nagios/etc/nagios.cfg
 ├─75301 /usr/local/nagios/bin/nagios --worker /usr/local/nagios/var/rw/nagios.qh
 ├─75302 /usr/local/nagios/bin/nagios --worker /usr/local/nagios/var/rw/nagios.qh
 ├─75303 /usr/local/nagios/bin/nagios --worker /usr/local/nagios/var/rw/nagios.qh
 ├─75304 /usr/local/nagios/bin/nagios --worker /usr/local/nagios/var/rw/nagios.qh
 └─75305 /usr/local/nagios/bin/nagios -d /usr/local/nagios/etc/nagios.cfg

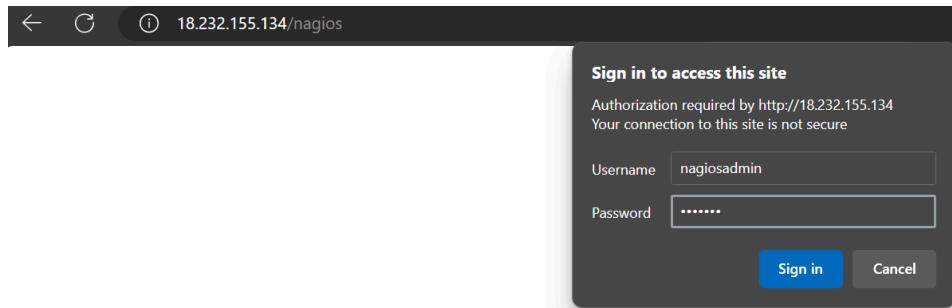
Oct 04 04:14:29 ip-172-31-39-90.ec2.internal nagios[75300]: wproc: Successfully registered manager as @wproc with query handler
Oct 04 04:14:29 ip-172-31-39-90.ec2.internal nagios[75300]: wproc: Registry request: name=Core Worker 75301;pid=75301
Oct 04 04:14:29 ip-172-31-39-90.ec2.internal nagios[75300]: wproc: Registry request: name=Core Worker 75302;pid=75302
Oct 04 04:14:29 ip-172-31-39-90.ec2.internal nagios[75300]: wproc: Registry request: name=Core Worker 75304;pid=75304
Oct 04 04:14:29 ip-172-31-39-90.ec2.internal nagios[75300]: wproc: Registry request: name=Core Worker 75303;pid=75303
Oct 04 04:14:29 ip-172-31-39-90.ec2.internal nagios[75300]: Successfully launched command file worker with pid 75305
Oct 04 04:14:29 ip-172-31-39-90.ec2.internal nagios[75300]: HOST ALERT: localhost;DOWN;SOFT;1;(No output on stdout) stderr: execvp(<
Oct 04 04:15:06 ip-172-31-39-90.ec2.internal nagios[75300]: SERVICE ALERT: localhost;Current Load;CRITICAL;HARD;1;(No output on std
Oct 04 04:15:29 ip-172-31-39-90.ec2.internal nagios[75300]: HOST ALERT: localhost;DOWN;SOFT;2;(No output on stdout) stderr: execvp(<
Oct 04 04:15:44 ip-172-31-39-90.ec2.internal nagios[75300]: SERVICE ALERT: localhost;Current Users;CRITICAL;HARD;1;(No output on std
[ec2-user@ip-172-31-39-90 ~]$ |
```

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



## 23. Open up your browser and look for [http://<your\\_public\\_ip\\_address>/nagios](http://<your_public_ip_address>/nagios)

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



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

The screenshot shows the Nagios Core 4.5.5 dashboard. At the top right, it displays "Nagios® Core™ Version 4.5.5" and the date "September 17, 2024". A green checkmark indicates "Daemon running with PID 75300". The left sidebar contains navigation links for General, Current Status, Problems, Reports, and System. The main content area includes a "Get Started" section with bullet points about monitoring infrastructure, changing the look, extending with addons, and getting support. It also features "Quick Links" to Nagios Library, Labs, Exchange, Support, and the official websites. Below these are sections for "Latest News" and "Don't Miss...". A "Page Tour" button is located at the bottom right.

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

## Conclusion:

**Aim:** To perform Port, Service monitoring, and Windows/Linux server monitoring using Nagios.

### Theory:

#### Port and Service Monitoring

Port and service monitoring in Nagios involves checking the availability and responsiveness of network services running on specific ports. This ensures that critical services (like HTTP, FTP, or SSH) are operational. Nagios uses plugins to ping the ports and verify whether services are up and responding as expected, allowing administrators to be alerted in case of outages.

#### Windows/Linux Server Monitoring

Windows/Linux server monitoring with Nagios entails tracking the performance and health of servers running these operating systems. It includes monitoring metrics such as CPU usage, memory consumption, disk space, and system logs. Nagios employs various plugins to gather data, enabling administrators to ensure optimal performance, identify potential issues, and maintain uptime across their server infrastructure.

#### Prerequisites:

AWS Academy or Personal account.

Nagios Server running on Amazon Linux Machine. (Refer Experiment No 9)

#### Monitoring Using Nagios:

**Step 1:** To Confirm Nagios is running on the server side Perform the following command on your Amazon Linux Machine (Nagios-host).

**sudo systemctl status nagios**

```
[ec2-user@ip-172-31-39-90 ~]$ sudo systemctl status nagios
● nagios.service - Nagios Core 4.5.5
 Loaded: loaded (/usr/lib/systemd/system/nagios.service; disabled; preset: disabled)
 Active: active (running) since Fri 2024-10-04 04:14:29 UTC; 9min ago
 Docs: https://www.nagios.org/documentation
 Process: 75298 ExecStartPre=/usr/local/nagios/bin/nagios -v /usr/local/nagios/etc/nagios.cfg (code=exited, status=0/SUCCESS)
 Process: 75299 ExecStart=/usr/local/nagios/bin/nagios -d /usr/local/nagios/etc/nagios.cfg (code=exited, status=0/SUCCESS)
 Main PID: 75300 (nagios)
 Tasks: 6 (limit: 1112)
 Memory: 5.6M
 CPU: 164ms
 CGroup: /system.slice/nagios.service
 ├─75300 /usr/local/nagios/bin/nagios -d /usr/local/nagios/etc/nagios.cfg
 ├─75301 /usr/local/nagios/bin/nagios --worker /usr/local/nagios/var/rw/nagios.qh
 ├─75302 /usr/local/nagios/bin/nagios --worker /usr/local/nagios/var/rw/nagios.qh
 ├─75303 /usr/local/nagios/bin/nagios --worker /usr/local/nagios/var/rw/nagios.qh
 ├─75304 /usr/local/nagios/bin/nagios --worker /usr/local/nagios/var/rw/nagios.qh
 └─75305 /usr/local/nagios/bin/nagios -d /usr/local/nagios/etc/nagios.cfg

Oct 04 04:17:36 ip-172-31-39-90.ec2.internal nagios[75300]: SERVICE ALERT: localhost;Root Partition;CRITICAL;HARD;1;(No output on std>
Oct 04 04:18:14 ip-172-31-39-90.ec2.internal nagios[75300]: SERVICE ALERT: localhost;SSH;CRITICAL;HARD;1;(No output on stdout) stderr>
Oct 04 04:18:29 ip-172-31-39-90.ec2.internal nagios[75300]: HOST ALERT: localhost;DOWN;SOFT;5;(No output on stdout) stderr: execvp(>
Oct 04 04:18:51 ip-172-31-39-90.ec2.internal nagios[75300]: SERVICE ALERT: localhost;Swap Usage;CRITICAL;HARD;1;(No output on std>
Oct 04 04:19:29 ip-172-31-39-90.ec2.internal nagios[75300]: SERVICE ALERT: localhost;Total Processes;CRITICAL;HARD;1;(No output on s>
Oct 04 04:19:29 ip-172-31-39-90.ec2.internal nagios[75300]: HOST ALERT: localhost;DOWN;SOFT;6;(No output on stdout) stderr: execvp(>
Oct 04 04:20:29 ip-172-31-39-90.ec2.internal nagios[75300]: HOST ALERT: localhost;DOWN;SOFT;7;(No output on stdout) stderr: execvp(>
Oct 04 04:21:29 ip-172-31-39-90.ec2.internal nagios[75300]: HOST ALERT: localhost;DOWN;SOFT;8;(No output on stdout) stderr: execvp(>
Oct 04 04:22:29 ip-172-31-39-90.ec2.internal nagios[75300]: HOST ALERT: localhost;DOWN;SOFT;9;(No output on stdout) stderr: execvp(>
Oct 04 04:23:29 ip-172-31-39-90.ec2.internal nagios[75300]: HOST ALERT: localhost;DOWN;HARD;10;(No output on stdout) stderr: execvp(>
lines 1-28/28 (END)
```

You can now proceed if you get the above message/output.

**Step 2:** Now Create a new EC2 instance. Name: Nagios-client, AMI: Ubuntu Instance Type: t2.micro.

The screenshot shows the AWS Lambda console interface. At the top, there's a search bar with the placeholder "Search our full catalog including 1000s of application and OS images". Below it, there are two tabs: "Recents" and "Quick Start", with "Quick Start" being the active tab. On the left, there's a sidebar with a "Create function" button and several recent projects listed. The main area displays a list of available operating systems (OSes) for creating a Lambda function. The "Ubuntu" option is highlighted with a blue border. Other options include Amazon Linux, macOS, Windows, Red Hat, and SUSE Linux. To the right of the OS list, there's a search icon and a link to "Browse more AMIs", which includes AMIs from AWS Marketplace and the Community.

**For Key pair :** Click on create key and make key of type RSA with extension .pem . Key will be downloaded to your local machine.

Now select that key in key pair if you already have key with type RSA and extension .pem no need to create new key but you must have that key downloaded.

The screenshot shows the AWS Lambda console interface. The "Instance type" section is open, showing the "t2.micro" option selected. It provides details about the instance: Family: t2, 1 vCPU, 1 GiB Memory, Current generation: true, Free tier eligible. It also lists On-Demand Windows base pricing: 0.0162 USD per Hour, On-Demand SUSE base pricing: 0.0116 USD per Hour, On-Demand RHEL base pricing: 0.026 USD per Hour, and On-Demand Linux base pricing: 0.0116 USD per Hour. A note at the bottom states "Additional costs apply for AMIs with pre-installed software". The "Key pair (login)" section is also visible, with a note that you can use a key pair to securely connect to your instance. It shows a dropdown menu where "server" is selected, and a "Create new key pair" button is available.

Select the Existing Security Group and select the Security Group that we have created in Experiment no 9 or the same one you have used for the Nagios server (Nagios-host).

▼ Network settings [Info](#)

[Edit](#)

Network | [Info](#)  
vpc-051bba342b3626898

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

Common security groups [Info](#)  
Select security groups ▾

Nagios sg-0b1355e80625c05ee X  
VPC: vpc-051bba342b3626898

Compare security group rules

Security groups that you add or remove here will be added to or removed from all your network interfaces.

**Step 3:** Now After creating the EC2 Instance click on connect and then copy the command which is given as example in the SSH Client section .

Now open the terminal in the folder where your key(RSA key with .pem) is located. and paste that copied command.

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

Instance ID  
 i-06e099f1c55d0ec8d (Nagios-client)

1. Open an SSH client.  
2. Locate your private key file. The key used to launch this instance is server.pem  
3. Run this command, if necessary, to ensure your key is not publicly viewable.  
 chmod 400 "server.pem"  
4. Connect to your instance using its Public DNS:  
 ec2-54-157-1-59.compute-1.amazonaws.com

Example:  
 ssh -i "server.pem" ubuntu@ec2-54-157-1-59.compute-1.amazonaws.com

Successfully connected to the instance.

