SINHGAD TECHNICAL EDUCATION SOCIETY'S

SINHGAD INSTITUTE OF TECHNOLOGY

Kusgaon (Bk), Lonavala 410401

DEPARTMENT OF INFORMATION TECHNOLOGY

LABORATORY MANUAL V-1.0

Laboratory Practices-II

Part 1: Cloud Computing [314458]

T.E. IT (SEM –II) (2019 Course)

AY 2023-24

Developed By

Prof. Sagar D.Bhopale

TEACHING SCHEME

Practical: 3 Hrs/Week

EXAMINATION SCHEME

Term Work: 50Marks

Practical: 25Marks

Vision and Mission of Institute

VISION

उत्तमपुरुषान् उत्तमाभियंतृन् निर्मातु कटिबद्धाःवयम्

"We are committed to produce not only good engineers but good human beings, also."

MISSION

- We believe in and work for the holistic development of students and teachers.
- We strive to achieve this by imbibing a unique value system, transparent work culture, excellent academic and physical environment conducive to learning, creativity and technology transfer.

Vision and Mission of the Department

VISION

To provide excellent Information Technology education by building teaching and research environment.

MISSION

1) To transform the students into innovative, competent and high quality IT professionals to meet the growing global challenges.
2) To achieve and impart quality education with an emphasis on practical skills and social relevance.
3) To endeavour for continuous up-gradation of technical expertise of students to cater to the needs of the society.
4) To achieve an effective interaction with industry for mutual benefits.

Program Educational Objectives (PEO's)

PEO1	Possess strong fundamental concepts in mathematics, science, engineering and Technology to address technological challenges.
PEO2	Possess knowledge and skills in the field of Computer Science and Information Technology for analyzing, designing and implementin g complex engineering problems of any domain with innovative approaches.
PEO3	Possess an attitude and aptitude for research, entrepreneurship and higher studies in the field of Computer Science and Information Technology.
PEO4	Have commitment ethical practices, societal contributions through communities and life-long learning.
PEO5	Possess better communication, presentation, time management and team work skills leading to responsible & competent professional sand will be able to address challenges in the field of IT at global level.

Program Outcomes: POs

PO1	**Engineering knowledge**	An ability to apply knowledge of mathematics, computing, science, engineering and technology.
PO2	**Problem analysis**	An ability to define a problem and provide a systematic solution with the help of conducting experiments, analyzing the problem and interpreting the data.
PO3	**Design / Development of Solutions**	An ability to design, implement, and evaluate software or a software /hardware system, component, or process to meet desired need switch in realistic constraints.
PO4	**Conduct Investigation of Complex Problems**	An ability to identify, formulates, and provide essay schematic solutions to complex engineering /Technology problems.
PO5	**Modern Tool Usage**	An ability to use the techniques, skills, and modern engineering technology tools, standard processes necessary for practice as a IT

		professional.
PO6	**The Engineer and Society**	An ability to apply mathematical foundations, algorithmic principles, and computer science theory in the modeling and design of computer-based systems with necessary constraints and assumptions.
PO7	**Environment and Sustainability**	An ability to analyze and provide solution for the local and global impact of information technology on individuals, organizations and society.
PO8	**Ethics**	An ability to understand professional, ethical, legal, security and social issues and responsibilities.
PO9	**Individual and Team Work**	An ability to function effectively as an individual or as a team member to accomplish a desired goal(s).
PO10	**Communication Skills**	An ability to engage in life-long learning and continuing professional development to cope up with fast changes in the technologies /tools with the help of electives, profession along animations and extra- curricular activities.
PO11	**Project Management and Finance**	An ability to communicate effectively in engineering community at large by means of effective presentations, report writing, paper publications, demonstrations.
PO12	**Life-long Learning**	An ability to understand engineering, management, financial aspects, performance, optimizations and time complexity necessary for professional practice.

Program Specific Outcomes: PSOs

PSO1	An ability to apply the theoretical concepts and practical knowledge of Information Technology in analysis, design, development and management of information processing systems and applications in the interdisciplinary domain.
PSO2	An ability to analyze a problem, and identify and define the computing infrastructure and operations requirements appropriate to its solution. IT graduates should be able to work on large-scale computing systems.
PSO3	An understanding of professional, business and business processes, ethical, legal, security and social issues and responsibilities.
PSO4	Practice communication and decision-making skills through the use of appropriate technology and be ready for professional responsibilities.

Prerequisites:

- Basics of Computer Networks
- Operating Systems

Course Description :

Cloud computing is the on-demand solution for storing and retrieving data globally. Cloud computing is fast emerging as an essential component for IT-based enterprises and as a whole. Organizations, both big and small, have deployed cloud technology in a suitable capacity.

Course Objectives:

1. To develop web applications in cloud.
2. To learn the design and development process involved in creating a cloud based application.

Course Outcomes:

On completion of the course, students will be able to–

CO1: To design and develop cloud-based applications.

CO2: To Simulate a cloud scenario using Cloud Sim.

CO3: To design and deploy web applications in cloud environment.

Sinhgad Institutes

CERTIFICATE

This is to certify that Mr. /Ms _____ _____of class TEIT Div _____ Roll No._____ Examination Seat No./PRN No._____ has completed all the practical work in the **Laboratory Practices- II(Cloud Computing)** satisfactorily, as prescribed by Savitribai Phule Pune University , Pune in academic year 2023 - 24 (Semester II).

