

[Show Submission Credentials](#)

P0. Google Cloud Platform APIs

A brief overview of using Google Cloud Platform's APIs

✓ Google Cloud Platform API

✓ GCP Java API

✓ GCP Python API

✓ GCP Cloud SDK

Google Cloud Platform API

GCP SDK and Developer Tools

Similarly to AWS and Azure, Google offers Cloud Client Libraries (<https://cloud.google.com/apis/docs/cloud-client-libraries>) and command line tools, the Google Cloud SDK (<https://cloud.google.com/sdk/docs/>). Client libraries are offered for many languages, including Go, Java, Node.js, Python, and others.

In this primer we will use the Google Cloud Client Library for Java (<https://googlecloudplatform.github.io/google-cloud-java/google-cloud-clients/index.html>) and the Google Cloud Client Library for Python (<https://googlecloudplatform.github.io/google-cloud-python/>) to launch virtual machines. The Cloud SDK is a set of tools that you can use to manage resources and applications hosted on Google Cloud Platform and their usage will be explored later in this primer.

GCP Java API

Google Cloud Client Library for Java

We need you to install Maven. Maven is a software project management and comprehension tool. To learn more about Maven you may refer to the Intro to Maven and Checkstyle primer (<https://theproject.zone/s21-15619/maven-checkstyle-primer>) but for now, you just need to know that all required dependencies are specified in `pom.xml` of the code

```
sudo apt-get update
sudo apt-get install maven
```

As in the previous AWS and Azure API examples, we will be providing a Maven project to demonstrate using the Java client library to manage cloud resources. We will take a dependency on the latest version of the `google-cloud` package. The project's ***pom.xml*** will have the following dependency:

```
<dependency>
  <groupId>com.google.cloud</groupId>
  <artifactId>google-cloud</artifactId>
  <version>0.32.0-alpha</version>
</dependency>
```

The main documentation for the `google-cloud` package can be found on GitHub (<http://googlecloudplatform.github.io/google-cloud-java>). Additional Java samples are hosted in the `GoogleCloudPlatform/java-docs-samples` (<https://github.com/GoogleCloudPlatform/java-docs-samples>) repository.

Cloud SDK and environment setup

Install and initialize Cloud SDK

```
$ gcloud init
$ gcloud projects create gcp-java-p1-xxxxxx (e.g., gcp-java-p1-123456)
```

Visit the GCP web console to enable billing and enable the Compute Engine API for this GCP project.

The GCP Java SDK depends on an environment variable `GOOGLE_CLOUD_PROJECT` to decide the GCP project to work with:

```
$ export GOOGLE_CLOUD_PROJECT=gcp-java-p1-xxxxxx
```

Client Authentication

The recommended method for authenticating with the Cloud API is via the use of Service Accounts (<https://cloud.google.com/compute/docs/access/service-accounts>). The steps to create a new service account can be found in the Getting Started with Authentication (<https://cloud.google.com/docs/authentication/getting-started>) documentation:

1. Visit the create service account page (<https://console.cloud.google.com/apis/credentials/serviceaccountkey>) in the GCP console.
2. Select **New service account** from the Service account drop down.

3. Provide a name for the service account.
4. Set the Project / Owner role for the service account.
5. **Create** the service account and download the JSON file.

To use the newly created service account in your project, you must set the `GOOGLE_APPLICATION_CREDENTIALS` to be the path of the JSON file downloaded in the previous step.

```
export GOOGLE_APPLICATION_CREDENTIALS=<PATH_TO_SERVICE_ACCOUNT_FILE>
```

The client authentication described above is currently the recommended method for GCP authentication. Alternatively, run `gcloud auth application-default login` if you are running the code on your local machine.

