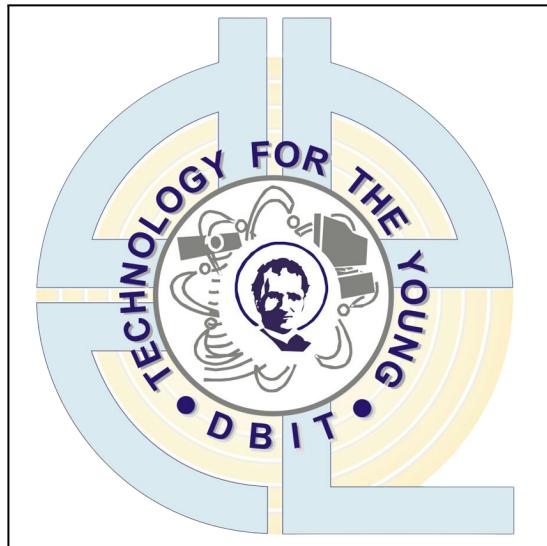


Don Bosco Institute of Technology(DBIT), Mumbai

Department of Information Technology



LAB JOURNAL

On

I.T/T.E/Sem V/ITL504 : AdvDevOps Lab

By

43 Ravi Pandey

Academic Year : Nov,2022
INDEX

Sr no	Exp No	Name of the experiment	Date
1.	1	To understand the benefits of Cloud Infrastructure and Setup AWS Cloud9 IDE, Launch AWS Cloud9 IDE and Perform Collaboration Demonstration	22/07/2022
2.	2	To Build Your Application using AWS CodeBuild and Deploy on S3 / SEBS using AWS CodePipeline, deploy Sample Application on EC2 instance using AWS CodeDeploy.	31/07/2022
3	5	To understand terraform lifecycle, core concepts/terminologies and install it on a Linux Machine.	20/08/2022
4	6	To Build, change, and destroy AWS / GCP /Microsoft Azure/ DigitalOcean infrastructure Using Terraform.	27/09/2022
5	7	To understand Static Analysis SAST process and learn to integrate Jenkins SAST to SonarQube/GitLab.	03/09/2022
6	8	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	10/09/2022
7	9	To Understand Continuous monitoring and Installation and configuration of Nagios Core, Nagios Plugins and NRPE (Nagios Remote Plugin	17/09/2022

		Executor) on Linux Machine	
8	10	To perform Port, Service monitoring, Windows/Linux server monitoring using Nagios	24/09/2022
9	11	To understand AWS Lambda, its workflow, various functions and create your first Lambda functions using Python / Java / Nodejs	01/10/2022
10	12	To create a Lambda function which will log “An Image has been added” once you add an object to a specific bucket in S3.	08/10/2022

AdvDevOps

Experiment No 1

Name - Ravi Pandey

Roll no - 43

Batch - B

Date - 22/07/2022

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

- **What is AWS?**

- Amazon Web Services, Inc. (AWS) is a subsidiary of [Amazon](#) that provides [on-demand cloud computing platforms](#) and [APIs](#) to individuals, companies, and governments, on a metered pay-as-you-go basis. These cloud computing [web services](#) provide [distributed computing](#) processing capacity and software tools via AWS [server farms](#). One of these services is [Amazon Elastic Compute Cloud](#) (EC2), which allows users to have at their disposal a [virtual cluster of computers](#), available all the time, through the Internet. AWS's virtual computers emulate most of the attributes of a real computer, including hardware [central processing units](#) (CPUs) and [graphics processing units](#) (GPUs) for processing; local/[RAM](#) memory; hard-disk/[SSD storage](#); a choice of operating systems; networking; and pre-loaded application software such as [web servers](#), [databases](#), and [customer relationship management](#) (CRM).

- **Step - 1**

- **Create a free account on AWS**

The screenshot shows the 'Sign up for AWS' page. It has two main input fields: 'Root user email address' and 'AWS account name'. Below these is a large orange button labeled 'Verify email address'. At the bottom, there is a horizontal line with the word 'OR' in the center, and below it is another input field labeled 'Sign in to an existing AWS account'.

Sign up for AWS

Root user email address
Used for account recovery and some administrative functions

AWS account name
Choose a name for your account. You can change this name in your account settings after you sign up.

Verify email address

OR

Sign in to an existing AWS account

- Step -2
-Search for cloud9

The screenshot shows the AWS Cloud search interface. In the top navigation bar, there is a search bar containing the text "cloud9". Below the search bar, the sidebar lists various categories: Services (39), Features (22), Blogs (4,551), Documentation (36,118), Knowledge Articles (30), Tutorials (24), Events (254), and Marketplace (11). The main content area is titled "Services" and displays four results: "Cloud9" (A Cloud IDE for Writing, Running, and Debugging Code), "AWS Cloud Map" (Build a dynamic map of your cloud), "Lightsail" (Launch and Manage Virtual Private Servers), and "WorkSpaces". A link "See all 39 results ▶" is located at the top right of the results list.

- Step-3
- Create environment and write sample code

The screenshot shows the "Name environment" step of the AWS Cloud9 environment creation wizard. On the left, a vertical navigation bar indicates "Step 1 Name environment", "Step 2 Configure settings", and "Step 3 Review". The main content area is titled "Name environment" and contains a section titled "Environment name and description". It includes fields for "Name" (with placeholder text "The name needs to be unique per user. You can update it at any time in your environment settings.") and "Description - Optional" (with placeholder text "This will appear on your environment's card in your dashboard. You can update it at any time in your environment settings."). Both fields have character limits: 60 for the name and 200 for the description. At the bottom right of the form are "Cancel" and "Next step" buttons.

Environment name and settings

Name
Lab1

Description
DevOps-1

Environment type
EC2

Instance type
t2.micro

Subnet

Platform
Amazon Linux 2 (recommended)

Cost-saving settings
After 30 minutes (default)

IAM role
AWSServiceRoleForAWSCloud9 (generated)

File Edit Find View Go Run Tools Window Support Preview Run

Go to Anything (Ctrl-P)

Lab1 - /home/ec2

- hello.py
- README.md

```
1 print("Hello This is AWS!")
```

bash - "ip-172-31-46-166.x" hello.py - Stopped x +

Run Command: hello.py

Hello This is AWS!

Process exited with code: 0

● Step - 4

- Add new IAM User

Set user details

You can add multiple users at once with the same access type and permissions. [Learn more](#)

User name* tushar4303

[+ Add another user](#)

Select AWS access type

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

Select AWS credential type*

Access key - Programmatic access

Enables an **access key ID** and **secret access key** for the AWS API, CLI, SDK, and other development tools.

Password - AWS Management Console access

Enables a **password** that allows users to sign-in to the AWS Management Console.

Console password*

Autogenerated password

Custom password

.....

Show password

Require password reset

User must create a new password at next sign-in

Users automatically get the [IAMUserChangePassword](#) policy to allow them to change their own password.

Review

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

User details

User name tushar4303

AWS access type Programmatic access and AWS Management Console access

Console password type Custom

Require password reset Yes

Permissions boundary Permissions boundary is not set

Permissions summary

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

Type	Name
Managed policy	IAMUserChangePassword

Tags

No tags were added.

- Step-5
 - Download the credentials

Success
You successfully created the users shown below. You can view and download user security credentials. You can also email users instructions for signing in to the AWS Management Console. This is the last time these credentials will be available to download. However, you can create new credentials at any time.

Users with AWS Management Console access can sign-in at: <https://696940912922.signin.aws.amazon.com/console>

[Download .csv](#)

	User	Access key ID	Secret access key	Email login instructions
▶	<input checked="" type="checkbox"/> tushar4303	AKIA2ERHKMENB5QDTINS	***** Show	Send email

- Step-6
 - Copy the console link and open in other browser

Summary

User ARN: arn:aws:iam::696940912922:user/tushar4303

Path: /

Creation time: 2022-07-22 22:00 UTC+0530

Permissions Groups Tags Security credentials Access Advisor

Sign-in credentials

Summary: Console sign-in link: <https://696940912922.signin.aws.amazon.com/console>

Console password: Enabled (last signed in Today) | Manage

Assigned MFA device: Not assigned | Manage

Signing certificates: None

Reset to default layout [+ Add widgets](#)

Welcome to AWS

Getting started with AWS [↗](#)
Learn the fundamentals and find valuable information to get the most out of AWS.

Training and certification [↗](#)
Learn from AWS experts and advance your skills and knowledge.

Account ID: 6969-4091-2922 [↗](#)
IAM user: tushar4303 [↗](#)

Account
Organization
Service Quotas
Billing Dashboard
Security credentials
Settings

- **Step-7**

- Copy the environment ID

Share this environment ×

Links to share

Environment: <https://us-east-1.console.aws.amazon.com/cloud9/ide/6046934d7a32>

Application: 54.175.214.198

To make your application accessible from the internet, please follow [our documentation](#).

Who has access

▼ ReadWrite

- You (online) RW
- tushar4303 (online) R RW

Don't allow members to save their tab state

Invite Members

IAM username R RW Invite

Invite an existing IAM user or [create a new user](#).

Done

- **Step-8**

-Paste the environment ID in other browser and make some changes to verify the connection

The screenshot shows the AWS Cloud9 IDE interface. The top bar displays the URL: us-east-1.console.aws.amazon.com/cloud9/ide/6046934d7a32433abc5e3b1f252d83... The main workspace shows a file named 'aloha.py' with the following code:

```
1 print("Aloha Protocol. Hello this is Iam-User")
```

The terminal below shows the output of the script:

```
Aloha Protocol. Hello this is Iam-User
```

Process exited with code: 0

ADVANCE DEVOPS LAB

Name : Ravi Pandey

Roll No. - 43

Experiment No.: 02

Aim : To Build Your Application using AWS CodeBuild and Deploy on S3 / SEBS using AWS CodePipeline, deploy Sample Application on EC2 instance using AWS CodeDeploy.

OutPut :

STEP 1 :

The screenshot shows the AWS S3 Buckets page. At the top, there is a green banner with the message "Successfully created bucket 'ravi-artifact'. To upload files and folders, or to configure additional bucket settings choose View details." Below this, a blue banner says "Learn how to effectively use the S3 Storage Classes." The main content area shows an "Account snapshot" with a link to "View Storage Lens dashboard". The "Buckets" section has a heading "Buckets (1) Info". It lists one bucket: "ravi-artifact" (Name), "US East (N. Virginia) us-east-1" (AWS Region), "Objects can be public" (Access), and "July 29, 2022, 01:09:49 (UTC+05:30)" (Creation date). There are buttons for "Copy ARN", "Empty", "Delete", and "Create bucket". A search bar at the bottom left and navigation controls at the bottom right are also visible.

STEP 2

The screenshot shows the AWS S3 Buckets page again. The top banner now says "Learn how to effectively use the S3 Storage Classes." The main content area shows an "Account snapshot" and the "Buckets" section. The "Buckets" section has a heading "Buckets (2) Info". It lists two buckets: "ravi-artifact" and "ravi-destination", both from "US East (N. Virginia) us-east-1". Both buckets have "Objects can be public" access and were created on "July 29, 2022, 01:09:49 (UTC+05:30)". The same set of buttons ("Copy ARN", "Empty", "Delete", "Create bucket") and search/nav controls are present as in the first screenshot.

STEP 3:

Edit static website hosting Info

Static website hosting
Use this bucket to host a website or redirect requests. [Learn more](#)

Static website hosting
 Disable
 Enable

Hosting type
 Host a static website
 Use the bucket endpoint as the web address. [Learn more](#)
 Redirect requests for an object
 Redirect requests to another bucket or domain. [Learn more](#)

For your customers to access content at the website endpoint, you must make all your content publicly readable. To do so, you can edit the S3 Block Public Access settings for the bucket. For more information, see [Using Amazon S3 Block Public Access](#)

Index document
Specify the home or default page of the website.

Error document - *optional*
This is returned when an error occurs.

STEP 4:

Create build project

Project configuration

Project name

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

Description - *optional*

Build badge - *optional*
 Enable build badge

Enable concurrent build limit - *optional*
Limit the number of allowed concurrent builds for this project.
 Restrict number of concurrent builds this project can start

► Additional configuration
tags

STEP 5

Source

Add source

Source 1 - Primary

Source provider

GitHub

Repository

Public repository Repository in my GitHub account

GitHub repository

https://github.com/Raviipandey/It-s_Me.io.git X C

https://github.com/<user-name>/<repository-name>

Connection status

You are connected to GitHub using OAuth.

Disconnect from GitHub

Source version - *optional info*

Enter a pull request, branch, commit ID, tag, or reference and a commit ID.

▶ Additional configuration
Git clone depth, Git submodules, Build status config

STEP 6:

Environment type

Linux

Privileged

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

Service role

New service role
Create a service role in your account Existing service role
Choose an existing service role from your account

Role name

codebuild-ravi-lab2-service-role

Type your service role name

▶ Additional configuration
Timeout, certificate, VPC, compute type, environment variables, file systems

Buildspec

Build specifications

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

Buildspec name - *optional*

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

rvices, features, blogs, docs, and more [Alt+S]

Project created You have successfully created the following project: ravi-lab2

Create a notification rule for this project X

Developer Tools > CodeBuild > Build projects > ravi-lab2

ravi-lab2

Notify Share Edit Delete build project Start build with overrides Start build

Configuration

Source provider GitHub	Primary repository Raviipandey/It-s_Me.io	Artifacts upload location -	Build badge Disabled
Public builds Disabled			

Build history Batch history Build details Build triggers Metrics

Build history

C Stop build View artifacts View logs Delete builds Retry build < 1 > ⌂

Build run	Status	Build number	Source version	Submitter	Duration	Completed
No results There are no results to display.						

Developer Tools > CodePipeline > Pipelines > Create new pipeline

Step 1 Choose pipeline settings

Step 2 Add source stage

Step 3 Add build stage

Step 4 Add deploy stage

Step 5 Review

Add source stage Info

Source

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

GitHub (Version 1)

Grant AWS CodePipeline access to your GitHub repository. This allows AWS CodePipeline to upload commits from GitHub to your pipeline.

Connected

You have successfully configured the action with the provider. X

i The GitHub (Version 1) action is not recommended The selected action uses OAuth apps to access your GitHub repository. This is no longer the recommended method. Instead, choose the GitHub (Version 2) action to access your repository by creating a connection. Connections use GitHub Apps to manage authentication and can be shared with other resources. [Learn more](#)

Repository Raviipandey/It-s_Me.io

Branch main

Change detection options Choose a detection mode to automatically start your pipeline when a change occurs in the source code.

GitHub webhooks (recommended) Use webhooks in GitHub to automatically start my pipeline when a change occurs

AWS CodePipeline Use AWS CodePipeline to check periodically for changes

STEP 7:

Developer Tools > CodePipeline > Pipelines > Create new pipeline

Step 1 Choose pipeline settings

Step 2 Add source stage

Step 3 **Add build stage**

Step 4 Add deploy stage

Step 5 Review

Add build stage Info

Build - optional

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

AWS CodeBuild

Region
US East (N. Virginia)

Project name
Choose a build project that you have already created in the AWS CodeBuild console. Or create a build project in the AWS CodeBuild console and then return to this task.
ravi-lab2 or

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

Build type

Single build
Triggers a single build.

Batch build
Triggers multiple builds as a single execution.

Developer Tools > CodePipeline > Pipelines > Create new pipeline

Step 1 Choose pipeline settings

Step 2 Add source stage

Step 3 **Add build stage**

Step 4 **Add deploy stage**

Step 5 Review

Add deploy stage Info

Deploy - optional

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

Amazon S3

Region
US East (N. Virginia)

Bucket
ravi-destination

Deployment path - optional

Extract file before deploy
The deployed artifact will be unzipped before deployment.