```
PS C:\Users\Sadneya\downloads> ssh -i "server.pem" ec2-user@ec2-3-81-91-101.compute-1.amazonaws.com
The authenticity of host 'ec2-3-81-91-101.compute-1.amazonaws.com (3.81.91.101)' can't be established.
ED25519 key fingerprint is SHA256:6UVLjB6FbGB7A92sIEobs4886tozb5yML0ekn5Xzfrw.
This host key is known by the following other names/addresses:
 C:\Users\Sadneya\.ssh\known_hosts:68: ec2-18-232-155-134.compute-1.amazonaws.com
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added 'ec2-3-81-91-101.compute-1.amazonaws.com' (ED25519) to the list of known hosts.

#_
__ #####_ Amazon Linux 2023
~~ _\#\#\#__
~~ \#\#\|_
~~ \#/ ___ https://aws.amazon.com/linux/amazon-linux-2023
~~ V~`-->
~~ /_
~~ .-_-/_/
~~ /_/_/
~/m/`_
Last login: Fri Oct 4 03:36:10 2024 from 125.99.93.18
```

## Now perform all the commands on the Nagios-host till step 10

**Step 4:** Now on the server Nagios-host run the following command.

```
ps -ef | grep nagios
```

```
[ec2-user@ip-172-31-39-90 ~]$ ps -ef | grep nagios
ec2-user 2377 2350 0 09:15 pts/0 00:00:00 grep --color=auto nagios
[ec2-user@ip-172-31-39-90 ~]$
```

**Step 5:** Now Become root user and create root directories.

```
sudo su
```

```
mkdir /usr/local/nagios/etc/objects/monitorhosts
```

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

```
[ec2-user@ip-172-31-39-90 ~]$ sudo su
[root@ip-172-31-39-90 ec2-user]# mkdir /usr/local/nagios/etc/objects/monitorhosts
[root@ip-172-31-39-90 ec2-user]# mkdir /usr/local/nagios/etc/objects/monitorhosts/linuxhosts
[root@ip-172-31-39-90 ~]#
```

**Step 6:** Copy the sample localhost.cfg to linuxhost.cfg by running the following command.(Below command should come in one line see screenshot below)

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

```
[root@ip-172-31-39-90 ec2-user]# cp /usr/local/nagios/etc/objects/localhost.cfg /usr/local/nagios/etc/objects/monitorhosts/linuxhosts/linuxserver.cfg
```

**Step 7:** Open `linuxserver.cfg` using nano and make the following changes in all positions?everywhere in file.

```
[root@ip-172-31-39-90 ec2-user]# nano /usr/local/nagios/etc/objects/monitorhosts/linuxhosts/linuxserver.cfg
[root@ip-172-31-39-90 ec2-user]# |
```

## Change `hostname` to `linuxserver`

Change `address` to the public IP of your Linux client.

Set hostgroup name to **linux-servers1**.

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

```
Windows PowerShell x root@ip-172-31-39-90:/home, ~ x ubuntu@ip-172-31-42-255: ~ x + -
GNU nano 5.8 /usr/local/nagios/etc/objects/monitorhosts/linuxhosts/linuxserver.cfg Modified
#####
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 linuxserver
 alias localhost
 address 98.83.6.103
}

#####

#
HOST GROUP DEFINITION
#
| # Define an optional hostgroup for Linux machines

define hostgroup {

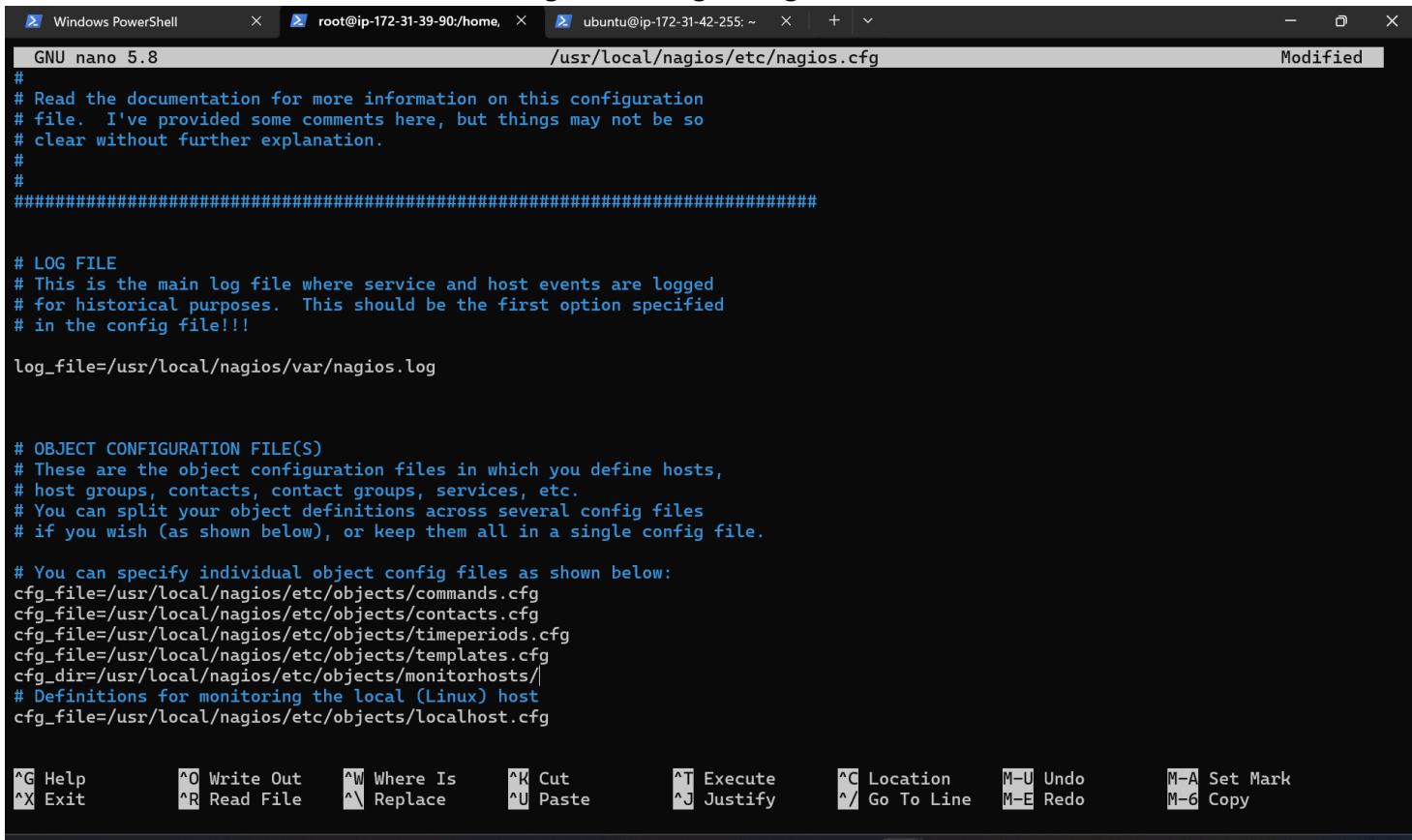
 hostgroup_name linux-servers1 ; The name of the hostgroup
 alias Linux Servers ; Long name of the group
 members localhost ; Comma separated list of hosts that belong to this group
}
}

^G Help ^O Write Out ^W Where Is ^K Cut ^T Execute ^C Location M-U Undo M-A Set Mark
^X Exit ^R Read File ^\ Replace ^U Paste ^J Justify ^/ Go To Line M-E Redo M-6 Copy
```

**Step 8:** Now update the Nagios config file .Add the following line in the file.

**Line to add : cfg\_dir=/usr/local/nagios/etc/objects/monitorhosts/**

Run the command : nano /usr/local/nagios/etc/nagios.cfg



The screenshot shows a terminal window with three tabs: "Windows PowerShell", "root@ip-172-31-39-90:/home", and "ubuntu@ip-172-31-42-255: ~". The current tab is "ubuntu@ip-172-31-42-255: ~" and it displays the contents of the "/usr/local/nagios/etc/nagios.cfg" file in the nano editor. The file contains configuration for Nagios, including sections for log files, object configuration files, and specific host monitoring definitions. At the bottom of the screen, there is a menu bar with various keyboard shortcuts for navigating and modifying the file.

