

EXPERIMENT 1

1) To Develop A Website And Host it on a VM.

Steps:

Create An Instance:

The screenshot shows the AWS EC2 Dashboard. On the left, a sidebar lists various EC2 services: Global View, Events, Instances (with sub-options like Instances, Instance Types, Launch Templates, Spot Requests, Savings Plans, Reserved Instances, Dedicated Hosts, Capacity, and Reservations), Images (AMIs, AMI Catalog), and Elastic Block Store (Volumes, Snapshots). The main area is titled 'Resources' and displays a summary of Amazon EC2 resources in the US East (N. Virginia) Region. It includes tables for Instances (running: 2), Auto Scaling Groups (0), Capacity Reservations (0), Dedicated Hosts (0), Elastic IPs (0), Instances (2), Key pairs (1), Load balancers (0), Placement groups (0), Security groups (1), Snapshots (0), and Volumes (2). To the right, there's an 'Account attributes' section showing the Default VPC (vpc-0004ba2bb735a928f) and a 'Settings' section with links for Data protection and security, Zones, EC2 Serial Console, Default credit specification, and EC2 console preferences. Below these are sections for 'Launch instance' (with 'Launch instance' and 'Migrate a server' buttons) and 'Service health' (showing AWS Health Dashboard, Region: US East (N. Virginia), Status: This service is operating normally, and Zones). A 'Explore AWS' sidebar on the right promotes better price performance and spot instances, and a footer at the bottom provides copyright information and navigation links.

The screenshot shows the 'Launch an instance' wizard. The top navigation bar includes tabs for New Tab, Launch AWS Academy Learner, Host your personal site on AWS, and Launch an instance | EC2 | us-east-1. The main content area has a breadcrumb trail: EC2 > Instances > Launch an instance. The first step, 'Launch an instance' (Info), describes creating virtual machines and provides simple steps. The second step, 'Name and tags' (Info), shows a 'Name' field with 'Web-Server' and a 'Add additional tags' button. The third step, 'Application and OS Images (Amazon Machine Image)' (Info), includes a search bar for AMIs and a 'Recent' tab. The fourth step, 'Summary' (Info), shows settings for 1 instance, using the Amazon Linux 2023 AMI (ami-066784287e3558dad1), t2.micro instance type, and a new security group. It also highlights a 'Free tier' offer for t2.micro instances. The fifth step, 'Launch instance' (Info), contains a 'Launch instance' button and a 'Review commands' link. The footer at the bottom includes standard AWS links for Privacy, Terms, and Cookie preferences, along with system status icons.

Application and OS Images (Amazon Machine Image)

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

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

Recents Quick Start

Amazon Linux macOS Ubuntu Windows Red Hat SUSE Linux

Browse more AMIs Including AMIs from AWS, Marketplace and the Community

Amazon Machine Image (AMI)

Amazon Linux 2023 AMI
ami-066784287e358dad1 (64-bit (x86), uefi-preferred) / ami-023508951a94f0c71 (64-bit (Arm), uefi)
Virtualization: hvm ENA enabled: true Root device type: ebs

Free tier eligible

Description

Amazon Linux 2023 is a modern, general purpose Linux-based OS that comes with 5 years of long term support. It is optimized for AWS and designed to provide a secure, stable and high-performance execution environment to

Summary

Number of instances: 1

Software Image (AMI): Amazon Linux 2023 AMI 2023.5.2...
Virtual server type (instance type): t2.micro
Firewall (security group): New security group
Storage (volumes): 1 volume(s) - 8 GiB

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

Cancel Launch instance Review commands

Instance type

t2.micro

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)

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

vokey

Create new key pair

Network settings

Network

Summary

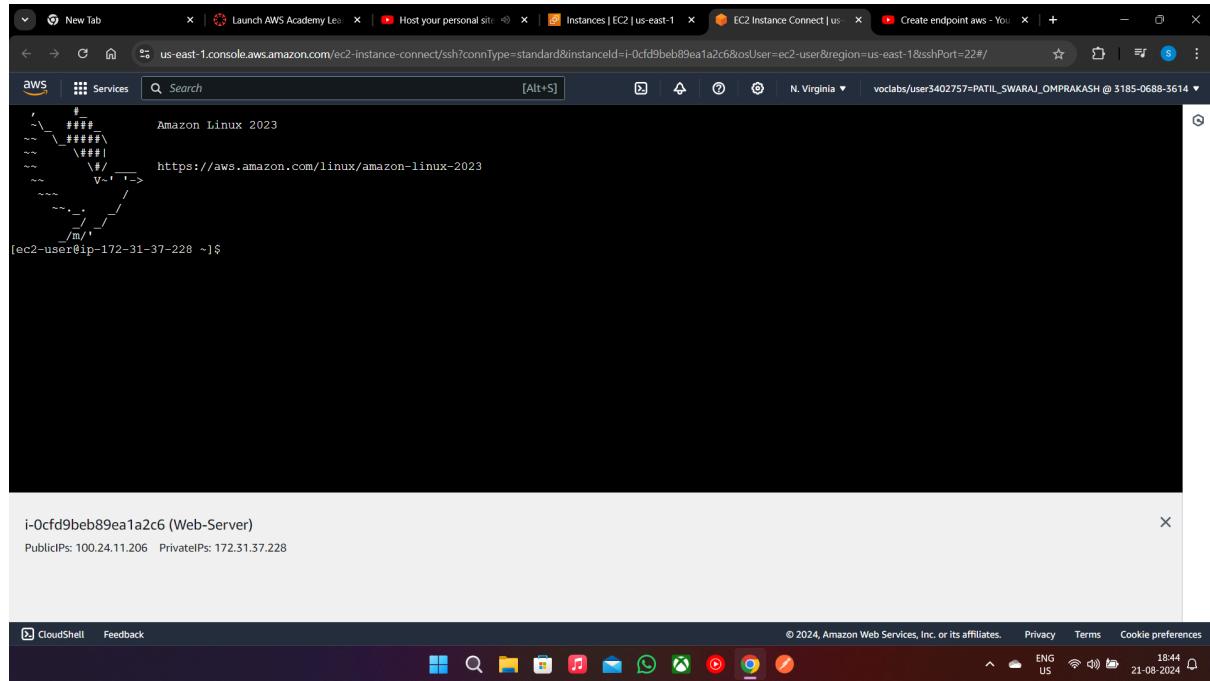
Number of instances: 1

Software Image (AMI): Amazon Linux 2023 AMI 2023.5.2...
Virtual server type (instance type): t2.micro
Firewall (security group): New security group
Storage (volumes): 1 volume(s) - 8 GiB

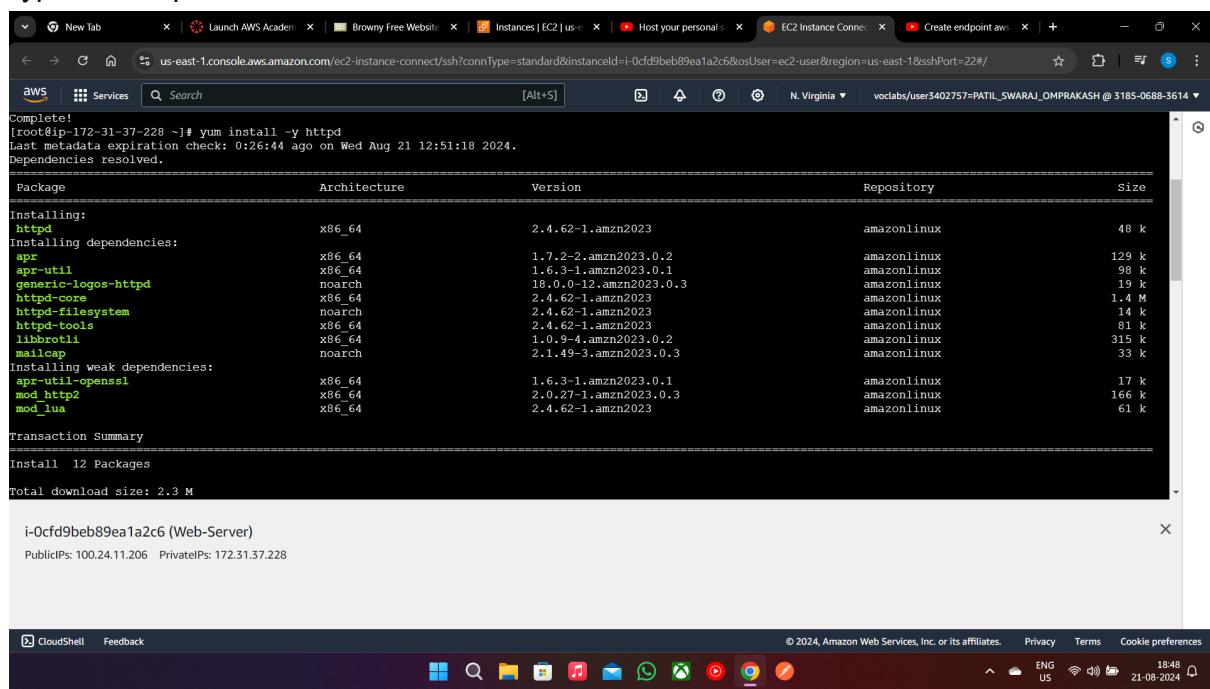
Free tier: In your first year includes 750 hours of t2.micro (or t3.micro in the Regions in which t2.micro is unavailable) instance

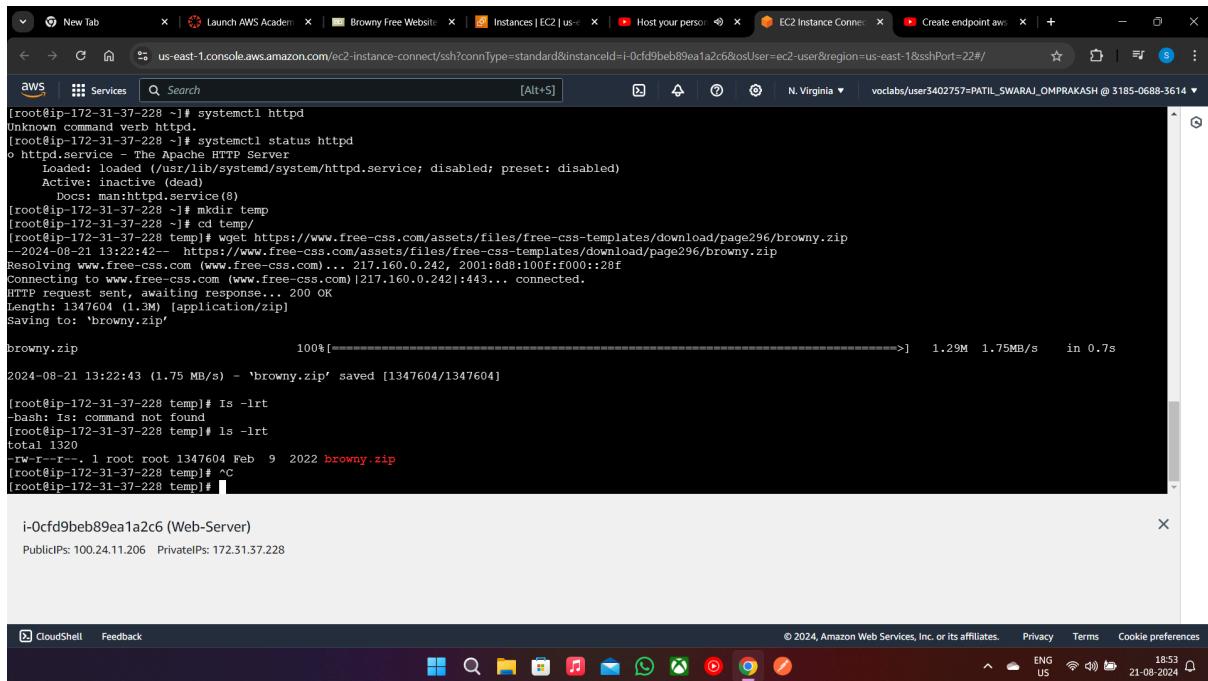
Cancel Launch instance Review commands

Step 2: Connect To the Instance:



Type the Required Commands:





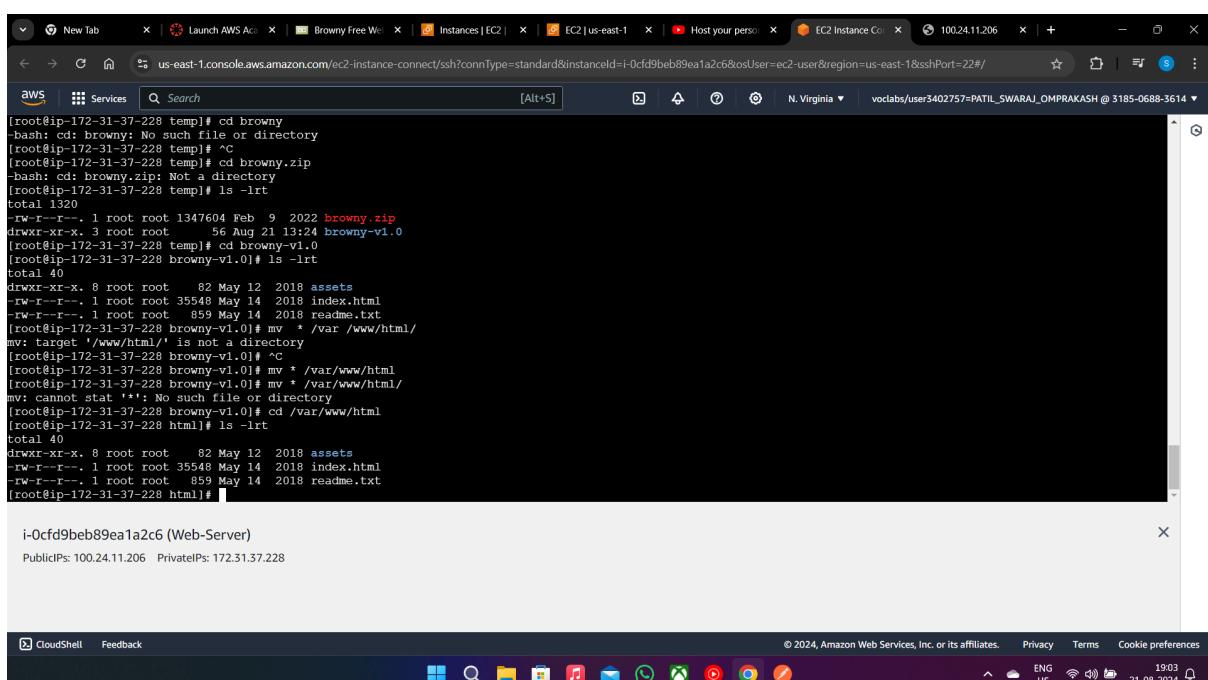
```
[root@ip-172-31-37-228 ~]# systemctl httpd
Unknown command verb httpd.
[root@ip-172-31-37-228 ~]# systemctl status httpd
● httpd.service - The Apache HTTP Server
   Loaded: loaded (/usr/lib/systemd/system/httpd.service; disabled; preset: disabled)
     Active: inactive (dead)
       Docs: man:httpd.service(8)
[root@ip-172-31-37-228 ~]# mkdir temp
[root@ip-172-31-37-228 temp]# wget https://www.free-css.com/assets/files/free-css-templates/download/page296/browny.zip
--2024-08-21 13:22:42-- https://www.free-css.com/assets/files/free-css-templates/download/page296/browny.zip
Resolving www.free-css.com [www.free-css.com]... 217.160.0.242, 2001:8d8:100f:f000::28E
Connecting to www.free-css.com [www.free-css.com] 217.160.0.242:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 1347604 (1.3M) [application/zip]
Saving to: 'browny.zip'

browny.zip          100%[=====] 1.29M  1.75MB/s    in 0.7s

2024-08-21 13:22:43 (1.75 MB/s) - 'browny.zip' saved [1347604/1347604]

[root@ip-172-31-37-228 temp]# ls -lrt
-bash: ls: command not found
[root@ip-172-31-37-228 temp]# ls -lrt
total 1320
-rw-r--r--. 1 root root 1347604 Feb  9  2022 browny.zip
[root@ip-172-31-37-228 temp]# ^C
[root@ip-172-31-37-228 temp]# 
```

i-0cf9beb89ea1a2c6 (Web-Server)
PublicIPs: 100.24.11.206 PrivateIPs: 172.31.37.228



```
[root@ip-172-31-37-228 temp]# cd browny
-bash: cd: browny: No such file or directory
[root@ip-172-31-37-228 temp]# ^C
[root@ip-172-31-37-228 temp]# cd browny.zip
-bash: cd: browny.zip: Not a directory
[root@ip-172-31-37-228 temp]# ls -lrt
total 1320
-rw-r--r--. 1 root root 1347604 Feb  9  2022 browny.zip
drwxr-xr-x. 3 root root      56 Aug 21 13:24 browny-v1.0
[root@ip-172-31-37-228 temp]# cd browny-v1.0
[root@ip-172-31-37-228 browny-v1.0]# ls -lrt
total 40
drwxr-xr-x. 8 root root     82 May 12  2018 assets
-rw-r--r--. 1 root root 35548 May 14  2018 index.html
-rw-r--r--. 1 root root    859 May 14  2018 readme.txt
[root@ip-172-31-37-228 browny-v1.0]# mv * /var/www/html/
mv: target '/var/www/html/' is not a directory
[root@ip-172-31-37-228 browny-v1.0]# ^C
[root@ip-172-31-37-228 browny-v1.0]# mv * /var/www/html/
[root@ip-172-31-37-228 browny-v1.0]# mv * /var/www/html/
mv: cannot stat '*': No such file or directory
[root@ip-172-31-37-228 browny-v1.0]# cd /var/www/html
[root@ip-172-31-37-228 html]# ls -lrt
total 40
drwxr-xr-x. 8 root root     82 May 12  2018 assets
-rw-r--r--. 1 root root 35548 May 14  2018 index.html
-rw-r--r--. 1 root root    859 May 14  2018 readme.txt
[root@ip-172-31-37-228 html]# 
```

i-0cf9beb89ea1a2c6 (Web-Server)
PublicIPs: 100.24.11.206 PrivateIPs: 172.31.37.228

Give it Permission For HTTP And Https:

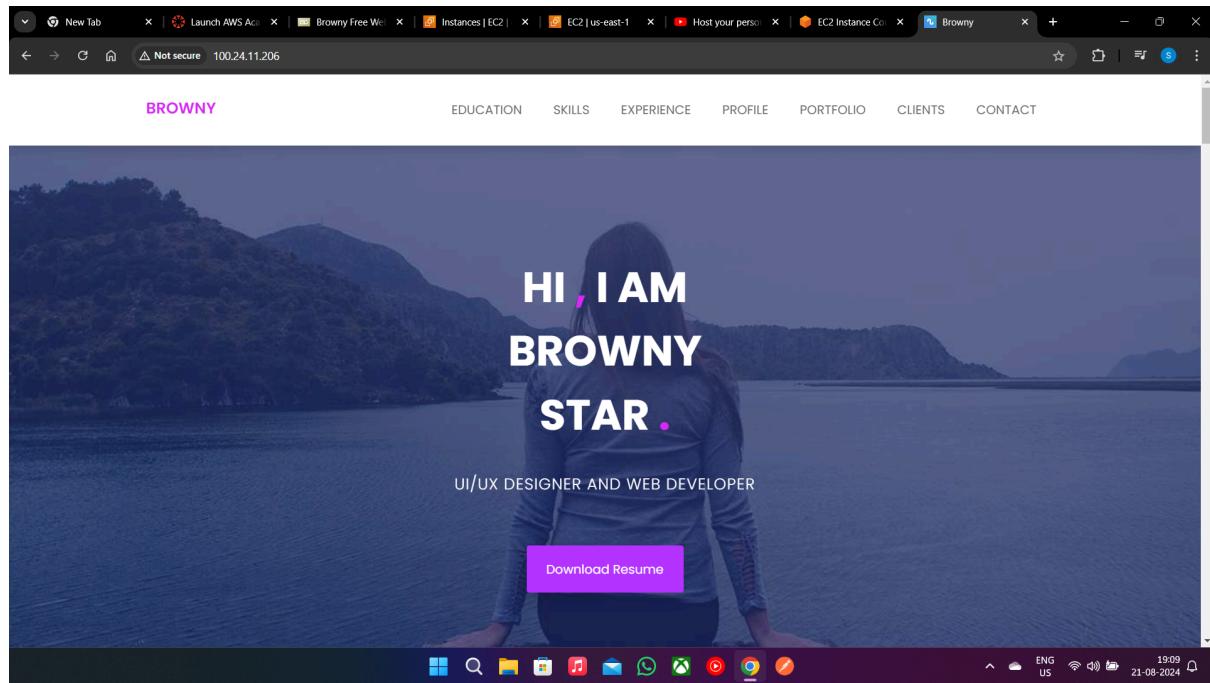
The screenshot shows the AWS CloudShell interface. The user is navigating through the AWS services, specifically focusing on the EC2 security group configuration. The security group named 'sg-0f1d7bd3441ae56c8' is selected. The 'Inbound rules' tab is active, displaying a single rule: 'sgr-0d819c4a450ad39...' (IP version: IPv4, Type: SSH, Protocol: TCP, Port range: 22). The 'Outbound rules' tab and 'Tags' tab are also visible. The bottom of the screen shows the Windows taskbar with various pinned icons.

The screenshot shows the AWS CloudShell interface. The user has run the command `systemctl status httpd` to check the status of the Apache HTTP Server. The output indicates that the service is inactive (dead) and has been loaded. The Public IP address of the instance is shown as 100.24.11.206. The bottom of the screen shows the Windows taskbar with various pinned icons.

```
--image=PATH      Edit/enable/disable/mask unit files in the specified image
--n --lines=INTEGER Number of journal entries to show
--o --output=STRING Change journal output mode (short, short-precise,
                     short-iso, short-iso-precise, short-full,
                     short-monotonic, short-unix, short-delta,
                     verbose, export, json, json-pretty, json-sse, cat)
--firmware-setup   Tell the firmware to show the setup menu on next boot
--boot-loader-menu=TIME Boot into boot loader menu on next boot
--boot-loader-entry=NAME Boot into a specific boot loader entry on next boot
--plain           Print unit dependencies as a list instead of a tree
--timestamp=FORMAT Change format of printed timestamps (pretty, unix,
                     us, utc, us+utc)
--read-only        Create read-only bind mount
--mkdir            Create directory before mounting, if missing
--marked          Restart/reload previously marked units

See the systemctl(1) man page for details.
log file: /var/log/systemctl.log
^C
[root@ip-172-31-37-228 ~]# systemctl status httpd
● httpd.service - The Apache HTTP Server
   Loaded: loaded (/usr/lib/systemd/system/httpd.service; disabled; preset: disabled)
     Active: inactive (dead)
       Docs: man:httpd.service(8)
[root@ip-172-31-37-228 ~]# 
```

THE HOSTED WEBSITE:



2) HOSTING THROUGH CLOUD9:

Step 1: CREATE A CLOUD9 ENVIRONMENT:

A screenshot of a web browser showing the AWS Cloud9 'Create environment' interface. The URL is 'us-east-1.console.aws.amazon.com/cloud9control/home?region=us-east-1#/create/'. The page shows a success message: 'Successfully created AWS Cloud9. To get the most out of your environment, see Best practices for using AWS Cloud9'. It also provides information about AWS Toolkits and CloudShell. Below this, there's a 'Create environment' form with a 'Details' section. It asks for a 'Name' (input field), a 'Description - optional' (input field), and an 'Environment type' section. The 'New EC2 instance' option is selected, with a note that Cloud9 creates an EC2 instance in your account. The bottom of the screen shows the AWS navigation bar and system status indicators.

STEP 2:

The screenshot shows the 'Network settings' step of the AWS Cloud9 environment creation wizard. It includes options for 'AWS Systems Manager (SSM)' (selected) and 'Secure Shell (SSH)'. A note states: 'How your environment is accessed.' Under 'Tags - optional', it says: '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.' A callout box informs: 'The following IAM resources will be created in your account'.