Additional configuration

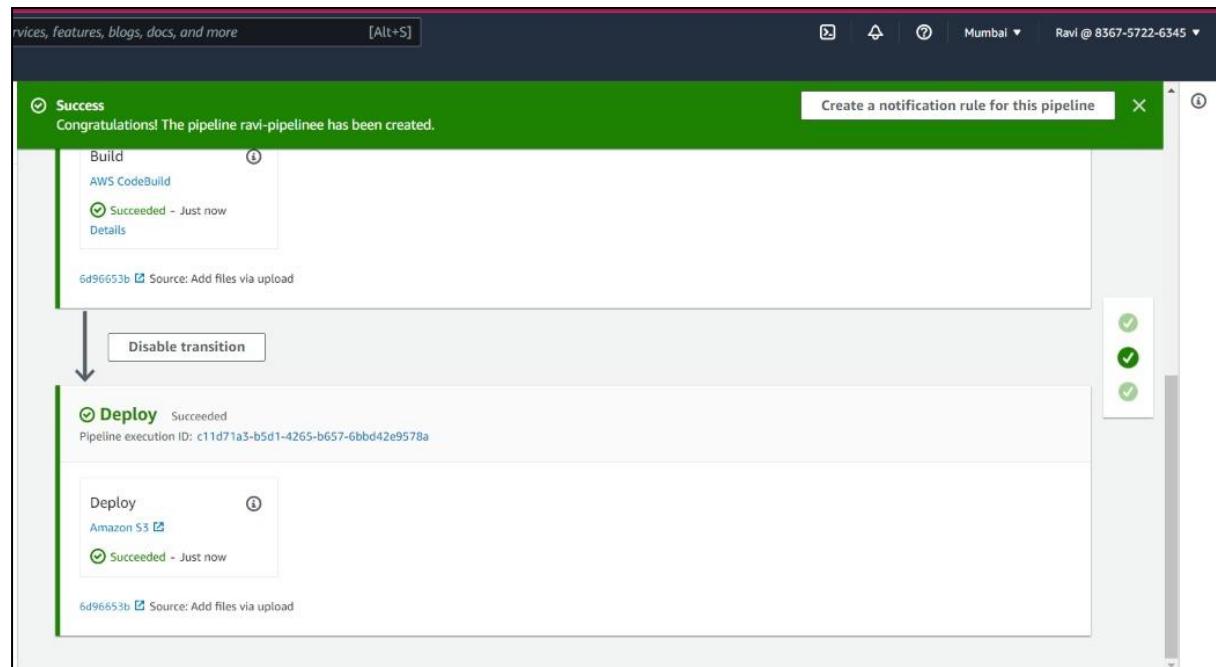
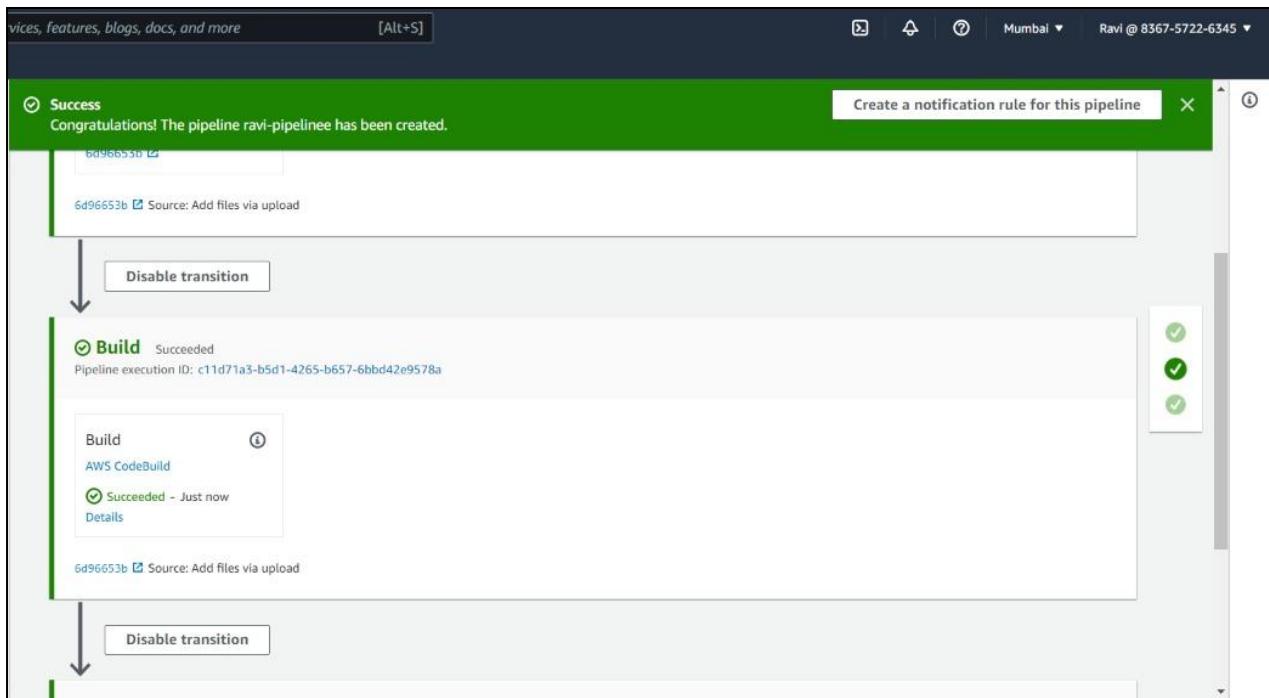
KMS encryption key ARN - optional
Encrypt your object using a KMS encryption key. If no key is provided, objects remain un-encrypted.

arn:aws:kms:<region-ID>:<account-ID>/key/<key-ID> OR arn:aws:kms:<region-ID>:<account-ID>:alias/<alias-name>

Canned ACL - optional
Specify an Amazon S3 canned access control list (ACL) for your bucket.
public-read-write

Cache control - optional
Set cache control for objects requested from your Amazon S3 bucket.

FINAL OUTPUT:



ADVANCE DEVOPS LAB

Name : Ravi Pandey

Roll No: 43

Exp No - 05

Aim:

To understand terraform lifecycle, core concepts/terminologies and install it on a Linux Machine

Output:

STEP 1 - Install the HashiCorp GPG key

```
(aloha@Kali)-[~/Downloads]
$ wget -O- https://apt.releases.hashicorp.com/gpg | \
  gpg --dearmor | \
  sudo tee /usr/share/keyrings/hashicorp-archive-keyring.gpg
--2022-08-12 02:44:16-- https://apt.releases.hashicorp.com/gpg
Resolving apt.releases.hashicorp.com (apt.releases.hashicorp.com) ... 13.227.138.116, 13.227.138.47, 13.227.138.63, .
...
Connecting to apt.releases.hashicorp.com (apt.releases.hashicorp.com)|13.227.138.116|:443 ... connected.
HTTP request sent, awaiting response... 200 OK
Length: 3195 (3.1K) [binary/octet-stream]
Saving to: 'STDOUT'

[  100%[=====] 3.12K --.-KB/s   in 0s

2022-08-12 02:44:16 (272 MB/s) - written to stdout [3195/3195]

^NKSz'4xqo
-----[REDACTED]-----
```

STEP 2 - Unzip terraform

```
└─(aloha㉿Kali)-[~/Downloads]
└─$ unzip terraform_1.2.7_linux_amd64.zip
Archive:  terraform_1.2.7_linux_amd64.zip
  inflating: terraform
```

STEP 3 - Verify the version

```
└─(aloha㉿Kali)-[~/Downloads]
└─$ terraform -version
Terraform v1.2.7
on linux_amd64

└─(aloha㉿Kali)-[~/Downloads]
└─$ terraform -h
Usage: terraform [global options] <subcommand> [args]

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

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

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

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

ADVANCE DEVOPS LAB

**Name : Ravi Pandey
Roll No: 43
Exp No - 06**

Aim:

To Build, change, and destroy AWS / GCP /Microsoft Azure/ DigitalOcean infrastructure Using Terraform

Output:

Building Infrastructure

STEP 1 - Installing AWS CLI

```
(aloha㉿Kali)-[~/Downloads] curl "https://awscli.amazonaws.com/awscli-exe-linux-x86_64.zip" -o "awscliv2.zip"
(aloha㉿Kali)-[~/Downloads] unzip awscliv2.zip
(aloha㉿Kali)-[~/Downloads] sudo ./aws/install
(aloha㉿Kali)-[~/Downloads] % Total    % Received % Xferd
(aloha㉿Kali)-[~/Downloads] 100 44.9M  100 44.9M      0      0 2495k      0  0:00:18  0:00:18 --:--:-- 2399k
(aloha㉿Kali)-[~/Downloads] Archive: awscliv2.zip
(aloha㉿Kali)-[~/Downloads]   creating: aws/
(aloha㉿Kali)-[~/Downloads]   creating: aws/dist/
(aloha㉿Kali)-[~/Downloads] inflating: aws/install
```

STEP 2 - Verification of AWS-CLI

```
inflating: aws/dist/lib/python3.9/config-3.9_x86_64-linux-gnu/makefile
creating: aws/dist/include/python3.9/
inflating: aws/dist/include/python3.9/pyconfig.h
You can now run: /usr/local/bin/aws --version
```

Callout file in one of the following ways:

(aloha㉿Kali)-[~/Downloads]

\$ aws --version

aws-cli/2.7.22 Python/3.9.11 Linux/5.18.0-kali5-amd64 exe/x86_64.kali.2022 prompt/off

(aloha㉿Kali)-[~/Downloads]

\$ [REDACTED]

- Use the curl command – The -o option specifies the file name that the download will be saved as. In this example command write the downloaded file to the current directory with the name makefile.

STEP 3 - AWS Configure(Access and Secret key)

```
└─(aloha㉿Kali)-[~/Downloads] HashiCorp Learn     Browse products ▾
  $ export AWS_ACCESS_KEY_ID= AKIA2ERHKMENOQE3C62PB ↗ Google Hacking DB ↗ OffSec
  └─(aloha㉿Kali)-[~/Downloads] HashiCorp Learn     Browse products ▾
  $ export AWS_SECRET_ACCESS_KEY=fmvqjdGPRFSqZCnJqPP4mOTJtpneyj0wAY2JMWB6
  └─(aloha㉿Kali)-[~/Downloads] Terraform
  $ |
```

STEP 4 - Making Folders

```
└─(aloha㉿Kali)-[~/Desktop] Terraform
  $ mkdir Terraform-AWS

  └─(aloha㉿Kali)-[~/Desktop] AWS
    $ cd Terraform-AWS

  └─(aloha㉿Kali)-[~/Desktop/Terraform-AWS]
    $ touch main.tf
```

STEP 5 - Configuration

```
(aloha㉿Kali)-[~/Desktop/Terraform-AWS]
$ nano main.tf

(aloha㉿Kali)-[~/Desktop/Terraform-AWS]
$ cat main.tf
terraform {
    required_providers {
        aws = {
            source  = "hashicorp/aws"
            version = "~> 4.16"      • Build Infrastructure
        }
    }
    required_version = "≥ 1.2.0"          Destroy Infrastructure
}                                         Define Input Variables

provider "aws" {                         Query Data with Output
    region  = "us-west-2"                  Store Remote State
}

resource "aws_instance" "app_server" {
    ami              = "ami-830c94e3"
    instance_type   = "t2.micro"

    tags = {
        Name = "ExampleAppServerInstance"
    }
}
```

STEP 6 - Terraform init

```
(aloha㉿Kali)-[~/Desktop/Terraform-AWS]
$ terraform init
      What is Infrastructure-as Code
      with Terraform?

Initializing the backend ...           Install Terraform

Initializing provider plugins ...     Build Infrastructure
- Finding hashicorp/aws versions matching "~ 4.16" ...
- Installing hashicorp/aws v4.25.0 ...
- Installed hashicorp/aws v4.25.0 (signed by HashiCorp)

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

Terraform has been successfully initialized!

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

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

STEP 7- Format and Validation

```
(aloha㉿Kali)-[~/Desktop/Terraform-AWS]
$ terraform fmt
main.tf

(aloha㉿Kali)-[~/Desktop/Terraform-AWS]
$ terraform validate
Success! The configuration is valid.
```

STEP 8 - Terraform Apply

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

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

Enter a value: yes

aws_instance.app_server: Creating ...
aws_instance.app_server: Still creating ... [10s elapsed]
aws_instance.app_server: Still creating ... [20s elapsed]
aws_instance.app_server: Still creating ... [30s elapsed]
aws_instance.app_server: Still creating ... [40s elapsed]
aws_instance.app_server: Creation complete after 46s [id=i-0c85527a94c57b64a]

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

STEP 9 - Terraform Show

```
└─(aloha㉿Kali)-[~/Desktop/Terraform-AWS]
$ terraform show
# aws_instance.app_server:
resource "aws_instance" "app_server" {
    ami                               = "ami-052efd3df9dad4825"
    arn                             = "arn:aws:ec2:us-east-1:696940912922:instance/i-0c85527a94c57b64a"
    associate_public_ip_address      = true
    availability_zone                = "us-east-1a"
    cpu_core_count                  = 1
    cpu_threads_per_core            = 1
    disable_api_stop                = false
    disable_api_termination         = false
    ebs_optimized                   = false
    get_password_data               = false
    hibernation                     = false
    id                               = "i-0c85527a94c57b64a"
    instance_initiated_shutdown_behavior = "stop"
    instance_state                  = "running"
    instance_type                   = "t2.micro"
    ipv6_address_count              = 0
    ipv6_addresses                  = []
    monitoring                      = false
    primary_network_interface_id   = "eni-0fd411e5af6a5f488"
    private_dns                     = "ip-172-31-24-218.ec2.internal"
    private_ip                      = "172.31.24.218"
    public_dns                      = "ec2-54-208-111-160.compute-1.amazonaws.com"
    public_ip                       = "54.208.111.160"
    secondary_private_ips           = []
    security_groups                 = [
        "default",
    ]
    source_dest_check               = true
    subnet_id                       = "subnet-08d1a332d67d1ca8d"
    tags                            = {
        "Name" = "ExampleAppServerInstance"
    }
    tags_all                         = {
        "Name" = "ExampleAppServerInstance"
    }
    tenancy                          = "default"
    user_data_replace_on_change     = false
    vpc_security_group_ids          = [
        "sg-046fc22136f55a9c4",
    ]
    capacity_reservation_specification {
        capacity_reservation_preference = "open"
    }
    credit_specification {
        cpu_credits = "standard"
    }
}
```

STEP 10 - Terraform State

```
(aloha㉿Kali)-[~/Desktop/Terraform-AWS]
$ terraform state list
aws_instance.app_server
```

STEP 11 - Checking Instance in AWS

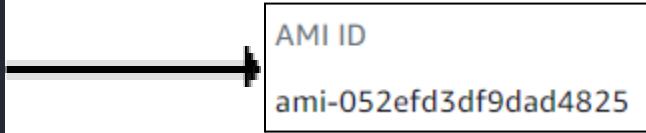
The screenshot shows the AWS EC2 Instances details page for an instance named `i-0c85527a94c57b64a`. The instance is currently running. Key details include:

- Public IPv4 address:** 54.208.111.160
- Private IP DNS name (IPv4 only):** ip-172-31-24-218.ec2.internal
- Instance type:** t2.micro
- VPC ID:** vpc-08e24b0b056f7a73a
- Subnet ID:** subnet-08d1a332d67d1ca8d
- AMI ID:** ami-052efd3df9dad4825
- AMI name:** ubuntu/images/hvm-ssd/ubuntu-jammy-22.04-amd64-server-20220609
- Launch time:** Sun Aug 14 2022 15:39:47 GMT-0400 (Eastern Daylight Time) (10 minutes)
- Lifecycle:** normal
- Monitoring:** disabled
- Termination protection:** Disabled
- AMI location:** 099720109477/ubuntu/images/hvm-ssd/ubuntu-jammy-22.04-amd64-server-20220609
- Stop-hibernate behavior:** disabled

Changing Infrastructure

STEP 1 - Change AMI number according to Selected region in main.tf

```
resource "aws_instance" "app_server" {  
    ami          = "ami-830c94e3"  
    instance_type = "t2.micro"
```



AMI ID
ami-052efd3df9dad4825

STEP 2 - Terraform Apply

```
# aws_instance.app_server must be replaced  
-/+ resource "aws_instance" "app_server" {  
    ~ ami  
    ~ arn  
  (known after apply)
```

```
Plan: 1 to add, 0 to change, 1 to destroy.  
  
Do you want to perform these actions?  
Terraform will perform the actions described above.  
Only 'yes' will be accepted to approve.  
  
Enter a value: yes  
  
aws_instance.app_server: Destroying ... [id=i-0a3ceb8ea6695c613]  
aws_instance.app_server: Still destroying ... [id=i-0a3ceb8ea6695c613, 10s elapsed]  
aws_instance.app_server: Still destroying ... [id=i-0a3ceb8ea6695c613, 20s elapsed]  
aws_instance.app_server: Still destroying ... [id=i-0a3ceb8ea6695c613, 30s elapsed]  
aws_instance.app_server: Destruction complete after 34s  
aws_instance.app_server: Creating ...  
aws_instance.app_server: Still creating ... [10s elapsed]  
aws_instance.app_server: Still creating ... [20s elapsed]  
aws_instance.app_server: Still creating ... [30s elapsed]  
aws_instance.app_server: Still creating ... [40s elapsed]  
aws_instance.app_server: Still creating ... [50s elapsed]  
aws_instance.app_server: Still creating ... [1m0s elapsed]  
aws_instance.app_server: Still creating ... [1m10s elapsed]  
aws_instance.app_server: Still creating ... [1m20s elapsed]  
aws_instance.app_server: Creation complete after 1m25s [id=i-04d3a3aad27e610f6]  
  
Apply complete! Resources: 1 added, 0 changed, 1 destroyed.
```

Destroying Infrastructure

```
|__ (aloha㉿Kali)-[~/Desktop/Terraform-AWS]
|__ (aloha㉿Kali)-[~/Desktop/Terraform-AWS]
└ $ terraform destroy
aws_instance.app_server: Refreshing state ... [id=i-04d3a3aad27e610f6]

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

Terraform will perform the following actions:

# aws_instance.app_server will be destroyed
- resource "aws_instance" "app_server" {
    - ami = "ami-08d70e59c07c61a3a" → null
    - arn = "arn:aws:ec2:us-west-2:696940912922:instance/i-04d3a3aad27e610f6" →
...11}
```

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

Do you really want to destroy all resources?