Course In-charge Head of Department Principal

Date:

INDEX [LP-II Lab CC]

SN	Title of experiment	Date of Submission	Marks Obtained (10)	Sign of Faculty
1	Install Google App Engine. Create hello world app and other simple web applications using python/java.			
2	Use GAE launcher to launch the web applications.			
3	Simulate a cloud scenario using Cloud Sim and run a scheduling algorithm that is not present in Cloud Sim.			
4	Find a procedure to transfer the files from one virtual machine to another virtual machine.			
5	Find a procedure to launch virtual machine using try stack (Online Open stack Demo Version)			
6	Design and deploy a web application in a PaaS environment.			
7	Design and develop custom Application (Mini Project) using Salesforce Cloud.			
8	Design an Assignment to retrieve, verify, and store user credentials using Firebase Authentication, the Google App Engine standard environment, and Google Cloud Data store.			

Name & Signature of Course In-charge

INDEX [LP-II Lab CC]

Sr.No	Title of experiment	No. of Hrs.	Week No.
1	Install Google App Engine. Create hello world app and other simple web applications using python/java.	2	WK1
2	Use GAE launcher to launch the web applications.	2	WK2
3	Simulate a cloud scenario using Cloud Sim and run a scheduling algorithm that is not present in Cloud Sim.	2	WK3
4	Find a procedure to transfer the files from one virtual machine to another virtual machine.	2	WK4
5	Find a procedure to launch virtual machine using try stack (Online Open stack Demo Version)	2	WK5
6	Design and deploy a web application in a PaaS environment.	2	WK6
7	Design and develop custom Application (Mini Project) using Salesforce Cloud.	4	WK7, WK8
8	Design an Assignment to retrieve, verify, and store user credentials using Firebase Authentication, the Google App Engine standard environment, and Google Cloud Data store.	4	WK9,WK10

Sinhgad Institutes

Name of the Student: _____

Roll no: _____

CLASS: - T..E. IT

Subject Name: - CC Lab

Assignment No. 01

** Install and explore the Google App Engine : CO1 **

Date of Performance: / /2024

Marks out of 10:

Sign with Date:

Assignment No. 1

Aim: Install Google App Engine. Create hello world app and other simple web applications using python/java.

Objective:

- Installing Google App Engine.

Theory:

What is google app engine?

Google App Engine is a platform by Google Cloud that simplifies web application development and hosting. It manages the infrastructure, automatically scales applications, supports various programming languages like Python and Java, integrates with Google Cloud services, and offers development tools for streamlined deployment. With built-in security features and compliance certifications, it ensures application security and regulatory compliance. Additionally, it provides both standard and flexible environments, catering to different application requirements and allowing developers to focus on coding while Google handles the backend complexities.

STEPS TO FOLLOW :

1. Go to Google.

2. Search for "Google Cloud Console."

3. Sign in.

4. Click the three lines menu on the left.

5. Select "Cloud Overview & Dashboard."

6. Choose "Create New Project" from the dropdown menu.

7. Give your project a name.

8. Leave location and organization as they are.

9. Enable at least one project.

10. Search for "Admin API."

11. Click on "App Engine Admin API."

12. Enable the API.

13. Activate Cloud Shell on the right.

14. Go to GitHub.

15. Log in or sign in.

16. Create a project and add a file.

17. Copy the path of the code file.

18. Keep the URL as HTTPS and copy it.

19. Go back to Cloud Shell.

20. Type "git clone."

21. If you've cloned before, remove the previous one: "rm -rf <directory name>," then "cd <directory name>," and again "rm -rf <directory name>." Use "ls" to confirm it's removed.

22. Clone the GitHub repo with "git clone <link of GitHub repo>."

23. Copy the code.

24. To see files, type "ls."

25. Navigate to the directory with "cd <directory>." For example, ">Python <script.py>" to execute a Python script named "<script.py>" and see "Hello" printed.

OUTPUTS :

<Images of actual implementation>

CONCLUSION :

Completing this assignment will give you hands-on experience with Google Cloud Platform, including Google App Engine and GitHub. You'll learn to navigate cloud services, manage code repositories, and deploy applications, gaining valuable skills for software development and cloud computing roles.

Sinhgad Institutes

Name of the Student: _____

Roll no: _____

CLASS: - T..E. IT

Subject Name: - CC Lab

Assignment No. 02

**** Use GAE launcher to launch the web applications: CO1 ****

Date of Performance:

/ /2024

Marks out of 10:

Sign with Date:

Assignment No. 02

Aim: Use GAE launcher to launch the web applications.

Objective:

Creating and deploying an application on GAE

Theory:

1. Use GAE launcher to launch the web applications

What is google app engine?

Google App Engine is a platform by Google Cloud that simplifies web application development and hosting. It manages the infrastructure, automatically scales applications, supports various programming languages like Python and Java, integrates with Google Cloud services, and offers development tools for streamlined deployment. With built-in security features and compliance certifications, it ensures application security and regulatory compliance. Additionally, it provides both standard and flexible environments, catering to different application requirements and allowing developers to focus on coding while Google handles the backend complexities.

STEPS TO FOLLOW :

1. Go to Google.

2. Search for "Google Cloud ."

3.Go to console.

4. Choose "Create New Project" from the dropdown menu.

5. Give your project a name.

6. Activate Billing account or add Billing.

7. Manage Billing account er dropdown select Biling -> create account.

-> Default ->Continue ->Do credit card.

