TY B.Tech Computer Engineering
2018 Course Cloud Computing

SUBMITTED TO: PROF. VISHAL MESHRAM

VISHWAKARMA INSTITUTE OF INFORMATION TECHNOLOGY, PUNE

COMPUTER ENGINEERING DEPARTMENT

SUBMITTED BY:

NAME: AMEY SUNIL BOBADE

G.R NO.: 21810866

ROLL NO.: 321007

CLASS: COMP A

BATCH: A1

Assignment No 1

Title: Study the Google Cloud Platform.
Theory : link
(https://cloud.google.com/docs)
(https://www.youtube.com/playlist?list=PLIivdWyY5sqIUkH9XhgMkAOyIwCN4H0As)

What Is Cloud Computing?

Cloud computing, in simple words, is accessing and storing data over the Internet instead of doing it on your personal hard drive.
It offers services like storage, database, networking, and more over the Internet to provide faster, innovative, and flexible resources to its customers. The customers get to pay only for the resources they use, hence helping them lower their operating costs and run their business infrastructure more efficiently.

Now, among various cloud providers like AWS, Microsoft, VMWare, IBM, etc., Google Cloud has been the talk of the town in recent years and there are enough reasons behind it. Let's begin with starters and understand what Google Cloud is.

What Is Google Cloud Platform?

Google Cloud is a suite of Cloud Computing services offered by Google. The platform provides various services like compute, storage, networking, Big Data, and many more that run on the same infrastructure that Google uses internally for its end users like Google Search and YouTube.

Google server hasn't gone down in years. So, if you are planning to run your application on the Google Cloud infrastructure, then you can be assured of your applications being safe and secure.
ABOUT GOOGLE CLOUD SERVICES
(HTTPS://CLOUD.GOOGLE.COM/DOCS/OVERVIEW/CLOUD-PLATFORM-SERVICES)

This overview introduces some of the commonly used Google Cloud services. For the full list of services, see the Products and services page.

This overview covers the following types of services:

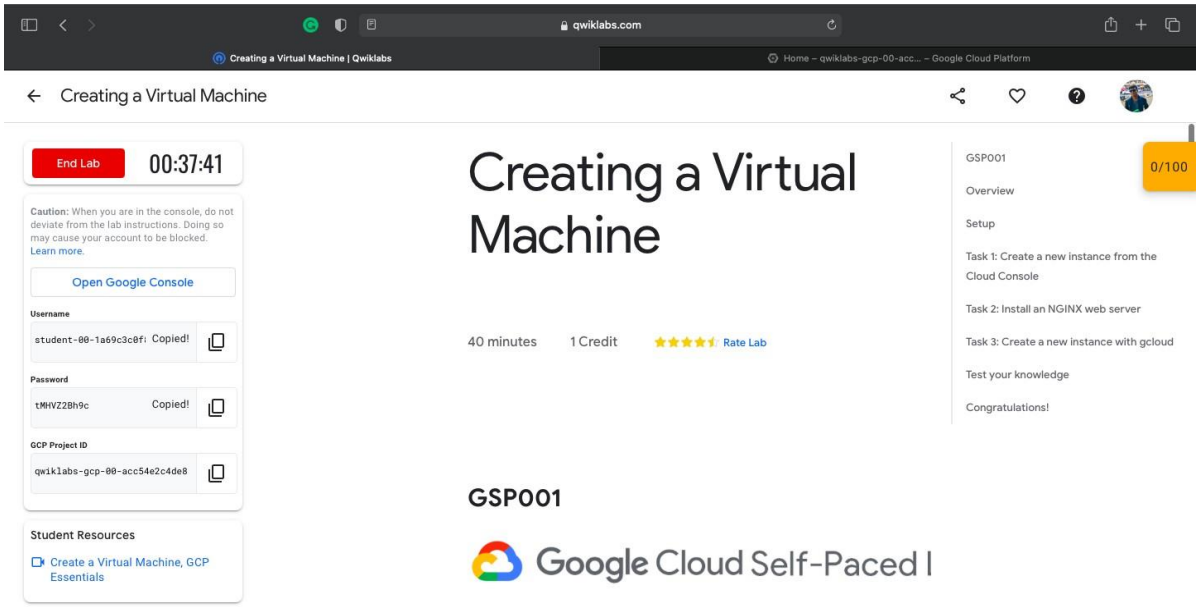
- Computing and hosting
- Storage
- Databases
- Networking
- Big data
- Machine learning

OUT of Google Product you have to study 3 services in detail:

1. **Compute** (https://cloud.google.com/docs#section-7)

a)In compute write detail steps of how to create a Linux virtual machine instance in Compute Engine using the Google Cloud Console. (Reference: https://cloud.google.com/compute/docs/quickstart-linux)

1. Click the **Start Lab** button. If you need to pay for the lab, a pop-up opens for you to select your payment method. On the left is a panel populated with the temporary credentials that you must use for this lab.



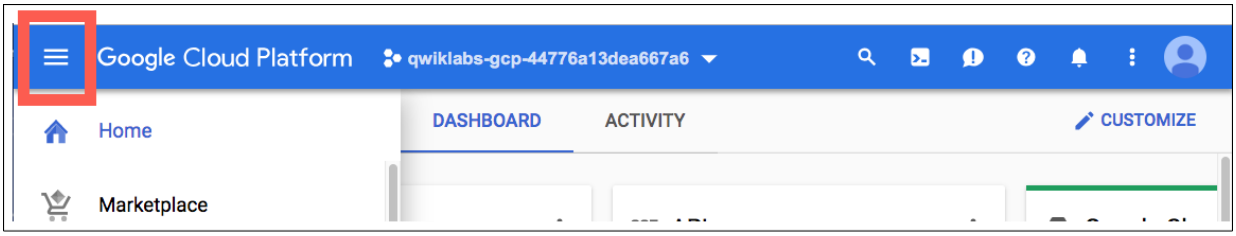
2. Copy the username, and then click **Open Google Console**. The lab spins up resources, and then opens another tab that shows the **Sign in** page.

    1. In the **Sign in** page, paste the username that you copied from the Connection Details panel. Then copy and paste the password.

    2. Click through the subsequent pages:

        - Accept the terms and conditions.
        - Do not add recovery options or two-factor authentication (because this is a temporary account).
        - Do not sign up for free trials.

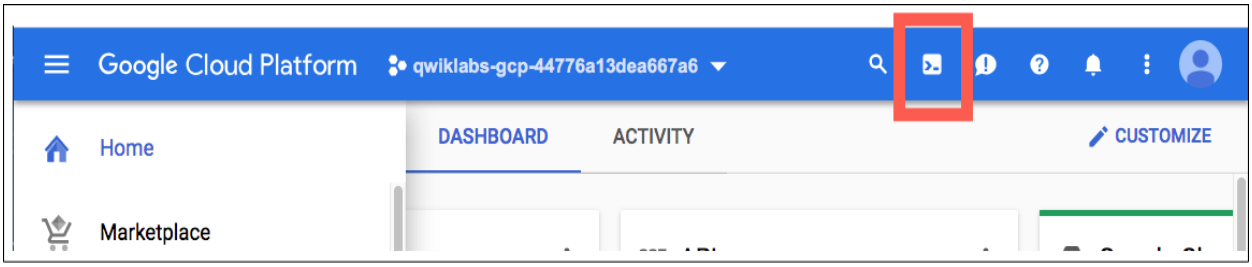
After a few moments, the Cloud Console opens in this tab.



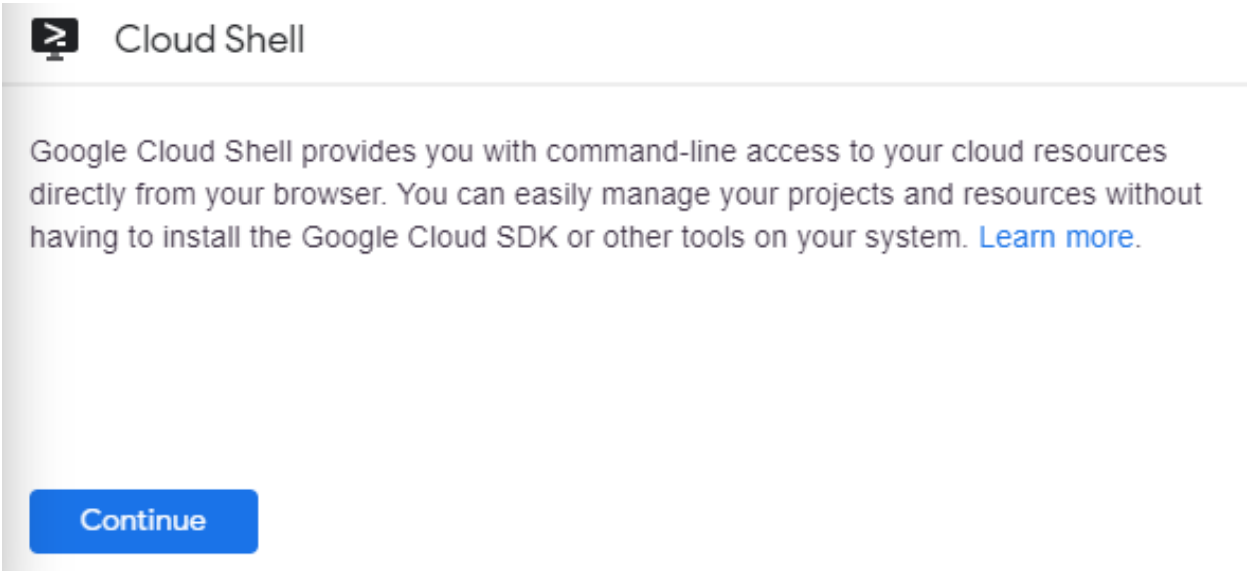
Activate Cloud Shell

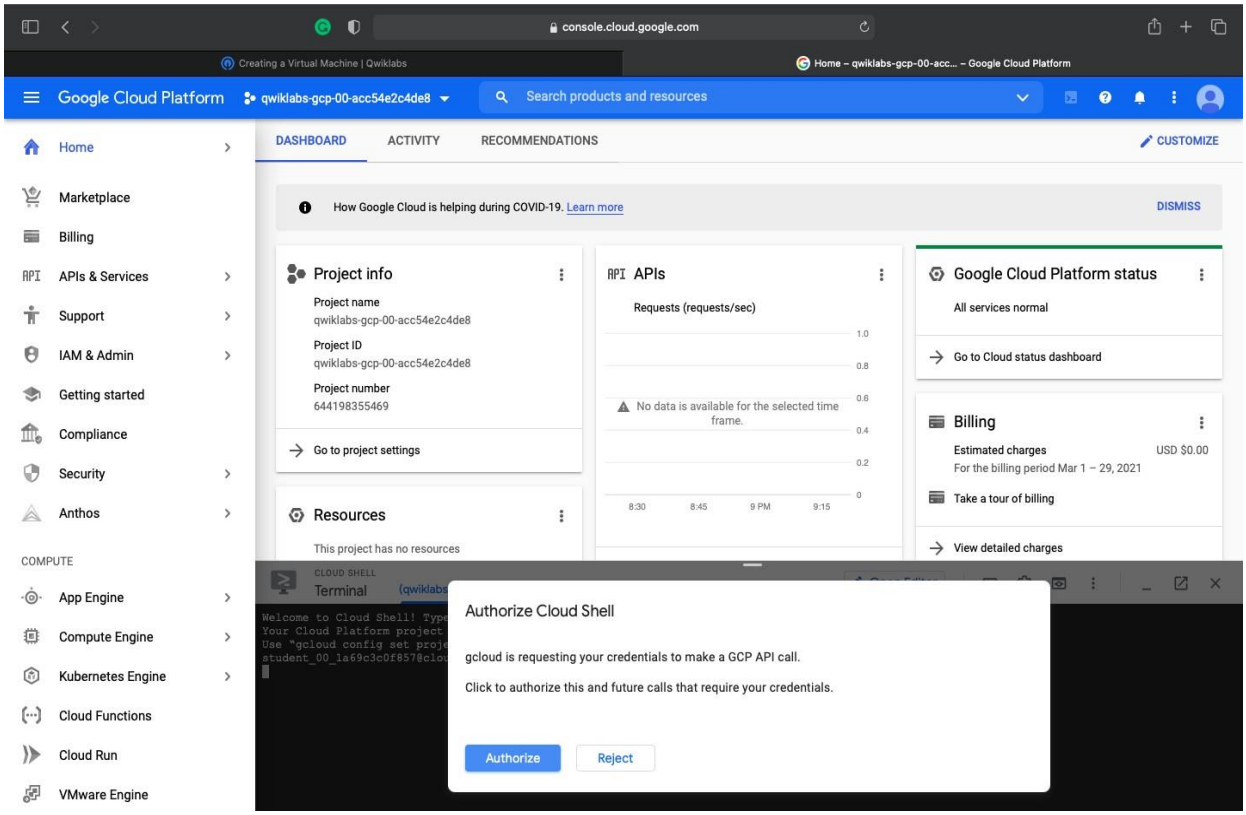
Cloud Shell is a virtual machine that is loaded with development tools. It offers a persistent 5GB home directory and runs on the Google Cloud. Cloud Shell provides command-line access to your Google Cloud resources.

In the Cloud Console, in the top right toolbar, click the **Activate Cloud Shell** button.

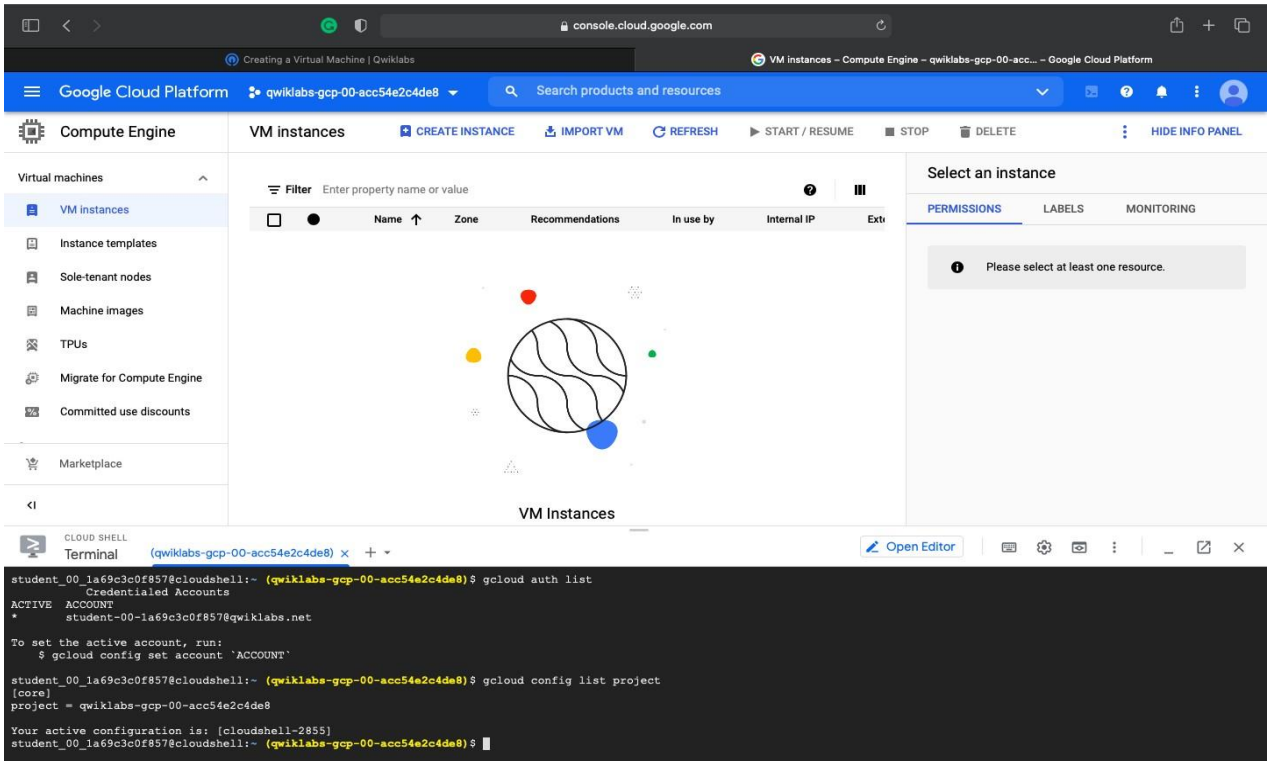


Click **Continue**.





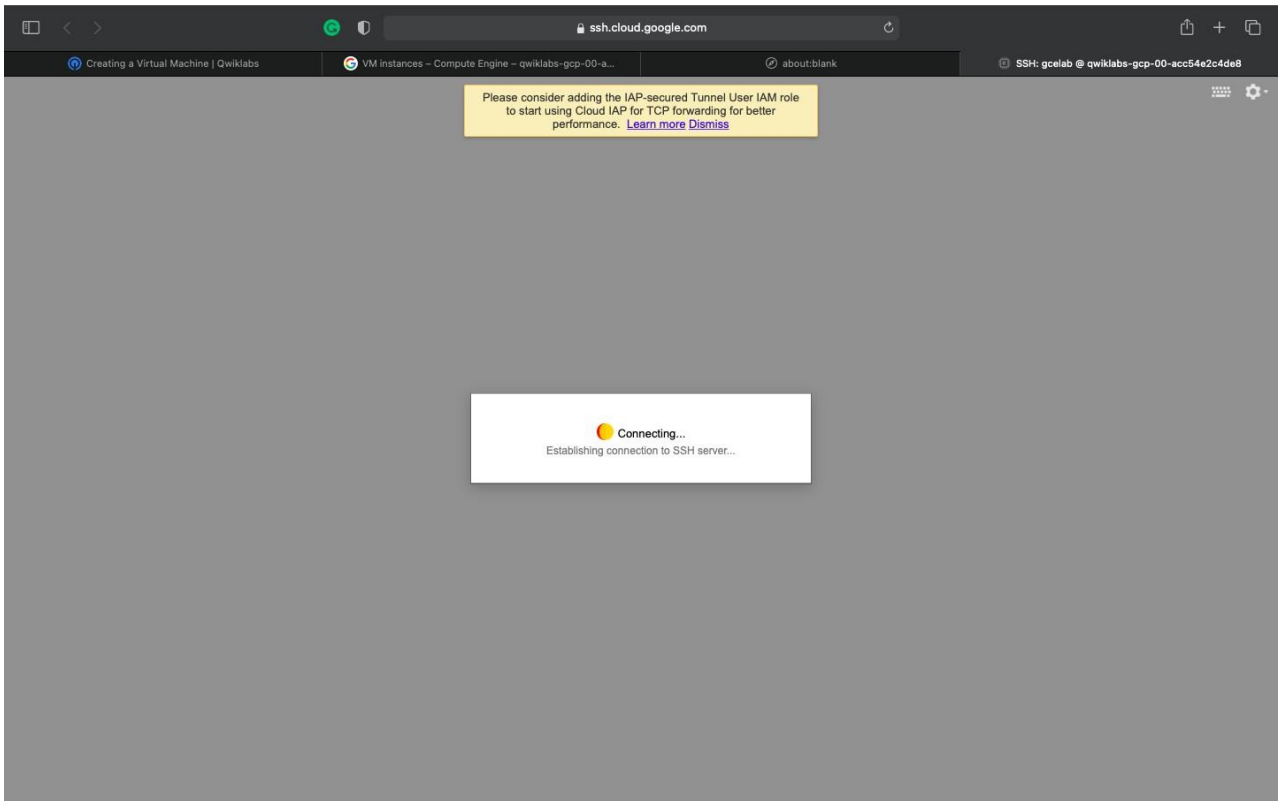
You can list the active account name with the written commands in console :



a) Write detail steps for Deploy a simple Apache web server to learn the basics of running a server on a virtual machine instance. (Reference: https://cloud.google.com/compute/docs/tutorials/basic-webserver-apache)

Now you'll install an NGINX web server, one of the most popular web servers in the world, to connect your virtual machine to something.

1. In the SSH terminal, to get root access, run the following command:



In the SSH terminal, to get root access, run the following command:

```
sudo su -
```

- As the `root` user, update your OS:

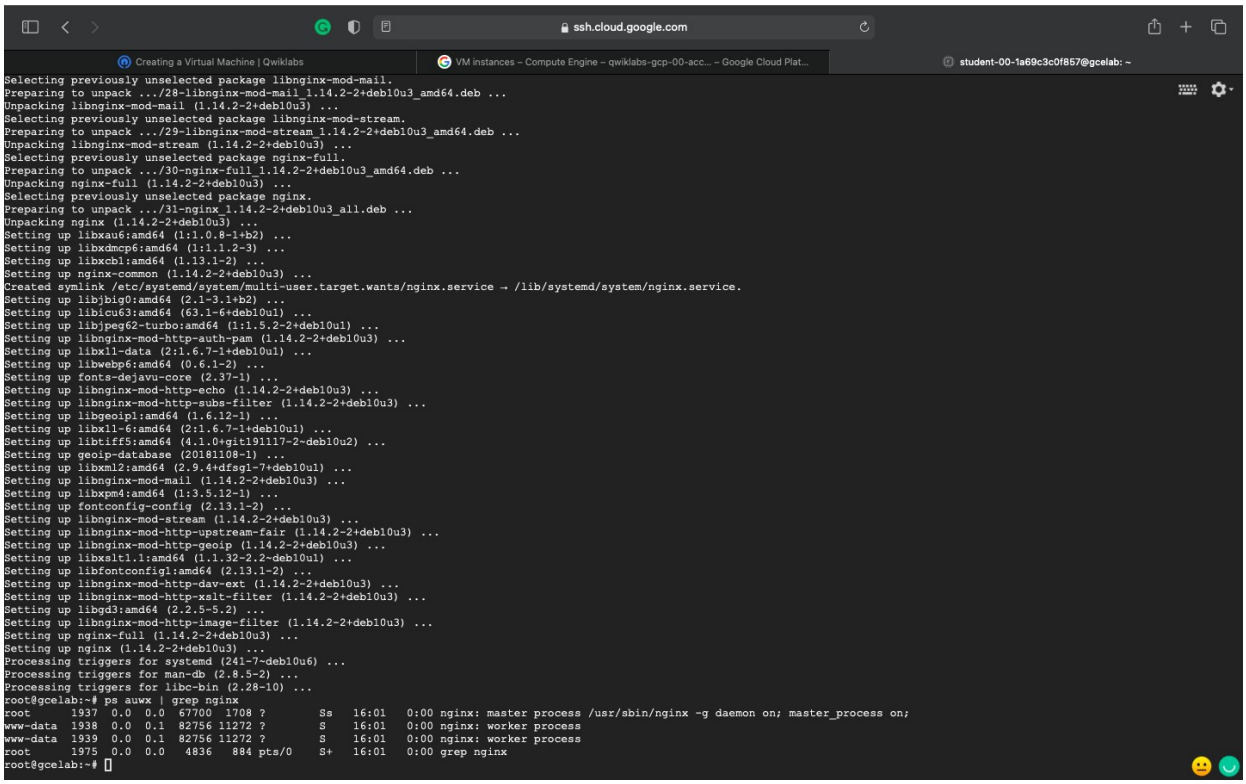
```
apt-get update


Get:1 http://security.debian.org stretch/updates InRelease [94.3 kB]
Ign http://deb.debian.org strech InRelease
Get:2 http://deb.debian.org strech-updates InRelease [91.0 kB]
```

1. Install **NGINX**:

```
apt-get install nginx -y
```

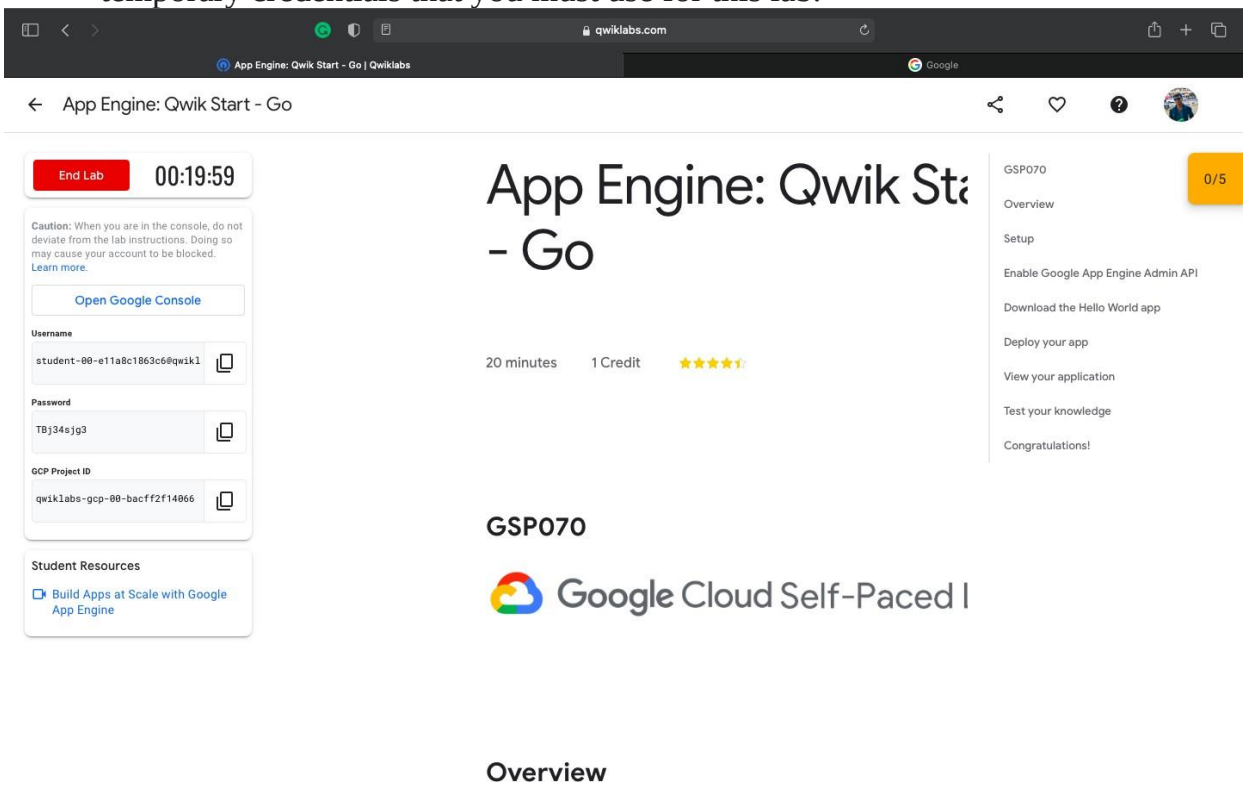
content_copy

```
Reading package lists... Done
Building dependency tree
```



2. **GOOGLE APP ENGINE** (REF: [HTTPS://CLOUD.GOOGLE.COM/APPENGINE/DOCS)](HTTPS://CLOUD.GOOGLE.COM/APPENGINE/DOCS)

- Click the **Start Lab** button. If you need to pay for the lab, a pop-up opens for you to select your payment method. On the left is a panel populated with the temporary credentials that you must use for this lab.



Copy the username, and then click **Open Google Console**. The lab spins up resources, and then opens another tab that shows the **Sign in** page.

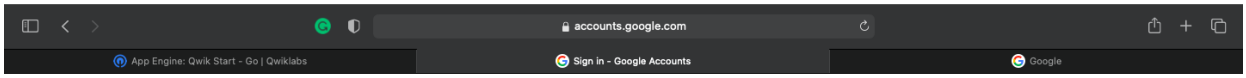
1. In the **Sign in** page, paste the username that you copied from the Connection Details panel. Then copy and paste the password.

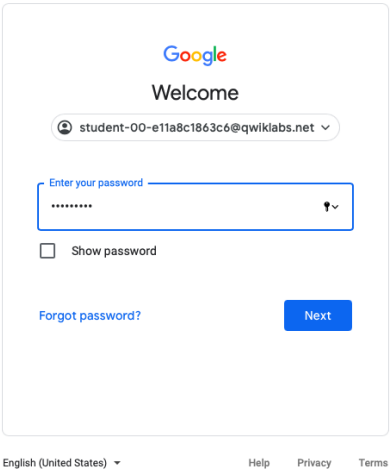
   Important: You must use the credentials from the Connection Details panel. Do not use your Qwiklabs credentials. If you have your own Google Cloud account, do not use it for this lab (avoids incurring charges).

2. Click through the subsequent pages:

    - Accept the terms and conditions.
    - Do not add recovery options or two-factor authentication (because this is a temporary account).
    - Do not sign up for free trials.

