GLOBAL ACADEMY OF TECHNOLOGY

An Autonomous Institute, Affiliated to VTU Belagavi,
Approved by AICTE, Accredited by NAAC,'A' Grade,
Ideal Homes Township, Raja Rajeshwari Nagar, Bengaluru – 560 098
Tel: +91 80 28603158 Email: hodaiml@gat.ac.in Web: www.gat.ac.in

Department of Artificial Intelligence and Machine Learning

LAB MANUAL

Cloud Computing Lab Manual (Integrated)

VI Semester

Course Code:22AML61

Editorial Committee

Cloud Computing Laboratory

Dept. of AIML

Approved by

Dr. Roopa B S

Professor & Head,
Department of AIML

GLOBAL ACADEMY OF TECHNOLOGY

An Autonomous Institute, Affiliated to VTU Belagavi,
Approved by AICTE, Accredited by NAAC,'A' Grade,
Ideal Homes Township, Raja Rajeshwari Nagar, Bengaluru – 560 098
Tel: +91 80 28603158 Email: hodaiml@gat.ac.in Web: www.gat.ac.in

Department of Artificial Intelligence and Machine Learning

LABORATORY CERTIFICATE

This is to certify that Mr./ Ms. .....

bearing USN..... of Artificial Intelligence & Machine Learning

and the Engineering Department has satisfactorily completed the course of Cloud

Computing Laboratory (Integrated) prescribed by Global Academy of Technology

(Autonomous Institute, under VTU) for the Academic Year 2024-2025

Marks	
Maximum	**Obtained**
20	

Signature of the Faculty

Signature of the HOD

Date:

DOCUMENT LOG

Name of the document	Cloud Computing Laboratory (Integrated)
Syllabus Scheme	2022
Semester	V1
Subject code	22AML61
Editorial Committee	Cloud Computing Laboratory
Approved by	HOD, Dept. of AIML
Lab Faculty	Prof. Prasanna N
Computer Programmer	Ms. Ramyashree M R

Table of Contents

Sl.No.	Particulars.		Page No.
1	**Vision and Mission of The Department**		**1**
2	**PEOs, PSOs, POs**		**2-3**
3	Syllabus • Course Objectives • Course Outcomes • Conduction of Practical Examination • CO-PO-PSO Mapping		**4-5**
4	Scheme of Evaluation		**6**
5	Marks Distribution		**6**
6	Rubrics		**7**
7	**PROGRAMS**		
	Program 1	Install Virtual Box or VMware Workstation on a Windows 11 system, set up a virtual machine with a Linux or Windows operating system, install a C compiler, and execute simple C programs.	**8**
	Program 2	Find a procedure to transfer the files from One Virtual Machine to another Virtual Machine.	**15**
	Program 3	Deploy a Flask web application to Google Cloud using Google Cloud CLI as PaaS.	**23**
	Program 4	Build a Docker image for a simple Flask web application and push it to your Docker Hub repository.	**40**
	Program 5	Pull a Docker image from Docker Hub and deploy your Flask app using Docker Swarm**.**	**59**
	Program 6	Find vulnerabilities, misconfigurations and secrets in files, Docker, Docker Hub and Git Repo using Trivy .and generate a summary report in HTML format.	**62**
	Program 7	Simulate and Manage Amazon S3 buckets and EC2 using Local Stack locally. Perform operations using both the Local Stack Cloud Dashboard and AWS CLI.	**71**
	Program 8	Simulate and manage a Amazon Dynamo DB Using AWS CLI with AWS LocalStack Cloud.	**86**

Type equation here.

Table of Contents

Sl.No.	Particulars.		Page No.
1	**Vision and Mission of The Department**		1
2	**PEOs, PSOs, POs**		2-3
3	Syllabus • Course Objectives • Course Outcomes • Conduction of Practical Examination • CO-PO-PSO Mapping		4-5
4	Scheme of Evaluation		6
5	Marks Distribution		6
6	Rubrics		7
7	**PROGRAMS**		
	Program 1	Install Virtual Box or VMware Workstation on a Windows 11 system, set up a virtual machine with a Linux or Windows operating system, install a C compiler, and execute simple C programs.	8
	Program 2	Find a procedure to transfer the files from One Virtual Machine to another Virtual Machine .	15
	Program 3	Deploy a Flask web application to Google Cloud using Google Cloud CLI as PaaS .	23
	Program 4	Build a Docker image for a simple Flask web application and push it to your Docker Hub repository.	40
	Program 5	Pull a Docker image from Docker Hub and deploy your Flask app using Docker Swarm**.**	59
	Program 6	Find vulnerabilities, misconfigurations and secrets in files, Docker, Docker Hub and Git Repo using Trivy .and generate a summary report in HTML format.	62
	Program 7	Simulate and Manage Amazon S3 buckets and EC2 using LocalStack locally. Perform operations using both the LocalStack Cloud Dashboard and AWS CLI.	71
	Program 8	Simulate and manage a Amazon Dynamo DB Using AWS CLI with AWS LocalStack Cloud.	86

Vision of the Department

To provide progressive education and create transformative professionals and leaders to harness the power of technology and make a positive impact on society.

Mission of the Department

M1 - Quality Education: To adopt a student centric curriculum delivery process with emphasis on problem solving and programming skills.

M2 - Innovation: To collaborate with industries and professional bodies and make the students industry ready. To drive innovation through multi-disciplinary research and development activities.

M3 - Skill Development & Ethics: To endorse additional skill development through student forums and experiential learning. Inculcate human values for a smarter and ethical world.

Program Outcomes (POs)

Engineering Graduates will be able to:

1. **Engineering knowledge:** Apply the knowledge of mathematics, science, engineering fundamentals, and an engineering specialization to the solution of complex engineering problems.

2. **Problem analysis**: Identify, formulate, review research literature, and analyze complex engineering problems reaching substantiated conclusions using first principles of mathematics, natural sciences, and engineering sciences.

3. **Design/development of solutions**: Design solutions for complex engineering problems and design system components or processes that meet the specified needs with appropriate consideration for the public health and safety, and the cultural, societal, and environmental considerations.

4. **Conduct investigations of complex problems**: Use research-based knowledge and research methods including design of experiments, analysis and interpretation of data, and synthesis of the information to provide valid conclusions.

5. **Modern tool usage**: Create, select, and apply appropriate techniques, resources, and modern engineering and IT tools including prediction and modeling to complex engineering activities with an understanding of the limitations.

6. **The engineer and society**: Apply reasoning informed by the contextual knowledge to assess societal, health, safety, legal and cultural issues and the consequent responsibilities relevant to the professional engineering practice.

7. **Environment and sustainability**: Understand the impact of the professional engineering solutions in societal and environmental contexts, and demonstrate the knowledge of, and need for sustainable development.

8. **Ethics**: Apply ethical principles and commit to professional ethics and responsibilities and norms of the engineering practice.

9. **Individual and team work**: Function effectively as an individual, and as a member or leader in diverse teams, and in multidisciplinary settings.

10. **Communication**: Communicate effectively on complex engineering activities with the engineering community and with society at large, such as, being able to comprehend and write effective reports and design documentation, make effective presentations, and give and receive clear instructions.

11. **Project management and finance**: Demonstrate knowledge and understanding of the engineering and management principles and apply these to one's own work, as a member and leader in a team, to manage projects and in multidisciplinary environments.

12. **Life-long learning**: Recognize the need for, and have the preparation and ability to engage in independent and life-long learning in the broadest context of technological change.

Program Educational Objectives (PEOs) of Department

Graduates in Artificial Intelligence and Data Science will be able to:

PEO1. Able to practice and implement their success skills like problem solving, communication and collaboration for providing innovative engineering solutions.

PEO2. Contribute their AI & ML expertise grounded in computer science as members and leaders of professional engineering teams in multidisciplinary applications.

PEO3. Demonstrate lifelong learning through continued professional development and higher education in top graduate technical, research and management programs.

PEO4. Demonstrate commitment to society by applying the skills and knowledge for a smarter and ethical world.

Program Specific Outcomes (PSOs)

After successful completion of Artificial Intelligence and Machine Learning program

PSO1. Graduates will be Proficient in programming and problem-solving skills for developing, managing software and distributed systems.

PSO2. Graduates will be able to identify, formulate, predict and solve real world problems by applying principles of Artificial Intelligence & Machine learning.

SYLLABUS

Subject Code	22AML61	CIE Marks	50
Hours/Week (L: T: P)	0:0:2	SEE Marks	50
Credits	4	Examination Hours	03

Course Learning Objectives

Sl. No	Course Learning Objectives (CLO)
1	To understand the fundamental ideas behind cloud computing, evolution of the paradigm, its applicability, benefits and challenges.
2	Discuss virtualization and outline their role in enabling the cloud computing system model and different platforms in regard to industry relevance.
3	Illustrate the security aspects for building cloud-based applications and outline the applications of cloud in various industries.