8. Do not forget to disable Billing account for selected Project.

9. Search APP engine ->select app engine ->select project or create

10.Activate Cloud shell ->in open terminal create folder by typing command mkdir assignment2-> cd assignment2

11.Open editor -> from the drop down menu -> file -> open folder -> select previously created folder assignment2 -> click ok.

12.create new files -> 1.main.py , 2. app.yaml , 3. requirement.txt

In main.py

from flask import Flask

app=Flask(_name_)

@app.route('/')

def hello_world():

return 'my first google app engine app'

In the preceding code block, you first import the Flask object from the flask package. You then use it to create your Flask application instance with the name app. You pass the special variable __name__ that holds the name of the current Python module. It's used to tell the instance where it's located—you need this because Flask sets up some paths behind the scenes.

Once you create the app instance, you use it to handle incoming web requests and send responses to the user. @app.route is a decorator that turns a regular Python function into a Flask *view function*, which converts the function's return value into an HTTP response to be displayed by an HTTP client, such as a web browser. You pass the value '/' to @app.route() to signify that this function will respond to web requests for the URL /, which is the main URL.

The hello() view function returns the string 'Hello, World!' as a response.

In app.yaml

runtime:python39

In requirement.txt

Flask>=2.0

Werkzeug>=2.2

13.open terminal -> gcloud app deploy -v v01 -> Y -> deploy app on cloud -> using URL given in shell

14. type gcloud app browse -> click on URL

OUTPUT:

<Images of actual implementation>

CONCLUSION :

Completing this assignment will give you hands-on experience with Google Cloud Platform, including Google App Engine. You'll learn to navigate cloud services, manage code repositories, and deploy applications, gaining valuable skills for software development and cloud computing roles.

Sinhgad Institutes

Name of the Student: _____

Roll no: _____

CLASS: - T..E. IT

Subject Name: - CC Lab

Assignment No. 03

**** Simulate a cloud scenario using CloudSim and run a scheduling algorithm that is not present in CloudSim: CO2 ****

Date of Performance: / /2024

Marks out of 10:

Sign with Date:

Assignment No. 3

Aim: Simulate a cloud scenario using CloudSim and run a scheduling algorithm that is not present in CloudSim.

Objective:

- Install CloudSim on system.
- Run Scheduling algorithm that is not present in CloudSim.

Theory:

Aim: Simulate a cloud scenario using CloudSim and run a scheduling algorithm that is not present in CloudSim.

Theory:

What is CloudSim?

CloudSim is an open-source framework, which is used to simulate cloud computing infrastructure and services. It is developed by the CLOUDS Lab organization and is written entirely in Java. It is used for modelling and simulating a cloud computing environment as a means for evaluating a hypothesis prior to software development in order to reproduce tests and results.

For example, if you were to deploy an application or a website on the cloud and wanted to test the services and load that your product can handle and also tune its performance to overcome bottlenecks before risking deployment, then such evaluations could be performed by simply coding a simulation of that environment with the help of various flexible and scalable classes provided by the CloudSim package, free of cost.

Benefits of Simulation over the Actual Deployment:

Following are the benefits of CloudSim:

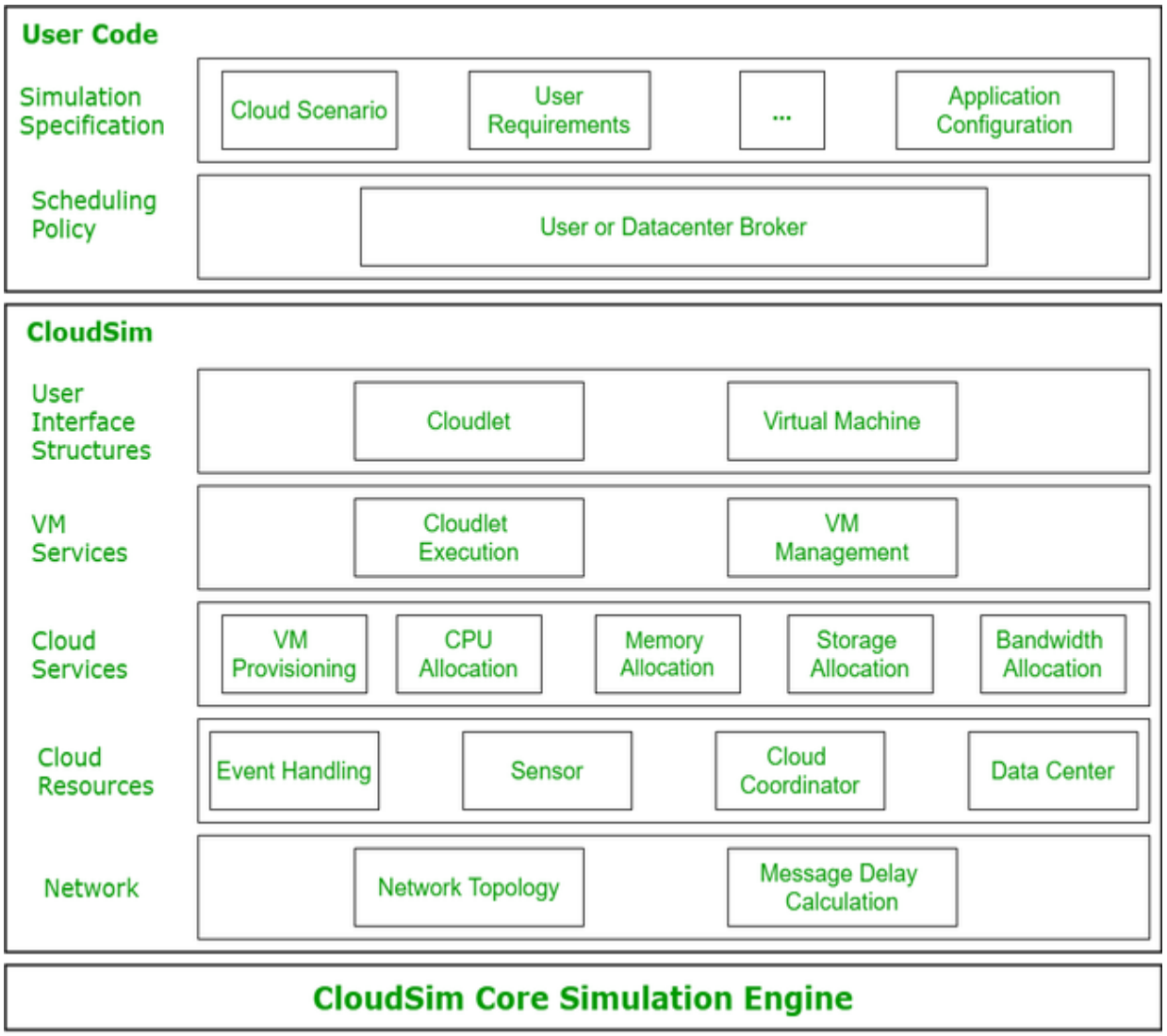
- **No capital investment involved**. With a simulation tool like CloudSim there is no installation or maintenance cost.
- **Easy to use and Scalable**. You can change the requirements such as adding or deleting resources by changing just a few lines of code.
- **Risks can be evaluated at an earlier stage**. In Cloud Computing utilization of real testbeds limits the experiments to the scale of the testbed and makes the reproduction of results an extremely difficult undertaking. With simulation, you can test your product against test cases and resolve issues before actual deployment without any limitations.
- **No need for try-and-error approaches**. Instead of relying on theoretical and imprecise evaluations which can lead to inefficient service performance and revenue generation, you can test your services in a repeatable and controlled environment free of cost with CloudSim.

