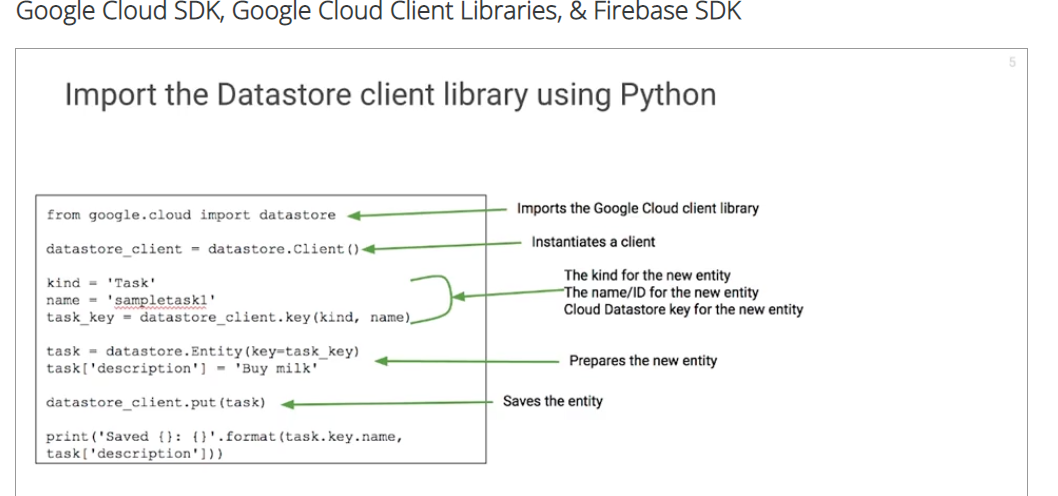
**Google Cloud client libraries** are the recommended method to invoke Google Cloud APIs. The client libraries are available in a variety of popular programming languages. Using these libraries, you can write application code that can easily be executed in a computer environment such as Google App Engine flexible environment, Google container engine or Google compute engine.

The Google Cloud SDK, contains command line tools to work with GCP services. You can use a Firebase SDK to implement Federated identity management. Google has all the client libraries and SDKs that you need to get your job done.

In the module Google Cloud client libraries, Google Cloud SDK and Google Firebase SDK, you'll learn how to setup and use client libraries and SDKs. You'll also learn about API Explorer that you can use to try Google Cloud APIs. Hi.

# Google Cloud client libraries

## What are the Google Cloud client libraries?

* The Google Cloud client libraries are the latest and recommended approach to making requests to the server.
* The cloud libraries make it easier to access API codes using your favorite programming language. Google Cloud client libraries handle low level communication with the server including authentication with Google.
* And they can be installed using familiar installation packages such as NPM and PIP. The client libraries also provide retry logic for transient network failures. Consult the client libraries documentation for more details. The links are available in the download pane below.
* Google Cloud client libraries provide idiomatic code in supported languages which makes it easier to work with.
* Some libraries give you performance benefits from gRPC. gRPC is the Google remote procedure calls. It is an open source remote procedure call framework that can be run anywhere. gRPC makes it easier to build connected systems because it enables client and server applications to communicate transparently.
* Google API client libraries should only be used if your programming language of choice isn't supported by the Google Cloud client libraries yet. They provide access to rest APIs only and do not support gRPC. For more information and resources, please click on the links provided in the download pane below.
* The language is supported by the Google Cloud client libraries are.Net, Go, Java, Node.js, PHP, Python and Ruby.
* To get started with the Google Cloud client libraries, you can pull the repo for the Google Cloud Client libraries from GitHub. It is provided for each of the supported programming languages. The GitHub repo page lists services and API supported by each languages Cloud Client Library and provides installation instructions. You can also download Cloud Client Libraries for individual cloud platform services. Reference libraries contain links to documentation as well as to relevant StackOverflow posts and provide code examples. The reference libraries are your one stop shop for information on a language specific Google Cloud client library. Every package uses a client as a base for interacting with an API.
* If your application is running on App Engine or Compute engine, authentication for your application will just work. If you don't explicitly provide credentials, the client will reuse the credentials from the gCloud tool assuming it has already been installed and authorized. The example provided demonstrates importing the data store client library, instantiating the client using default credentials and adding an entity to the data store. For more information on this example and for further resources, please click on the downloads pane below. 