```
GNU nano 5.8 /usr/local/nagios/etc/nagios.cfg Modified

#
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

^G Help ^O Write Out ^W Where Is ^K Cut ^T Execute ^C Location M-U Undo
^X Exit ^R Read File ^\ Replace ^U Paste ^J Justify ^/ Go To Line M-E Redo
 M-A Set Mark
 M-6 Copy
```

**Step 9:** Now Verify the configuration files by running the following commands.

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

```
[root@ip-172-31-39-90 ec2-user]# nano /usr/local/nagios/etc/nagios.cfg
[root@ip-172-31-39-90 ec2-user]# /usr/local/nagios/bin/nagios -v /usr/local/nagios/etc/nagios.cfg

Nagios Core 4.5.5
Copyright (c) 2009-present Nagios Core Development Team and Community Contributors
Copyright (c) 1999-2009 Ethan Galstad
Last Modified: 2024-09-17
License: GPL

Website: https://www.nagios.org
Reading configuration data...
 Read main config file okay...
Warning: Duplicate definition found for service 'HTTP' on host 'localhost' (config file '/usr/local/nagios/etc/objects/monitorhosts/linuxhosts/linuxserver.cfg', starting on line 152)
Warning: Duplicate definition found for service 'SSH' on host 'localhost' (config file '/usr/local/nagios/etc/objects/monitorhosts/linuxhosts/linuxserver.cfg', starting on line 138)
Warning: Duplicate definition found for service 'Swap Usage' on host 'localhost' (config file '/usr/local/nagios/etc/objects/monitorhosts/linuxhosts/linuxserver.cfg', starting on line 125)
Warning: Duplicate definition found for service 'Current Load' on host 'localhost' (config file '/usr/local/nagios/etc/objects/monitorhosts/linuxhosts/linuxserver.cfg', starting on line 112)
Warning: Duplicate definition found for service 'Total Processes' on host 'localhost' (config file '/usr/local/nagios/etc/objects/monitorhosts/linuxhosts/linuxserver.cfg', starting on line 100)
Warning: Duplicate definition found for service 'Root Partition' on host 'localhost' (config file '/usr/local/nagios/etc/objects/monitorhosts/linuxhosts/linuxserver.cfg', starting on line 72)
Warning: Duplicate definition found for service 'PING' on host 'localhost' (config file '/usr/local/nagios/etc/objects/monitorhosts/linuxhosts/linuxserver.cfg', starting on line 58)
 Read object config files okay...

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-172-31-39-90 ec2-user]# |
```

**Step 10:** Now restart the services of nagios by running the following command.

```
service nagios restart
```

```
[root@ip-172-31-39-90 ec2-user]# service nagios restart
Redirecting to /bin/systemctl restart nagios.service
```

**Step 11:** Now Go to the Nagios-client ssh terminal and update and install the packages by running the following command.

```
sudo apt update -y
```

```
ubuntu@ip-172-31-42-255:~$ sudo apt update -y
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://security.ubuntu.com/ubuntu noble-security/main Translation-en [83.9 kB]
Get:8 http://security.ubuntu.com/ubuntu noble-security/main amd64 c-n-f Metadata [4704 B]
Get:9 http://security.ubuntu.com/ubuntu noble-security/universe amd64 Packages [277 kB]
Get:10 http://security.ubuntu.com/ubuntu noble-security/universe Translation-en [117 kB]
Get:11 http://security.ubuntu.com/ubuntu noble-security/universe amd64 Components [8632 B]
Get:12 http://security.ubuntu.com/ubuntu noble-security/universe amd64 c-n-f Metadata [10.4 kB]
Get:13 http://security.ubuntu.com/ubuntu noble-security/multiverse amd64 Packages [10.9 kB]
Get:14 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble/universe Translation-en [5982 kB]
Get:15 http://security.ubuntu.com/ubuntu noble-security/multiverse Translation-en [2808 B]
Get:16 http://security.ubuntu.com/ubuntu noble-security/multiverse amd64 Components [208 B]
Get:17 http://security.ubuntu.com/ubuntu noble-security/multiverse amd64 c-n-f Metadata [344 B]
Get:18 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble/universe amd64 Components [3871 kB]
Get:19 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble/universe amd64 c-n-f Metadata [301 kB]

Get:34 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates/multiverse amd64 c-n-f Metadata [532 B]
Get:35 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-backports/main amd64 Components [208 B]
Get:36 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-backports/main amd64 c-n-f Metadata [112 B]
Get:37 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-backports/universe amd64 Packages [10.6 kB]
Get:38 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-backports/universe Translation-en [10.8 kB]
Get:39 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-backports/universe amd64 Components [17.6 kB]
Get:40 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-backports/universe amd64 c-n-f Metadata [1104 B]
Get:41 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-backports/restricted amd64 Components [216 B]
Get:42 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-backports/restricted amd64 c-n-f Metadata [116 B]
Get:43 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-backports/multiverse amd64 Components [212 B]
Get:44 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-backports/multiverse amd64 c-n-f Metadata [116 B]
Fetched 28.2 MB in 6s (4883 kB/s)
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
6 packages can be upgraded. Run 'apt list --upgradable' to see them.
```

```
sudo apt install gcc -y
```

```

ubuntu@ip-172-31-42-255:~$ sudo apt install gcc -y
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following additional packages will be installed:
 binutils binutils-common binutils-x86_64-linux-gnu cpp cpp-13 cpp-13-x86_64-linux-gnu cpp-x86_64-linux-gnu fontconfig-config
 fonts-dejavu-core fonts-dejavu-mono gcc-13 gcc-13-base gcc-13-x86_64-linux-gnu gcc-x86_64-linux-gnu libaom3 libasan8 libatomic1c
 libbinutils libc-dev-bin libc-devtools libc6-dev libgcc1-0 libcrypt-dev libctf-nobfd0 libctf0 libde265-0 libdeflate0
 libfontconfig1 libgcc-13-dev libgd3 libgomp1 libgprofng0 libheif-plugin-aomdec libheif-plugin-aomenc libheif-plugin-libde265
 libheif1 libhwasan0 libisl23 libitm1 libjbig0 libjpeg-turbo8 libjpeg8 liblrc4 liblsan0 libmpc3 libquadmath0 libsframe1
 libsharpyuv0 libtiff6 libtsan2 libubsan1 libwebp7 libxpm4 linux-libc-dev manpages-dev rpcsvc-proto
Suggested packages:
 binutils-doc gprofng-gui cpp-doc gcc-13-locales cpp-13-doc gcc-multilib make autoconf automake libtool flex bison gdb gcc-doc
 gcc-13-multilib gcc-13-doc gdb-x86_64-linux-gnu glibc-doc libgd-tools libheif-plugin-x265 libheif-plugin-ffmpegdec
 libheif-plugin-jpegdec libheif-plugin-jpegenc libheif-plugin-j2kdec libheif-plugin-j2kenc libheif-plugin-ravle
 libheif-plugin-svtenc
The following NEW packages will be installed:
 binutils binutils-common binutils-x86_64-linux-gnu cpp cpp-13 cpp-13-x86_64-linux-gnu cpp-x86_64-linux-gnu fontconfig-config
 fonts-dejavu-core fonts-dejavu-mono gcc gcc-13 gcc-13-base gcc-13-x86_64-linux-gnu gcc-x86_64-linux-gnu libaom3 libasan8
 libatomic1c libbinutils libc-dev-bin libc-devtools libc6-dev libgcc1-0 libcrypt-dev libctf-nobfd0 libctf0 libde265-0 libdeflate0
 libfontconfig1 libgcc-13-dev libgd3 libgomp1 libgprofng0 libheif-plugin-aomdec libheif-plugin-aomenc libheif-plugin-libde265
 libheif1 libhwasan0 libisl23 libitm1 libjbig0 libjpeg-turbo8 libjpeg8 liblrc4 liblsan0 libmpc3 libquadmath0 libsframe1
 libsharpyuv0 libtiff6 libtsan2 libubsan1 libwebp7 libxpm4 linux-libc-dev manpages-dev rpcsvc-proto
0 upgraded, 57 newly installed, 0 to remove and 6 not upgraded.
Need to get 62.8 MB of archives.
After this operation, 222 MB of additional disk space will be used.
Get:1 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble/main amd64 binutils-common amd64 2.42-4ubuntu2 [239 kB]
Get:2 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble/main amd64 libsframe1 amd64 2.42-4ubuntu2 [14.8 kB]
Get:3 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble/main amd64 libbinutils amd64 2.42-4ubuntu2 [572 kB]
Get:4 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble/main amd64 libctf-nobfd0 amd64 2.42-4ubuntu2 [97.1 kB]

```

```

Setting up gcc (4:13.2.0-0ubuntu1) ...
Setting up libheif-plugin-aomdec:amd64 (1.17.6-1ubuntu4) ...
Setting up libheif1:amd64 (1.17.6-1ubuntu4) ...
Setting up libheif-plugin-libde265:amd64 (1.17.6-1ubuntu4) ...
Setting up libheif-plugin-aomenc:amd64 (1.17.6-1ubuntu4) ...
Processing triggers for libc-bin (2.39-0ubuntu8.3) ...
Processing triggers for man-db (2.12.0-4build2) ...
Processing triggers for sgml-base (1.31) ...
Setting up libfontconfig1:amd64 (2.15.0-1.1ubuntu2) ...
Setting up libgd3:amd64 (2.3.3-9ubuntu5) ...
Setting up libc-devtools (2.39-0ubuntu8.3) ...
Processing triggers for libc-bin (2.39-0ubuntu8.3) ...
Scanning processes...
Scanning linux images...