Terraform will destroy all your managed infrastructure, as shown above.
There is no undo. Only 'yes' will be accepted to confirm.

Enter a value: yes

```
aws_instance.app_server: Destroying ... [id=i-04d3a3aad27e610f6]
aws_instance.app_server: Still destroying ... [id=i-04d3a3aad27e610f6, 10s elapsed]
aws_instance.app_server: Still destroying ... [id=i-04d3a3aad27e610f6, 20s elapsed]
aws_instance.app_server: Still destroying ... [id=i-04d3a3aad27e610f6, 30s elapsed]
aws_instance.app_server: Destruction complete after 33s
```

Destroy complete! Resources: 1 destroyed.

ADVANCED DEVOPS LAB

Name : Ravi Pandey

Roll no : 43

Experiment No : 08

Aim: Perform static analysis using sonarqube and show the analysis

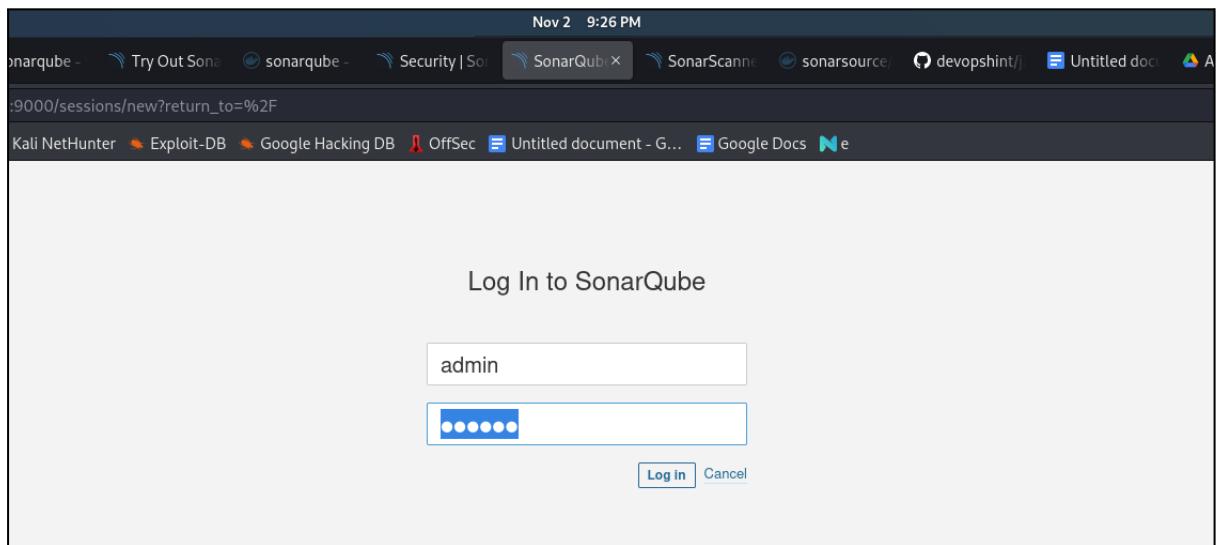
Output: Start sonarqube container in docker

```
(root@kali):~/home/kali]
└─$ docker ps -a
CONTAINER ID IMAGE COMMAND CREATED STATUS PORTS NAMES
e0c2aeb32f79 1d0a26834cb "/opt/sonarqube/bin/..." 37 minutes ago Created
a32c2773b07c hello-world "/hello" 25 hours ago Exited (0) 25 hours ago
37460966be62 hello-world "/Ravi" 25 hours ago Created
cd105e19e91f hello-world "/Hello" 25 hours ago Exited (0) 25 hours ago
a876a53ace2e hello-world "/Hello" 25 hours ago Exited (0) 25 hours ago
ac81020f7aa5 Hello-world "/Hello" 25 hours ago Created
51276421da5d docker/whalesay "cowsay boo" 26 hours ago Exited (0) 26 hours ago
b484ec2bb46a hello-world "/hello" 26 hours ago Exited (0) 26 hours ago
9dc2b74b2c8c ubuntu "bash" 3 weeks ago Exited (0) 3 weeks ago
d68dd3f81696 nginx "/docker-entrypoint..." 3 weeks ago Exited (255) 3 weeks ago 0.0.0.0:5000->80/tcp, :::5000->80/tcp NAMES
9cra2d57983 3728f8fc7302 "/usr/bin/tini -- /..." 3 weeks ago Exited (137) 5 days ago jenkins
2ccc06f04449 postgres:12 "/docker-entrypoint.s..." 3 weeks ago Exited (137) 3 weeks ago db

[root@kali]:~/home/kali]
└─$ docker start e0c2aeb32f79
e0c2aeb32f79 started at 2023-11-01T20:51:52+05:30

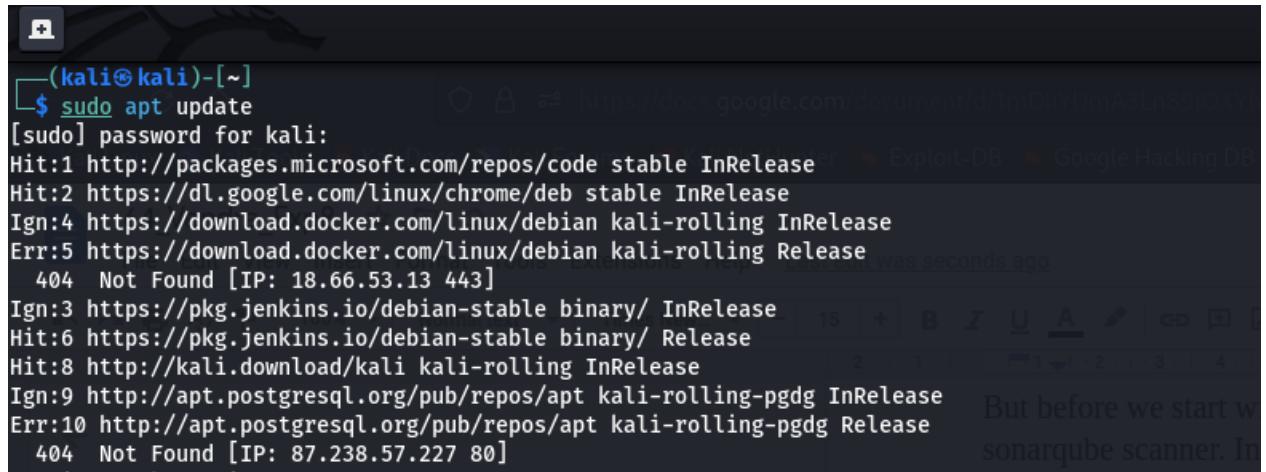
[root@kali]:~/home/kali]
└─$ docker ps
CONTAINER ID IMAGE COMMAND CREATED STATUS PORTS NAMES
e0c2aeb32f79 1d0a26834cb "/opt/sonarqube/bin/..." 38 minutes ago Up 5 seconds 0.0.0.0:9000->9000/tcp, :::9000->9000/tcp, 0.0.0.0:9092->9092/tcp, :::9092->9092/tcp NAMES
sonarqube
```

After the service is up, write in the browser `localhost:9000` and you will see the following



But before we start with the static code analysis, we need to download the sonarqube scanner. In order to install sonarqube we need to follow the steps given below:

1. sudo apt-get update



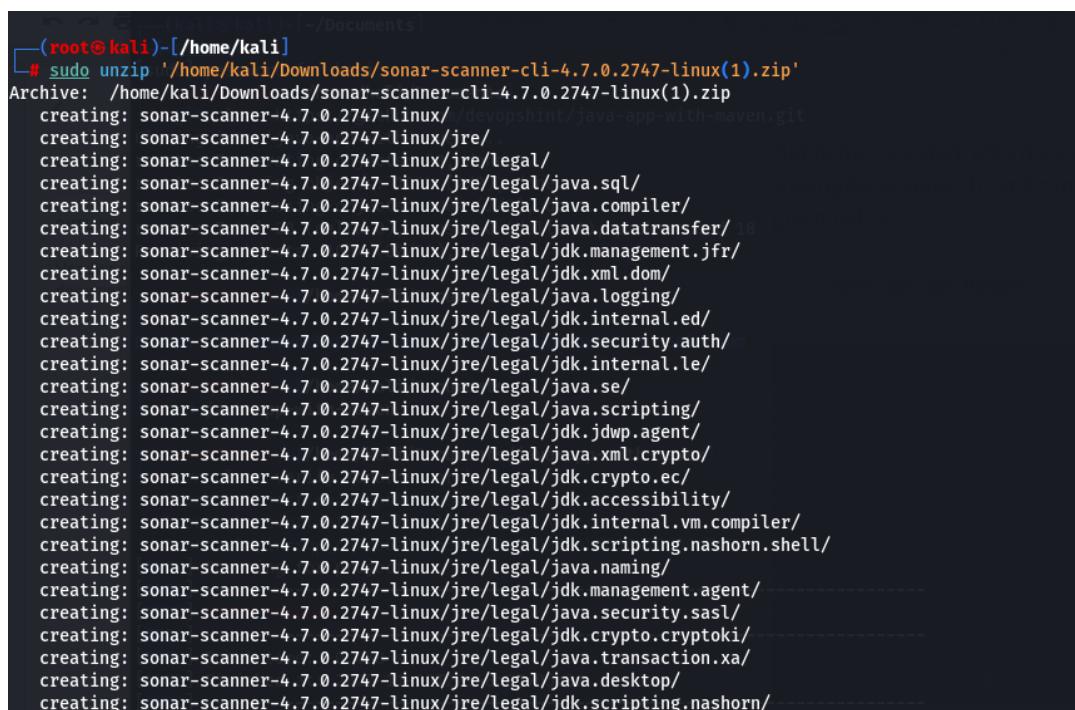
```
(kali㉿kali)-[~]
$ sudo apt update
[sudo] password for kali:
Hit:1 http://packages.microsoft.com/repos/code stable InRelease
Hit:2 https://dl.google.com/linux/chrome/deb stable InRelease
Ign:4 https://download.docker.com/linux/debian kali-rolling InRelease
Err:5 https://download.docker.com/linux/debian kali-rolling Release
  404  Not Found [IP: 18.66.53.13 443]
Ign:3 https://pkg.jenkins.io/debian-stable binary/ InRelease
Hit:6 https://pkg.jenkins.io/debian-stable binary/ Release
Hit:8 http://kali.download/kali kali-rolling InRelease
Ign:9 http://apt.postgresql.org/pub/repos/apt kali-rolling-pgdg InRelease
Err:10 http://apt.postgresql.org/pub/repos/apt kali-rolling-pgdg Release
   404  Not Found [IP: 87.238.57.227 80]
```

2. Install sonar scanner from the official link

<https://docs.sonarqube.org/latest/analysis/scan/sonarscanner/>



3. Unzip the installed zip file



```
(root㉿kali)-[~/Documents]
$ sudo unzip '/home/kali/Downloads/sonar-scanner-cli-4.7.0.2747-linux(1).zip'
Archive: /home/kali/Downloads/sonar-scanner-cli-4.7.0.2747-linux(1).zip
  creating: sonar-scanner-4.7.0.2747-linux/
  creating: sonar-scanner-4.7.0.2747-linux/jre/
  creating: sonar-scanner-4.7.0.2747-linux/jre/legal/
  creating: sonar-scanner-4.7.0.2747-linux/jre/legal/java.sql/
  creating: sonar-scanner-4.7.0.2747-linux/jre/legal/java.compiler/
  creating: sonar-scanner-4.7.0.2747-linux/jre/legal/java.datatransfer/
  creating: sonar-scanner-4.7.0.2747-linux/jre/legal/jdk.management.jfr/
  creating: sonar-scanner-4.7.0.2747-linux/jre/legal/jdk.xml.dom/
  creating: sonar-scanner-4.7.0.2747-linux/jre/legal/java.logging/
  creating: sonar-scanner-4.7.0.2747-linux/jre/legal/jdk.internal.ed/
  creating: sonar-scanner-4.7.0.2747-linux/jre/legal/jdk.security.auth/
  creating: sonar-scanner-4.7.0.2747-linux/jre/legal/jdk.internal.le/
  creating: sonar-scanner-4.7.0.2747-linux/jre/legal/java.se/
  creating: sonar-scanner-4.7.0.2747-linux/jre/legal/java.scripting/
  creating: sonar-scanner-4.7.0.2747-linux/jre/legal/jdk.jdwp.agent/
  creating: sonar-scanner-4.7.0.2747-linux/jre/legal/java.xml.crypto/
  creating: sonar-scanner-4.7.0.2747-linux/jre/legal/jdk.crypto.ec/
  creating: sonar-scanner-4.7.0.2747-linux/jre/legal/jdk.accessibility/
  creating: sonar-scanner-4.7.0.2747-linux/jre/legal/jdk.internal.vm.compiler/
  creating: sonar-scanner-4.7.0.2747-linux/jre/legal/jdk.scripting.nashorn.shell/
  creating: sonar-scanner-4.7.0.2747-linux/jre/legal/java.naming/
  creating: sonar-scanner-4.7.0.2747-linux/jre/legal/jdk.management.agent/
  creating: sonar-scanner-4.7.0.2747-linux/jre/legal/java.security.sasl/
  creating: sonar-scanner-4.7.0.2747-linux/jre/legal/jdk.crypto.cryptoki/
  creating: sonar-scanner-4.7.0.2747-linux/jre/legal/java.transaction.xa/
  creating: sonar-scanner-4.7.0.2747-linux/jre/legal/java.desktop/
  creating: sonar-scanner-4.7.0.2747-linux/jre/legal/jdk.scripting.nashorn/
```

4. Move the installed file into sonar-scanner in opt directory where sonarqube is present

```
(root㉿kali)-[~/home/kali]
└─# ls
AndroRAT  Documents  Music  Osintgram  snap  into the sonar-scanner direc...  Videos
Csi-webApp  Downloads  myMar2022WeekdayBatchRepo-main  Pictures  sonar-scanner-4.7.0.2747-linux  'VS Code'
Desktop  eclipse-workspace  nagios-4.4.6  Public  Templates

└─# (root㉿kali)-[~/home/kali]
└─# sudo mv sonar-scanner-4.7.0.2747-linux /
```

5. Go into the sonar-scanner directory and open the sonar-scanner.properties file

```
root@kali:~/sonar-scanner-4.7.0.2747-linux]
└─# ls
bin  conf  jre  lib

root@kali:[/sonar-scanner-4.7.0.2747-linux]
└─# cd conf

root@kali:[/sonar-scanner-4.7.0.2747-linux/conf]
└─# ls
sonar-scanner.properties

root@kali:[/sonar-scanner-4.7.0.2747-linux/conf]
└─# sudo nano sonar-scanner.properties
```

6. Add the localhost and encoding

```
root@kali:~/sonar-scanner-4.7.0.2747-linux/conf
GNU nano 6.4
sonar-scanner.properties *
#Configure here general information about the environment, such as SonarQube server connection details for example
#No information about specific project should appear here
#----- Default SonarQube server
sonar.host.url=http://localhost:9000
#----- Default source code encoding
sonar.sourceEncoding=UTF-8
```

7. Now make the scanner binary executable & Then create a symbolic link so that you can call the scanner without specifying the path

```
(root㉿kali)-[/>
└─# ls
0  dev  initrd.img  lib32  lost+found  opt  run  sonarqube  sys  var
bin  etc  initrd.img.old  lib64  media  proc  sbin  sonar-scanner-4.7.0.2747-linux  tmp  vmlinuz
boot  home  lib  libx32  mnt  root  snap  srv  usr  vmlinuz.old

└─# (root㉿kali)-[/>
└─# chmod +x sonar-scanner-4.7.0.2747-linux/bin/sonar-scanner
```

Now that our sonar-scanner is installed, we will create a project in our sonarqube that is logged in through <http://localhost:9000/projects/create>

The screenshot shows the SonarQube interface for creating a new project. At the top, there's a navigation bar with links for Projects, Issues, Rules, Quality Profiles, Quality Gates, and Administration. A notification bar at the top says "There's a new version of SonarQube available. Update to enjoy the latest features." Below this, a message asks how to create the project, mentioning SonarQube's features like repository import and Pull Request decoration. It then asks to set up a DevOps platform configuration. Four options are provided: "From Azure DevOps" (with a blue icon), "From Bitbucket" (with a blue icon), "From GitHub" (with a black icon), and "From GitLab" (with an orange icon). Each option has a "Set up global configuration" link below it. At the bottom, there's a section for manual creation with a "Manually" button and a double-angle bracket icon.