After a few moments, the Cloud Console opens in this tab.

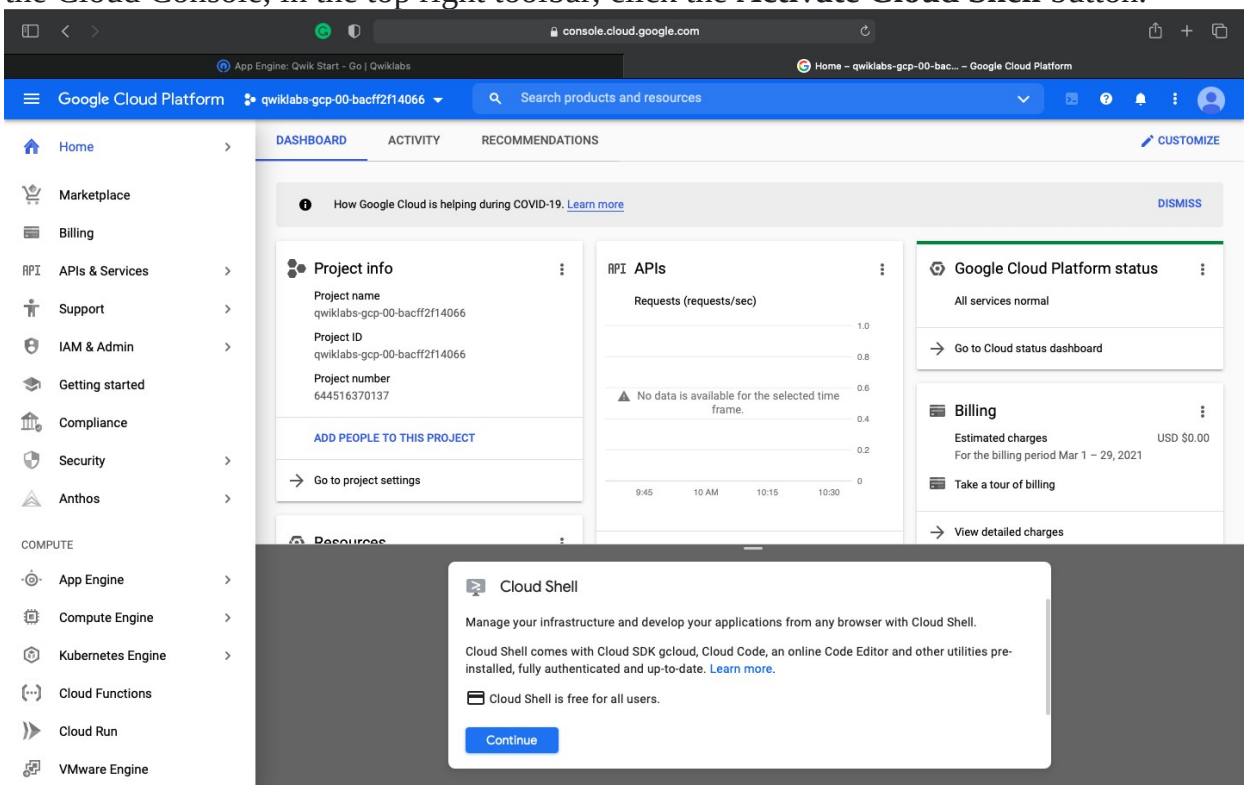




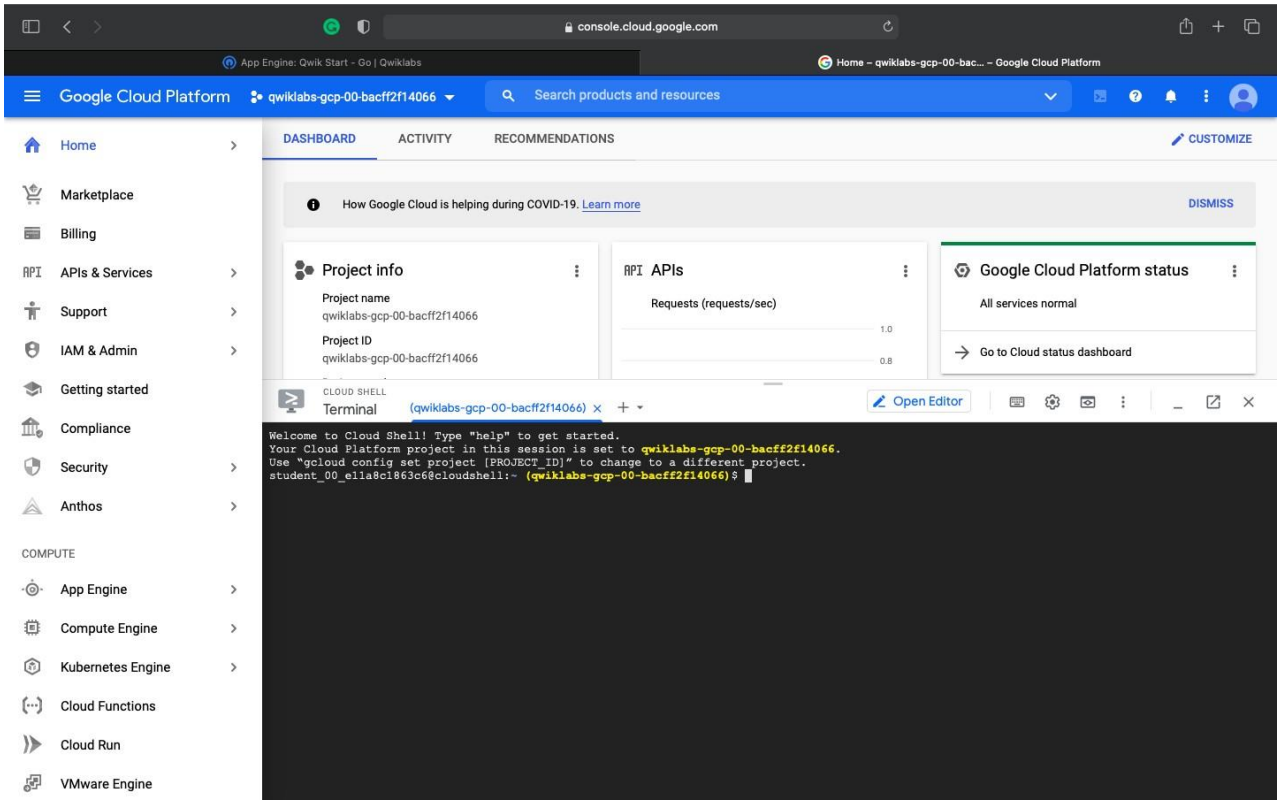
ACTIVATE CLOUD SHELL

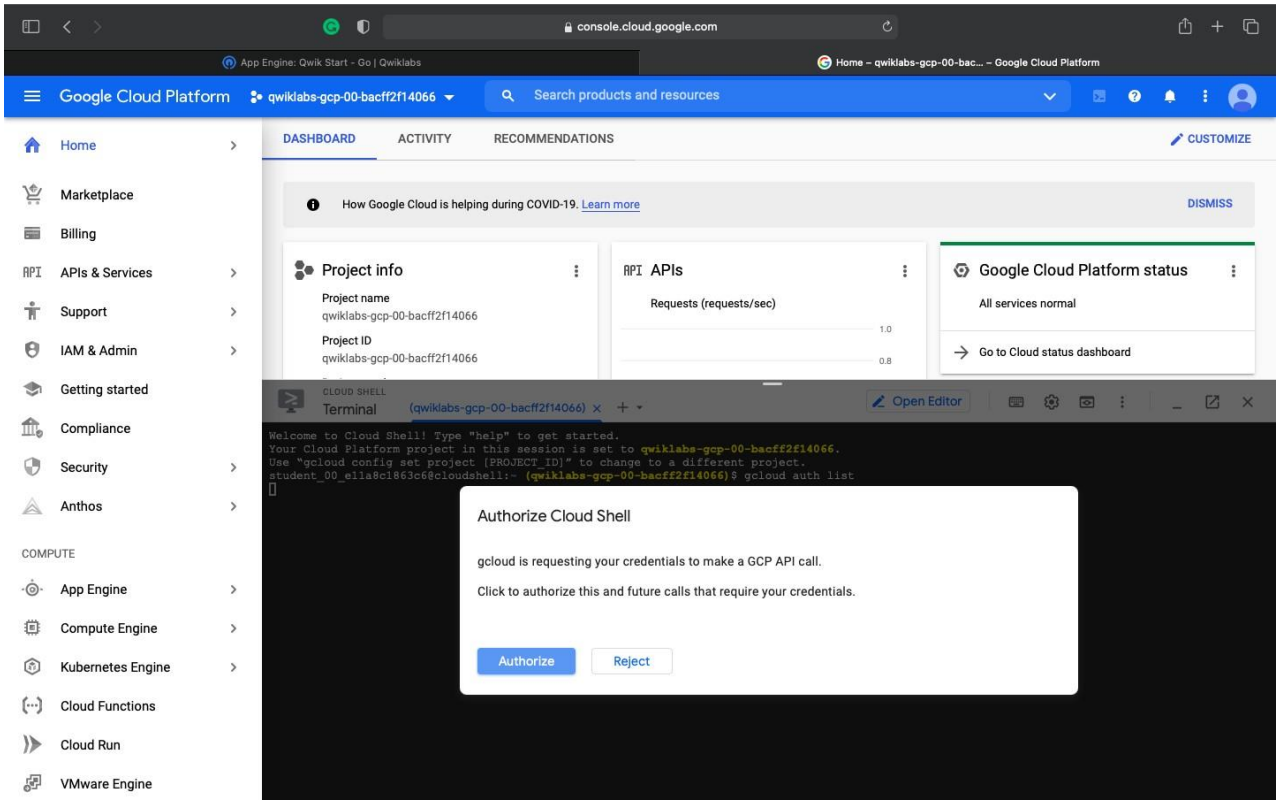
Cloud Shell is a virtual machine that is loaded with development tools. It offers a persistent 5GB home directory and runs on the Google Cloud. Cloud Shell provides command-line access to your Google Cloud resources.

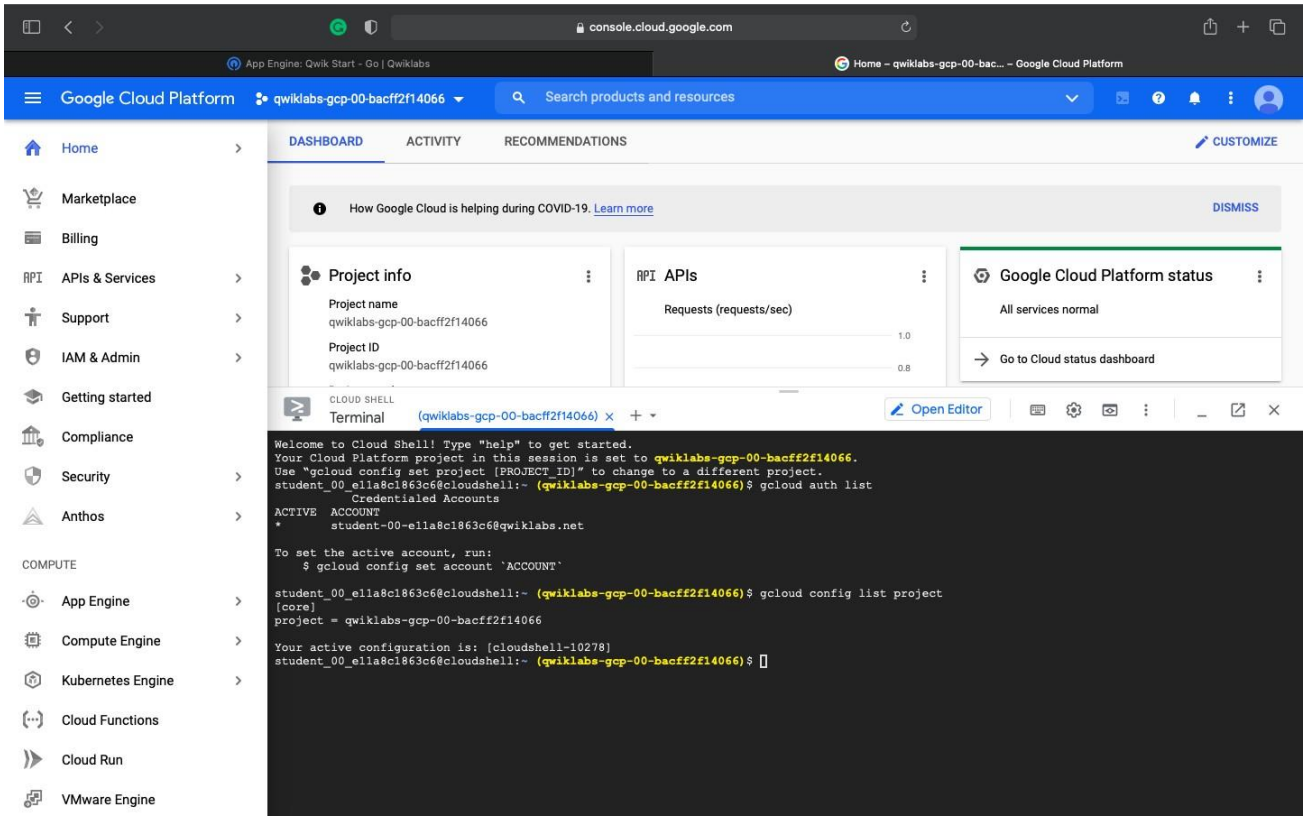
In the Cloud Console, in the top right toolbar, click the **Activate Cloud Shell** button.



It takes a few moments to provision and connect to the environment. When you are connected, you are already authenticated, and the project is set to your *PROJECT_ID*. For example:


console.cloud.google.com
App Engine: Qwik Start - Go | Qwiklabs
Home – qwiklabs-gcp-00-bac... – Google Cloud Platform
Google Cloud Platform
qwiklabs-gcp-00-bacff2f14066
Search products and resources
Home
Marketplace
Billing
APIs & Services
Support
IAM & Admin
Getting started
Compliance
Security
Anthos
COMPUTE
App Engine
Compute Engine
Kubernetes Engine
Cloud Functions
Cloud Run
VMware Engine
DASHBOARD
ACTIVITY
RECOMMENDATIONS
CUSTOMIZE
How Google Cloud is helping during COVID-19. Learn more
DISMISS
Project info
Project name
qwiklabs-gcp-00-bacff2f14066
Project ID
qwiklabs-gcp-00-bacff2f14066
APIs
Requests (requests/sec)
1.0
0.8
Google Cloud Platform status
All services normal
Go to Cloud status dashboard
CLOUD SHELL
Terminal
(qwiklabs-gcp-00-bacff2f14066)
Open Editor
Welcome to Cloud Shell! Type "help" to get started.
Your Cloud Platform project in this session is set to qwiklabs-gcp-00-bacff2f14066.
Use "gcloud config set project [PROJECT_ID]" to change to a different project.
student_00_e11a8c1863c6@cloudshell:~ (qwiklabs-gcp-00-bacff2f14066)$

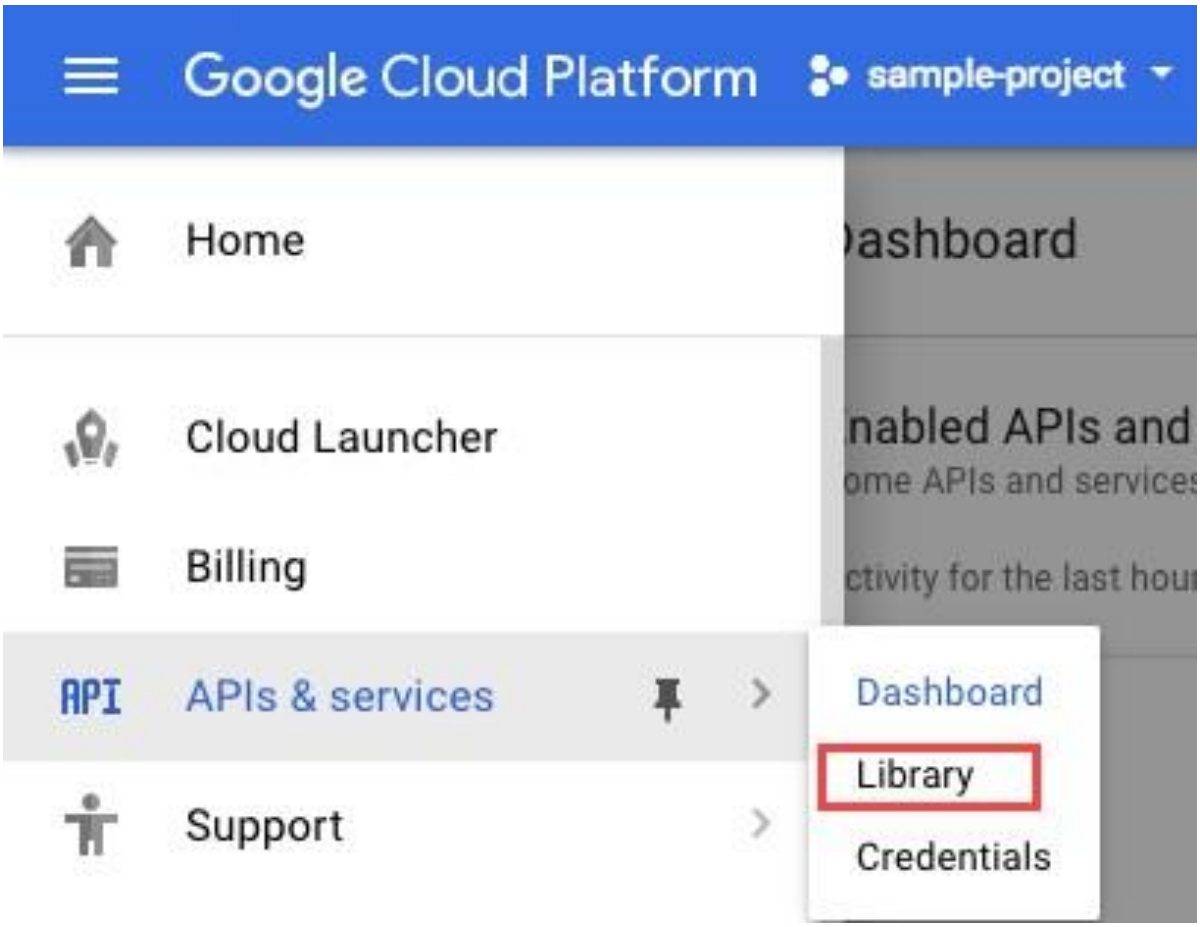

console.cloud.google.com
App Engine: Qwik Start - Go | Qwiklabs
Home – qwiklabs-gcp-00-bac... – Google Cloud Platform
Google Cloud Platform
qwiklabs-gcp-00-bacff2f14066
Search products and resources
Home
Marketplace
Billing
APIs & Services
Support
IAM & Admin
Getting started
Compliance
Security
Anthos
COMPUTE
App Engine
Compute Engine
Kubernetes Engine
Cloud Functions
Cloud Run
VMware Engine
DASHBOARD
ACTIVITY
RECOMMENDATIONS
CUSTOMIZE
How Google Cloud is helping during COVID-19. Learn more
DISMISS
Project info
Project name
qwiklabs-gcp-00-bacff2f14066
Project ID
qwiklabs-gcp-00-bacff2f14066
APIs
Requests (requests/sec)
1.0
0.8
Google Cloud Platform status
All services normal
Go to Cloud status dashboard
CLOUD SHELL
Terminal
(qwiklabs-gcp-00-bacff2f14066)
Open Editor
Welcome to Cloud Shell! Type "help" to get started.
Your Cloud Platform project in this session is set to qwiklabs-gcp-00-bacff2f14066.
Use "gcloud config set project [PROJECT_ID]" to change to a different project.
student_00_e11a8c1863c6@cloudshell:~ (qwiklabs-gcp-00-bacff2f14066)$ gcloud auth list
Authorize Cloud Shell
gcloud is requesting your credentials to make a GCP API call.
Click to authorize this and future calls that require your credentials.
Authorize
Reject


console.cloud.google.com
App Engine: Qwik Start - Go | Qwiklabs
Home – qwiklabs-gcp-00-bac... – Google Cloud Platform
Google Cloud Platform
qwiklabs-gcp-00-bacff2f14066
Search products and resources
Home
Marketplace
Billing
APIs & Services
Support
IAM & Admin
Getting started
Compliance
Security
Anthos
COMPUTE
App Engine
Compute Engine
Kubernetes Engine
Cloud Functions
Cloud Run
VMware Engine
DASHBOARD
ACTIVITY
RECOMMENDATIONS
CUSTOMIZE
How Google Cloud is helping during COVID-19. Learn more
DISMISS
Project info
Project name
qwiklabs-gcp-00-bacff2f14066
Project ID
qwiklabs-gcp-00-bacff2f14066
APIs
Requests (requests/sec)
1.0
0.8
Google Cloud Platform status
All services normal
Go to Cloud status dashboard
CLOUD SHELL
Terminal
(qwiklabs-gcp-00-bacff2f14066)
Open Editor
Welcome to Cloud Shell! Type "help" to get started.
Your Cloud Platform project in this session is set to qwiklabs-gcp-00-bacff2f14066.
Use "gcloud config set project [PROJECT_ID]" to change to a different project.
student_00_e11a8c1863c6@cloudshell:~ (qwiklabs-gcp-00-bacff2f14066)$ gcloud auth list
Credentialed Accounts
ACTIVE ACCOUNT
* student-00-e11a8c1863c6@qwiklabs.net
To set the active account, run:
$ gcloud config set account `ACCOUNT`
student_00_e11a8c1863c6@cloudshell:~ (qwiklabs-gcp-00-bacff2f14066)$ gcloud config list project
[core]
project = qwiklabs-gcp-00-bacff2f14066
Your active configuration is: [cloudshell-10278]
student_00_e11a8c1863c6@cloudshell:~ (qwiklabs-gcp-00-bacff2f14066)$

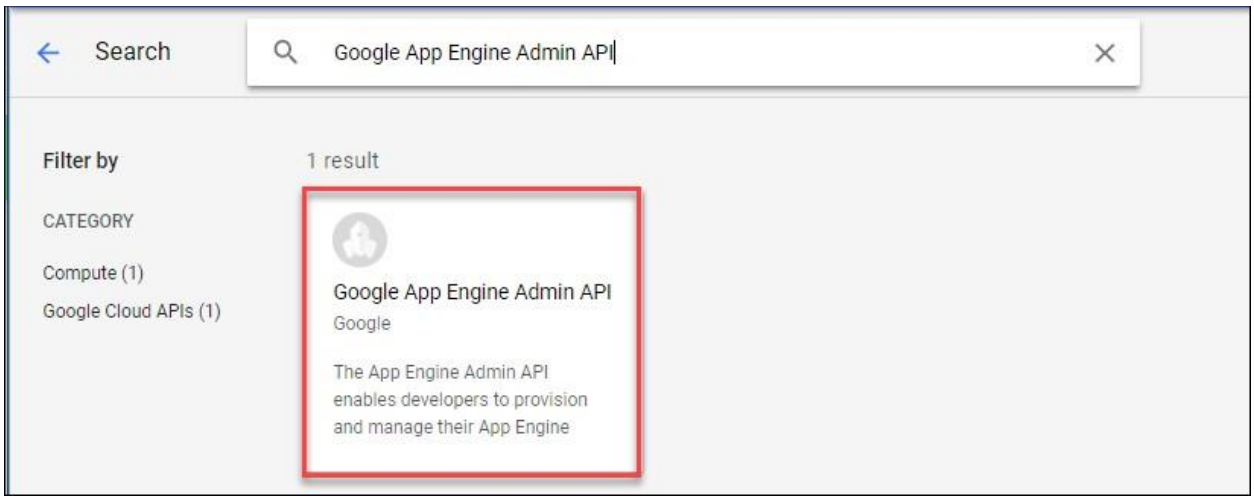
Enable Google App Engine Admin API

The App Engine Admin API enables developers to provision and manage their App Engine Applications.

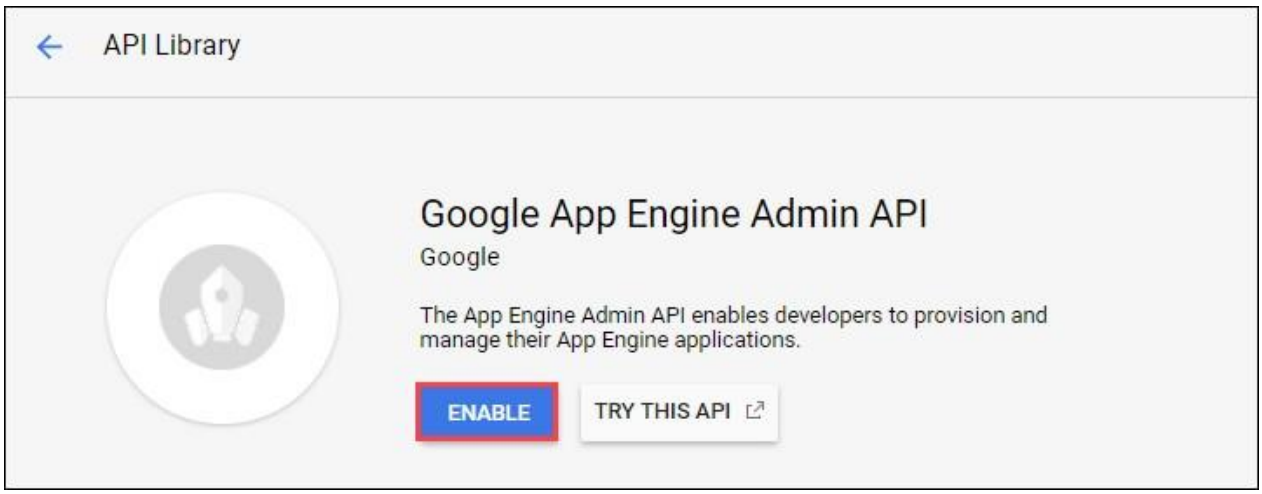
1. In the left menu, click **APIs & Services** > **Library**.