Why use CloudSim?

Below are a few reasons to opt for CloudSim:

- *Open source* and *free of cost*, so it favours researchers/developers working in the field.
- Easy to download and set-up.
- It is more *generalized* and *extensible* to support modelling and experimentation.
- Does not require any high-specs computer to work on.
- Provides *pre-defined allocation policies* and *utilization models* for managing resources, and allows implementation of user-defined algorithms as well.
- The documentation provides *pre-coded examples* for new developers to get familiar with the basic classes and functions.
- Tackle bottlenecks before deployment to reduce risk, lower costs, increase performance, and raise revenue.

CloudSim Architecture:



CloudSim Layered Architecture

CloudSim Core Simulation Engine provides interfaces for the management of resources such as VM, memory and bandwidth of virtualized Datacenters.

CloudSim layer manages the creation and execution of core entities such as VMs, Cloudlets, Hosts etc. It also handles network-related execution along with the provisioning of resources and their execution and management.

User Code is the layer controlled by the user. The developer can write the requirements of the hardware specifications in this layer according to the scenario.

Some of the most common classes used during simulation are:

- **Datacenter**: used for modelling the foundational hardware equipment of any cloud environment, that is the Datacenter. This class provides methods to specify the functional requirements of the Datacenter as well as methods to set the allocation policies of the VMs etc.
- **Host**: this class executes actions related to management of virtual machines. It also defines policies for provisioning memory and bandwidth to the virtual machines, as well as allocating CPU cores to the virtual machines.
- **VM**: this class represents a virtual machine by providing data members defining a VM's bandwidth, RAM, mips (million instructions per second), size while also providing setter and getter methods for these parameters.
- **Cloudlet**: a cloudlet class represents any task that is run on a VM, like a processing task, or a memory access task, or a file updating task etc. It stores parameters defining the characteristics of a task such as its length, size, mi (million instructions) and provides methods similarly to VM class while also providing methods that define a task's execution time, status, cost and history.
- **DatacenterBroker**: is an entity acting on behalf of the user/customer. It is responsible for functioning of VMs, including VM creation, management, destruction and submission of cloudlets to the VM.
- **CloudSim**: this is the class responsible for initializing and starting the simulation environment after all the necessary cloud entities have been defined and later stopping after all the entities have been destroyed.

STEPS TO FOLLOW :

1. Download and install eclipse

2.Open eclipse and go to File -> new->java project-> give any project name-> next-> finish.

3.Expand project by clicking on project name (SJFCloudSim) arrow -> select src-> right click on src->new->package->give any name to package(Assignment3)->finish

4.Right click on Assignment3(package)->select show in -> system explorer->it will open src folder -> open Assignment3 package->download sample code from GitHub and paste 5 Files in Assignment3 package folder-> close

5.Go to eclipse -> right click on project name(SJFCloudSim)->select close project

6.Right click on project name(SJFCloudSim)->open project-> click on arrow before project name->click on src arrow->Assignment3 arrow->open all 5 files and write package name(Assignment3)in first statement of each file and save each file.

7.Select project tab on upper menu bar ->properties->from left side select java build path->select libraries -> click on add external JARs-> select already downloaded CloudSim JARs(CloudSim4.0 and CloudSim-examples4.0)->open ->apply->apply and close. If not downloaded then go to google ->CloudSim download-> select GitHub link releases –cloudslab/CloudSim->go to CloudSim 4.0-> select assets arrow-> download all 4 files -> extract CloudSim 4.0.tar

8.Run the file which contains main function(SJF_Schedule.java) by right click -> run as -> java application.