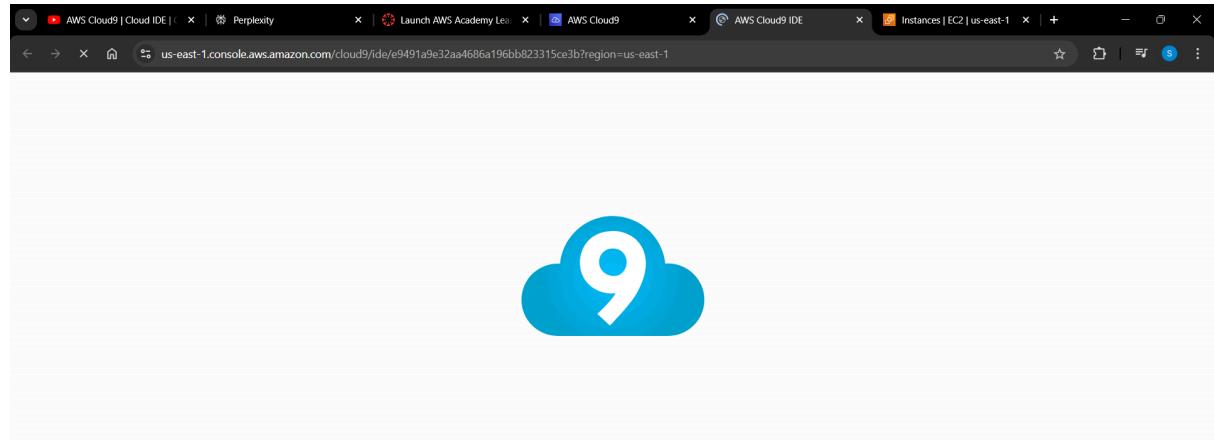
- AWSServiceRoleForAWSCloud9 - AWS Cloud9 creates a service-linked role for you. This allows AWS Cloud9 to call other AWS services on your behalf. You can delete the role from the AWS IAM console once you no longer have any AWS Cloud9 environments. [Learn more](#)
- AWSCloud9SSMAccessRole and AWSCloud9SSMInstanceProfile - A service role and an instance profile are automatically created if Cloud9 accesses its EC2 instance through AWS Systems Manager. If your environments no longer require EC2 instances that block incoming traffic, you can delete these roles using the AWS IAM console. [Learn more](#)

Buttons at the bottom: 'Cancel' and 'Create'.

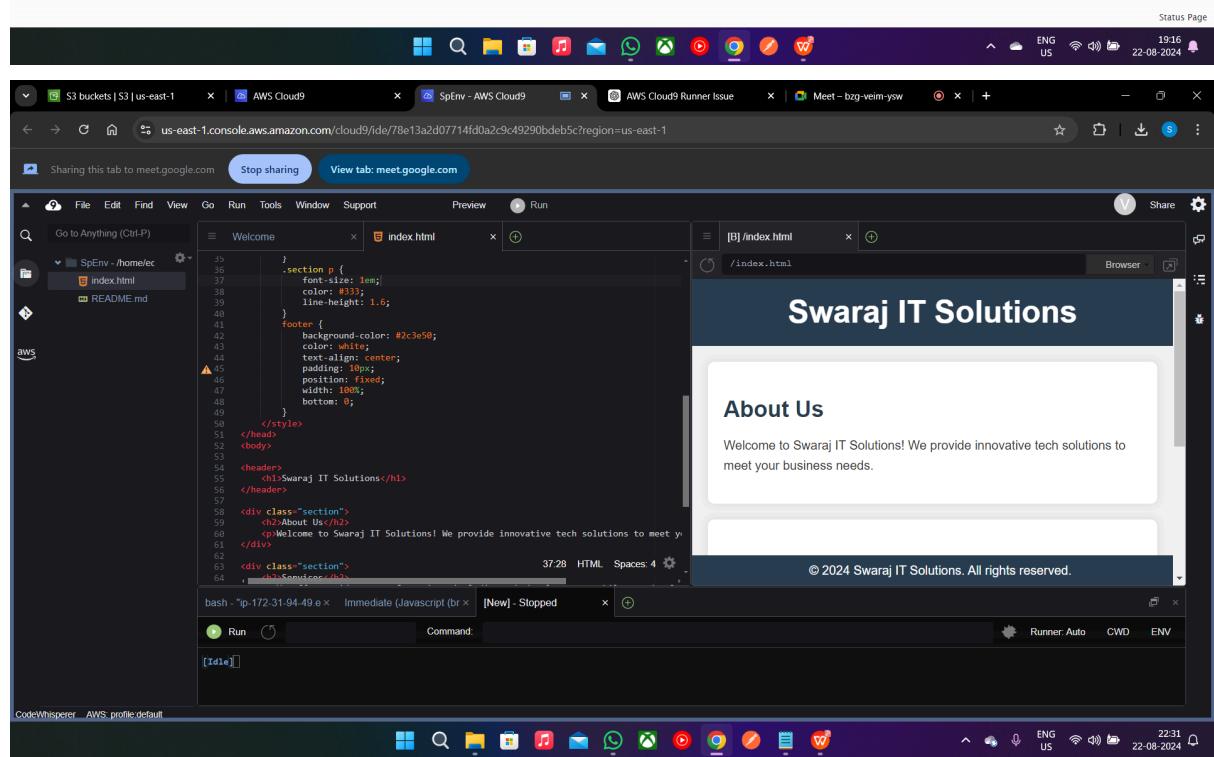
The screenshot shows the 'Environments' list in the AWS Cloud9 console. It displays one environment named 'AWS Cloud9' which was just created. The table columns are: Name, Cloud9 IDE, Environment type, Connection, Permission, and Owner ARN. The environment details are: Name - AWS Cloud9, Cloud9 IDE - Open, Environment type - EC2 instance, Connection - Secure Shell (SSH), Permission - Owner, and Owner ARN - arn:aws:sts::318506883614:assumed-role/voclabs/user3402757=PATIL_SWARAJ_OMPRAKASH.

Buttons at the bottom: 'Delete', 'View details', 'Open in Cloud9', and 'Create environment'.

STEP 3: OPEN IN THE CLOUD9 IDE:



To rename a variable, highlight it then press Ctrl-Alt-R.



HOSTING USING S3 BUCKET:

CREATE A BUCKET:

The screenshot shows the 'Create bucket' configuration page in the AWS S3 console. The 'General configuration' section is selected. Under 'AWS Region', 'US East (N. Virginia) us-east-1' is chosen. Under 'Bucket type', 'General purpose' is selected, which is described as recommended for most use cases and access patterns. A second option, 'Directory - New', is also listed. The 'Bucket name' field contains 'MyBucket'. Below it, a note states that the name must be unique within the global namespace and follow bucket naming rules. There is a 'Copy settings from existing bucket - optional' section with a 'Choose bucket' button and a dropdown for 'Format: s3://bucket/prefix'. The bottom of the page includes standard AWS navigation links like CloudShell and Feedback.

SELECT THE BUCKET:

The screenshot shows the 'Buckets' list page in the AWS S3 console. A green banner at the top indicates that a bucket named 'bucky11' has been successfully created. Below the banner, there is an 'Account snapshot' section with a 'Storage lens' link. The main table lists three buckets: 'bucky11', 'elasticbeanstalk-us-east-1-318506883614', and 'hutyapa'. The 'bucky11' bucket is highlighted with a blue border. The table includes columns for Name, AWS Region, IAM Access Analyzer, and Creation date. The bottom of the page includes standard AWS navigation links like CloudShell and Feedback.

Name	AWS Region	IAM Access Analyzer	Creation date
bucky11	US East (N. Virginia) us-east-1	View analyzer for us-east-1	August 22, 2024, 22:47:31 (UTC+05:30)
elasticbeanstalk-us-east-1-318506883614	US East (N. Virginia) us-east-1	View analyzer for us-east-1	August 8, 2024, 15:02:46 (UTC+05:30)
hutyapa	US East (N. Virginia) us-east-1	View analyzer for us-east-1	August 8, 2024, 14:35:56 (UTC+05:30)

UPLOAD THE HTML FILE:

The screenshot shows two screenshots of the AWS S3 console. The top screenshot shows the 'Objects' page for the 'bucky11' bucket, which is currently empty. The bottom screenshot shows the details for the uploaded 'index.html' file.

Top Screenshot (Objects Page):

- Bucket: bucky11
- Region: N. Virginia
- Actions: Copy S3 URI, Copy URL, Download, Open, Delete, Actions, Create folder, Upload
- Search bar: Find objects by prefix
- Table Headers: Name, Type, Last modified, Size, Storage class
- Message: No objects. You don't have any objects in this bucket.
- Upload button

Bottom Screenshot (Object Details):

- Bucket: bucky11
- Object: index.html
- Properties tab selected
- Object overview table:

Owner	s3://bucky11/index.html
AWS Region	Amazon Resource Name (ARN)
Last modified	Entity tag (Etag)
Size	arn:aws:s3:::bucky11/index.html
Type	d2789291e219bc55a8cdf3c2c088d2b1
Key	Object URL
	https://bucky11.s3.amazonaws.com/index.html
- CloudShell and Feedback buttons at the bottom

OUTPUT PAGE:

The screenshot shows a web browser window with multiple tabs open. The active tab displays the homepage of "Swaraj IT Solutions". The page has a dark header with the company name. Below the header is a light gray content area containing three sections: "About Us", "Services", and "Contact Us". Each section has a heading and a brief description. At the bottom of the page is a dark footer bar with the copyright notice "© 2024 Swaraj IT Solutions. All rights reserved." and various system icons.

Swaraj IT Solutions

About Us
Welcome to Swaraj IT Solutions! We provide innovative tech solutions to meet your business needs.

Services
We offer a wide range of services including web development, mobile app development, and IT consulting.

Contact Us
Reach out to us via email at contact@swrajitsolutions.com or call us at +123-456-7890.

© 2024 Swaraj IT Solutions. All rights reserved.

ADVANCE

DEVOPS EXPERIMENT NO.2

Name: Swaraj Patil

Class:D15A

Roll No:40

Aim:-

To Build Your Application using AWS CodeBuild and Deploy on S3 / SEBS using AWSCodePipeline, deploy Sample Application on EC2 instance using AWS CodeDeploy

Step 1
Configure environment

Step 2
Configure service access

Step 3 - optional
Set up networking, database, and tags

Step 4 - optional
Configure instance traffic and scaling

Step 5 - optional
Configure updates, monitoring, and logging

Step 6
Review

Configure environment [Info](#)

Environment tier [Info](#)
Amazon Elastic Beanstalk has two types of environment tiers to support different types of web applications.

Web server environment
Run a website, web application, or web API that serves HTTP requests. [Learn more](#)

Worker environment
Run a worker application that processes long-running workloads on demand or performs tasks on a schedule. [Learn more](#)

Application information [Info](#)

Application name

Maximum length of 100 characters.

► Application tags (optional)

Environment information [Info](#)
Choose the name, subdomain and description for your environment. These cannot be changed later.

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

Services Search [Alt+S] N. Virginia v vocabs/user3402784=PATIL_BHAGYESH_BHARAT_MADHURI @ 4693-7468... ▾

Application tags (optional)

Environment information [Info](#)
Choose the name, subdomain and description for your environment. These cannot be changed later.

Environment name

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

Domain
 .us-east-1.elasticbeanstalk.com [Check availability](#)

Environment description

Platform [Info](#)

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

Custom platform

Platform version
3.8.2 (Recommended)

Application code [Info](#)

- Sample application
- Existing version Application versions that you have uploaded.
- Upload your code Upload a source bundle from your computer or copy one from Amazon S3.

Version label Unique name for this version of your application code.
v1

Source code origin. Maximum size 500 MB

- Local file
- Upload application [Choose file](#)
- File name: **bp.php** File must be less than 500MB max file size
- Public S3 URL

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

aws Services [Search](#) [Alt+S] N. Virginia vclabs/user3402784=Pa

Configure service access [Info](#)

Step 2 **Configure service access**

Step 3 - optional [Set up networking, database, and tags](#)

Step 4 - optional [Configure instance traffic and scaling](#)

Step 5 - optional [Configure updates, monitoring, and logging](#)

Step 6 [Review](#)

Service access
IAM roles, assumed by Elastic Beanstalk as a service role, and EC2 instance profiles allow Elastic Beanstalk to create and manage your environment. Both the IAM role and instance profile must be attached to IAM managed policies that contain the required permissions. [Learn more](#)

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.

EMR_EC2_DefaultRole

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

vockey

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

EMR_EC2_DefaultRole

[View permission details](#)

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

Step 5 - optional [Configure updates, monitoring, and logging](#)

Step 6 [Review](#)

Instance settings
Choose a subnet in each AZ for the instances that run your application. To avoid exposing your instances to the Internet, run your instances in private subnets and load balancer in public subnets. To run your load balancer and instances in the same public subnets, assign public IP addresses to the instances. [Learn more](#)

Public IP address
Assign a public IP address to the Amazon EC2 instances in your environment.

Activated

Instance subnets

<input type="checkbox"/>	Availability Zone	Subnet	CIDR	Name
<input type="checkbox"/>	us-east-1b	subnet-0280aadd4...	172.31.32.0/20	
<input checked="" type="checkbox"/>	us-east-1a	subnet-02fd58f51...	172.31.16.0/20	
<input type="checkbox"/>	us-east-1e	subnet-030f247bc...	172.31.48.0/20	
<input type="checkbox"/>	us-east-1f	subnet-087b16f5d...	172.31.64.0/20	
<input type="checkbox"/>	us-east-1d	subnet-09207757c...	172.31.80.0/20	
<input type="checkbox"/>	us-east-1c	subnet-0eb3eafea...	172.31.0.0/20	

Databases [+ Add](#)

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

Database Info

Integrate an RDS SQL database with your environment. Learn more [\[external link\]](#)

Database subnets

If your Elastic Beanstalk environment is attached to an Amazon RDS, choose subnets for your database instances. Learn more [\[external link\]](#)

Choose database subnets (6)

Availability Zone	Subnet	CIDR	Name
<input type="checkbox"/> us-east-1b	subnet-0280aadd4...	172.31.32.0/20	
<input checked="" type="checkbox"/> us-east-1a	subnet-02fd58f51...	172.31.16.0/20	
<input type="checkbox"/> us-east-1e	subnet-030f247bc...	172.31.48.0/20	
<input type="checkbox"/> us-east-1f	subnet-087b16f5d...	172.31.64.0/20	
<input type="checkbox"/> us-east-1d	subnet-09207757c...	172.31.80.0/20	
<input type="checkbox"/> us-east-1c	subnet-0eb3eafea...	172.31.0.0/20	

Enable database

Restore a snapshot - optional

Restore an existing snapshot from a previously used database.

Snapshot

© 2024, Amazon Web Services, Inc. or its affiliates. [Privacy](#) [Terms](#) [Cookie preferences](#)

Instance Metadata Service (IMDS)

Your environment's platform supports both IMDSv1 and IMDSv2. To enforce IMDSv2, deactivate IMDSv1. Learn more [\[external link\]](#)

IMDSv1

With the current setting, the environment enables only IMDSv2.

Deactivated

EC2 security groups

Select security groups to control traffic.

Group name	Group ID	Name
<input type="checkbox"/> aws-cloud9-demo-1-91871...	sg-0382a792a5b32ba40	
<input checked="" type="checkbox"/> default	sg-07e95b9dea58eaa28	
<input type="checkbox"/> launch-wizard-1	sg-02503bfbcc4f23b2a	

Capacity Info

Configure the compute capacity of your environment and auto scaling settings to optimize the number of instances used.

(standard S3 charges apply.)

Activated

Instance log streaming to CloudWatch logs

Configure the instances in your environment to stream logs to CloudWatch logs. You can set the retention up to 10 years and configure Elastic Beanstalk to delete the logs when you terminate your environment. Learn more [\[external link\]](#)

Log streaming
(standard CloudWatch charges apply.)

Activated

Retention

7

Lifecycle

Keep logs after terminating envir...

Environment properties

The following properties are passed in the application as environment properties. Learn more [\[external link\]](#)

No environment properties have been configured.

Add environment property

Cancel **Previous** **Next**

CloudShell Feedback

28°C Mostly cloudy

Search

ENG IN 19:35 21-08-2024

Review Info

Step 1: Configure environment

Edit

Environment information

Environment tier	Application name
Web server environment	Aryan27
Environment name	Application code
Aryan27-env	Sample application
Platform	
arn:aws:elasticbeanstalk:us-east-1::platform/PHP 8.3 running on 64bit Amazon Linux 2023/4.3.2	

Step 2: Configure service access

Edit

Service access Info

Configure the service role and EC2 instance profile that Elastic Beanstalk uses to manage your environment. Choose an EC2 key pair to securely log in to your EC2 instances.

Service role	EC2 instance profile
arn:aws:iam::405894863107:role/service-role/aws-elasticbeanstalk-service-role	aws-elasticbeanstalk-ec2-role

Step 3: Set up networking, database, and tags

Edit

Networking, database, and tags Info

Configure VPC settings, and subnets for your environment's EC2 instances and load balancer. Set up an Amazon RDS database that's integrated with your environment.

Network

VPC	Public IP address	Instance subnets
vpc-0bf7d7d872a737f13	false	subnet-035fe38d8d742329e,subnet-0a7c9c6dedec1325d

Step 5: Configure updates, monitoring, and logging

Edit

Updates, monitoring, and logging Info

Define when and how Elastic Beanstalk deploys changes to your environment. Manage your application's monitoring and logging settings, instances, and other environment resources.

Monitoring

System enhanced	Cloudwatch custom metrics - instance	Cloudwatch custom metrics - environment
—	—	—
Log streaming	Retention	Lifecycle
Deactivated	7	false
Updates		
Managed updates	Deployment batch size	Deployment batch size type
Activated	100	Percentage

Platform software

Lifecycle	Log streaming	Allow URL fopen
false	Deactivated	On
Display errors	Document root	Max execution time
Off	—	60
Memory limit	Zlib output compression	Proxy server
256M	Off	nginx
Logs retention	Rotate logs	Update level
7	Deactivated	minor
X-Ray enabled		

Choose identity type

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

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

Service or use case

EC2

Services				Search	[Alt+S]	Global ▾	AryanPatankar
<input type="checkbox"/>	<input type="checkbox"/>	 AWSElasticBeanstalkCustomPlatformforEC2Role	AWS managed	Provide the instance			
<input type="checkbox"/>	<input type="checkbox"/>	 AWSElasticBeanstalkEnhancedHealth	AWS managed	AWS Elastic Beanstal			
<input type="checkbox"/>	<input type="checkbox"/>	 AWSElasticBeanstalkManagedUpdatesCustomerR...	AWS managed	This policy is for the			
<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	 AWSElasticBeanstalkMulticontainerDocker	AWS managed	Provide the instance			
<input type="checkbox"/>	<input type="checkbox"/>	 AWSElasticBeanstalkReadOnly	AWS managed	Grants read-only per			
<input type="checkbox"/>	<input type="checkbox"/>	 AWSElasticBeanstalkRoleCore	AWS managed	AWElasticBeanstalk			
<input type="checkbox"/>	<input type="checkbox"/>	 AWSElasticBeanstalkRoleCWL	AWS managed	(Elastic Beanstalk op			
<input type="checkbox"/>	<input type="checkbox"/>	 AWSElasticBeanstalkRoleECS	AWS managed	(Elastic Beanstalk op			
<input type="checkbox"/>	<input type="checkbox"/>	 AWSElasticBeanstalkRoleRDS	AWS managed	(Elastic Beanstalk op			
<input type="checkbox"/>	<input type="checkbox"/>	 AWSElasticBeanstalkRoleSNS	AWS managed	(Elastic Beanstalk op			
<input type="checkbox"/>	<input type="checkbox"/>	 AWSElasticBeanstalkRoleWorkerTier	AWS managed	(Elastic Beanstalk op			
<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	 AWSElasticBeanstalkWebTier	AWS managed	Provide the instance			
<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	 AWSElasticBeanstalkWorkerTier	AWS managed	Provide the instance			

Services Search [Alt+S] Global AryanPatankar

Step 2 Add permissions

Step 3 Name, review, and create

Role details

Role name: Enter a meaningful name to identify this role.
aws-elastic-beanstalk-ec2-role

Description: Add a short explanation for this role.
Allows EC2 instances to call AWS services on your behalf.

Maximum 1000 characters. Use letters (A-Z and a-z), numbers (0-9), tabs, new lines, or any of the following characters: _+=,. @-[]{}#\$%^&{}=

Step 1: Select trusted entities

Edit

Identity and Access Management (IAM)

Role aws-elastic-beanstalk-ec2-role created. View role

IAM Roles (4) Info Create role

An IAM role is an identity you can create that has specific permissions with credentials that are valid for short durations. Roles can be assumed by entities that you trust.

Role name	Trusted entities	Last activity
AWSServiceRoleForTrustedAdvisor	AWS Service: trustedadvisor (Service)	-
AWSServiceRoleForSupport	AWS Service: support (Service-Linker)	-
aws-elasticbeanstalk-service-role	AWS Service: elasticbeanstalk	-
aws-elastic-beanstalk-ec2-role	AWS Service: ec2	-

Services Search cloud formation Stockholm

CloudFormation

Search results for 'cloud formation'

See all 66 results

CloudFormation Create and Manage Resources with Templates

Application Composer Visually design and build modern applications quickly

Athena Serverless interactive analytics service

Actions Upload Change

64bit Amazon Linux 2023/4.3.2

Screenshot of the AWS CloudFormation Stacks console showing a single stack named "awseb-e-rh8w3tywxk-stack" in the "CREATE_COMPLETE" status.

Stack name	Status	Created time	Description
awseb-e-rh8w3tywxk-stack	CREATE_COMPLETE	2024-08-21 14:36:08 UTC+0530	AWS Elastic Beanstalk environment (Name: 'Aryan27-env-1' Id: 'e-rh8w3tywxk')

Screenshot of the AWS CloudFormation Stacks console showing the same stack "awseb-e-rh8w3tywxk-stack" in the "CREATE_COMPLETE" status.

Stack name	Status	Created time	Description
awseb-e-rh8w3tywxk-stack	CREATE_COMPLETE	2024-08-21 14:36:08 UTC+0530	AWS Elastic Beanstalk environment (Name: 'Aryan27-env-1' Id: 'e-rh8w3tywxk')

Screenshot of the AWS Deploy stage configuration screen.

Add deploy stage Step 4 of 5

Deploy provider: AWS Elastic Beanstalk

Region: Asia Pacific (Mumbai)

Input artifacts: SourceArtifact

Application name: WebServer

Environment name: WebServer-env

Configure automatic rollback on stage failure:

Info message: 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.

The screenshot shows the AWS Elastic Beanstalk console. The left sidebar has 'Elastic Beanstalk' selected. Under 'Recent environments', 'WebServer-env' is listed. The main area shows a table of environments:

Environment name	Health	Date created	Domain	Running vers
WebServer-env	Green	August 17, 2024 22:...	WebServer-env.eba-227p9xyx...	code-pipeline

At the bottom, the URL is https://ap-south-1.console.aws.amazon.com/elasticbeanstalk/home?region=ap-south-1#.

The screenshot shows a web browser window with the following details:

- Address bar: Not secure webserver-env.eba-227p9xyx.ap-south-1.elasticbeanstalk.com
- Header: © 2024, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences
- Main Content:

Congratulations!

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

For next steps, read the AWS CodePipeline Documentation. Inedge 2020

Using s3 bucket

The screenshot shows the Amazon S3 console with the 'Create a bucket' wizard open. The left sidebar includes links for Buckets, Access Grants, Access Points, Object Lambda Access Points, Multi-Region Access Points, Batch Operations, IAM Access Analyzer for S3, Block Public Access settings for this account, Storage Lens, Dashboards, Storage Lens groups, AWS Organizations settings, Feature spotlight, CloudShell, and Feedback.

The main content area displays the 'Amazon S3' landing page with the heading 'Store and retrieve any amount of data from anywhere'. It includes a brief description of Amazon S3 as an object storage service and a 'Create a bucket' button. Below this, there's a 'How it works' section featuring a video thumbnail titled 'Introduction to Amazon S3' with a 'Copy link' button.

A pricing section on the right indicates that there are no minimum fees and provides a link to the AWS Simple Monthly Calculator. The footer contains copyright information and links for Privacy, Terms, and Cookie preferences.

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

Copy settings from existing bucket - optional
Only the bucket settings in the following configuration are copied.
[Choose bucket](#)
Format: s3://bucket/prefix

Object Ownership Info

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

ACLs disabled (recommended)

All objects in this bucket are owned by this account. Access to this bucket and its objects is specified using only policies.

ACLs enabled

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

Object Ownership

Bucket owner enforced

Block Public Access settings for this bucket

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

Block all public access

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

Block public access to buckets and objects granted through new access control lists (ACLs)

S3 will block public access permissions applied to newly added buckets or objects, and prevent the creation of new public access.

Encryption type Info

Server-side encryption with Amazon S3 managed keys (SSE-S3)

Server-side encryption with AWS Key Management Service keys (SSE-KMS)

Dual-layer server-side encryption with AWS Key Management Service keys (DSSE-KMS)

Secure your objects with two separate layers of encryption. For details on pricing, see DSSE-KMS pricing on the Storage tab of the [Amazon S3 pricing page](#).