Lab Programs:

1. Install Virtual Box or VMware Workstation on a Windows 11 system, set up a virtual machine with a Linux or Windows operating system, install a C compiler and execute simple C programs.
2. Find a procedure to transfer the files from One Virtual Machine to another Virtual Machine.
3. Deploy a Flask web application to Google Cloud using Google Cloud CLI as PaaS.
4. Build a Docker image for a simple Flask web application and push it to your Docker Hub repository.
5. Pull a Docker image from Docker Hub and deploy your Flask app using Docker Swarm.
6. Find vulnerabilities, misconfigurations and secrets in files, Docker, Docker Hub and Git Repo using Aqua Trivy. and generate a summary report in HTML format.
7. Simulate and Manage Amazon S3 buckets and EC2 using LocalStack locally. Perform operations using both the LocalStack Cloud Dashboard and AWS CLI.
8. Simulate and manage an Amazon Dynamo DB Using AWS CLI with AWS LocalStack Cloud.

Course Outcomes

Upon successful completion of this course, students are able to

CO61.1	Understand the core concepts of cloud computing paradigm and Describe the various cloud computing platforms.
CO61.2	Outline the various virtualization technologies and Illustrate a virtual instance using virtualization.
CO61.3	Explain the architecture, infrastructure and deployment models of cloud computing.
CO61.4	Understand the security aspects of cloud and Identify the platforms for development of cloud applications.
CO61.5	Obtain an insight on applications of cloud and Develop a project for the applications

CO-PO Mapping

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	O12	SO1	SO2
CO61.1	3						1					3		
CO61.2	3	2	3									3	2	
CO61.3	3	2	2		1		1					3	3	
CO61.4	3	3	3		2	1						3	3	
CO61.5	3	3	3		2	1	1					3	3	
Average	3	2	3		2	1	1					3	3	

Scheme of Evaluation:

(Integrated courses)

Semester End Examination (SEE):

SEE Question paper is to be set for 100 marks and the marks scored will be proportionately reduced to 50. There will be two full questions (with a maximum of four sub questions) from each module carrying 20 marks each. Students are required to answer any **five full questions** choosing at least **one full question from each module.**

The laboratory assessment would be restricted to only the CIE evaluation.

Continuous Internal Evaluation (CIE):

Continuous Internal Evaluation (CIE): Three Tests are to be conducted for 40 marks each. The average of the three tests are taken for computation of CIE on a scale of 30, the CIE would also include laboratory evaluation for 20 marks. The laboratory marks of 20 would comprise of 10 marks for regular laboratory assessment to include lab record and observation. 10 marks would be exclusive for laboratory internal assessment test to be conducted at the end of the semester.

Table: Distribution of weightage for CIE & SEE of Integrated courses

	Component	Marks	Total Marks
CIE	Average of Three CIEs (Min Passing Marks:12)	**30**	**50**
	Laboratory (Min Passing Marks:8)	**20**	
SEE	Semester End Examination (Min Passing Mark:40)	**100**	**50**
		Grand Total	**100**

LAB EVALUATION PROCESS

Sl.No	Parameter	Max. Marks (Evaluation)	Reduced to
1	Record & Observation	30 Marks per week	10
2	Final Lab Internal Assessment	100	10
	Total		**20**

LAB RUBRICS

Rubrics for Evaluation of Record and Observation

Attribute	Max. Marks	Good	Satisfactory	Poor
		4-5	2-3	0-1
Write-up	**05**	Write the program without errors.	Writes the program with few mistakes.	Unable to write the program.
	Max. Marks	**10-20**	**5-9**	**0-4**
Execution of program	**20**	• Debugs the program independently. • Executed the program for all possible inputs.	• Works with little help from faculty. • All possible input cases are not covered.	• Unable to complete the execution of the program within the lab session.
	Max. Marks	**3-5**	**1-2**	**0**
Viva Voce	**05**	• Able to explain the logic of the program. • Answered all questions.	• Partially understood the logic of the program. • Answered few questions.	• Not understood the logic of the program. • Not answering Sany questions

Rubrics for Evaluation of Internal Test Part A

Attribute	Max Marks	Good	Satisfactory	Poor
		4-5	2-3	0-1
Write-up	**10**	• Completes source code. • No syntax and logical errors. • All possible inputs listed with expected output.	• Completes source code. • Syntax and logical errors exist. • All possible inputs listed with expected output.	• Incomplete source code. • Change of program.
	Max Marks	**21-30**	**6-20**	**1-5**
Execution	**30**	Able to debug the program and get the right set of outputs.	• Program, executed for only few input cases.	Not executed.
	Max Marks	**3-5**	**1-2**	**0**
Viva Voce	**10**	Answering all questions.	Answering few questions.	Not Able to answer basic questions.

Program 1:

Install VirtualBox or VMware Workstation on a Windows 11 system, set up a virtual machine with a Linux or Windows operating system, install a C compiler, and execute simple C programs.

Prerequisite Software

1. Windows 11 OS

2. VMware Workstation

3. Ubuntu 18.04

Implementation Steps

Step1: Install VMware Workstation

Step2. Click on Create a New Virtual Machine

Step3: Select Types of Configuration and click Next

Step4: Select the Disc image of Ubuntu 18 and click Next

New Virtual Machine Wizard
Browse for ISO Image
Desktop
Search Desktop
Organize
New folder
Home
Gallery
OneDrive
Desktop
Downloads
Documents
Pictures
Name
Date modified
Type
119
12-03-2025 02:15 PM
File folder
403
13-02-2025 10:05 AM
File folder
Abhi003
18-02-2025 11:42 AM
File folder
all dataset
20-08-2024 02:52 PM
File folder
cloud
21-02-2025 11:53 AM
File folder
cloudLab
18-02-2025 11:55 AM
File folder
Downloads
27-08-2024 02:43 PM
File folder
ubuntu-18.04.6-desktop-amd64
13-03-2025 11:57 AM
Disc Image File
File name:
ubuntu-18.04.6-desktop-amd64
CD-ROM images (*.iso)
Open
Cancel

New Virtual Machine Wizard
Guest Operating System Installation
A virtual machine is like a physical computer; it needs an operating system. How will you install the guest operating system?
Install from:
Installer disc:
No drives available
Installer disc image file (iso):
C:\Users\Admin\Desktop\ubuntu-18.04.6-desktop-am
Browse...
Ubuntu 64-bit 18.04.6 detected.
This operating system will use Easy Install. (What's this?)
I will install the operating system later.
The virtual machine will be created with a blank hard disk.
Help
< Back
Next >
Cancel

Step5: Enter the Personalize details.

Ex:

Full Name:

User name:

Password:

Conform:

and click Next.

Step5: Enter the Virtual Machine Name, keep default Location and Click Next.

Step6: Choose The Required Hardware Requirements and click Next
EX:
Processor: 2 CPUs
RAM : 2 GB
Hard Disk: 20 GB

Step6: Click Finish

To compile and run a C program in an Ubuntu virtual machine, follow these steps:

Step 1: Install the GCC Compiler (if not already installed)

The GCC (GNU Compiler Collection) is usually pre-installed on Ubuntu, but if it's not, you can install it with the following command:

Open a terminal window on your Ubuntu VM.

Run the following command to install GCC:

1. `sudo apt update`
2. `sudo apt install gcc` This will install GCC along with other essential tools for development.

Step 2: Create a C Program

Open a text editor like `nano`, `vim`, or `gedit` to write your C program. For example, to use `nano`, run:

```
nano my_program.c
```

//Write your C code in the editor. For example, you can write a simple "Hello, //World!" program:

```
#include <stdio.h>
int main ()
{
 printf ("Hello, World! \n");
 return 0;
 }
```

Save the file and exit. In `nano`, press `Ctrl + O` to save and `Ctrl + X` to exit.

Step 3: Compile the C Program

1. To compile your C program, run the following command in the terminal:

```
gcc my_program.c -o my_program
```

   - `gcc` is the C compiler.
   - `my_program.c` is the name of your C source file.
   - `-o my_program` tells the compiler to output the executable with the name `my_program`.

If there are no errors in your code, this will create an executable file named `my_program`.

Step 4: Run the Program

To run the compiled C program, use the following command:

```
. /my_program
```

This will execute the program, and you should see the output:

```
Hello, World!
```

Program 2:

Find a procedure to transfer the files from One Virtual Machine to another Virtual Machine.

Step 1: Creation of Virtual Machine 1

Full name: Global10

User name: gat1

Step 2: Creation of Virtual Machine 2

Full name : Global2

User name : gat2

Step 3: sudo apt install net-tools is used. This command installs essential networking utilities like ifconfig, netstat, and route. Here, it is used to know the IP address of the respective Virtual machines.

Step 4: The IP address of the Virtual Machine 1 is 192.168.90.128

Step 5: sudo apt install openssh-server -y.

This command is used to install and start the OpenSSH Server

Step 6: Creation of Virtual Machine 2

Ubuntu 18-2 - VMware Workstation
File Edit View VM Tabs Help
Library
Type here to search
My Computer
Ubuntu18-1
Ubuntu 18-2
Home
Ubuntu 18-2
Activities
Terminal
Wed 22:31
gat2@ubuntu: ~
File Edit View Search Terminal Help
gat2@ubuntu:~$ ifconfig
ens33: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
inet 192.168.90.129 netmask 255.255.255.0 broadcast 192.168.90.255
inet6 fe80::d524:11a0:1898:2075 prefixlen 64 scopeid 0x20<link>
ether 00:0c:29:99:ed:42 txqueuelen 1000 (Ethernet)
RX packets 5692 bytes 7529135 (7.5 MB)
RX errors 0 dropped 0 overruns 0 frame 0
TX packets 2760 bytes 384064 (384.0 KB)
TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0
lo: flags=73<UP,LOOPBACK,RUNNING> mtu 65536
inet 127.0.0.1 netmask 255.0.0.0
To direct input to this VM, move the mouse pointer inside or press Ctrl+G.