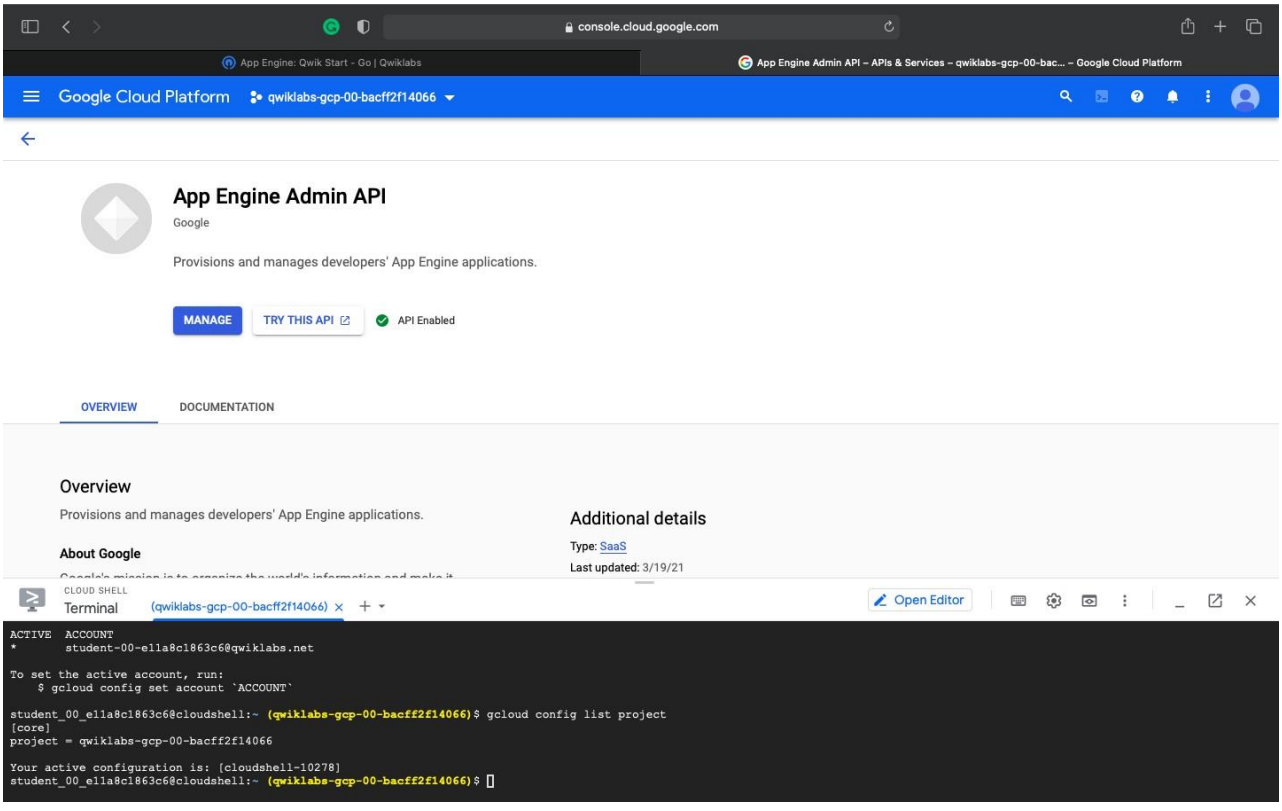


2. Type "App Engine Admin API" in search box.
3. Click **App Engine Admin API**.



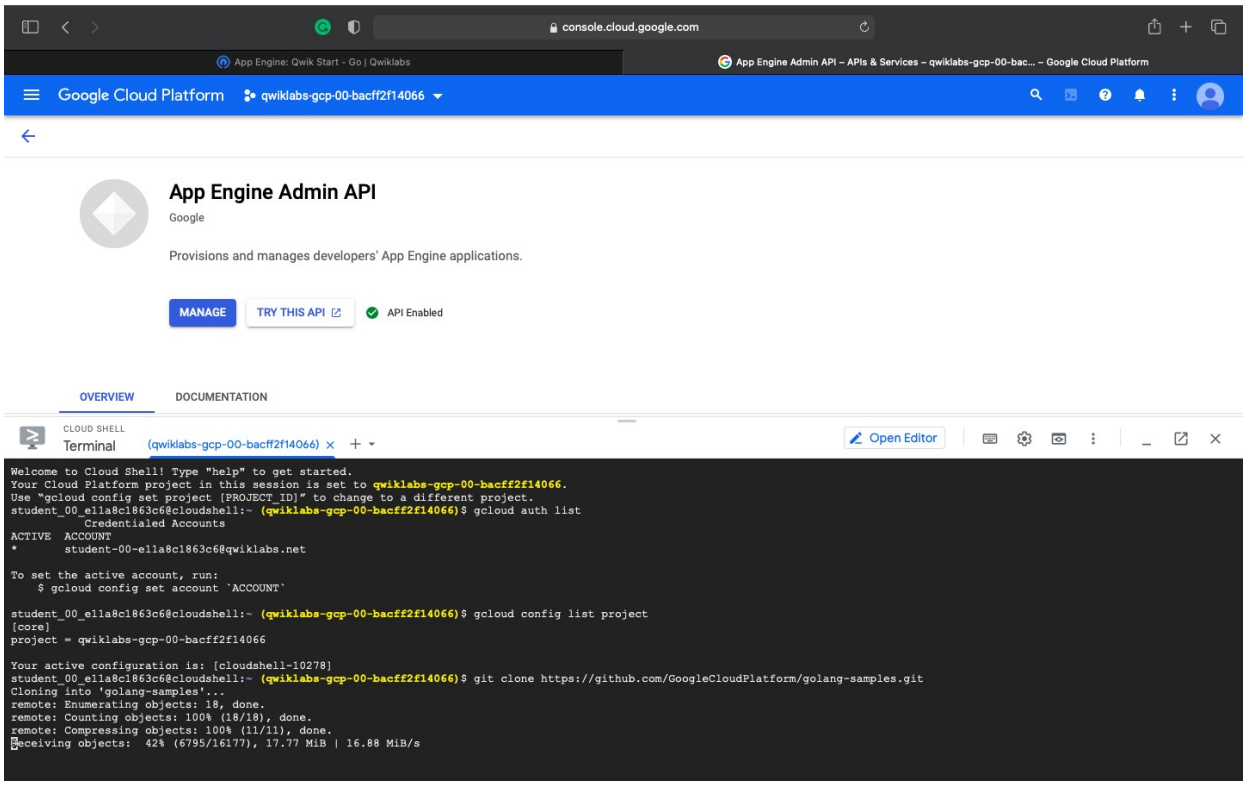
4. Click Enable.


API Library
Google App Engine Admin API
Google
The App Engine Admin API enables developers to provision and manage their App Engine applications.
ENABLE
TRY THIS API

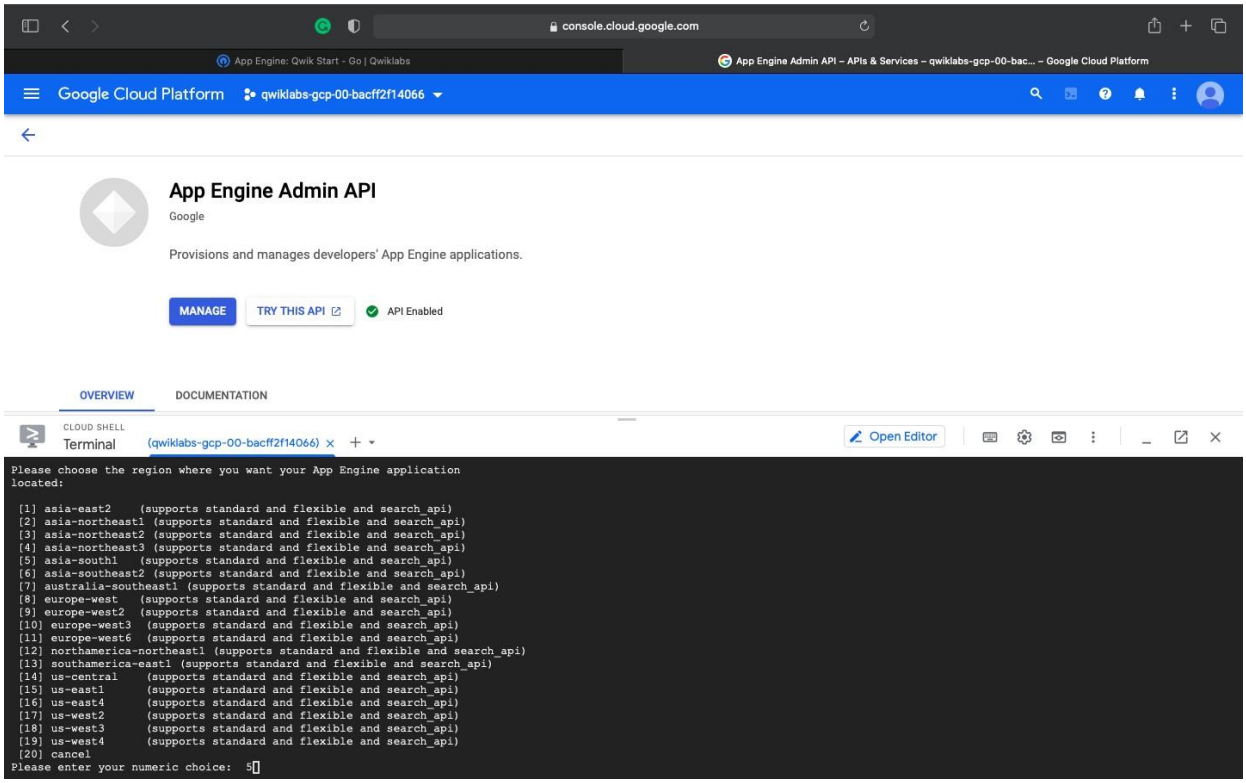

console.cloud.google.com
App Engine: Qwik Start - Go | Qwiklabs
App Engine Admin API – APIs & Services – qwiklabs-gcp-00-bac... – Google Cloud Platform
Google Cloud Platform
qwiklabs-gcp-00-bacff2f14066
App Engine Admin API
Google
Provisions and manages developers' App Engine applications.
MANAGE
TRY THIS API
API Enabled
OVERVIEW
DOCUMENTATION
Overview
Provisions and manages developers' App Engine applications.
About Google
Additional details
Type: SaaS
Last updated: 3/19/21
CLOUD SHELL
Terminal
(qwiklabs-gcp-00-bacff2f14066)
Open Editor
ACTIVE ACCOUNT
* student-00-e11a8c1863c6@qwiklabs.net
To set the active account, run:
$ gcloud config set account `ACCOUNT`
student_00_e11a8c1863c6@cloudshell:~ (qwiklabs-gcp-00-bacff2f14066)$ gcloud config list project
[core]
project = qwiklabs-gcp-00-bacff2f14066
Your active configuration is: [cloudshell-10278]
student_00_e11a8c1863c6@cloudshell:~ (qwiklabs-gcp-00-bacff2f14066)$

Deploy your app

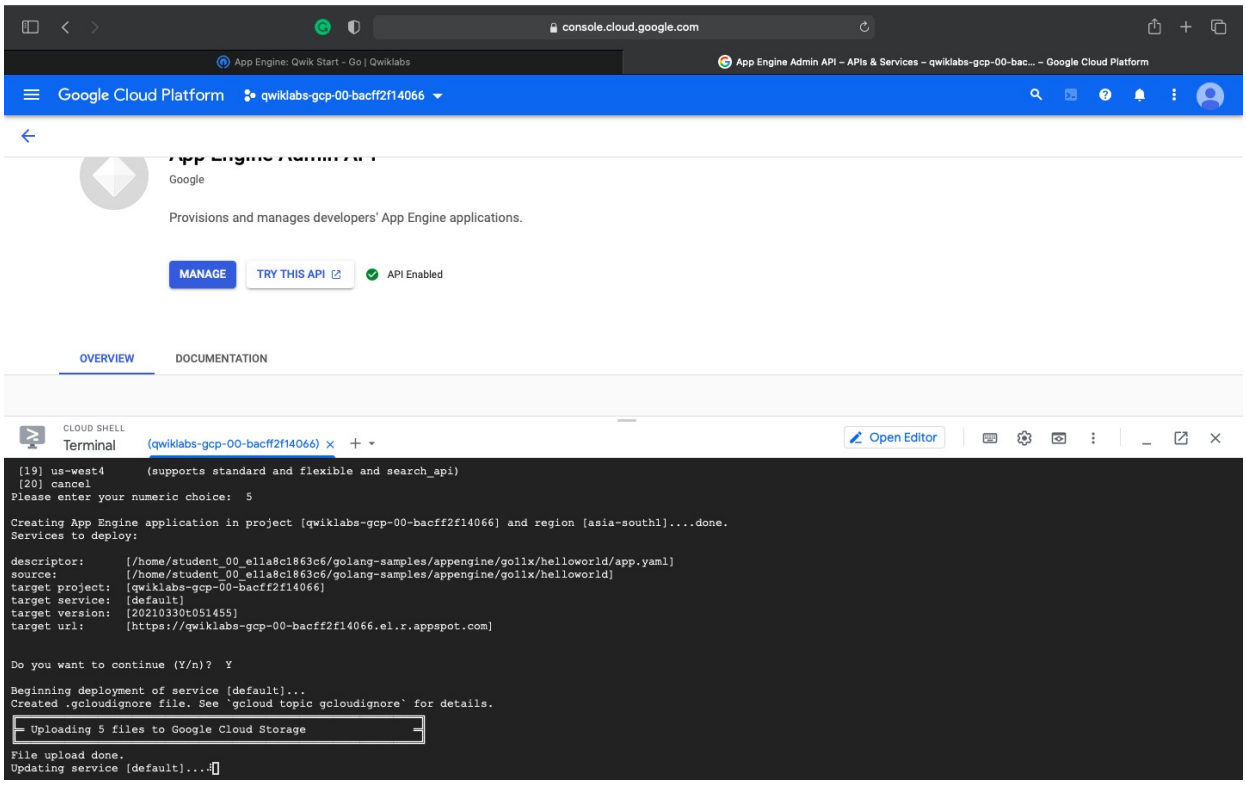
1. To deploy your app to App Engine, run the following command from within the root directory of your application where the app.yaml file is located:

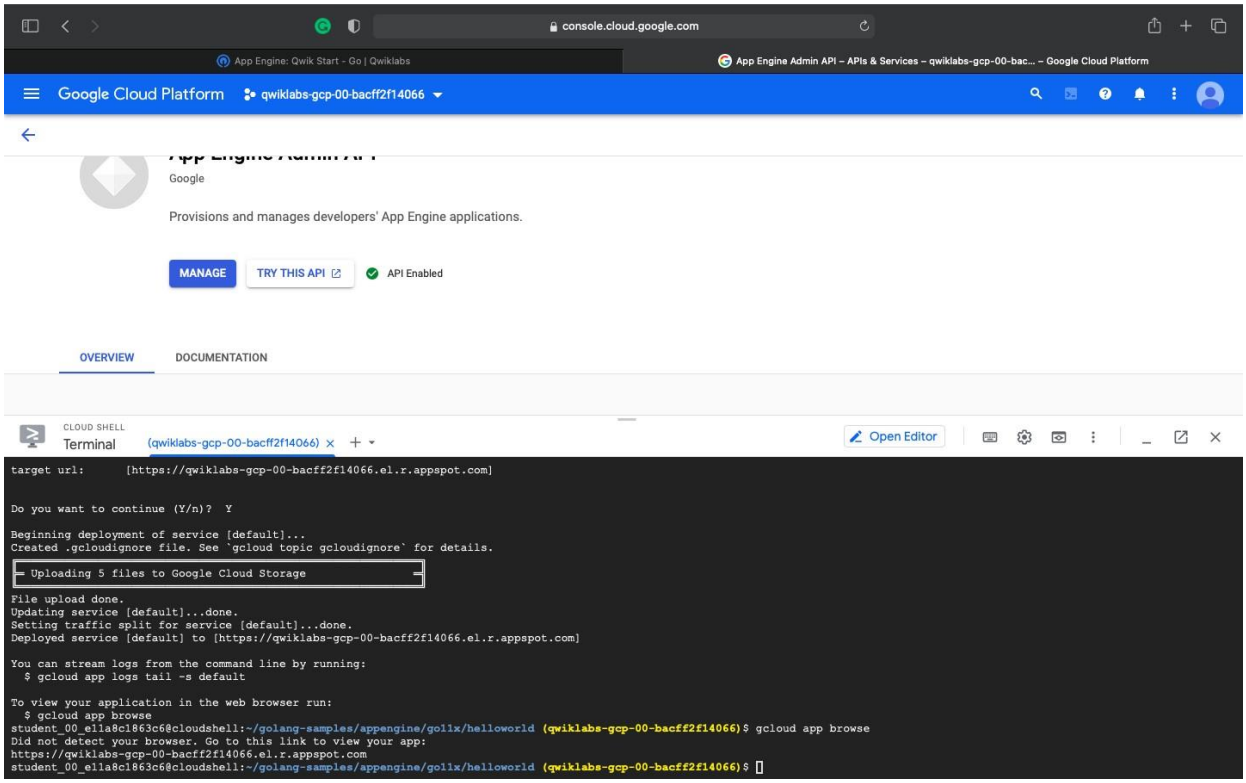


2. Type the number for your region.



3. Enter **Y** to confirm the deployment of service when prompted.




console.cloud.google.com
App Engine: Qwik Start - Go | Qwiklabs
App Engine Admin API – APIs & Services – qwiklabs-gcp-00-bac... – Google Cloud Platform
Google Cloud Platform
qwiklabs-gcp-00-bacff2f14066
Google
Provisions and manages developers' App Engine applications.
MANAGE
TRY THIS API
API Enabled
OVERVIEW
DOCUMENTATION
CLOUD SHELL
Terminal
(qwiklabs-gcp-00-bacff2f14066)
Open Editor
target url: [https://qwiklabs-gcp-00-bacff2f14066.el.r.appspot.com]
Do you want to continue (Y/n)? Y
Beginning deployment of service [default]...
Created .gcloudignore file. See `gcloud topic gcloudignore` for details.
Uploading 5 files to Google Cloud Storage
File upload done.
Updating service [default]...done.
Setting traffic split for service [default]...done.
Deployed service [default] to [https://qwiklabs-gcp-00-bacff2f14066.el.r.appspot.com]
You can stream logs from the command line by running:
$ gcloud app logs tail -s default
To view your application in the web browser run:
$ gcloud app browse
student_00_e11a8c1863c6@cloudshell:~/golang-samples/appengine/go11x/helloworld (qwiklabs-gcp-00-bacff2f14066)$ gcloud app browse
Did not detect your browser. Go to this link to view your app:
https://qwiklabs-gcp-00-bacff2f14066.el.r.appspot.com
student_00_e11a8c1863c6@cloudshell:~/golang-samples/appengine/go11x/helloworld (qwiklabs-gcp-00-bacff2f14066)$

VIEW YOUR APPLICATION

Your application is deployed and you can read the short message in your browser.

Click **Check my progress** to verify the objective.

3) Firebase database service (https://firebase.google.com/docs/database/)

CHOOSE A DATABASE: CLOUD FIRESTORE OR REALTIME DATABASE

Firebase offers two cloud-based, client-accessible database solutions that support realtime data syncing:

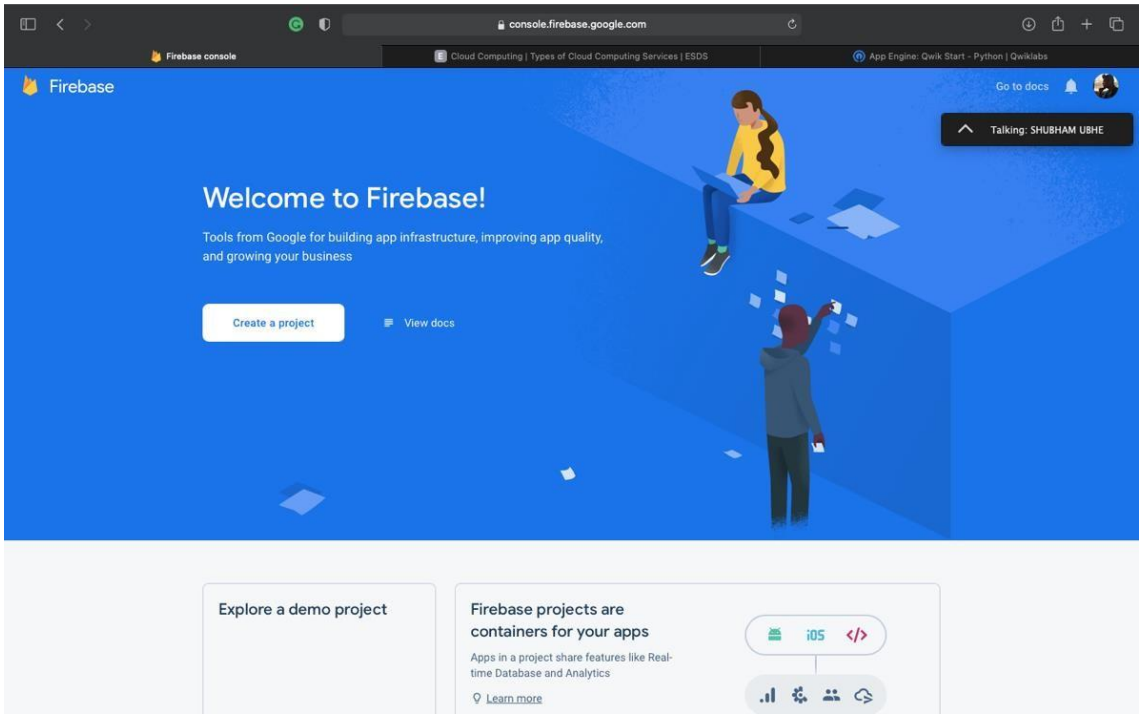
- **Cloud Firestore** is Firebase's newest database for mobile app development. It builds on the successes of the Realtime Database with a new, more intuitive data model. Cloud Firestore also features richer, faster queries and scales further than the Realtime Database.
- **Realtime Database** is Firebase's original database. It's an efficient, low-latency solution for mobile apps that require synced states across clients in realtime.

Firebase Realtime Database :
Store and sync data with our NoSQL cloud database. Data is synced across all clients in Realtime, and remains available when your app goes offline.

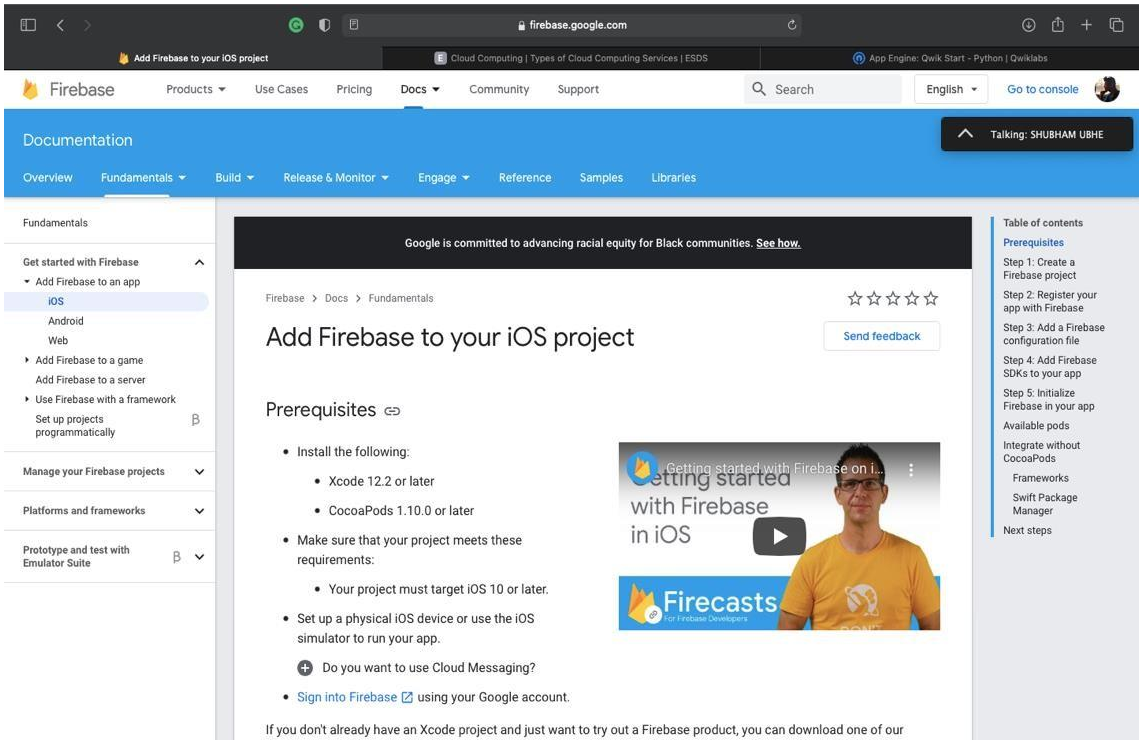
The Firebase Realtime Database is a cloud-hosted database. Data is stored as JSON and synchronized in Realtime to every connected client. When you build cross-platform apps with our iOS, Android, and JavaScript SDKs, all of your clients share one Realtime Database instance and automatically receive updates with the newest data.

iOS setup | Android setup | Web setup | REST API | C++ setup | Unity setup | Admin setup

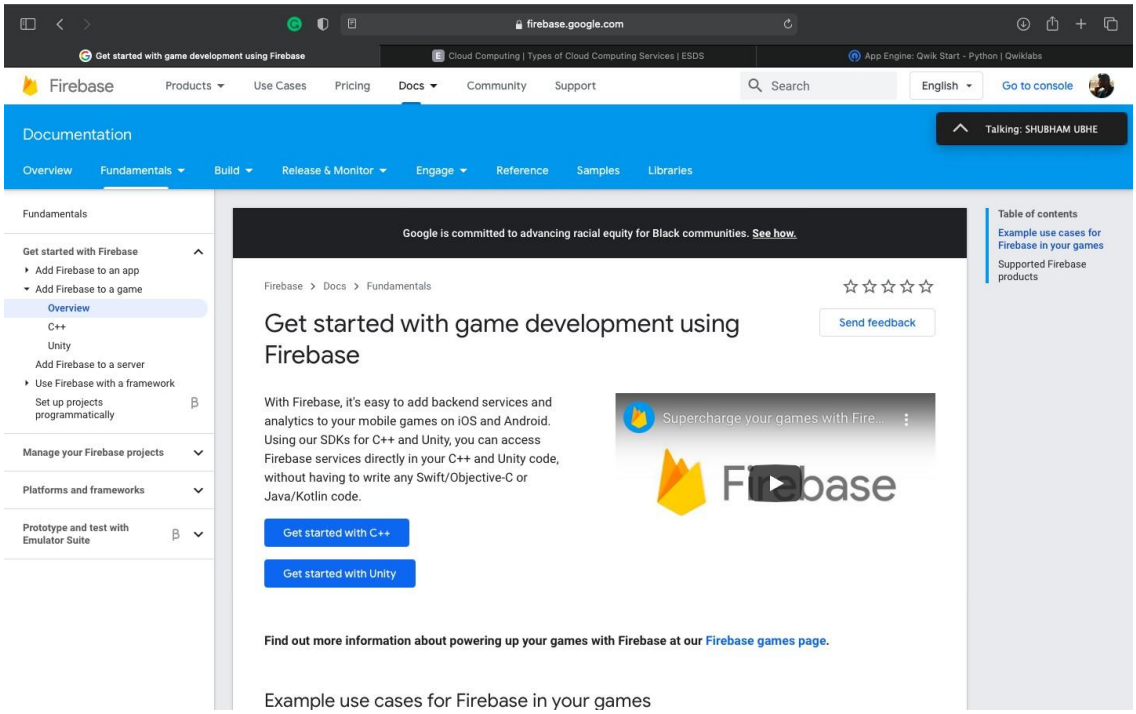
You can create a Firebase Project:



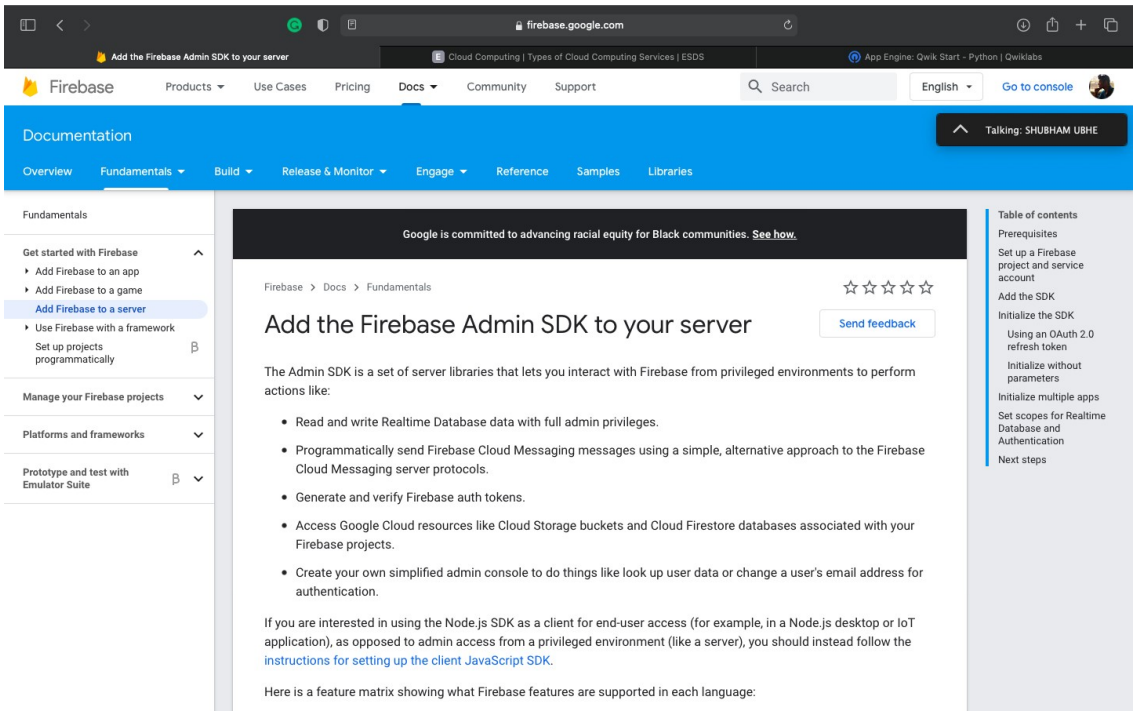
1) You can add firebase to an app.



2)You can add Firebase to an game.



3)You can add Firebase to a server.



Assignment No 2

Title: Develop an application and use Google Firbase service to store your data.

Theory: Cloud Firestore is a cloud-hosted, NoSQL database that your iOS, Android, and web apps can access directly via native SDKs. Cloud Firestore is also available in native Node.js, Java, Python, Unity, C++ and Go SDKs, in addition to REST and RPC APIs.

