**Applications in the cloud**

- (App Engine) App Engine automatically manages server provisioning based on demand and supports popular coding languages. It offers built-in services and APIs, and developers can use SDKs for local development. The Cloud Console handles app management in production, and the Security Command Center ensures web app security by scanning for vulnerabilities.

- (App Engine environments) In Google App Engine, there are two types of environments: standard and flexible.

1. App Engine Standard Environment:

* Based on container instances running on Google's infrastructure.
* Pre-configured runtimes from a supported list of languages and versions.
* Features include automatic scaling, load balancing, task queues, scheduled tasks, and integration with Google Cloud services.
* Requirements: Must use specified versions of supported languages and conform to sandbox constraints.
* Advantages: Fast startup, secure sandbox environment, and automatic server scaling.
* Limitations: Less control over infrastructure, no SSH access, and limited local disk access.

1. App Engine Flexible Environment:

* Allows specifying the container type for the web application, running in Docker containers on Google Cloud's Compute Engine VMs.
* App Engine manages Compute Engine instances.
* Supports more customization, including custom runtimes, libraries, and Docker files.
* Advantages: More control over infrastructure, SSH access, and local disk usage.
* Limitations: Slower startup time compared to the standard environment.
* Pricing: Resource allocation per hour with no automatic shutdown.

Comparing to Google Kubernetes Engine (GKE):

* App Engine Standard: Ideal for simplified deployment and scaling with less control.
* Google Kubernetes Engine: Offers full flexibility and control using Kubernetes.
* App Engine Flexible: A middle ground between the two options, providing more control than standard but less than GKE.

- (Google Cloud API management tools). An API (Application Programming Interface) is a set of rules that allows software applications to communicate with each other. Google Cloud offers three API management tools:

1. Cloud Endpoints: Manages APIs with low latency and high performance, supporting OpenAPI Specification. Works with App Engine, Google Kubernetes Engine, and Compute Engine.
2. API Gateway: Provides a consistent REST API for various backend services, regardless of implementation, enabling secure access for clients.
3. Apigee API Management: Focuses on business problems like rate limiting, quotas, and analytics. Useful for handling legacy applications and decomposing them into microservices gradually.

- (Cloud Run) Cloud Run is a managed serverless compute platform that runs stateless containers in response to web requests or Pub/Sub events. It is built on Knative, an open API and runtime environment based on Kubernetes, providing flexibility to move workloads across different environments and platforms.

Key features of Cloud Run:

1. Serverless: No infrastructure management required, allowing developers to focus on application development.
2. Fast Scaling: Scales up and down from zero almost instantaneously, charging only for the resources used down to the nearest 100 milliseconds.
3. Developer Workflow: Write the application in your preferred programming language, package it into a container image, and deploy it to Cloud Run via Artifact Registry.
4. Source-Based Workflow: Alternatively, deploy source code instead of a container image, and Cloud Run will build the application using buildpacks.
5. HTTPS Serving: Cloud Run handles HTTPS serving, providing a unique subdomain or allowing the use of custom domains. It manages SSL certificates and request handling.
6. Unique Pricing: Pay only for system resources used during handling of web requests, starting, or shutting down the container, with a small fee per one million requests served.
7. Language Support: Cloud Run can run web applications written in popular languages like Java, Python, Node.js, PHP, Go, and C++, as well as less common languages like Cobol, Haskell, and Perl, as long as they handle web requests.