Running kernel seems to be up-to-date.

No services need to be restarted.

No containers need to be restarted.

No user sessions are running outdated binaries.

No VM guests are running outdated hypervisor (qemu) binaries on this host.
ubuntu@ip-172-31-42-255:~|

```

**sudo apt install -y nagios-nrpe-server nagios-plugins**

```
ubuntu@ip-172-31-42-255:~$ sudo apt install -y nagios-nrpe-server nagios-plugins
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
Note, selecting 'monitoring-plugins' instead of 'nagios-plugins'
The following additional packages will be installed:
 libavahi-client3 libavahi-common-data libavahi-common3 libcurl2t64 libdbdilt64 libldb2 libmysqlclient21 libnet-snmp-perl libpq5
 libradcli4 libsmclient0 libsnmp-base libsnmp40t64 libtalloc2 libtdb1 libevent0t64 liburiparser1 libwbcclient0
 monitoring-plugins-basic monitoring-plugins-common monitoring-plugins-standard mysql-common python3-gpg python3-ldb
 python3-markdown python3-samba python3-talloc python3-tdb rpcbind samba-common samba-common-bin samba-dsdb-modules samba-libs
 smbclient snmp
Suggested packages:
 cups-common libcrypt-des-perl libdigest-hmac-perl libio-socket-inet6-perl snmp-mibs-downloader icinga2 nagios-plugins-contrib
 fping postfix | sendmail-bin | exim4-daemon-heavy | exim4-daemon-light qstat xinetd | inetd python-markdown-doc heimdal-clients
 python3-dnspython cifs-utils
The following NEW packages will be installed:
 libavahi-client3 libavahi-common-data libavahi-common3 libcurl2t64 libdbdilt64 libldb2 libmysqlclient21 libnet-snmp-perl libpq5
 libradcli4 libsmclient0 libsnmp-base libsnmp40t64 libtalloc2 libtdb1 libevent0t64 liburiparser1 libwbcclient0 monitoring-plugins
 monitoring-plugins-basic monitoring-plugins-common monitoring-plugins-standard mysql-common nagios-nrpe-server python3-gpg
 python3-ldb python3-markdown python3-samba python3-talloc python3-tdb rpcbind samba-common samba-common-bin samba-dsdb-modules
 samba-libs smbclient snmp
0 upgraded, 37 newly installed, 0 to remove and 6 not upgraded.
Need to get 16.1 MB of archives.
After this operation, 72.0 MB of additional disk space will be used.
Get:1 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble/universe amd64 nagios-nrpe-server amd64 4.1.0-1ubuntu3 [356 kB]
Get:2 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble/main amd64 rpcbind amd64 1.2.6-7ubuntu2 [46.5 kB]
Get:3 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble/main amd64 libavahi-common-data amd64 0.8-13ubuntu6 [29.7 kB]
Get:4 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble/main amd64 libavahi-common3 amd64 0.8-13ubuntu6 [23.3 kB]
Get:5 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble/main amd64 libavahi-client3 amd64 0.8-13ubuntu6 [26.8 kB]
Get:6 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates/main amd64 libcurl2t64 amd64 2.4.7-1.2ubuntu7.3 [272 kB]
Get:7 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble/main amd64 libdbdilt64 amd64 0.9.0-6.1build1 [25.7 kB]

Creating config file /etc/nagios-plugins/config/netware.cfg with new version

Creating config file /etc/nagios-plugins/config/nt.cfg with new version

Creating config file /etc/nagios-plugins/config/pgsql.cfg with new version

Creating config file /etc/nagios-plugins/config/radius.cfg with new version

Creating config file /etc/nagios-plugins/config/rpc-nfs.cfg with new version

Creating config file /etc/nagios-plugins/config/snmp.cfg with new version
Setting up monitoring-plugins (2.3.5-1ubuntu3) ...
Setting up libldb2:amd64 (2:2.8.0+samba4.19.5+dfsg-4ubuntu9) ...
Setting up libavahi-client3:amd64 (0.8-13ubuntu6) ...
Setting up samba-libs:amd64 (2:4.19.5+dfsg-4ubuntu9) ...
Setting up python3-ldb (2:2.8.0+samba4.19.5+dfsg-4ubuntu9) ...
Setting up samba-dsdb-modules:amd64 (2:4.19.5+dfsg-4ubuntu9) ...
Setting up libsmclient0:amd64 (2:4.19.5+dfsg-4ubuntu9) ...
Setting up libcurl2t64:amd64 (2.4.7-1.2ubuntu7.3) ...
Setting up python3-samba (2:4.19.5+dfsg-4ubuntu9) ...
Setting up smbclient (2:4.19.5+dfsg-4ubuntu9) ...
Setting up samba-common-bin (2:4.19.5+dfsg-4ubuntu9) ...
Processing triggers for man-db (2.12.0-4build2) ...
Processing triggers for libc-bin (2.39-0ubuntu8.3) ...
Scanning processes...
Scanning linux images...

Running kernel seems to be up-to-date.

No services need to be restarted.

No containers need to be restarted.

No user sessions are running outdated binaries.

No VM guests are running outdated hypervisor (qemu) binaries on this host.
ubuntu@ip-172-31-42-255:~$ |
```

**Step 12:** Open nrpe.cfg file to make changes.Under allowed\_hosts, add your nagios host IP address.  
**sudo nano /etc/nagios/nrpe.cfg**

```
GNU nano 7.2 /etc/nagios/nrpe.cfg
#####
Sample NRPE Config File
#
Notes:
#
This is a sample configuration file for the NRPE daemon. It needs to be
located on the remote host that is running the NRPE daemon, not the host
from which the check_nrpe client is being executed.
#
#####

LOG FACILITY
The syslog facility that should be used for logging purposes.

log_facility=daemon

LOG FILE
If a log file is specified in this option, nrpe will write to
that file instead of using syslog.

#log_file=/var/log/nrpe.log

DEBUGGING OPTION
This option determines whether or not debugging messages are logged to the
syslog facility.
Values: 0=debugging off, 1=debugging on
```

```
GNU nano 7.2 /etc/nagios/nrpe.cfg *
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,::1,3.81.91.101

COMMAND ARGUMENT PROCESSING
This option determines whether or not the NRPE daemon will allow clients
```

**Step 13:** Now restart the NRPE server by this command.

```
sudo systemctl restart nagios-nrpe-server
```

```
ubuntu@ip-172-31-42-255:~$ sudo systemctl restart nagios-nrpe-server
```

**Step 14:** Now again check the status of Nagios by running this command on Nagios-host and also check httpd is active and run the command to active it.

```
sudo systemctl status nagios
```

```
[root@ip-172-31-39-90 ec2-user]#
sudo systemctl status nagios
● nagios.service - Nagios Core 4.5.5
 Loaded: loaded (/usr/lib/systemd/system/nagios.service; disabled; preset: disabled)
 Active: active (running) since Fri 2024-10-04 09:24:28 UTC; 9min ago
 Docs: https://www.nagios.org/documentation
 Process: 2725 ExecStartPre=/usr/local/nagios/bin/nagios -v /usr/local/nagios/etc/nagios.cfg (code=exited, status=0/SUCCESS)
 Process: 2727 ExecStart=/usr/local/nagios/bin/nagios -d /usr/local/nagios/etc/nagios.cfg (code=exited, status=0/SUCCESS)
 Main PID: 2729 (nagios)
 Tasks: 6 (limit: 1112)
 Memory: 4.2M
 CPU: 114ms
 CGroup: /system.slice/nagios.service
 ├─2729 /usr/local/nagios/bin/nagios -d /usr/local/nagios/etc/nagios.cfg
 ├─2730 /usr/local/nagios/bin/nagios --worker /usr/local/nagios/var/rw/nagios.qh
 ├─2731 /usr/local/nagios/bin/nagios --worker /usr/local/nagios/var/rw/nagios.qh
 ├─2732 /usr/local/nagios/bin/nagios --worker /usr/local/nagios/var/rw/nagios.qh
 ├─2733 /usr/local/nagios/bin/nagios --worker /usr/local/nagios/var/rw/nagios.qh
 └─2742 /usr/local/nagios/bin/nagios -d /usr/local/nagios/etc/nagios.cfg