Following Cloud Firestore's NoSQL data model, you store data in documents that contain fields mapping to values. These documents are stored in collections, which are containers for your documents that you can use to organize your data and build queries. Documents support many different data types, from simple strings and numbers, to complex, nested objects. You can also create subcollections within documents and build hierarchical data structures that scale as your database grows. The Cloud Firestore data model supports whatever data structure works best for your app.

Additionally, querying in Cloud Firestore is expressive, efficient, and flexible. Create shallow queries to retrieve data at the document level without needing to retrieve the entire collection, or any nested subcollections. Add sorting, filtering, and limits to your queries or cursors to paginate your results. To keep data in your apps current, without retrieving your entire database each time an update happens, add realtime listeners. Adding realtime listeners to your app notifies you with a data snapshot whenever the data your client apps are listening to changes, retrieving only the new changes.

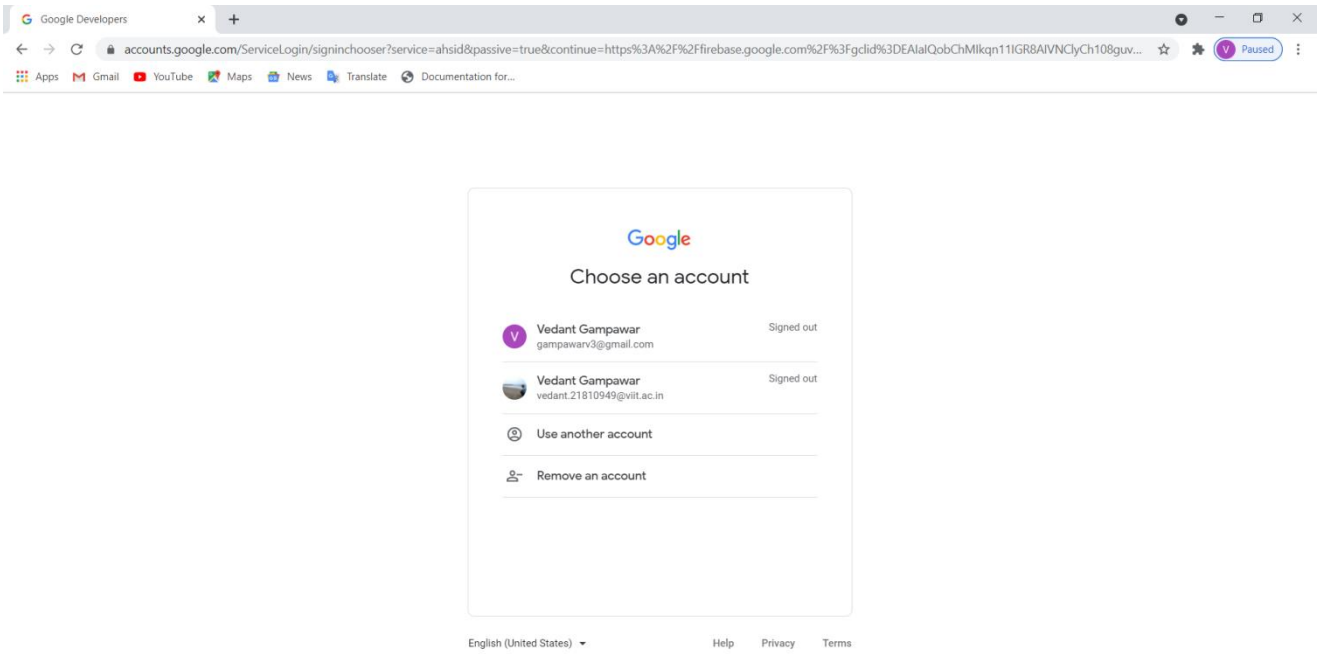
Protect access to your data in Cloud Firestore with Firebase Authentication and Cloud Firestore Security Rules for Android, iOS, and JavaScript, or Identity and Access Management (IAM) for server-side languages.

What is Firebase?

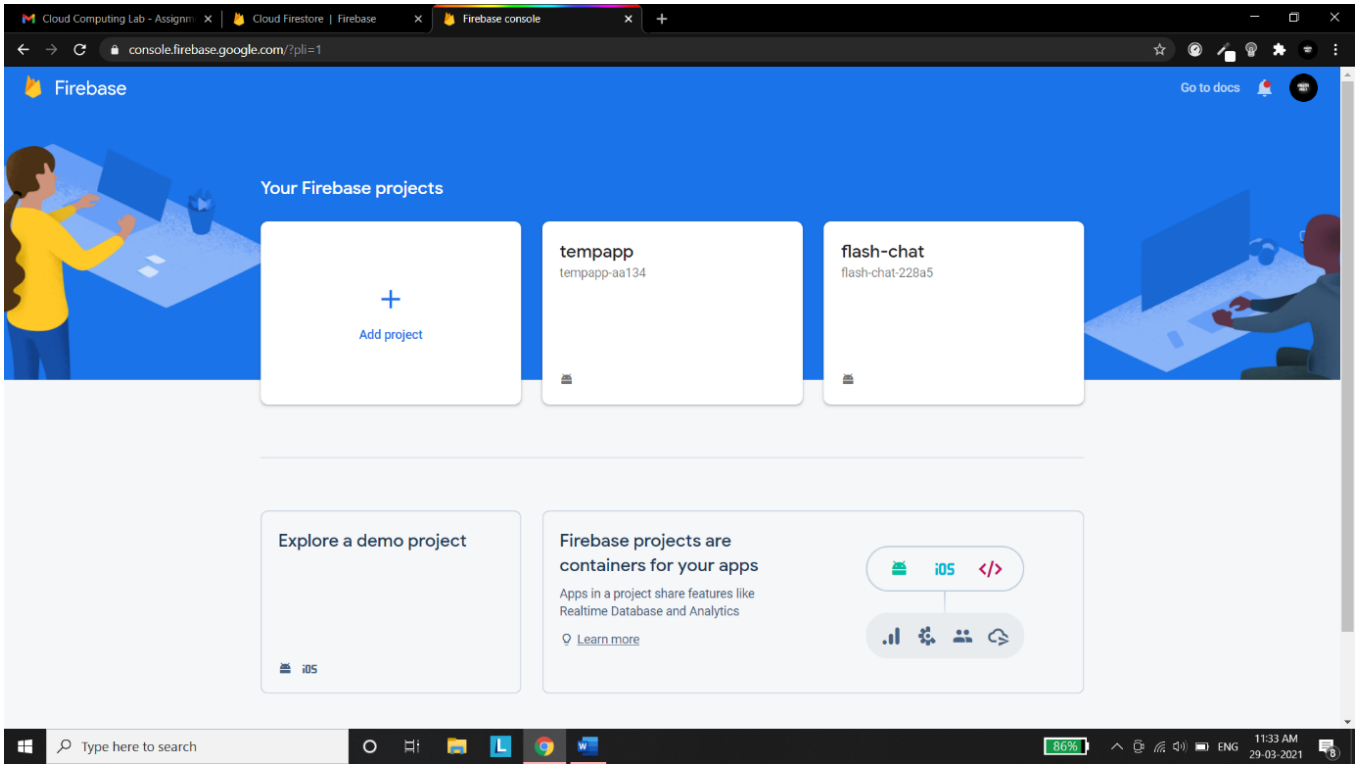
Cloud Firestore is a flexible, scalable database for mobile, web, and server development from Firebase and Google Cloud. Like Firebase Realtime Database, it keeps your data in sync across client apps through realtime listeners and offers offline support for mobile and web so you can build responsive apps that work regardless of network latency or Internet connectivity. Cloud Firestore also offers seamless integration with other Firebase and Google Cloud products, including Cloud Functions.

How to use Firebase:

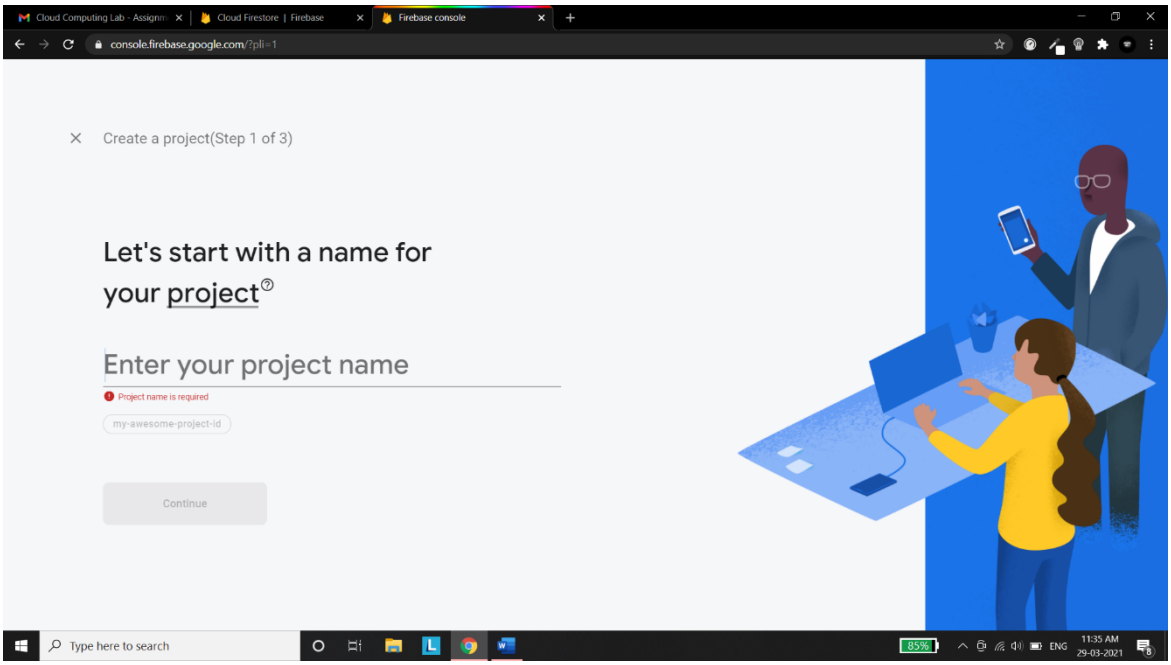
STEP 1): Sign In with Your Google account



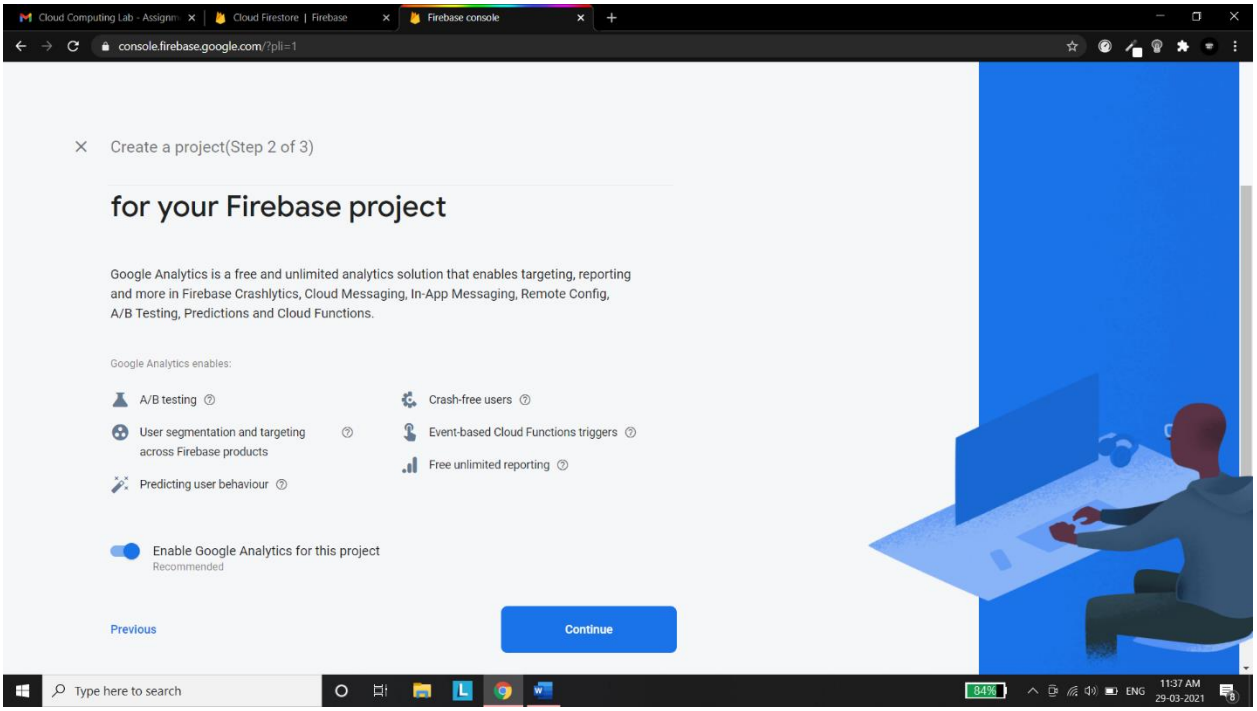
Step 2): Navigate to firebase console and add your project



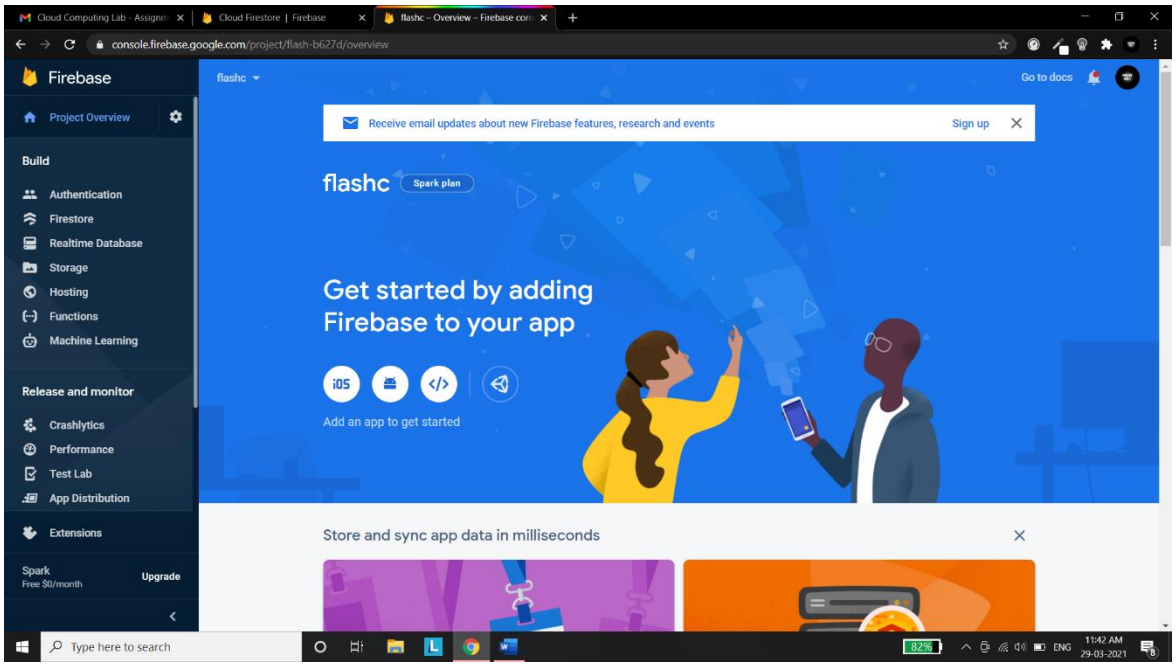
Step 3): Enter The project name



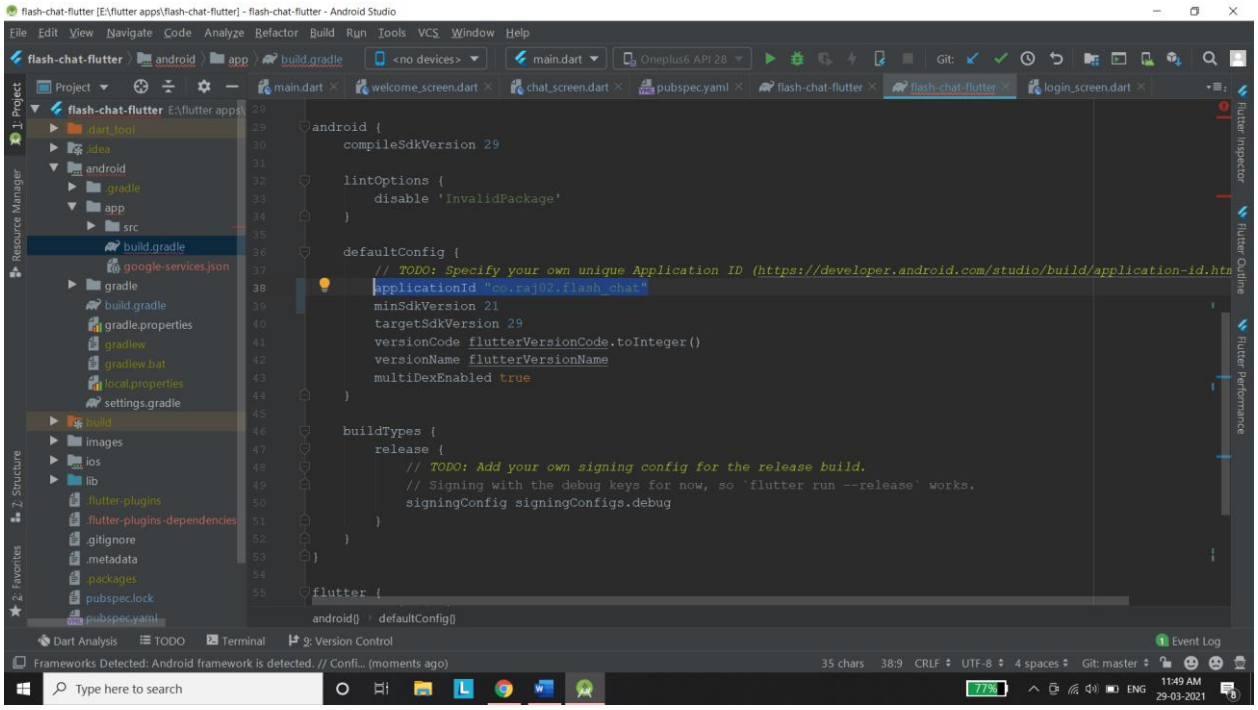
Step 4): If you want google analytics for the project then select the option or else press continue

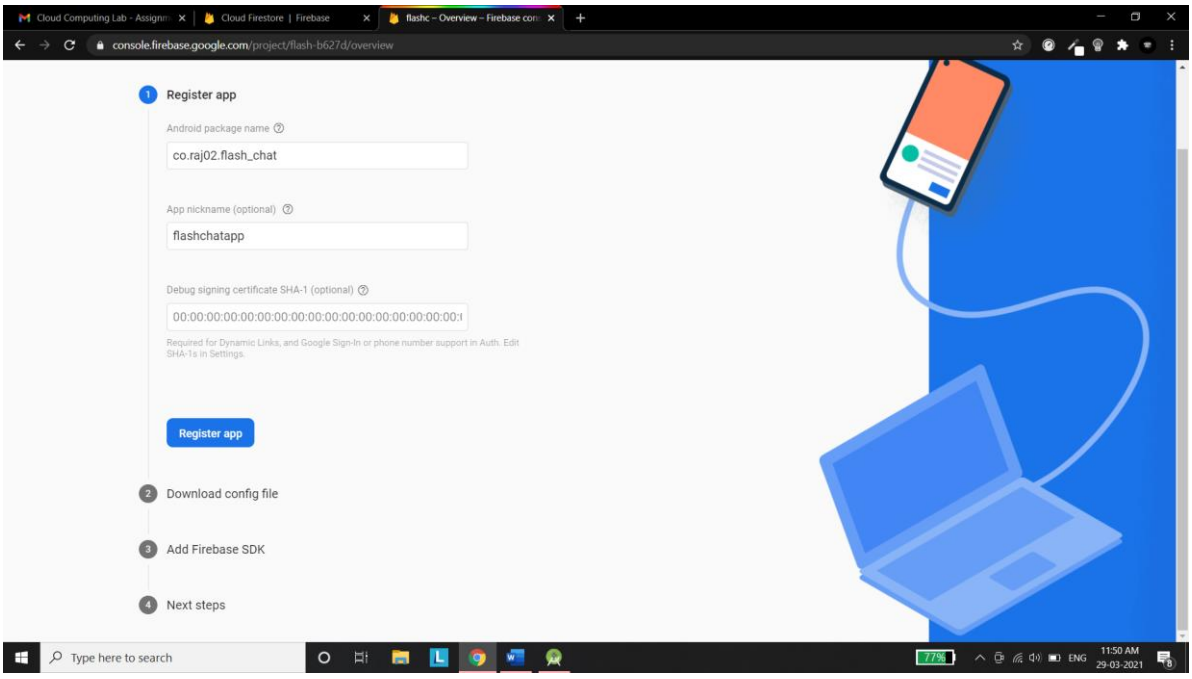


Step 5): After Pressing on continue your project will be created now add an android or an ios or a web based app to the project . In my case I will add an android app.

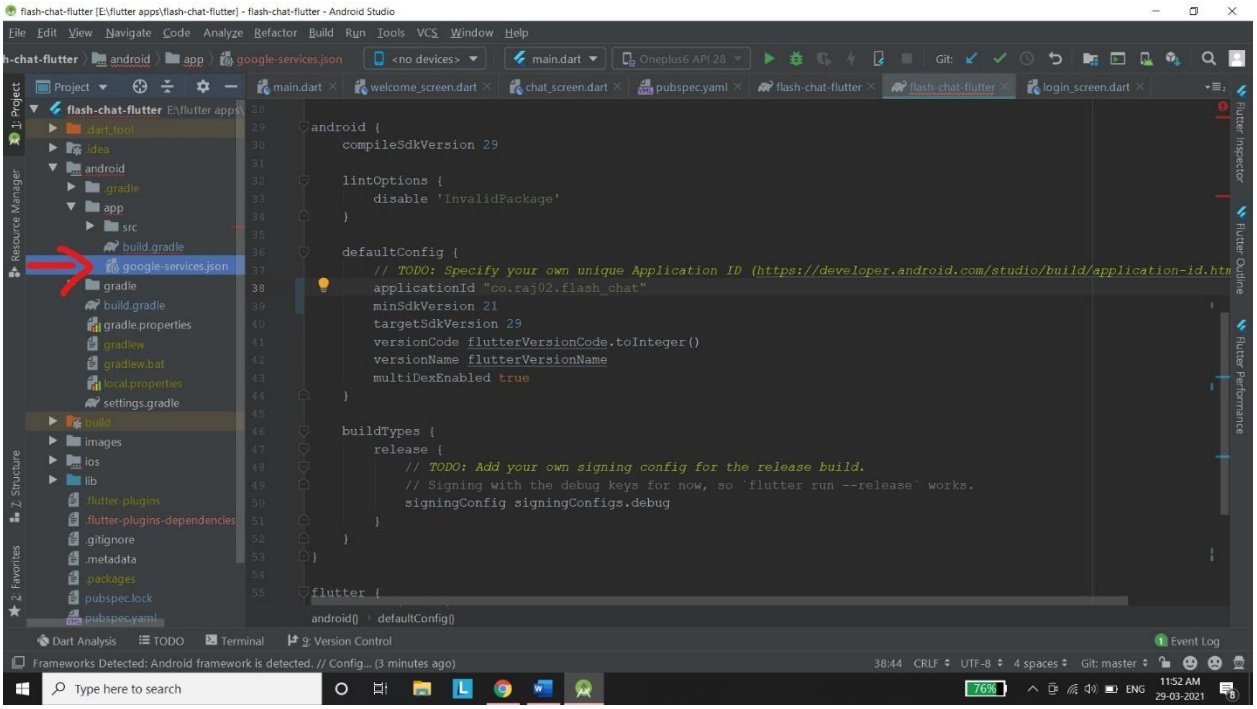


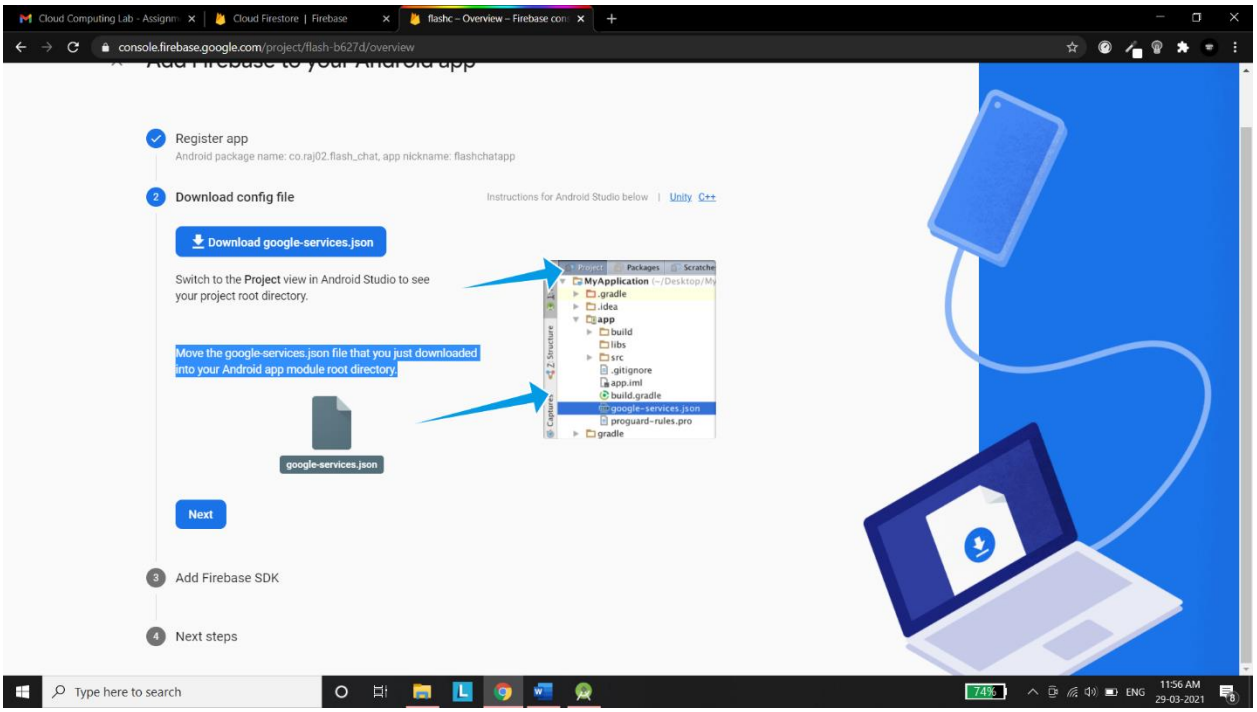
Step 6): Now you need to add an android package name , your package name is generally the application id in your app-level build.gradle file then you need to type the app nickname which is an optional field after performing the above mentioned procedure you need to press on register app button