Bucket Key

Using an S3 Bucket Key for SSE-KMS reduces encryption costs by lowering calls to AWS KMS. S3 Bucket Keys aren't supported for DSSE-KMS. [Learn more](#)

Disable

Enable

► Advanced settings

i After creating the bucket, you can upload files and folders to the bucket, and configure additional bucket settings.

Cancel

Create bucket

Amazon S3 > Buckets > hutyapa

hutyapa Info

Objects Properties Permissions Metrics Management Access Points

Objects (2) Info

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

<input type="checkbox"/>	Name	Type	Last modified	Size	Storage class
<input checked="" type="checkbox"/>	chutiya/	Folder	-	-	-
<input type="checkbox"/>	index.html	html	August 8, 2024, 14:38:58 (UTC+05:30)	2.2 KB	Standard

Amazon S3 > Buckets > hutyapa > chutiya/

chutiya/

Objects Properties

Objects (1) Info

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

<input type="checkbox"/>	Name	Type	Last modified	Size	Storage class
<input type="checkbox"/>	index.html	html	August 8, 2024, 14:38:59 (UTC+05:30)	2.2 KB	Standard

About This Page

This is a simple and elegant web page designed with a modern and clean UI. The layout includes a header, a content area, and a footer, all styled to provide a pleasant user experience.

[Learn More](#)

© 2024 My Beautiful Web Page

ADVANCE DEVOPS EXP 3

Name :- Swaraj Patil

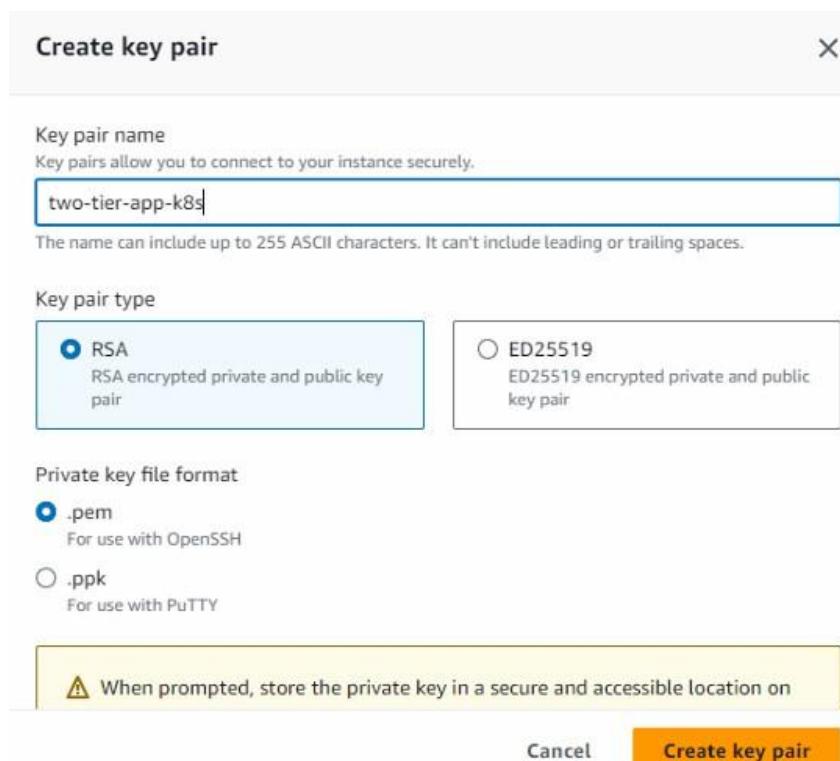
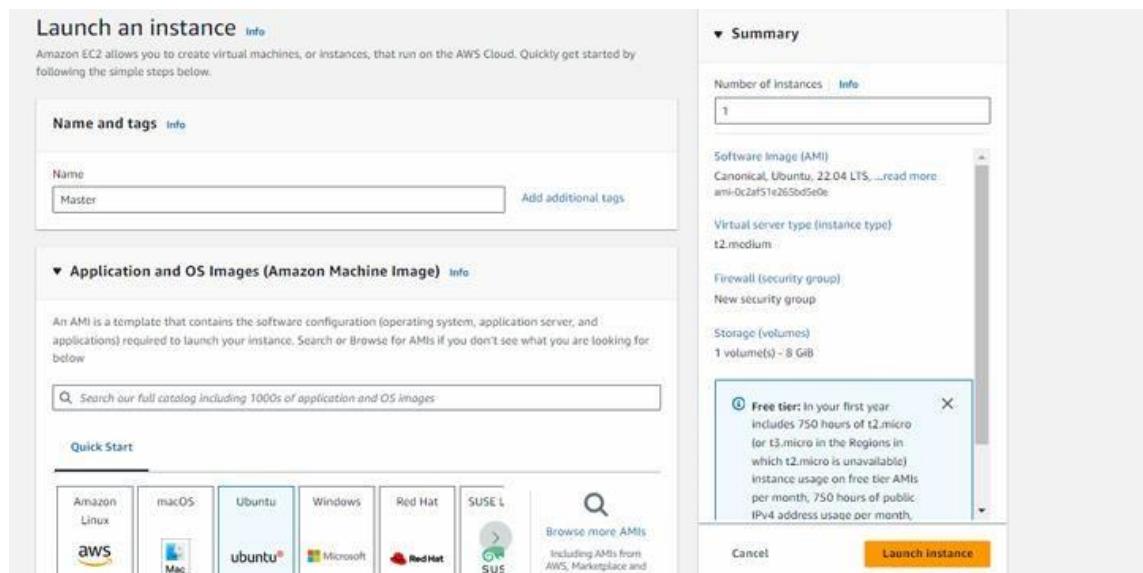
Roll no :- 40

Aim:-

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

Step 1: Pre-requisites

1.1 Create 3 EC2 instances, one for the master node and two for the worker nodes.



Create 3 EC2 Ubuntu Instances of Ubuntu version 20.04 and keep all the instances in the same security group on AWS. (Name 1 as Master, the other 2 as worker-1 and worker-2)

Instances (3) Info		C	Connect	Instance state ▾	Actions ▾	Launch instance	
<input type="text"/> Find instance by attribute or tag (case-sensitive)							
Instance state = running X		Clear filters					
	Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone
<input type="checkbox"/>	Worker2	i-0454af7c775a97b9b	Running Details	t2.micro	2/2 checks passed Details	No alarms +	us-east-1c
<input type="checkbox"/>	Worker-1	i-031892fec4e1152a7	Running Details	t2.micro	2/2 checks passed Details	No alarms +	us-east-1c
<input type="checkbox"/>	Master	i-02750ab2198a11dca	Running Details	t2.micro	2/2 checks passed Details	No alarms +	us-east-1b

Now the ssh created, copy the text given in the example

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

Instance ID
 i-0e3930ceb2d892d01 (Worker-2)

1. Open an SSH client.
2. Locate your private key file. The key used to launch this instance is two-tier-app-k8s.pem
3. Run this command, if necessary, to ensure your key is not publicly viewable.
 chmod 400 "two-tier-app-k8s.pem"
4. Connect to your instance using its Public DNS:
 ec2-13-234-226-219.ap-south-1.compute.amazonaws.com

Example:
 ssh -i "two-tier-app-k8s.pem" ubuntu@ec2-13-234-226-219.ap-south-1.compute.amazonaws.com

```
acer@TMP214-53 MINGW64 ~/Downloads
$ ssh -i "two-tier-app-k8s.pem" ubuntu@ec2-13-232-36-34.ap-south-1.compute.amazonaws.com
The authenticity of host 'ec2-13-232-36-34.ap-south-1.compute.amazonaws.com (13.232.36.34)' can't be established.
ED25519 key fingerprint is SHA256:uVGEO+FWYefj60j0ft70Sralv8NrzEi/IwxAtBY+EPE.
This key is not known by any other names.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added 'ec2-13-232-36-34.ap-south-1.compute.amazonaws.com' (ED25519) to the list of known hosts.
Welcome to Ubuntu 22.04.4 LTS (GNU/Linux 6.5.0-1022-aws x86_64)

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

System information as of Wed Sep 11 14:07:10 UTC 2024

System load: 0.0          Processes:           106
Usage of /: 20.7% of 7.57GB  Users Logged in:      0
Memory usage: 5%           IPv4 address for eth0: 172.31.45.227
Swap usage: 0%
```

Expanded Security Maintenance for Applications is not enabled.

0 updates can be applied immediately.

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

```
ubuntu@ip-172-31-81-188:~$ docker --version
Docker version 20.10.12, build 20.10.12-0ubuntu2~20.04.1
ubuntu@ip-172-31-81-188:~$ █
```

```
Amazon | Services | Search for services, features, blogs, docs, and more
ubuntu@ip-172-31-23-53:~$ docker --version
Docker version 20.10.12, build 20.10.12-0ubuntu2~20.04.1
ubuntu@ip-172-31-23-53:~$ █
```

```
ubuntu@ip-172-31-21-143:~$ docker --version
Docker version 20.10.12, build 20.10.12-0ubuntu2~20.04.1
ubuntu@ip-172-31-21-143:~$ █
```

Kubernetes Installation

Go to official documentation off kubeadm

The screenshot shows the official Kubernetes Documentation website. The top navigation bar includes links for Documentation, Kubernetes Blog, Training, Partners, Community, Case Studies, Versions, and English. A sidebar on the left provides navigation links for Documentation, Getting started, Production environment, and specific kubeADM pages like Troubleshooting kubeADM and Creating a cluster with kubeADM. The main content area displays the 'Installing kubeADM' page, which includes a sidebar with edit and creation options, a 'Before you begin' section, and a 'kubeADM' logo.

Kubernetes Documentation / Getting started / Production environment
/ Installing Kubernetes with deployment tools / Bootstrapping clusters with kubeADM
/ Installing kubeADM

Installing kubeADM

This page shows how to install the `kubeADM` toolbox. For information on how to create a cluster with kubeADM once you have performed this installation process, see the [Creating a cluster with kubeADM](#) page.

This installation guide is for Kubernetes v1.31. If you want to use a different Kubernetes version, please refer to the following pages instead:

- [Installing kubeADM \(Kubernetes v1.30\)](#)
- [Installing kubeADM \(Kubernetes v1.29\)](#)
- [Installing kubeADM \(Kubernetes v1.28\)](#)
- [Installing kubeADM \(Kubernetes v1.27\)](#)

Before you begin

1. \$sudo apt-get install kubeadm kubelet kubectl -y

```
ubuntu@ip-172-31-188:~$ sudo apt-get install kubeadm kubelet kubectl -y
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following additional packages will be installed:
  conntrack cri-tools ebttables kubernetes-cni socat
Suggested packages:
  nftables
The following NEW packages will be installed:
  conntrack cri-tools ebttables kubeadm kubectl kubelet kubernetes-cni socat
0 upgraded, 8 newly installed, 0 to remove and 62 not upgraded.
Need to get 75.9 MB of archives.
After this operation, 310 MB of additional disk space will be used.
Get:1 http://us-east-1.ec2.archive.ubuntu.com/ubuntu focal/main amd64 conntrack amd64 1:1.4.5-2
Get:2 http://us-east-1.ec2.archive.ubuntu.com/ubuntu focal/main amd64 ebttables amd64 2.0.11-3+buil
```

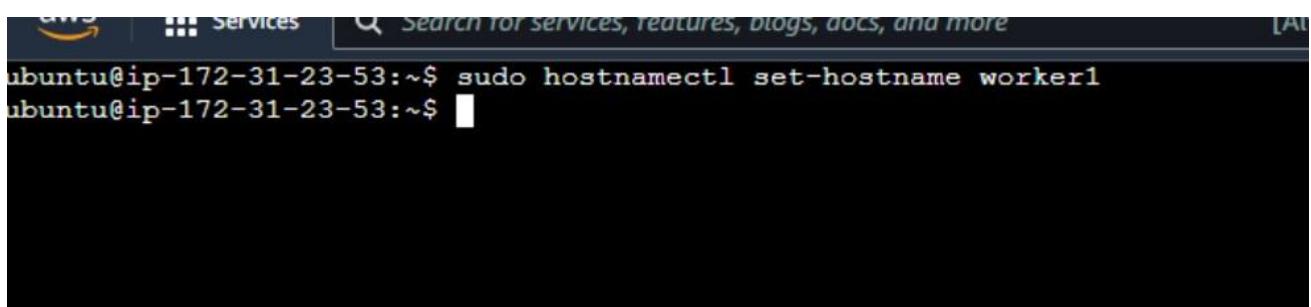
2. Verify the installation with

```
ubuntu@ip-172-31-23-53:~$ kubeadm version
kubeadm version: &version.Info{Major:"1", Minor:"25", GitVersion:"v1.25.0", GitCommit:"a866cbe2e5bbfa01cf5e969aa3e033f3282a8a2", GitTreeState:"clean",
BuildDate:"2022-08-23T17:43:25Z", GoVersion:"go1.19", Compiler:"gc", Platform:"linux/amd64"}
ubuntu@ip-172-31-23-53:~$
```

3. \$sudo swapoff -a

```
ubuntu@ip-172-31-22-29:~$ sudo swapoff -a
sudo sed -i '/ swap / s/^/#/' /etc/fstab
```

4. \$sudo hostnamectl set-hostname



```
ubuntu@ip-172-31-23-53:~$ sudo hostnamectl set-hostname worker1
ubuntu@ip-172-31-23-53:~$
```

```
cat <<EOF | sudo tee /etc/sysctl.d/k8s.conf
net.bridge.bridge-nf-call-ip6tables = 1
net.bridge.bridge-nf-call-iptables = 1
EOF
sudo sysctl—system
```

```
ubuntu@ip-172-31-81-188:~$ sudo kubeadm init --pod-network-cidr=10.244.0.0/16 --ignore-preflight-errors=all
[init] Using Kubernetes version: v1.25.0
[preflight] Running pre-flight checks
  [WARNING NumCPU]: the number of available CPUs 1 is less than the required 2
  [WARNING Mem]: the system RAM (967 MB) is less than the minimum 1700 MB
[preflight] Pulling images required for setting up a Kubernetes cluster
[preflight] This might take a minute or two, depending on the speed of your internet connection
[preflight] You can also perform this action in beforehand using 'kubeadm config images pull'
[certs] Using certificateDir folder "/etc/kubernetes/pki"
[certs] Generating "ca" certificate and key
[certs] Generating "apiserver" certificate and key
[certs] apiserver serving cert is signed for DNS names [kubernetes kubernetes.default kubernetes.default.svc kubernetes.default.svc.kube
6.0.1 172.31.81.188]
```

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 172.31.81.188:6443 --token n46tzy.ocnrf7wkiyk0t0xu \
  --discovery-token-ca-cert-hash sha256:59c2fec9fc69aa85d306f8bfkadac2d827699b0db3d87e13192873a1044f86e2
ubuntu@ip-172-31-81-188:~$
```

Deploy Pod Network to Cluster A

Pod Network is a way to allow communication between different nodes in the cluster. This tutorial uses the flannel virtual network.

```
ubuntu@ip-172-31-81-188:~$ sudo kubectl apply -f https://raw.githubusercontent.com/coreos/flannel/master/Documentation/kube-flannel.yaml
The connection to the server localhost:8080 was refused - did you specify the right host or port?
ubuntu@ip-172-31-81-188:~$ kubectl apply -f https://raw.githubusercontent.com/coreos/flannel/master/Documentation/kube-flannel.yaml
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
ubuntu@ip-172-31-81-188:~$
```

Join Worker Nodes to the Cluster

On the worker nodes, run the command provided by the master node during initialization . It looks something like this: sudo kubeadm join :6443--token --discovery-token-ca-cert-hash sha256:

```
root@worker1:~# kubeadm join 172.31.81.188:6443 --token n46tzy.ocnrf7wkiyk0t0xu      --discovery-token-ca-cert-hash sha256:59c2fec9fc69aa85d306f8bfkadac2d827699b0db3d87e13192873a1044f86e2 --ignore-preflight-errors=all
[preflight] Running pre-flight checks
error execution phase preflight: couldn't validate the identity of the API Server: Get "https://172.31.81.188:6443/api/v1/namespaces/kube-public/configmaps/cluster-info?timeout=10s": net/http: request canceled while waiting for connection (Client.Timeout exceeded while awaiting headers)
o see the stack trace of this error execute with --v=5 or higher
root@worker1:~# kubeadm join 172.31.81.188:6443 --token n46tzy.ocnrf7wkiyk0t0xu      --discovery-token-ca-cert-hash sha256:59c2fec9fc69aa85d306f8bfkadac2d827699b0db3d87e13192873a1044f86e2 --ignore-preflight-errors=all
[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-start] Waiting for the kubelet to perform the TLS Bootstrap...
his 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.
```

Verify the Cluster

```
Using cluster from kubectl context: workshop.k8s.local

Validating cluster workshop.k8s.local

INSTANCE GROUPS
NAME          ROLE    MACHINETYPE   MIN   MAX   SUBNETS
master-us-west-2a  Master  t3.medium    1     1     us-west-2a
nodes-us-west-2a   Node   t3.medium    1     1     us-west-2a

NODE STATUS
NAME          ROLE    READY
ip-172-20-40-55.us-west-2.compute.internal  master  True
ip-172-20-58-174.us-west-2.compute.internal  node    True

Your cluster workshop.k8s.local is ready
```

ADVANCE DEVOPS EXP 4

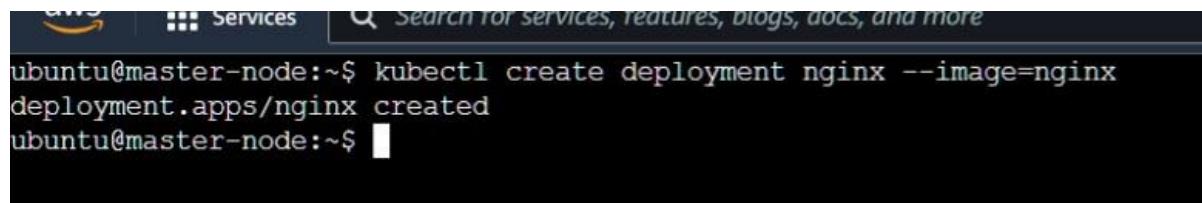
Name :- Swaraj Patil

Roll no :- 40

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

Step 1: As the cluster is up and running, we can deploy our nginx server on this cluster. Apply this deployment file using this command to create a deployment.

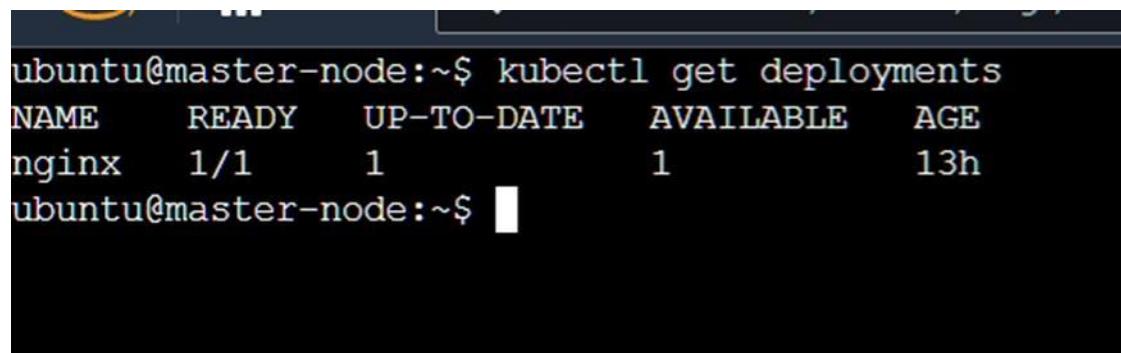
```
$kubectl create deployment nginx --image=nginx
```



```
ubuntu@master-node:~$ kubectl create deployment nginx --image=nginx
deployment.apps/nginx created
ubuntu@master-node:~$
```

Step 2: Verify the deployment using the command:

```
$kubectl get deployments
```

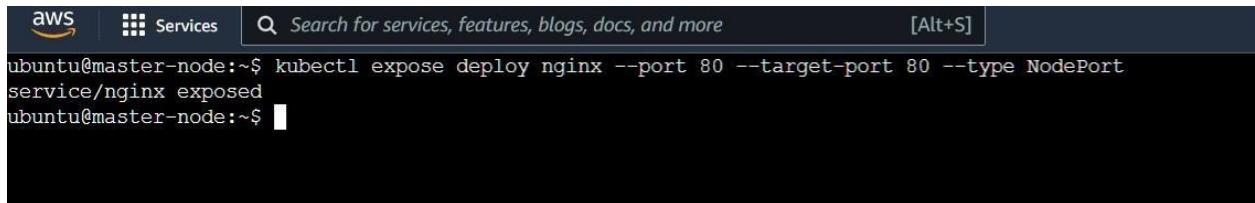


```
ubuntu@master-node:~$ kubectl get deployments
NAME      READY    UP-TO-DATE   AVAILABLE   AGE
nginx     1/1      1           1           13h
ubuntu@master-node:~$
```

Step 3: Next, run the following command to create a service named nginx that will expose the app publicly. It will do so through a NodePort, a scheme that will make the pod accessible through an arbitrary port opened on each node of the cluster

with this service-type, Kubernetes will assign this service on ports on the **30000+** range.

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

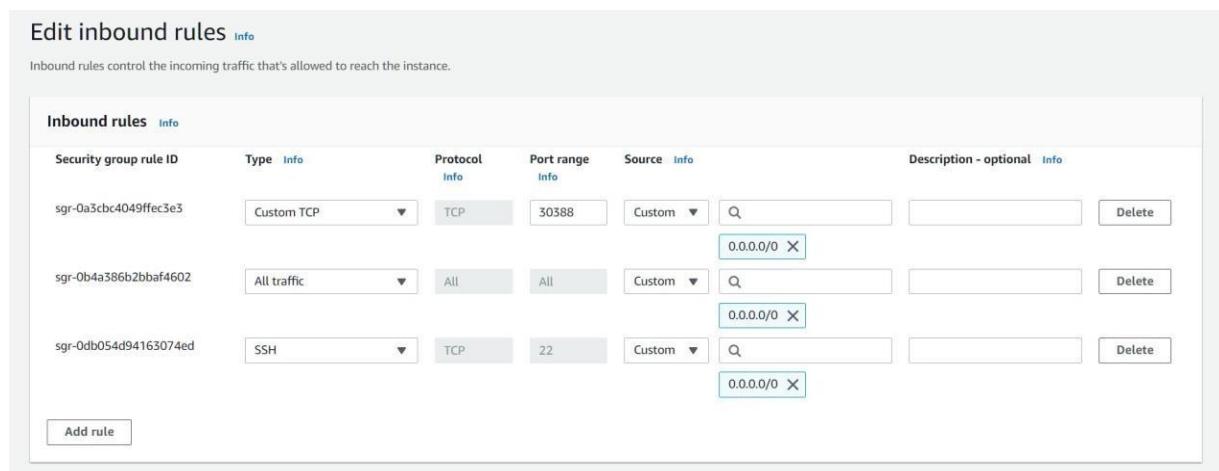


A screenshot of the AWS CloudShell interface. At the top, there's a navigation bar with the AWS logo, 'Services' (selected), a search bar ('Search for services, features, blogs, docs, and more'), and a keyboard shortcut '[Alt+S]'. Below the bar, the terminal window shows the command being run: 'ubuntu@master-node:~\$ kubectl expose deploy nginx --port 80 --target-port 80 --type NodePort'. The response 'service/nginx exposed' is shown, followed by the prompt 'ubuntu@master-node:~\$'. The rest of the terminal window is blacked out.

Step 4: Run this command to see a summary of the service and the ports exposed.