Click on manually and create the project by giving in the values for the asked fields

The screenshot shows the "Create a project" form. At the top, a message says "There's a new version of SonarQube available. Update to enjoy the latest features." The main title is "Create a project". Below it, a note says "All fields marked with * are required". The first field is "Project display name *", which has "AdopsAnalysis" entered and a green checkmark indicating it's valid. A note below says "Up to 255 characters. Some scanners might override the value you provide.". The second field is "Project key *", which also has "AdopsAnalysis" entered and a green checkmark. A note below says "The project key is a unique identifier for your project. It may contain up to 400 characters. Allowed characters are alphanumeric, '-' (dash), '_' (underscore), '.' (period) and ':' (colon), with at least one non-digit.". At the bottom, there's a "Set Up" button.

It will look something like this after clicking set up, now in order to analyze the code, click on locally

The screenshot shows the SonarQube interface for the 'AdopsAnalysis' project. At the top, there's a banner about a new version of SonarQube available. Below it, the project details ('AdopsAnalysis', 'master') are shown. The 'Overview' tab is selected. A section titled 'How do you want to analyze your repository?' offers integration with various CI systems: Jenkins, GitHub Actions, Bitbucket Pipelines, GitLab CI, Azure Pipelines, and Other CI. Below this, a section for local analysis is shown, featuring an icon of a computer monitor with arrows and the text 'Locally'.

Click on generate a token

The screenshot shows the 'Provide a token' step in the project setup process. It asks for a token name ('Analyze "AdopsAnalysis"') and an expiration period ('30 days'). A 'Generate' button is present. A note below explains that the token identifies the user performing the analysis and can be revoked if compromised. The next step, 'Run analysis on your project', is partially visible below.

The following output will be received

sonarcube Projects Issues Rules Quality Profiles Quality Gates Administration

There's a new version of SonarQube available. Update to enjoy the latest updates and features. Learn More

AdopsAnalysis master

Overview Issues Security Hotspots Measures Code Activity Project Settings Project Information

Analyze your project
We initialized your project on SonarQube, now it's up to you to launch analyses!

1 Provide a token
Analyze "AdopsAnalysis": **sqp_cd7da6a14e55458124f69a2aabf626c70800c7f5**

The token is used to identify you when an analysis is performed. If it has been compromised, you can revoke it at any point in time in your user account.

Continue

2 Run analysis on your project

After the token is copied and clicked on continue, we need to enter the following

sonarcube Projects Issues Rules Quality Profiles Quality Gates Administration

There's a new version of SonarQube available. Update to enjoy the latest updates and features. Learn More

AdopsAnalysis master

Overview Issues Security Hotspots Measures Code Activity Project Settings Project Information

Analyze your project
We initialized your project on SonarQube, now it's up to you to launch analyses!

1 Provide a token Analyze "AdopsAnalysis":**sqp_cd7da6a14e55458124f69a2aabf626c70800c7f5**

2 Run analysis on your project

What option best describes your build?
 Maven Gradle .NET Other (for JS, TS, Go, Python, PHP, ...)

Execute the Scanner for Maven
Running a SonarQube analysis with Maven is straightforward. You just need to run the following command in your project's folder.

```
mvn clean verify sonar:sonar \
-Dsonar.projectKey=AdopsAnalysis \
-Dsonar.host.url=http://localhost:9000 \
-Dsonar.login=sqp_cd7da6a14e55458124f69a2aabf626c70800c7f5
```

Please visit the [official documentation of the Scanner for Maven](#) for more details.

Is my analysis done? If your analysis is successful, this page will automatically refresh in a few moments.
You can set up Pull Request Decoration under the project settings. To set up analysis with your favorite CI tool, see the tutorials.
Check these useful links while you wait: [Branch Analysis](#), [Pull Request Analysis](#).

The code that we received of sonar-scanner has to be copied and then go into the directory where our project is and paste the sonar-scanner code (project : <https://github.com/devopshint/java-app-with-maven>)

```
(kali㉿kali)-[~]
$ cd Documents

(kali㉿kali)-[~/Documents]
$ sudo su
[sudo] password for kali:
(root㉿kali)-[/home/kali/Documents]
# git clone https://github.com/devopshint/java-app-with-maven.git
Cloning into 'java-app-with-maven'...
remote: Enumerating objects: 20, done.
remote: Counting objects: 100% (2/2), done.
remote: Compressing objects: 100% (2/2), done.
remote: Total 20 (delta 0), reused 0 (delta 0), pack-reused 18
Receiving objects: 100% (20/20), done.

(root㉿kali)-[/home/kali/Documents]
# ls
Devops Eclipseworkspace Java java-app-with-maven Selenium

(root㉿kali)-[/home/kali/Documents]
# cd java-app-with-maven
```

```
(root㉿kali)-[/home/kali/Documents/java-app-with-maven]
# ls
my-app README.md

(root㉿kali)-[/home/kali/Documents/java-app-with-maven]
# cd my-app

(root㉿kali)-[/home/kali/Documents/java-app-with-maven/my-app]
# mvn clean verify sonar:sonar \
-Dsonar.projectKey=AdopsAnalysis \
-Dsonar.host.url=http://localhost:9000 \
-Dsonar.login=sqp_cd7da6a14e55458124fe9a2aabf626c70800c7fs
[INFO] Scanning for projects...
[INFO] Downloading from central: https://repo.maven.apache.org/maven2/org/apache/maven/plugins/maven-clean-plugin/3.1.0/maven-clean-plugin-3.1.0.pom
[INFO] Downloaded from central: https://repo.maven.apache.org/maven2/org/apache/maven/plugins/maven-clean-plugin/3.1.0/maven-clean-plugin-3.1.0.pom (5.2 kB at 4.0 kB/s)
[INFO] Downloading from central: https://repo.maven.apache.org/maven2/org/apache/maven/plugins/maven-plugins/31/maven-plugins-31.pom
[INFO] Downloaded from central: https://repo.maven.apache.org/maven/plugins/maven-plugins/31/maven-plugins-31.pom (10 kB at 27 kB/s)
[INFO] Downloading from central: https://repo.maven.apache.org/maven/maven-parent/31/maven-parent-31.pom
[INFO] Downloaded from central: https://repo.maven.apache.org/maven2/org/apache/maven/maven-parent/31/maven-parent-31.pom (43 kB at 71 kB/s)
[INFO] Downloading from central: https://repo.maven.apache.org/maven2/org/apache/apache/19/apache-19.pom
[INFO] Downloaded from central: https://repo.maven.apache.org/maven2/org/apache/apache/19/apache-19.pom (15 kB at 39 kB/s)
[INFO] Downloading from central: https://repo.maven.apache.org/maven2/org/apache/maven/plugins/maven-clean-plugin/3.1.0/maven-clean-plugin-3.1.0.jar
[INFO] Downloaded from central: https://repo.maven.apache.org/maven2/org/apache/maven/plugins/maven-clean-plugin/3.1.0/maven-clean-plugin-3.1.0.jar (30 kB at 81 kB/s)
[INFO] Downloading from central: https://repo.maven.apache.org/maven2/org/apache/maven/plugins/maven-resources-plugin/3.0.2/maven-resources-plugin-3.0.2.pom
[INFO] Downloaded from central: https://repo.maven.apache.org/maven2/org/apache/maven/plugins/maven-resources-plugin/3.0.2/maven-resources-plugin-3.0.2.pom (7.1 kB at 18 kB/s)
[INFO] Downloading from central: https://repo.maven.apache.org/maven2/org/apache/maven/plugins/30/maven-plugins-30.pom
[INFO] Downloaded from central: https://repo.maven.apache.org/maven2/org/apache/maven/plugins/30/maven-plugins-30.pom (10 kB at 25 kB/s)
```

After it is run, the output is successful

```
[INFO] Sensor VB.NET Properties [vbnet] (done) | time=0ms
[INFO] ----- Run sensors on project
[INFO] Sensor Analysis Warnings import [csharp]
[INFO] Sensor Analysis Warnings import [csharp] (done) | time=1ms
[INFO] Sensor Zero Coverage Sensor
[INFO] Sensor Zero Coverage Sensor (done) | time=6ms
[INFO] Sensor Java CPD Block Indexer
[INFO] Sensor Java CPD Block Indexer (done) | time=8ms
[INFO] SCM Publisher SCM provider for this project is: git
[INFO] SCM Publisher 3 source files to be analyzed
[INFO] SCM Publisher 3/3 source files have been analyzed (done) | time=163ms
[INFO] CPD Executor 1 file had no CPD blocks
[INFO] CPD Executor Calculating CPD for 0 files
[INFO] CPD Executor CPD calculation finished (done) | time=0ms
[INFO] Analysis report generated in 80ms, dir size=124.2 kB
[INFO] Analysis report compressed in 8ms, zip size=19.9 kB
[INFO] Analysis report uploaded in 80ms
[INFO] ANALYSIS SUCCESSFUL, you can find the results at: http://localhost:9000/dashboard?id=AdopsAnalysis
[INFO] Note that you will be able to access the updated dashboard once the server has processed the submitted analysis report
[INFO] More about the report processing at http://localhost:9000/api/ce/task?id=AYQ47zny_F50yC7zKzSM
[INFO] Analysis total time: 4.251 s
[INFO] -----
[INFO] BUILD SUCCESS
[INFO] -----
[INFO] Total time: 03:11 min
[INFO] Finished at: 2022-11-02T20:51:52+05:30
[INFO] -----
```

Now if localhost:9000 is opened, we will see the following

The screenshot shows the SonarQube dashboard for the 'my-app' project. At the top, there's a message about a new version available. Below that, the project details show 'my-app' in 'master' branch, last updated on November 2, 2022, at 8:51 PM, with a version of 1.0-SNAPSHOT. The 'Overview' tab is selected. On the left, a green box indicates 'Passed' quality gate status with 'All conditions passed.' In the center, under 'MEASURES', there are several cards: 'Overall Code' (0 bugs, Reliability A), 'New Code' (0 vulnerabilities, Security A), 'Security Hotspots' (0 hotspots, Security Review A), 'Debt' (10min), 'Code Smells' (2), 'Maintainability' (A), 'Coverage' (0.0%, 2 lines to cover, 1 unit test), and 'Duplications' (0.0%, 73 lines, 0 duplicated blocks).

The screenshot shows the SonarQube interface for the project 'my-app' on the 'Issues' tab. A message at the top indicates a new version is available. The main area displays two issues: one in 'pom.xml' and one in 'src/main/java/com/mycompany/app/App.java'. The first issue is a 'Code Smell' (Major) with a fix comment. The second issue is a 'Code Smell' (Major) related to System.out/System.err usage. A sidebar on the left provides filtering options for issues based on various criteria like Type, Severity, and Scope.

The screenshot shows the SonarCloud interface for the project 'my-app'. It displays a summary of the analysis results, including counts for Bugs, Vulnerabilities, Hotspots Reviewed, Code Smells, Coverage, Duplications, and Lines of code. The coverage is shown as 0.0% with a red circle icon. The interface includes a search bar, project count, and sorting/filtering options.

Conclusion:

The analysis is performed and displayed in sonarqube successfully

ADVANCE DEVOPS LAB
Name : Ravi Pandey
Roll no: 43
Exp - 09

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

Theory : Continuous monitoring is a process of constant detecting, reporting, and responding to risks and events within an IT system.

This process is a vital DevOps security practice and has multiple goals:

- Provide real-time insight into system performance.
 - Offer feedback on the overall health and security of IT infrastructure.
 - Enhance visibility across IT operations and the DevOps pipeline.
 - Identify the cause of incidents and apply mitigation before the problem results in downtime or a data breach

Output :

STEP 1 : Install Prerequisite Packages

```
root@Aloha:~# sudo apt install -y autoconf bc gawk dc build-essential gcc libc6 make wget unzip apache2
Reading package lists... Done
Building dependency tree
Reading state information... Done
bc is already the newest version (1.07.1-2build1).
bc set to manually installed.
gawk is already the newest version (1:5.0.1+dfsg-1).
gawk set to manually installed.
gcc is already the newest version (4:9.3.0-1ubuntu2).
gcc set to manually installed.
make is already the newest version (4.2.1-1.2).
make set to manually installed.
unzip is already the newest version (6.0-25ubuntu1).
unzip set to manually installed.
build-essential is already the newest version (12.8ubuntu1.1).
build-essential set to manually installed.
libc6 is already the newest version (2.31-0ubuntu9.9).
libc6 set to manually installed.
wget is already the newest version (1.20.3-1ubuntu2).
wget set to manually installed.
The following package was automatically installed and is no longer required:
  libfwupdplugin1
Use 'sudo apt autoremove' to remove it.
The following additional packages will be installed:
  apache2-bin apache2-data apache2-utils automake autotools-dev libapache2-mod-php7.4 libapr1 libap
```

STEP 2 : Install Nagios core

```
root@Aloha:~# mkdir nagios
root@Aloha:~# cd nagios/
root@Aloha:~/nagios# sudo wget https://assets.nagios.com/downloads/nagioscore/releases/nagios-4.4.6.tar.gz
--2022-09-17 23:02:47-- https://assets.nagios.com/downloads/nagioscore/releases/nagios-4.4.6.tar.gz
Resolving assets.nagios.com (assets.nagios.com)... 205.251.196.107, 205.251.198.242, 205.251.193.140, ...
Connecting to assets.nagios.com (assets.nagios.com)|205.251.196.107|:443... ^Z
[1]+  Stopped                  sudo wget https://assets.nagios.com/downloads/nagioscore/releases/nagios-4.4.6.tar.gz
root@Aloha:~/nagios# sudo wget https://assets.nagios.com/downloads/nagioscore/releases/nagios-4.4.6.tar.gz
--2022-09-17 23:03:41-- https://assets.nagios.com/downloads/nagioscore/releases/nagios-4.4.6.tar.gz
Resolving assets.nagios.com (assets.nagios.com)... 45.79.49.120, 205.251.196.107, 205.251.198.242, ...
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          100%[=====] 10.81M  517KB/s   in 34s

2022-09-17 23:04:17 (325 KB/s) - 'nagios-4.4.6.tar.gz' saved [11333414/11333414]

root@Aloha:~/nagios#
```

```
root@Aloha:~/nagios# ls
nagios-4.4.6.tar.gz
root@Aloha:~/nagios# sudo tar -xvf 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/
```

```
useradd -g nagios nagios
root@Aloha:~/nagios/nagios-4.4.6# sudo ./configure --with-httpd-conf=/etc/apache2/sites-enabled
```

```
*** Configuration summary for nagios 4.4.6 2020-04-28 ***:

General Options:
-----
    Nagios executable:    nagios
    Nagios user/group:   nagios,nagios
    Command user/group:  nagios,nagios
        Event Broker:    yes
    Install ${prefix}:   /usr/local/nagios
                         ^ "nagios"           .ude/nagios
root@Aloha:~/nagios/nagios-4.4.6# sudo make all

    Check result directory: /usr/local/nagios/var/spool/checkresults
        Init directory:    /etc/init.d
    Apache conf.d directory: /etc/apache2/sites-enabled
        Mail program:     /bin/mail
        Host OS:          linux-gnu
    IOBroker Method:      epoll

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

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

STEP 3: Run the make all command to compile the program alongside the CGIs:

```
*** 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.
```

STEP 4: Create group users and install sample config files

```
root@Aloha:~/nagios/nagios-4.4.6# sudo make install-groups-users
groupadd -r nagios
useradd -g nagios nagios
root@Aloha:~/nagios/nagios-4.4.6#
```

```
root@Aloha:~/nagios/nagios-4.4.6# sudo usermod -a -G nagios www-data
root@Aloha:~/nagios/nagios-4.4.6#
```

```
root@Aloha:~/nagios/nagios-4.4.6# sudo make install
cd ./base && make install
root@Aloha:~/nagios/nagios-4.4.6# sudo make install-init
/usr/bin/install -c -m 755 -d -o root -g root /etc/init.d
/usr/bin/install -c -m 755 -o root -g root startup/default-init /etc/init.d/nagios
root@Aloha:~/nagios/nagios-4.4.6#
```



```
make[2]: Entering directory '/root/nagios/nagios-4.4.6/cgi'
/usr/bin/install -c -m 775 -o nagios -g nagios -d /usr/local/nagios/sbin
for file in *.cgi; do \
    /usr/bin/install -c -s -m 775 -o nagios -g nagios $file /usr/local/nagios/sbin; \
done
make[2]: Leaving directory '/root/nagios/nagios-4.4.6/cgi'
make[1]: Leaving directory '/root/nagios/nagios-4.4.6/cgi'
cd ./html && make install
root@Aloha:~/nagios/nagios-4.4.6#
```

```
root@Aloha:~/nagios/nagios-4.4.6# sudo make install-commandmode  
/usr/bin/install -c -m 775 -o nagios -g nagios -d /usr/local/nagios/var/rw  
chmod g+s /usr/local/nagios/var/rw  
  
*** External command directory configured ***  
  
root@Aloha:~/nagios/nagios-4.4.6#
```

```
root@Aloha:~/nagios/nagios-4.4.6# sudo make install-config  
/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/  
/usr/bin/install -c -b -m 664 -o nagios -g nagios sample-config/nagios.c  
/usr/bin/install -c -b -m 664 -o nagios -g nagios sample-config/cgi.cfg  
/usr/bin/install -c -b -m 660 -o nagios -g nagios sample-config/resource  
/usr/bin/install -c -b -m 664 -o nagios -g nagios sample-config/template  
objects/templates.cfg  
/usr/bin/install -c -b -m 664 -o nagios -g nagios sample-config/template  
objects/commands.cfg  
/usr/bin/install -c -b -m 664 -o nagios -g nagios sample-config/template  
objects/contacts.cfg  
/usr/bin/install -c -b -m 664 -o nagios -g nagios sample-config/template  
objects/timeperiods.cfg  
/usr/bin/install -c -b -m 664 -o nagios -g nagios sample-config/template  
objects/localhost.cfg  
/usr/bin/install -c -b -m 664 -o nagios -g nagios sample-config/template  
objects/windows.cfg  
/usr/bin/install -c -b -m 664 -o nagios -g nagios sample-config/template  
objects/printer.cfg  
/usr/bin/install -c -b -m 664 -o nagios -g nagios sample-config/template  
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.
```

STEP 5: Set up Apache and Nagios UI

```
root@Aloha:~/nagios/nagios-4.4.6# sudo make install-webconf
/usr/bin/install -c -m 644 sample-config/httpd.conf /etc/apache2/sites-enabled/nagios.conf
if [ 0 -eq 1 ]; then \
    ln -s /etc/apache2/sites-enabled/nagios.conf /etc/apache2/sites-enabled/nagios.conf; \
fi
*** Nagios/Apache conf file installed ***
root@Aloha:~/nagios/nagios-4.4.6#
```

```
root@Aloha:~/nagios/nagios-4.4.6# sudo a2enmod rewrite cgi
Enabling module rewrite.
Enabling module cgi.
To activate the new configuration, you need to run:
  service apache2 restart
root@Aloha:~/nagios/nagios-4.4.6#
```

STEP 6: Create nagios user and set password

```
root@Aloha:~/nagios/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
root@Aloha:~/nagios/nagios-4.4.6#
```

Step 7: Install nagios plugin

```
root@Aloha:~/nagios# sudo apt install monitoring-plugins nagios-nrpe-plugin -y
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following package was automatically installed and is no longer required:
  libfwupdplugin1
Use 'sudo apt autoremove' to remove it.
The following additional packages will be installed:
  libdbi1 libradcli4 libtirpc-common libtirpc3 monitoring-plugins-basic monitoring-plugins-common
  monitoring-plugins-standard python3-crypto python3-gpg python3-samba python3-tdb rpcbind samba-common
  samba-common-bin samba-dsdb-modules smbclient
Suggested packages:
  icinga | icinga2 nagios-plugins-contrib fping postfix | sendmail-bin | exim4-daemon-heavy | exim4-daemon-light
  qstat heimdal-clients python3-markdown python3-dnspython cifs-utils
The following NEW packages will be installed:
  libdbi1 libradcli4 libtirpc-common libtirpc3 monitoring-plugins monitoring-plugins-basic
  monitoring-plugins-common monitoring-plugins-standard nagios-nrpe-plugin python3-crypto python3-gpg python3-samba
  python3-tdb rpcbind samba-common samba-common-bin samba-dsdb-modules smbclient
0 upgraded, 18 newly installed, 0 to remove and 23 not upgraded.
Need to get 4992 kB of archives.
After this operation, 33.4 MB of additional disk space will be used.
Get:1 http://archive.ubuntu.com/ubuntu focal-updates/main amd64 libtirpc-common all 1.2.5-1ubuntu0.1 [7712 B]
Get:2 http://archive.ubuntu.com/ubuntu focal-updates/main amd64 libtirpc3 amd64 1.2.5-1ubuntu0.1 [77.9 kB]
Get:3 http://archive.ubuntu.com/ubuntu focal/main amd64 rpcbind amd64 1.2.5-8 [42.8 kB]
Get:4 http://archive.ubuntu.com/ubuntu focal-updates/main amd64 samba-common all 2:4.13.17~dfsg-0ubuntu1.20.04.1 [69.1 kB]
```

```
root@Aloha:/usr/local/nagios/etc# ll
total 84
drwxrwxr-x 3 nagios nagios 4096 Sep 17 23:21 .
drwxr-xr-x 8 root root 4096 Sep 17 23:17 ..
-rw-rw-r-- 1 nagios nagios 13710 Sep 17 23:17 cgi.cfg
-rw-r--r-- 1 root root 50 Sep 17 23:22 htpasswd.users
-rw-rw-r-- 1 nagios nagios 45843 Sep 17 23:17 nagios.cfg
drwxrwxr-x 2 nagios nagios 4096 Sep 17 23:17 objects/
-rw-rw---- 1 nagios nagios 1312 Sep 17 23:17 resource.cfg
root@Aloha:/usr/local/nagios/etc#
```

```
50 # directive as shown below:
51
52 cfg_dir=/usr/local/nagios/etc/servers
53 cfg_dir=/usr/local/nagios/etc/printers
54 cfg_dir=/usr/local/nagios/etc/switches
55 cfg_dir=/usr/local/nagios/etc/routers
56
57
58
```

```
# 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-
bove)
    alias             Nagios Admin        ; Full name of user
    email             rshanker084@gmail.com ; <***** CHANGE THIS TO YOUR EMAIL ADDI
}

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


```

```
root@Aloha: /usr/local/nagios/etc
GNU nano 4.8                                     resource.cfg
#####
#
# RESOURCE.CFG - Sample Resource File for Nagios 4.4.6
#
#
# You can define $USERx$ macros in this file, which can in turn be used
# in command definitions in your host config file(s). $USERx$ macros
# are useful for storing sensitive information such as usernames, passwords
# etc. They are also handy for specifying the path to plugins and
# event handlers - if you decide to move the plugins or event handlers
# to a different directory in the future, you can just update one or two
# $USERx$ macros, instead of modifying a lot of command definitions.
#
# The CGIs will not attempt to read the contents of resource files, so
# you can set restrictive permissions (600 or 660) on them.
#
# Nagios supports up to 256 $USERx$ macros ($USER1$ through $USER256$)
#
# Resource files may also be used to store configuration directives for
# external data sources like MySQL...
#
#####
#
# Sets $USER1$ to be the path to the plugins
#$USER1$=/usr/local/nagios/libexec
$USER1$=/usr/lib/nagios/plugins
# Sets $USER2$ to be the path to event handlers
#$USER2$=/usr/local/nagios/libexec/eventhandlers

# Store some usernames and passwords (hidden from the CGIs)
#$USER3$=someuser
#$USER4$=somepassword
```

```
#####
#
# SAMPLE PERFORMANCE DATA COMMANDS
#
# These are sample performance data commands that can be used to send performance
# data output to two text files (one for hosts, another for services). If you
# plan on simply writing performance data out to a file, consider using the
# host_perfdata_file and service_perfdata_file options in the main config file.
#
#####

define command {
    command_name      process-host-perfdata
    command_line      /usr/bin/printf "%b" "$LASTHOSTCHECK$\t$HOSTNAME$\t$HOSTSTATE$\t$HOSTADDRESS$\t$HOSTPERF$"
}

define command {
    command_name      process-service-perfdata
    command_line      /usr/bin/printf "%b" "$LASTSERVICECHECK$\t$HOSTNAME$\t$SERVICEDESC$\t$SERVICESTATE$\t$SERVICESTATUSTIME$\t$SERVICESTATUSTIME$"
}

define command{
    command_name     check_nrpe
    command_line     $USER1$/check_nrpe -H $HOSTADDRESS$ -c $ARG1$
}
```

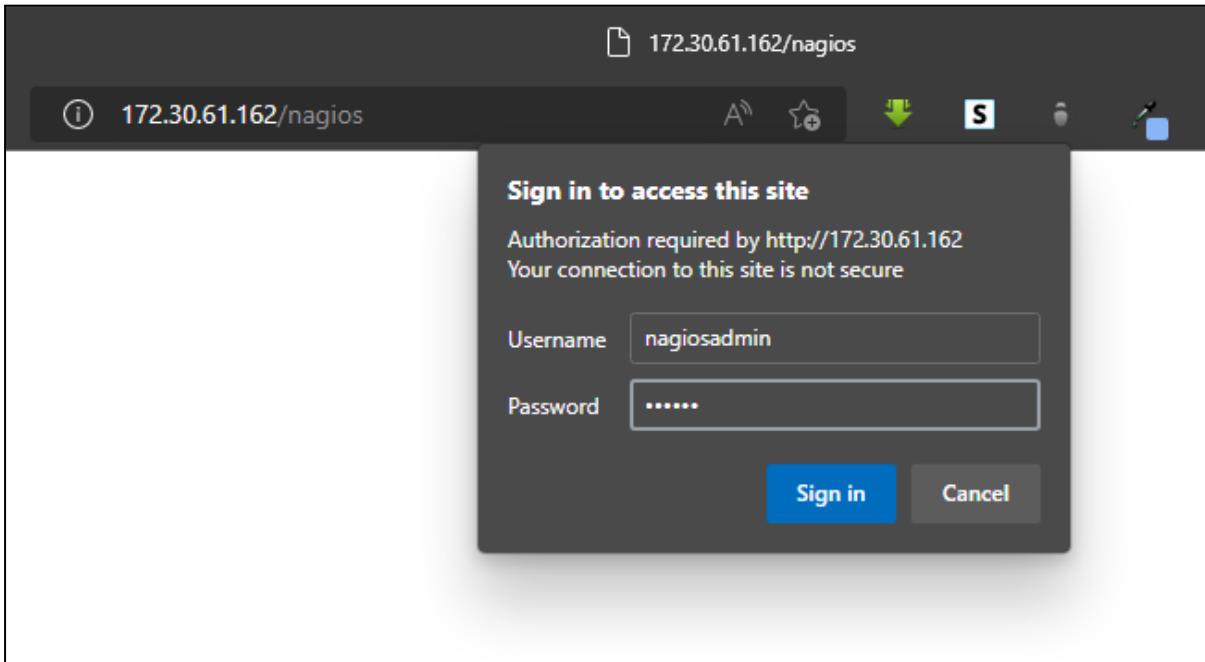
Step 8 : Start the Nagios Daemon and check status

```
root@Aloha:~# sudo systemctl restart apache2
root@Aloha:~# sudo systemctl restart nagios
root@Aloha:~# sudo systemctl enable apache2
Synchronizing state of apache2.service with SysV service script with /lib/systemd/systemd-sysv-install.
Executing: /lib/systemd/systemd-sysv-install enable apache2
root@Aloha:~# sudo systemctl enable nagios
nagios.service is not a native service, redirecting to systemd-sysv-install.
Executing: /lib/systemd/systemd-sysv-install enable nagios
```

```
root@Aloha:~# sudo systemctl status apache2
● apache2.service - The Apache HTTP Server
  Loaded: loaded (/lib/systemd/system/apache2.service; enabled; vendor preset: enabled)
  Active: active (running) since Sat 2022-09-17 23:40:55 IST; 2min 30s ago
    Docs: https://httpd.apache.org/docs/2.4/
 Main PID: 709 (apache2)
   Tasks: 6 (limit: 4314)
  Memory: 11.4M
 CGroup: /system.slice/apache2.service
         ├─709 /usr/sbin/apache2 -k start
         ├─710 /usr/sbin/apache2 -k start
         ├─711 /usr/sbin/apache2 -k start
         ├─712 /usr/sbin/apache2 -k start
         ├─713 /usr/sbin/apache2 -k start
         └─714 /usr/sbin/apache2 -k start

Sep 17 23:40:55 Aloha systemd[1]: Starting The Apache HTTP Server...
Sep 17 23:40:55 Aloha systemd[1]: Started The Apache HTTP Server.
root@Aloha:~# sudo systemctl status nagios
● nagios.service - LSB: Starts and stops the Nagios monitoring server
  Loaded: loaded (/etc/init.d/nagios; generated)
  Active: active (running) since Sat 2022-09-17 23:41:05 IST; 2min 40s ago
    Docs: man:systemd-sysv-generator(8)
   Tasks: 20 (limit: 4314)
  Memory: 6.1M
 CGroup: /system.slice/nagios.service
         ├─939 /usr/local/nagios/bin/nagios -d /usr/local/nagios/etc/nagios.cfg
         ├─940 /usr/local/nagios/bin/nagios --worker /usr/local/nagios/var/rw/nagios.qh
         ├─941 /usr/local/nagios/bin/nagios --worker /usr/local/nagios/var/rw/nagios.qh
```

Access the tool by opening the browser and navigating to the `http://server-IP/nagios` URL.

A screenshot of the Nagios Core 4.4.6 dashboard. The top right features the Nagios Core logo with a green checkmark and the text "Daemon running with PID 939". The center displays the Nagios Core logo and version information: "Nagios® Core™ Version 4.4.6" and "April 28, 2020 Check for updates". A blue banner at the bottom left says "A new version of Nagios Core is available! Visit [nagios.org](#) to download Nagios 4.4.7." On the left side, there's a sidebar with links: General (Home, Documentation), Current Status (Tactical Overview, Map (Legacy), Hosts, Services, Host Groups, Service Groups, Problems), Reports (Availability, Trends (Legacy), Alerts, Notifications, Event Log), and System. In the center, there are three download links for additional tools: "Nagios XI" (Easy Configuration Advanced Reporting), "Nagios Log Server" (Monitor and analyze logs from anywhere), and "Nagios Network Analyzer" (Real-time netflow and bandwidth analysis).

ADVANCED DEVOPS LAB

Name : Ravi Pandey

Roll no : 43

Experiment No : 10

Aim:

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

Theory:

Nagios is an open source monitoring system for computer systems. It was designed to run on the Linux operating system and can monitor devices running Linux, Windows and Unix operating systems (OSes).

Nagios runs both agent-based and agentless configurations. Independent agents are installed on any hardware or software system to collect data that is then reported back to the management server. Agentless monitoring uses existing protocols to emulate an agent. Both approaches can monitor file system usage, OS metrics, service and process states and more.

Here are the important reasons to use Nagios monitoring tool:

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

Output:

Install all plugins