Ubuntu 18-2 - VMware Workstation
File Edit View VM Tabs Help
Library
Type here to search
My Computer
Ubuntu18-1
Ubuntu 18-2
Home
Ubuntu 18-2
Activities
Terminal
Wed 22:25
gat2@ubuntu: ~
File Edit View Search Terminal Help
gat2@ubuntu:~$ sudo apt install openssh-server -y
To direct input to this VM, move the mouse pointer inside or press Ctrl+G.

Step 7: Creating a file named p1.txt on Virtual Machine 1

Step 8: The scp command in Linux is a command-line utility that uses the Secure Shell (SSH) protocol to securely copy files or directories between a local and a remote system, or between two remote systems.

Step 9: Transferring the file p1.txt from Virtual Machine 1 to Virtual Machine 2

Step 10: Successful file transfer from Virtual Machine 1 to Virtual Machine 2

Step 11: The command rsync or remote synchronization that efficiently sync files and directories between two hosts or machines.

Program 3:

Deploy a Flask web application to Google Cloud using Google Cloud CLI as PaaS.

Step1: Download the Google Cloud CLI installer.

URL: https://cloud.google.com/sdk/docs/install

Install the gcloud CLI

Send feedback

This page contains instructions for choosing and maintaining a Google Cloud CLI installation. The Google Cloud CLI includes the `gcloud`, `gsutil` and `bq` command-line tools. For a list of gcloud CLI features, see All features.

To access the Google Cloud APIs using a supported programming language, you can download the Cloud Client Libraries.

Installation instructions

These instructions are for installing the Google Cloud CLI. For information about installing additional components, such as gcloud CLI commands at the alpha or beta release level, see Managing gcloud CLI components.

Note: If you are behind a proxy or firewall, see the proxy settings page for more information on installation.

Linux | Debian/Ubuntu | Red Hat/Fedora/CentOS | macOS | Windows

The Google Cloud CLI works on Windows 8.1 and later and Windows Server 2012 and later.

1. Download the Google Cloud CLI installer.

Alternatively, open a PowerShell terminal and run the following PowerShell commands:

Step2: Install the Google Cloud CLI installer.

2.1 Click Next

2.2 Click on I Agree

2.3 Select user and click next

2.4 SELECT THE DEFAULT DESTINATION FOLDER AND CLICK NEXT

2.5 CLICK ALL THE CHECKBOXES AND PRESS INSTALL BUTTON

2.6 CLICK ON NEXT FOR THE SUCCESSFUL INSTALLATION

2.7 CLICK ON FINISH

2.8 Gcloud run commands run in terminal, Press: Y

2.9 AFTER PRESSSING THE FINISH BUTTON,U GO TO COMMND PROMPT AND PRESS 'y' WHICH TAKES TO THE SISN-IN MODE AND SIGN-UP TO THE ANY OF YOUR PERSONAL GMAIL ACCOUNT

U WILL GO TO THE PASSWORD SCREEN , TYPE UR PASSWORD AND THEN CLICK CONTINUE

2.10 PRESS CONTINUE AFTER SUCCESSFULLY LOGIN

2.11 AFTER CONTINUE BUTTON PRESSING, U WILL GET THE BELOW SCREEN AND PRESS ALLOW AND U WILL GET THE AUTHENTICATED SCREEN

2.12 AUTHENTICATED SCREEN

Cloud SDK > Documentation

Was this helpful?

You are now authenticated with the gcloud CLI!

Send feedback

The authentication flow has completed successfully. You may close this window, or check out the resources below.

Information about command-line tools and client libraries

To learn more about Google Cloud CLI commands, see the gcloud CLI guide.

To learn more about the command-line tools for App Engine, Compute Engine, Cloud Storage, BigQuery, Cloud SQL, and Cloud DNS (which are all bundled with the gcloud CLI), see Accessing services with the gcloud CLI.

If you're a client application developer and want to find out more about accessing Google Cloud services with a programming language or framework, see Client Libraries Explained.

2.13 AFTER AUTHENTICATION SCREEN,

YOU WILL GET THE BELOW SCREEN, WITH **LIST OF PROJECTS ID**,

SELECT ANY ONE (AS YOUR WISH) WITH THE NUMBER OF ENTRIES

```
C:\WINDOWS\SYSTEM32\cmd

Reachability Check passed.
Network diagnostic passed (1/1 checks passed).

You must sign in to continue. Would you like to sign in (Y/n)?  y

Your browser has been opened to visit:

    https://accounts.google.com/o/oauth2/auth?response_type=code&client_id=32555940559.apps.goog
leusercontent.com&redirect_uri=http%3A%2F%2Flocalhost%3A8085%2F&scope=openid+https%3A%2F%2Fwww.g
oogleapis.com%2Fauth%2Fuserinfo.email+https%3A%2F%2Fwww.googleapis.com%2Fauth%2Fcloud-platform+h
ttps%3A%2F%2Fwww.googleapis.com%2Fauth%2Fappengine.admin+https%3A%2F%2Fwww.googleapis.com%2Fauth
%2Fsqlservice.login+https%3A%2F%2Fwww.googleapis.com%2Fauth%2Fcompute+https%3A%2F%2Fwww.googleap
is.com%2Fauth%2Faccounts.reauth&state=pfHGcf3gU3sszw9iplbNd6S2aHJopG&access_type=offline&code_ch
allenge=-hvKM_n0qk3ttxU-kH826zj0-E2pBASjs5sW3nJp-dY&code_challenge_method=S256

You are signed in as: [psmurthycs@gmail.com].

Pick cloud project to use:
 [1] cobalt-vector-454407-p1
 [2] elite-advice-454411-b7
 [3] plasma-ripple-455514-n1
 [4] proud-plane-451708-j7
 [5] shining-bazaar-450804-q0
 [6] soft-1274
 [7] still-protocol-454411-a4
 [8] stone-timing-454903-b7
 [9] Enter a project ID
 [10] Create a new project
Please enter numeric choice or text value (must exactly match list item):  1
```

2.14 YOU GET THE BELOW WINDOW AND CHK THE MESSAGE ABOUT YOUR CURRENT PROJECT has BEEN SET TO

```
Please enter numeric choice or text value (must exactly match list item):  1

Your current project has been set to: [cobalt-vector-454407-p1].

Not setting default zone/region (this feature makes it easier to use
[gcloud compute] by setting an appropriate default value for the
--zone and --region flag).
See https://cloud.google.com/compute/docs/gcloud-compute section on how to set
default compute region and zone manually. If you would like [gcloud init] to be
able to do this for you the next time you run it, make sure the
Compute Engine API is enabled for your project on the
https://console.developers.google.com/apis page.

Created a default .boto configuration file at [C:\Users\Admin\.boto]. See this file and
[https://cloud.google.com/storage/docs/gsutil/commands/config] for more
information about configuring Google Cloud Storage.
The Google Cloud CLI is configured and ready to use!

* Commands that require authentication will use psmurthycs@gmail.com by default
* Commands will reference project `cobalt-vector-454407-p1` by default
Run `gcloud help config` to learn how to change individual settings

This gcloud configuration is called [default]. You can create additional configurations if you work with multiple accoun
ts and/or projects.
Run `gcloud topic configurations` to learn more.

Some things to try next:
```

2.15 If a Google Cloud Project ID hasn't been created,

you'll need to create a new project

URL : https://console.cloud.google.com/terms

click on Select Project

2.16 Click on New project

2.17 Enter Project Name and Click on Create

2.18 Open Terminal, Type gcloud init to reinitialize

2.19 Choices 1 to reinitialize

Command Prompt - gcloud init

```
Microsoft Windows [Version 10.0.19045.5854]
(c) Microsoft Corporation. All rights reserved.

C:\Users\Parashivamurthy>gcloud init
Welcome! This command will take you through the configuration of gcloud.

Settings from your current configuration [default] are:
accessibility:
  screen_reader: 'True'
core:
  account: psmurthycs@gmail.com
  disable_usage_reporting: 'False'
  project: cobalt-vector-454407-p1

Pick configuration to use:
 [1] Re-initialize this configuration [default] with new settings
 [2] Create a new configuration
Please enter your numeric choice:  1
```

2.20 select an your Account

Command Prompt - gcloud init

```
 [2] Create a new configuration
Please enter your numeric choice:  1

Your current configuration has been set to: [default]

You can skip diagnostics next time by using the following flag:
  gcloud init --skip-diagnostics

Network diagnostic detects and fixes local network connection issues.

Checking network connection...done.
Reachability Check passed.
Network diagnostic passed (1/1 checks passed).

Choose the account you want to use for this configuration.
To use a federated user account, exit this command and sign in to the
 gcloud CLI with your login configuration file,
then run this command again.

Select an account:
 [1] psmurthycs@gmail.com
 [2] Sign in with a new Google Account
 [3] Skip this step
Please enter your numeric choice:  1
```

2.21 Select Project ID

```
Pick cloud project to use:
 [1] animated-tracer-457707-s6
 [2] crested-unity-454403-e7
 [3] delta-geode-454407-q4
 [4] global123-457707
 [5] keen-metric-454403-t3
 [6] Enter a project ID
 [7] Create a new project
Please enter numeric choice or text value (must exactly match list item):  4

Your current project has been set to: [global123-457707].

Not setting default zone/region (this feature makes it easier to use
[gcloud compute] by setting an appropriate default value for the
--zone and --region flag).
See https://cloud.google.com/compute/docs/gcloud-compute section on how to set
default compute region and zone manually. If you would like [gcloud init] to be
able to do this for you the next time you run it, make sure the
Compute Engine API is enabled for your project on the
https://console.developers.google.com/apis page.
```

Step3: Download and Install Python 3.9.0 or 3.10

3.1 Check the Version

```
C:\Users\Admin>python --version
Python 3.9.0
```

3.2 Install the App Engine component:

Ex: gcloud components install app-engine-python

Note

(For Java, use app-engine-java instead of app-engine-python.).