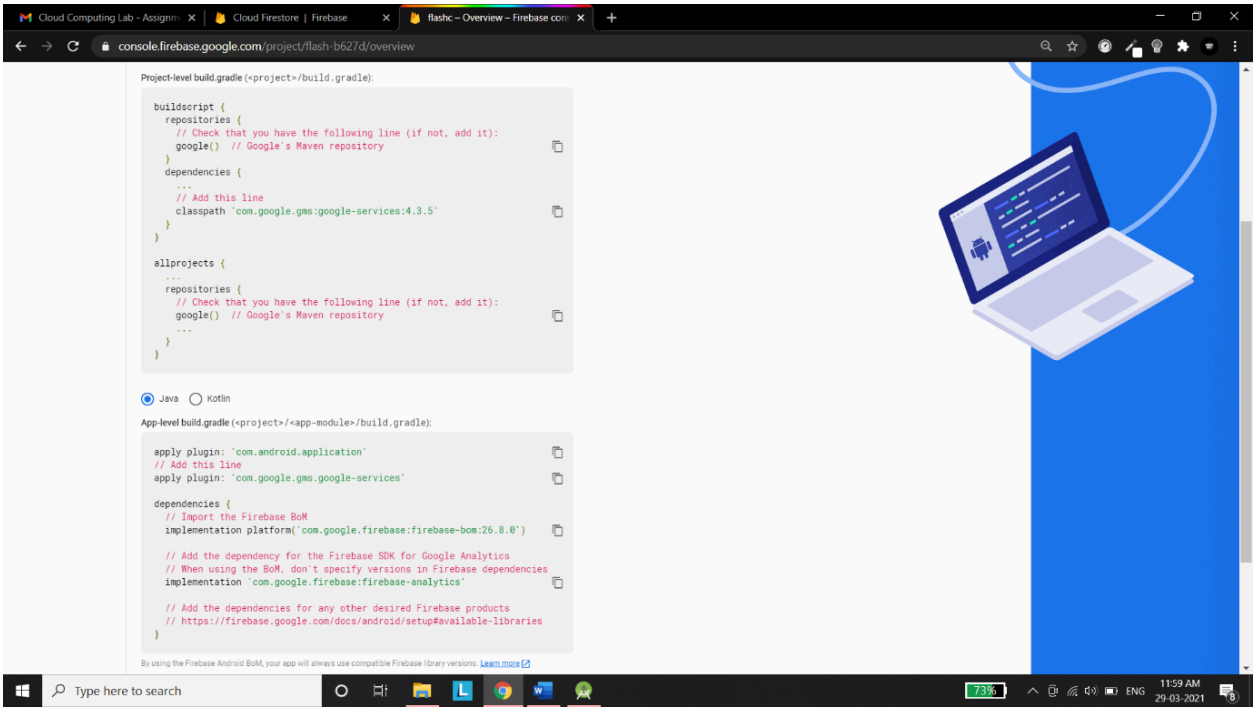


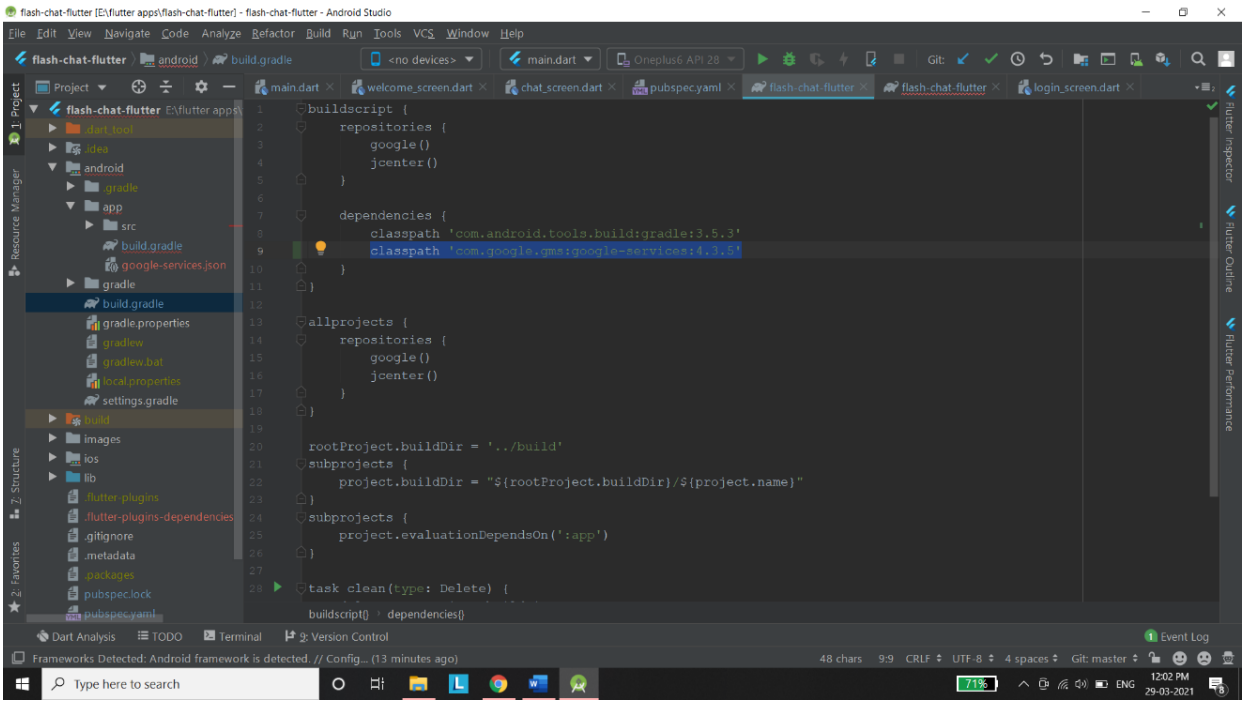
Step 7): Download the config file and move the google-services.json file that you just downloaded into your Android app module root directory and then click on next.

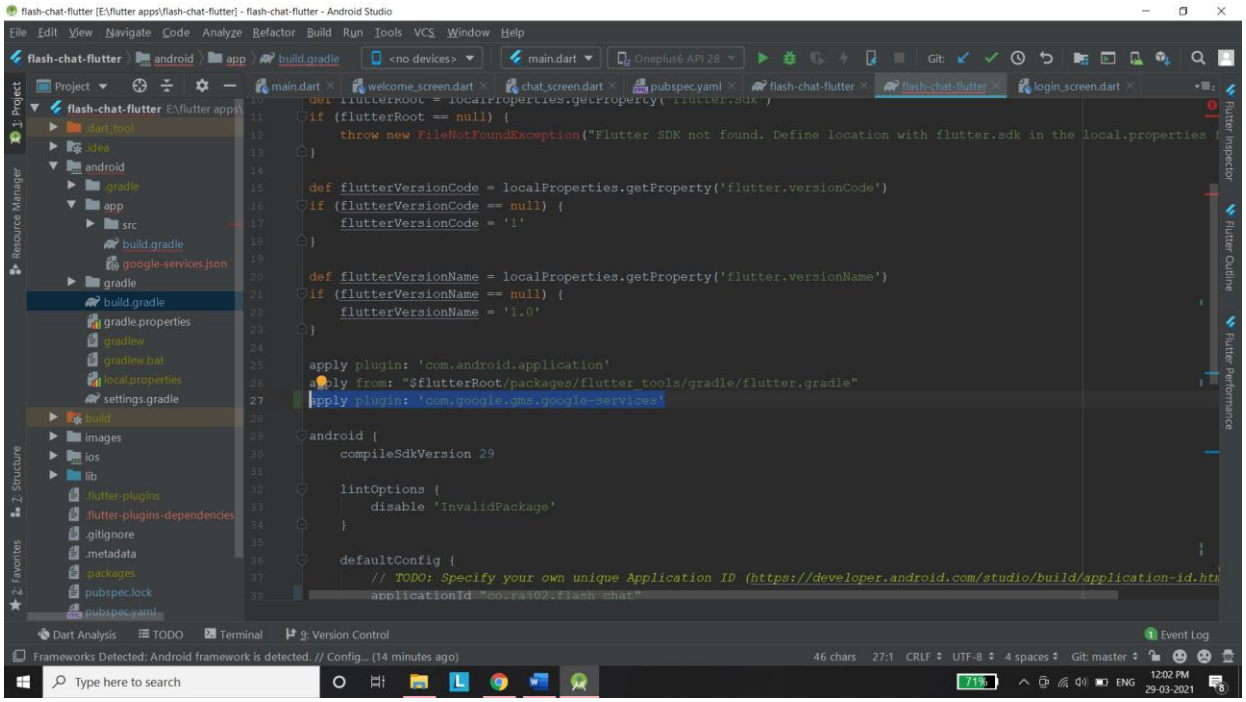




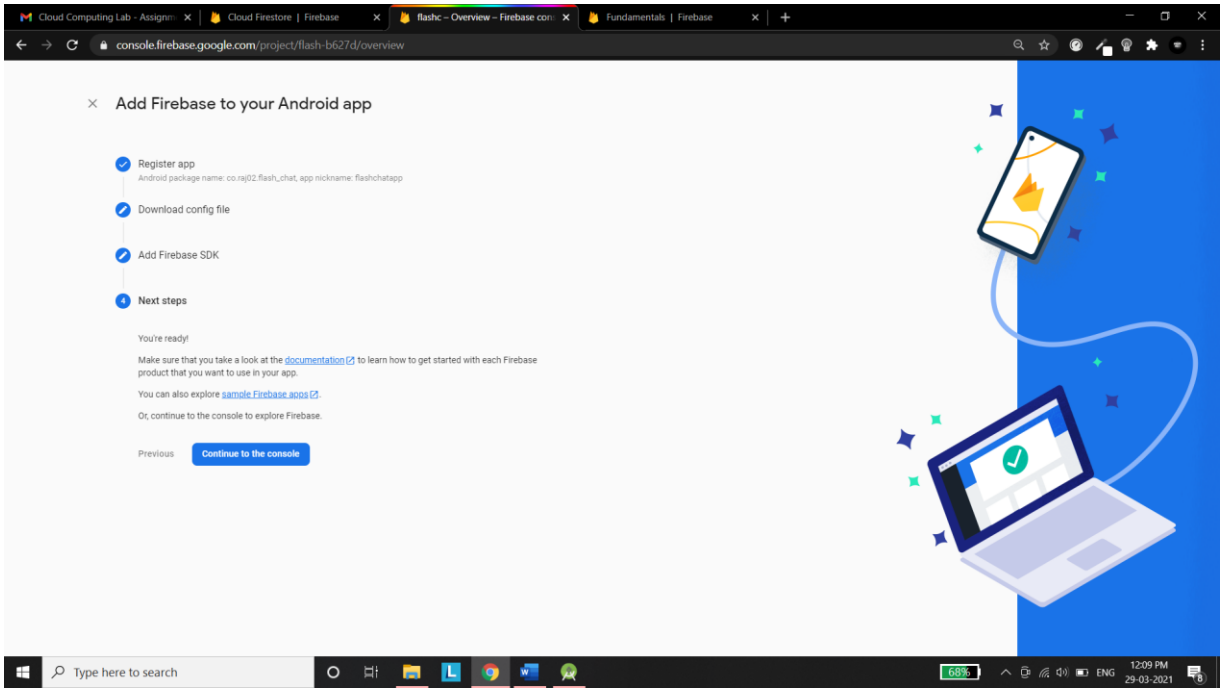
Step 8): Add the following lines of code in your app which are mentioned on the website and then press the next button.


Cloud Computing Lab - Assignm
Cloud Firestore | Firebase
flashc – Overview – Firebase con
console.firebase.google.com/project/flash-b627d/overview
Project-level build.gradle (<project>/build.gradle):
buildscript {
repositories {
// Check that you have the following line (if not, add it):
google() // Google's Maven repository
}
dependencies {
...
// Add this line
classpath 'com.google.gms:google-services:4.3.5'
}
}
allprojects {
...
repositories {
// Check that you have the following line (if not, add it):
google() // Google's Maven repository
...
}
}
Java
Kotlin
App-level build.gradle (<project>/<app-module>/build.gradle):
apply plugin: 'com.android.application'
// Add this line
apply plugin: 'com.google.gms.google-services'
dependencies {
// Import the Firebase BoM
implementation platform('com.google.firebase:firebase-bom:26.8.0')
// Add the dependency for the Firebase SDK for Google Analytics
// When using the BoM, don't specify versions in Firebase dependencies
implementation 'com.google.firebase:firebase-analytics'
// Add the dependencies for any other desired Firebase products
// https://firebase.google.com/docs/android/setup#available-libraries
}
By using the Firebase Android BoM, your app will always use compatible Firebase library versions. Learn more
Type here to search
73%
ENG
11:59 AM
29-03-2021

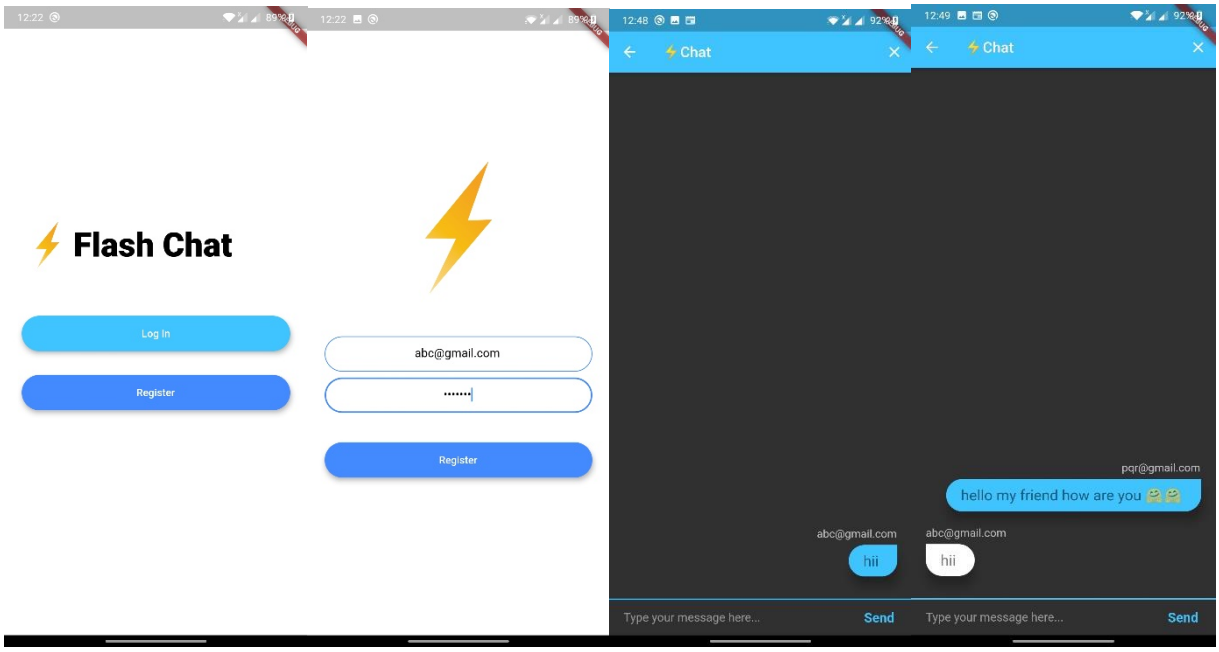

flash-chat-flutter [E:\flutter apps\flash-chat-flutter] - flash-chat-flutter - Android Studio
File Edit View Navigate Code Analyze Refactor Build Run Tools VCS Window Help
flash-chat-flutter android build.gradle
<no devices>
main.dart
Oneplus6 API 28
Git:
Project
main.dart
welcome_screen.dart
chat_screen.dart
pubspec.yaml
flash-chat-flutter
flash-chat-flutter
login_screen.dart
flash-chat-flutter E:\flutter apps\
.dart_tool
.idea
android
.gradle
app
src
build.gradle
google-services.json
gradle
build.gradle
gradle.properties
gradlew
gradlew.bat
local.properties
settings.gradle
build
images
ios
lib
.flutter-plugins
.flutter-plugins-dependencies
.gitignore
.metadata
.packages
pubspec.lock
pubspec.yaml
buildscript {
repositories {
google()
jcenter()
}
dependencies {
classpath 'com.android.tools.build:gradle:3.5.3'
classpath 'com.google.gms:google-services:4.3.5'
}
}
allprojects {
repositories {
google()
jcenter()
}
}
rootProject.buildDir = '../build'
subprojects {
project.buildDir = "${rootProject.buildDir}/${project.name}"
}
subprojects {
project.evaluationDependsOn(':app')
}
task clean(type: Delete) {
buildscript{} dependencies{}
Dart Analysis
TODO
Terminal
9: Version Control
Event Log
Frameworks Detected: Android framework is detected. // Config... (13 minutes ago)
48 chars
9:9
CRLF
UTF-8
4 spaces
Git: master
Type here to search
71%
ENG
12:02 PM
29-03-2021



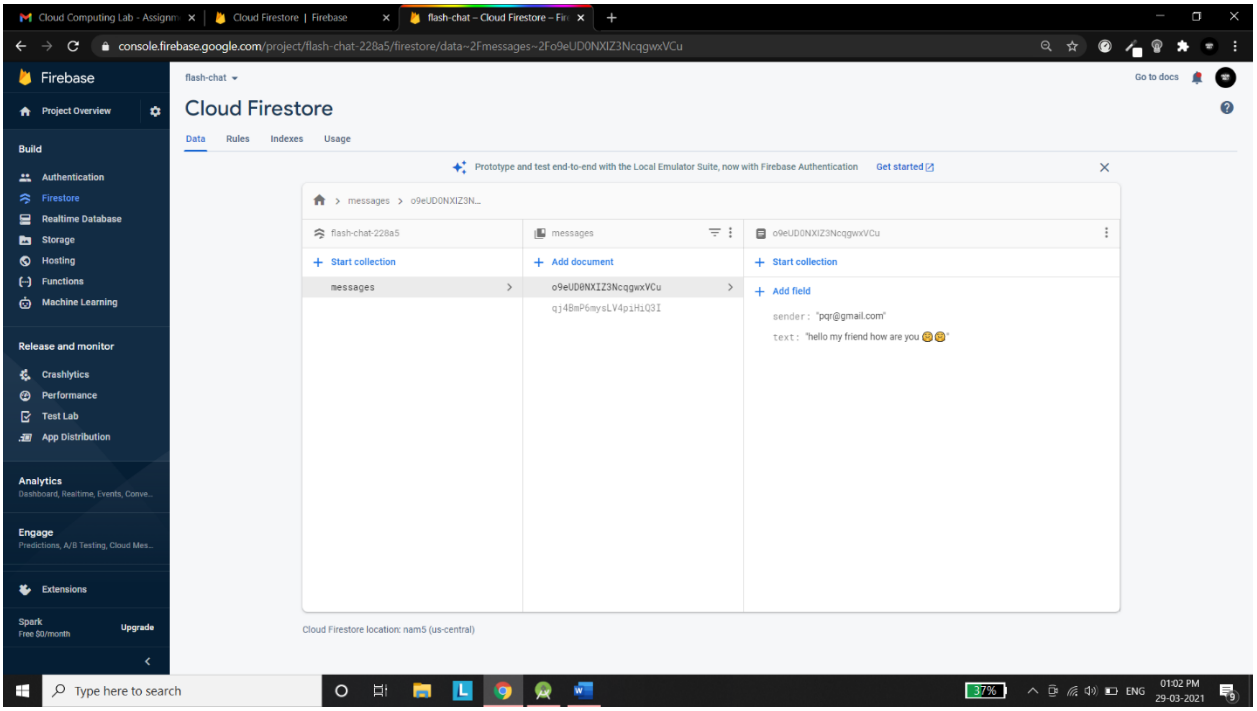
Step 9): You have now integrated firebase in your android app now click on continue to navigate to the console

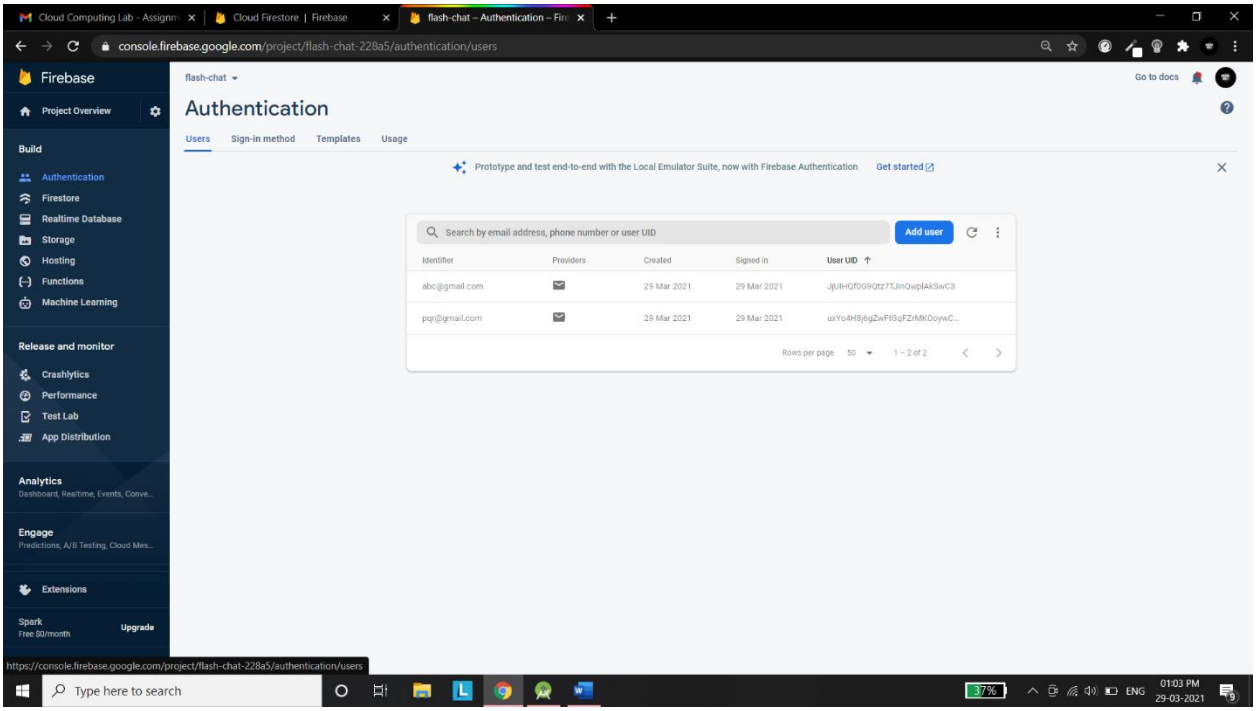


You can now add code in your android app and can access the firebase services. In my app I have used the firestore and the authentication services of the firebase. Here are some screen shots from my android app.



Data Which is stored on the firestore

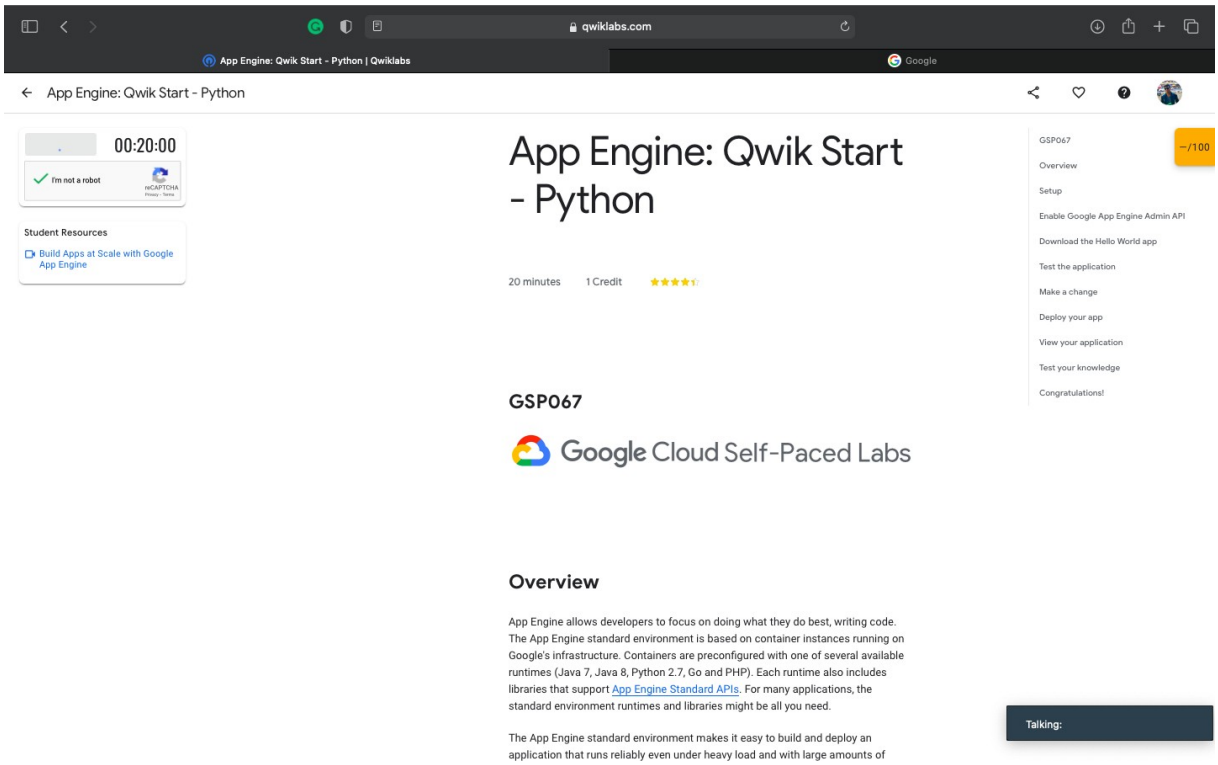




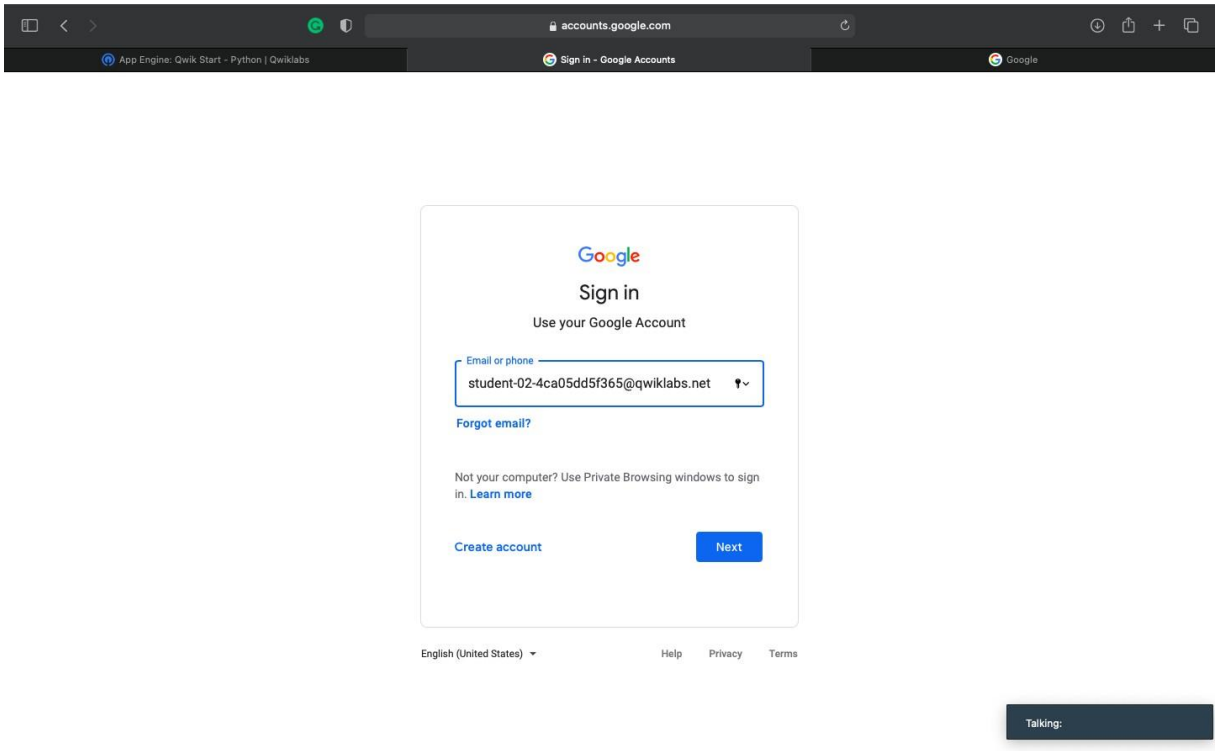
Assignment No 3

Create a sample web-based application using Python and deploy it on AWS.

Click the Start Lab button. If you need to pay for the lab, a pop-up opens for you to select your payment method. On the left is a panel populated with the temporary credentials that you must use for this lab.