```
(root@kali)-[/home/kali]
└─# sudo apt install monitoring-plugins nagios-nrpe-plugin -y
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
monitoring-plugins is already the newest version (2.3.1-1).
The following NEW packages will be installed:
  nagios-nrpe-plugin
0 upgraded, 1 newly installed, 0 to remove and 662 not upgraded.
Need to get 31.0 kB of archives.
After this operation, 85.0 kB of additional disk space will be used.
Get:1 http://http.kali.org/kali kali-rolling/main amd64 nagios-nrpe-plugin amd64 4.1.0-1+b1 [31.0 kB]
Fetched 31.0 kB in 1s (22.5 kB/s)
Selecting previously unselected package nagios-nrpe-plugin.
(Reading database ... 344263 files and directories currently installed.)
Preparing to unpack .../nagios-nrpe-plugin_4.1.0-1+b1_amd64.deb ...
Unpacking nagios-nrpe-plugin (4.1.0-1+b1) ...
Setting up nagios-nrpe-plugin (4.1.0-1+b1) ...
Scanning processes...
Scanning candidates...
Scanning processor microcode...
Scanning linux images...
```

Go and check the files that are present in the following directory

```
(root@kali)-[/home/kali/nagios]
└─# cd /usr/local/nagios/etc
23:51:23 resource.cfg

[root@kali)-[/usr/local/nagios/etc]
└─# ls
cgi.cfg  htpasswd.users  nagios.cfg  objects  resource.cfg
```

Uncomment these lines

```
# You can specify individual object config files as shown below:
cfg_file=/usr/local/nagios/etc/objects/commands.cfg
cfg_file=/usr/local/nagios/etc/objects/contacts.cfg
cfg_file=/usr/local/nagios/etc/objects/timeperiods.cfg
cfg_file=/usr/local/nagios/etc/objects/templates.cfg
```

Then create the following directories

```
[root@kali]# mkdir servers printers switches routers
```

Then add this path in resource.cfg file for \$USER1\$

```
# Sets $USER1$ to be the path to the plugins  
#$USER1$=/usr/local/nagios/libexec  
$USER1$=/usr/lib/nagios/plugins
```

Now go to objects directory

```
[root@kali]# cd objects  
noe with your email id  
[root@kali]# ls  
commands.cfg contacts.cfg localhost.cfg printer.cfg switch.cfg templates.cfg timeperiods.cfg windows.cfg
```

Open file contacts.cfg in any editor and change with your email id

```
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             rshanker084@gmail.com ; <***** CHANGE THIS TO YOUR EMAIL ADDRESS *****  
}oe with your email id
```

Now go to commands.cfg and enter the following define at the end

```
define command{  
    command_name check_nrpe  
    command_line $USER1$/check_nrpe -H $HOSTADDRESS$ -c $ARG1$  
}
```

Check and update the firewall in the following manner

```
(kali㉿kali)-[~]
└─$ sudo ufw status
Status: inactive
```

Restart and enable both apache and nagios

```
(kali㉿kali)-[~]
└─$ sudo systemctl restart apache2

(kali㉿kali)-[~]
└─$ sudo systemctl restart nagios

(kali㉿kali)-[~]
└─$ sudo systemctl enable apache2
Synchronizing state of apache2.service with SysV service script with /lib/systemd/systemd-sysv-install.
Executing: /lib/systemd/systemd-sysv-install enable apache2
Created symlink /etc/systemd/system/multi-user.target.wants/apache2.service → /lib/systemd/system/apache2.service.

(kali㉿kali)-[~]
└─$ sudo systemctl enable nagios
```

Check their status

Apache2

```
(kali㉿kali)-[~]
└─$ sudo systemctl status apache2
● apache2.service - The Apache HTTP Server
   Loaded: loaded (/lib/systemd/system/apache2.service; enabled; preset: disabled)
   Active: active (running) since Fri 2022-11-04 00:11:18 IST; 1min 47s ago
     Docs: https://httpd.apache.org/docs/2.4/
 Main PID: 14558 (apache2)
    Tasks: 7 (limit: 8666)
   Memory: 13.3M
      CPU: 35ms
     CGroup: /system.slice/apache2.service
             └─14558 /usr/sbin/apache2 -k start
                 ├─14560 /usr/sbin/apache2 -k start
                 ├─14561 /usr/sbin/apache2 -k start
                 ├─14562 /usr/sbin/apache2 -k start
                 ├─14563 /usr/sbin/apache2 -k start
                 ├─14564 /usr/sbin/apache2 -k start
                 ├─14565 /usr/sbin/apache2 -k start
                 └─14566 /usr/sbin/apache2 -k start

Nov 04 00:11:18 kali systemd[1]: Starting The Apache HTTP Server...
Nov 04 00:11:18 kali systemd[1]: Started The Apache HTTP Server.
```

nagios

```
(kali㉿kali)-[~]
└─$ sudo systemctl status nagios
● nagios.service - Nagios
   Loaded: loaded (/etc/systemd/system/nagios.service; enabled; preset: disabled)
   Active: active (running) since Fri 2022-11-04 00:11:40 IST; 1min 29s ago
     Main PID: 14595 (nagios)
        Tasks: 20 (limit: 8666)
       Memory: 5.9M
          CPU: 30ms
        CGroup: /system.slice/nagios.service
                └─14595 /usr/local/nagios/bin/nagios /usr/local/nagios/etc/nagios.cfg
                    ├─14597 /usr/local/nagios/bin/nagios --worker /usr/local/nagios/var/rw/nagios.qh
                    ├─14598 /usr/local/nagios/bin/nagios --worker /usr/local/nagios/var/rw/nagios.qh
                    ├─14599 /usr/local/nagios/bin/nagios --worker /usr/local/nagios/var/rw/nagios.qh
                    ├─14600 /usr/local/nagios/bin/nagios --worker /usr/local/nagios/var/rw/nagios.qh
                    ├─14601 /usr/local/nagios/bin/nagios --worker /usr/local/nagios/var/rw/nagios.qh
                    ├─14602 /usr/local/nagios/bin/nagios --worker /usr/local/nagios/var/rw/nagios.qh
                    ├─14603 /usr/local/nagios/bin/nagios --worker /usr/local/nagios/var/rw/nagios.qh
                    ├─14604 /usr/local/nagios/bin/nagios --worker /usr/local/nagios/var/rw/nagios.qh
                    ├─14605 /usr/local/nagios/bin/nagios --worker /usr/local/nagios/var/rw/nagios.qh
                    ├─14606 /usr/local/nagios/bin/nagios --worker /usr/local/nagios/var/rw/nagios.qh
                    ├─14607 /usr/local/nagios/bin/nagios --worker /usr/local/nagios/var/rw/nagios.qh
                    ├─14608 /usr/local/nagios/bin/nagios --worker /usr/local/nagios/var/rw/nagios.qh
                    ├─14609 /usr/local/nagios/bin/nagios --worker /usr/local/nagios/var/rw/nagios.qh
                    ├─14610 /usr/local/nagios/bin/nagios --worker /usr/local/nagios/var/rw/nagios.qh
                    ├─14611 /usr/local/nagios/bin/nagios --worker /usr/local/nagios/var/rw/nagios.qh
                    ├─14612 /usr/local/nagios/bin/nagios --worker /usr/local/nagios/var/rw/nagios.qh
                    ├─14613 /usr/local/nagios/bin/nagios --worker /usr/local/nagios/var/rw/nagios.qh
                    └─14614 /usr/local/nagios/bin/nagios --worker /usr/local/nagios/var/rw/nagios.qh
                        └─14621 /usr/local/nagios/bin/nagios /usr/local/nagios/etc/nagios.cfg
```

Now in order to do port monitoring do the following

Go into the following directories

```
(kali㉿kali)-[~]
└─$ sudo -i
(_Message from Kali developers_)
This is a minimal installation of Kali Linux, you likely
want to install supplementary tools. Learn how:
⇒ https://www.kali.org/docs/troubleshooting/common-minimum-setup/
(Run: "touch ~/.hushlogin" to hide this message)
(root㉿kali)-[~]
└─# cd /usr/local
(root㉿kali)-[/usr/local]
└─# cd nagios
(root㉿kali)-[/usr/local/nagios]
└─# cd etc
(root㉿kali)-[/usr/local/nagios/etc]
└─# ls
cgi.cfg htpasswd.users nagios.cfg objects printers resource.cfg routers servers switches
```

In servers directory, create a file called as hosts.cfg

```
[root@kali]~-[/usr/local/nagios/etc]
[root@kali]# cd servers
[root@kali]~-[/usr/local/nagios/etc/servers]
[root@kali]# nano hosts.cfg
```

Add the following ip and name according to your machine and ip address (in our case, we have used a mobile and used its wifi IP)

```
GNU nano 6.4
define host {
    use                 linux-server
    host_name           Xiaomi
    alias               phone
    address             192.168.0.103
    max_check_attempts  5
    check_period        24x7
    notification_interval 30
    notification_period 24x7
}
```

Now restart the nagios service

```
[root@kali]~-[/usr/local/nagios/etc/servers]
[root@kali]# service nagios start
[root@kali]# service nagios start
[root@kali]~-[/usr/local/nagios/etc/servers]
[root@kali]# service nagios restart
```

Open (ip_addr/nagios)

Put in your credentials, you will see the following screen

The screenshot shows the Nagios Core 4.4.6 dashboard. At the top right, it displays "Nagios® Core™ Version 4.4.6 April 28, 2020 Check for updates". Below this, there are three promotional cards for Nagios XI, Log Server, and Network Analyzer, each with a "Download" button. To the left, the main navigation menu includes sections like General, Current Status, Reports, and System. A central "Get Started" section provides links to monitor infrastructure, change look and feel, extend with add-ons, get support, and get certified. On the right, a "Quick Links" sidebar lists Nagios Library, Labs, Exchange, Support, and the official project page.

Now, go into hosts, we will be able to monitor our client

The screenshot shows the "Host Status Details For All Host Groups" page. It displays two hosts: "Kali" and "Kali2", both in "UP" status. The "Status Information" column indicates PING OK - Packet loss = 0%, RTA = 181.96 ms for Kali and PING OK - Packet loss = 0%, RTA = 0.04 ms for Kali2. Above the table, a "Current Network Status" box shows last update information and a host status totals bar with 0 Down, 0 Unreachable, and 0 Pending hosts. The left sidebar contains the same navigation menu as the main dashboard.

Conclusion:

Thus port monitoring is completed successfully.

Reference:

<https://www.youtube.com/watch?v=UMHgRnPXoEw>

https://www.youtube.com/watch?v=6T_RCywnLB8&list=PLvoGk4CSiH0sfH21iVgaPKx3DII8dJYte&index=4

ADVANCED DEVOPS LAB

Name : Ravi Pandey

Roll no : 43

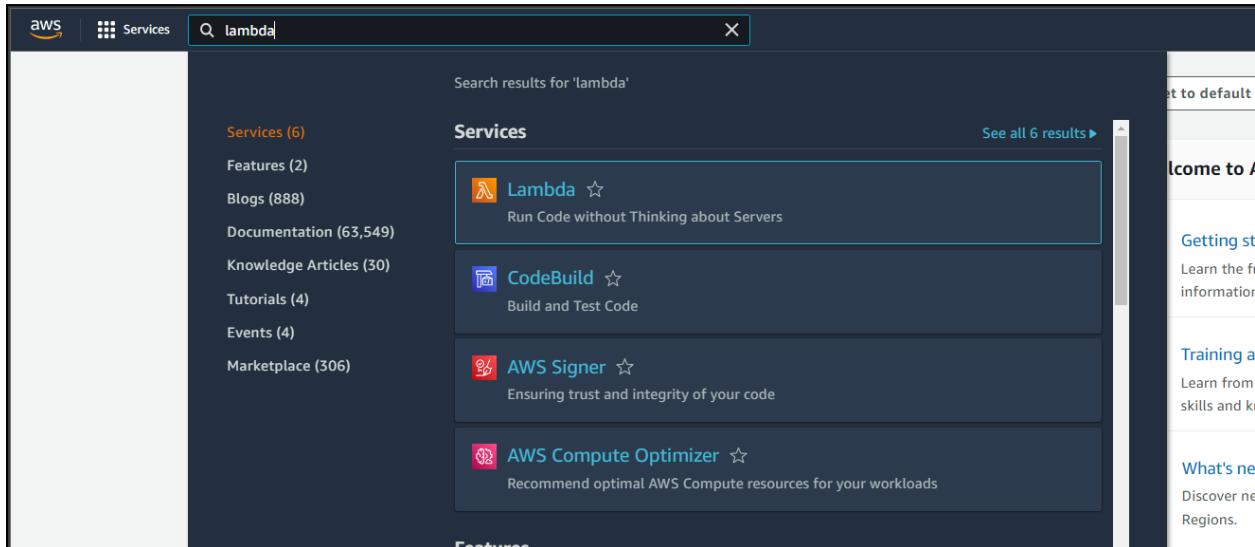
Experiment No : 11

Aim:

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

Output:

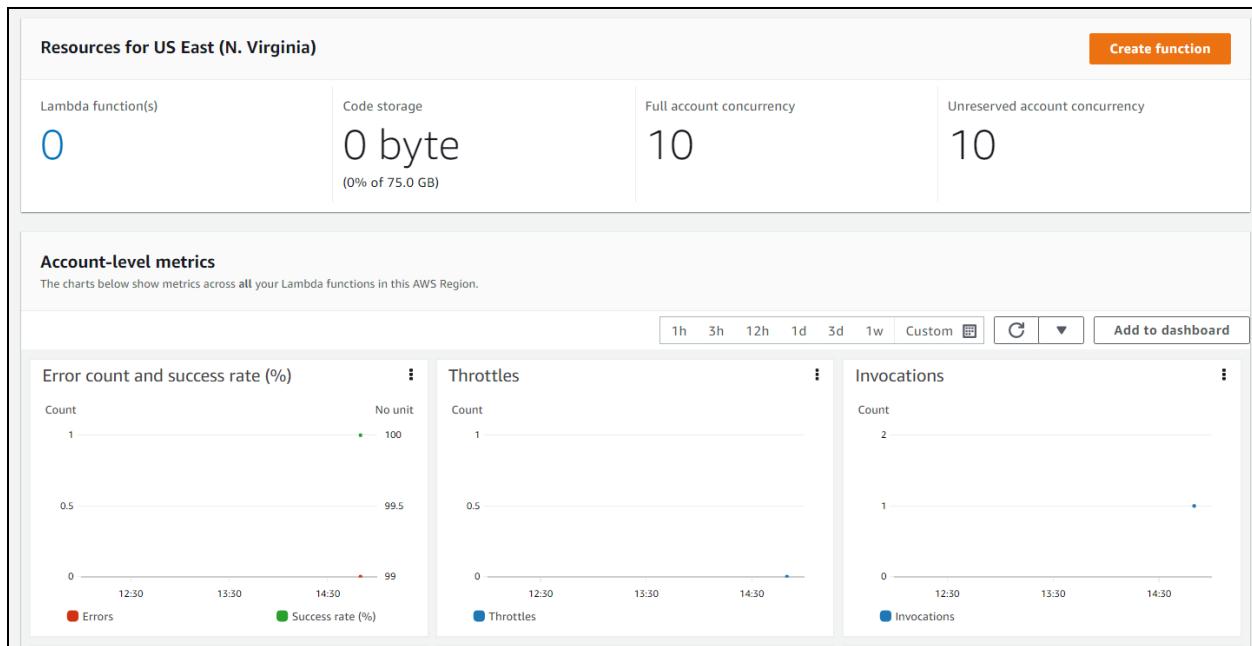
Step 1 : Enter The lambda console



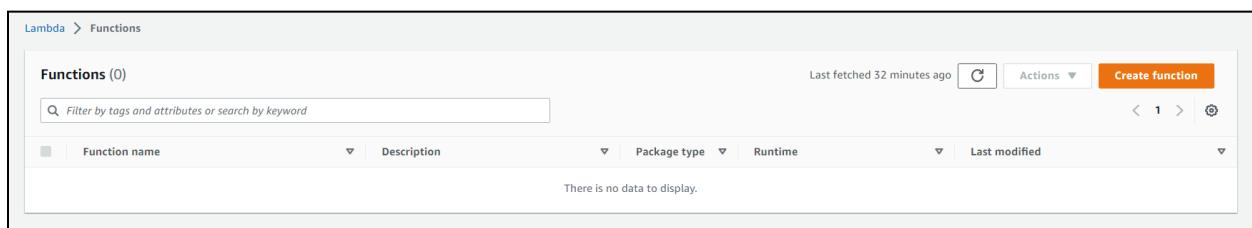
Step 2 : Select a lambda blueprint

Blueprints provide example code to do some minimal processing. Most blueprints process events from specific event sources, such as Amazon S3, Amazon DynamoDB, or a custom application.

- In the AWS Lambda console, choose Create function.



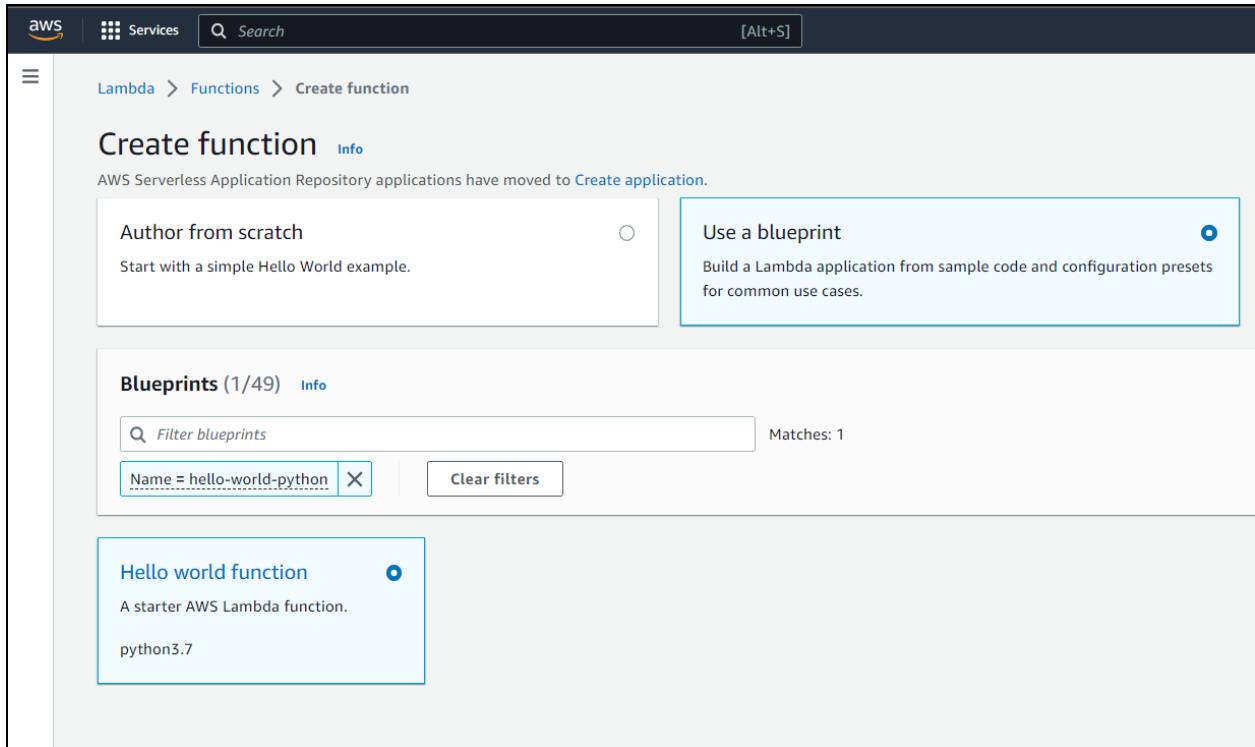
- In functions, the create function is present



- Select use a blueprint.

- In the Filter box, enter hello-world-python and select the hello-world-python blueprint.

e. Then choose Configure.



Step 3 : Configure and create your lambda function

A Lambda function consists of code you provide, associated dependencies, and configuration. The configuration information you provide includes the compute resources you want to allocate (for example, memory), execution timeout, and an IAM role that AWS Lambda can assume to execute your Lambda function on your behalf.

a. You will now enter Basic information about your Lambda function.

Basic information:

Name: You can name your Lambda function here. For this tutorial, enter hello-world-python.

Role: You will create an IAM role (referred to as the execution role) with the necessary permissions that AWS Lambda can assume to invoke your Lambda

function on your behalf. Select Create a new role from AWS policy templates.

Role name: type lambda_basic_execution.

Lambda function code:

b. Go to the bottom of the page and choose the Create function.

The screenshot shows the 'Basic information' section of the AWS Lambda 'Create function' configuration page. The 'Function name' field contains 'hello-python'. Under 'Execution role', the 'Create a new role from AWS policy templates' option is selected. A note below states: 'Role creation might take a few minutes. Please do not delete the role or edit the trust or permissions policies in this role.' The 'Role name' field is set to 'Adops_lambda'. The 'Policy templates - optional' section is present but empty. A 'Create' button is visible at the bottom right.

Lambda > Functions > Create function > Configure blueprint hello-world-python

Basic information [Info](#)

Function name

hello-python

Execution role

Choose a role that defines the permissions of your function. To create a custom role, go to the [IAM console](#).

Create a new role with basic Lambda permissions

Use an existing role

Create a new role from AWS policy templates

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

Role name

Enter a name for your new role.

Adops_lambda

Use only letters, numbers, hyphens, or underscores with no spaces.

Policy templates - *optional* [Info](#)

Choose one or more policy templates.

▼ C

Lambda function code

Code is preconfigured by the chosen blueprint. You can configure it after you create the function. [Learn more about deploying Lambda functions.](#)

ⓘ This function contains external libraries. X

Runtime	Architecture
Python 3.7	x86_64