GO TO THE CMD APROMPT AND EXECUTE THE FOLLOWING COMMAND

```
C:\Users\Student>gcloud components install app-engine-python
```

3.3 All the below components in the box will be installed with sdk installer AN DPRESS 'y'

```
cmd.exe /c ""C:\Users\Admin

Your current Google Cloud CLI version is: 519.0.0
Installing components from version: 519.0.0

These components will be installed.

Name: Cloud Datastore Emulator
Version: 2.3.1
Size: 36.2 MiB

Name: gRPC Python library (Platform Specific)
Version: 1.70.0
Size: 4.0 MiB

Name: gcloud app Python Extensions
Version: 1.9.114
Size: 3.8 MiB

For the latest full release notes, please visit:
  https://cloud.google.com/sdk/release_notes

Once started, canceling this operation may leave your SDK installation in an inconsistent state.

Do you want to continue (Y/n)?  Y
```

```
Installing: gcloud app Python Extensions
5%
10%
15%
20%
25%
30%
35%
40%
46%
51%
56%
61%
66%
71%
76%
81%
86%
91%
96%
100%
100%
100%

Performing post processing steps...done.

Update done!

Press any key to continue . . .
```

Press any key to exit

Step4: Install Flask:

```
C:\Users\Student>pip install flask
```

4.1 CHK THE PYTHON VERSION USING THE FOLLOWING COMMAND TYPE FLASK --VERSON

```
C:\Users\Admin>flask --version
Python 3.9.0
Flask 3.1.0
Werkzeug 3.1.3
```

Step 5: AFTER THE SUCCESFUL INSTALLATION, GO TO C DRIVE

5.1 Create a new folder in C drive: WITH A NAME AS flask-app

Change THE DEFAULT DIRECTORT PATH TO THE FLASK-APP DIRECTORY Flask-app

```
Command Prompt

C:\Users\Admin>cd ../..

C:\>cd flask-app

C:\flask-app>
```

CREATE THE FOLLOWING FILES MENTIONED BELOW IN THE FLASK-APP DIRECTORY , AS SHOWN BELOW BY CLICKING ON THE TEXT DOCUMENT AND save THE FILES

AS app.yaml, main.py and requirements.txt

5.2 Create a file named app.yaml

Note:

To determine the Python version you're using, open a terminal or command prompt and type python --version or python -V. This command will display the installed Python version. For example, you might see "Python 3.9.0"

5.3 Then in your `app.yaml`, you should specify the runtime as: runtime: python39

```
runtime: python39
entrypoint: python main.py  # Directly run Flask

handlers:
- url: /.*
  script: auto
```

5.4 Create a file named main.py

```
from flask import Flask

app = Flask(__name__)

@app.route('/')
def hello():
    return 'Hello, World!'

if __name__ == '__main__':
    app.run(host='0.0.0.0', port=8080)
```

5.5 Create a file named: requirements.txt

```
flask
```

Step 6: Run the app locally - before deploying to Google Cloud

Note: The installed path of the Google Cloud CLI depends on your operating system and how you installed it. Here are the typical default installation paths

6.1 python "C:\Users\Admin\AppData\Local\Google\Cloud SDK\google-cloud-sdk\bin\dev_appserver.py" app.yaml

```
C:\flask-app>python "C:\Users\Admin\AppData\Local\Google\Cloud SDK\google-cloud-sdk\bin\dev_appserver.py" app.yaml
```

```
C:\Windows\System32\cmd.e

WARNING  2025-04-02 08:18:41,834 urlfetch_stub.py:97] C:\Users\Student\AppData\Local\Google\Cloud SDK\google-cloud-sdk\p
latform\google_appengine\google\appengine\tools\devappserver2\..\..\..\..\lib\cacerts\urlfetch_cacerts.txt missing; with
out this urlfetch will not be able to validate SSL certificates.
INFO     2025-04-02 08:18:41,834 <string>:391] Starting API server at: http://localhost:55845
INFO     2025-04-02 08:18:41,882 instance_factory.py:127] Detected python version "b'Python 3.12.9\r\n'" for runtime "py
thon310" at "C:\Users\Student\AppData\Local\Microsoft\WindowsApps\PythonSoftwareFoundation.Python.3.12_qbz5n2kfra8p0\pyt
hon.exe".
INFO     2025-04-02 08:18:48,153 instance_factory.py:280] Using pip to install dependency libraries; pip stdout is redir
ected to C:\Users\Student\AppData\Local\Temp\tmp7ylodc2y
INFO     2025-04-02 08:18:48,153 instance_factory.py:310] Running python -m pip install --upgrade pip
Requirement already satisfied: pip in c:\users\student\appdata\local\packages\pythonsoftwarefoundation.python.3.12_qbz5n


INFO     2025-04-02 08:18:49,703 instance_factory.py:310] Running C:\Users\Student\AppData\Local\Temp\tmpscmnvfxx\Script
s\pip install -r requirements.txt
Collecting MarkupSafe>=2.0 (from Jinja2>=3.1.2->flask->-r requirements.txt (line 3))
[notice] A new release of pip is available: 24.3.1 -> 25.0.1
[notice] To update, run: python.exe -m pip install --upgrade pip

INFO     2025-04-02 08:18:54,560 dispatcher.py:267] Starting module "default" running at: http://localhost:8080
INFO     2025-04-02 08:18:54,569 admin_server.py:67] Starting admin server at: http://localhost:8000
INFO     2025-04-02 08:18:55,572 instance.py:559] Cannot decide GOOGLE_CLOUD_PROJECT, using "test" as a fake value
 * Serving Flask app 'main'
 * Debug mode: off
WARNING: This is a development server. Do not use it in a production deployment. Use a production WSGI server instead.
 * Running on all addresses (0.0.0.0)
 * Running on http://127.0.0.1:8080
 * Running on http://192.168.2.65:8080
Press CTRL+C to quit
```

6.2 type on browser 192.168.2.65:8080

Step 7: Deploy the app to Google Cloud

7.1 Deploy your app:

gcloud app deploy

```
C:\flask-app>gcloud app deploy
```

7.2 Select any region by pressing the relevant button

```
C:\Windows\System32\cmd.e

 [1] asia-east1    (supports standard and flexible)
 [2] asia-east2    (supports standard and flexible and search_api)
 [3] asia-northeast1 (supports standard and flexible and search_api)
 [4] asia-northeast2 (supports standard and flexible and search_api)
 [5] asia-northeast3 (supports standard and flexible and search_api)
 [6] asia-south1   (supports standard and flexible and search_api)
 [7] asia-southeast1 (supports standard and flexible)
 [8] asia-southeast2 (supports standard and flexible and search_api)
 [9] australia-southeast1 (supports standard and flexible and search_api)
 [10] europe-central2 (supports standard and flexible)
 [11] europe-west   (supports standard and flexible and search_api)
 [12] europe-west2  (supports standard and flexible and search_api)
 [13] europe-west3  (supports standard and flexible and search_api)
 [14] europe-west6  (supports standard and flexible and search_api)
 [15] northamerica-northeast1 (supports standard and flexible and search_api)
 [16] southamerica-east1 (supports standard and flexible and search_api)
 [17] us-central    (supports standard and flexible and search_api)
 [18] us-east1      (supports standard and flexible and search_api)
 [19] us-east4      (supports standard and flexible and search_api)
 [20] us-west1      (supports standard and flexible)
 [21] us-west2      (supports standard and flexible and search_api)
 [22] us-west3      (supports standard and flexible and search_api)
 [23] us-west4      (supports standard and flexible and search_api)
 [24] cancel
Please enter your numeric choice:  1

ERROR: (gcloud.app.deploy) INVALID_ARGUMENT: The project must have a billing account attached.

C:\flask-app>
```

If the user has successful billing with the selected region then only we can go to view option with the following command or else an error message is received as shown above

7.3 View the live app:

gcloud app browse

Program 4:

Build a Docker image for a simple Flask web application and push it to your Docker Hub repository.