# Google Cloud SDK

* The Google SDK consists of three command line tools: gCloud, BQ and gsUtil. These tools allow you to access GCP products and services. You can run your tools interactively or in your automated scripts.
* GCloud is a command line tool that allows you to perform common task on GCP. It allows you to create and manage GCP resources. In the example, we are listing all the compute engine VM instances for your project. <https://cloud.google.com/sdk/gcloud/reference/>
* BQ is a command line tool used to work with Google BigQuery. BQ's primary purpose is running queries and it can also be used to manage data sets, tables and other BigQuery entities.
* gsutil is a command line tool used to perform tasks in Google Cloud storage. You can use gsutil to create and manage buckets, upload, download and delete objects, move, copy and rename objects and manage access to stored objects. In the example, we are copying logo from our desktop to a bucket called my-awesome-bucket. You can download and install cloud SDK on Linux, Mac, OS X and Windows. You can install cloud SDK using apt get on DBN and Ubuntu. Initialize the Cloud SDK by running the command *gcloud init*. Once it is initialized, you can start using it. You can install and manage SDK components and use the gcloud interactive shell which provides prompt completion and suggest flags. You can even script gcloud commands to automate your processes.
* Google Cloud Shell is a free admin machine with browser based command line access. It allows you to easily manage your infrastructure and application on GCP. It gives you access to a temporary virtual machine instance with five gigs of persistent disk storage. When you start Cloud Shell, it provisions an F1 micro Google compute engine virtual machine running a Debian based Linux OS.
* Cloud shell instances are provisioned on a per user per session basis. The instances persist only while your cloud shell session is active and terminate after an hour of inactivity.
* Google Cloud SDK comes pre-installed in Cloud Shell. Additionally, Cloud Shell has built-in automation to your cloud platform console projects and resources.
* Cloud Shell comes with a built-in code editor based on Orion to browse file directories and view and edit files with continued access to Cloud Shell.
* Cloud Shell provides pre-installed language support for Java, Go, Python, Node.js, Ruby, PHP and.Net. Google Cloud platform provides cloud tools for various IDEs to facilitate development on Google Cloud Platform.
* Cloud's tools are supported for Android Studio, Eclipse, IntelliJ, PowerShell and Visual Studio. In the example, cloud developer tools for Visual Studio is allowing the developer to browse compute engine resources, storage buckets and cloud SQL instances from the Visual Studio IDE.

# Google Firebase SDK

Google firebase is a mobile and web application development platform. Supported platforms include Android, iOS, Web, C++, Unity and Node.js. Firebase is integrated with various services in GCP. The Firebase SDK for cloud storage store files directly in Google Cloud Storage buckets. And you can use the Google Cloud Storage APIs to access files uploaded via the Firebase SDKs for Cloud Storage. Firebase SDKs for cloud storage use the default bucket for Google App Engine standard environment. So you can use the built in App Engine API to share data between Firebase and your app engine application. Additionally, you can retrieve, verify and store user credentials using Firebase authentication, the Google App Engine standard environment and Google Cloud datastore. Google cloud functions is GCP's serverless offering. Cloud functions for Firebase lets you automatically run backend code in response to events triggered by Firebase features and HTTP requests and the code is stored in Google's cloud and run in a managed environment. No management or scaling of your own servers is required. For more information and use cases for a Firebase app with Google Cloud Vision and Speech APIs, please view the resources available in the downloads pane below.

# Lab

# App Dev: Setting up a Development Environment v1.1

2 hours1 Credit

Rate Lab

## Overview

In this lab, you will provision a Google Compute Engine virtual machine and install software libraries for Node.js software development on Google Cloud Platform.

## Objectives

In this lab, you will learn how to perform the following tasks:

* Provision a Google Compute Engine instance.
* Connect to the instance using SSH.
* Install software on the instance.
* Verify the software installation.

## Setup

#### What you'll need

To complete this lab, you’ll need:

* Access to a standard internet browser (Chrome browser recommended).
* Time. Note the lab’s **Completion** time in Qwiklabs. This is an estimate of the time it should take to complete all steps. Plan your schedule so you have time to complete the lab. Once you start the lab, you will not be able to pause and return later (you begin at step 1 every time you start a lab).
* The lab's **Access** time is how long your lab resources will be available. If you finish your lab with access time still available, you will be able to explore the Google Cloud Platform or work on any section of the lab that was marked "if you have time". Once the Access time runs out, your lab will end and all resources will terminate.
* You **DO NOT** need a Google Cloud Platform account or project. An account, project and associated resources are provided to you as part of this lab.
* If you already have your own GCP account, make sure you do not use it for this lab.
* If your lab prompts you to log into the console, **use only the student account provided to you by the lab**. This prevents you from incurring charges for lab activities in your personal GCP account.