```
1 import json
2
3 print('Loading function')
4
5
6 * def lambda_handler(event, context):
7     #print("Received event: " + json.dumps(event, indent=2))
8     print("value1 = " + event['key1'])
9     print("value2 = " + event['key2'])
10    print("value3 = " + event['key3'])
11    return event['key1'] # Echo back the first key value
12    #raise Exception('Something went wrong')
13
```

Cancel Create function

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

Lambda > Functions > hello-python

hello-python

Function overview Info

 hello-python	Description A starter AWS Lambda function.
 Layers (0)	Last modified 27 seconds ago
+ Add trigger	+ Add destination
	Function ARN arn:aws:lambda:us-east-1:696940912922:function:hello-python
	Function URL Info

c. Runtime: Currently, you can author your Lambda function code in Java, Node.js, C#, Go, or Python. For this tutorial, use Python 3.7 as the runtime.

d. Handler: You can specify a handler (a method/function in your code) where AWS Lambda can begin executing your code. AWS Lambda provides event data as input to this handler, which processes the event.

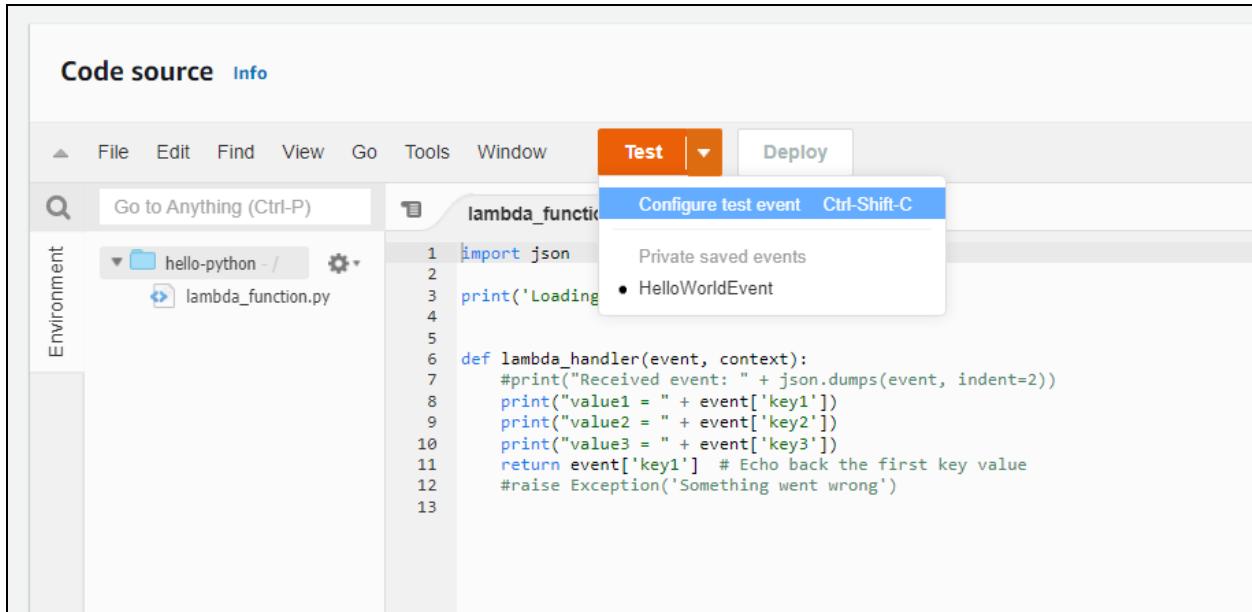
In this example, Lambda identifies this from the code sample and this should be pre-populated with `lambda_function.lambda_handler`.

The screenshot shows the 'Code properties' section of the AWS Lambda function configuration. It includes three main sections: 'Code properties', 'Runtime settings', and 'Layers'.
- In 'Code properties', there is information about the package size (443.0 byte), SHA256 hash (fAbN/OSYMyE4KT5AqnIcbAZ7EE4R5eznxmLw0+jbo=), and last modified time (November 3, 2022 at 08:17 PM GMT+5:30).
- In 'Runtime settings', the runtime is set to Python 3.7, the handler is lambda_function.lambda_handler, and the architecture is x86_64.
- In 'Layers', there is no data to display, indicated by the message 'There is no data to display.'

Step 4 : Invoke lambda function and verify results

The console shows the hello-world-python Lambda function. You can now test the function, verify results, and review the logs.

- a. Select Configure Test Event from the drop-down menu called Test.



- b. The editor pops up so you can enter an event to test your function.
- Select Create new event.
 - Type in an event name like HelloWorldEvent.
 - Retain default setting of Private for Event sharing settings.
 - Choose hello-world from the template list.
 - You can change the values in the sample JSON, but don't change the event structure.

For now, replace value1 with hello world!. Select Create.

Configure test event

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.

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

Test event action

Create new event Edit saved event

Event name

HelloLambda

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

Event sharing settings

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

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

Template - *optional*

hello-world

Event JSON

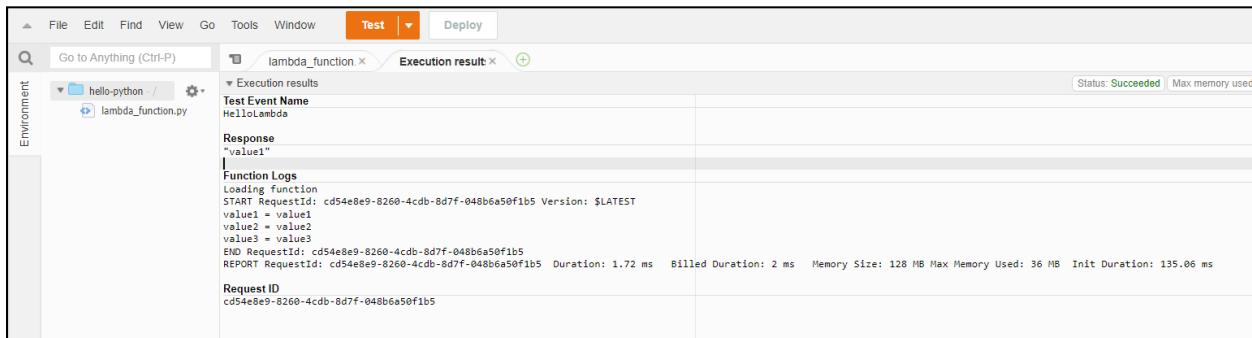
Format JSON

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

c. Choose Test.

d. Upon successful execution, view the results in the console:

- The Execution results tab verifies that the execution succeeded.
- The Function Logs section will show the logs generated by the Lambda function execution as well as key information reported in the Log output.



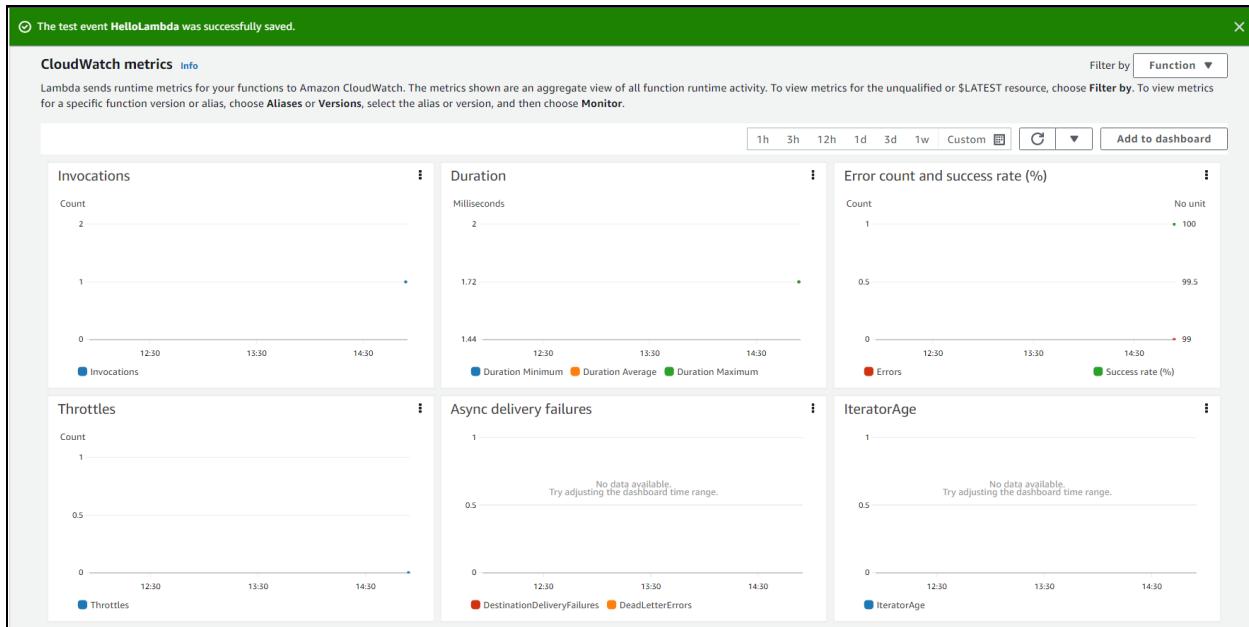
The screenshot shows the AWS Lambda Test interface. At the top, there's a navigation bar with File, Edit, Find, View, Go, Tools, Window, a Test button, and Deploy. Below the navigation bar, there's a search bar labeled 'Go to Anything (Ctrl-P)'. The main area has tabs for 'Execution results' and 'Function Logs'. Under 'Execution results', it shows a test event named 'HelloLambda' and a response object containing 'value1'. Under 'Function Logs', it displays the log output for a successful execution. The log includes details like RequestId, Duration, Billed Duration, Memory Size, Max Memory Used, and Init Duration. The status is listed as 'Succeeded'.

Step 5 : Monitor your metrics

AWS Lambda automatically monitors Lambda functions and reports metrics through Amazon CloudWatch. To help you monitor your code as it executes, Lambda automatically tracks the number of requests, the latency per request, and the number of requests resulting in an error and publishes the associated metrics.

- Invoke the Lambda function a few more times by repeatedly choosing the Test button. This will generate the metrics that can be viewed in the next step.
- Select the Monitor tab to view the results.
- Scroll down to view the metrics for your Lambda function. Lambda metrics are reported through Amazon CloudWatch. You can leverage these metrics to set custom alarms. For more information about CloudWatch, see the Amazon CloudWatch Developer Guide.

The Monitoring tab will show seven CloudWatch metrics: Invocations, Duration, Error count and success rate (%), Throttles, Async delivery failures, IteratorAge, and Concurrent executions.



Step 6 : Delete the lambda function

- Select the Actions button and select Delete function.

The Lambda function overview page for hello-python shows the following details:

- Function name:** hello-python
- Description:** A starter AWS Lambda function.
- Last modified:** 10 minutes ago
- Function ARN:** arn:aws:lambda:us-east-1:696940912922:function:hello-python
- Function URL:** Info

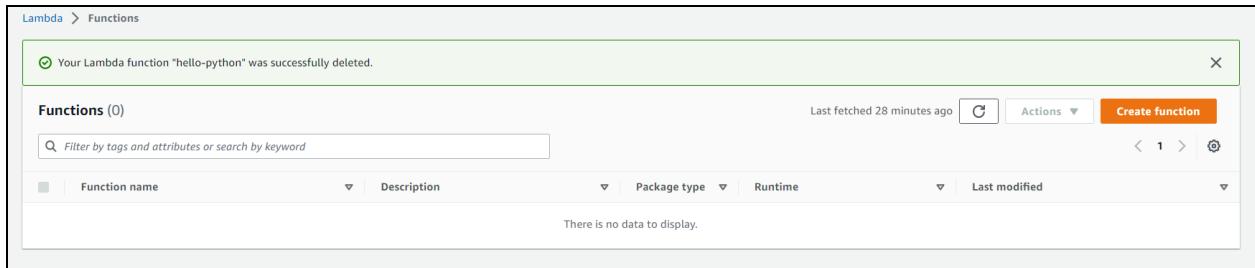
The Actions menu is open, showing options: Throttle, Copy ARN, Actions ▾, Publish new version, Create alias, Export function, Capabilities, Deploy to Lambda@Edge, and Delete function. Delete function is highlighted.

The delete function confirmation dialog displays the following message:

Delete function hello-python

⚠️ Deleting a function permanently removes the function code. The related logs, roles, test event schemas, and triggers are retained in your account.

Cancel **Delete**



Conclusion:

First AWS Lambda function is created successfully.

Reference:

[Run a Serverless "Hello, World!" with AWS Lambda](#)

ADVANCED DEVOPS LAB

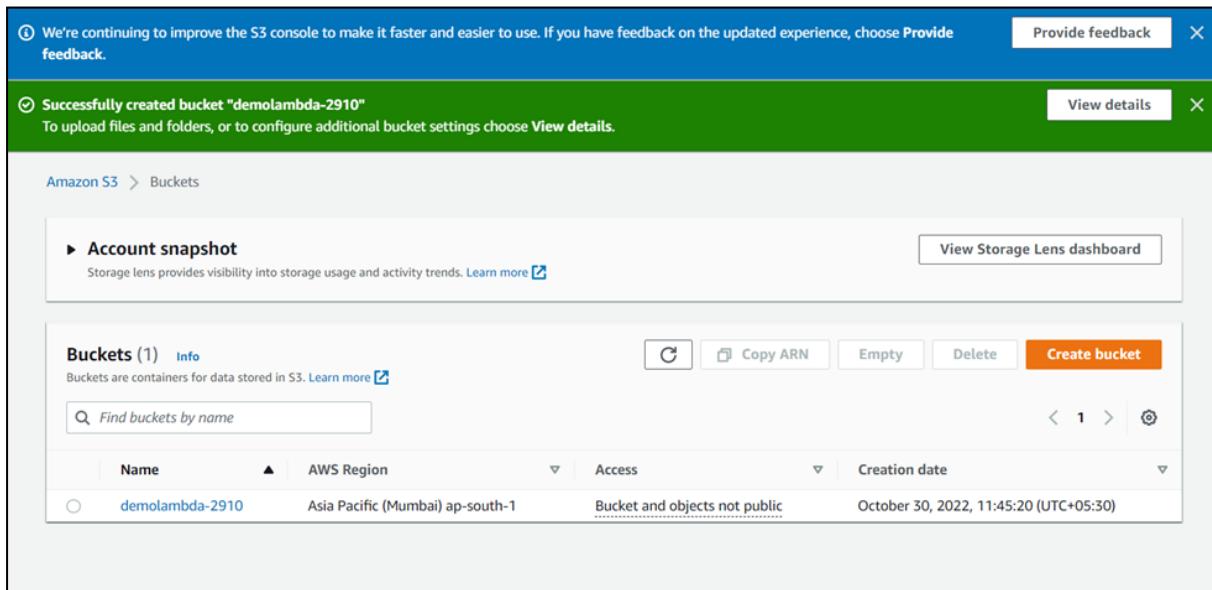
Name : Ravi Pandey
Roll no : 43
Experiment No : 12

To start using AWS Lambda with Amazon S3, we need the following –

- Create S3 Bucket
- Create role which has permission to work with s3 and lambda
- Create a lambda function and add s3 as the trigger.

Creating S3 Bucket

Step 1



The screenshot shows the AWS S3 console interface. At the top, there is a blue header bar with a message about improving the S3 console and a 'Provide feedback' button. Below this is a green success banner stating 'Successfully created bucket "demolambda-2910"' and a 'View details' button. The main content area shows an 'Account snapshot' section with a 'View Storage Lens dashboard' button. Below it is a table titled 'Buckets (1)'. The table has a 'Find buckets by name' search bar at the top. It contains one row with the following data:

Name	AWS Region	Access	Creation date
demolambda-2910	Asia Pacific (Mumbai) ap-south-1	Bucket and objects not public	October 30, 2022, 11:45:20 (UTC+05:30)

Create Role that Works with S3 and Lambda

To create role that works with S3 and Lambda, please follow the Steps given below –

Step 1

Go to AWS services and select IAM as shown below –

The screenshot shows the AWS IAM dashboard. On the left, there are two red warning boxes: one for 'Add MFA for root user' and another for 'Deactivate or delete access keys for root user'. In the center, it displays 'IAM resources' with counts: 0 User groups, 1 User, 3 Roles, 0 Policies, and 0 Identity providers. On the right, the 'AWS Account' section shows the Account ID (625453636107), Account Alias (625453636107), and a 'Create' link for sign-in. Below that is a 'Quick Links' section for 'My security credentials'.

Step 2

Now, click IAM -> Roles as shown below –

The screenshot shows the 'Roles' list page. At the top, there are 'Create role' and 'Delete role' buttons. A search bar is present at the top left. The main table lists five roles:

Role name	Description	Trusted entities
lambdaapipolicy	Allows Lambda functions to call AWS servic...	AWS service: lambda
lambdapolicyjava	Allows Lambda functions to call AWS servic...	AWS service: lambda
lambdawithdynamodb	Allows Lambda functions to call AWS servic...	AWS service: lambda
lambdawiths3	Allows Lambda functions to call AWS servic...	AWS service: lambda
roleforlambdatesting	Allows Lambda functions to call AWS servic...	AWS service: lambda

Step 3

Add the permission from below and click Review.

The screenshot shows a list of AWS IAM policies. The columns are 'Policy name', 'Attachments', and 'Description'. The 'Policy name' column lists various policies like 'AdministratorAccess', 'AlexaForBusinessDeviceSetup', etc. The 'Attachments' column shows the count of attached resources for each policy. The 'Description' column provides a brief summary of each policy's function. A search bar at the top right allows filtering by policy type. At the bottom, there are buttons for 'Cancel', 'Previous', and 'Next: Review'.

	Policy name	Attachments	Description
<input type="checkbox"/>	AdministratorAccess	0	Provides full access to AWS services and resources.
<input type="checkbox"/>	AlexaForBusinessDeviceSetup	0	Provide device setup access to AlexaForBusiness services
<input type="checkbox"/>	AlexaForBusinessFullAccess	0	Grants full access to AlexaForBusiness resources and acc...
<input type="checkbox"/>	AlexaForBusinessGatewayExecution	0	Provide gateway execution access to AlexaForBusiness s...
<input type="checkbox"/>	AlexaForBusinessReadOnlyAccess	0	Provide read only access to AlexaForBusiness services
<input type="checkbox"/>	AmazonAPIGatewayAdministrator	1	Provides full access to create/edit/delete APIs in Amazon ...
<input type="checkbox"/>	AmazonAPIGatewayInvokeFullAccess	2	Provides full access to invoke APIs in Amazon API Gateway.
<input type="checkbox"/>	AmazonAPIGatewayPushToCloudWatchLogs	0	Allows API Gateway to push logs to user's account.
<input type="checkbox"/>	AmazonAppStreamFullAccess	0	Provides full access to Amazon AppStream via the AWS ...
<input type="checkbox"/>	AmazonAppStreamReadOnlyAccess	0	Provides read only access to Amazon AppStream via the ...
<input type="checkbox"/>	AmazonAppStreamServiceAccess	0	Default policy for Amazon AppStream service role.

Step 4

Observe that we have chosen the following permissions –

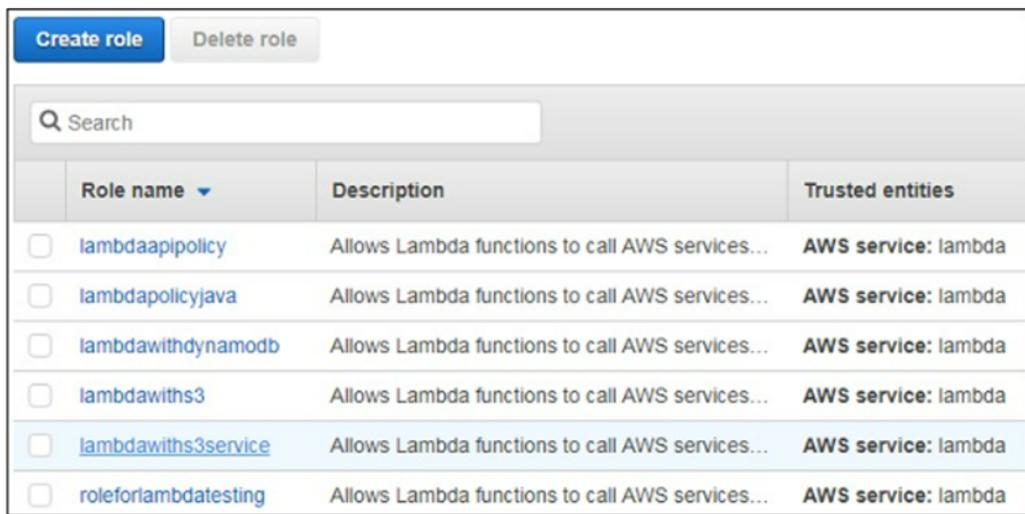
We have selected AmazonS3FullAccess, AWSLambdaFullAccess and CloudWatchFullAccess.

Step 5

Create Lambda function and Add S3 Trigger

In this section, let us see how to create a Lambda function and add a S3 trigger to it. For this purpose, you will have to follow the Steps given below –

Now, enter the Role name, Role description and click **Create Role** button at the bottom.

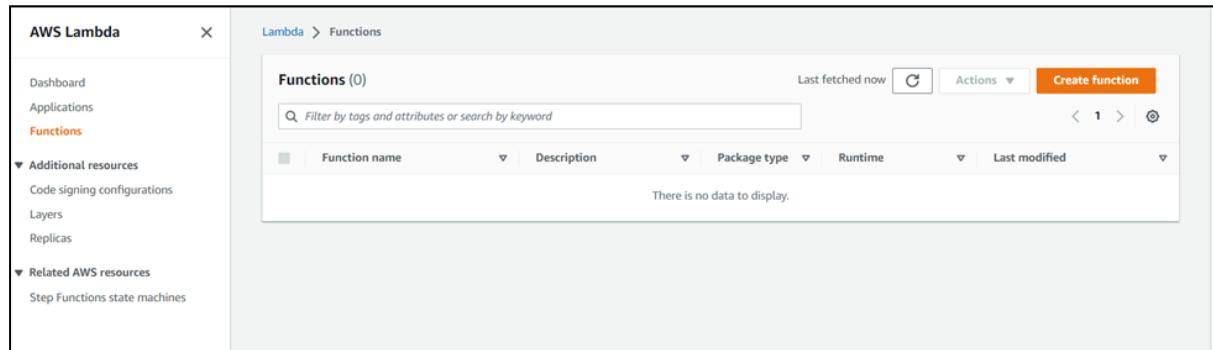


The screenshot shows the AWS IAM Roles list page. At the top, there are two buttons: "Create role" (blue) and "Delete role". Below is a search bar with the placeholder "Search". A table lists six roles:

Role name	Description	Trusted entities
lambdaapipolicy	Allows Lambda functions to call AWS services...	AWS service: lambda
lambdapolicyjava	Allows Lambda functions to call AWS services...	AWS service: lambda
lambdawithdynamodb	Allows Lambda functions to call AWS services...	AWS service: lambda
lambdawiths3	Allows Lambda functions to call AWS services...	AWS service: lambda
lambdawiths3service	Allows Lambda functions to call AWS services...	AWS service: lambda
roleforlambdatesting	Allows Lambda functions to call AWS services...	AWS service: lambda

Step 1

Go to AWS Services and select Lambda as shown below –



The screenshot shows the AWS Lambda Functions list page. On the left, there is a sidebar with the following navigation:

- AWS Lambda
- Dashboard
- Applications
- Functions** (highlighted)
- Additional resources
- Related AWS resources

The main area shows the "Functions (0)" list. It includes a search bar, a toolbar with "Actions" and "Create function" buttons, and a table header with columns: Function name, Description, Package type, Runtime, and Last modified. A message at the bottom states: "There is no data to display."

Step 2

Click Lambda and follow the process for adding Name. Choose the Runtime, Role etc. and create the function. The Lambda function that we have created is shown in the screenshot below –

Use cases for other AWS services:

Lambda

Lambda
Allows Lambda functions to call AWS services on your behalf

Basic information

Function name
Enter a name that describes the purpose of your function.
demolambdacode

Use only letters, numbers, hyphens, or underscores with no spaces.

Runtime [Info](#)
Choose the language to use to write your function. Note that the console code editor supports only Node.js, Python, and Ruby.
Node.js 16.x

Architecture [Info](#)

Architecture [Info](#)
Choose the instruction set architecture you want for your function code.
 x86_64
 arm64

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

▼ Change default execution role

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

Existing role
Choose an existing role that you've created to be used with this Lambda function. The role must have permission to upload logs to Amazon CloudWatch Logs.
demolambdarole [View](#) [Edit](#)

[Lambda](#) > [Functions](#)

Functions (1)						Last fetched 3 minutes ago	Actions	Create function
<input type="checkbox"/> Function name	Description	Package type	Runtime	Last modified				
demolambda	-	Zip	Node.js 16.x	2 minutes ago				

demolambdacode

▼ Function overview [Info](#)

 demolambdacode  Layers (0) + Add trigger	Description - Last modified 28 seconds ago Function ARN arn:aws:lambda:ap-south-1:625453636107:function:demolambdacode Function URL Info -
--	--

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

Hence a lambda function is created

Now Let's Create a trigger

The screenshot shows the 'Trigger configuration' step of creating a new Lambda function. It is specifically for triggering from an S3 bucket.

Bucket: s3/demolambda-2910 (selected)

Event type: All object create events

Prefix - optional: e.g. images/

Suffix - optional: e.g. jpg

Recursive invocation: A note about using the same S3 bucket for both input and output, with an acknowledgement checkbox.

I acknowledge that using the same S3 bucket for both input and output is not recommended and that this configuration can cause recursive invocations, increased costs.

Below is the code written in node.js to pinned a log about an object that has been added to s3.

The screenshot shows the AWS Lambda code editor with the 'index.js' file open. The code is a simple event handler for S3 object creation events.