GCP Java Compute API

Download the code using the commands given below, use the Maven command line to execute the `LaunchInstance` class. This class will create a new Ubuntu virtual machine and output the name of the newly create instance (e.g. **cc-instance-xxxxxxxxxxxx**). You may replace the image project and image name to launch custom images, which will be discussed in a later project.

```
$ wget https://clouddeveloper.blob.core.windows.net/s21-15619/gcp-primer/gcp-primer-java.tar.gz
...

$ tar xzf gcp-primer-java.tar.gz
$ cd gcp-primer-java
```

Execute the `LaunchInstance` class to provision a GCP VM.

```
$ mvn compile && mvn exec:java -Dexec.mainClass="edu.cmu.cs.cloud.samples.gcp.LaunchInstance"
[INFO] Scanning for projects...
[INFO]
[INFO] -----
[INFO] Building GCP-API-Samples 0.0.1-SNAPSHOT
[INFO] -----
[INFO]
[INFO] --- exec-maven-plugin:1.6.0:java (default-cli) @ GCP-API-Samples ---
Disk DiskId{project=null, zone=us-east1-d, disk=cc-disk-xxxxxxxxxxxx} was successfully created
Address RegionAddressId{project=null, address=cc-address-xxxxxxxxxxxx, region=us-east1} was successfully created
Instance InstanceId{zone=us-east1-d, instance=cc-instance-xxxxxxxxxxxx} was successfully created
[INFO] -----
[INFO] BUILD SUCCESS
[INFO] -----
[INFO] Total time: 21.566 s
[INFO] Finished at: 2017-07-18T10:01:07-04:00
[INFO] Final Memory: 17M/133M
[INFO] -----
```

Note the instance id provided in the console output; alternatively you may use the `gcloud` CLI or GCP Console to determine the IP address of the machine. Connect to the machine using the `gcloud compute ssh` command. The `gcloud` CLI will generate a SSH key based on your input, which allow you to remotely access the VM.

```
$ gcloud compute ssh cc-instance-xxxxxxxxxxxx
Warning: Permanently added 'compute.2477870006613027445' (ECDSA) to the list of
known hosts.
Welcome to Ubuntu 16.10 (GNU/Linux 4.8.0-56-generic x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:    https://landscape.canonical.com
 * Support:       https://ubuntu.com/advantage

Get cloud support with Ubuntu Advantage Cloud Guest:
http://www.ubuntu.com/business/services/cloud

0 packages can be updated.
0 updates are security updates.

The programs included with the Ubuntu system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by
applicable law.

$ ubuntu@cc-instance-xxxxxxxxxxxx
```

GCP Python API

Google Cloud Client Library for Python

GCP offers a Python Client library (https://cloud.google.com/compute/docs/api/libraries#google_apis_python_client_library) for managing cloud resources from a Python environment. In this section we will be using the latest version of the `google-api-python-client`.

Client Authentication

Refer to the Client Authentication discussion in the previous section. The recommended methods of authentication are:

1. To create new service account with appropriate permissions and set the `GOOGLE_APPLICATION_CREDENTIALS` environment variable.

2. **Alternatively**, if you are running the code on your local machine, you can run `gcloud auth application-default login`.

GCP Python Compute API

In the Creating and Starting an Instance

(<https://cloud.google.com/compute/docs/instances/create-start-instance>) documentation Google provides an overview of launching a Compute instance using the Python client library. The scripts provided below are recreated based on the public samples provided in `GoogleCloudPlatform/python-docs-samples` (<https://github.com/GoogleCloudPlatform/python-docs-samples>).

```
import googleapiclient.discovery

def create_instance(compute, project, zone, name, bucket):

def wait_for_operation(compute, project, zone, operation):

if __name__ == '__main__':
    compute = googleapiclient.discovery.build('compute', 'v1')

    print('Creating instance.')

    create_instance(...)
```