Oct 04 09:30:28 ip-172-31-39-90.ec2.internal nagios[2729]: HOST ALERT: linuxserver;DOWN;SOFT;7;(No output on stdout) stderr>
Oct 04 09:31:28 ip-172-31-39-90.ec2.internal nagios[2729]: HOST ALERT: linuxserver;DOWN;SOFT;8;(No output on stdout) stderr>
Oct 04 09:32:28 ip-172-31-39-90.ec2.internal nagios[2729]: HOST ALERT: linuxserver;DOWN;SOFT;9;(No output on stdout) stderr>
Oct 04 09:33:28 ip-172-31-39-90.ec2.internal nagios[2729]: HOST NOTIFICATION: nagiosadmin;linuxserver;DOWN;notify-host-by-email>
Oct 04 09:33:28 ip-172-31-39-90.ec2.internal nagios[2729]: HOST ALERT: linuxserver;DOWN;HARD;10;(No output on stdout) stderr>
Oct 04 09:33:28 ip-172-31-39-90.ec2.internal nagios[2729]: wproc: NOTIFY job 6 from worker Core Worker 2732 is a non-check>
Oct 04 09:33:28 ip-172-31-39-90.ec2.internal nagios[2729]: wproc: host=linuxserver; service=(none); contact=nagiosadmin>
Oct 04 09:33:28 ip-172-31-39-90.ec2.internal nagios[2729]: wproc: early_timeout=0; exited_ok=1; wait_status=32512; error=>
Oct 04 09:33:28 ip-172-31-39-90.ec2.internal nagios[2729]: wproc: stderr line 01: /bin/sh: line 1: /bin/mail: No such file or directory>
Oct 04 09:33:28 ip-172-31-39-90.ec2.internal nagios[2729]: wproc: stderr line 02: /usr/bin/printf: write error: Broken pipe>
[root@ip-172-31-39-90 ec2-user]#
```

```
sudo systemctl status httpd
sudo systemctl start httpd
sudo systemctl enable httpd
```

```
[root@ip-172-31-39-90 ec2-user]# sudo systemctl status httpd
● httpd.service - The Apache HTTP Server
 Loaded: loaded (/usr/lib/systemd/system/httpd.service; enabled; preset: disabled)
 Drop-In: /usr/lib/systemd/system/httpd.service.d
 └─php-fpm.conf
 Active: active (running) since Fri 2024-10-04 09:45:14 UTC; 33min ago
 Docs: man:httpd.service(8)
 Main PID: 4146 (httpd)
 Status: "Total requests: 19; Idle/Busy workers 100/0;Requests/sec: 0.0095; Bytes served/sec: 70 B/sec"
 Tasks: 230 (limit: 1112)
 Memory: 17.3M
 CPU: 1.428s
 CGroup: /system.slice/httpd.service
 ├─4146 /usr/sbin/httpd -DFOREGROUND
 ├─4148 /usr/sbin/httpd -DFOREGROUND
 ├─4149 /usr/sbin/httpd -DFOREGROUND
 ├─4150 /usr/sbin/httpd -DFOREGROUND
 ├─4151 /usr/sbin/httpd -DFOREGROUND
 └─4533 /usr/sbin/httpd -DFOREGROUND

Oct 04 09:45:14 ip-172-31-39-90.ec2.internal systemd[1]: Stopped httpd.service - The Apache HTTP Server.
Oct 04 09:45:14 ip-172-31-39-90.ec2.internal systemd[1]: Starting httpd.service - The Apache HTTP Server...
Oct 04 09:45:14 ip-172-31-39-90.ec2.internal systemd[1]: Started httpd.service - The Apache HTTP Server.
Oct 04 09:45:14 ip-172-31-39-90.ec2.internal httpd[4146]: Server configured, listening on: port 80
[root@ip-172-31-39-90 ec2-user]# sudo systemctl start httpd
[root@ip-172-31-39-90 ec2-user]# sudo systemctl enable httpd
```

**Step 15:** Now to check Nagios dashboard go to <http://<Nagios-host ip>/nagios> .

The screenshot shows the Nagios Core dashboard. On the left, there's a vertical navigation bar with sections for General, Current Status, Reports, and System. The 'Current Status' section is expanded, showing options like Tactical Overview, Map, Hosts, Services, Host Groups, Service Groups, and Problems. The 'Reports' section includes Availability, Trends, Alerts, and Notifications. The 'System' section includes Comments, Downtime, Process Info, Performance Info, and Scheduling Queue.

The main content area features the Nagios Core logo at the top right. Below it, a message says "Daemon running with PID 2729". To the right of the message, the text "Nagios® Core™ Version 4.5.5" is displayed, along with the date "September 17, 2024" and a "Check for updates" link. The main area is divided into several boxes: "Get Started" (with a bulleted list of steps), "Latest News" (empty), "Don't Miss..." (empty), and "Quick Links" (listing Nagios Library, Labs, Exchange, Support, and org). At the bottom, there's a copyright notice and a license statement.

**Now Click on Hosts from left side panel**

Not secure | 3.81.91.101/nagios/

# Nagios®

**General**

- [Home](#)
- [Documentation](#)

**Current Status**

- [Tactical Overview](#)
- [Map](#)
- [Hosts](#)
- [Services](#)
- [Host Groups](#)
  - [Summary](#)
  - [Grid](#)
- [Service Groups](#)
  - [Summary](#)
  - [Grid](#)
- [Problems](#)
  - [Services](#)
  - [\(Unhandled\)](#)
  - [Hosts \(Unhandled\)](#)
  - [Network Outages](#)
- [Quick Search:](#)

**Reports**

- [Availability](#)
- [Trends](#)
- [Alerts](#)
  - [History](#)
  - [Summary](#)
  - [Histogram](#)
- [Notifications](#)
- [Event Log](#)

**System**

- [Comments](#)
- [Downtime](#)
- [Process Info](#)
- [Performance Info](#)
- [Scheduling Queue](#)

**Current Network Status**

Last Updated: Fri Oct 4 10:43:54 UTC 2024  
 Updated every 90 seconds  
 Nagios® Core™ 4.5.5 - www.nagios.org  
 Logged in as: nagiosadmin

**Host Status Totals**

| Up | Down | Unreachable | Pending |
|----|------|-------------|---------|
| 2  | 0    | 0           | 0       |

All Problems All Types  
0 2

**Service Status Totals**

| Ok | Warning | Unknown | Critical | Pending |
|----|---------|---------|----------|---------|
| 8  | 1       | 0       | 7        | 0       |

All Problems All Types  
8 16

**Host Status Details For All Host Groups**

Limit Results:  ▾

| Host        | Status | Last Check          | Duration     | Status Information                        |
|-------------|--------|---------------------|--------------|-------------------------------------------|
| linuxserver | UP     | 10-04-2024 10:41:21 | 0d 0h 2m 33s | PING OK - Packet loss = 0%, RTA = 1.23 ms |
| localhost   | UP     | 10-04-2024 10:43:13 | 0d 0h 3m 11s | PING OK - Packet loss = 0%, RTA = 0.06 ms |

Results 1 - 2 of 2 Matching Hosts

Page Tour

We can see our linuxserver now click on it we can see the host information.

The screenshot shows the Nagios web interface at [3.81.91.101/nagios/](http://3.81.91.101/nagios/). The left sidebar has sections for General, Current Status (Tactical Overview, Map, Hosts, Services, Host Groups, Service Groups, Problems), Reports (Availability, Trends, Alerts, History, Summary, Histogram, Notifications, Event Log), and System (Comments, Downtime, Process Info, Performance Info, Scheduling Queue). The main content area shows 'Host Information' for 'localhost (linuxserver)'. It includes a status bar with 'Last Updated: Fri Oct 4 10:44:12 UTC 2024', 'Updated every 90 seconds', 'Nagios® Core™ 4.5.5 - www.nagios.org', and 'Logged in as nagiosadmin'. To the right, there's a 'Host Commands' panel with various options like Locate host on map, Disable active checks, and Stop accepting passive checks. Below the host info is 'Host State Information' with details like 'Status: UP (for 0d 0h 2m 51s)', 'Performance Data: rta=1.227000ms;3000.000000;5000.000000;0.000000 pl=%;80;100;1/10 (HARD state)', and a table of active checks (all enabled). At the bottom is a 'Host Comments' section with a link to add a new comment.

The screenshot shows the Nagios web interface at [3.81.91.101/nagios/](http://3.81.91.101/nagios/). The left sidebar is identical to the previous screenshot. The main content area shows 'Current Network Status' with last updated time, version, and logon information. It includes 'Host Status Totals' (Up: 2, Down: 0, Unreachable: 0, Pending: 0) and 'Service Status Totals' (Ok: 10, Warning: 1, Unknown: 0, Critical: 5, Pending: 0). Below is a table titled 'Service Status Details For All Hosts' with columns for Host, Service, Status, Last Check, Duration, Attempt, and Status Information. The table lists services for 'linuxserver' and 'localhost', including CPU load, user counts, and various system checks. A note at the bottom says 'Results 1 - 16 of 16 Matching Services'.

**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.

### Theory:

#### AWS Lambda

A fully managed, serverless computing service where you run code without provisioning or managing servers. Lambda automatically scales your application based on the number of incoming requests or events, ensuring efficient resource utilization. You are only charged for the time your code is running, with no upfront cost, making it cost-effective for on-demand workloads.

#### Lambda Workflow

- **Create a Function:** Write the function code and define its handler (entry point). You can use the AWS Console, CLI, or upload a deployment package.
- **Set Event Sources:** Define how the function is triggered (e.g., when an object is uploaded to S3 or a DynamoDB table is updated).
- **Execution:** When triggered, Lambda runs your function, executes the logic, and automatically scales to handle the incoming event volume.
- **Scaling and Concurrency:** Lambda scales automatically by launching more instances of the function to handle simultaneous invocations. There are also options for configuring **reserved concurrency** to manage traffic.
- **Monitoring and Logging:** Lambda integrates with Amazon CloudWatch for logging and monitoring. Logs for each invocation are sent to CloudWatch, allowing you to track performance and troubleshoot errors.

#### AWS Lambda Functions

- **Python:** Great for quick development with its rich standard library and support for lightweight tasks.
- **Java:** Typically used for more complex, compute-intensive tasks. While it's robust, cold start times can be higher.
- **Node.js:** Excellent for I/O-bound tasks like handling APIs or streaming data, with fast startup times and efficient memory usage.

**Prerequisites:** AWS Personal/Academy Account

Name:Bhushan Mukund Kor

Academic Year:2024-2025

Division: D15C

Roll No: 28

### Steps To create the lambda function:

**Step 1:** Login to your AWS Personal/Academy Account. Open lambda and click on create function button.

The screenshot shows the AWS Lambda Functions list page. It displays five existing Lambda functions: "MainMonito", "ringFunction", "ModLabRole", "Create", and "Redshift". Each function entry includes its name, description, package type (Zip), runtime (Python 3.8), and last modified date (2 months ago). The "Create function" button is prominently displayed at the top right of the list.

**Step 2:** Now Give a name to your Lambda function, Select the language to use to write your function. Note that the console code editor supports only Node.js, Python, and Ruby. So will select Python 3.12 , Architecture as x86, and Execution role to Create a new role with basic Lambda permissions.

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 "Basic information" step shows the function name "anshi-lambda" entered. The "Runtime" step shows "Python 3.12" selected. The "Architecture" step shows "x86\_64" selected.

**Function overview** [Info](#)

[Throttle](#) [Copy ARN](#) [Actions ▾](#)

[Export to Application Composer](#) [Download ▾](#)

**Diagram** [Template](#)

**anshi-lambda**

**Layers** (0)

[+ Add trigger](#) [+ Add destination](#)

**Description**  
-

Last modified  
2 seconds ago

**Function ARN**  
arn:aws:lambda:us-east-1:708398963195:function:anshi-lambda

**Function URL** [Info](#)

[Code source](#) [Info](#)

[Upload from ▾](#)

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

Environment Go to Anything (Ctrl-P) lambda\_function Environment Variants