```
$kubectl get services
```

Step 5: Add the port which is displayed i.e. 30388 (in our case) in the inbound rules of the security group.

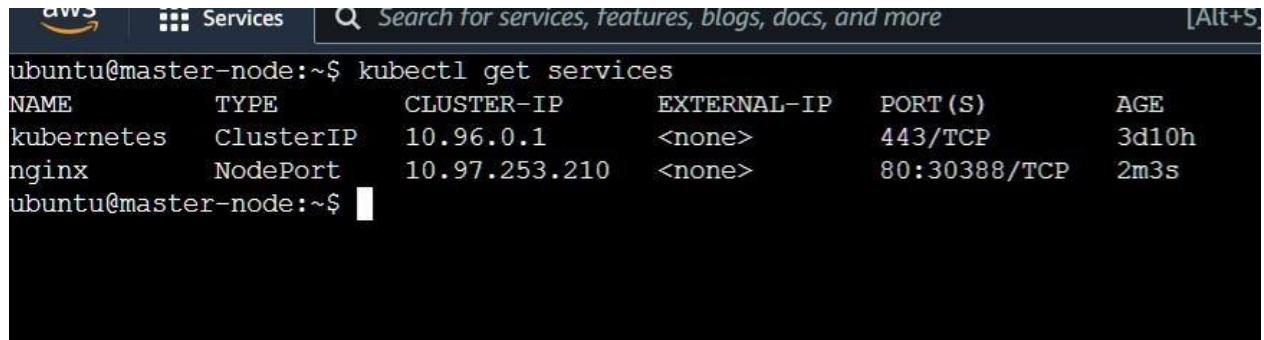


A screenshot of the AWS CloudFront 'Edit inbound rules' page. The title is 'Edit inbound rules Info'. A note below says 'Inbound rules control the incoming traffic that's allowed to reach the instance.' The main table has columns: 'Inbound rules Info', 'Security group rule ID', 'Type Info', 'Protocol Info', 'Port range Info', 'Source Info', and 'Description - optional Info'. There are three rows of rules:

Inbound rules <small>Info</small>	Security group rule ID	Type <small>Info</small>	Protocol <small>Info</small>	Port range <small>Info</small>	Source <small>Info</small>	Description - optional <small>Info</small>
	sgr-0a3cbc4049ffec3e3	Custom TCP	TCP	30388	Custom	<input type="text"/> 0.0.0.0/0 <small>X</small>
	sgr-0b4a386b2bbaf4602	All traffic	All	All	Custom	<input type="text"/> 0.0.0.0/0 <small>X</small>
	sgr-0db054d94163074ed	SSH	TCP	22	Custom	<input type="text"/> 0.0.0.0/0 <small>X</small>

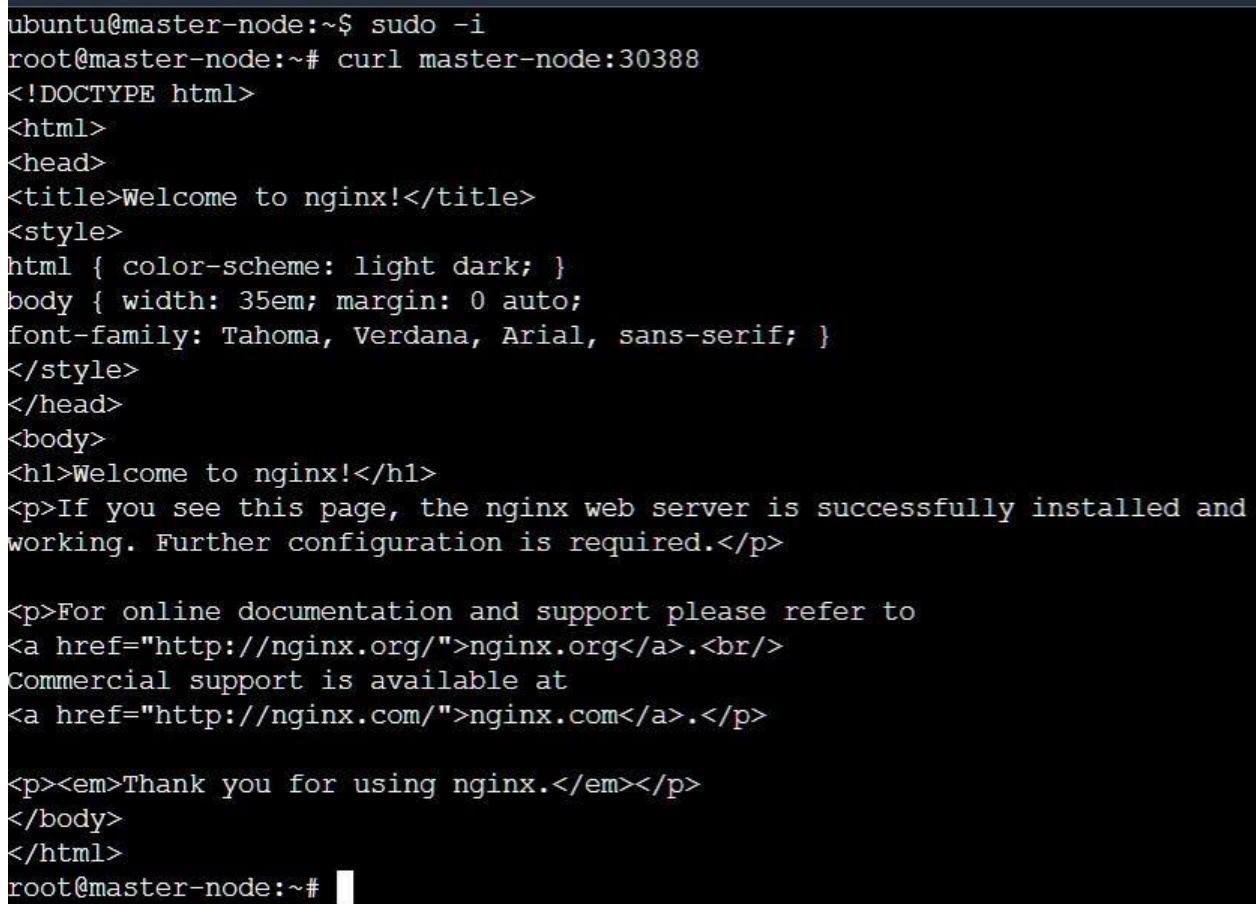
At the bottom left is a button 'Add rule'.

Step 6: Now you can verify that the Nginx page is reachable on all nodes using the curl command.



A screenshot of a web-based terminal interface. At the top, there's a navigation bar with the AWS logo, a "Services" button, a search bar containing "Search for services, features, blogs, docs, and more", and a keyboard shortcut "[Alt+S]". Below the bar is a black terminal window. The terminal output shows the user running the command "kubectl get services". The output lists two services: "kubernetes" (ClusterIP, 10.96.0.1, 443/TCP, 3d10h) and "nginx" (NodePort, 10.97.253.210, 80:30388/TCP, 2m3s).

```
ubuntu@master-node:~$ kubectl get services
NAME      TYPE      CLUSTER-IP      EXTERNAL-IP      PORT(S)      AGE
kubernetes  ClusterIP  10.96.0.1      <none>        443/TCP      3d10h
nginx      NodePort   10.97.253.210  <none>        80:30388/TCP  2m3s
ubuntu@master-node:~$
```



A screenshot of a terminal window. The user runs "sudo -i" to become root. Then, they run "curl master-node:30388", which returns the default Nginx welcome page. The page content includes HTML code for the title, styles, head, body, and various paragraphs, along with links to nginx.org and nginx.com. The terminal prompt "root@master-node:~#" is visible at the bottom.

```
ubuntu@master-node:~$ sudo -i
root@master-node:~# curl master-node:30388
<!DOCTYPE html>
<html>
<head>
<title>Welcome to nginx!</title>
<style>
html { color-scheme: light dark; }
body { width: 35em; margin: 0 auto;
font-family: Tahoma, Verdana, Arial, sans-serif; }
</style>
</head>
<body>
<h1>Welcome to nginx!</h1>
<p>If you see this page, the nginx web server is successfully installed and
working. Further configuration is required.</p>
<p>For online documentation and support please refer to
<a href="http://nginx.org/">nginx.org</a>.<br/>
Commercial support is available at
<a href="http://nginx.com/">nginx.com</a>.</p>
<p><em>Thank you for using nginx.</em></p>
</body>
</html>
root@master-node:~#
```

As you can see, the “**WELCOME TO NGINX!**” page can be reached.

Step 7: To test that everything is working, visit `http://worker_1_ip:nginx_port` or `http://worker_2_ip:nginx_port` through a browser on your local machine. You will see Nginx’s familiar welcomepage.

<http://52.90.129.234:30388>

Welcome to nginx!

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

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

Thank you for using nginx.

ADVANCE DEVOPS EXP-7

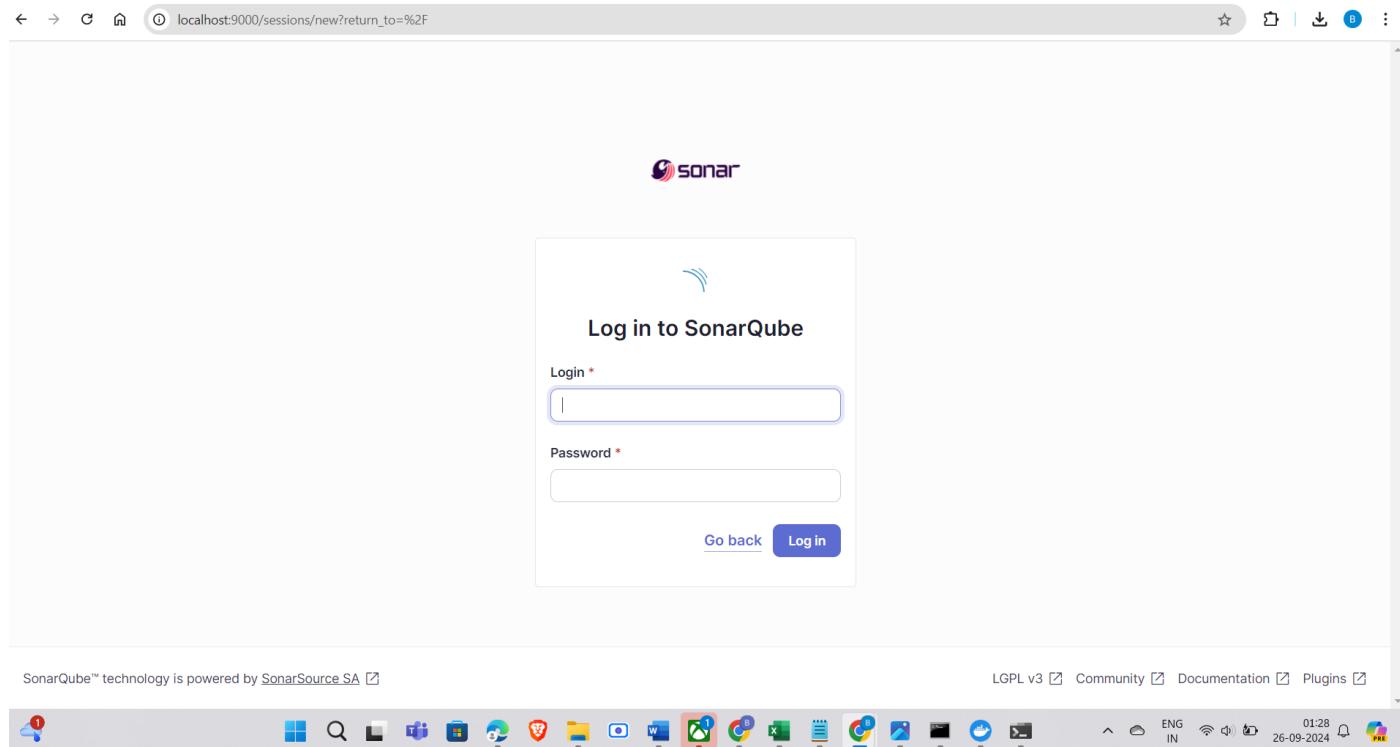
Aim :- To understand Static Analysis SAST process and learn to integrate Jenkins SAST to SonarQube/GitLab.

Step 1: Installing SonarQube from the Docker Image

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

```
PS C:\Windows\system32> docker run -d --name sonarqube -e SONAR_ES_BOOTSTRAP_CHECKS_DISABLE=true -p 9000:9000 sonarqube:latest  
Unable to find image 'sonarqube:latest' locally  
latest: Pulling from library/sonarqube  
7478e0ac0f23: Pull complete  
90a925ab929a: Pull complete  
7d9a34308537: Pull complete  
80338217a4ab: Pull complete  
1a5fd5c7e184: Pull complete  
7b87d6fa783d: Pull complete  
bd819c9b5ead: Pull complete  
4f4fb700ef54: Pull complete  
Digest: sha256:72e9feec71242af83faf65f95a40d5e3bb2822a6c3b2cda8568790f3d31aecde  
Status: Downloaded newer image for sonarqube:latest  
22ee358a0f88adabcb823e6eccebb28975b916b4b116b619e567666c8d3860d1  
PS C:\Windows\system32>
```

Step 2: After installation of SonarQube, go to the SonarQube page by typing: <http://localhost:9000/> on your browser. If you see such page then you have successfully installed it.



Step 3: Login using the username as “admin” and password as “admin”. And then you will see the home page of SonarQube

The screenshot shows the SonarQube interface with the title "How do you want to create your project?". It lists several import options: "Import from Azure DevOps" (Setup), "Import from Bitbucket Cloud" (Setup), "Import from Bitbucket Server" (Setup), "Import from GitHub" (Setup), "Import from GitLab" (Setup), and a "Create a local project" button. A warning message at the bottom left states: "⚠️ Embedded database should be used for evaluation purposes only. The embedded database will not scale, it will not support upgrading to newer versions of SonarQube, and there is no support for migrating your data out of it into a different database engine." The footer includes links for SonarSource SA, Community Edition v10.6 (ACTIVE), LGPL v3, Community, Documentation, Plugins, and Web API.

Step 4 :- give the name project

The screenshot shows the "Create a local project" step of the setup wizard. It asks for a "Project display name" (sonarqube-test) and a "Project key" (sonarqube-test). It also has a field for "Main branch name" (main) with a note about the default branch. At the bottom are "Cancel" and "Next" buttons. A warning message at the bottom left states: "⚠️ Embedded database should be used for evaluation purposes only. The embedded database will not scale, it will not support upgrading to newer versions of SonarQube, and there is no support for migrating your data out of it into a different database engine." The footer includes links for SonarSource SA, Community Edition v10.6 (ACTIVE), LGPL v3, Community, Documentation, Plugins, and Web API.

Step 5: Go to Dashboard > Manage Jenkins > Plugin Manager and search for SonarQube Scanner under Available plugins for Jenkins and install it

The screenshot shows the Jenkins Plugins page. A search bar at the top right contains the text "SONAR". Below it, a sidebar on the left lists "Updates" (8), "Available plugins", "Installed plugins" (selected), and "Advanced settings". The main content area displays the "SonarQube Scanner for Jenkins" plugin version 2.17.2. It has a brief description: "This plugin allows an easy integration of SonarQube, the open source platform for Continuous Inspection of code quality." Below the description is a "Report an issue with this plugin" link. To the right of the description are two buttons: a blue "Enabled" button with a checkmark and a red "Disabled" button with a crossed-out X. At the bottom right of the page are links for "REST API" and "Jenkins 2.462.2".

Step 6:- add authentication all details

The screenshot shows the Jenkins "New credentials" page under "Manage Jenkins > Credentials > System > Global credentials (unrestricted)". The "Kind" dropdown is set to "Secret text". The "Scope" dropdown is set to "Global (Jenkins, nodes, items, all child items, etc)". The "Secret" field contains a masked password. The "ID" field is filled with "reatil-tokcen". The "Description" field contains "reatil tokcen". At the bottom is a "Create" button.

Step 7: Search SonarQube Scanner under Dashboard > Manage Jenkins > Global Tool Configuration. Choose the latest configuration and choose Install Automatically

Dashboard > Manage Jenkins > Global Tool Configuration

SonarQube Scanner installations

List of SonarQube Scanner installations on this system

Add SonarQube Scanner

☰ SonarQube Scanner

Name: AdvDevops-EXP7

Install automatically ?

☰ Install from Maven Central

Version: SonarQube Scanner 4.7.0.2747

Add Installer

Add SonarQube Scanner

Save Apply

This screenshot shows the Jenkins Global Tool Configuration page. Under the 'SonarQube Scanner installations' section, there is a form to add a new installation. The 'Name' field contains 'AdvDevops-EXP7'. The 'Install automatically?' checkbox is checked. Below it, the 'Install from Maven Central' section is expanded, showing the selected version 'SonarQube Scanner 4.7.0.2747'. There are 'Save' and 'Apply' buttons at the bottom.

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

Jenkins

Search (CTRL+K) ? 🔍

bbpatil

Dashboard > All > New Item

New Item

Enter an item name

Select an item type

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

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

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

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

OK

This screenshot shows the 'New Item' dialog in Jenkins. It asks for an item name and provides four options for item type: 'Freestyle project', 'Maven project', 'Pipeline', and 'Multi-configuration project'. Each option has a brief description. At the bottom is an 'OK' button.

Step 9 :- add the git url and also all details

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

**step 10 :- under the bulid command all the details plz correctly and give
Mention the SonarQube Project Key, Login, Password, and
Host URL. sonar.projectKey=retail-stock
sonar.login=*your username*
sonar.password=*your password*
sonar.hosturl=http://localhost:9000**

Repositories ?

Repository URL ?

Credentials ? Advanced ▾

Add Repository

Branches to build ?

Branch Specifier (blank for 'any') ?

Build Steps

Execute SonarQube Scanner

JDK ?

SonarQube Scanner
! Jenkins needs to know where your SonarQube Scanner is installed.
Please do so from the global tool configuration.

Path to project properties ?

Analysis properties ?

```
sonar.projectKey=sonarqube-test
sonar.projectName=sonarqube-test
sonar.projectVersion=1.0
sonar.language=java
sonar.tests=src/test/java
```

Additional arguments ?

Step 11: Go to <http://localhost:9000/> and enter your previously created username. Go to Permissions and grant the Admin user Execute Permissions.

Permissions

Grant and revoke project-level permissions. Permissions can be granted to groups or individual users. This project is public. Anyone can browse and see the source code.

Public Private

All	Users	Groups	Search for users or groups...	Administer Issues	Administer Security Hotspots	Administer	Execute Analysis
			sonar-administrators System administrators	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
			sonar-users Any new users created will automatically join this group	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
			Administrator admin	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>
			Anyone	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

4 of 4 shown

! Embedded database should be used for evaluation purposes only
The embedded database will not scale, it will not support upgrading to newer versions of SonarQube, and there is no support for migrating your data out of it into a different database engine.

SonarQube™ technology is powered by SonarSource SA
Community Edition - Version 9.6.1 (build 59631) - LGPL v3 - Community - Documentation - Plugins - Web API

Step 12: Run The Build

Dashboard > reatil-store-jenkin-project >

Status reatil-store-jenkin-project Add description

<> Changes
Workspace
Build Now
Configure
Delete Project
SonarQube
Rename

SonarQube
Permalinks

Build History trend ▾
No builds

No builds
Atom feed for all Atom feed for failures

REST API Jenkins 2.462.2

console output

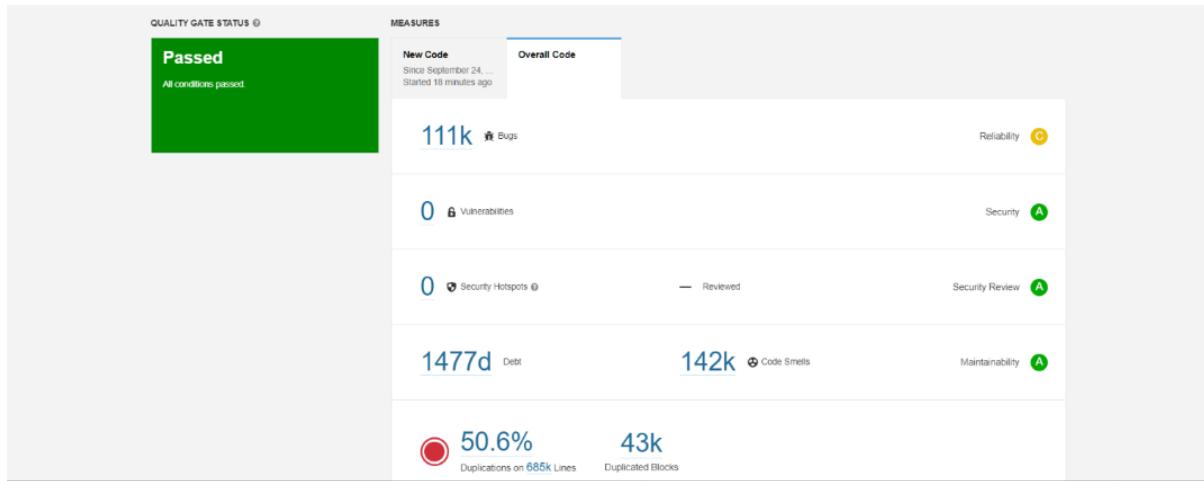
```
Started by user bpatil
Running as SYSTEM
Building in workspace C:\ProgramData\Jenkins\jenkins\workspace\plzhoha
The recommended git tool is: NONE
No credentials specified
Cloning the remote Git repository
Cloning repository https://github.com/shazforiot/MSBuild_firstproject.git
> git.exe init C:\ProgramData\Jenkins\jenkins\workspace\plzhoha # 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 config remote.origin.url https://github.com/shazforiot/MSBuild_firstproject.git # timeout=10
> git.exe config --add remote.origin.fetch +refs/heads/*:refs/remotes/origin/* # timeout=10
Avoid second fetch
> 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"
First time build. Skipping changelog.
[plzhoha] $ C:\ProgramData\Jenkins\jenkins\tools\hudson.plugins.sonar.SonarRunnerInstallation\sonarqube\bin\sonar-scanner.bat -
-Dsonar.host.url=http://localhost:9000 ***** "-Dsonar.projectKey=pipeline ok1" -Dsonar.host.url=http://localhost:9000 -Dsonar.source=HelloWorldCare
-Dsonar.projectBaseDir=C:\ProgramData\Jenkins\jenkins\workspace\plzhoha
00:39:45.044 WARN Property 'sonar.host.url' with value 'http://localhost:9000' is overridden with value 'http://localhost:9000'
00:39:45.059 INFO Scanner configuration file:
C:\ProgramData\Jenkins\jenkins\tools\hudson.plugins.sonar.SonarRunnerInstallation\sonarqube\bin..\conf\sonar-scanner.properties
00:39:45.059 INFO Configuration file loaded from C:\ProgramData\Jenkins\jenkins\tools\hudson.plugins.sonar.SonarRunnerInstallation\sonarqube\bin..\conf\sonar-scanner.properties
```