OUTPUTS :

<Images of actual implementation>

Conclusion: We successfully installed and implemented the Scheduling alogithm that was not in the CloudSim environment.

Sinhgad Institutes

Name of the Student: _____

Roll no: _____

CLASS: - T..E. IT

Subject Name: - CC Lab

Assignment No. 04

**** Find a procedure to transfer the filesfrom one virtual machine to another virtual machine: CO3 ****

Date of Performance: / /2024

Marks out of 10:

Sign with Date:

Assignmnet No. 4

Aim: Find a procedure to transfer the files from one virtual machine to another virtual machine.

Theory:

Transferring files between virtual machines (VMs) can be done through several methods depending on the virtualization platform you're using and the specific requirements of your setup. Here's a general procedure that can work across different virtualization platforms:

1. **Shared Folders or Drives**: Many virtualization platforms offer the ability to create shared folders or drives that can be accessed by multiple VMs. You can set up a shared folder on one VM and then access it from another VM to transfer files.

2. **Network File Transfer**: Use network file transfer protocols like FTP (File Transfer Protocol), SCP (Secure Copy Protocol), or SFTP (SSH File Transfer Protocol) to transfer files between VMs over the network. Ensure both VMs are on the same network and configure the necessary network settings.

3. **Cloud Storage**: If both VMs have internet access, you can upload the files to a cloud storage service (e.g., Google Drive, Dropbox, OneDrive) from one VM and then download them on the other VM.

4. **Email or Messaging Services**: For small files, you can email them to yourself or use messaging services like Slack or Discord to transfer files between VMs.

5. **External Storage**: Save the files to an external storage device (e.g., USB drive) from one VM and then connect the storage device to the other VM to access the files.

6. **Virtual Machine Import/Export**: Some virtualization platforms allow you to export VMs to a file and then import them into another VM. This method may be suitable for transferring larger amounts of data or entire VM images.

7. **Command Line Tools**: Depending on the operating systems running on your VMs, you can use command-line tools like `scp` (secure copy) or `rsync` to transfer files over SSH.

8. **Virtualization Management Tools**: Virtualization management tools like VMware vSphere, Microsoft Hyper-V, or VirtualBox may offer features or plugins for transferring files between VMs within the same host environment.

9. **APIs and Scripts**: For advanced users, you can use APIs provided by the virtualization platform to write scripts or programs that automate the file transfer process between VMs.

Before implementing any of these methods, ensure you have the necessary permissions and security measures in place to protect your data and maintain compliance with any applicable policies or regulations.

Procedure

1) Install Oracle VM Virtual box

2) Create two Virtual machines VM1&VM2. Download 23.10.1.Desktop—amd64 ISO Disk image file & add path of above ISO Disk image file path & select type linux and version Ubuntu(64-bit)

3) In virtual box select tools-> create Nat network

4) IN VM1 go to settings→ network → attached two Nat network → ok. Repeat same procedure for VM2

5) Open VM1 & VM2

6) Open terminal in both virtual machine

7) Type ls and check the files present

8) Create any file by using command touch filename.txt → type cat filename.txt to see the contents of the file

9) To add data to filename.txt type nano.txt which will open an editor in which you can type any txt message → ctrl S→ ctrl X to exit.

10) Type cat filename.txt. It will show entered text

11) To find ip address type ifconfig . if you get an error saying ifconfig not found then type sudo apt install net-tools→ wait for 100% completions after which again type ifconfig and note IP address of machine . repeat the same procedure for other virtual machine

12) Type systemctl status ssh → if it gives error type sudo apt-get install openssh-server openssh-client→ select install the package maintainers version. Repeat the same for both virtual machine

13) Type systemctl status ssh→ if status is inactive then type systemctl restart ssh→ authenticate → check whether the status has become active by command systemctl status ssh ctrl C to exit . repeat the same procedure for both virtual machines

14) Type command scp filename.txt ubuntu@IPAddress of receiver virtual machine:/home/Ubuntu→ type yes. It will ask for password -→ type any password →it will deny → now change password by command passwd and change password .repaeat the same procedure for both virtual machine

15) Run command scp filename.txt ubuntu@IPAddress of receiver virtual machine:/home/Ubuntu and Enter new password of receiving virtual machine .

16) Check whether the file is received on receiving virtual machine by commands ls & cat.

OUTPUTS:

<Images of actual implementation>

Conclusion: T h u s w e have learned how to transfer files from one virtual machine to another.

Sinhgad Institutes

Name of the Student: _____

Roll no: _____

CLASS: - T..E. IT

Subject Name: - CC Lab

Assignment No. 05

**** Find a procedure to launch a virtual machine using trystack:CO3 ****

Date of Performance: / /2024

Marks out of 10:

Sign with Date:

Assignment No. 5

Aim: Find a procedure to launch a virtual machine using trystack (Online Openstack Demo Version)

Theory:

Virtual Machines: VM is no different than any other physical computer like a laptop, smartphone or server. It has a CPU, memory, and disks by which you can store your files and can connect to the internet if needed. In the VM world Operating System running on your computer is called a host and any operating system running inside VMs is called a guest.

