

2. Use GAE launcher to launch the web applications.

****Question:** What is GAE Launcher?**

****Answer:**** GAE Launcher, or Google App Engine Launcher, is a graphical user interface (GUI) tool provided by Google for managing Google App Engine applications locally on a developer's machine. It allows developers to run, deploy, and manage their web applications built on Google App Engine.

****Question:** What are the steps to launch a web application using GAE Launcher?**

****Answer:****

1. Install GAE Launcher and set up the Google App Engine SDK.
2. Create or open an existing Google App Engine project in GAE Launcher.
3. Configure project settings, including the application ID and version.
4. Run the application locally using the built-in development server.
5. Deploy the application to Google App Engine's cloud infrastructure directly from GAE Launcher.

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

****Question:** What is CloudSim?**

****Answer:**** CloudSim is a framework for modeling and simulating cloud computing infrastructures and services. It provides support for modeling various cloud components such as data centers, virtual machines, and applications, allowing researchers and developers to experiment with different cloud scenarios and algorithms.

****Question:** How can you run a scheduling algorithm that is not present in CloudSim?**

****Answer:****

1. Implement the custom scheduling algorithm using Java or any other supported programming language.
2. Integrate the custom algorithm into the CloudSim framework by extending the existing scheduling classes or implementing new ones.
3. Configure CloudSim to use the custom scheduling algorithm when simulating cloud scenarios by specifying it in the simulation parameters or configuration files.
4. Run the CloudSim simulation with the custom scheduling algorithm to evaluate its performance and behavior in the simulated cloud environment.

4. Find a procedure to transfer files from one virtual machine to another virtual machine.

****Question:** What is a common procedure to transfer files between virtual machines?**

****Answer:**** One common procedure to transfer files between virtual machines is using SSH (Secure Shell) and SCP (Secure Copy Protocol).

1. Use SSH to connect to the source virtual machine.
2. Use SCP to securely copy files from the source virtual machine to the destination virtual machine using the command line or a graphical SCP client.
3. Authenticate with the destination virtual machine using SSH keys or login credentials.
4. Transfer the files by specifying the source and destination paths and filenames.

5. Find a procedure to launch a virtual machine using Trystack (Online Openstack Demo Version).

****Question:** What is Trystack?**

****Answer:**** Trystack is an online OpenStack demo environment provided by the OpenStack community for users to explore and experiment with OpenStack cloud computing platform features and functionalities.

****Question:** How can you launch a virtual machine using Trystack?**

****Answer:****

1. Sign up for an account on the Trystack website and log in to the dashboard.
2. Navigate to the Compute section of the dashboard and click on the Instances tab.
3. Click on the Launch Instance button to launch a new virtual machine.
4. Specify the instance details such as flavor, image, and key pair.
5. Configure networking options and security groups as needed.
6. Launch the instance and wait for it to become active.
7. Access the virtual machine using SSH or remote desktop protocols.

6. Design and deploy a web application in a PaaS environment.

****Question:** What is PaaS?**

****Answer:**** PaaS, or Platform as a Service, is a cloud computing service model that provides a platform allowing customers to develop, run, and manage applications without the complexity of building and maintaining the underlying infrastructure.

****Question:** What are the steps involved in designing and deploying a web application in a PaaS environment?**

****Answer:****

1. Choose a PaaS provider that meets your requirements and sign up for an account.
2. Develop the web application using supported programming languages, frameworks, and tools provided by the PaaS platform.
3. Configure application settings such as environment variables, database connections, and scaling options.
4. Test the application locally and ensure it meets performance and reliability requirements.
5. Deploy the application to the PaaS environment using the platform's deployment tools or APIs.
6. Monitor the application's performance, availability, and resource usage in the PaaS dashboard.
7. Scale the application horizontally or vertically as needed to handle increased workload or demand.