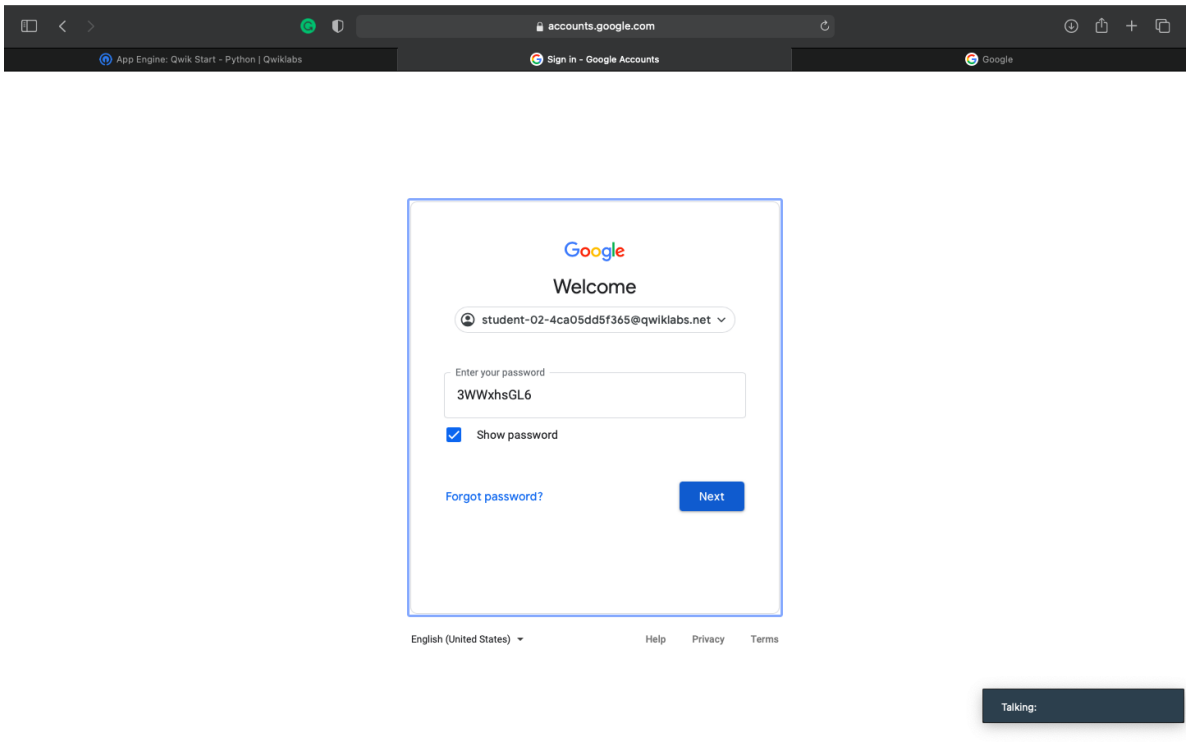


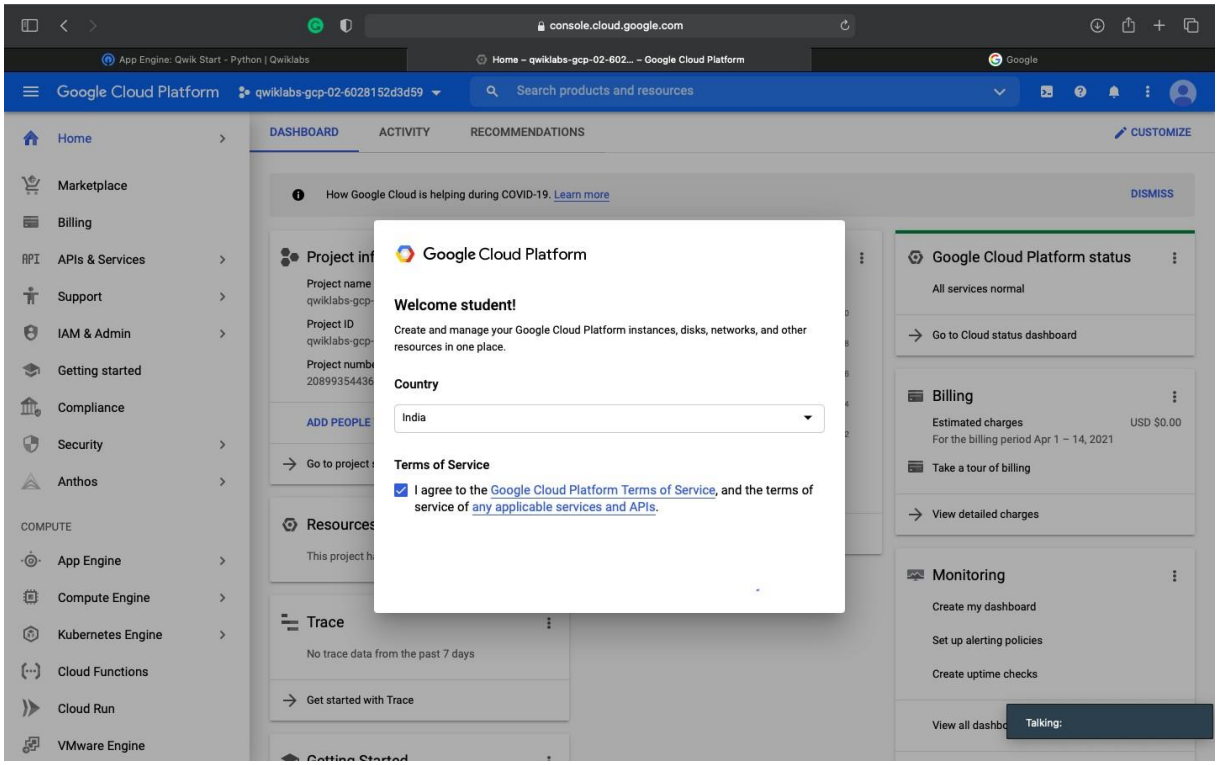
Copy the username, and then click Open Google Console. The lab spins up resources, and then opens another tab that shows the Sign in page.

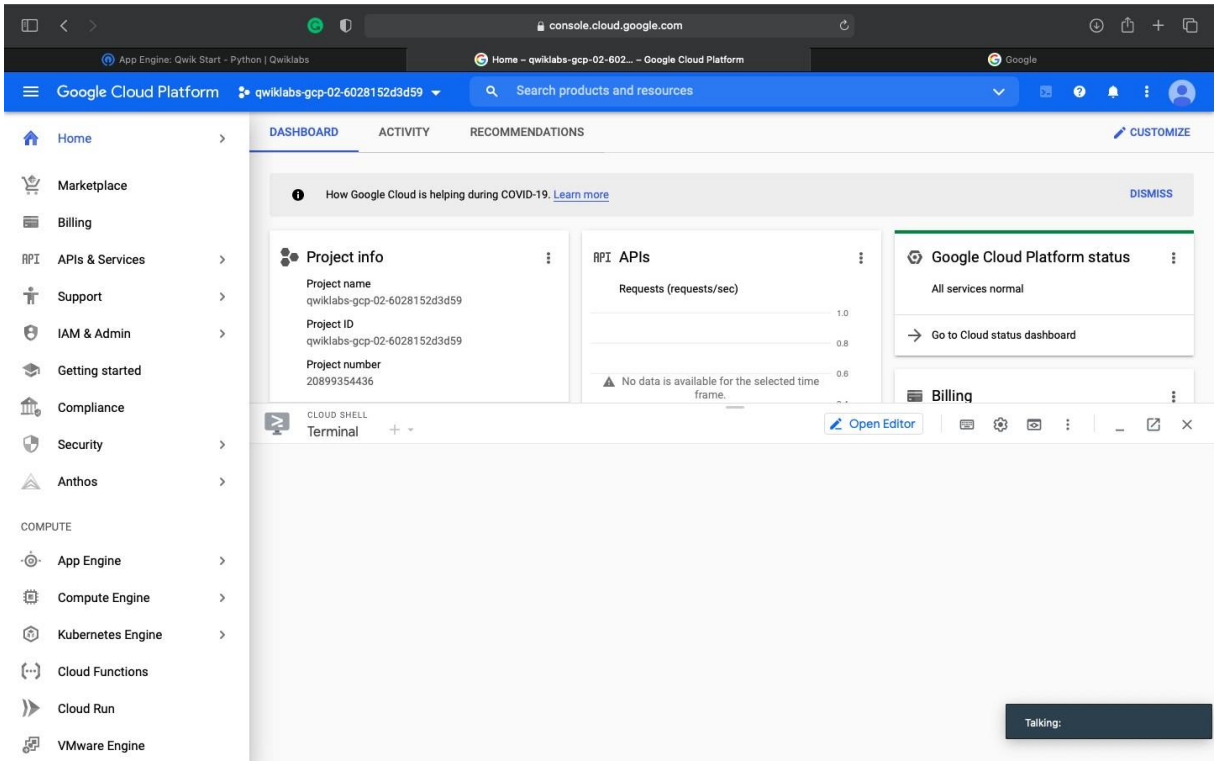


In the **Sign in** page, paste the username that you copied from the Connection Details panel. Then copy and paste the password.

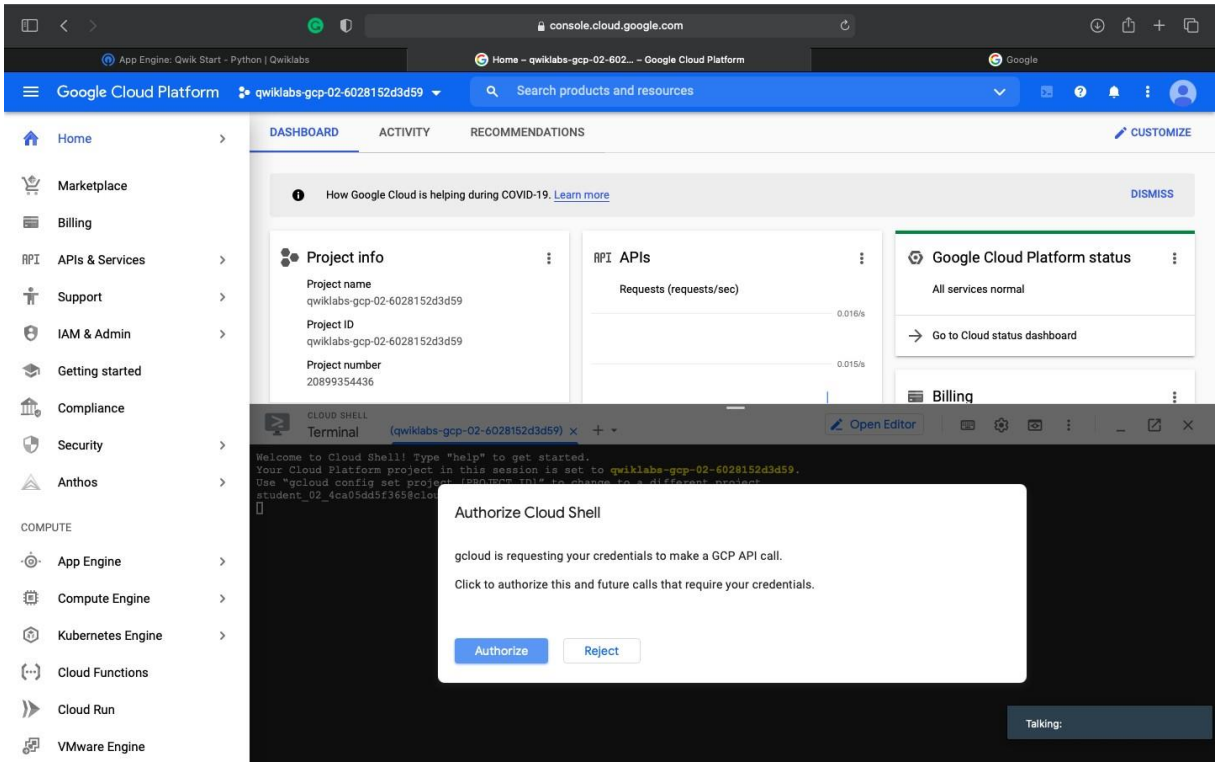
Important: You must use the credentials from the Connection Details panel. Do not use your Qwiklabs credentials. If you have your own Google Cloud account, do not use it for this lab (avoids incurring charges).

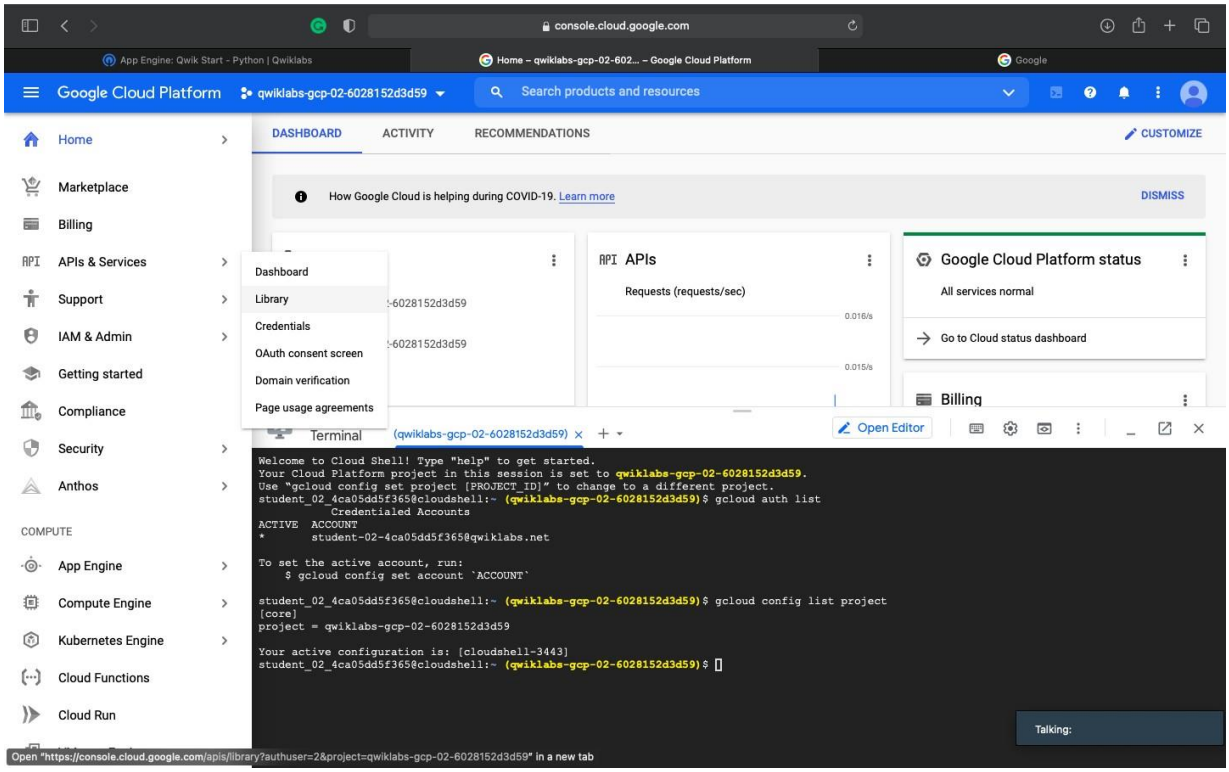



console.cloud.google.com
App Engine: Qwik Start - Python | Qwiklabs
Home – qwiklabs-gcp-02-602... – Google Cloud Platform
Google
Google Cloud Platform
qwiklabs-gcp-02-6028152d3d59
Search products and resources
Home
Marketplace
Billing
APIs & Services
Support
IAM & Admin
Getting started
Compliance
Security
Anthos
COMPUTE
App Engine
Compute Engine
Kubernetes Engine
Cloud Functions
Cloud Run
VMware Engine
DASHBOARD
ACTIVITY
RECOMMENDATIONS
CUSTOMIZE
How Google Cloud is helping during COVID-19. Learn more
DISMISS
Google Cloud Platform
Welcome student!
Create and manage your Google Cloud Platform instances, disks, networks, and other resources in one place.
Country
India
Terms of Service
I agree to the Google Cloud Platform Terms of Service, and the terms of service of any applicable services and APIs.
Google Cloud Platform status
All services normal
Go to Cloud status dashboard
Billing
Estimated charges
USD $0.00
For the billing period Apr 1 – 14, 2021
Take a tour of billing
View detailed charges
Monitoring
Create my dashboard
Set up alerting policies
Create uptime checks
Trace
No trace data from the past 7 days
Get started with Trace
Talking:


console.cloud.google.com
App Engine: Qwik Start - Python | Qwiklabs
Home – qwiklabs-gcp-02-602... – Google Cloud Platform
Google
Google Cloud Platform
qwiklabs-gcp-02-6028152d3d59
Search products and resources
Home
Marketplace
Billing
APIs & Services
Support
IAM & Admin
Getting started
Compliance
Security
Anthos
COMPUTE
App Engine
Compute Engine
Kubernetes Engine
Cloud Functions
Cloud Run
VMware Engine
DASHBOARD
ACTIVITY
RECOMMENDATIONS
CUSTOMIZE
How Google Cloud is helping during COVID-19. Learn more
DISMISS
Project info
Project name
qwiklabs-gcp-02-6028152d3d59
Project ID
qwiklabs-gcp-02-6028152d3d59
Project number
20899354436
APIs
Requests (requests/sec)
1.0
0.8
0.6
No data is available for the selected time frame.
Google Cloud Platform status
All services normal
Go to Cloud status dashboard
Billing
CLOUD SHELL
Terminal
Open Editor
Talking:

Now Activate Cloud Shell,

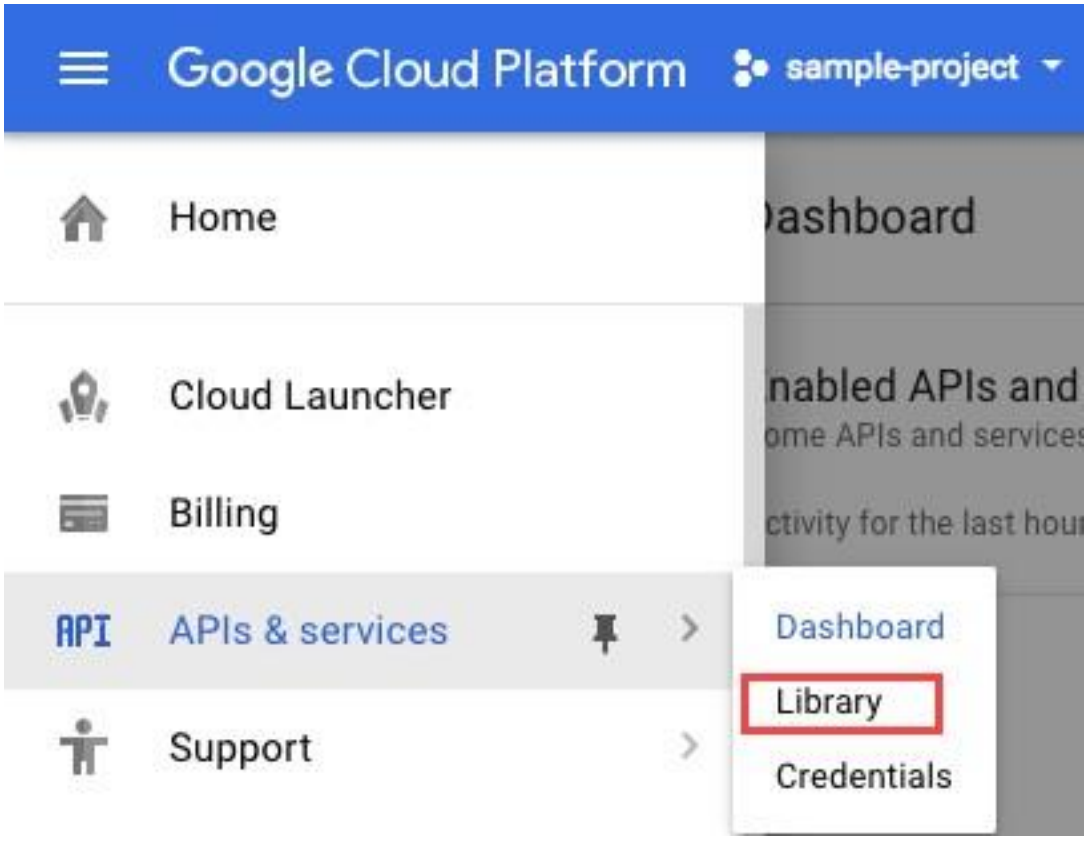




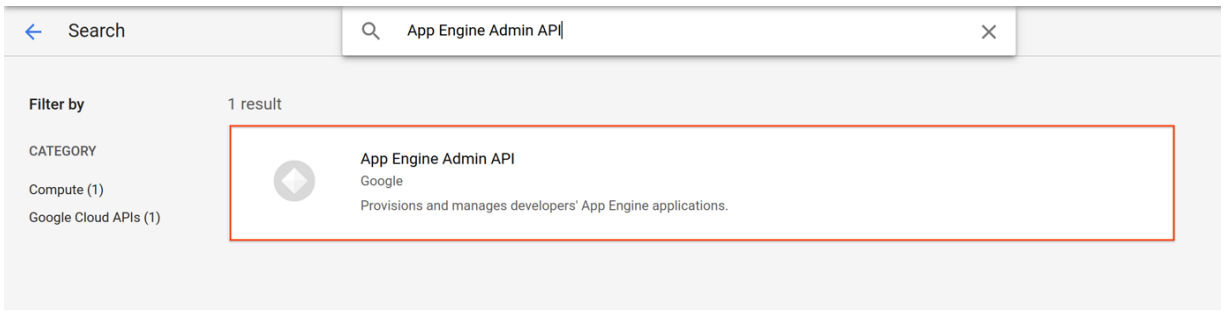
Enable Google App Engine Admin API

The App Engine Admin API enables developers to provision and manage their App Engine Applications.

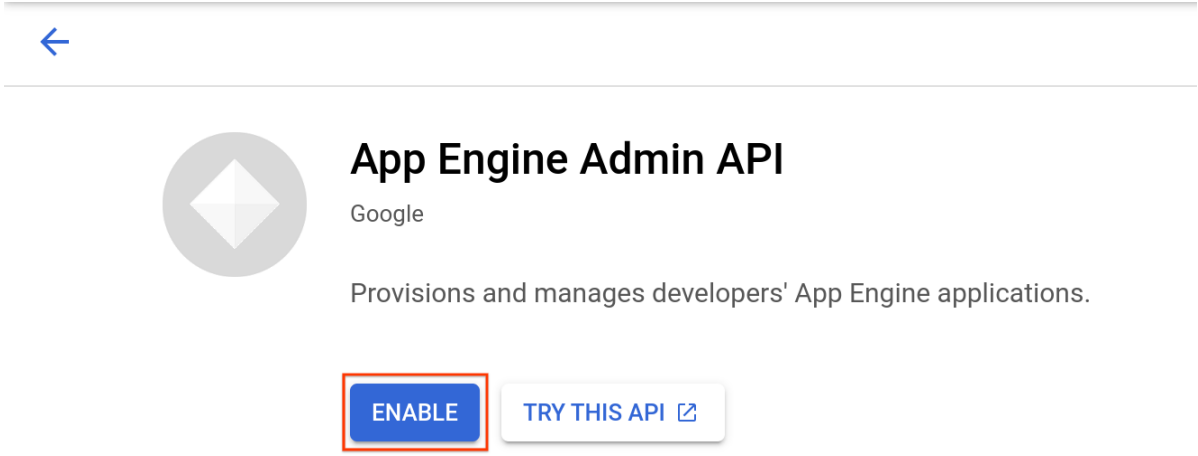
1. In the left menu click **APIs & Services** > **Library**.

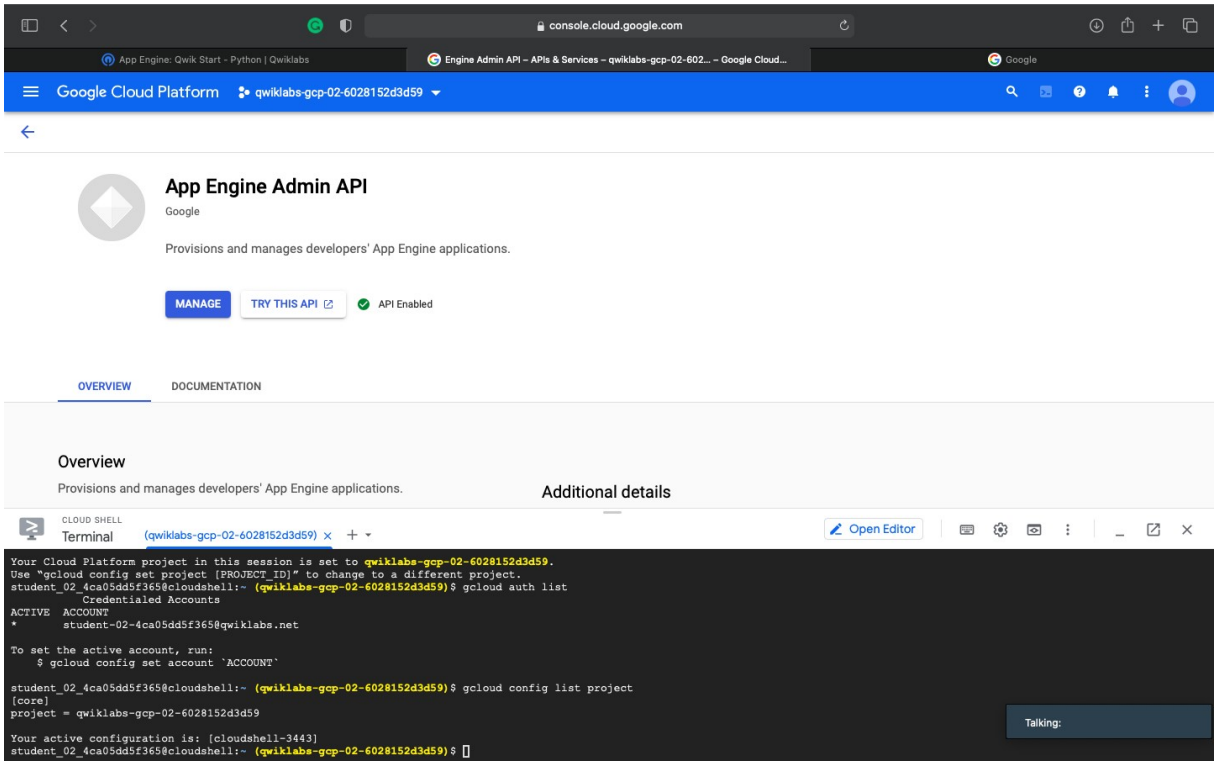


2. Type "App Engine Admin API" in search box.
3. Click **App Engine Admin API**.



4. Click **Enable**. If there is no prompt to enable the API, then it is already enabled and no action is needed.



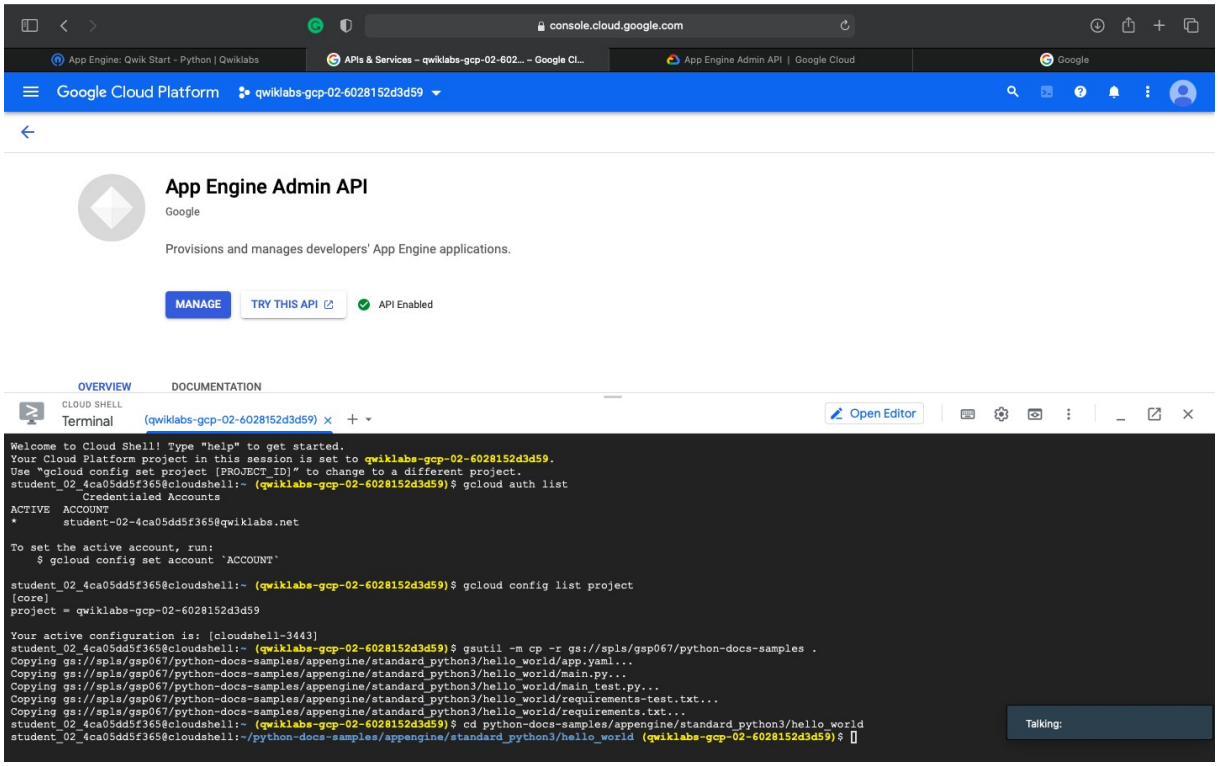


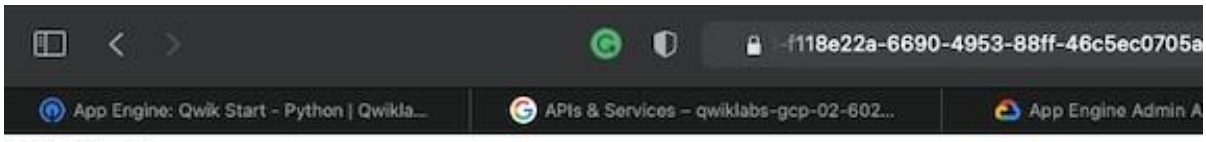
Download the Hello World app

There is a simple Hello World app for Python you can use to quickly get a feel for deploying an app to Google Cloud. Follow these steps to download Hello World to your Google Cloud instance.