```
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
```

So See or Edit the basic settings go to configuration then click on edit general setting.

**General configuration** [Info](#)

[Edit](#)

|             |                                |                   |
|-------------|--------------------------------|-------------------|
| Description | Memory                         | Ephemeral storage |
| -           | 128 MB                         | 512 MB            |
| Timeout     | SnapStart <a href="#">Info</a> |                   |
| 0 min 3 sec | None                           |                   |

Here, you can enter a description and change Memory and Timeout. I've changed the Timeout period to 1 sec since that is sufficient for now.

**Basic settings** [Info](#)

**Description - optional**  
Basic settings

**Memory** [Info](#)  
Your function is allocated CPU proportional to the memory configured.  
128 MB  
Set memory to between 128 MB and 10240 MB.

**Ephemeral storage** [Info](#)  
You can configure up to 10 GB of ephemeral storage (/tmp) for your function. View pricing [View pricing](#)  
512 MB  
Set ephemeral storage (/tmp) to between 512 MB and 10240 MB.

**SnapStart** [Info](#)  
Reduce startup time by having Lambda cache a snapshot of your function after the function has initialized. To evaluate whether your function code is resilient to snapshot operations, review the [SnapStart compatibility considerations](#).  
None

Supported runtimes: Java 11, Java 17, Java 21.

**Timeout**  
0 min 1 sec

**Execution role**  
Choose a role that defines the permissions of your function. To create a custom role, go to the [IAM console](#).  
 Use an existing role  
 Create a new role from AWS policy templates

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

**Step 3:** Now Click on the Test tab then select Create a new event, give a name to the event and select Event Sharing to private, and select hello-world template.

Test event [Info](#)

To invoke your function without saving an event, configure the JSON event, then choose Test.

Test event action  Create new event  Edit saved event

Event name `our_event`

Maximum of 25 characters consisting of letters, numbers, dots, hyphens and underscores.

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

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

Template - optional `hello-world` [Format JSON](#)

Event JSON

```
1 * []
2 "key1": "value1",
3 "key2": "value2",
4 "key3": "value3"
5 []
```

**Step 4:** Now In Code section select the created event from the dropdown of test then click on test . You will see the below output.

Code source [Info](#)

File Edit Find View Go Tools Window [Test](#) Deploy

Configure test event Ctrl-Shift-C Execution result: x +

Test Event Name (unsaved) test event

- (unsaved) test event
- Private saved events
- our\_event**

Response

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

Code source [Info](#)

File Edit Find View Go Tools Window [Test](#) Deploy

Go to Anything (Ctrl-P) lambda\_function Environment Var Execution result: x +

Execution results

Test Event Name our\_event

Response

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

Function Logs

```
START RequestId: acba54af-dad3-4830-b390-c09f019d1192 Version: $LATEST
END RequestId: acba54af-dad3-4830-b390-c09f019d1192
REPORT RequestId: acba54af-dad3-4830-b390-c09f019d1192 Duration: 1.76 ms Billed Duration: 2 ms Memory Size: 128 MB Max Memory Used: 32 MB
```

Request ID acba54af-dad3-4830-b390-c09f019d1192

**Step 5:** You can edit your lambda function code. I have changed the code to display the new String.

The screenshot shows the AWS Lambda code editor interface. The file path is 'anshi-lambda / lambda\_function.py'. The code is as follows:

```
1 import json
2
3 def lambda_handler(event, context):
4 # TODO implement
5 new_string="This is a lambda function by Anshi and Sadneya"
6 return {
7 'statusCode': 200,
8 'body': json.dumps('Hello from Lambda!')
9 }
10
```

Now ctrl+s to save and click on deploy to deploy the changes.

The screenshot shows the AWS Lambda code editor interface after deployment. The file path is 'anshi-lambda / lambda\_function.py'. The code is as follows:

```
1 import json
2
3 def lambda_handler(event, context):
4 # Implementing the custom string in the response
5 new_string = "This is a lambda function by Anshi and Sadneya"
6 return {
7 'statusCode': 200,
8 'body': json.dumps(new_string) # Returning the custom string in the response
9 }
10
```

**Step 6:** Now click on the test and observe the output. We can see the status code 200 and your string output and function logs. On successful deployment.

The screenshot shows the AWS Lambda test results interface. The file path is 'anshi-lambda / lambda\_function.py'. The test event name is 'our\_event'. The response is:

```
{"statusCode": 200,
"body": "\"This is a lambda function by Anshi and Sadneya\""
}
```

The function logs show the execution details:

