

Google Cloud Platform (GCP) Practicals Q&A

Let me provide comprehensive Q&A for each of the practical tasks you've mentioned:

1. Google App Engine Launcher

Q: What is Google App Engine Launcher and how do you use it to deploy web applications? A:

Google App Engine Launcher is a graphical interface tool that helps deploy, manage, and test web applications locally before pushing them to the Google Cloud Platform. To launch an application, install the GAE SDK, open the launcher, create or import a project, configure application settings in app.yaml, test locally using the built-in development server, and finally deploy to production using the "Deploy" button.

Q: What are the key configuration files needed for a GAE application? A: The primary configuration file is app.yaml, which specifies runtime environment, application ID, version, instance class, and URL handlers. For Python applications, you'll also need main.py containing your application code and requirements.txt listing dependencies. Java applications require a WEB-INF directory with web.xml and appengine-web.xml files for servlet configuration and runtime settings.

Q: How do you troubleshoot common deployment issues with GAE Launcher? A: Common deployment issues can be diagnosed by checking the logs in the GAE Launcher console output window. Verify your application ID matches your GCP project, ensure proper SDK version compatibility, check quota limits in your GCP console, and validate your app.yaml configuration for syntax errors. For permissions issues, make sure you're authenticated with the correct Google account that has appropriate IAM roles.

2. File Transfer Between Virtual Machines

Q: How can you transfer files between two GCP virtual machines using gcloud commands? A: Files can be transferred between GCP VMs using the gcloud compute scp command with the syntax: `gcloud compute scp [LOCAL_FILE_PATH] [INSTANCE_NAME]:[DESTINATION_PATH] --zone=[ZONE]`. For transferring from one VM to another, first SSH into the source VM, then use `gcloud compute scp` to copy files directly to the destination VM. Authentication is handled automatically using your GCP credentials.

Q: What secure options exist for transferring files between VMs in different cloud environments?

A: For cross-cloud transfers, set up an SFTP server on the destination VM and use SFTP client on the source VM. Alternatively, use SCP with proper key authentication or rsync over SSH with the command `rsync -avz -e ssh /source/path user@destination-vm:/destination/path`. For large or frequent transfers, consider using a shared cloud storage bucket (like Google Cloud Storage) as an intermediary.

Q: How would you automate regular file transfers between virtual machines? A: Automate transfers by creating a bash script incorporating scp/rsync commands and schedule it using cron jobs with appropriate timing parameters. For more complex synchronization, implement tools like Syncthing or set up a continuous synchronization service. In production environments, consider using configuration management tools like Ansible or creating a CI/CD pipeline that handles file distribution.

3. Virtual Machine Deployment with TryStack (OpenStack)

Q: What is TryStack and how do you register for access? A: TryStack is a free demonstration environment of OpenStack that allows users to experiment with OpenStack's features without setting

up their own infrastructure. To access TryStack, register on their website providing required information, await approval, and then receive login credentials. TryStack resources are typically limited and temporary, designed for learning and testing purposes.

Q: What is the procedure to launch a virtual machine in TryStack? A: After logging into the TryStack dashboard, navigate to "Project > Compute > Instances" and click "Launch Instance". Select an appropriate image (OS template), flavor (VM size), configure networking settings, add a key pair for SSH access, and specify security groups. Finally, click "Launch" to provision the VM and wait for it to initialize before accessing it via SSH using your key pair.

Q: How do you attach persistent storage to a TryStack virtual machine? A: In the TryStack dashboard, go to "Project > Volumes > Volumes" and create a new volume specifying size and name. Once created, select "Attach" from the volume's dropdown menu and choose your running instance. After attachment, SSH into your VM, format the volume using tools like `fdisk` and `mkfs`, then mount it to a directory with the `mount` command. For persistence across reboots, add the mount entry to `/etc/fstab`.

4. Web Application Deployment in PaaS Environment

Q: What are the key components needed to design a web application for PaaS deployment? A: A PaaS-ready web application requires a clearly defined architecture with stateless application logic, externalized configuration for different environments, proper handling of temporary file storage, a dependency management system, and database connection pooling. The application should follow a 12-factor app methodology, including environment variable configuration and appropriate logging, while adhering to the specific PaaS provider's guidelines.

Q: How do you deploy a Python Flask application to Google App Engine? A: Create a Flask application with necessary routes and logic, then define an `app.yaml` file specifying "python39" (or your version) runtime. Include `requirements.txt` with all dependencies, create an `appengine_config.py` for vendor configurations, and implement a `main.py` file with a WSGI-compatible application variable. Use the `gcloud` CLI with `gcloud app deploy` to push your application to GAE, where it will automatically provision infrastructure and scale based on traffic.

Q: What strategies should be implemented for monitoring and scaling a PaaS-deployed application? A: Implement logging that sends structured data to your PaaS provider's logging service, set up health checks for application status monitoring, and integrate application performance monitoring (APM) tools. Configure appropriate scaling parameters in your PaaS configuration (like `instance_class` and scaling settings in `app.yaml` for GAE). Create alerts for key metrics such as latency, error rates, and resource utilization, and use the PaaS provider's dashboard for real-time performance analysis.

5. Firebase Authentication with GAE and Datastore

Q: How do you integrate Firebase Authentication with a Google App Engine application? A: Add Firebase SDK to your GAE project using `npm` or the Firebase CLI, then include Firebase configuration in your application code. Implement sign-in methods (email/password, Google, etc.) using Firebase Authentication API, and verify Firebase ID tokens on your server-side GAE application. Create middleware to extract and validate user information from ID tokens and include Firebase Admin SDK in your GAE application for secure server-side verification.

Q: What is the process for storing authenticated user credentials in Google Cloud Datastore? A: After successful Firebase authentication, retrieve the user's unique ID (UID) and profile information

from the authentication response. Create a Datastore entity with the UID as the key and store necessary user profile information as properties. Implement data access methods that query Datastore based on authenticated user context, ensuring proper error handling and transaction management. Set appropriate security rules to restrict data access based on user authentication status.

Q: How would you implement secure sessions and role-based access control using Firebase and Datastore? A: Use Firebase custom claims to store user roles and permissions in the ID token, configuring them through Firebase Admin SDK. Store detailed role information in Datastore entities linked to user profiles. Implement middleware that examines token claims and retrieves additional permission details from Datastore when needed. Create decorators or guards that check permissions before allowing access to protected routes, and establish clear role hierarchies with inheritance for more complex permission structures.