```

Commit message: "updated"
First time build. Skipping changelog.
[plzhoha] $ C:\ProgramData\Jenkins\jenkins\tools\hudson.plugins.sonar.SonarRunnerInstallation\sonarqube\bin\sonar-scanner.bat -
Dsonar.host.url=http://localhost:9000 ***** "-Dsonar.projectKey=pipeline ok1" -Dsonar.host.url=http://localhost:9000 -Dsonar.source=HelloWorldCare
-Dsonar.projectBaseDir=C:\ProgramData\Jenkins\jenkins\workspace\plzhoha
00:39:45.044 WARN Property 'sonar.host.url' with value 'http://localhost:9000' is overridden with value 'http://localhost:9000'
00:39:45.059 INFO Scanner configuration file:
C:\ProgramData\Jenkins\jenkins\tools\hudson.plugins.sonar.SonarRunnerInstallation\sonarqube\bin..\conf\sonar-scanner.properties
00:39:45.060 INFO Project root configuration file: NONE
00:39:45.076 INFO SonarScanner CLI 6.1.0.4477
00:39:45.077 INFO Java 21.0.4 Eclipse Adoptium (64-bit)
00:39:45.082 INFO Windows 11 10.0 amd64
00:39:45.109 INFO User cache: C:\Windows\system32\config\systemprofile\.sonar\cache
00:39:45.909 INFO JRE provisioning: os[windows], arch[amd64]
00:39:48.935 INFO Communicating with SonarQube Server 10.6.0.92116
00:39:49.380 INFO Starting SonarScanner Engine...
00:39:49.381 INFO Java 17.0.11 Eclipse Adoptium (64-bit)
00:39:50.215 INFO Load global settings
00:39:50.408 INFO Load global settings (done) | time=191ms
00:39:50.417 INFO Server id: 147B411E-AZIp-aMmENyc0d53ifbB
00:39:50.430 INFO Loading required plugins
00:39:50.430 INFO Load plugins index
00:39:50.554 INFO Load plugins index (done) | time=123ms
00:39:50.554 INFO Load/download plugins
00:39:51.123 INFO Load/download plugins (done) | time=570ms
00:39:51.447 INFO Process project properties
00:39:51.449 INFO Process project properties (done) | time=2ms
00:39:51.475 ERROR Validation of project failed:
  o "pipeline ok1" is not a valid project key. Allowed characters are alphanumeric, '-', '_', '.' and ':', with at least one non-digit.
00:39:51.547 INFO EXECUTION FAILURE
00:39:51.548 INFO Total time: 6.491s

```

Due to blocking system git this screenshot form clg pc



Name:- Swaraj Patil

Roll :- 40

Experiment no 8

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.

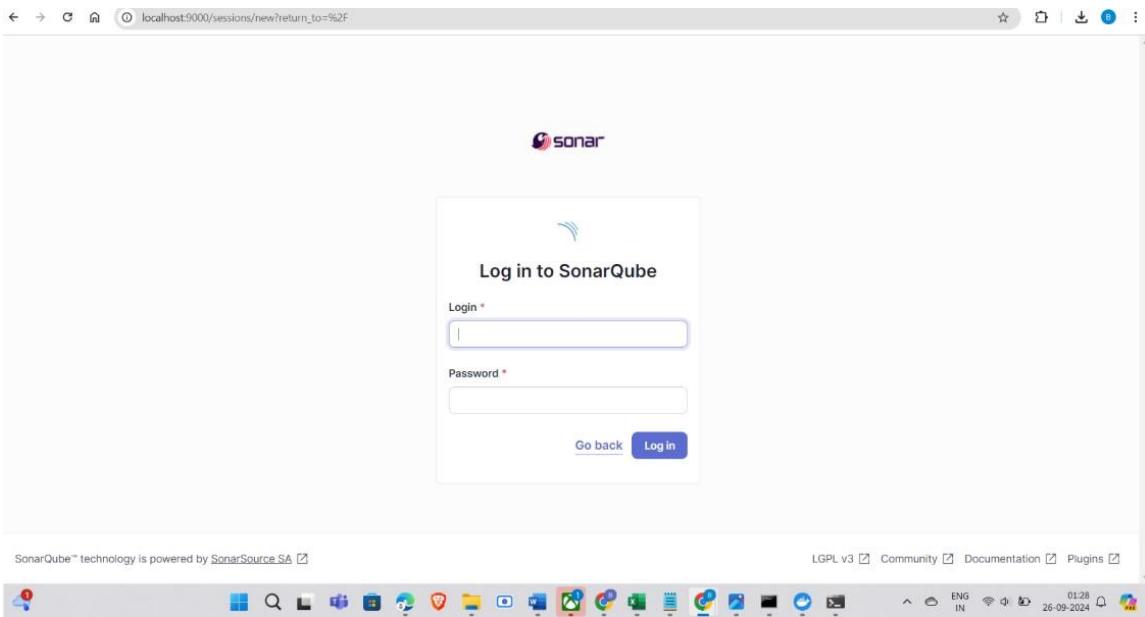
Step 1: Download sonar scanner

The screenshot shows the SonarScanner CLI page on the SonarQube website. The left sidebar includes links for 'Homepage', 'Try out SonarQube', 'Server installation and setup', 'Analyzing source code' (selected), 'Scanners' (selected), 'Scanner environment', 'SonarScanner CLI' (selected), 'SonarQube extension for Azure DevOps', 'SonarQube extension for Jenkins', 'SonarScanner for .NET', 'SonarScanner for Maven', 'SonarScanner for Gradle', 'SonarScanner for NPM' (selected), 'SonarScanner for Ant (Deprecated)', 'SonarScanner for Python (Beta)', and 'Analysis parameters'. The main content area displays the 'SonarScanner CLI' section, which includes a table for version 6.2 (released 2024-09-17) showing supported platforms: Linux x64, Linux AArch64, Windows x64, macOS x64, macOS AArch64, Docker Any (Requires a pre-installed JVM), and Release notes. A note at the bottom states: 'The SonarScanners run on code that is checked out. See Verifying the code checkout step of your build.' On the right, there's a sidebar titled 'On this page' with links to 'Configuring your project', 'Running SonarScanner CLI from the zip file', 'Running SonarScanner CLI from the Docker image', 'Scanning C, C++, or Objective-C projects', 'Sample projects', 'Alternatives to sonar-project.properties', 'Alternate analysis directory', 'Advanced configuration', and 'Troubleshooting'.

1. Command: docker pull sonarqube

```
C:\Windows\System32>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
```

2. Login the details



3. Give the name of the project

1 of 2

Create a local project

Project display name *

 (Valid)

Project key *

 (Valid)

Main branch name *

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

Cancel Next

⚠️ **Embedded database should be used for evaluation purposes only.**
The embedded database will not scale, it will not support upgrading to newer versions of SonarQube, and there is no support for migrating your data out of it into a different database engine.

SonarQube™ technology is powered by SonarSource SA [Community Edition v10.6 \(92116\) ACTIVE](#) [LGPL v3](#) [Community](#) [Documentation](#) [Plugins](#) [Web API](#)

4. Open up Jenkins Dashboard on localhost, port 8081 or which ever port it is for you.

[+ New Item](#)[Build History](#)[Manage Jenkins](#)[My Views](#)[All](#)[+](#)[Add description](#)

S	W	Name ↓	Last Success	Last Failure	Last Duration	
✓	☀	bbparil	10 days #1	N/A	5.3 sec	▶
...	☀	bhagyesh	N/A	N/A	N/A	▶
✗	☁	bhagyesh_patil_opo	N/A	1 day 0 hr #2	8.7 sec	▶
✗	☁	bhagyesh_mavev	N/A	1 day 0 hr #2	7.8 sec	▶
...	☀	bhagyesh_niraj	N/A	N/A	N/A	▶
✗	☁	bhagyesh_Niraj_final	N/A	6 hr 13 min #1	4.7 sec	▶
✓	☀	bhagyeshpipi	8 days 10 hr #1	N/A	5.5 sec	▶
✗	☁	bp_nk	N/A	6 hr 13 min #1	9.4 sec	▶
✓	☀	lipo	1 day 0 hr #1	N/A	0.36 sec	▶

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

Name [X](#)

Server URL [1.Under Jenkins 'Manage Jenkins' the n go to 'system', scroll and look for Default is http://localhost:9000](#)

Server authentication token
SonarQube authentication token. Mandatory when anonymous access is disabled.

[+ Add ▾](#)

[Advanced ▾](#)

2. SonarQube Servers

and enter the details.

Enter the Server Authentication token if needed.

In SonarQube installations: Under **Name** add <project name of sonarqube> for me
adv_devops_7_sonarqube

In **Server URL** Default is <http://localhost:9000>

The screenshot shows the Jenkins 'Manage Jenkins' > 'Plugins' page. A search bar at the top contains the text 'sonarq'. Below it, a sidebar has tabs for 'Updates' (25), 'Available plugins' (selected), 'Installed plugins', and 'Advanced settings'. The main area lists the 'SonarQube Scanner' plugin, version 2.17.2, released 6 months and 29 days ago. It includes links for 'Install', 'External Site/Tool Integrations', and 'Build Reports'. A note below the plugin says: 'This plugin allows an easy integration of SonarQube, the open source platform for Continuous Inspection of code quality.'

6. After configuration, create a New Item → choose a pipeline project.

The screenshot shows the Jenkins 'New Item' dialog. In the 'Enter an item name' field, 'lipo1' is typed. Under 'Select an item type', the 'Pipeline' option is selected, which is described as 'Orchestrates long-running activities that can span multiple build agents. Suitable for building pipelines (formerly known as workflows) and/or organizing complex activities that do not easily fit in free-style job type.' Other options shown include 'Freestyle project', 'Maven project', 'Multi-configuration project', and 'Folder'. A blue 'OK' button is at the bottom right.

This is java code put in the pipeline script

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=<SonarQube_URL>(default:
http://localhost:9000/)"""
}

}

```

Definition

Pipeline script

Script ?

```

1+ node {
2+ stage('Cloning the GitHub Repo') {
3+   git 'https://github.com/shazforiot/GOL.git'
4+ }
5+
6+ stage('SonarQube analysis') { withSonarQubeEnv('<Name_of_SonarQube_environment_on_Jenkins>') {
7+   sh """
8+     <PATH_TO SONARQUBE_SCANNER_FOLDER>/bin/sonar-scanner \
9+       -D sonar.login=admin \
10+      -D sonar.password=admin \
11+      -D sonar.projectKey=sonarqube \
12+      -D sonar.exclusions=vendor/**,resources/**,**/*.java \
13+      -D sonar.host.url=http://localhost:9000
14+ """
15+ }
16+ }
17+ }
18+

```

Use Groovy Sandbox ?

[Pipeline Syntax](#)

7. Build project

Stage View



Permalinks



- Last build (#2), 1 day 1 hr ago
- Last failed build (#2), 1 day 1 hr ago
- Last unsuccessful build (#2), 1 day 1 hr ago
- Last completed build (#2), 1 day 1 hr ago

This error come in pc but next step output come in clg pc

✖ Console Output

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

```
Started by user bbpatil
[Pipeline] Start of Pipeline
[Pipeline] node
Running on Jenkins in C:\ProgramData\Jenkins\.jenkins\workspace\lipo12
[Pipeline] {
[Pipeline] stage
[Pipeline] { (Cloning the GitHub Repo)
[Pipeline] git
The recommended git tool is: NONE
No credentials specified
Cloning the remote Git repository
Cloning repository https://github.com/shazforiot/GOL.git
> git.exe init C:\ProgramData\Jenkins\.jenkins\workspace\lipo12 # timeout=10
Fetching upstream changes from https://github.com/shazforiot/GOL.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/GOL.git +refs/heads/*:refs/remotes/origin/* # timeout=10
> git.exe config remote.origin.url https://github.com/shazforiot/GOL.git # timeout=10
> git.exe config --add remote.origin.fetch +refs/heads/*:refs/remotes/origin/* # timeout=10
Avoid second fetch
> git.exe rev-parse "refs/remotes/origin/master^{commit}" # timeout=10
Checking out Revision ba799ba7e1b576f04a4612322b0412c5e6e1e5e4 (refs/remotes/origin/master)
> git.exe config core.sparsecheckout # timeout=10
> git.exe checkout -f ba799ba7e1b576f04a4612322b0412c5e6e1e5e4 # timeout=10
> git.exe branch -a -v --no-abbrev # timeout=10
> git.exe checkout -b master ba799ba7e1b576f04a4612322b0412c5e6e1e5e4 # timeout=10
Commit message: "Update Jenkinsfile"
```

8. Check Console

>Status

Changes

Console Output

View as plain text

Edit Build Information

Delete build '#9'

Timings

Git Build Data

Pipeline Overview

Pipeline Console

Replay

Pipeline Steps

Workspaces

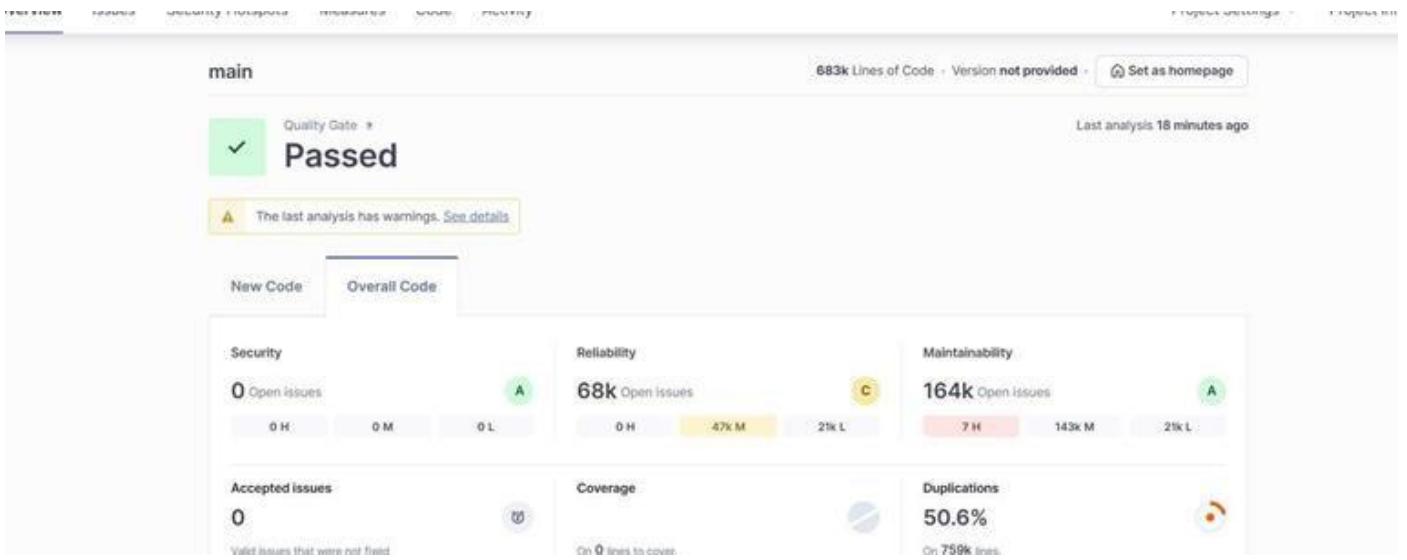
Previous Build

Console Output

Skipping 4,246 KB. Full Log

```
16:19:49.751 WARN Too many duplication references on file gameoflife-web/tools/jmeter/docs/api/org/apache/jmeter/visualizers/PropertyControlGui.html for block at line 512. Keep only the first 100 references.
16:19:49.751 WARN Too many duplication references on file gameoflife-web/tools/jmeter/docs/api/org/apache/jmeter/visualizers/PropertyControlGui.html for block at line 248. Keep only the first 100 references.
16:19:49.751 WARN Too many duplication references on file gameoflife-web/tools/jmeter/docs/api/org/apache/jmeter/visualizers/PropertyControlGui.html for block at line 886. Keep only the first 100 references.
16:19:49.751 WARN Too many duplication references on file gameoflife-web/tools/jmeter/docs/api/org/apache/jmeter/visualizers/PropertyControlGui.html for block at line 249. Keep only the first 100 references.
16:19:49.751 WARN Too many duplication references on file gameoflife-web/tools/jmeter/docs/api/org/apache/jmeter/visualizers/PropertyControlGui.html for block at line 662. Keep only the first 100 references.
16:19:49.751 WARN Too many duplication references on file gameoflife-web/tools/jmeter/docs/api/org/apache/jmeter/visualizers/PropertyControlGui.html for block at line 615. Keep only the first 100 references.
16:19:49.751 WARN Too many duplication references on file gameoflife-web/tools/jmeter/docs/api/org/apache/jmeter/visualizers/PropertyControlGui.html for block at line 664. Keep only the first 100 references.
16:19:49.751 WARN Too many duplication references on file gameoflife-web/tools/jmeter/docs/api/org/apache/jmeter/visualizers/PropertyControlGui.html for block at line 913. Keep only the first 100 references.
16:19:49.752 WARN Too many duplication references on file gameoflife-web/tools/jmeter/docs/api/org/apache/jmeter/visualizers/PropertyControlGui.html for block at line 810. Keep only the first 100 references.
16:19:49.752 WARN Too many duplication references on file gameoflife-web/tools/jmeter/docs/api/org/apache/jmeter/visualizers/PropertyControlGui.html for block at line 668. Keep only the first 100 references.
16:19:49.752 WARN Too many duplication references on file gameoflife-web/tools/jmeter/docs/api/org/apache/jmeter/visualizers/PropertyControlGui.html for block at line 548. Keep only the first 100 references.
16:19:49.752 WARN Too many duplication references on file gameoflife-web/tools/jmeter/docs/api/org/apache/jmeter/visualizers/PropertyControlGui.html for block at line 543. Keep only the first 100 references.
16:19:49.752 WARN Too many duplication references on file gameoflife-web/tools/jmeter/docs/api/org/apache/jmeter/visualizers/PropertyControlGui.html for block at line 152. Keep only the first 100 references.
```

9. Now, check the project in SonarQube



10 .Consistency

Overview Issues Security measures Measures User Activity Project Overview Project Management

My Issues All

Filters Clear All Filters

Issues in new code

↳ Clean Code Attribute 1 X

- Consistency 107k
- Intentionality 14k
- Adaptability 0
- Responsibility 0

Add to selection Ctrl + click

Bulk Change Select issues ▾ Navigate to issue ▾ 196,662 issues 3075d effort

gameoflife-core/build/reports/tests/all-tests.html

Insert a <!DOCTYPE> declaration to before this <html> tag. Consistency

Reliability ⓘ user-experience ⓘ

Open Not assigned L1 - 5min effort - 4 years ago - ⚡ Bug - ⚡ Major

Remove this deprecated "width" attribute. Consistency

Maintainability ⓘ HTML5 - obsolete ⓘ

Open Not assigned L1 - 5min effort - 4 years ago - ⚡ Code Smell - ⚡ Major

Remove this deprecated "align" attribute. Consistency

Maintainability ⓘ HTML5 - obsolete ⓘ

Open Not assigned L11 - 5min effort - 4 years ago - ⚡ Code Smell - ⚡ Major

11. Intentionality

The screenshot shows a software interface for managing code quality and security. At the top, there are tabs for Overview, Issues, Security Hotspots, Measures, Code, and Activity. On the right, there are Project Settings and Project Information buttons. The main area displays a list of issues under the heading "gameoflife-acceptance-tests/Dockerfile".
Filters: My Issues, All, Clear All Filters.
Issues in new code:
Clean Code Attribute: 197k total, with Intentionality set to 14k. Other categories like Consistency and Adaptability have 0 issues.
Add to selection: Ctrl + click.
Software Quality: Security has 0 issues, Reliability has 14k issues.
Issues List:

- Use a specific version tag for the image. **Intentionality** (highlighted). **Maintainability**. Status: Open. Last updated: 4 years ago. Effort: 5min. Priority: Code Smell - Major.
- Surround this variable with double quotes; otherwise, it can lead to unexpected behavior. **Intentionality**. **Maintainability**. Status: Open. Last updated: 4 years ago. Effort: 5min. Priority: Code Smell - Major.
- Surround this variable with double quotes; otherwise, it can lead to unexpected behavior. **Intentionality**. **Maintainability**. Status: Open. Last updated: 4 years ago. Effort: 5min. Priority: Code Smell - Major.

12. Bugs

The screenshot shows a software interface for managing bugs and test reports. At the top, there are buttons for Bulk Change, Select issues, Navigate to issue, and statistics: 67,624 issues and 1648d effort. The main area displays a list of bugs for the file "gameoflife-core/build/reports/tests/all-tests.html".
Bulk Change button.
Select issues and **Navigate to issue** buttons.
Issues List:

- Add "lang" and/or "xml:lang" attributes to this "<html>" element. **Intentionality**. **Reliability**. **accessibility** (highlighted) **wcag2-a**. Status: Open. Last updated: 4 years ago. Effort: 2min. Priority: Bug - Major.
- Insert a <!DOCTYPE> declaration to before this <html> tag. **Consistency**. **Reliability**. **user-experience**. Status: Open. Last updated: 4 years ago. Effort: 5min. Priority: Bug - Major.
- Add "<th>" headers to this "<table>". **Intentionality**. **Reliability**. **accessibility** (highlighted) **wcag2-a**. Status: Open. Last updated: 4 years ago. Effort: 2min. Priority: Bug - Major.

13. Code smell

The screenshot shows the SonarQube interface for the project "gameoflife-acceptance-tests/Dockerfile". The left sidebar displays various filters and issue counts. The main area lists three code smell issues under the "Clean Code Attribute" category:

- Issue: Use a specific version tag for the image. Status: Open, Not assigned. Intentionality: No tags. Impact: Minor effort / 4 years ago. Type: Code Smell, Major.
- Issue: Surround this variable with double quotes; otherwise, it can lead to unexpected behavior. Status: Open, Not assigned. Intentionality: No tags. Impact: Minor effort / 4 years ago. Type: Code Smell, Major.
- Issue: Surround this variable with double quotes; otherwise, it can lead to unexpected behavior. Status: Open, Not assigned. Intentionality: No tags. Impact: Minor effort / 4 years ago. Type: Code Smell, Major.

14. Cyclomatic Complexities

The screenshot shows the SonarQube interface for the project "gameoflife-acceptance-tests/Dockerfile". The left sidebar lists various measures, with "Cyclomatic Complexity" selected. The main area displays the cyclomatic complexity for different components:

Component	Cyclomatic Complexity
gameoflife-acceptance-tests	—
gameoflife-build	—
gameoflife-core	18
gameoflife-deploy	—
gameoflife-web	1,094
pom.xml	—

Bottom right corner: 6 of 6 shown.

ADVANCE DEVOPS EXP 9

Name :- Swaraj Patil

Roll_no :- 40

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

1. Create an Amazon Linux EC2 Instance

The screenshot shows the AWS EC2 Instances page. On the left, there's a navigation sidebar with options like EC2 Dashboard, EC2 Global View, Events, Console-to-Code Preview, Instances (selected), Instance Types, Launch Templates, Spot Requests, Savings Plans, Reserved Instances, Dedicated Hosts, Capacity Reservations, Images (AMIs, AMI Catalog), and Elastic Block Store (Volumes, Snapshots). The main content area displays a table titled 'Instances (1/1)'. It shows one instance: 'nagios-host' (Instance ID: i-Oce5684c780a9ed59, Status: Running, Instance type: t2.micro, Status check: 2/2 checks passed, Availability Zone: us-east-1c, Public IP: ec2-54-17). There are buttons for 'Connect', 'Actions', and 'Launch instances' at the top of the table. At the bottom of the table, there's a link to 'View alarms' and the instance ID 'i-Oce5684c780a9ed59 (nagios-host)'. The footer includes links for CloudShell, Feedback, and copyright information: © 2024, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences.

2. Configure Security Group

- Ensure HTTP, HTTPS, SSH, and ICMP are open from everywhere.
- Edit the inbound rules of the specified Security Group

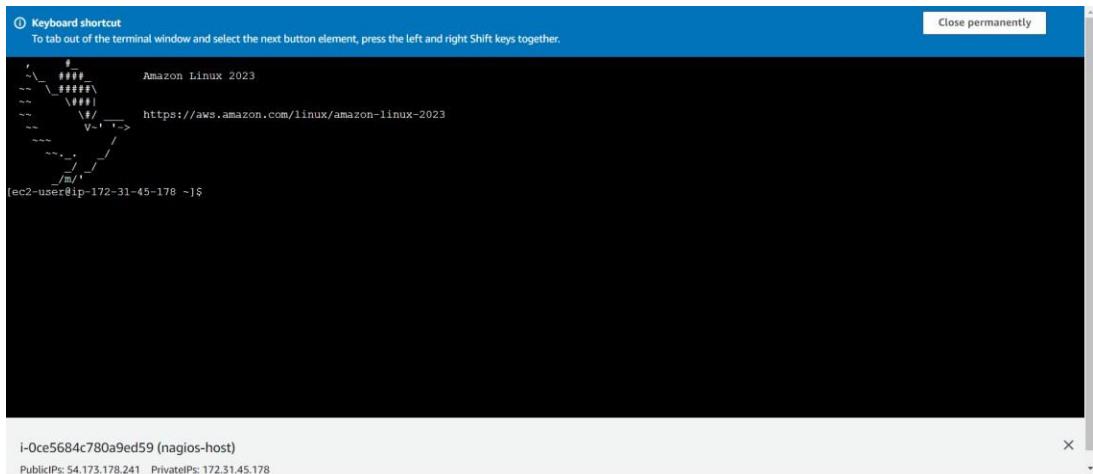
The screenshot shows the AWS Security Groups page. The URL is EC2 > Security Groups > sg-0b6d663161209f32d - launch-wizard-11 > Edit inbound rules. The page title is 'Edit inbound rules' with a 'Info' link. A note says 'Inbound rules control the incoming traffic that's allowed to reach the instance.' Below this is a table titled 'Inbound rules' with an 'Info' link. The columns are: Security group rule ID, Type, Protocol, Port range, Source, and Description - optional. There are five rows in the table:

- sgr-00d59807a11e23687: All ICMP - IPv4, ICMP, All, Custom, 0.0.0.0/0, Delete
- sgr-0ae620ec0b187c4a7: All traffic, All, All, Custom, 0.0.0.0/0, Delete
- sgr-0775d4388ffe14db6: SSH, TCP, 22, Custom, 0.0.0.0/0, Delete
- sgr-0ebadedcb97cb60fc: HTTP, TCP, 80, Custom, 0.0.0.0/0, Delete
- sgr-08983e0020306b273: HTTPS, TCP, 443, Custom, 0.0.0.0/0, Delete

The footer includes links for CloudShell, Feedback, and copyright information: © 2024, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences.

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.