Prerequisite Software

Download docker desktop using following link

https://www.docker.com/products/docker-desktop/

Step 2:

On successful installation we get the following screen and click ok

Step 3:

Open windows PowerShell and install wsl

Command: wsl --install

Step 4:

Docker is installed and added in the desktop

Step 5:

Click on the docker icon and click on accept shown in the below screen

Step 6:

Click on skip

Click on skip shown in the below screen

Step 7:

Docker desktop will be running in show hidden icon

Step 8:

Click on the docker and below screen is displayed

Step 9:

Click on Docker hub as shown in the below image

Step 10:

Sign up by creating account or sign up by google

Step 11:

Sign in on Docker desktop app as well

Step 12:

After sign in click on open Docker desktop

Step 13:

Below screen will be seen after clicking open Docker desktop

Implementation Steps

Step 14:

Create a new folder “flask-app” in C drive
Create a simple python file inside flask-app “app.py”
App.py

```
from flask import Flask

app = Flask(__name__)

@app.route('/')
def home():
    return "Hello, Docker & Kubernetes!"

if __name__ == '__main__':
    app.run(host='0.0.0.0', port=5000)
```

Step 15:

Create a new text file in “flask-app” has “requirement.txt”

```
flask
```

Step 16:

Inside “flask-app” Create a Dockerfile should be in file format and write the below code in Dockerfile

```
# Use an official Python runtime as a parent image
FROM python:3.9

# Set the working directory
WORKDIR /app

# Copy the current directory contents into the container
COPY . /app

# Install the dependencies
RUN pip install --no-cache-dir -r requirements.txt

# Expose port 5000 for the Flask application
EXPOSE 5000

# Define the command to run the application
CMD ["python", "app.py"]
```

Step 17:

Create Build a Docker image by Clicking Terminal inside the Desktop docker

```
PS C:\> cd flask-app
PS C:\flask-app> docker build -t flask-app .
```

Step 18:

To check the images

Here “flask-app” is our image

```
PS C:\Users\Admin> docker images
REPOSITORY   TAG       IMAGE ID       CREATED          SIZE
flask-app    latest    20dc1143c938   6 minutes ago    1.46GB
PS C:\Users\Admin>
```

Step 19:

Run a Docker image in container

Terminal

```
PS C:\flask-app> docker run -p 5000:5000 -d flask-app
98b8ef7e12b8263558fee8f6f8bfb63368ded0552b01eb022645d380584bd431
PS C:\flask-app>
```

To check image is running in container docker ps -a is the command

```
PS C:\flask-app> docker ps -a
CONTAINER ID   IMAGE      COMMAND           CREATED          STATUS          PORTS                    NAMES
98b8ef7e12b8   flask-app  "python app.py"   59 seconds ago   Up 58 seconds   0.0.0.0:5000->5000/tcp   lucid_blackwell
PS C:\flask-app>
```

Step 20:

Type http://localhost:5000 on your browser

Hello, Docker & Kubernetes!

Steps to push image to the Docker Hub

Step 21:

Push the image to the Docker hub

a) To Tag the image use below command line

```
docker tag my-image your-dockerhub-username/my-image:latest
```

Type the command in the Terminal

Name	Tag	Image ID	Created	Size	Actions
flask-app	latest	d1605ee7171e	15 minutes ag	1.46 GB	
pras0406/flask-app	latest	d1605ee7171e	15 minutes ag	1.46 GB	

b) Command to push a image to docker hub

```
docker push your-dockerhub-username/my-image:latest
```

Terminal

```
PS C:\flask-app> docker push pras0406/flask-app:latest
The push refers to repository [docker.io/pras0406/flask-app]
32fd46cb499e: Waiting
32fd46cb499e: Pushed
f299e0671245: Pushed
```

c) Verify the image in docker hub

Step 22:

To pull the image from Docker hub

```
docker pull your-dockerhub-username/my-image:latest
```

Program 7:

Find vulnerabilities, misconfigurations and secrets in files, Docker, Docker Hub and Git Repo using Trivy . and generate a summary report in HTML format.

Install Trivy on Windows Using Scoop

Scoop is a Windows package manager that makes it easy to install CLI tools.

Step 1: Install Scoop

Open **PowerShell** as Administrator and run:

```
Set-ExecutionPolicy RemoteSigned -scope CurrentUser

iwr -useb get.scoop.sh | iex
```

Step 2: Install Trivy using Scoop

```
scoop install trivy
```

Step 3: Verify installation

```
trivy  --version
```

scenario-1 : Secrets

Scenario-2 : Misconfigurations

*Dockerfile - Notepad

```
# Use an official Python runtime as a parent image
FROM python:3.9

# Set the working directory inside the container
WORKDIR /app

# Copy app files
COPY . /app

# Install the dependencies
RUN pip install -r requirements.txt

# Expose port 5000 for the Flask app
EXPOSE 5000

# Run the Flask app
CMD ["python", "app.py"]
```

Ln 7, Col 17 | 100% | Windows (CRLF) | UTF-8

*Dockerfile - Notepad

```
FROM python:3.9
# Create a non-root user
RUN useradd -m appuser

# Set working directory
WORKDIR /app

#Copy the current directory contents into the container
COPY . /app

# Install dependencies
RUN pip install --no-cache-dir -r requirements.txt

# Add HEALTHCHECK
HEALTHCHECK --interval=30s --timeout=10s --start-period=5s --retries=3 \
  CMD curl --fail http://localhost:5000/health || exit 1

# Run as non-root user
USER appuser

EXPOSE 5000

# Start Flask app
CMD ["python", "app.py"]
```

Ln 2, Col 1 | 100% | Windows (CRLF) | UTF-8

Scenario-3 : Vulnerabilities

requirement...

```
flask==2.3.0
```

Windows (CRLF) | UTF-8

*requirements...

```
flask
```

1 Windows (CRLF) | UTF-8

Report Summary in CLI Mode

Note 1: Create a directory named `trivy in C drive`. Within this directory, create the following three files: `app.py`, `Dockerfile` , and `requirements.txt`."

To scan all files for secrets, misconfigurations and vulnerabilities

EX:

```
trivy fs --security-checks secret,config,vuln .
```

```
C:\Windows\System32\cmd.exe
Microsoft Windows [Version 10.0.19045.5737]
(c) Microsoft Corporation. All rights reserved.

C:\Trivy>trivy fs --security-checks secret,config,vuln .
```

Report Summary in CLI Mode

Report Summary

Target	Type	Secrets	Misconfigurations	Vulnerabilities
requirements.txt	pip	-	-	1
Dockerfile	dockerfile	-	2	-
app.py	text	1	-	-

```
app.py (secrets)
================
Total: 1 (UNKNOWN: 0, LOW: 0, MEDIUM: 0, HIGH: 0, CRITICAL: 1)

CRITICAL: Stripe (stripe-secret-token)
==============================================
Stripe Secret Key
----------------------------------------------
 app.py:8
----------------------------------------------
   6
   7   # API Key stored directly in code (SECRET)
   8 [ API_KEY = "************************"
   9
----------------------------------------------
```

```
Dockerfile (dockerfile)
=======================
Tests: 28 (SUCCESSES: 26, FAILURES: 2)
Failures: 2 (UNKNOWN: 0, LOW: 1, MEDIUM: 0, HIGH: 1, CRITICAL: 0)

AVD-DS-0002 (HIGH): Specify at least 1 USER command in Dockerfile with non-root user as argument
==================================================
Running containers with 'root' user can lead to a container escape situation. It is a best practice to run containers as n
USER' statement to the Dockerfile.

See https://avd.aquasec.com/misconfig/ds002
--------------------------------------------------


AVD-DS-0026 (LOW): Add HEALTHCHECK instruction in your Dockerfile
==================================================
You should add HEALTHCHECK instruction in your docker container images to perform the health check on running containers.

See https://avd.aquasec.com/misconfig/ds026
--------------------------------------------------
```

```
requirements.txt (pip)
======================
Total: 1 (UNKNOWN: 0, LOW: 0, MEDIUM: 0, HIGH: 1, CRITICAL: 0)
```

Library	Vulnerability	Severity	Status	Installed Version	Fixed Version
flask	CVE-2023-30861	HIGH	fixed	1.0.2	2.3.2, 2.2.5

Note for Individual Scan

Individual Scan for secret

```
trivy fs --security-checks secret .
```

Individual Scan for misconfigurations

```
trivy fs --security-checks config .
```

Individual Scan for vulnerabilities

```
trivy fs --security-checks vuln .
```

To generate a Summary Report in HTML format

Note2: First, download the original HTML template from Trivy:

☞ https://github.com/aquasecurity/trivy/blob/main/contrib/html.tpl

Click **"Raw"** → Right-click → **"Save As"** → Save as `html.tpl`. inside of **Trivy** Folder

Run in CLI Mode to generate the Report Summary in Trivy Folder

EX:

trivy fs --format template --template "@html.tpl" -o report.html `--security-checks secret,config,vuln .`

```
C:\Trivy>trivy fs --format template --template "@html.tpl" -o  report.html  --security-checks secret,config,vuln .
```

Output

Check the report.html and open it

Trivy Security Scan Report

Report Summary for Target: requirements.txt

Target	Type	Vulnerabilities	Misconfigurations	Secrets
requirements.txt	pip	1	0	0

Vulnerabilities

Package	Vulnerability ID	Severity	Installed Version	Fixed Version	Title
flask	CVE-2023-30861	HIGH	1.0.2	2.3.2, 2.2.5	flask: Possible disclosure of permanent session cookie due to missing Vary: Cookie header

Report Summary for Target: Dockerfile

Target	Type	Vulnerabilities	Misconfigurations	Secrets
Dockerfile	dockerfile	0	2	0

Vulnerabilities

No vulnerabilities found

Misconfigurations

- DS002: Specify at least 1 USER command in Dockerfile with non-root user as argument
- DS026: Add HEALTHCHECK instruction in your Dockerfile

Report Summary for Target: app.py

Target	Type	Vulnerabilities	Misconfigurations	Secrets
app.py		0	0	1

Vulnerabilities

No vulnerabilities found

Secrets

- stripe-secret-token at

Note for **Individual Scan**

Individual Scan for secret

trivy fs --format template --template "@html.tpl" -o secret-report.html --security-checks secret .