```
START RequestId: 501daf29-295f-46c5-b191-6fc0fc5fd69e Version: $LATEST
END RequestId: 501daf29-295f-46c5-b191-6fc0fc5fd69e
REPORT RequestId: 501daf29-295f-46c5-b191-6fc0fc5fd69e Duration: 1.94 ms Billed Duration: 2 ms Me
Request ID
501daf29-295f-46c5-b191-6fc0fc5fd69e|
```

**Conclusion:** In this experiment, we successfully created an AWS Lambda function and walked through its essential steps. After setting up the function with Python, we configured the basic settings, including adjusting the timeout to 1 second. We then created a test event, deployed the function, and validated the output. Additionally, we modified the Lambda function's code and redeployed it to observe the changes in real-time.

This practical experience demonstrated the simplicity and flexibility of AWS Lambda in creating serverless applications, allowing you to focus on code while AWS manages the infrastructure and scaling.

**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

**Theory:**

**AWS Lambda and S3 Integration:**

AWS Lambda allows you to execute code in response to various events, including those triggered by Amazon S3. When an object is added to an S3 bucket, it can trigger a Lambda function to execute, allowing for event-driven processing without managing servers.

**Workflow:**

**1. Create an S3 Bucket:**

- First, create an S3 bucket that will store the objects. This bucket will act as the trigger source for the Lambda function.

**2. Create the Lambda Function:**

- Set up a new Lambda function using AWS Lambda’s console. You can choose a runtime environment like Python, Node.js, or Java.
- Write code that logs a message like “An Image has been added” when triggered.

**3. Set Up Permissions:**

- Ensure that the Lambda function has the necessary permissions to access S3. You can do this by attaching an IAM role with policies that allow reading from the bucket and writing logs to CloudWatch.

**4. Configure S3 Trigger:**

- Link the S3 bucket to the Lambda function by setting up a trigger. Specify that the function should be triggered when an object is created in the bucket (e.g., when an image is uploaded).

**5. Test the Setup:**

- Upload an object (e.g., an image) to the S3 bucket to test the trigger. The Lambda function should execute and log the message “An Image has been added” in AWS CloudWatch Logs.

**Steps To create the lambda function:**

**Step 1:** Login to your AWS Personal account. Now open S3 from services and click on create S3 bucket.

Amazon S3

► 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    Directory buckets

General purpose buckets (1) [Info](#) [All AWS Regions](#)

Buckets are containers for data stored in S3.

Find buckets by name

| Name                                                    | AWS Region                      | IAM Access Analyzer                         | Creation date                        |
|---------------------------------------------------------|---------------------------------|---------------------------------------------|--------------------------------------|
| <a href="#">elasticbeanstalk-us-east-1-708398963195</a> | US East (N. Virginia) us-east-1 | <a href="#">View analyzer for us-east-1</a> | August 3, 2024, 22:25:17 (UTC+05:30) |

[Create bucket](#)

**Step 2:** Now Give a name to the Bucket, select general purpose project and deselect the Block public access and keep other this to default.

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](#)  
 lambda\_buche

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

Block Public Access settings for this bucket

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

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

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

**⚠ Turning off block all public access might result in this bucket and the objects within becoming public**  
AWS recommends that you turn on block all public access, unless public access is required for specific and verified use cases such as static website hosting.

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

| General purpose buckets                                                          |                                 | Directory buckets                           |                                       |   |   |                               |  |  |  |  |
|----------------------------------------------------------------------------------|---------------------------------|---------------------------------------------|---------------------------------------|---|---|-------------------------------|--|--|--|--|
| General purpose buckets (2) <a href="#">Info</a> <a href="#">All AWS Regions</a> |                                 |                                             |                                       |   |   | <a href="#">Create bucket</a> |  |  |  |  |
| Buckets are containers for data stored in S3.                                    |                                 |                                             |                                       |   |   |                               |  |  |  |  |
| <input type="text"/> Find buckets by name                                        |                                 |                                             |                                       |   |   |                               |  |  |  |  |
| Name                                                                             | AWS Region                      | IAM Access Analyzer                         | Creation date                         | < | 1 | >                             |  |  |  |  |
| <input type="radio"/> anshi-sadneya-lambda-bucket                                | US East (N. Virginia) us-east-1 | <a href="#">View analyzer for us-east-1</a> | October 5, 2024, 11:28:16 (UTC+05:30) |   |   |                               |  |  |  |  |
| <input type="radio"/> elasticbeanstalk-us-east-1-708398963195                    | US East (N. Virginia) us-east-1 | <a href="#">View analyzer for us-east-1</a> | August 3, 2024, 22:25:17 (UTC+05:30)  |   |   |                               |  |  |  |  |

**Step 3:** Open lambda console and click on create function button.

The screenshot shows the AWS Lambda console homepage. It features a dark header bar with the AWS logo and navigation links. Below the header, there's a large central area with the heading "AWS Lambda" and the subtext "lets you run code without thinking about servers." A "Get started" box contains the text "Author a Lambda function from scratch, or choose from one of many preconfigured examples." and a prominent orange "Create a function" button. At the bottom of the page, there's a "How it works" section with a code editor showing a Node.js script, and a footer with copyright information and links to CloudShell, Feedback, and various AWS services.

**Step 4:** Now Give a name to your Lambda function, Select the language to use to write your function. Note that the console code editor supports only Node.js, Python, and Ruby. So will select Python 3.12 , Architecture as x86, and Execution role to Create a new role with basic Lambda permissions.

The screenshot shows the "Create function" wizard. In the "Basic information" step, the user has selected "Author from scratch" and entered the function name "anshi-sadneya-lambda". The "Runtime" is set to "Python 3.12", and the "Architecture" is set to "x86\_64". Other options like "Container image" and "Use a blueprint" are also visible.

The screenshot shows the AWS Lambda console interface. At the top, it displays the function name 'anshi-sadneya-lambda'. Below the function name are buttons for Throttle, Copy ARN, and Actions. A 'Function overview' section includes a 'Diagram' tab (selected) and a 'Template' tab. The diagram shows a single function icon labeled 'anshi-sadneya-lambda' and a 'Layers' section with '(0)' next to it. There are buttons for '+ Add trigger' and '+ Add destination'. To the right, there's a 'Description' field with a '-' sign, 'Last modified' (2 seconds ago), 'Function ARN' (arn:aws:lambda:us-east-1:708398963195:function:anshi-sadneya-lambda), and a 'Function URL' link. Below these are tabs for Code, Test, Monitor, Configuration, Aliases, and Versions. Under the Code tab, the 'Code source' section shows the code editor with the file 'lambda\_function.py' containing:

```

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

```

So See or Edit the basic settings go to configuration then click on edit general setting.

The screenshot shows the AWS Lambda Configuration page. The 'Configuration' tab is active. On the left, a sidebar lists General configuration, Triggers, Permissions, Destinations, and Function URL. The main area displays 'General configuration' with the following settings:

| Setting           | Value       |
|-------------------|-------------|
| Description       | -           |
| Memory            | 128 MB      |
| Ephemeral storage | 512 MB      |
| Timeout           | 0 min 3 sec |
| SnapStart         | None        |

Here, you can enter a description and change Memory and Timeout. I've changed the Timeout period to 1 sec since that is sufficient for now.

The screenshot shows the 'Edit basic settings' dialog. It includes the following fields:

- Basic settings**: Description - optional: Basic Settings (input field).
- Memory**: Info: 128 MB. Note: Your function is allocated CPU proportional to the memory configured. Set memory to between 128 MB and 10240 MB.
- Ephemeral storage**: Info: 512 MB. Note: You can configure up to 10 GB of ephemeral storage (/tmp) for your function. View pricing.
- SnapStart**: Info: None. Note: Reduce startup time by having Lambda cache a snapshot of your function after the function has initialized. To evaluate whether your function code is resilient to snapshot operations, review the SnapStart compatibility considerations.
- Timeout**: 0 min 1 sec.
- Execution role**: Choose a role that defines the permissions of your function. To create a custom role, go to the IAM console. Options: Use an existing role (selected), Create a new role from AWS policy templates.

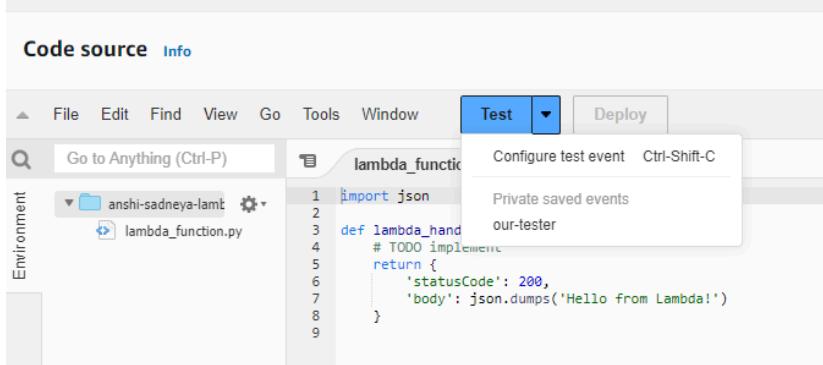
**Step 5:** Now Click on the Test tab then select Create a new event, give a name to the event and select Event Sharing to private, and select s3 put template.

The screenshot shows the 'Test event' configuration screen. At the top, there are 'Save' and 'Test' buttons. Below them, a note says 'To invoke your function without saving an event, configure the JSON event, then choose Test.' Under 'Test event action', the 'Create new event' option is selected. The 'Event name' field contains 'our-tester'. In the 'Event sharing settings' section, 'Private' is selected, with a note that it's available in the Lambda console and event creator. 'Shareable' is also listed with its own note. Under 'Template - optional', the 'hello-world' template is selected. The 'Event JSON' section contains a sample JSON array:

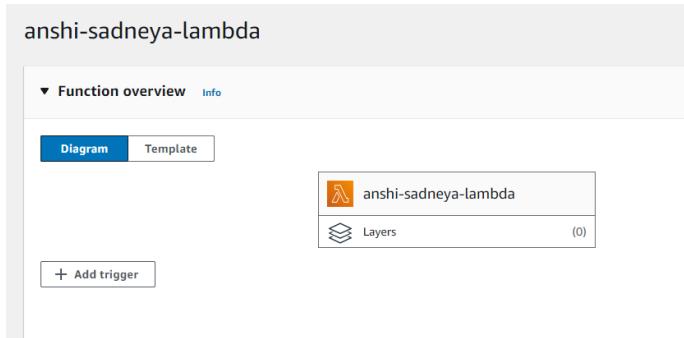
```
1 * []
2 "key1": "value1",
3 "key2": "value2",
4 "key3": "value3"
5 []
```

A 'Format JSON' button is located at the top right of the JSON editor.

**Step 6:** Now In Code section select the created event from the dropdown .



**Step 7:** Now In the Lambda function click on add trigger.



Now select the source as S3 then select the bucket name from the dropdown, keep other things to default and also you can add prefix to image.

Lambda > Add triggers

## Add trigger

**Trigger configuration** [Info](#)

**S3** aws asynchronous storage

**Bucket**  
Choose or enter the ARN of an S3 bucket that serves as the event source. The bucket must be in the same region as the function.

[X](#) [C](#)

Bucket region: us-east-1

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

All object create events [X](#)

**Prefix - optional**  
Enter a single optional prefix to limit the notifications to objects with keys that start with matching characters. Any [special characters](#) must be URL encoded.

**Function overview** [Info](#)

[Diagram](#) [Template](#)

**anshi-sadneya-lambda**

The trigger anshi-sadneya-lambda-bucket was successfully added to function anshi-sadneya-lambda. The function is r

**Layers** (0)

**S3**

[+ Add trigger](#)

**Configuration** [Code](#) [Test](#) [Monitor](#) [Aliases](#) [Versions](#)

**Triggers (1) Info**

Trigger

**S3: anshi-sadneya-lambda-bucket**  
arn:aws:s3:::anshi-sadneya-lambda-bucket

[Details](#)

**Step 8:** Now Write code that logs a message like “An Image has been added” when triggered. Save the file and click on deploy.

**Code source** [Info](#)

[File](#) [Edit](#) [Find](#) [View](#) [Go](#) [Tools](#) [Window](#) [Test](#) [Deploy](#) **Changes not deployed**

**lambda\_function** Environment Var [+](#)