Additional information about using the Python client library, references, and code samples can be found on the Google API Client Libraries (<https://developers.google.com/api-client-library/python/>). To run the provided example code on your local machine, complete the following steps after setting up a Python 3 virtual environment and configuring the authentication settings:

```
$ wget https://clouddeveloper.blob.core.windows.net/s21-15619/gcp-primer/gcp-primer-python.tar.gz
...

$ tar xzf gcp-primer-python.tar.gz
$ cd gcp-primer-python
$ virtualenv env

$ source env/bin/activate

(env)$ pip install -r requirements.txt

(env)$ python create_instance.py --zone us-east1-d --name cc-demo-instance-1 [PROJECT-ID]
Creating instance.
Waiting for operation to finish...
done.
Instances in project [PROJECT-ID] and zone us-east1-d:
- cc-demo-instance-1
```

Once the instance has successfully launch, you can connect to it using the `gcloud compute ssh` command as shown below:

```
$ gcloud compute ssh cc-demo-instance-1
Warning: Permanently added 'compute.1820383717008777741' (ECDSA) to the list of
known hosts.
```

```
The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.
```

```
Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
user@cc-demo-instance-1:~$
```

After you are finished with this exercise, remember to delete all your GCP resources.

GCP Cloud SDK

Cloud SDK Components

Cloud SDK provides the following tools:

- `gcloud`
- `bq`
- `gsutil`

Some additional components are installed automatically when you run different `gcloud` commands. For example, `gcloud` automatically installs per-language Google App Engine extensions when you use it to deploy an App Engine application.

```
gcloud
```

`gcloud` is a command-line tool that you can use to perform many common tasks on Google Cloud Platform. You can use `gcloud` to create and manage:

- Google Compute Engine virtual machine instances and other resources
- Google Cloud SQL instances
- Google Container Engine clusters
- Google Cloud Dataproc clusters and jobs
- Google Cloud DNS managed zones and record sets
- Google Cloud Deployment manager deployments

You can also use `gcloud` to deploy App Engine applications and perform other tasks. Additional functionality is available in the `gcloud` Alpha Commands and `gcloud` Beta Commands components. These allow you to work with Google Cloud Bigtable, Google Cloud Dataflow and other parts of the Cloud Platform at earlier release levels than General Availability.

```
bq
```

`bq` is a command-line tool that you can use to work with data in Google BigQuery. You can use `bq` to manage datasets, tables and other entities in BigQuery, as well as run queries on your data.

gsutil

`gsutil` is a command-line tool that you can use to perform tasks in Google Cloud Storage. You can use `gsutil` to:

- Create and manage Cloud Storage buckets
- Upload objects to buckets, and download and delete them
- Move, copy and rename objects
- Manage access to stored data

Installation

The GCP SDK runs on Linux, Mac OS X and Windows, and requires Python 2.7.9 or higher. For installing the GCP SDK, follow the instructions mentioned on Google's Installing Cloud SDK (<https://cloud.google.com/sdk/downloads>) doc.

Setup

Now that you've finished installing the SDK, we have to initialize it. Run `gcloud init` to configure your terminal session to work with GCP. Additionally, you can refer to Google's Initializing Cloud SDK document (<https://cloud.google.com/sdk/docs/initializing>).

After you've initialized your SDK and have explored some of the config options in `gcloud config --help` and `gsutil config --help`, it's important to note how to manage GCP configurations and components. Please refer to Google's Managing Cloud SDK Configurations (<https://cloud.google.com/sdk/docs/managing-configurations>), the Managing Cloud SDK Properties (<https://cloud.google.com/sdk/docs/managing-properties>), and the `gcloud` components (<https://cloud.google.com/sdk/gcloud/reference/components/>) docs.

Usage

Once you have set up Cloud SDK, you can use the tools in the same way you use other command-line tools. This includes running commands from the command line, and running them in scripts and other automations (<https://cloud.google.com/sdk/gcloud/guide/scripting-gcloud>) -- for example, when using scripts, applications, and tools to drive automation of Cloud Platform tasks.

For further documentation and resources on leveraging your GCP SDK, please refer to the Google Cloud SDK Documentation (<https://cloud.google.com/sdk/docs/>).