Advantages of VMs:

- Cost Saving
- Speed
- Lowered downtime
- Secure Environment
- Access Remotely

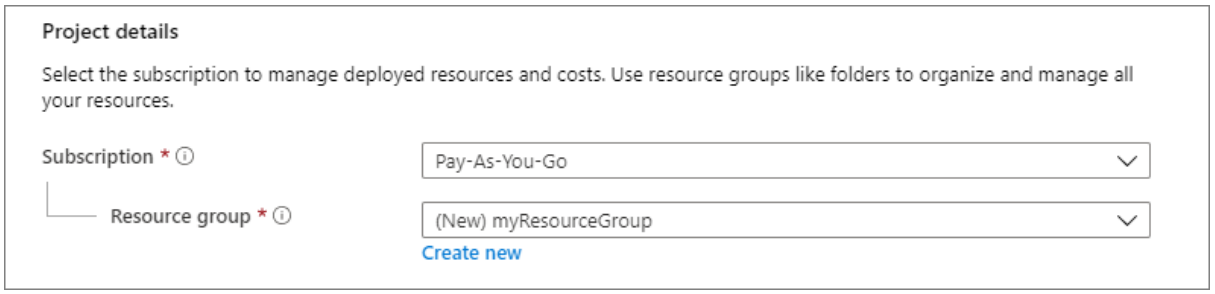
TryStack: TryStack is a free and easy way for users to try OpenStack, and setup their cloud with networking, storage and computer instances.

Requirements: Account on AWS/ Google Cloud/ Azure

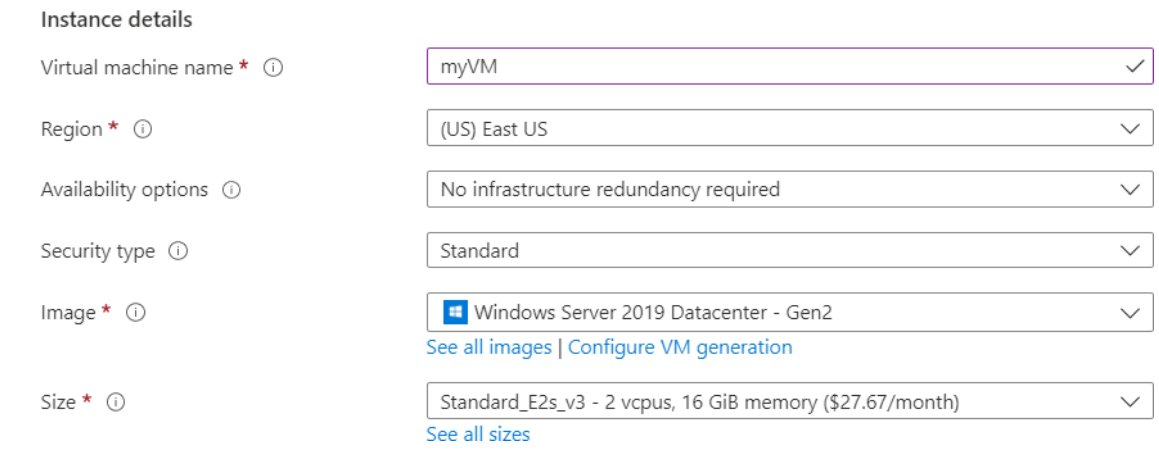
Steps:

Step1:- Create a virtual machine.

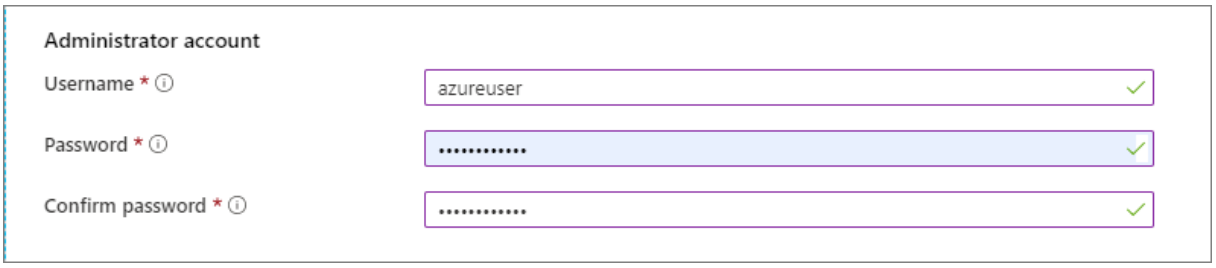
1. Enter *virtual machines* in the search.
2. Under Services, select Virtual machines.
3. In the Virtual machines page, select Create and then Virtual machine. The Create a virtual machine page opens.
4. In the Basics tab, under Project details, make sure the correct subscription is selected and then choose to Create a new resource group. Enter *myResourceGroup* for the name.