Individual Scan for misconfigurations

trivy fs --format template --template "@html.tpl" -o misscon-report.html --security-checks config .

Individual Scan for vulnerabilities

trivy fs --format template --template "@html.tpl" -o vuln-report.html --security-checks vuln .

Scan a Docker image

To scan Docker Image for secrets, misconfigurations and vulnerabilities

Note: Make sure Docker Desktop is installed and running if you're scanning Docker images.

Example:
Docker image is: trivy:latest

Report Summary in CLI Mode

---

full scan for vulnerabilities, misconfigurations and secrets

trivy image --scanners vuln,secret,misconfig trivy:latest

Individual Scan for secret

trivy image --scanners secret trivy:latest

Individual Scan for misconfigurations

trivy image --scanners misconfig trivy:latest

Individual Scan for vulnerabilities

trivy image --scanners vuln trivy:latest

To generate a Summary Report in HTML format

---

full scan for vulnerabilities, misconfigurations and secrets

trivy image --format template --template "@html.tpl" -o docfullscan.html --scanners vuln,secret,misconfig trivy:latest

Individual Scan for secret

trivy image --format template --template "@html.tpl" -o docsec.html --scanners secret trivy:latest

Individual Scan for misconfigurations

trivy image --format template --template "@html.tpl" -o docmiss.html --scanners misconfig trivy:latest

Individual Scan for vulnerabilities

trivy image --format template --template "@html.tpl" -o docvul.html --scanners vuln trivy:latest

Scan a Docker Hub image

To scan Docker Hub Image for secrets, misconfigurations and vulnerabilities

Example:

Docker hub image is: shwethapmurthy/trivy:latest

Report Summary in CLI Mode

---

full scan for vulnerabilities, misconfigurations and secrets

trivy image --scanners vuln,secret,misconfig shwethapmurthy/trivy:latest

Individual Scan for secret

trivy image --scanners secret shwethapmurthy/trivy:latest

Individual Scan for vulnerabilities

trivy image --scanners vuln **shwethapmurthy/trivy:latest**

Individual Scan for misconfigurations

trivy image --scanners misconfig shwethapmurthy/trivy:latest

To generate a Summary Report in HTML format

---

full scan for vulnerabilities, misconfigurations and secrets

trivy image --format template --template "@html.tpl" -o dhfullscan.html --scanners vuln,secret,misconfig shwethapmurthy/trivy:latest

Individual Scan for secret

trivy image --format template --template "@html.tpl" -o dhubvul.html --scanners vuln shwethapmurthy/trivy:latest

Individual Scan for misconfigurations

trivy image --format template --template "@html.tpl" -o dhubsec.html --scanners secret shwethapmurthy/trivy:latest

Individual Scan for vulnerabilities

trivy image --format template --template "@html.tpl" -o dhubmis.html --scanners misconfig shwethapmurthy/trivy:latest

Scanning a Remote Git Repository

To scan a public Git repository, use the following command:

This command enables scanning for vulnerabilities, secrets, and misconfigurations

---

Report Summary in CLI Mode

Ex:

trivy repo --scanners secret,misconfig,vuln https://github.com/ParashivamurthyC/Trivy-Security

```
C:\trivy>trivy repo --scanners secret,misconfig,vuln https://github.com/ParashivamurthyC/Trivy-Security
```

```
Report Summary
```

Target	Type	Secrets	Misconfigurations	Vulnerabilities
requirements.txt	pip	-	-	1
Dockerfile	dockerfile	-	2	-
app.py	text	1	-	-

To generate a Summary Report in HTML format

EX:

trivy repo --format template --template "@html.tpl" -o git.html --scanners vuln,secret,misconfig https://github.com/ParashivamurthyC/Trivy-Security

```
C:\trivy>trivy  repo --format template --template "@html.tpl" -o git.html  --scanners vuln,secret,misconfig
https://github.com/ParashivamurthyC/Trivy-Security
```

Program 8:

Simulate and manage Amazon S3 buckets and EC2 using LocalStack locally. Perform operations using both the LocalStack Cloud Dashboard and AWS CLI.

Prerequisite Software

1. **LocalStack Cloud**
2. **LocalStack**
3. **Docker**
4. **AWS CLI**

Step1: **create an account** on AWS LocalStack Cloud

Sign Up for LocalStack Cloud

https://app.localstack.cloud/sign-up

https://app.localstack.cloud/dashboard

LocalStack Cloud - UI dashboard

Check if LocalStack Cloud (Web UI) is Offline

Select "Local Instance"

Note: Now It is **offline**

Step2: **install LocalStack** on your system using **Python (pip)**:

pip install localstack-core

```
C:\Users\Admin>pip install localstack
```

Step3: Run Docker (No Sign-in Required)

Note: Make sure Docker is installed and running .

Step4: start LocalStack

```
C:\Users\Admin>localstack start
```

ℹ Note:

- Running LocalStack with Docker will **pull the LocalStack image** (if not already downloaded) and **start it inside a Docker container**.
- **Do not close the terminal** where LocalStack is running — that container will stop if you do.

Step5: Check if LocalStack Cloud (Web UI) is Online

https://app.localstack.cloud/dashboard

If you're using **https://app.localstack.cloud/dashboard** and **refresh the browser once**

Note : Now It is online

Make sure LocalStack is **running with HTTPS support**

https://localhost.localstack.cloud:4566

Simulate S3 Buckets using AWS LocalStack Cloud Dashboard

Step 1: Open the Dashboard

https://app.localstack.cloud/dashboard

Step 2: Select "Local Instance"

Step 3: Open the "Resource Browser"

Click on S3 from the services menu.

Click Create Bucket

Enter a bucket name

View S3 Buckets

To upload a file to a specific S3 bucket

view files (objects) inside an S3 bucket

delete a bucket

Note :, **you must first delete all objects and folders inside

---

Simulate S3 Buckets Using AWS CLI with LocalStack

Prerequisites - Install the AWS CLI version

```
pip3 install awscli
```

```
C:\Users\Admin>pip3 install awscli
```

```
C:\Users\Admin>localstack --version

C:\Users\Admin>python "C:\Users\Admin\AppData\Local\Programs\Python\Python310\Scripts\\localstack" --version
3.6.0
```

Prerequisites - configure dummy credentials using aws configure

AWS CLI configuration prompt: it's used to set up your AWS credentials and default settings.

aws configure

```
C:\Users\Admin>aws configure
```

AWS Access Key ID [None]: test

AWS Secret Access Key [None]: test

Default region name [None]: us-east-1

Default output format [None]:

Create a bucket

aws --endpoint-url=http://localhost:4566 s3 mb s3://dept-cse

```
C:\Users\Parashivamurthy>aws --endpoint-url=http://localhost:4566 s3 mb s3://dept-cse
make_bucket: dept-cse
```

List All Buckets

aws --endpoint-url=http://localhost:4566 s3 ls

```
C:\Users\Parashivamurthy>aws --endpoint-url=http://localhost:4566 s3 ls
2025-05-14 05:00:25 dept-cse
```

upload file to a specific "folder"

step1: create a text file

echo "Hello LocalStack" > test.txt

```
C:\Users\Parashivamurthy>echo "Hello LocalStack" > test.txt
```

Step2: upload / copy a file to dept-cse

aws --endpoint-url=http://localhost:4566 s3 cp test.txt s3://dept-cse

```
C:\Users\Parashivamurthy>aws --endpoint-url=http://localhost:4566 s3 cp test.txt s3://dept-cse
upload: .\test.txt to s3://dept-cse/test.txt
```

Delete File and Bucket

aws s3 rm s3://dept-cse --recursive --endpoint-url=http://localhost:4566

```
C:\Users\Parashivamurthy>aws s3 rm s3://dept-cse  --recursive --endpoint-url=http://localhost:4566
delete: s3://dept-cse/test.txt
```