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

```
sudo yum update
```

```
[ec2-user@ip-172-31-40-254 ~]$ sudo yum update
Last metadata expiration check: 0:01:31 ago on Wed Oct 2 05:48:47 2024.
Dependencies resolved.
Nothing to do.
Complete!
[ec2-user@ip-172-31-40-254 ~]$
```

```
sudo yum install httpd php
```

```
[ec2-user@ip-172-31-40-254 ~]$ sudo yum install httpd php
Last metadata expiration check: 0:01:59 ago on Wed Oct 2 05:48:47 2024.
Dependencies resolved.
=====
Package          Architecture Version       Repository   Size
=====
Installing:
httpd           x86_64      2.4.62-1.amzn2023
php8.3          x86_64      8.3.10-1.amzn2023.0.1
=====
Installing dependencies:
apr              x86_64      1.7.2-2.amzn2023.0.2
apr-util         x86_64      1.6.3-1.amzn2023.0.1
generic-logos-httdp x86_64      noarch      18.0.0-12.amzn2023.0.3
httpd-core       x86_64      2.4.62-1.amzn2023
httpd-filesystem x86_64      noarch      2.4.62-1.amzn2023
httpd-tools      x86_64      2.4.62-1.amzn2023
libbrotli        x86_64      1.0.9-4.amzn2023
libsodium         x86_64      1.0.19-4.amzn2023
libssl            x86_64      1.1.34-5.amzn2023.0.2
mailcap          noarch      2.1.49-3.amzn2023.0.3
nginx-filesystem x86_64      noarch      1:1.24.0-1.amzn2023.0.4
php8.3-cli       x86_64      8.3.10-1.amzn2023.0.1
php8.3-common    x86_64      8.3.10-1.amzn2023.0.1
php8.3-process   x86_64      8.3.10-1.amzn2023.0.1
=====
Repository      Size
amazonlinux      48 K
amazonlinux      10 K
amazonlinux      129 K
amazonlinux      98 K
amazonlinux      19 K
amazonlinux      1.4 M
amazonlinux      14 K
amazonlinux      81 K
amazonlinux      315 K
amazonlinux      176 K
amazonlinux      241 K
amazonlinux      33 K
amazonlinux      9.8 K
amazonlinux      3.7 M
amazonlinux      737 K
amazonlinux      45 K
```

```
sudo yum install gcc glibc glibc-common
```

```
[ec2-user@ip-172-31-40-254 ~]$ sudo yum install gcc glibc glibc-common
Last metadata expiration check: 0:02:41 ago on Wed Oct 2 05:48:47 2024.
Package glibc-2.34-52.amzn2023.0.11.x86_64 is already installed.
Package glibc-common-2.34-52.amzn2023.0.11.x86_64 is already installed.
Dependencies resolved.
=====
Package          Architecture Version       Repository   Size
=====
Installing:
gcc             x86_64      11.4.1-2.amzn2023.0.2
=====
Installing dependencies:
annobin-docs     noarch      10.93-1.amzn2023.0.1
annobin-plugin-gcc x86_64      10.93-1.amzn2023.0.1
cpp             x86_64      11.4.1-2.amzn2023.0.2
gc              x86_64      8.0.4-5.amzn2023.0.2
glibc-devel      x86_64      2.34-52.amzn2023.0.11
glibc-headers-x86 noarch      2.34-52.amzn2023.0.11
guile22         x86_64      2.2.7-2.amzn2023.0.3
kernel-headers   x86_64      6.1.109-118.189.amzn2023
libmpc          x86_64      1.2.1-2.amzn2023.0.2
libtool-ltdl    x86_64      2.4.7-1.amzn2023.0.3
libcrypt-devel   x86_64      4.4.33-7.amzn2023
make            x86_64      1:4.3-5.amzn2023.0.2
=====
Repository      Size
amazonlinux      32 M
amazonlinux      92 K
amazonlinux      887 K
amazonlinux      10 N
amazonlinux      105 K
amazonlinux      27 K
amazonlinux      427 K
amazonlinux      6.4 M
amazonlinux      1.4 M
amazonlinux      62 K
amazonlinux      38 K
amazonlinux      32 K
amazonlinux      534 K
```

```
sudo yum install gd gd-devel
```

Package	Architecture	Version	Repository	Size
Installing:				
gd	x86_64	2.3.3-5.amzn2023.0.3	amazonlinux	139
gd-devel	x86_64	2.3.3-5.amzn2023.0.3	amazonlinux	38
Installing dependencies:				
proto	x86_64	1.0.9-4.amzn2023.0.2	amazonlinux	314
brotli-devel	x86_64	1.0.9-4.amzn2023.0.2	amazonlinux	31
bzip2-devel	x86_64	1.0.8-6.amzn2023.0.2	amazonlinux	214
cairo	x86_64	1.17.6-2.amzn2023.0.1	amazonlinux	684
cmake-fs	x86_64	3.22.2-1.amzn2023.0.4	amazonlinux	16
fontconfig	x86_64	2.13.94-2.amzn2023.0.2	amazonlinux	273
fontconfig-devel	x86_64	2.13.94-2.amzn2023.0.2	amazonlinux	128
fonts-fs	noarch	1:2.0.5-12.amzn2023.0.2	amazonlinux	9.5
freetype	x86_64	2.13.2-5.amzn2023.0.1	amazonlinux	423
freetype-devel	x86_64	2.13.2-5.amzn2023.0.1	amazonlinux	912
glib2-devel	x86_64	2.74.7-68.amzn2023.0.2	amazonlinux	486
google-noto-fonts-common	noarch	20201206-2.amzn2023.0.2	amazonlinux	15
google-noto-sans-vf-fonts	noarch	20201206-2.amzn2023.0.2	amazonlinux	492
graphite2	x86_64	1.3.14-7.amzn2023.0.2	amazonlinux	97
graphite2-devel	x86_64	1.3.14-7.amzn2023.0.2	amazonlinux	21
harfbuzz	x86_64	7.0.0-2.amzn2023.0.1	amazonlinux	868
harfbuzz-devel	x86_64	7.0.0-2.amzn2023.0.1	amazonlinux	404
harfbuzz-icu	x86_64	7.0.0-2.amzn2023.0.1	amazonlinux	18

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
```

```
[ec2-user@ip-172-31-40-254 ~]$ sudo adduser -m nagios
[ec2-user@ip-172-31-40-254 ~]$ sudo passwd nagios
Changing password for user nagios.
New password:
BAD PASSWORD: The password is a palindrome
Retype new password:
Sorry, passwords do not match.
New password:
Retype new password:
passwd: all authentication tokens updated successfully.
[ec2-user@ip-172-31-40-254 ~]$
```

6. Create a new user group

```
sudo groupadd nagcmd
```

```
[ec2-user@ip-172-31-40-254 ~]$ sudo groupadd nagcmd
[ec2-user@ip-172-31-40-254 ~]$
```

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-172-31-40-254 ~]$ sudo groupadd nagcmd
[ec2-user@ip-172-31-40-254 ~]$ sudo usermod -a -G nagcmd nagios
sudo usermod -a -G nagcmd apache
[ec2-user@ip-172-31-40-254 ~]$
```

8. Create a new directory for Nagios downloads

```
mkdir ~/downloads
```

```
cd ~/downloads
```

```
[ec2-user@ip-172-31-40-254 ~]$ mkdir ~/downloads
cd ~/downloads
[ec2-user@ip-172-31-40-254 downloads]$
```

9. Use wget to download the source zip files.

Wget https://assets.nagios.com/downloads/nagioscore/releases/nagios-4.4.6.tar.gz wget

```
[ec2-user@ip-172-31-40-254 downloads]$ Wget https://assets.nagios.com/downloads/nagioscore/releases/nagios-4.4.6.tar.gz
--2024-10-02 06:15:45-- https://nagios-plugins.org/download/nagios-plugins-2.3.3.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|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 2782610 (2.7M) [application/x-gzip]
Saving to: 'nagios-plugins-2.3.3.tar.gz'

nagios-plugins-2.3.3.tar.gz      0%[=====] 632.00K 3.02MB/s
nagios-plugins-2.3.3.tar.gz    23%[=====] 1.26M 8.10MB/s   in 0.3s
nagios-plugins-2.3.3.tar.gz    100%[=====] 2.65M 8.10MB/s   in 0.3s

2024-10-02 06:15:46 (8.10 MB/s) - 'nagios-plugins-2.3.3.tar.gz' saved [2782610/2782610]
[ec2-user@ip-172-31-40-254 downloads]$
```

<https://nagios-plugins.org/download/nagios-plugins-2.3.3.tar.gz>

```
[ec2-user@ip-172-31-40-254 downloads]$ wget https://assets.nagios.com/downloads/nagioscore/releases/nagios-4.4.6.tar.gz
--2024-10-02 06:17:24-- https://assets.nagios.com/downloads/nagioscore/releases/nagios-4.4.6.tar.gz
Resolving assets.nagios.com (assets.nagios.com)... 45.79.49.120, 2600:3c00::f03c:92ff:fe17:45ce
Connecting to assets.nagios.com (assets.nagios.com)|45.79.49.120|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 11333414 (11M) [application/x-gzip]
Saving to: 'nagios-4.4.6.tar.gz'

nagios-4.4.6.tar.gz      0%[=====] 495.62K 2.40MB/s
nagios-4.4.6.tar.gz    4%[==>] 3.26M 7.99MB/s
nagios-4.4.6.tar.gz   30%[=====] 6.91M 11.0MB/s
nagios-4.4.6.tar.gz   63%[=====] 10.46M 12.6MB/s
nagios-4.4.6.tar.gz  96%[=====] 10.81M 12.9MB/s   in 0.8s
nagios-4.4.6.tar.gz  100%[=====] 10.81M 12.9MB/s   in 0.8s

2024-10-02 06:17:25 (12.9 MB/s) - 'nagios-4.4.6.tar.gz' saved [11333414/11333414]
[ec2-user@ip-172-31-40-254 downloads]$
```

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

tar zxvf nagios-4.4.6.tar.gz
cd nagios-4.4.6

```
[ec2-user@ip-172-31-40-254 downloads]$ tar zxvf nagios-4.4.6.tar.gz
nagios-4.4.6/
nagios-4.4.6/.gitignore
nagios-4.4.6/.travis.yml
nagios-4.4.6/CONTRIBUTING.md
nagios-4.4.6/Changelog
nagios-4.4.6/INSTALLING
nagios-4.4.6/LEGAL
nagios-4.4.6/LICENSE
nagios-4.4.6/Makefile.in
nagios-4.4.6/README.md
nagios-4.4.6/THANKS
nagios-4.4.6/UPGRADING
nagios-4.4.6/aclocal.m4
nagios-4.4.6/autoconf-macros/
nagios-4.4.6/autoconf-macros/.gitignore
nagios-4.4.6/autoconf-macros/CHANGELOG.md
nagios-4.4.6/autoconf-macros/LICENSE
nagios-4.4.6/autoconf-macros/LICENSE.md
nagios-4.4.6/autoconf-macros/README.md
nagios-4.4.6/autoconf-macros/add_group_user
nagios-4.4.6/autoconf-macros/ax_nagios_get_distrib
nagios-4.4.6/autoconf-macros/ax_nagios_get_files
nagios-4.4.6/autoconf-macros/ax_nagios_get_inetd
nagios-4.4.6/autoconf-macros/ax_nagios_get_init
nagios-4.4.6/autoconf-macros/ax_nagios_get_os
nagios-4.4.6/autoconf-macros/ax_nagios_get_paths
nagios-4.4.6/autoconf-macros/ax_nagios_get_ssl
nagios-4.4.6/base/
nagios-4.4.6/base/.gitignore
nagios-4.4.6/base/Makefile.in
nagios-4.4.6/base/broker.c
```

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

/configure --with-command-group=nagcmd

```
[ec2-user@ip-172-31-40-254 nagios-4.4.6]$ ./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 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 make sets $MAKE... yes
checking whether ln -s works... yes
checking for strip... /usr/bin-strip
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 ANSI C header files... yes
checking whether time.h and sys/time.h may both be included... yes
checking for sys/wait.h that is POSIX.1 compatible... yes
checking for sys/types.h... yes
checking for sys/stat.h... yes
checking for stdlib.h... yes
checking for string.h... yes
checking for memory.h... yes
checking for strings.h... yes
checking for inttypes.h... yes
checking for stdint.h... yes
checking for unistd.h... yes
checking arpa/inet.h usability... yes
checking arpa/inet.h presence... yes
checking for arpa/inet.h... yes
```

12. Compile the source code.

make all

```
[ec2-user@ip-172-31-40-254 nagios-4.4.6]$ make all
cd ./base && make
make[1]: Entering directory '/home/ec2-user/downloads/nagios-4.4.6/base'
gcc -Wall -I... -g -O2 -DHAVE_CONFIG_H -DSNCore -c -o nagios.o nagios.c
gcc -Wall -I... -g -O2 -DHAVE_CONFIG_H -DSNCore -c -o broker.o broker.c
gcc -Wall -I... -g -O2 -DHAVE_CONFIG_H -DSNCore -c -o nebmods.o nebmods.c
gcc -Wall -I... -g -O2 -DHAVE_CONFIG_H -DSNCore -c -o .../common/shared.o .../common/shared.c
gcc -Wall -I... -g -O2 -DHAVE_CONFIG_H -DSNCore -c -o query-handler.o query-handler.c
gcc -Wall -I... -g -O2 -DHAVE_CONFIG_H -DSNCore -c -o workers.o workers.c
In function 'get_wproc_list',
  inlined from 'get_worker' at workers.c:277:12:
workers.c:253:17: warning: '%s' directive argument is null [-Wformat-overflow=]
  253 |         log_debug_info(DEBUGL_CHECKS, 1, "Found specialized worker(s) for '%s'", (slash && *slash != '/') ? slash : cmd_name);
   |         ^
  |
  |         ~~~~~
gcc -Wall -I... -g -O2 -DHAVE_CONFIG_H -DSNCore -c -o checks.o checks.c
gcc -Wall -I... -g -O2 -DHAVE_CONFIG_H -DSNCore -c -o config.o config.c
gcc -Wall -I... -g -O2 -DHAVE_CONFIG_H -DSNCore -c -o commands.o commands.c
gcc -Wall -I... -g -O2 -DHAVE_CONFIG_H -DSNCore -c -o events.o events.c
gcc -Wall -I... -g -O2 -DHAVE_CONFIG_H -DSNCore -c -o flapping.o flapping.c
gcc -Wall -I... -g -O2 -DHAVE_CONFIG_H -DSNCore -c -o logging.o logging.c
gcc -Wall -I... -g -O2 -DHAVE_CONFIG_H -DSNCore -c -o macros-base.o .../common/macros.c
gcc -Wall -I... -g -O2 -DHAVE_CONFIG_H -DSNCore -c -o netutils.o netutils.c
netutils.c: in function 'my_tcp_connect':
netutils.c:50:47: warning: '%d' directive output may be truncated writing between 1 and 11 bytes into a region of size 6 [-Wformat-truncation=]
  50 |     sprintf(port_str, sizeof(port_str), "%d", port);
   |     ^
  |
  |     ~~~
netutils.c:50:46: note: directive argument in the range [-2147483648, 65535]
  50 |     sprintf(port_str, sizeof(port_str), "%d", port);
   |     ^
  |
  |     ~~~
netutils.c:50:9: note: 'sprintf' output between 2 and 12 bytes into a destination of size 6
  50 |     sprintf(port_str, sizeof(port_str), "%d", port);
   |     ^
  |
  |     ~~~~~
gcc -Wall -I... -g -O2 -DHAVE_CONFIG_H -DSNCore -c -o notifications.o notifications.c
```

```
make install-classicui
- This installs the classic theme for the Nagios
  web interface

*** 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

```
[ec2-user@ip-172-31-40-254 nagios-4.4.6]$ ./sudo make install
sudo make install-init
sudo make install-config
sudo make install-commandmode
-bash: ./sudo: No such file or directory
/usr/bin/install -c -m 755 -o root -g root /lib/systemd/system
/usr/bin/install -c -m 775 -o nagios -g nagios -d /usr/local/nagios/etc
/usr/bin/install -c -m 775 -o nagios -g nagios -d /usr/local/nagios/etc/objects
/usr/bin/install -c -b -m 664 -o nagios -g nagios sample-config/nagios.cfg /usr/local/nagios/etc/nagios.cfg
/usr/bin/install -c -b -m 664 -o nagios -g nagios sample-config/cgi.cgi /usr/local/nagios/etc/cgi.cgi
/usr/bin/install -c -b -m 664 -o nagios -g nagios sample-config/resource.cfg /usr/local/nagios/etc/resource.cfg
/usr/bin/install -c -b -m 664 -o nagios -g nagios sample-config/template-object/templates.cfg
/usr/bin/install -c -b -m 664 -o nagios sample-config/template-object/commands.cfg /usr/local/nagios/etc/objects/commands.cfg
/usr/bin/install -c -b -m 664 -o nagios sample-config/template-object/contacts.cfg /usr/local/nagios/etc/objects/contacts.cfg
/usr/bin/install -c -b -m 664 -o nagios sample-config/template-object/timeperiods.cfg /usr/local/nagios/etc/objects/timeperiods.cfg
/usr/bin/install -c -b -m 664 -o nagios sample-config/template-object/localhost.cfg /usr/local/nagios/etc/objects/localhost.cfg
/usr/bin/install -c -b -m 664 -o nagios sample-config/template-object/windows.cfg /usr/local/nagios/etc/objects/windows.cfg
/usr/bin/install -c -b -m 664 -o nagios sample-config/template-object/printer.cfg /usr/local/nagios/etc/objects/printer.cfg
/usr/bin/install -c -b -m 664 -o nagios sample-config/template-object/switch.cfg /usr/local/nagios/etc/objects/switch.cfg

*** 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 ***

```

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

sudo nano /usr/local/nagios/etc/objects/contacts.cfg

```
GNU nano 5.8                               /usr/local/nagios/etc/objects/contacts.cfg
# just one contact defined by default - the Nagios admin (that's you)
# This contact definition inherits a lot of default values from the
# 'generic-contact' template which is defined elsewhere.

define contact {
    contact_name      nagiosadmin          ; Short name of user
    use               generic-contact       ; Inherit default values from generic-contact template (defined above)
    alias             Nagios Admin         ; Full name of user
    email            bhagyeshpatil0702@gmail.com; <<***** CHANGE THIS TO YOUR EMAIL ADDRESS *****

#####
# CONTACT GROUPS
#####

# We only have one contact in this simple configuration file, so there is
# no need to create more than one contact group.

define contactgroup {

```

15. Configure the web interface.

sudo make install-webconf

```
[ec2-user@ip-172-31-40-254 nagios-4.4.6]$ 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 ***

[ec2-user@ip-172-31-40-254 nagios-4.4.6]$
```

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-172-31-40-254 nagios-4.4.6]$ sudo htpasswd -c /usr/local/nagios/etc/htpasswd.users nagiosadmin
New password:
Re-type new password:
Adding password for user nagiosadmin
[ec2-user@ip-172-31-40-254 nagios-4.4.6]$
```

17. Restart Apache

```
sudo systemctl restart httpd
```

```
[ec2-user@ip-172-31-40-254 nagios-4.4.6]$ sudo systemctl restart httpd
[ec2-user@ip-172-31-40-254 nagios-4.4.6]$
```

```
cd ~/downloads
```

```
tar zxvf nagios-plugins-2.3.3.tar.gz
```

```
cd nagios-plugins-2.3.3
```

```
[ec2-user@ip-172-31-40-254 nagios-4.4.6]$ sudo systemctl restart httpd
[ec2-user@ip-172-31-40-254 nagios-4.4.6]$ cd ~/downloads
tar zxvf nagios-plugins-2.3.3.tar.gz
cd nagios-plugins-2.3.3
nagios-plugins-2.3.3/
nagios-plugins-2.3.3/perlmods/
nagios-plugins-2.3.3/perlmods/Config-Tiny-2.14.tar.gz
nagios-plugins-2.3.3/perlmods/parent-0.226.tar.gz
nagios-plugins-2.3.3/perlmods/test-Simple-0.98.tar.gz
nagios-plugins-2.3.3/perlmods/Makefile.in
nagios-plugins-2.3.3/perlmods/version-0.9903.tar.gz
nagios-plugins-2.3.3/perlmods/Makefile.am
nagios-plugins-2.3.3/perlmods/Module-Runtime-0.013.tar.gz
nagios-plugins-2.3.3/perlmods/Module-Metadata-1.000014.tar.gz
nagios-plugins-2.3.3/perlmods/Params-Validate-0.000008.tar.gz
nagios-plugins-2.3.3/perlmods/Config-Parser-0.31.tar.gz
nagios-plugins-2.3.3/perlmods/Tiny-Tiny-0.18.tar.gz
nagios-plugins-2.3.3/perlmods/Module-Implementation-0.07.tar.gz
nagios-plugins-2.3.3/perlmods/Makefile
nagios-plugins-2.3.3/perlmods/Perl-OStype-1.003.tar.gz
nagios-plugins-2.3.3/perlmods/install_order
nagios-plugins-2.3.3/perlmods/Nagios-Plugin-0.36.tar.gz
nagios-plugins-2.3.3/perlmods/Math-Calc-Units-1.07.tar.gz
nagios-plugins-2.3.3/perlmods/Module-Build-0.4007.tar.gz
nagios-plugins-2.3.3/ABOUT-NLS
nagios-plugins-2.3.3/configure.ac
nagios-plugins-2.3.3/Makefile.in
nagios-plugins-2.3.3/config.h.in
nagios-plugins-2.3.3/ChangeLog
nagios-plugins-2.3.3/AUTHORS
nagios-plugins-2.3.3/lib/
nagios-plugins-2.3.3/lib/parse_ini.h
nagios-plugins-2.3.3/lib/extr_opts.c
nagios-plugins-2.3.3/lib/Makefile.in
```

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

```
./configure --with-nagios-user=nagios --with-nagios-group=nagios
```

```
make sudo
```

```
make install
```

```
[ec2-user@ip-172-31-80-22 nagios-plugins-2.3.3]$ ./configure --with-nagios-user=nagios --with-nagios-group=nagios
make
sudo make install
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 to disable maintainer-specific portions of Makefiles... yes
checking build system type... x86_64-unknown-linux-gnu
checking host system type... x86_64-unknown-linux-gnu
checking for gcc... gcc
checking for C compiler default output file name... a.out
checking whether the C compiler works... yes
checking whether we are cross compiling... no
checking for suffix of executables...
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 for style of include used by make... GNU
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
```

19. Compile and install plugins

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

```
sudo chkconfig nagios on
```

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

```
sudo systemctl start nagios
```

```
[ec2-user@ip-172-31-80-22 nagios-plugins-2.3.3]$ sudo chkconfig --add nagios
sudo chkconfig nagios on
sudo /usr/local/nagios/bin/nagios -v /usr/local/nagios/etc/nagios.cfg
sudo systemctl start nagios
error reading information on service nagios: No such file or directory
Note: Forwarding request to 'systemctl enable nagios.service'.
Created symlink /etc/systemd/system/multi-user.target.wants/nagios.service → /usr/lib/systemd/system/nagios.service.

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

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

Running pre-flight check on configuration data...

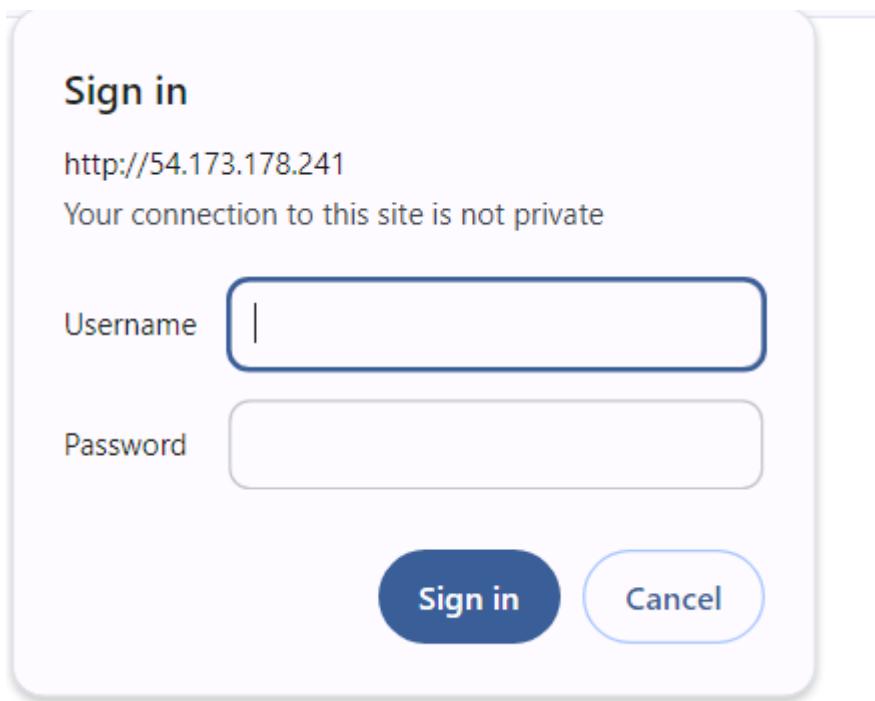
Checking objects...
  Checked 8 services.
  Checked 1 hosts.
  Checked 1 host groups.
```

20. Check the status of Nagios

```
things look okay - No serious problems were detected during the pre-flight check
[ec2-user@ip-172-31-45-178 nagios-plugins-2.3.3]$ sudo systemctl status nagios
nagios.service - Nagios Core 4.4.6
   Loaded: loaded (/usr/lib/systemd/system/nagios.service; enabled; preset: disabled)
     Active: active (running) since Wed 2024-10-02 05:37:36 UTC; 14s ago
       Docs: https://www.nagios.org/documentation
      Process: 67990 ExecStartPre=/usr/local/nagios/bin/nagios -v /usr/local/nagios/etc/nagios.cfg (code=exited, status=0/SUCCESS)
      Process: 67991 ExecStart=/usr/local/nagios/bin/nagios -d /usr/local/nagios/etc/nagios.cfg (code=exited, status=0/SUCCESS)
    Main PID: 67992 (nagios)
      Tasks: 6 (limit: 1112)
        Memory: 2.0M
          CPU: 16ms
        CGroup: /system.slice/nagios.service
                └─67992 /usr/local/nagios/bin/nagios -d /usr/local/nagios/etc/nagios.cfg
                  ├─67993 /usr/local/nagios/bin/nagios --worker /usr/local/nagios/var/rw/nagios.qh
                  ├─67994 /usr/local/nagios/bin/nagios --worker /usr/local/nagios/var/rw/nagios.qh
                  ├─67995 /usr/local/nagios/bin/nagios --worker /usr/local/nagios/var/rw/nagios.qh
                  ├─67996 /usr/local/nagios/bin/nagios --worker /usr/local/nagios/var/rw/nagios.qh
                  └─67997 /usr/local/nagios/bin/nagios -d /usr/local/nagios/etc/nagios.cfg

Oct 02 05:37:36 ip-172-31-45-178.ec2.internal nagios[67992]: qh: Socket '/usr/local/nagios/var/rw/nagios.qh' successfully initialized
Oct 02 05:37:36 ip-172-31-45-178.ec2.internal nagios[67992]: qh: core query handler registered
```

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



Not secure 54.173.178.241/nagios/

Nagios® Core™

Daemon running with PID 67992

Nagios® Core™ Version 4.4.6 April 28, 2020 Check for updates

A new version of Nagios Core is available! Visit nagios.org to download Nagios 4.5.5.