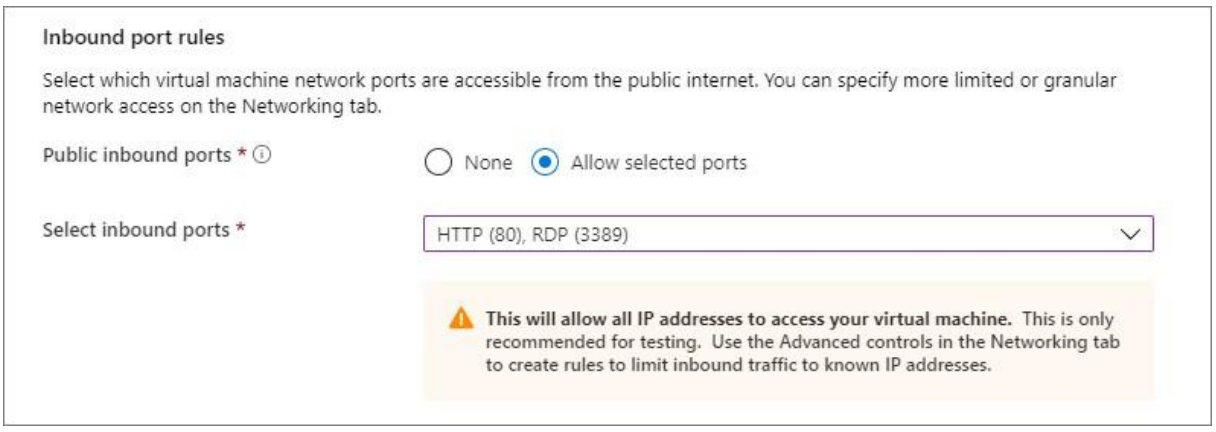
5. Under **Instance details**, enter *VM name* for the **Virtual machine name** and choose *Windows Server 2019 Datacenter - Gen2* for the **Image**. Leave the other defaults.



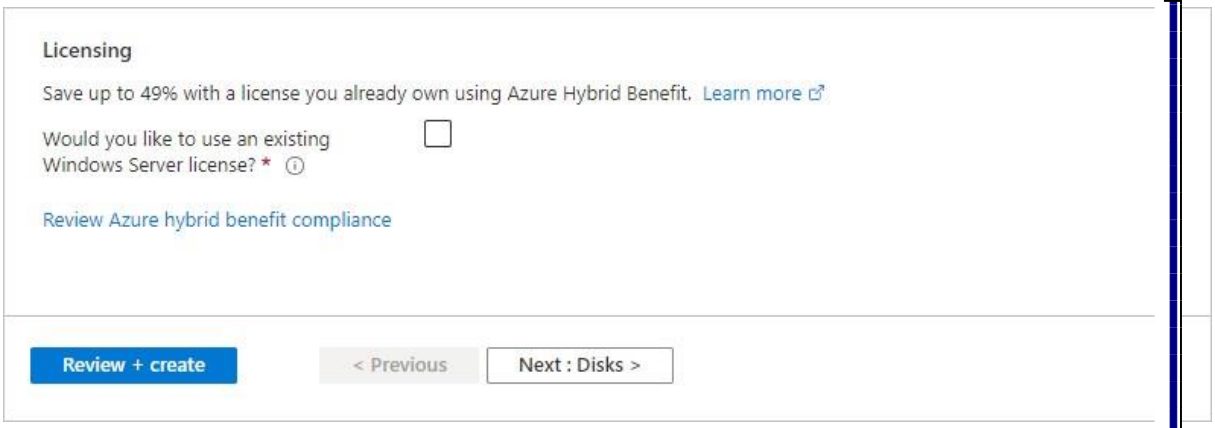
6. Under the **Administrator account**, select a password, and provide a username, such as *azureuser* , and a password. The password must be at least 12 characters long and meet the defined complexity requirements.



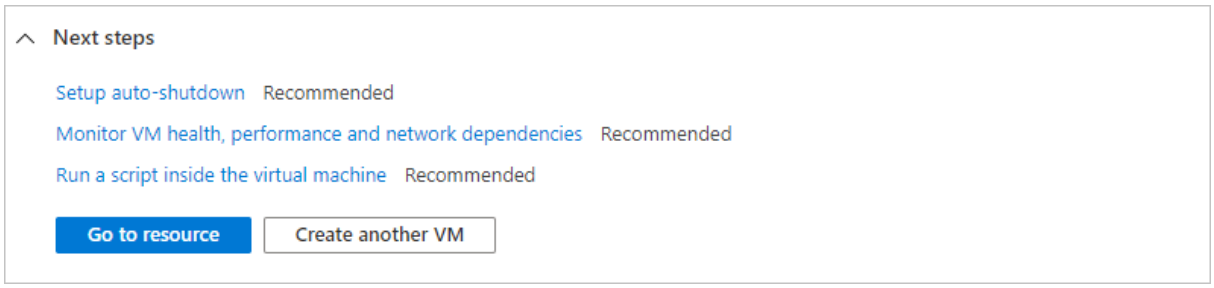
7. Under **Inbound port rules**, choose **Allow selected ports** and then select **RDP (3389)** and **HTTP (80)** from the drop-down.



8. Leave the remaining defaults and then select the **Review + create** button at the bottom of the page.

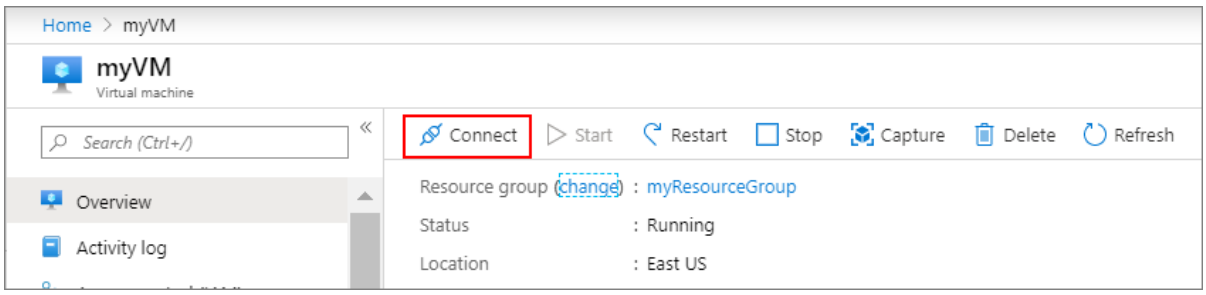


9. After validation runs, select the **Create** button at the bottom of the page.
10. After deployment is complete, select **Go to resource**.