1. Enter the following command to copy the Hello World sample app repository to your Google Cloud instance:

2. Go to the directory that contains the sample code:





Hello World!

\

Make a change

You can leave the development server running while you develop your application. The development server watches for changes in your source files and reloads them if necessary.

Let's try it. Leave the development server running. We'll open another command line window, then edit main.py to change "Hello World!" to "Hello, Cruel World!".

1. Click the (+) next to your Cloud Shell tab to open a new command line session.

2. Enter this command to go to the directory that contains the sample code.

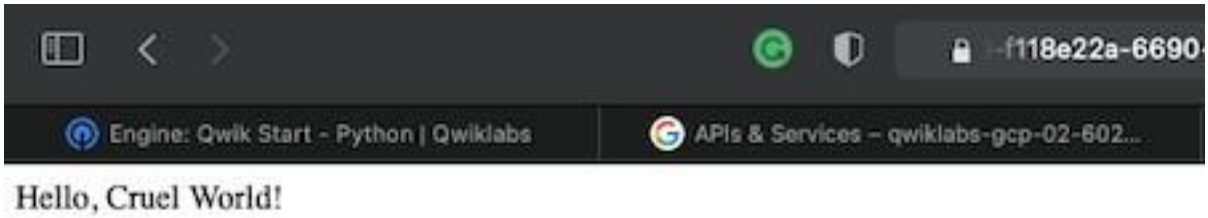
```
cd python-docs-samples/appengine/standard_python3/hello_world
```

3. Enter the following to open main.py in nano to edit the content.

```
nano main.py
```

4. Change "Hello World!" to "Hello, Cruel World!". Exit and save the file.

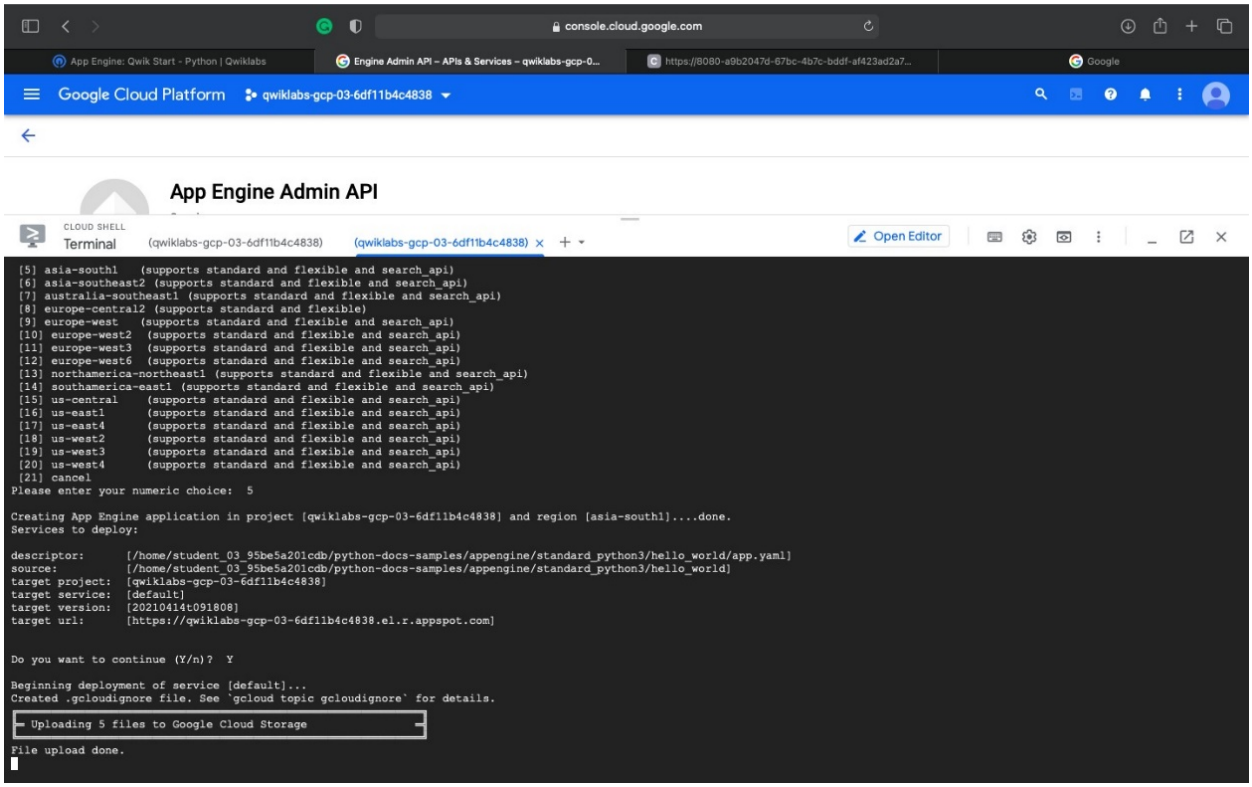
5. Reload the Hello World! Browser or click the **Web Preview** > **Preview on port 8080**to see the results.



Deploy your app

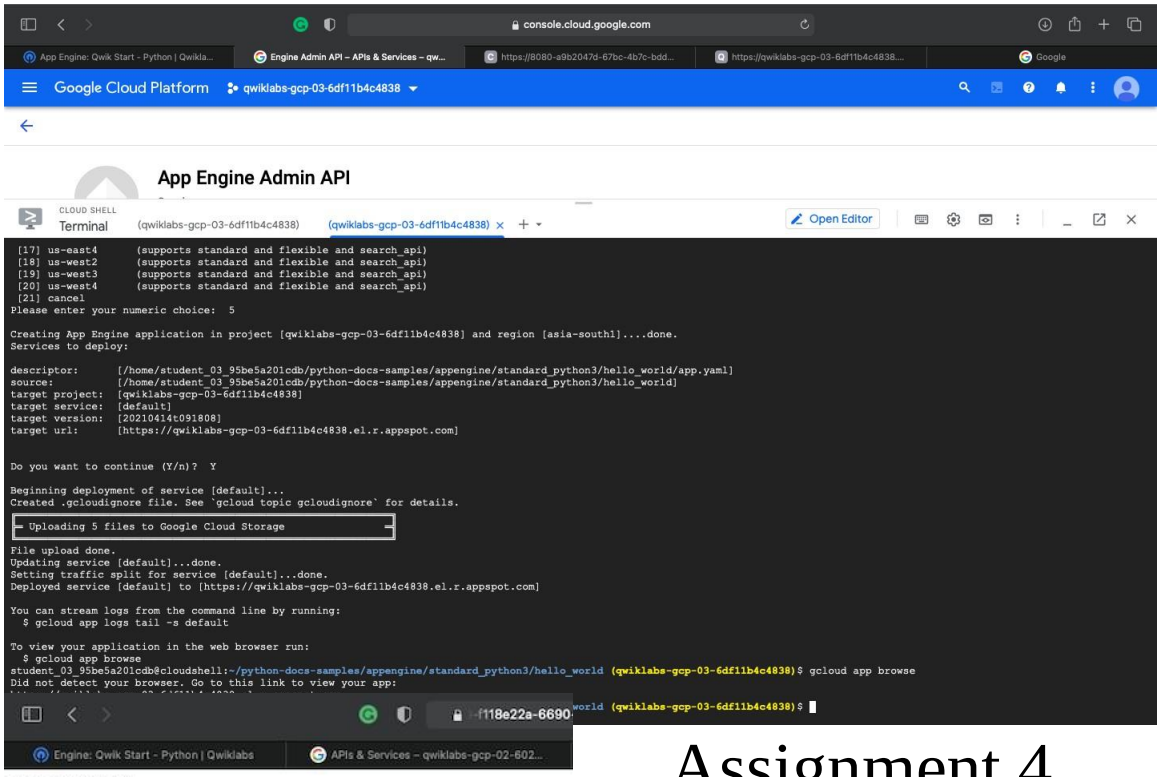
To deploy your app to App Engine, run the following command from within the root directory of your application where the app.yaml file is located:

gcloud app deploy



View your application

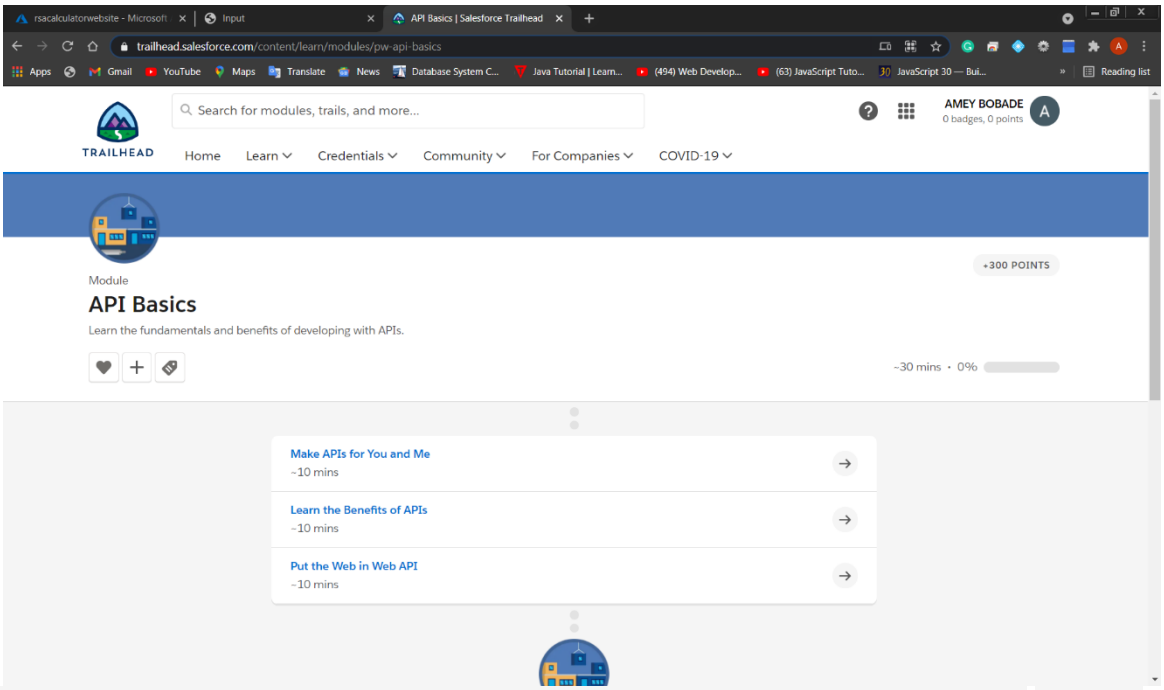
To launch your browser enter the following command, then click on the link it provides.



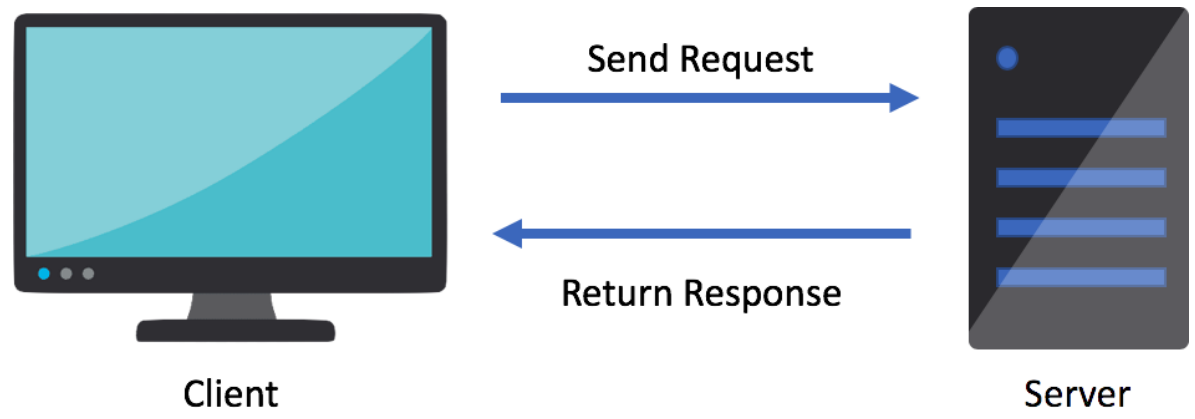
Assignment 4

Salesforce

MODULE : API Basics



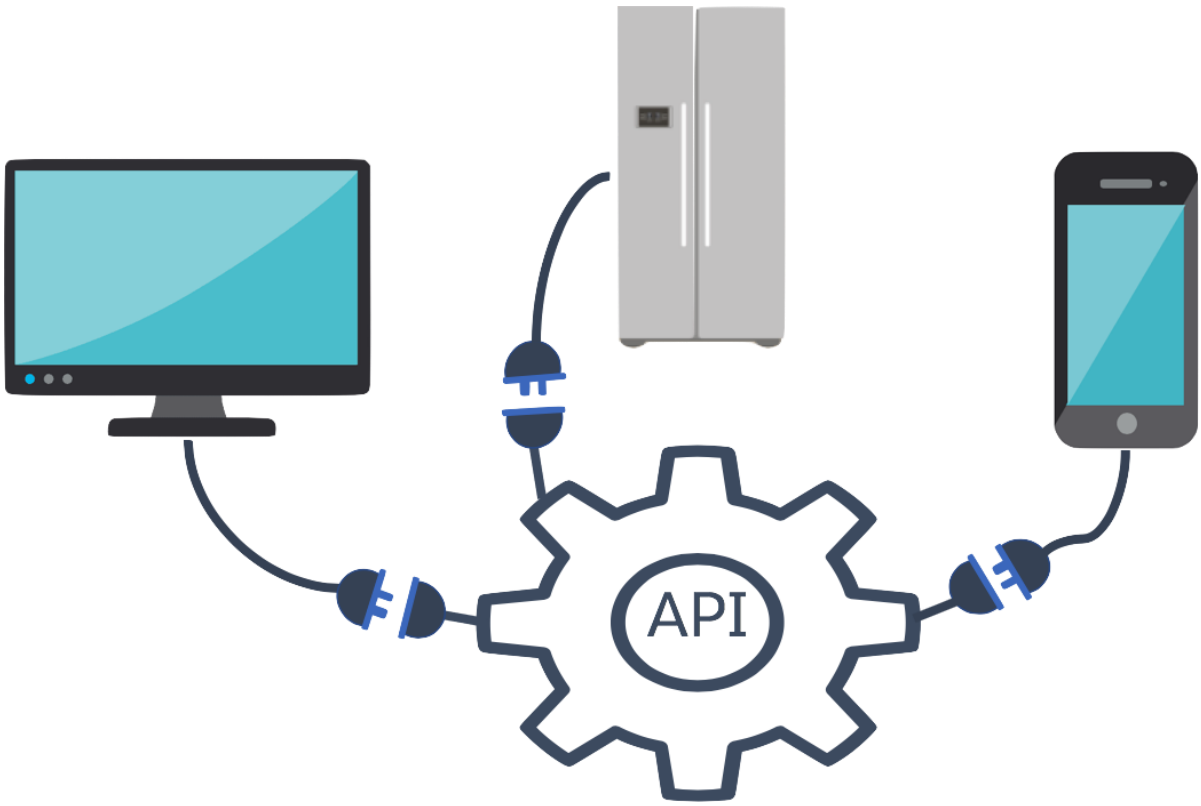
An API is equivalent to a user interface, except it's designed for software instead of humans. This is why APIs are often described in the media as technology that allows applications to talk to one another.



The client sends a request for specific information or functionality to another system. That system returns the data or functionality in a response. To send or receive data, there is an expectation that it will be in a specific format that both sides can understand. That format is often very sensitive to the context(s) it serves. Let's take a closer look.

APIs: The Electric Current Between Software

So what, if anything, do electric wall sockets have to do with APIs?



- The electricity coming from the wall is a service. It can stop and start at any time.
- The treadmill plugged into the wall uses the electricity to run.
- Since the treadmill does not have its own source of power, the treadmill is outsourcing the power it needs to a service provider, for example, to one that uses wind turbines or solar power.

While electrical sockets differ depending on where you are in the world, they have

predictable, standard patterns of openings; and the electrical plugs attached to devices like treadmills are designed to fit those standard patterns.

All these specifications essentially set expectations on behalf of any device that wants to use the service. The plugs and power supplies conform to the standard patterns and specifications (like 120 volts) for the service. The same goes for APIs.

The Benefits of Using APIs :

APIs help open up a plethora of opportunities. Here are ways that software, customers, citizen integrators, developers, and their teams can benefit from using APIs.

Outsourcing

In the name of repeatability, any compatible device (in this case, the gym equipment) can easily outsource its electrical requirements to a service, and those devices can expect to get the same results.

Similarly, APIs allow you to outsource key data and functionality through a predictable standard interface. Focus on making great applications, services, and customer experiences, not on figuring out how you're going to get common, yet nuanced information.

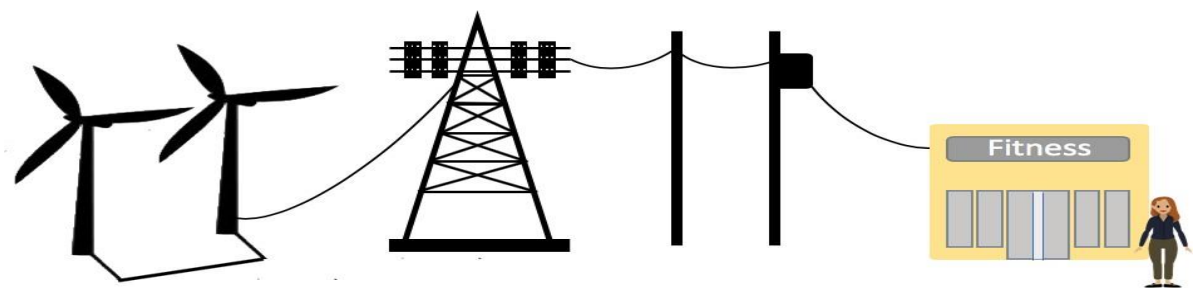
Increased Mobility

Consuming-devices are easily moved from one socket to another. For example, with no plug, matching socket or specifications, the gym owner may have to hardwire equipment into the walls of the building

Abstraction

When taking a look at the fitness club, the electrical socket is a layer of abstraction to the underlying service, or electricity. What is abstraction you ask? It is a way of hiding the working details of another system.

As long as the service delivers 120 volts of AC power to the wall socket in the standard way, the service provider is free to change anything and everything from just behind the socket all the way to the source of power. Any changes are opaque to consuming devices.



Increased Developer Productivity

When programmers write code, they rarely start from scratch. APIs are designed to take an existing code base and use it whenever, wherever instead of attempting to re-create those features.

While reusing existing code limits differentiation between applications, a reference to the API (more commonly referred to as "calling" an API) can supply the program with the expected data or functionality.

Using HTTP Protocols to Access Data

While there are no rules or laws deciding exactly how developers must connect their applications to an API, several standards have emerged.
For example, when applications connect to APIs from across the Internet, the majority of API providers make these connections available over HTTP, or hypertext transfer protocol, otherwise known as the World Wide Web.

HTTP Verbs	Descriptions
POST	Submit requested data to a server for processing
GET	Retrieve requested data from a server
PUT	Update and replace existing data with new data being sent in the request
DELETE	Remove the requested data from the server

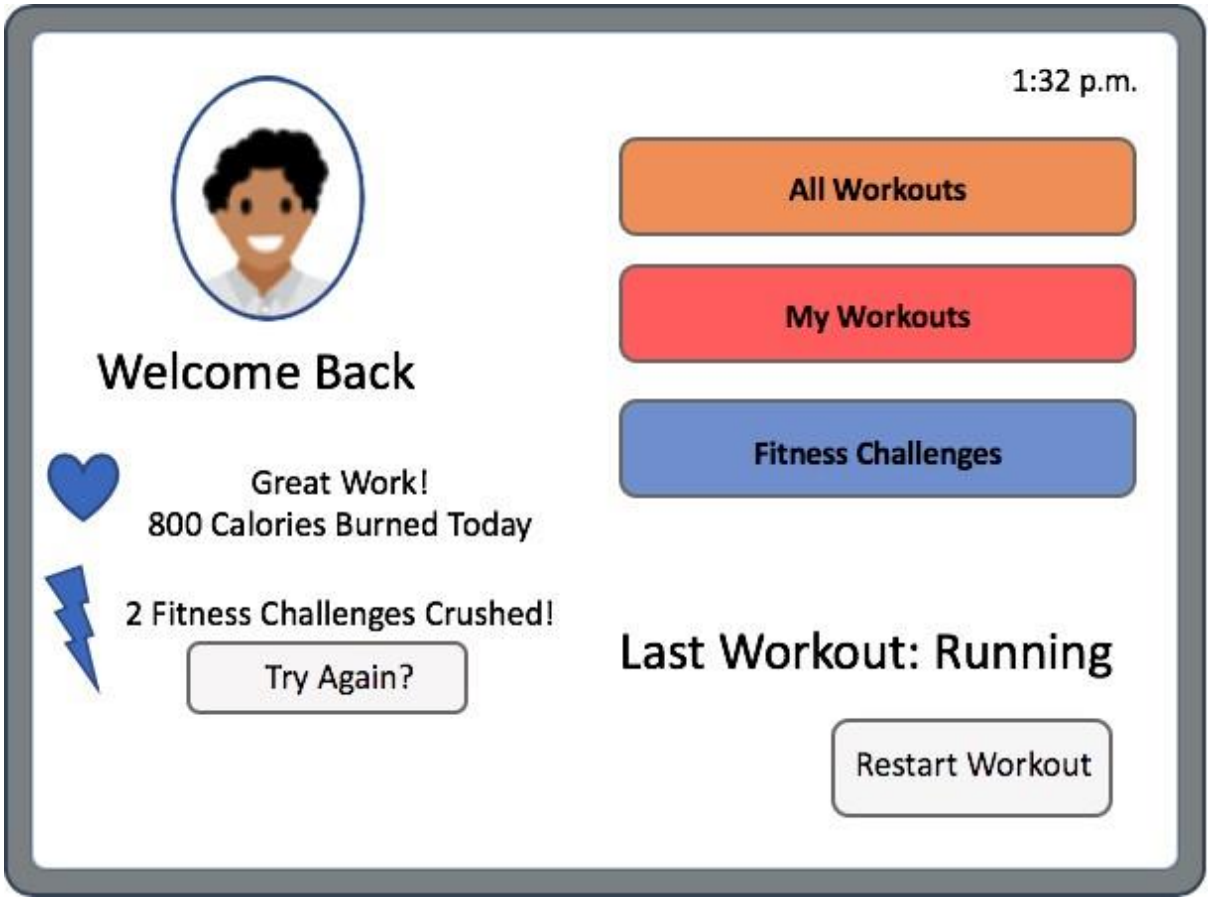
API


```
{
    "activities":[
        {
            "activityId":51007,
            "activityParentId":90019,
            "calories":800,
            "description":"7mph",
            "distance":3.04,
            "name":"Treadmill, 0% Incline",
            "startTime":"00:25",
            "steps":6783
        }
    ],

    “fitchallenge”:[
        {
            "caloriesOut":2826,
            "distance":8.05,
            "floors":150,
            "steps":10000
        },
        {
            "caloriesOut":2021,
            "distance":6.40,
            "floors":80,
            "steps":8000
        },
    ]
}
```

Of course, the above response isn’t very intuitive and would never be presented to the treadmill user. It is formatted according to another standard called JSON (JavaScript Object Notation) that is often used with HTTP.

Treadmill Interface

Once the treadmill receives a response, the user can see total calories burned, how they're tracking against the most recent fitness challenge, and more.



Thanks to the thousands of APIs that software developers and non-programmers equipped with citizen integration tools (citizen integrators) can reach over the Internet (more than 23,000 by last count according to ProgrammableWeb.com), the Web has turned into a programmable platform that’s equally, if not more, powerful than programmable platforms including Windows, Mac, and Linux.

2. Put the Web in Web API

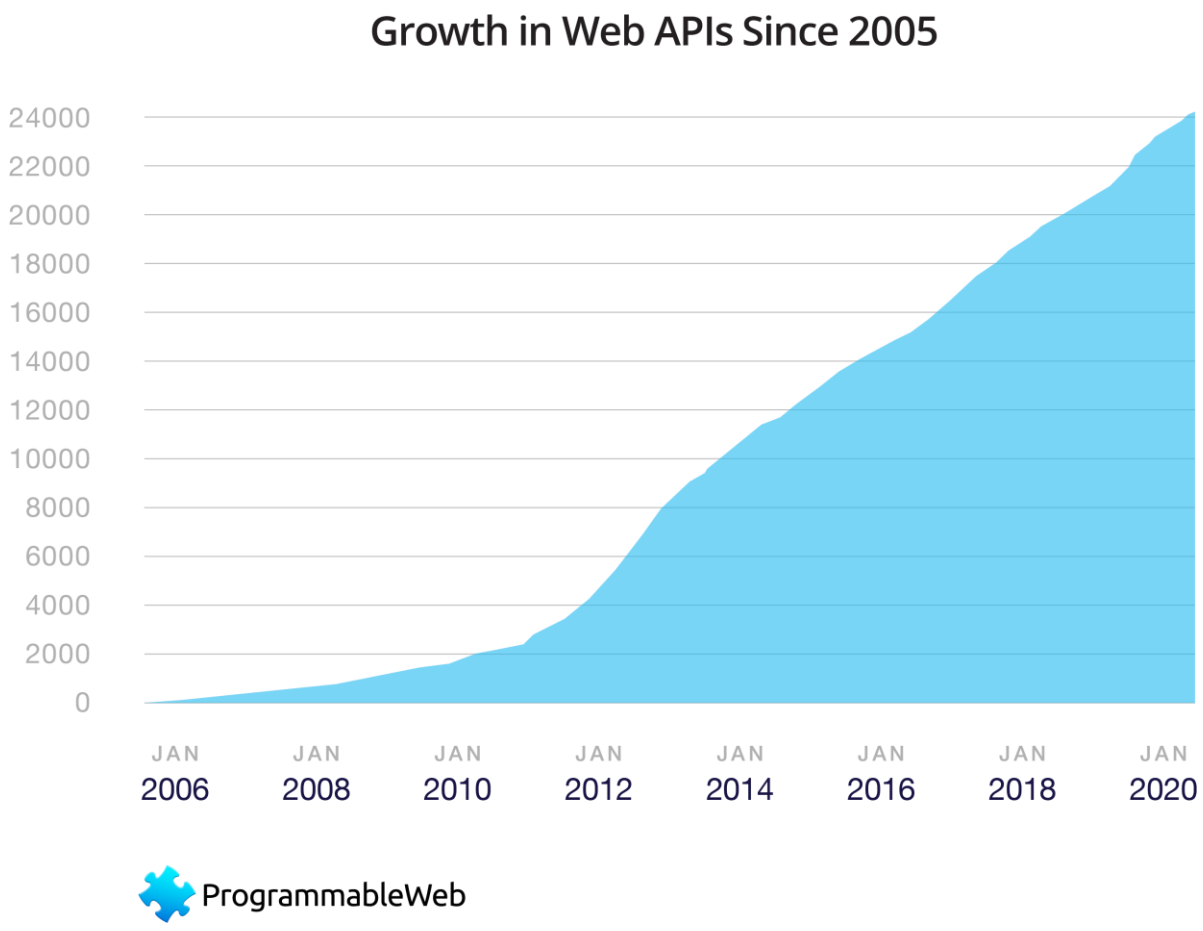
Networkable APIs Are Game-Changers :

APIs aren’t limited to what can be found in the same local network. Developers and citizen integrators can also consume APIs offered by remote systems and devices.