Get Started

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

Quick Links

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

Latest News

Don't Miss...

Copyright © 2010-2020 Nagios Core Development Team and Community Contributors. Copyright © 1999-2009 Ethan Galstad. See the THANKS file for more information on contributors.

Page Tour

General
Home
Documentation
Current Status
 General Overview
 Map (Legacy)
 Hosts
 Services
 Host Groups
 Summary
 Grid
 Service Groups
 Summary
 Grid
Problems
 Services (Unhandled)
 Hosts (Unhandled)
 Network Outages
Quick Search:

Reports
 Availability
 Trends (Legacy)
 Alerts
 History
 Summary
 Histogram (Legacy)
 Notifications
 Event Log

System
 Comments
 Downtime
 Process Info
 Performance Info
 Scheduling Queue
 Configuration

Name : Swaraj Patil

Class : D15A

Roll No. : 35

EXPERIMENT 10

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

Procedure:-

Check if the nagios service is running by executing following command

sudo systemctl status nagios

```
ubuntu@ip-172-31-89-161:~$ sudo systemctl status nagios
● nagios.service - Nagios Core 4.4.6
   Loaded: loaded (/usr/lib/systemd/system/nagios.service; enabled; preset: enabled)
   Active: active (running) since Sat 2024-09-28 16:08:58 UTC; 1min 2s ago
     Docs: https://www.nagios.org/documentation
 Process: 15743 ExecStartPre=/usr/local/nagios/bin/nagios -v /usr/local/nagios/etc/nagios.cfg (code=exited, status=0/SUCCESS)
 Process: 15753 ExecStart=/usr/local/nagios/bin/nagios -d /usr/local/nagios/etc/nagios.cfg (code=exited, status=0/SUCCESS)
 Main PID: 15764 (nagios)
   Tasks: 6 (limit: 1130)
  Memory: 2.4M (peak: 3.2M)
    CPU: 29ms
   CGroub: /system.slice/nagios.service
           ├─15764 /usr/local/nagios/bin/nagios -d /usr/local/nagios/etc/nagios.cfg
           ├─15765 /usr/local/nagios/bin/nagios --worker /usr/local/nagios/var/rw/nagios.qh
           ├─15766 /usr/local/nagios/bin/nagios --worker /usr/local/nagios/var/rw/nagios.qh
           ├─15767 /usr/local/nagios/bin/nagios --worker /usr/local/nagios/var/rw/nagios.qh
           ├─15768 /usr/local/nagios/bin/nagios --worker /usr/local/nagios/var/rw/nagios.qh
           └─15769 /usr/local/nagios/bin/nagios -d /usr/local/nagios/etc/nagios.cfg

Sep 28 16:08:58 ip-172-31-89-161 nagios[15764]: qh: Socket '/usr/local/nagios/var/rw/nagios.qh' successfully initialized
Sep 28 16:08:58 ip-172-31-89-161 nagios[15764]: qh: core query handler registered
Sep 28 16:08:58 ip-172-31-89-161 nagios[15764]: qh: echo service query handler registered
Sep 28 16:08:58 ip-172-31-89-161 nagios[15764]: qh: help for the query handler registered
Sep 28 16:08:58 ip-172-31-89-161 nagios[15764]: wproc: Successfully registered manager as @wproc with query handler
Sep 28 16:08:58 ip-172-31-89-161 nagios[15764]: wproc: Registry request: name=Core Worker 15765;pid=15765
Sep 28 16:08:58 ip-172-31-89-161 nagios[15764]: wproc: Registry request: name=Core Worker 15766;pid=15766
Sep 28 16:08:58 ip-172-31-89-161 nagios[15764]: wproc: Registry request: name=Core Worker 15767;pid=15767
```

Now, create a new EC2 instance on AWS

Instances (2) Info		Last updated less than a minute ago	C	Connect	Instance state ▾	Actions ▾	Launch instances	▼
					All states ▾			◀ 1 ▶ ⚙
<input type="checkbox"/>	Name ▾	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	P
<input type="checkbox"/>	nagios-host	i-09e8ea019f24f4be2	Running Q Q	t2.micro	2/2 checks passed View alarms +	us-east-1c	e	
<input type="checkbox"/>	linux-client	i-0ad38836f030e3784	Running Q Q	t2.micro	Initializing View alarms +	us-east-1c	e	

Now perform the following commands on nagios-host EC2 instance. On the server, run this command

ps -ef | grep nagios

```
ubuntu@ip-172-31-89-161:~$ ps -ef | grep nagios
nagios 15764 1 0 16:08 ? 00:00:00 /usr/local/nagios/bin/nagios -d /usr/local/nagios/etc/nagios.cfg
nagios 15765 15764 0 16:08 ? 00:00:00 /usr/local/nagios/bin/nagios --worker /usr/local/nagios/var/rw/nagios.qh
nagios 15766 15764 0 16:08 ? 00:00:00 /usr/local/nagios/bin/nagios --worker /usr/local/nagios/var/rw/nagios.qh
nagios 15767 15764 0 16:08 ? 00:00:00 /usr/local/nagios/bin/nagios --worker /usr/local/nagios/var/rw/nagios.qh
nagios 15768 15764 0 16:08 ? 00:00:00 /usr/local/nagios/bin/nagios --worker /usr/local/nagios/var/rw/nagios.qh
nagios 15769 15764 0 16:08 ? 00:00:00 /usr/local/nagios/bin/nagios -d /usr/local/nagios/etc/nagios.cfg
ubuntu 15957 1342 0 16:13 pts/0 00:00:00 grep --color=auto nagios
ubuntu@ip-172-31-89-161:~$
```

Sudo su

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

```
ubuntu@ip-172-31-89-161:~$ sudo su
mkdir /usr/local/nagios/etc/objects/monitorhosts
mkdir /usr/local/nagios/etc/objects/monitorhosts/linuxhosts
root@ip-172-31-89-161:/home/ubuntu#
```

Copy localhost.cfg file to the mentioned location

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

```
root@ip-172-31-89-161:/usr/local/nagios/etc/objects# cp /usr/local/nagios/etc/objects/localhost.cfg /usr/local/nagios/etc/objects/monitorhosts/linuxhosts
cp: cannot create regular file '/usr/local/nagios/etc/objects/monitorhosts/linuxhosts': No such file or directory
root@ip-172-31-89-161:/usr/local/nagios/etc/objects# sudo mkdir -p /usr/local/nagios/etc/objects/monitorhosts/linuxhosts
root@ip-172-31-89-161:/usr/local/nagios/etc/objects# cp /usr/local/nagios/etc/objects/localhost.cfg /usr/local/nagios/etc/objects/monitorhosts/linuxhosts
root@ip-172-31-89-161:/usr/local/nagios/etc/objects#
```

Open the nano editor for localhost.cfg file and make these changes. Add the Ip address of the linux-client for the address field.

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

```

GNU nano 7.2                               /usr/local/nagios/etc/nagios.cfg
#####
# HOST DEFINITION
#
#####

# Define a host for the local machine

define host {

    use          linux-server      ; Name of host template
    ; This host definition
    ; in (or inherits from) a group
    ; will inherit its host_name
    ; and host alias

    host_name    linuxserver
    alias        linuxserver
    address     52.207.253.18
}

#####

# HOST GROUP DEFINITION

^G Help          ^O Write Out      ^W Where Is      ^K Cut           ^T Ex
^X Exit         ^R Read File      ^\ Replace       ^U Paste         ^J Ju

```

Note - Here replace hostname with linuxserver

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

Add the following line to the nagios.cfg file

cfg_dir=/usr/local/nagios/etc/objects/monitorhosts/

```

# Definitions for monitoring a router/switch
#cfg_file=/usr/local/nagios/etc/objects/switch.cfg

# Definitions for monitoring a network printer
#cfg_file=/usr/local/nagios/etc/objects/printer.cfg

# You can also tell Nagios to process all config files (with a .cfg
# extension) in a particular directory by using the cfg_dir
# directive as shown below:

#cfg_dir=/usr/local/nagios/etc/servers
#cfg_dir=/usr/local/nagios/etc/printers
#cfg_dir=/usr/local/nagios/etc/switches
#cfg_dir=/usr/local/nagios/etc/routers

cfg_dir=/usr/local/nagios/etc/objects/monitorhosts/

```

After making the changes in nagios.cfg file now check validate the file by typing the following command in the terminal.

```
/usr/local/nagios/bin/nagios -v /usr/local/nagios/etc/nagios.cfg
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 16 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-172-31-89-161:/usr/local/nagios/etc/objects/monitorhosts/linuxhosts#
```

Now restart the service by using this command

```
service nagios restart
```

```

root@ip-172-31-89-161:/usr/local/nagios/etc/objects/monitorhosts/linuxhosts# service nagios restart
root@ip-172-31-89-161:/usr/local/nagios/etc/objects/monitorhosts/linuxhosts# systemctl status nagios
● nagios.service - Nagios Core 4.4.6
   Loaded: loaded (/usr/lib/systemd/system/nagios.service; enabled; preset: enabled)
   Active: active (running) since Sat 2024-09-28 17:36:35 UTC; 19s ago
     Docs: https://www.nagios.org/documentation
 Process: 1870 ExecStartPre=/usr/local/nagios/bin/nagios -v /usr/local/nagios/etc/nagios.cfg (code=exited, status=0/SUCCESS)
 Process: 1872 ExecStart=/usr/local/nagios/bin/nagios -d /usr/local/nagios/etc/nagios.cfg (code=exited, status=0/SUCCESS)
 Main PID: 1874 (nagios)
   Tasks: 8 (limit: 1130)
  Memory: 3.0M (peak: 3.2M)
    CPU: 24ms
   CGroup: /system.slice/nagios.service
           ├─1874 /usr/local/nagios/bin/nagios -d /usr/local/nagios/etc/nagios.cfg
           ├─1875 /usr/local/nagios/bin/nagios --worker /usr/local/nagios/var/rw/nagios.qh
           ├─1876 /usr/local/nagios/bin/nagios --worker /usr/local/nagios/var/rw/nagios.qh
           ├─1877 /usr/local/nagios/bin/nagios --worker /usr/local/nagios/var/rw/nagios.qh
           ├─1878 /usr/local/nagios/bin/nagios --worker /usr/local/nagios/var/rw/nagios.qh
           ├─1879 /usr/local/nagios/bin/nagios --worker /usr/local/nagios/etc/nagios.cfg
           ├─1880 /usr/local/nagios/libexec/check_ping -H 52.207.253.18 -w 3000.0,80% -c 5000.0,100% -p 5
           └─1881 /usr/bin/ping -n -U -w 30 -c 5 52.207.253.18

Sep 28 17:36:35 ip-172-31-89-161 nagios[1874]: qh: Socket '/usr/local/nagios/var/rw/nagios.qh' successfully initialized
Sep 28 17:36:35 ip-172-31-89-161 nagios[1874]: qh: core query handler registered
Sep 28 17:36:35 ip-172-31-89-161 nagios[1874]: qh: echo service query handler registered
Sep 28 17:36:35 ip-172-31-89-161 nagios[1874]: qh: help for the query handler registered
Sep 28 17:36:35 ip-172-31-89-161 nagios[1874]: wproc: Successfully registered manager as @wproc with query handler
Sep 28 17:36:35 ip-172-31-89-161 nagios[1874]: wproc: Registry request: name=Core Worker 1875,pid=1875
Lines 1-26

```

Now using this command update the apt repository of ubuntu (linux-client), install gcc, nagios-nrpe-server and nagios-plugin sudo apt update -y sudo apt install gcc -y

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

Now open nrpe.cfg file and add the ip address of the nagios host as shown. To open the nrpe.cfg file copy this command.

```

# Supported.
#
# Note: The daemon only does rudimentary checking
# address. I would highly recommend adding entries
# file to allow only the specified host to connect
# you are running this daemon on.
#
# NOTE: This option is ignored if NRPE is running
#       as a module in Apache.
allowed_hosts=127.0.0.1,54.167.169.0

#
# COMMAND ARGUMENT PROCESSING
# This option determines whether or not the NRPE
# to specify arguments to commands that are executed
# if the daemon was configured with the --enable-command
# option.

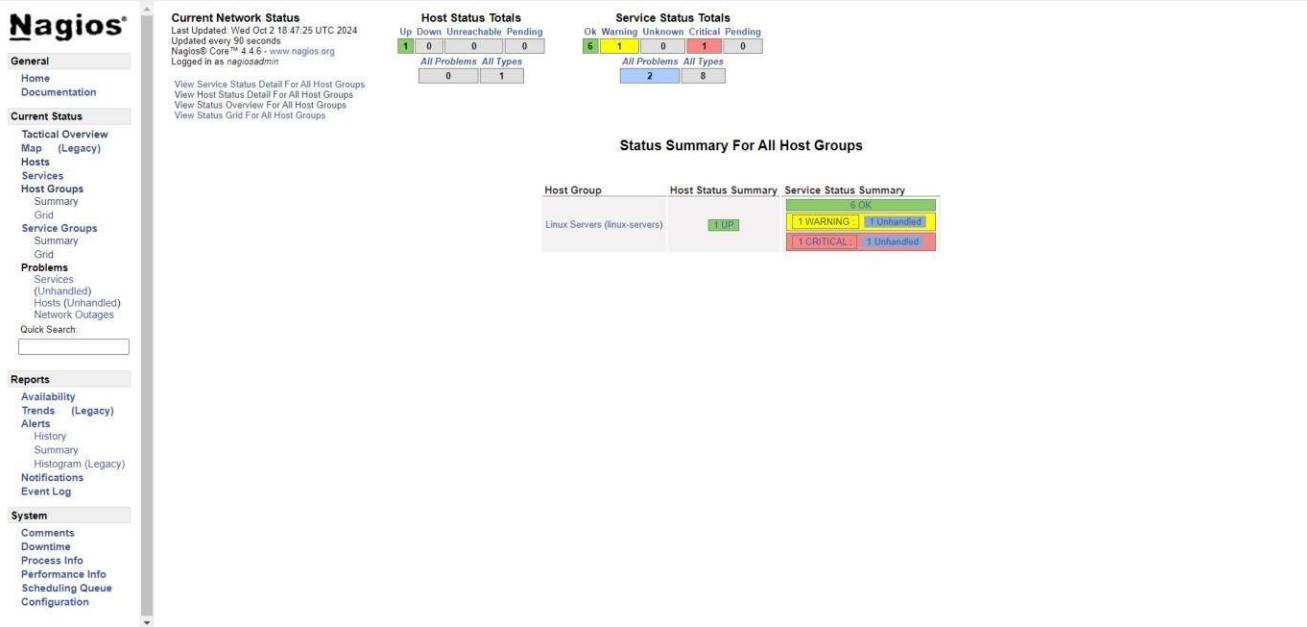
```

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

Now restart nrpe server by using this command

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

Now, check nagios dashboard, you should see linuxserver up and running, if not



Experiment 11

Swaraj Patil
D15A 40

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

AWS Lambda

AWS Lambda is a serverless computing service provided by Amazon Web Services (AWS). Users of AWS Lambda create functions, self-contained applications written in one of the supported languages and runtimes, and upload them to AWS Lambda, which executes those functions in an efficient and flexible manner. The Lambda functions can perform any kind of computing task, from serving web pages and processing streams of data to calling APIs and integrating with other AWS services.

The concept of “serverless” computing refers to not needing to maintain your own servers to run these functions. AWS Lambda is a fully managed service that takes care of all the infrastructure for you. And so “serverless” doesn’t mean that there are no servers involved: it just means that the servers, the operating systems, the network layer and the rest of the infrastructure have already been taken care of so that you can focus on writing application code.

Features of AWS Lambda

- AWS Lambda easily scales the infrastructure without any additional configuration. It reduces the operational work involved.
- It offers multiple options like AWS S3, CloudWatch, DynamoDB, API Gateway, Kinesis, CodeCommit, and many more to trigger an event.
- You don’t need to invest upfront. You pay only for the memory used by the lambda function and minimal cost on the number of requests hence cost-efficient.
- AWS Lambda is secure. It uses AWS IAM to define all the roles and security policies.
- It offers fault tolerance for both services running the code and the function. You do not have to worry about the application down.

Packaging Functions

Lambda functions need to be packaged and sent to AWS. This is usually a process of compressing the function and all its dependencies and uploading it to an S3 bucket.

And letting AWS know that you want to use this package when a specific event takes place. To help us with this process we use the Serverless Stack Framework (SST). We'll go over this in detail later on in this guide.

Execution Model

The container (and the resources used by it) that runs our function is managed completely by AWS. It is brought up when an event takes place and is turned off if it is not being used. If additional requests are made while the original event is being served, a new container is brought up to serve a request. This means that if we are undergoing a usage spike, the cloud provider simply creates multiple instances of the container without function to serve those requests.

This has some interesting implications. Firstly, our functions are effectively stateless.

Secondly, each request (or event) is served by a single instance of a Lambda function. This means that you are not going to be handling concurrent requests in your code.

AWS brings up a container whenever there is a new request. It does make some optimizations here. It will hang on to the container for a few minutes (5 - 15mins depending on the load) so it can respond to subsequent requests without a cold start.

Stateless Functions

The above execution model makes Lambda functions effectively stateless. This means that every time your Lambda function is triggered by an event it is invoked in a completely new environment. You don't have access to the execution context of the previous event.

However, due to the optimization noted above, the actual Lambda function is invoked only once per container instantiation. Recall that our functions are run inside containers. So when a function is first invoked, all the code in our handler function gets executed and the handler function gets invoked. If the container is still available for subsequent requests, your function will get invoked and not the code around it.

For example, the `createNewDbConnection` method below is called once per container instantiation and not every time the Lambda function is invoked. The `myHandler` function on the other hand is called on every invocation.

Common Use Cases for Lambda

Due to Lambda's architecture, it can deliver great benefits over traditional cloud computing setups for applications where:

1. Individual tasks run for a short time;
2. Each task is generally self-contained;

3. There is a large difference between the lowest and highest levels in the workload of the application.

Some of the most common use cases for AWS Lambda that fit these criteria are: Scalable APIs. When building APIs using AWS Lambda, one execution of a Lambda function can serve a single HTTP request. Different parts of the API can be routed to different Lambda functions via Amazon API Gateway. AWS Lambda automatically scales individual functions according to

the demand for them, so different parts of your API can scale differently according to current usage levels. This allows for cost-effective and flexible API setups.

Data processing. Lambda functions are optimized for event-based data processing. It is easy to integrate AWS Lambda with data sources like Amazon DynamoDB and trigger a Lambda function for specific kinds of data events. For example, you could employ Lambda to do some work every time an item in DynamoDB is created or updated, thus making it a good fit for things like notifications, counters and analytics.

Steps to create an AWS Lambda function

Step 1: Create a Lambda Function

1. Choose a Function Creation

Method:

Select Author from scratch.

2. Configure the Function:

Function name: Enter a name for your function (e.g., MyFirstLambda).

Runtime: Choose Python 3.x (the latest available version).

Permissions: Choose Create a new role with basic Lambda permissions (this creates a role with the necessary permissions).

3. Click on Create function.

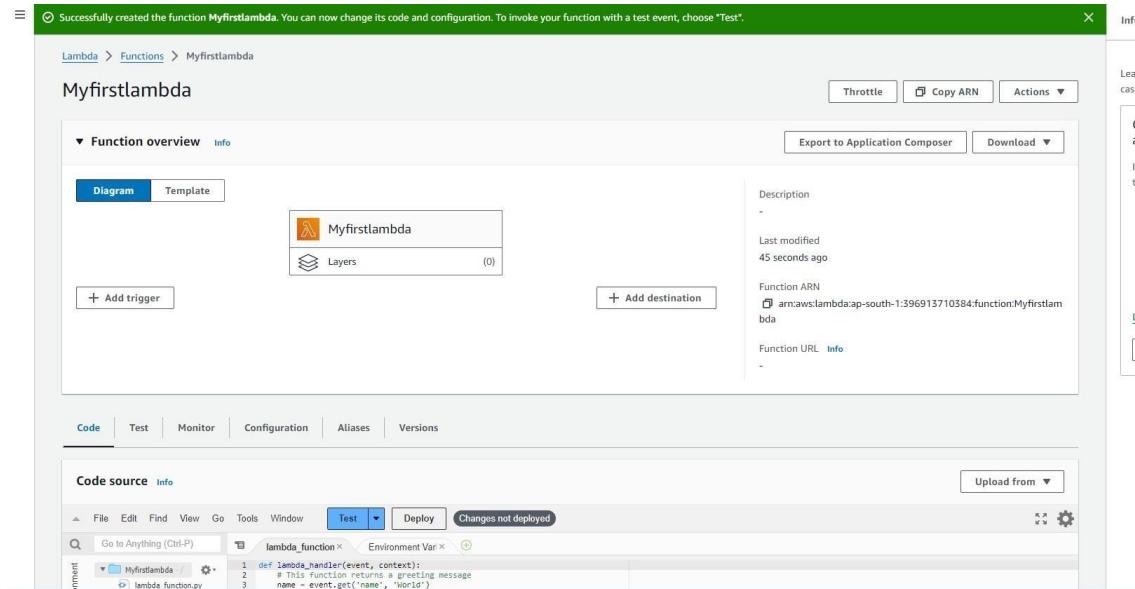
The screenshot shows the 'Create function' wizard in the AWS Lambda console. The 'Basic information' step is active. Under 'Function name', the value 'Myfirstlambda' is entered. The 'Runtime' dropdown is set to 'Python 3.12' and the 'Architecture' dropdown is set to 'x86_64'.

Step 2: Write Your Lambda Function Code

In the Function code section, you will see a code editor. Replace the default code with the following Python code:

```
python
Copy code
def lambda_handler(event, context):
    # This function returns a greeting message
    name = event.get('name', 'World')
    return {
        'statusCode': 200, 'body':
            f'Hello, {name}!'
    }
```

This function reads a name from the event and returns a greeting message. If no name is provided, it defaults to "World".



```

1 def lambda_handler(event, context):
2     # This function returns a greeting message
3     name = event.get('name', 'World')
4     return {
5         'statusCode': 200,
6         'body': f'Hello, {name}!'
7     }

```

Step3:

1. Configure a Test Event:

Click on the Test button.

