

EXPERIMENT NO. 9

AIM: Implementation of PaaS using Google App Engine/AWS/Azure

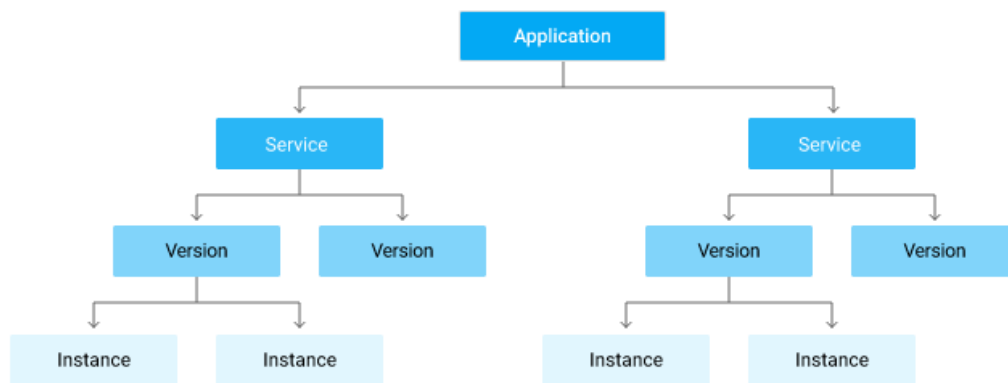
THEORY:

Platform as a service is a deployment and development environment within the cloud that delivers simple cloud based applications to complex, cloud enabled applications. PaAs is designed to support and complete the web application lifecycle, building, testing, deploying, managing and updating.

PaAs includes a complete infrastructure of servers, storages and middleware development tools like business intelligence services (CBI), database management systems etc. A complete platform is offered in PaAs in which the client can host their applications without the need to worry about maintenance of the servers and its operating system. However the user of the PaAs service should look after the implementation of the developed application to decide whether to scale it up or down depending on the traffic that the application receives.

The PaAs database utilizes visualization techniques, where the virtual machines were dependent on the actual hardware that hosts it.

An App Engine app is made up of a single application resource that consists of one or more *services*. Each service can be configured to use different runtimes and to operate with different performance settings. Within each service, you deploy *versions* of that service. Each version then runs within one or more *instances*, depending on how much traffic you configured it to handle. Google App Engine is a Platform as a Service (PaaS) product that provides Web app developers and enterprises with access to Google's scalable hosting and tier 1 Internet service. The App Engine requires that apps be written in Java or Python, store data in Google BigTable and use the Google query language. Non-compliant applications require modification to use App Engine. Google App Engine provides more infrastructure than other scalable hosting services such as Amazon Elastic Compute Cloud (EC2). The App Engine also eliminates some system administration and developmental tasks to make it easier to write scalable applications. The following diagram illustrates the hierarchy of an App Engine app running with multiple services. In this diagram, the app has two services that contain multiple versions, and two of those versions are actively running on multiple instances:



GAE runs in the Google Cloud Platform which is a worldwide network of cloud services that Google relies on to run its own products. Unlike other vendors who have built separate cloud service centers specifically to resell to you, GCP uses the exact same servers, routers, and other physical equipment for Google's products as yours. When you deploy an application into GAE you are relying on the same scaffolding that supports products such as Google Maps or Google Mail. It's entirely possible for your data to end up on the same disk as Google Search data. You're buying into the Google ecosystem in a way that makes you a partner with Google instead of a customer. GAE can also integrate tightly with other Google Cloud Platform products such as Cloud SQL, BigQuery, and Compute Engine. Google App Engine is free up to a certain amount of resource usage. Users exceeding the per-day or per-minute usage rates for CPU resources, storage, number of API calls or requests and concurrent requests can pay for more of these resources.

CONCLUSION: Hence successfully completed the implementation of PaAs using Google App engine.

OUTPUT:

The screenshot displays the Google Cloud Platform console interface for deploying a website. The top navigation bar shows "Google Cloud Platform" and "Deploying a website". The main content area is titled "App Engine" and "Next steps".

Resources:

- Documentation for Python in App Engine
- Visit GitHub for code samples in standard Python

Deploy with Google Cloud SDK

Download the Cloud SDK

Initialize your SDK

```
gcloud init
```

Deploy to App Engine

```
gcloud app deploy
```

DO THIS LATER

Your App Engine app has successfully been created

The bottom section shows a terminal window with the following commands and output:

```
-6, --ipv6 use IPv6 addresses only

anisha98 span@cloudshell:~$ git clone \
> https://github.com/GoogleCloudPlatform/python-docs-samples
Cloning into 'python-docs-samples'...
remote: Enumerating objects: 55, done.
remote: Counting objects: 100% (55/55), done.
remote: Compressing objects: 100% (43/43), done.
remote: Total 27148 (delta 20), reused 28 (delta 3), pack-reused 27093
Receiving objects: 100% (27148/27148), 38.99 MiB | 8.17 MiB/s, done.
Resolving deltas: 100% (14254/14254), done.
anisha98 span@cloudshell:~$ cd \
> python-docs-samples/appengine/standard_python37/hello_world
anisha98 span@cloudshell:~/python-docs-samples/appengine/standard_python37/hello_world$ cat main.py
```