Simulate EC2 Using AWS CLI with LocalStack

Create a Key Pair

aws --endpoint-url=http://localhost:4566 ec2 create-key-pair --key-name test-key

Command Prompt

```
C:\Users\Parashivamurthy>aws --endpoint-url=http://localhost:4566 ec2 create-key-pair --key-name test-key
{
    "KeyPairId": "key-5b5700c331bd652f7",
    "KeyName": "test-key",
    "KeyFingerprint": "6e:41:a4:dd:8a:ec:79:a7:cd:52:e3:a7:7a:4a:0a:85:36:79:dd:67",
    "KeyMaterial": "-----BEGIN RSA PRIVATE KEY-----\nMIIEpAIBAAKCAQEAuPRAAqlpeTwMLQCk/51VFEipmhLLUte0XxkHS+fKEp+M0GMv\ng
j52wgzeTKli5gn1XdxYkwBQDgZmfKBN0YMZqWzgqfKBGTL/9tvKEbGPxxfY0BgF\ny/QkVHg0lU3sxKz8qOy3D8BJwyW2BVZE/6flaIyHNXu5uLIvCtQaNKN
lztxKNGDw\nFrgT4VjQqYudcV5TJ7Y5yWubh73aNjWFrdCRfb7NjCfZEhwzvnWagDkNiYY/ufkK\nRSCykdzOwbC57mx5axOmley2V19mSzGW1lbRZCfTdTO
NRzpGEn+KQi2NFpOO3/vc\n786BFw8yYbpp21/gv19vaShxlluzNjbfU+/0JQIDAQABAoIBAATxZiqMC4kVbT/g\npgg81LLDlGDNQsjWgfgOq67AJZrkz7c
cB3+EGtlpzaEf38hFGN0mz+kx13Hggumu\nwRWp2sw34DM6i2+4eLXIvvhFFhIclYtOMQ1322a0hbc9y2egNepv1R6jJXWFMv2m\nhhaNDqkCJHTt9xM0E+T
GsWKZwVVLaxmEGA9gYWsbUa9xX8FiEKOSupzs8SDF/5rr\nlJuEMtgngNt0FydJnHgtWPUlKnYbAm5YKnhMo267ndhM17n7PX4UNfuD6HkDuUEM\nrSQIfyQ
Qyh20Xzxg8Fl2Dtjamdjuusotmwwl U+IIPvTZdSzfNcZ3kWNCL9wOKs0y\nmaUqZLsCgYEA/+prtMie83A2aJxR6B6D3xIIUuIdlQ8FoKonN09eU930PpL2X
qSP\nfV/iBUIx99y9mdttPUULl7ybb6Ok1smI2i1dWuC3Pp7VGndn5BEDX3uOiqAcq1Sr\nX8oy3xKS4U+/gt9A+Rcs2nhbgH/TNMdztDgCDc87CNBzHaCBk
ssPi58CgYEAuQPY\ngAE5+D/qrThoOz36iE7GNeFzSdHunT2KZGzMXnVtTnMr+0lDYBe3J0H0udi1kZMK\nQbbpzrPHBh4qSOXAsYmc2tbEx5vdM/aVkHcu4
TNeyzVWeJotvhpYdm9WdS5Mc1ob\n1bD3lsgxUkeR/gAPvhQj42rHnAo8EBZFSjr3qbsCgYBiNVXL9Ml65m8ayhJsn3k/\n262yEPAFSdSBCYsvIJg0WxLmC
IPfYgtms6sLFZITanSgXKKZRqviORD1ULlra7/G\n6m3O+DmaIF3rYDNQRG8k70eOT09xOHbjtba8AGa5bF80G/TVIl2mIVYMBAHLWVc/\nq9at8f7XrRbgI
b5X1PFMRQKBgQCSdpnoU5A5lxfHgxT4eQsg5wOAWfZzZO55qeMp\ntvted6VNal1R82CT7JJAULVUflrWYEnJVeznhmgyzqR4yvDtWtxCXayvvBKz9SvX\nj
pIdExai1Zmbpao6iQ6oy2DtKPa6hoXsd8i4o8hvs1q+cfwEZFs4yCo2u99qeMbn\niaDPywKBgQDZoRz/DyjPksLH3ENvFytdJ5meyO8zcKSmumuQC4j0P98
nF+MzfKNl\n6uPVhuEkOfsUMhWp+jeg4CJZWRHw2vhsOsmHbnfnxNuLQIhSRyPM0U7Gf8tRSwfV\nHEkt61GgqZP9qyucoSanJOPGLmb8deIwzx5iIBgPxoY
JOcDfxbZeHw==\n-----END RSA PRIVATE KEY-----\n"
}
```

Run an EC2 Instance

aws --endpoint-url=http://localhost:4566 ec2 run-instances --image-id ami-12345678 --instance-type t2.micro --key-name test-key

```
C:\Users\Parashivamurthy>aws --endpoint-url=http://localhost:4566 ec2 run-instances
--image-id ami-12345678 --instance-type t2.micro --key-name test-key
```

```
C:\Users\Parashivamurthy>aws --endpoint-url=http://localhost:4566 ec2 run-instances --image-id ami-12345678
 --instance-type t2.micro --key-name test-key
{
    "ReservationId": "r-90b571ff1d3739151",
    "OwnerId": "000000000000",
    "Groups": [
        {
            "GroupId": "sg-245f6a01",
            "GroupName": "default"
        }
```

Note :

image-id is OS id

Ex: Ubuntu 20.04 LTS, AWS gives the Image ID: ami-042e8287309f5df03

These IDs **change across regions** and **are maintained by AWS or the OS provider**

`t2.micro` is a **small, general-purpose** instance type (1 vCPU, 1 GiB RAM).

SSH key pair used to connect to the instance

List Running EC2 Instances

aws --endpoint-url=http://localhost:4566 ec2 describe-instances

```
},
"VirtualizationType": "paravirtual",
"InstanceId": "i-283153368cc22fb2c",
"ImageId": "ami-12345678",
"State": {
    "Code": 16,
    "Name": "running"
```

Note: **identify the `InstanceId`** to stop & Terminate the EC2 instance.

Example : **`InstanceId:`** i-283153368cc22fb2c

Stop the EC2 Instance

Note:Before stopping an EC2 instance, you **must first identify the `InstanceId`**.

aws --endpoint-url=http://localhost:4566 ec2 stop-instances --instance-ids i-283153368cc22fb2c

```
Command Prompt
C:\Users\Parashivamurthy>aws --endpoint-url=http://localhost:4566 ec2 stop-instances --instance-ids i-28315
3368cc22fb2c
{
    "StoppingInstances": [
        {
            "InstanceId": "i-283153368cc22fb2c",
            "CurrentState": {
                "Code": 64,
                "Name": "stopping"
            },
            "PreviousState": {
                "Code": 16,
                "Name": "running"
            }
        }
    ]
}
```

Terminate the EC2 Instance

Note:Before stopping an EC2 instance, you **must first identify the `InstanceId`**.

aws --endpoint-url=http://localhost:4566 ec2 terminate-instances --instance-ids i-283153368cc22fb2c

```
C:\Users\Parashivamurthy>aws --endpoint-url=http://localhost:4566 ec2 terminate-instances --instance-ids
i-283153368cc22fb2c
{
    "TerminatingInstances": [
        {
            "InstanceId": "i-283153368cc22fb2c",
            "CurrentState": {
                "Code": 32,
                "Name": "shutting-down"
            },
            "PreviousState": {
                "Code": 80,
                "Name": "stopped"
            }
        }
    ]
}
```

Delete Key Pair

Step1: Use this command to **list all key pairs**

aws --endpoint-url=http://localhost:4566 ec2 describe-key-pairs

```
C:\Users\Parashivamurthy>aws --endpoint-url=http://localhost:4566 ec2 describe-key-pairs
{
    "KeyPairs": [
        {
            "KeyPairId": "key-ece9dc10d880b1063",
            "CreateTime": "2025-05-14T00:31:05.962000+00:00",
            "KeyName": "test-key",
            "KeyFingerprint": "81:53:77:d0:5b:af:ab:7f:ed:ad:69:96:4d:d2:da:06:ca:73:88:fd"
        }
    ]
}
```

Note: To **delete a key pair** in AWS (or LocalStack), you must **first identify the key pair name**.

Step2: Delete the Key Pair

aws --endpoint-url=http://localhost:4566 ec2 delete-key-pair --key-name test-key

```
C:\Users\Parashivamurthy>aws --endpoint-url=http://localhost:4566 ec2 delete-key-pair --key-name test-key
{
    "Return": true
}
```

Simulate EC2 using AWS LocalStack Cloud Dashboard

Step 1: Open the Dashboard

https://app.localstack.cloud/dashboard

Step 2: Select "Local Instance"

Step 3: Open the "Resource Browser"

Step 4: Type `ec2` in the Search Bar

Step 5: Click on EC2

Click **"Launch Instance"**

fill in:

- **Image ID**: `ami-12345678`
- **Instance Type**: `t2.micro`
- **Key Name**: `test-key`

Click the Submit Button

view your EC2 instance

EC2 Instance Actions in LocalStack Cloud Dashboard

3 Actions :1. Stop

2. Start

3. Terminate

1. **Stop** Instance

2. **Start** Instance

3. **Terminate** Instance

Program 9:

Simulate and manage a Amazon DynamoDB Using AWS CLI with AWS LocalStack Cloud.

prerequisites:

1. **LocalStack Cloud**
2. **Python 3.6.9**
3. **LocalStack**
4. **Docker**
5. **AWS CLI**

Step1: **Sign Up for LocalStack Cloud**

create an account on LocalStack Cloud

https://app.localstack.cloud/sign-up

Step2: **Sign Up for LocalStack Cloud**

https://app.localstack.cloud/sign-in

Step3: **install LocalStack** on your system usining **Python (pip)**:

pip install localstack

```
C:\Users\Admin>pip install localstack
```

Step4: Run Docker (No Sign-in Required)

Note: Make sure Docker is installed and running .