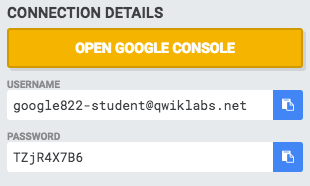
#### Start your lab

When you are ready, click **Start Lab**. You can track your lab’s progress with the status bar at the top of your screen.

**Important** What is happening during this time? Your lab is spinning up GCP resources for you behind the scenes, including an account, a project, resources within the project, and permission for you to control the resources needed to run the lab. This means that instead of spending time manually setting up a project and building resources from scratch as part of your lab, you can begin learning more quickly.

#### Find Your Lab’s GCP Username and Password

To access the resources and console for this lab, locate the Connection Details panel in Qwiklabs. Here you will find the account ID and password for the account you will use to log in to the Google Cloud Platform:



If your lab provides other resource identifiers or connection-related information, it will appear on this panel as well.

## Task 1: Creating a Compute Engine Virtual Machine Instance

In this section, you will use the Google Cloud Platform Console to provision a new Google Compute Engine virtual machine instance.

1. In the **Cloud Platform Console**, on the **Navigation menu**, click **Compute Engine**.
2. On the **VM Instances** page, click **Create**.
3. On the **Create an instance** page, for **Name** type **dev-instance**, select **us-central1** for region and **us-central1-a** for the zone.

**GCP Regions and Zones**

Google Cloud Platform offers products and services in multiple distinct geographic locations, called regions.

Each region has multiple distinct zones. Each zone is isolated from other zones in terms of power and internet connectivity.

1. In the **Identity and API access > Access Scopes** section, select **Allow full access to all Cloud APIs**.
2. In the **Firewall** section, enable **Allow HTTP traffic**.
3. Leave the remaining settings as their defaults, and click **Create**.

It takes about 20 seconds for the virtual machine to be provisioned and started.

1. On the **VM instances** page, in the row for the **dev-instance**, click **SSH** (in the **Connect**column).

This launches a browser-hosted SSH session. If you have a popup blocker, you may need to click twice.

There's no need to configure or manage SSH keys.

Click Check my progress to verify the objective.

Create a Compute Engine Virtual Machine Instance

Check my progress

## Task 2: Install software on the VM instance

1. In the SSH session, to update the Debian package list, execute the following command:
2. sudo apt-get update
3. To install Git, execute the following command:
4. sudo apt-get install git

If prompted, press Enter.

1. To download the Node.js setup script, execute the following command:
2. curl -sL https://deb.nodesource.com/setup\_6.x | sudo -E bash -
3. To install npm and Node.js, execute the following command:
4. sudo apt install nodejs

Click Check my progress to verify the objective.

Install software on the VM instance

Check my progress

## Task 3: Configure the VM to Run Application Software

In this section, you will verify the software installation and run some sample codes.

1. To check the version of Node.js, execute the following command:
2. node -v

You should see the Node.js version number for version 6.

1. To clone the class repository, execute the following command:
2. git clone https://github.com/GoogleCloudPlatform/training-data-analyst
3. To change the working directory, execute the following command:
4. cd ~/training-data-analyst/courses/developingapps/nodejs/devenv/
5. To run a simple web server, execute the following command:
6. sudo node server/app.js
7. Return to the Cloud Console VM instances list, and click on the External IP address for the dev-instance.

You should see a Hello GCP dev! message from Node.js.

1. Return to the SSH window, and stop the application by pressing **Ctrl+C**.
2. To install the Node.js library for Compute Engine, execute the following command:
3. npm install
4. To run a simple Node.js application that lists Compute Engine instances, execute the following command:
5. node list-gce-instances.js

Many details about your machine should appear in the terminal window.

Warning: If you try to do this on your own machine, it will not work if no credentials have been set up to access GCP on your machine.

Click Check my progress to verify the objective.

Clone the repository

Check my progress

## End your lab

When you have completed your lab, click **End Lab**. Qwiklabs removes the resources you’ve used and cleans the account for you.

You will be given an opportunity to rate the lab experience. Select the applicable number of stars, type a comment, and then click **Submit**.

The number of stars indicates the following:

* 1 star = Very dissatisfied
* 2 stars = Dissatisfied
* 3 stars = Neutral
* 4 stars = Satisfied
* 5 stars = Very satisfied

You can close the dialog box if you don't want to provide feedback.

For feedback, suggestions, or corrections, please use the **Support** tab.

©Google, Inc. or its affiliates. All rights reserved. Do not distribute.