```

1 import json
2
3 def lambda_handler(event, context):
4 # TODO implement
5 bucket_name=event['Records'][0]['s3']['bucket']['name']
6 object_key=event['Records'][0]['s3']['object']['key']
7 print(f'An image has been added to the bucket {bucket_name} : {object_key}')
8 return {
9 'statusCode': 200,
10 'body': json.dumps('Log entry created successfully!')
11 }
12

```

**Code source** [Info](#)

[File](#) [Edit](#) [Find](#) [View](#) [Go](#) [Tools](#) [Window](#) [Test](#) [Deploy](#)

**lambda\_function** Environment [+](#)

```

1 import json
2
3 def lambda_handler(event, context):
4 # TODO implement
5 bucket_name=event['Records'][0]['s3']['bucket']['name']
6 object_key=event['Records'][0]['s3']['object']['key']
7 print(f'An image has been added to the bucket {bucket_name} : {object_key}')
8 return {
9 'statusCode': 200,
10 'body': json.dumps('Log entry created successfully!')
11 }
12

```

## Step 9: Now upload any image to the bucket.

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, 1.5 MB)**  
All files and folders in this table will be uploaded.

| Name                                  | Type       | Size   | Status    | Error |
|---------------------------------------|------------|--------|-----------|-------|
| abigail-lynn-9JrBiphz0e0-unsplash.jpg | image/jpeg | 1.5 MB | Succeeded | -     |

**Destination** [Info](#)

Destination  
[s3://anshi-sadneya-lambda-bucket](#)

**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.

[Cancel](#) [Upload](#)

**Upload succeeded**  
View details below.

Upload: status [Close](#)

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

**Summary**

| Destination                                      | Succeeded                                | Failed                            |
|--------------------------------------------------|------------------------------------------|-----------------------------------|
| <a href="#">s3://anshi-sadneya-lambda-bucket</a> | <a href="#">1 file, 1.5 MB (100.00%)</a> | <a href="#">0 files, 0 B (0%)</a> |

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

**Files and folders (1 Total, 1.5 MB)**

| Name               | Folder | Type       | Size   | Status    | Error |
|--------------------|--------|------------|--------|-----------|-------|
| abigail-lynn-... - |        | image/jpeg | 1.5 MB | Succeeded | -     |

## Step 10: Now to click on test in lambda to check whether it is giving log when image is added to S3.

Code source [Info](#)

[Upload from](#)

File Edit Find View Go Tools Window [Test](#) Deploy

Go to Anything (Ctrl-P)

Execution results

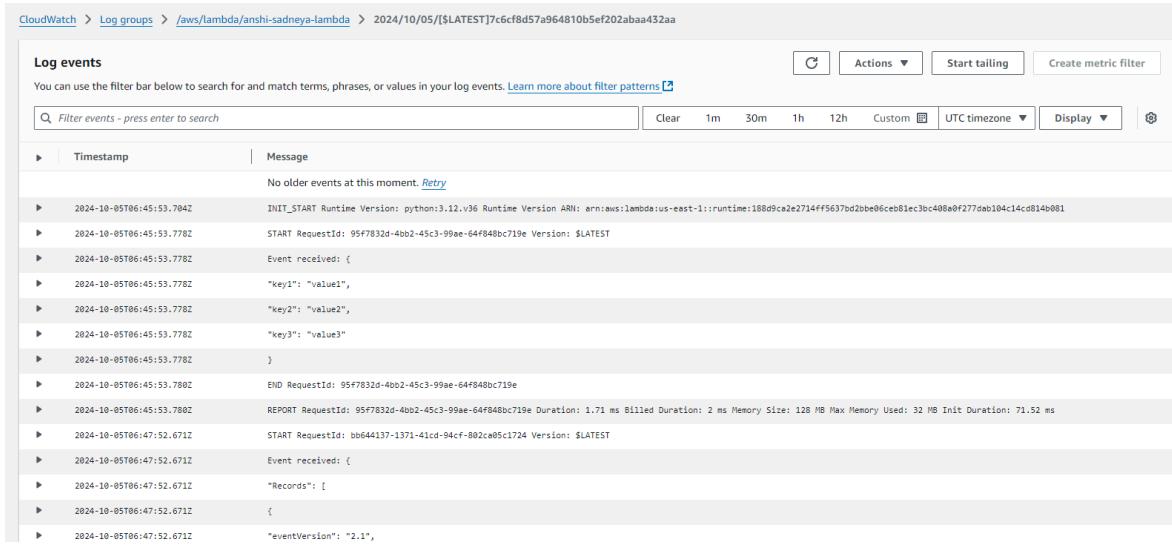
Test Event Name our-tester

Response

```
{ "statusCode": 200, "body": "\\"Log entry created successfully!\\\""}
Function Logs
START RequestId: bb644137-1371-41cd-94cf-802ca05c1724 Version: $LATEST
Event received:
{ "records": [{ "eventVersion": "2.1", "eventSource": "aws:s3", "awsRegion": "us-east-1", "eventTime": "2014-10-05T12:34:56.789Z", "eventName": "ObjectCreated:Put", "userIdentity": "AWSUSER", "principalId": "AWSUSER", "requestParameters": { "sourceIPAddress": "127.0.0.1" }, "responseElements": { "x-amz-request-id": "C3D13FES80E4C810", "x-amz-id-2": "EXAMPLE1234567890bcdef" }, "s3": { "s3SchemaVersion": "1.0", "bucket": { "name": "anshi-sadneya-lambda-bucket", "arn": "arn:aws:s3:::anshi-sadneya-lambda-bucket" }, "object": { "key": "abigail-lynn-9JrBiphz0e0-unsplash.jpg" } } } }
```

Status: [Succeeded](#) | Max memory used: 32 MB | Time: 12.69 ms

**Step 11:** Now Lets see the log on Cloud watch. To see it go to monitor section and then click on view cloudwatch logs.



The screenshot shows the AWS CloudWatch Log Events interface. The top navigation bar includes 'CloudWatch' > 'Log groups' > '/aws/lambda/anshi-sadneya-lambda' > '2024/10/05/[LATEST]7c6cf8d57a964810b5ef202abaa432aa'. Below the navigation is a search bar and filter controls for 'Actions', 'Start tailing', and 'Create metric filter'. The main area displays log events with columns for 'Timestamp' and 'Message'. The 'Message' column contains JSON log entries. One entry shows an 'Event received' message with a complex JSON payload containing keys like 'key1', 'key2', and 'key3' with their corresponding values. Another entry shows a 'REPORT' message with details such as RequestId, Duration, and Memory Size.

| Timestamp                | Message                                                                                                                                                             |
|--------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 2024-10-05T06:45:53.704Z | INIT_START Runtime Version: python:3.12.v36 Runtime Version ARN: arn:aws:lambda:us-east-1::runtime:188d9ce2e2714ff5637bd2bbe06ceb81ec3bc408a0f277dab104c14cd814b081 |
| 2024-10-05T06:45:53.778Z | START RequestId: 95f7832d-4bb2-45c3-99ae-64f848bc719e Version: \$LATEST                                                                                             |
| 2024-10-05T06:45:53.778Z | Event received: {                                                                                                                                                   |
| 2024-10-05T06:45:53.778Z | "key1": "value1",                                                                                                                                                   |
| 2024-10-05T06:45:53.778Z | "key2": "value2",                                                                                                                                                   |
| 2024-10-05T06:45:53.778Z | "key3": "value3"                                                                                                                                                    |
| 2024-10-05T06:45:53.778Z | }                                                                                                                                                                   |
| 2024-10-05T06:45:53.788Z | END RequestId: 95f7832d-4bb2-45c3-99ae-64f848bc719e                                                                                                                 |
| 2024-10-05T06:45:53.788Z | REPORT RequestId: 95f7832d-4bb2-45c3-99ae-64f848bc719e Duration: 1.71 ms Billed Duration: 2 ms Memory Size: 128 MB Max Memory Used: 32 MB Init Duration: 71.52 ms   |
| 2024-10-05T06:47:52.671Z | START RequestId: bb644137-1371-41cd-94cf-802ca05c1724 Version: \$LATEST                                                                                             |
| 2024-10-05T06:47:52.671Z | Event received: {                                                                                                                                                   |
| 2024-10-05T06:47:52.671Z | "Records": [                                                                                                                                                        |
| 2024-10-05T06:47:52.671Z | {                                                                                                                                                                   |
| 2024-10-05T06:47:52.671Z | "eventVersion": "2.1",                                                                                                                                              |

**Conclusion:** In this experiment, we successfully created an AWS Lambda function that logs a message whenever an image is uploaded to an S3 bucket. One key takeaway is that selecting the correct event template, specifically the S3 Put event, is essential. Using an incorrect event format initially caused errors due to the Lambda function not receiving the expected event structure.

After resolving these issues, the function was successfully triggered by S3 object uploads, confirming the effectiveness of Lambda's event-driven architecture. This experiment not only showcased Lambda's ability to respond to S3 events efficiently but also highlighted the importance of understanding and troubleshooting common event structure issues.