Step5: start LocalStack

```
C:\Users\Admin>localstack start
```

ℹ Note:

- Running LocalStack with Docker will **pull the LocalStack image** (if not already downloaded) and **start it inside a Docker container**.
- **Do not close the terminal** where LocalStack is running — that container will stop if you do.

Step6: Check if LocalStack Cloud (Web UI) is Online

Note :

Make sure LocalStack is **running with HTTPS support**

https://localhost.localstack.cloud:4566

Prerequisites - Install the AWS CLI version

```
pip3 install awscli
```

```
C:\Users\Admin>pip3 install awscli
```

```
C:\Users\Admin>localstack --version

C:\Users\Admin>python "C:\Users\Admin\AppData\Local\Programs\Python\Python310\Scripts\\localstack" --version
3.6.0
```

Prerequisites - configure dummy credentials using aws configure

aws configure

```
C:\Users\Admin>aws configure
```

AWS Access Key ID: test

AWS Secret Access Key: test

Default region name: us-east-1

Default output format: json

Implementation Steps

AWS CLI DynamoDB Actions			
#	Action	CLI Command	Description
1	Create Table	create-table	Create a new DynamoDB table
2	List Tables	list-tables	List all DynamoDB tables
3	Describe Table	describe-table	Show metadata (status, schema) of a table
4	Put Item	put-item	Insert a new item (record)
5	Get Item	get-item	Fetch an item by primary key
6	Update Item	update-item	Update specific attribute(s) of an item
7	Delete Item	delete-item	Delete a specific item
8	Delete Attribute	update-item with REMOVE	Remove a specific attribute from an item
9	Scan	scan	Read all items (like SELECT * in SQL)

1. Create a DynamoDB Table

```
aws --endpoint-url=http://localhost:4566 dynamodb create-table ^
  --table-name Students ^
  --attribute-definitions AttributeName=USN,AttributeType=S ^
  --key-schema  AttributeName=USN,KeyType=HASH  ^
  --provisioned-throughput  ReadCapacityUnits=5,WriteCapacityUnits=5
```

```
    "TableDescription": {
        "AttributeDefinitions": [
            {
                "AttributeName": "USN",
                "AttributeType": "S"
            }
        ],
        "TableName": "Students",
        "KeySchema": [
            {
                "AttributeName": "USN",
                "KeyType": "HASH"
            }
        ],
        "TableStatus": "ACTIVE",
        "CreationDateTime": "2025-06-06T10:41:41.079000+05:30",
        "ProvisionedThroughput": {
            "NumberOfDecreasesToday": 0,
            "ReadCapacityUnits": 5,
            "WriteCapacityUnits": 5
        },
```

2. List Tables

aws --endpoint-url=http://localhost:4566 dynamodb list-tables

```
C:\Users\Admin>aws --endpoint-url=http://localhost:4566 dynamodb list-tables
{
    "TableNames": [
        "Students"
    ]
}
```

3. Insert Data(PutItem)

Ex:1

aws --endpoint-url=http://localhost:4566 dynamodb put-item ^

--table-name Students ^

--item "{\"USN\": {\"S\": \"1ga25cs001\"}, \"Name\": {\"S\": \"Ramesh\"}, \"Address\": {\"S\": \"Bangalore\"}}"

Ex:2

aws --endpoint-url=http://localhost:4566 dynamodb put-item ^

--table-name Students ^

--item "{\"USN\": {\"S\": \"1ga25cs002\"}, \"Name\": {\"S\": \"Naresh\"}, \"age\": {\"N\": \"40\"}, \"Address\": {\"S\": \"Bangalore\"}}"

4. Retrieve Data (GetItem)

Ex1:

aws --endpoint-url=http://localhost:4566 dynamodb get-item ^

--table-name Students ^

--key "{\"USN\": {\"S\": \"1ga25cs001\"}}"

```
C:\Users\Admin>aws --endpoint-url=http://localhost:4566 dynamodb get-item ^
More?     --table-name Students ^
More?     --key "{\"USN\": {\"S\": \"1ga25cs001\"}}"
{
    "Item": {
        "Address": {
            "S": "Bangalore"
        },
        "USN": {
            "S": "1ga25cs001"
        },
        "Name": {
            "S": "Ramesh"
        }
    }
}
```

Ex2:

aws --endpoint-url=http://localhost:4566 dynamodb get-item ^

--table-name Students ^

--key "{\"USN\": {\"S\": \"1ga25cs002\"}}"

```
C:\Users\Admin>aws --endpoint-url=http://localhost:4566 dynamodb get-item ^
More?     --table-name Students ^
More?     --key "{\"USN\": {\"S\": \"1ga25cs002\"}}"
{
    "Item": {
        "Address": {
            "S": "Bangalore"
        },
        "USN": {
            "S": "1ga25cs002"
        },
        "Name": {
            "S": "Ramesh"
        }
    }
}
```

5. To Update / Rename item (i.e., a record)

Note: you have to copy value to new key, then remove old:

Let's say you want to update the **Address** of the student with **USN = "1ga25cs001"**.

```
aws --endpoint-url=http://localhost:4566 dynamodb update-item ^
  --table-name Students ^
 --key "{\"USN\": {\"S\": \"1ga25cs001\"}}" ^
  --update-expression "SET Address = :newaddress" ^
  --expression-attribute-values "{\":newaddress\": {\"S\": \"Mysore\"}}"
```

```
C:\Users\Admin>aws --endpoint-url=http://localhost:4566 dynamodb update-item ^
More?     --table-name Students ^
More?     --key "{\"USN\": {\"S\": \"1ga25cs001\"}}" ^
More?     --update-expression "SET Address = :newaddress" ^
More?     --expression-attribute-values "{\":newaddress\": {\"S\": \"Mysore\"}}"
```

Note :To check whether value has been updated or not

```
aws --endpoint-url=http://localhost:4566 dynamodb get-item ^
  --table-name Students ^
  --key "{\"USN\": {\"S\": \"1ga25cs001\"}}"
```

```
C:\Users\Admin>aws --endpoint-url=http://localhost:4566 dynamodb get-item ^
More?     --table-name Students ^
More?     --key "{\"USN\": {\"S\": \"1ga25cs001\"}}"
{
    "Item": {
        "Address": {
            "S": "Mysore"
        },
        "USN": {
            "S": "1ga25cs001"
        },
        "Name": {
            "S": "Ramesh"
        }
    }
}
```

6. To **delete an item** (i.e., a record)

```
aws --endpoint-url=http://localhost:4566 dynamodb delete-item ^
 --table-name Students ^
 --key "{\"USN\": {\"S\": \"1ga25cs002\"}}"
```

```
C:\Users\Admin>aws --endpoint-url=http://localhost:4566 dynamodb delete-item ^
More?   --table-name Students ^
More?   --key "{\"USN\": {\"S\": \"1ga25cs002\"}}"
```

Note :To check whether item has been deleted or not

aws --endpoint-url=http://localhost:4566 dynamodb scan ^

--table-name Students ^

--projection-expression "USN" ^

--output json

```
C:\Users\Admin>aws --endpoint-url=http://localhost:4566 dynamodb scan ^
More?   --table-name Students ^
More?   --projection-expression "USN" ^
More?   --output json
{
    "Items": [
        {
            "USN": {
                "S": "1ga25cs001"
            }
        }
    ],
    "Count": 1,
    "ScannedCount": 1,
    "ConsumedCapacity": null
}
```

7. To delete an attribute

EX: Address is an attribute

aws --endpoint-url=http://localhost:4566 dynamodb update-item ^

--table-name Students ^

--key "{\"USN\": {\"S\": \"1ga25cs001\"}}" ^

--update-expression "REMOVE Address"

```
C:\Users\Admin>aws --endpoint-url=http://localhost:4566 dynamodb update-item ^
More?     --table-name Students ^
More?     --key "{\"USN\": {\"S\": \"1ga25cs001\"}}" ^
More?     --update-expression "REMOVE Address"
```

Note:To check whether an attribute has been deleted or not

aws --endpoint-url=http://localhost:4566 dynamodb get-item ^

--table-name Students ^

--key "{\"USN\": {\"S\": \"1ga25cs001\"}}"

```
C:\Users\Admin>aws --endpoint-url=http://localhost:4566 dynamodb get-item ^
More?     --table-name Students ^
More?     --key "{\"USN\": {\"S\": \"1ga25cs001\"}}"
{
    "Item": {
        "USN": {
            "S": "1ga25cs001"
        },
        "Name": {
            "S": "Ramesh"
        }
    }
}
```

5. Delete Table

aws --endpoint-url=http://localhost:4566 dynamodb delete-table --table-name Students

```
C:\Users\Admin>aws --endpoint-url=http://localhost:4566 dynamodb delete-table --table-name Students
```

Note:To check whether a Table has been deleted or not

```
C:\Users\Admin>aws --endpoint-url=http://localhost:4566 dynamodb list-tables
{
    "TableNames": []
}
```