Step 2:- Connect to a virtual machine

1. On the overview page for your virtual machine, select the **Connect** > **RDP**.



2. In the **Connect with RDP** page, keep the default options to connect by IP address, over port 3389, and click **Download RDP file**.
3. Open the downloaded RDP file and click **Connect** when prompted.
4. In the **Windows Security** window, select **More choices** and then **Use a different account**. Type the username as **localhost***username*, enter the password you created for the virtual machine, and then click **OK**.
5. You may receive a certificate warning during the sign-in process. Click **Yes** or **Continue** to create the connection.

Conclusion: Hence we have learned the procedure to launch a virtual machine using trystack.

Sinhgad Institutes

Name of the Student: _____

Roll no: _____

CLASS: - T..E. IT

Subject Name: - CC Lab

Assignment No. 06

**** Design and deploy a web application in a PaaS environment: CO3 ****

/ /2024

Date of Performance:

Marks out of 10:

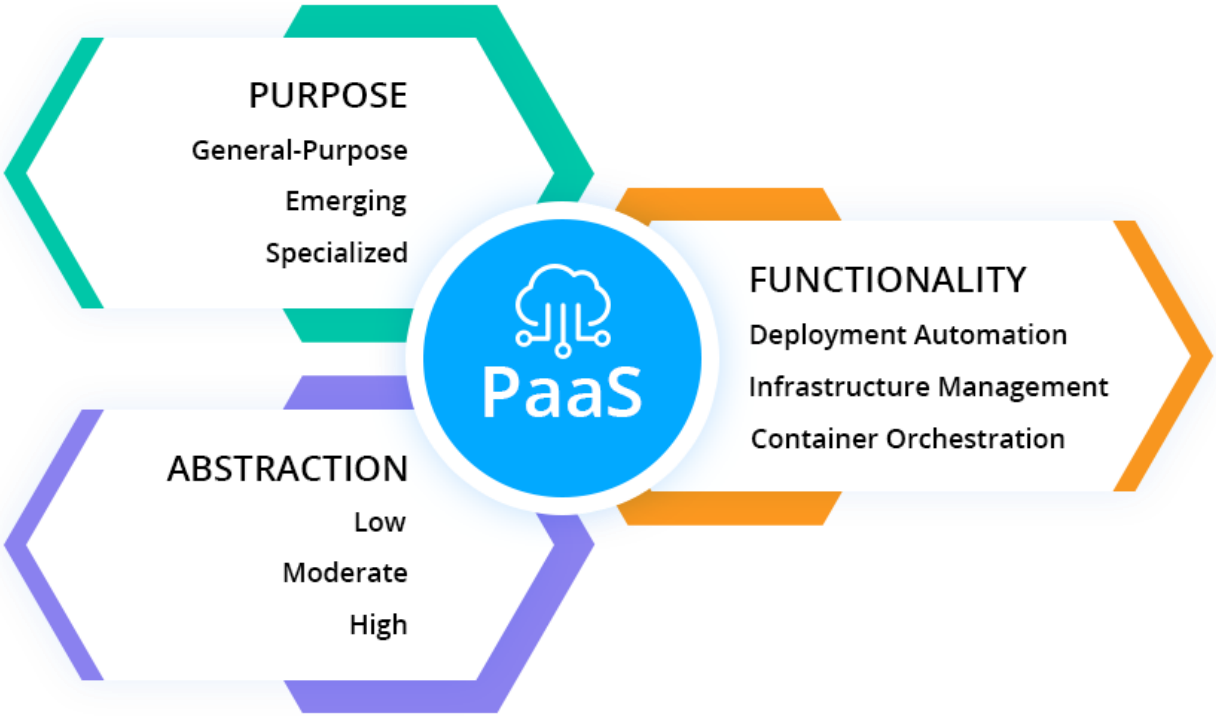
Assignment No. 6

Aim: Design and deploy a web application in a PaaS environment.

Theory:

What is PaaS?

Platform as a service (PaaS) is a category of cloud computing services that provides a platform allowing customers to develop, run, and manage applications without the complexity of building and maintaining the infrastructure typically associated with developing and launching an app.



What AWS Amplify is?

AWS Amplify is a set of purpose-built tools and features that lets frontend web and mobile developers quickly and easily build full-stack applications on AWS, with the flexibility to leverage the breadth of AWS services as your use cases evolve. With Amplify, you can configure a web or mobile app backend, connect your app in minutes, visually build a web frontend UI, and easily manage app content outside the AWS console. Ship faster and scale effortlessly—with no cloud expertise needed.

❖ **Procedure**

1) In google search Aws amplify
2) Create a new aws account
3) In github create new repository and add all the required files into it
4) In google search Aws amplify & click on host a web application with AWS amplify
5) Click on github & continue
6) Click on “only select repositories”
7) In recently updated repositories,select your repository
8) Tick connecting a monorepo & write path of the repository → next → save &deploy
9) Wait till deployment process is complete
10) Open the generated link

OUTPUTS:

<Images of actual implementation>

Conclusion: Thus we have deployed a web application in a PaaS environment.