In the Configure test event dialog, give your event a name (e.g., TestEvent). Replace the default JSON with the following:

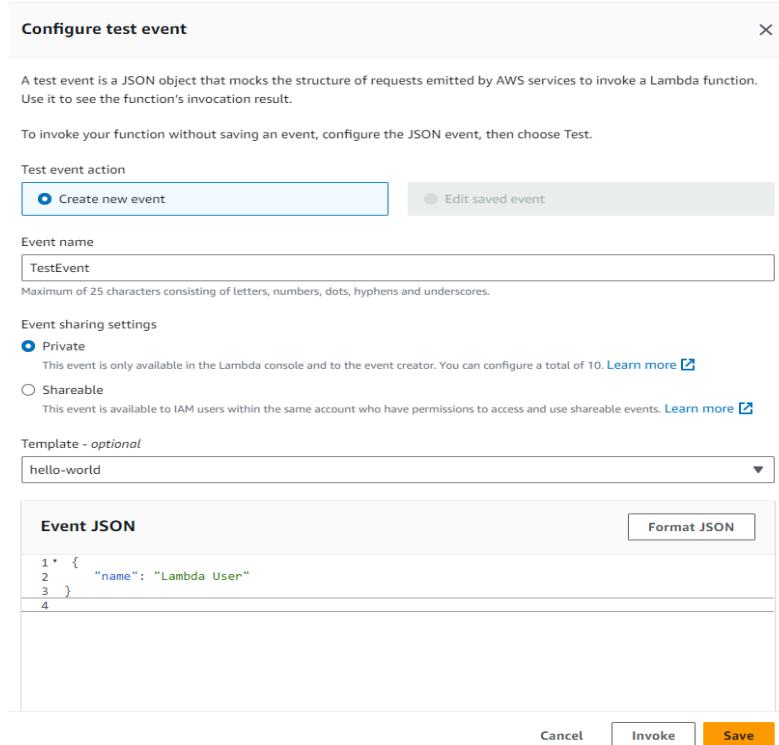
```
{
  "name": "Lambda User"
}
```

2. Run the Test:

Click on the Test button again to execute your Lambda function.

You should see the execution results below the code editor, including the response:
Copy code

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



The screenshot shows the AWS Lambda Code source interface. At the top, there are tabs for Code, Test, Monitor, Configuration, Aliases, and Versions. The Code tab is selected. Below the tabs, there's a toolbar with File, Edit, Find, View, Go, Tools, Window, Test, Deploy, and a status message 'Changes not deployed'. On the left, there's a sidebar labeled 'Environment' with a search bar 'Go to Anything (Ctrl-P)' and a dropdown menu showing 'Myfirstlambda' and 'lambda_function.py'. The main area displays 'Execution results' for a 'Test Event Name' named 'TestEvent'. It shows the 'Response' code:

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

Below the response, 'Function Logs' show the request and response details:

```
START RequestId: 36449800-5b8a-496e-83f6-7de19be2aa3c Version: $LATEST  
END RequestId: 36449800-5b8a-496e-83f6-7de19be2aa3c  
REPORT RequestId: 36449800-5b8a-496e-83f6-7de19be2aa3c Duration: 2.08 ms Billed Duration: 3 ms Mem  
Request ID  
36449800-5b8a-496e-83f6-7de19be2aa3c
```

Conclusion:

AWS Lambda is a serverless computing service that allows you to run code without managing servers, making it highly scalable, cost-effective, and easy to use. It automatically manages the compute resources, executes your code in response to specific events such as API calls, file uploads, or database updates, and scales based on the demand.

Adv. DevOps Exp. 12

Swaraj Patil
D15A 40

Step 1: Open the IAM (user)

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

Role name	Trusted entities	Last activity
aws-elasticbeanstalk-service-role-2	AWS Service: elasticbeanstalk	40 days ago
AWSServiceRoleForAutoScaling	AWS Service: autoscaling (Service-Link)	40 days ago
AWSServiceRoleForSupport	AWS Service: support (Service-Link)	-
AWSServiceRoleForTrustedAdvisor	AWS Service: trustedadvisor (Service-Link)	-
myPythonLambdaFunction-role-a2x7el65	AWS Service: lambda	-
test-2-role	AWS Service: ec2	40 days ago

Step 2: Under Attach Policies, add S3-ReadOnly and CloudWatchFull permissions to this role.

The screenshot shows the AWS IAM Role Summary page for the role "myPythonLambdaFunction-role-a2x7el65". The left sidebar has the same navigation as the previous screen. The main area shows the role's summary information and a "Permissions" tab.

Summary

Creation date	ARN
October 07, 2023, 16:05 (UTC+05:30)	arn:aws:iam::447953971928:role/service-role/myPythonLambdaFunction-role-a2x7el65

Permissions

Permissions policies (1) Info
You can attach up to 10 managed policies.

Add permissions ▲
Attach policies
Create inline policy

S3-ReadOnly

IAM > Roles > myPythonLambdaFunction-role-a2x7el65 > Add permissions

Attach policy to myPythonLambdaFunction-role-a2x7el65

▶ Current permissions policies (1)

Other permissions policies (882)

Filter by Type: All types | 1 match

Policy name	Type	Description
AmazonS3ReadOnlyAccess	AWS managed	Provides read only access to all bucket...

Cancel Add permissions

CloudWatchFull

IAM > Roles > myPythonLambdaFunction-role-a2x7el65 > Add permissions

Attach policy to myPythonLambdaFunction-role-a2x7el65

▶ Current permissions policies (2)

Other permissions policies (881)

Filter by Type: All types | 2 matches

Policy name	Type	Description
CloudWatchFullAccess	AWS managed	Provides full access to CloudWatch.
CloudWatchFullAccessV2	AWS managed	Provides full access to CloudWatch.

Cancel Add permissions

After successful attachment of policy you will see something like this you will be able to see the updated policies.

Identity and Access Management (IAM)

Policy was successfully attached to role.

Last activity: 1 hour ago (Duration: 1 hour)

Permissions | Trust relationships | Tags | Access Advisor | Revoke sessions

Permissions policies (3) Info

You can attach up to 10 managed policies.

Filter by Type: All types

Policy name	Type	Attached entities
AmazonS3ReadOnlyAccess	AWS managed	1
AWSLambdaBasicExecutionRole	Customer managed	1
CloudWatchFullAccess	AWS managed	1

Permissions boundary (not set)

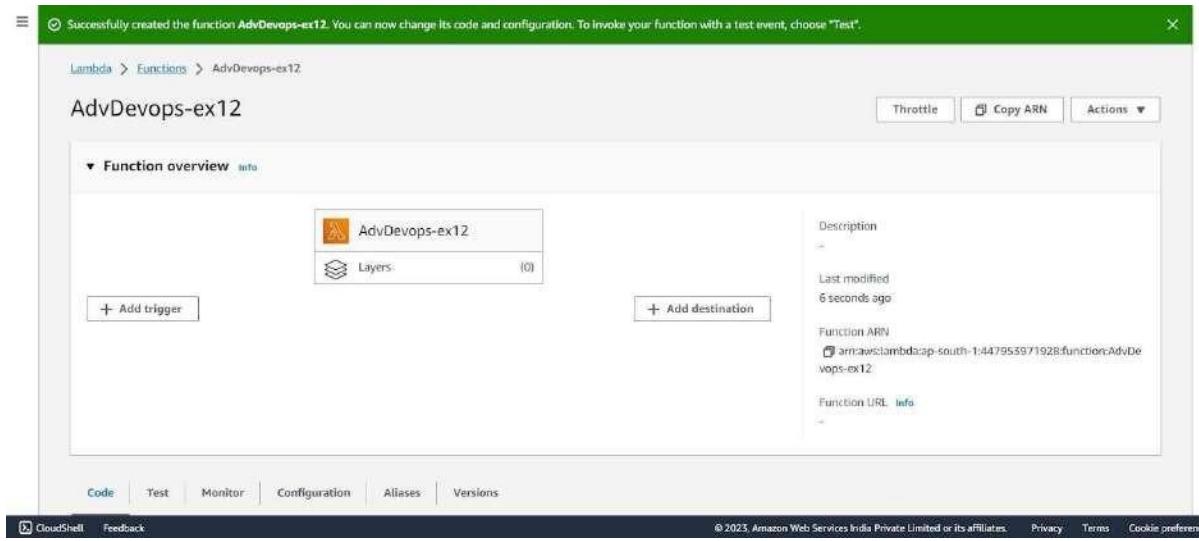
Step 3: Open up AWS Lambda and create a new Python function.

The screenshot shows the 'Create function' wizard in the AWS Lambda console. It includes sections for 'Basic information' (Function name: AdvDevOps-est12, Runtime: Python 3.11), 'Architecture' (x86_64 selected), and 'Permissions' (Use an existing role: service-role/myPythonLambdaFunctionRole-a2x7el6S selected). The bottom navigation bar includes CloudShell, Feedback, and links to Privacy, Terms, and Cookie preference.

Under Execution Role, choose the existing role, then select the one which was previously created and to which we just added permissions.

The screenshot shows the 'Create function' wizard in the AWS Lambda console, specifically the 'Execution role' configuration step. It shows the 'Use an existing role' option selected (service-role/myPythonLambdaFunctionRole-a2x7el6S) and the 'Advanced settings' section expanded. The bottom navigation bar includes CloudShell, Feedback, and links to Privacy, Terms, and Cookie preference.

Step 4: The function is up and running.



Step 5: Make the following changes to the function and click on the deploy button. This code basically logs a message and logs the contents of a JSON file which is uploaded to an S3 Bucket and then deploy the code.

```
1 import json
2 import boto3
3 import urllib
4
5 def lambda_handler(event, context):
6
7     s3_client = boto3.client('s3')
8     bucket_name = event['Records'][0]['s3']['bucket']['name']
9     key_event = event['Records'][0]['s3']['object']['key']
10    key_unquote_plus = key_event.unquote_plus(key_event, encoding='utf-8')
11    message = "An file has been added with key " + key_event + " to the bucket " + bucket_name
12    print(message)
13    response_s3_client = s3_client.get_object(Bucket=bucket_name, Key=key_event)
14    contents_response = response_s3_client['Body'].read().decode()
15    contents = json.loads(contents_response)
16
17    print("These are the Contents of the File: \n", contents)
18
19
```

The code is written in Python. It uses the boto3 library to interact with S3. The lambda_handler function takes an event and context as parameters. It retrieves the bucket name and key from the event. It then prints a message indicating a file has been added. It uses the s3_client to get the object content and decodes it. Finally, it prints the contents of the file.

Step 6: Click on Test and choose the 'S3 Put' Template.

The screenshot shows the AWS Lambda console interface. At the top, there's a navigation bar with the AWS logo, 'Services' dropdown, a search bar, and a keyboard shortcut '[Alt+S]'. A green success message states: 'Successfully created the function AdvDevops-ex12. You can now change its code and configuration. To invoke your function, click on the "Test" tab.' Below the message, there are tabs for 'Code', 'Test', 'Monitor', 'Configuration', 'Aliases', and 'Versions'. The 'Code' tab is selected, showing the 'Code source' section. It includes a file browser with a folder named 'AdvDevops-ex12' containing a file 'lambda_function.py'. The code editor displays the following Python code:

```
1 import json
2 import boto3
3 import urllib
4
5 def lambda_handler(event, context):
```

Below the code editor is a 'Configure test event' dialog. The dialog title is 'Configure test event' with a close button 'X'. The main text in the dialog says: 'A test event is a JSON object that mocks the structure of requests emitted by AWS services to invoke a Lambda function. Use it to see the function's invocation result.' Below this, there's a note: 'To invoke your function without saving an event, configure the JSON event, then choose Test.' Under 'Test event action', there are two options: 'Create new event' (selected) and 'Edit saved event'. The 'Event name' field is set to 'test'. A note below it says: 'Maximum of 25 characters consisting of letters, numbers, dots, hyphens and underscores.' Under 'Event sharing settings', there are two options: 'Private' (selected) and 'Shareable'. A note for 'Private' says: 'This event is only available in the Lambda console and to the event creator. You can configure a total of 10.' A link 'Learn more' is provided. A note for 'Shareable' says: 'This event is available to IAM users within the same account who have permissions to access and use shareable events.' Another link 'Learn more' is provided. Under 'Template - optional', there's a dropdown menu currently set to 's3-put'. At the bottom of the dialog, there are buttons for 'Format JSON', 'Cancel', 'Invoke' (disabled), and 'Save'.

And Save it.

Step 7: Open up the S3 Console and create a new bucket.

The screenshot shows the AWS S3 Buckets page. At the top, there's an 'Account snapshot' section with a link to 'Learn more'. To the right is a 'View Storage Lens dashboard' button. Below this is a table titled 'Buckets (3) info'. The table has columns for Name, AWS Region, Access, and Creation date. The buckets listed are:

Name	AWS Region	Access	Creation date
elasticbeanstalk-ap-south-1-447953971928	Asia Pacific (Mumbai) ap-south-1	Objects can be public	August 7, 2023, 14:24:02 (UTC+05:30)
www.hellorchane.com	Asia Pacific (Mumbai) ap-south-1	Public	July 30, 2023, 15:05:54 (UTC+05:30)
www.htmlwebside.com	Asia Pacific (Mumbai) ap-south-1	Public	July 30, 2023, 15:49:06 (UTC+05:30)

At the bottom of the page are links for CloudShell, Feedback, and cookie preferences.

Step 8: With all general settings, create the bucket in the same region as the function.

The screenshot shows the 'Create bucket' wizard. The first step, 'General configuration', is completed with the bucket name 'AdDevspsepg12' and the AWS Region set to 'Asia Pacific (Mumbai) ap-south-1'. The 'Copy settings from existing bucket - optional' section is present but empty. The second step, 'Object Ownership', is partially visible below. At the bottom of the page are links for CloudShell, Feedback, and cookie preferences.

Step 9: Click on the created bucket and under properties, look for events.

The screenshot shows the properties page for a newly created bucket. Under 'Event notifications (0)', there's a note to 'Send a notification when specific events occur in your bucket'. A 'Create event notification' button is available. The 'Amazon EventBridge' section indicates 'Off' for sending notifications to EventBridge. The 'Transfer acceleration' section notes that an accelerated endpoint is used for faster data transfers. At the bottom of the page are links for CloudShell, Feedback, and cookie preferences.

Click on Create Event Notification.

Step 10: Mention an event name and check Put under event types.

The screenshot shows the 'General configuration' section of the AWS S3 console. The 'Event name' field contains 'S3putrequest'. Under 'Event types', the 'Put' checkbox is checked, with the sub-type 's3:ObjectCreated:Put' selected. Other options like 'Post' and 'All object create events' are unselected. The status bar at the bottom indicates '© 2023, Amazon Web Services India Private Limited'.

Choose Lambda function as destination and choose your lambda function and save the changes.

The screenshot shows the 'Destination' configuration page. A note states that before publishing messages, permissions must be granted to the Amazon S3 principal. The 'Destination' section shows 'Lambda function' selected as the destination type. Under 'Specify Lambda function', 'Choose from your Lambda functions' is selected. The Lambda function dropdown shows 'AdvDevops-ex12'. At the bottom right are 'Cancel' and 'Save changes' buttons.

Step 11: Refresh the Lambda function console and you should be able to see an S3 Trigger in the overview.

The screenshot shows the AWS Lambda Function Overview page for a function named 'bp_nk'. At the top, there are buttons for Throttle, Copy ARN, and Actions. Below that is a 'Function overview' section with tabs for Diagram (selected) and Template. The Diagram view shows a main function box labeled 'bp_nk' with a 'Layers' section below it, and an 'S3' trigger box connected to it. There are '+ Add destination' and '+ Add trigger' buttons. To the right, there's a 'Description' field, 'Last modified' (6 days ago), 'Function ARN' (arn:aws:lambda:us-east-1:061051254449:function:bp_nk), and a 'Function URL' link. Below the overview are tabs for Code, Test, Monitor, Configuration, Aliases, and Versions. The 'Code source' tab is selected, showing an 'Upload from' button. The bottom navigation bar includes links for Home, Lambda, Functions, and API Gateway.

Step 12: Now, create a dummy JSON file locally.

Step 13: Go back to your S3 Bucket and click on Add Files to upload a new file.

Step 14: Select the dummy data file from your computer and click Upload.

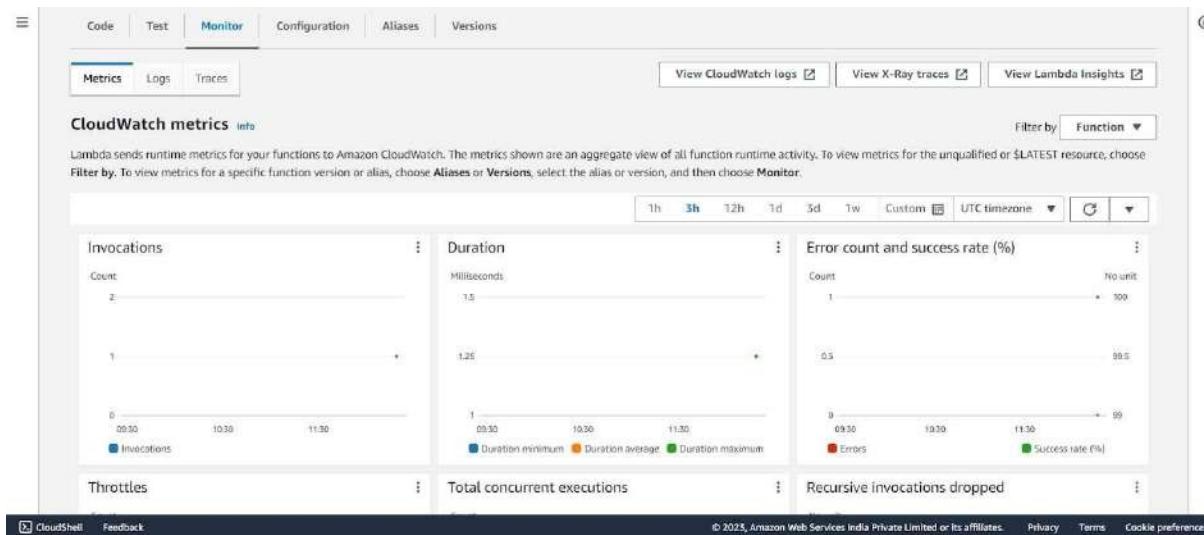
The screenshot shows the AWS S3 'Upload' interface. At the top, there's a navigation bar with 'Services' and a search bar. Below it, the path 'Amazon S3 > Buckets > advopssexp12 > Upload' is visible. The main area is titled 'Upload' with an 'Info' link. A note says: 'Add the files and folders you want to upload to S3. To upload a file larger than 160GB, use the AWS CLI, AWS SDK or Amazon S3 REST API. Learn more.' Below this is a large input box with the placeholder 'Drag and drop files and folders you want to upload here, or choose Add files or Add folder.' Underneath is a table titled 'Files and folders (1 Total, 89.0 B)'. It contains one item: 'dummy.json' (application/json, 89.0 B). There are 'Remove', 'Add files', and 'Add folder' buttons above the table. A search bar labeled 'Find by name:' is also present. The 'Destination' section shows 'Destination' set to 's3://advopssexp12'. At the bottom, there are 'CloudShell' and 'Feedback' links, and a copyright notice: '© 2023, Amazon Web Services India Private Limited or its affiliates.'

Step 15: After this make the necessary changes in the Test configuration file which we created it previously by replacing the Bucket Name and the ARN of Bucket.

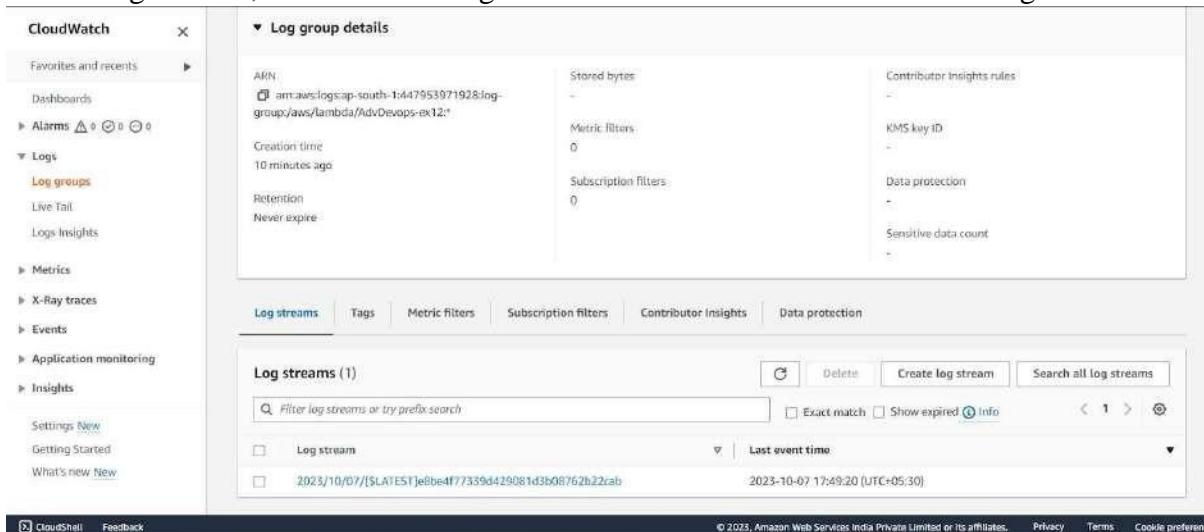
The screenshot shows the Lambda Test Configuration JSON editor. The title is 'Event JSON'. The content is a multi-line JSON document with line numbers from 10 to 38. The JSON describes an S3 object creation event. Key parts include:

```
10     "principalId": "EXAMPLE"
11 },
12 "requestParameters": {
13   "sourceIPAddress": "127.0.0.1"
14 },
15 "responseElements": {
16   "x-amz-request-id": "EXAMPLE123456789",
17   "x-amz-id-2": "EXAMPLE123/5678abcdefhijklambdaisawesome/mnopqrstuvwxyzABCDEFGHIJ"
18 },
19 "s3": {
20   "s3SchemaVersion": "1.0",
21   "configurationId": "testConfigRule",
22   "bucket": {
23     "name": "advopssexp12",
24     "ownerIdentity": {
25       "principalId": "EXAMPLE"
26     },
27     "arn": "arn:aws:s3:::advopssexp12"
28   },
29   "object": {
30     "key": "test%2Fkey",
31     "size": 1024,
32     "eTag": "0123456789abcdef0123456789abcdef",
33     "sequencer": "0A1B2C3D4E5F678901"
34   }
35 },
36 [
37 ]
38 }
```

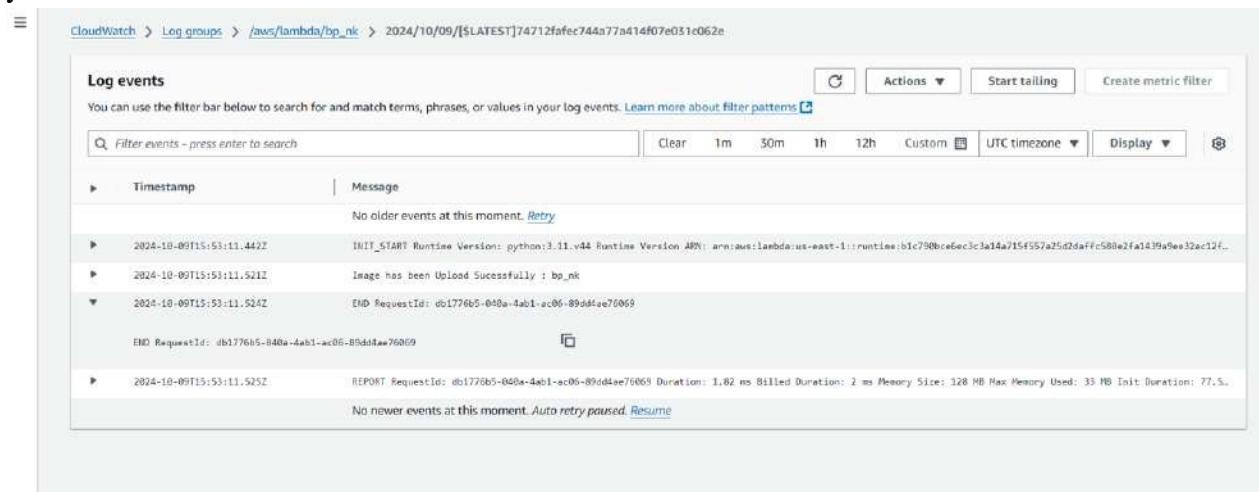
Step 16: Go back to your Lambda function , Refresh it and check the Monitor tab.



Under Log streams, click on View logs in Cloudwatch to check the Function logs.



Step 17: Click on this log Stream that was created to view what was logged by your function.



Conclusion: Thus, we have created a Lambda function which logs “An Image has been added” once you add an object to a specific bucket in S3.