```
1 exports.handler = function(event, context, callback) {
2     console.log("Incoming Event: " + event);
3     const bucket = event.Records[0].s3.bucket.name;
4     const filename = decodeURIComponent(event.Records[0].s3.object.key.replace(/\+/g, ' '));
5     const message = `An image has been added to - ${bucket} - ${filename}`;
6     console.log(message);
7     callback(null, message);
8 };
```

The screenshot shows the 'Triggers' section of the AWS Lambda console. It lists one trigger, 'S3: demolambda-2910', which is associated with the ARN 'arn:aws:s3:::demolambda-2910'. There are buttons for 'Fix errors', 'Edit', 'Delete', and 'Add trigger' at the top right.

Now to test our lambda function with s3 trigger.

Step 1.add a image to s3 bucket

The screenshot shows the 'Functions' section of the AWS Lambda console. It lists one function, 'demolambda', which was last modified 2 minutes ago. The function uses a 'Zip' package type and 'Node.js 16.x' runtime.

Step 2.

Now go to cloudwatch you will see the logs of newly happened activity in s3.

The screenshot shows the 'Log groups' section of the AWS CloudWatch console. It displays one log group, '/aws/lambda/demolambda', which contains one log stream. The log stream has a last event time of '2022-10-29 20:01:38 (UTC+05:30)'.

The screenshot shows the 'Log group details' page for the '/aws/lambda/demolambda' log group. It shows a single log entry:

```
START RequestId: effb334d-d472-4932-8971-aca4c35b7552 Version: $LATEST
END RequestId: effb334d-d472-4932-8971-aca4c35b7552
REPORT RequestId: effb334d-d472-4932-8971-aca4c35b7552 Duration: 18.15 ms Billed Duration: 19 ms Memory Size: 128 MB Max Memory Used: 57 MB Init Duration: 138.57 ms
No newer events at this moment. Auto retry paused. Resume
```