The number and types of devices that can be plugged into electrical sockets are limited only by the imaginations of inventors and the capacity of the utility.

API Economy

Depending on the volume of calls or some other way of breaking down different tiers of service, a provider like Google might charge the application developer a fee for using the API. This gives rise to the idea of an API economy.



API Growth Then and Now

You may be thinking networkable APIs are the greatest thing since sliced bread. You also may be wondering, if they're so great, why didn't the tech industry come up with them earlier? As it turns out, it did.

Back in the days when Unix first came out, it was not uncommon for programmers to remotely invoke business logic from another machine across a network through a technology called RPC, or remote procedure call.

The (Possible) Future of Integration

So if history is any kind of indicator, the way we integrate between systems may be due for a change. There are now two relatively new API-like technologies that part ways with the currently favored web approach. One comes from Facebook, called GraphQL, and the other is from Google, called gRPC.

Assignment No 5

Title: create a VM using Docker and write your own docker file and publish it.

References : https://docs.docker.com/get-started/overview/

Note:- This assignemnt has 2 parts:

1) Install and use docker

2) Create your own image and shre it or publish it. (Image containing base : ubuntu and softwre isntalled: Apache web server)

- **What is Docker:**

Docker is a set of platform as a service (PaaS) products that use OS-level virtualization to deliver software in packages called containers. Containers are isolated from one another and bundle their own software, libraries and configuration files; they can communicate with each other through well-defined channels. All containers are run by a single operating system kernel and therefore use fewer resources than virtual machines.

- **What is the use of Docker:**

Fast, consistent delivery of your applications

Docker streamlines the development lifecycle by allowing developers to work in standardized environments using local containers which provide your applications and services. Containers are great for continuous integration and continuous delivery (CI/CD) workflows.

Consider the following example scenario:

- Your developers write code locally and share their work with their colleagues using Docker containers.
- They use Docker to push their applications into a test environment and execute automated and manual tests.
- When developers find bugs, they can fix them in the development environment and redeploy them to the test environment for testing and validation.
- When testing is complete, getting the fix to the customer is as simple as pushing the updated image to the production environment.

Responsive deployment and scaling

Docker's container-based platform allows for highly portable workloads. Docker containers can run on a developer's local laptop, on physical or virtual machines in a data center, on cloud providers, or in a mixture of environments.

Docker's portability and lightweight nature also make it easy to dynamically manage workloads, scaling up or tearing down applications and services as business needs dictate, in near real time.

Running more workloads on the same hardware

Docker is lightweight and fast. It provides a viable, cost-effective alternative to hypervisor-based virtual machines, so you can use more of your compute capacity to achieve your business goals. Docker is perfect for high density environments and for small and medium deployments where you need to do more with fewer resources.

- **Docker Architecture / Docker Engine**

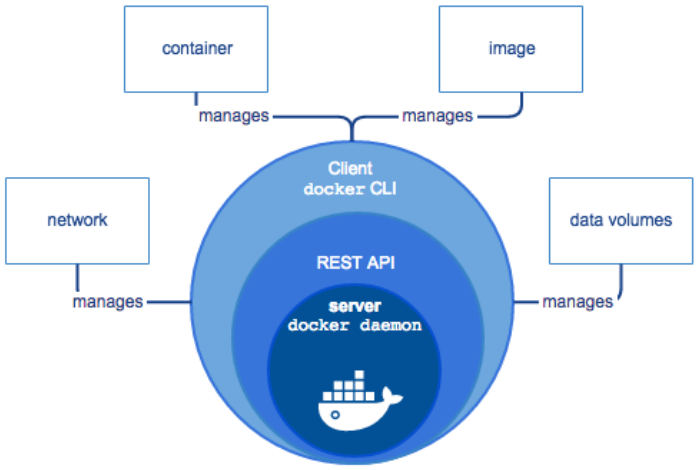


Fig: Docker Engine

Docker Engine is a client-server application with these major components:

- A server which is a type of long-running program called a daemon process (the dockerd command).
- A REST API which specifies interfaces that programs can use to talk to the daemon and instruct it what to do.
- A command line interface (CLI) client (the docker command).

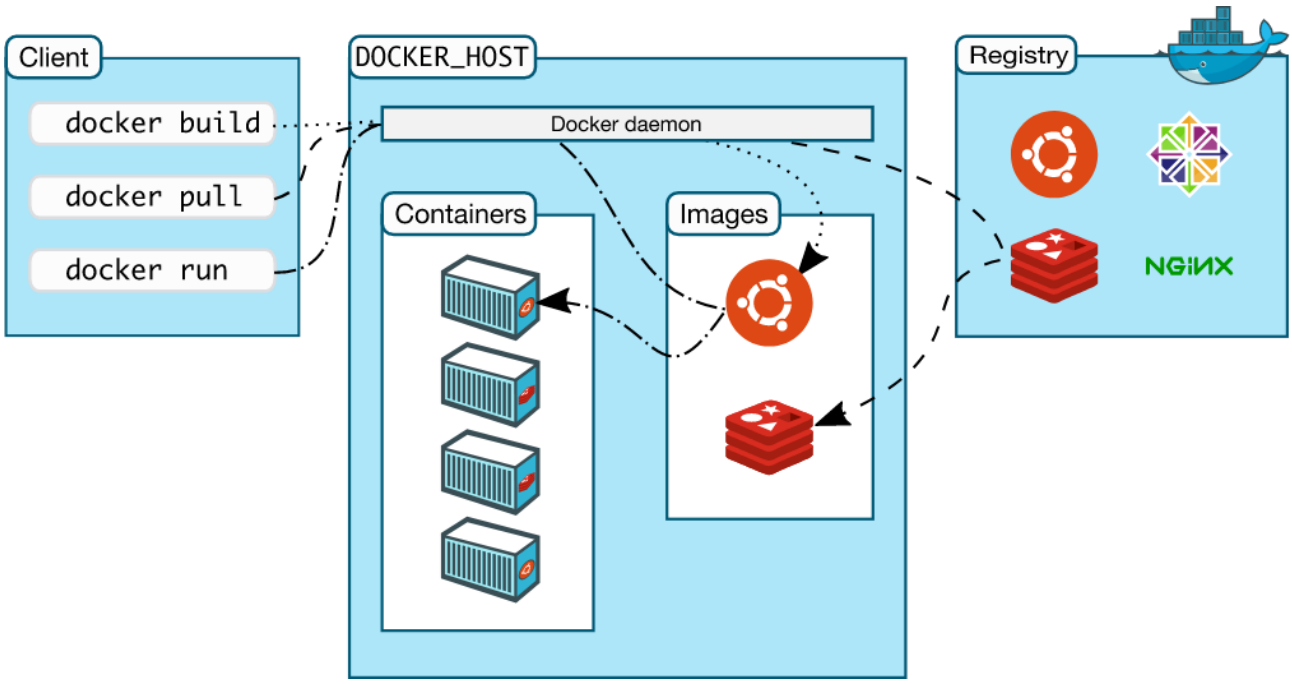


Fig: Docker Architecture

The Docker daemon

The Docker daemon (dockerd) listens for Docker API requests and manages Docker objects such as images, containers, networks, and volumes. A daemon can also communicate with other daemons to manage Docker services.

The Docker client

The Docker client (docker) is the primary way that many Docker users interact with Docker. When you use commands such as docker run, the client sends these commands to dockerd, which carries them out. The docker command uses the Docker API. The Docker client can communicate with more than one daemon.

Docker registries

A Docker *registry* stores Docker images. Docker Hub is a public registry that anyone can use, and Docker is configured to look for images on Docker Hub by default. You can even run your own private registry.

When you use the docker pull or docker run commands, the required images are pulled from your configured registry. When you use the docker push command, your image is pushed to your configured registry.

Docker objects

When you use Docker, you are creating and using images, containers, networks, volumes,

plugins, and other objects. This section is a brief overview of some of those objects.

Images

An *image* is a read-only template with instructions for creating a Docker container. Often, an image is *based on* another image, with some additional customization. For example, you may build an image which is based on the ubuntu image, but installs the Apache web server and your application, as well as the configuration details needed to make your application run. You might create your own images or you might only use those created by others and published in a registry. To build your own image, you create a *Dockerfile* with a simple syntax for defining the steps needed to create the image and run it. Each instruction in a Dockerfile creates a layer in the image. When you change the Dockerfile and rebuild the image, only those layers which have changed are rebuilt. This is part of what makes images so lightweight, small, and fast, when compared to other virtualization technologies.

Containers

A container is a runnable instance of an image. You can create, start, stop, move, or delete a container using the Docker API or CLI. You can connect a container to one or more networks, attach storage to it, or even create a new image based on its current state.

- **Steps to isntall Docker on Ubuntu** (https://docs.docker.com/engine/install/ubuntu/)

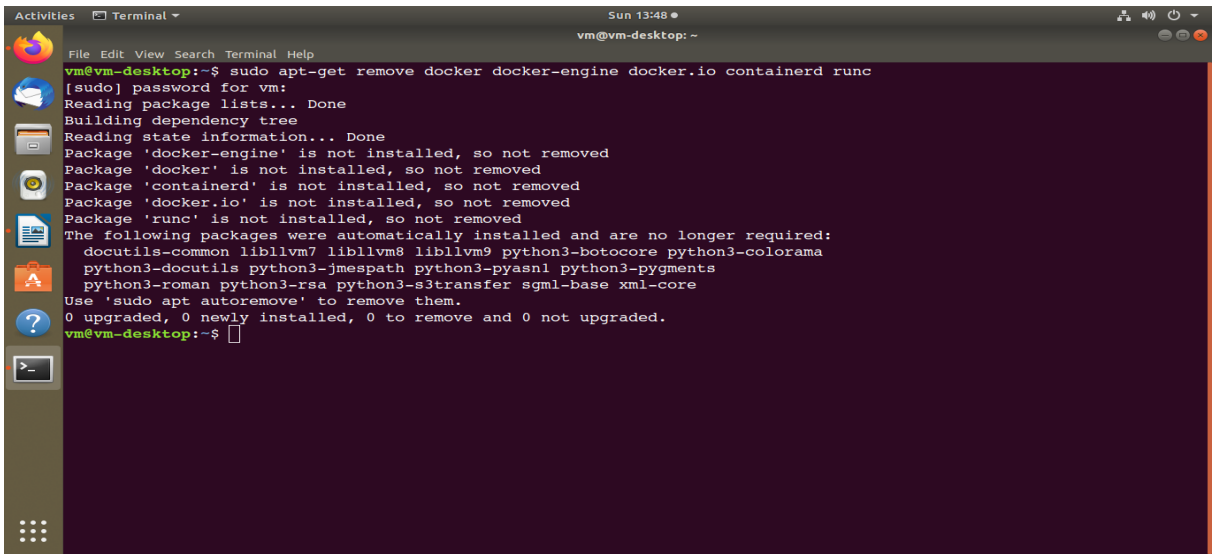
You can install docker in 3 ways:

1) Install using the repository (This is the recommended approach)
2) Install from a package
3) Install using the convenience script (most easiest way)

we will use fist method i.e. Install using respository

Step1: Begore staring isntallation remove the old installation if any, by using command:

$ sudo apt-get remove docker docker-engine docker.io containerd runc

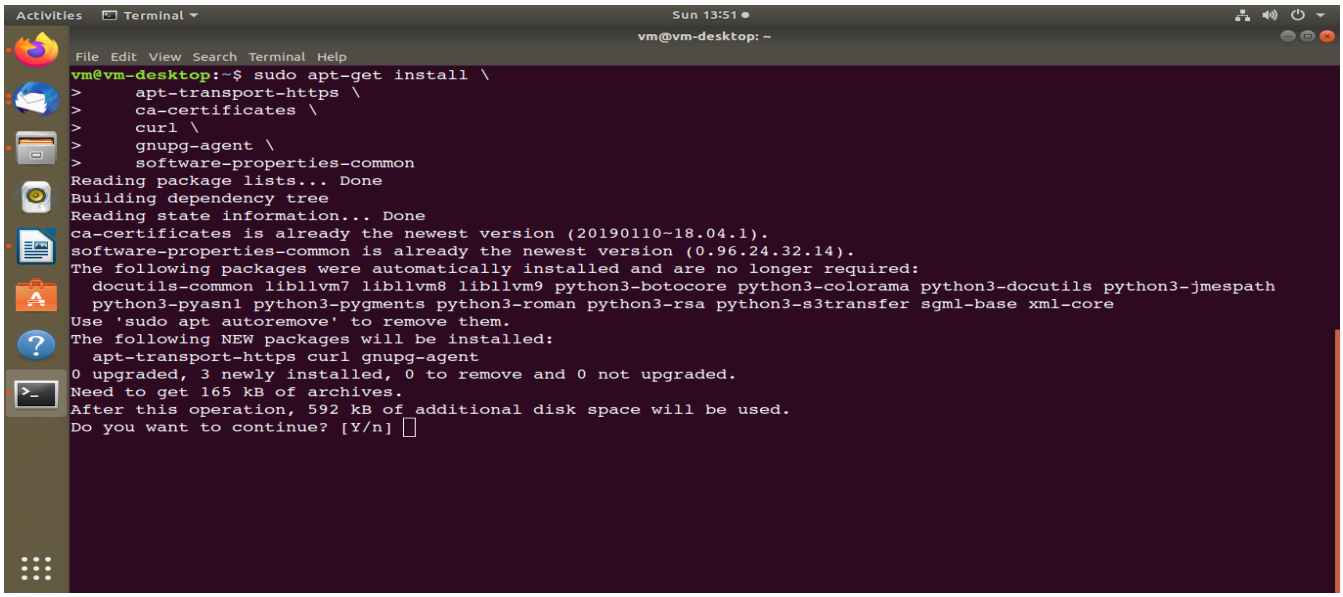


Step 2: update the packages by using command:

sudo apt-get update

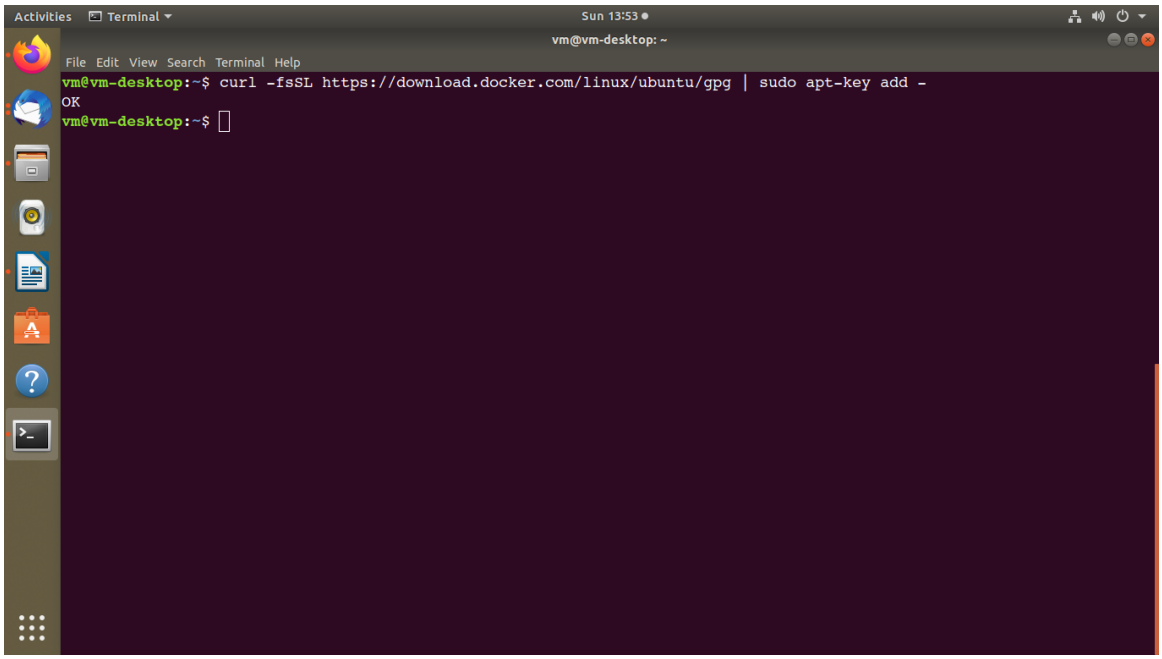
Step 3: install packages to allow apt to use a repository over HTTPS:

```
sudo apt-get install \
    apt-transport-https \
    ca-certificates \
    curl \
    gnupg-agent \
    software-properties-common
```



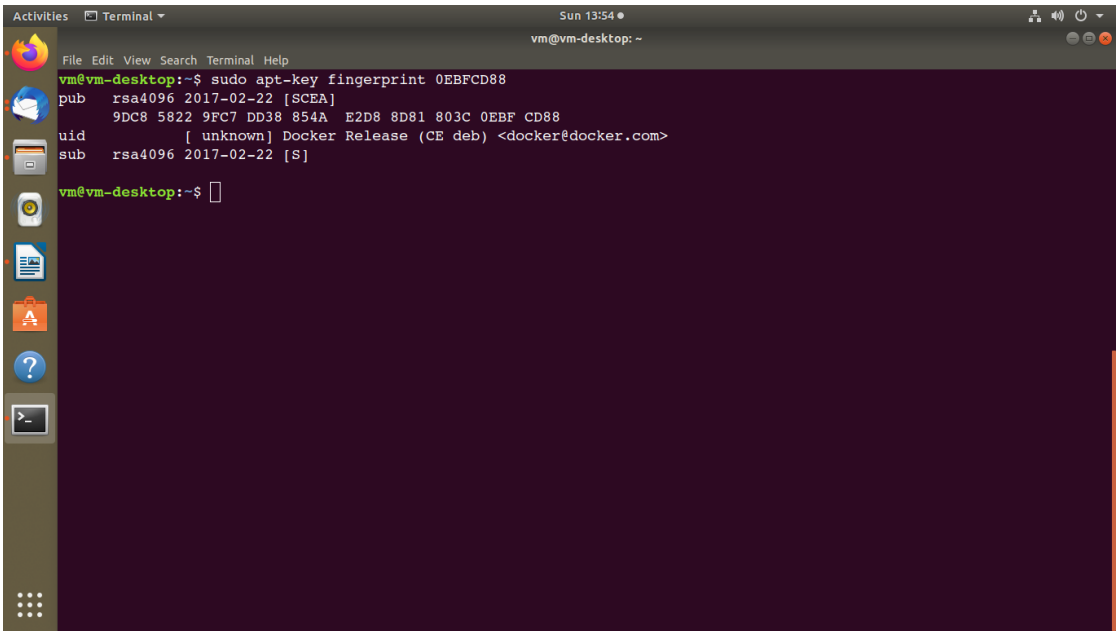
Step 4: Add Docker's official GPG key:

$ curl -fsSL https://download.docker.com/linux/ubuntu/gpg | sudo apt-key add –



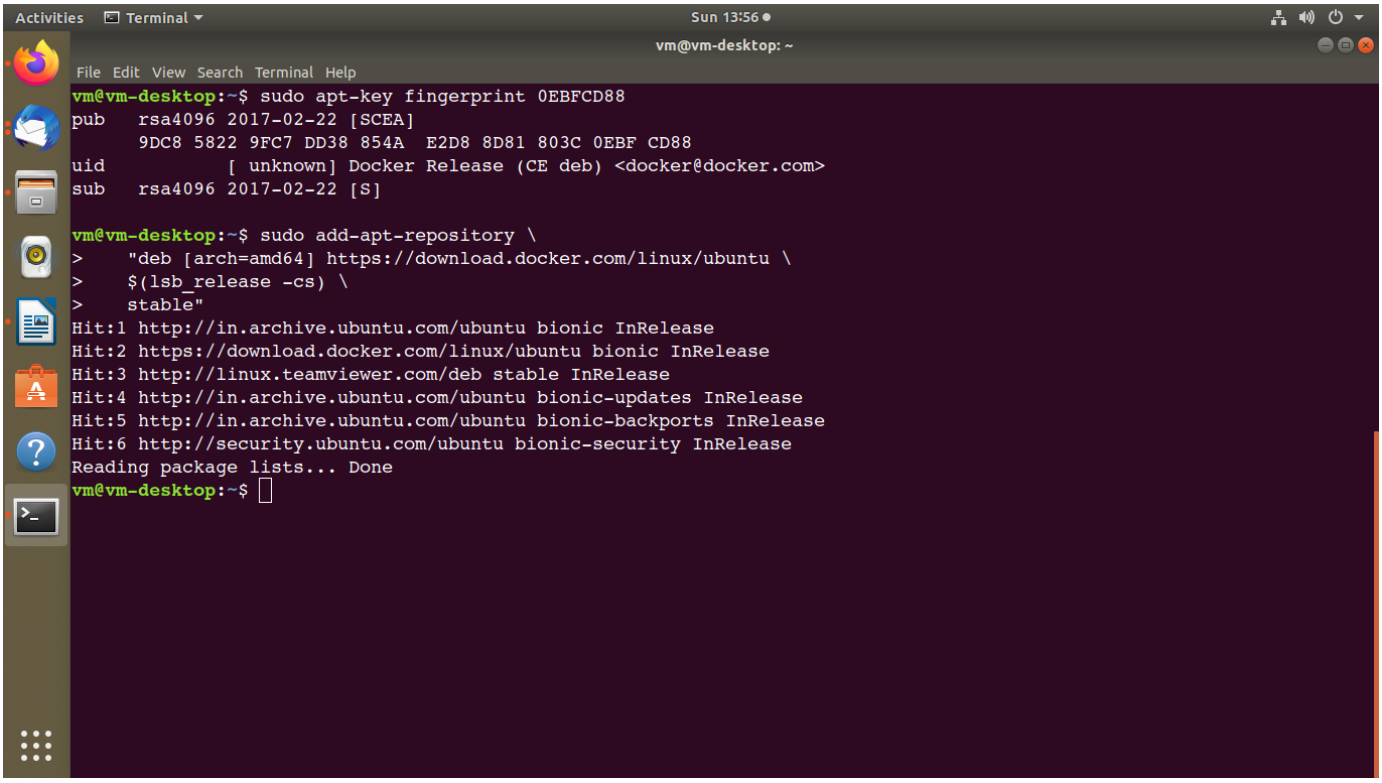
Step 5: Verify that you now have the key with the fingerprint 9DC8 5822 9FC7 DD38 854A E2D8 8D81 803C 0EBF CD88, by searching for the last 8 characters of the fingerprint.

$ sudo apt-key fingerprint 0EBFCD88



Step 6: Use the following command to set up the **stable** repository. To add the **nightly** or **test** repository, add the word nightly or test (or both) after the word stable in the commands below.

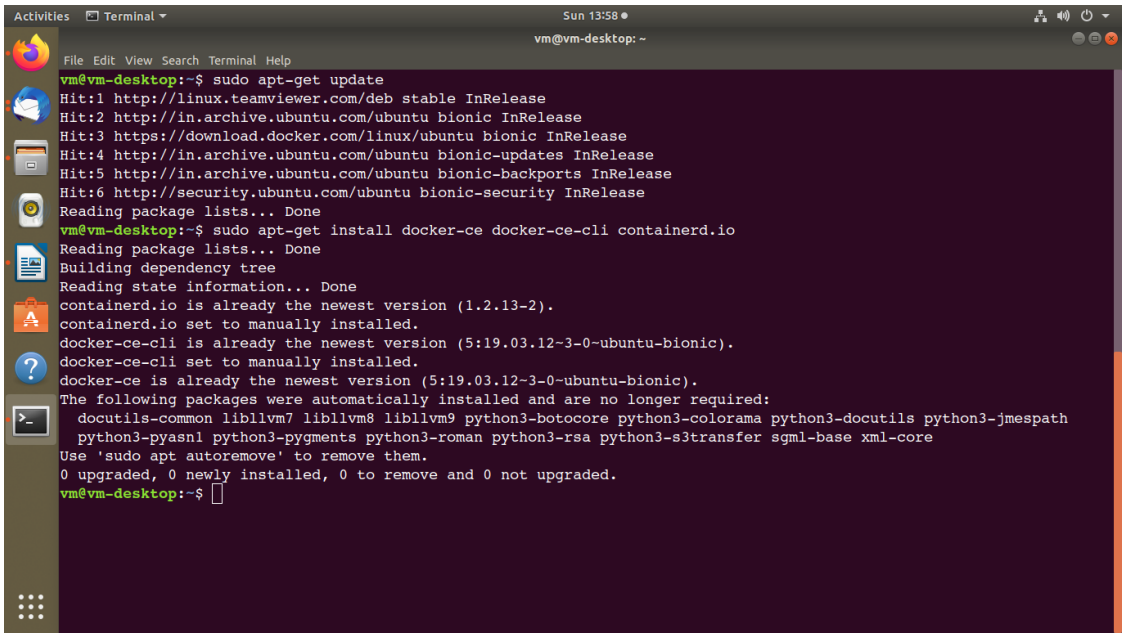
```
$ sudo add-apt-repository \
  "deb [arch=amd64] https://download.docker.com/linux/ubuntu \
  $(lsb_release -cs) \
  stable"
```



Install Docker Engine

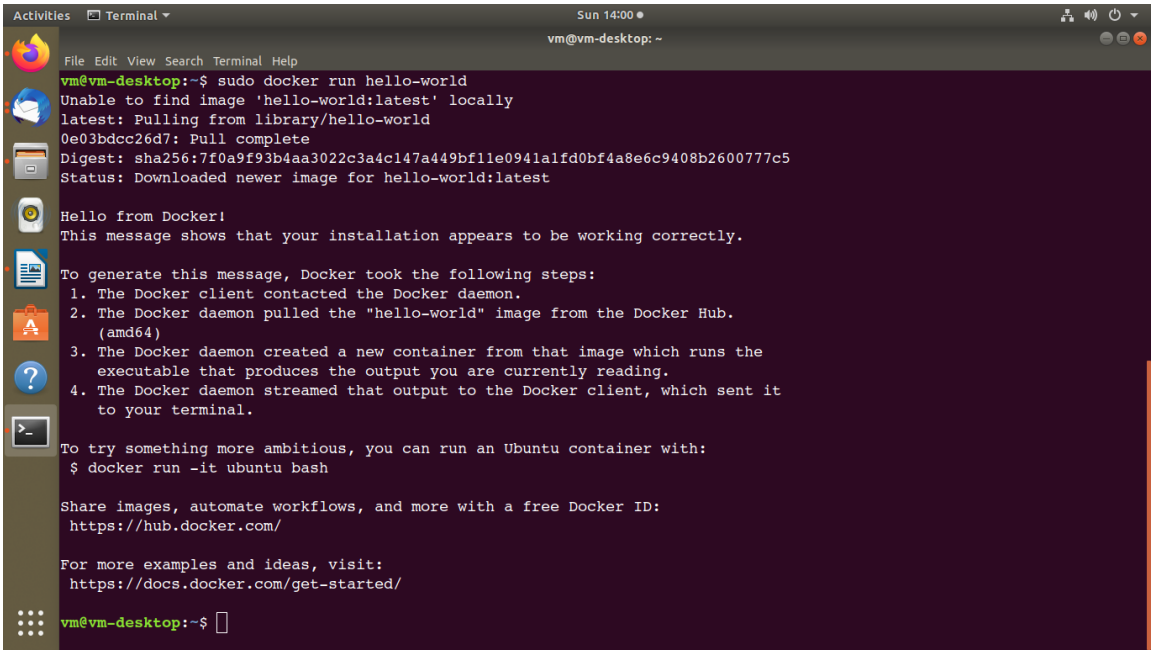
Step 7: Update the apt package index, and install the *latest version* of Docker Engine and containerd, or go to the next step to install a specific version:

$ sudo apt-get update
$ sudo apt-get install docker-ce docker-ce-cli containerd.io



Step 8: Verify that Docker Engine is installed correctly by running the hello-world image.

$ sudo docker run hello-world



Uninstall Docker Engine

Step 1: Uninstall the Docker Engine, CLI, and Containerd packages:

$ sudo apt-get purge docker-ce docker-ce-cli containerd.io

Step 2: Images, containers, volumes, or customized configuration files on your host are not automatically removed. To delete all images, containers, and volumes:

```
$ sudo rm -rf /var/lib/docker
```

Docker Compose

Compose is a tool for defining and running multi-container Docker applications. With Compose, you use a YAML file to configure your application's services. Then, with a single command, you create and start all the services from your configuration. To learn more about all the features of Compose, see the list of features.
Compose works in all environments: production, staging, development, testing, as well as CI workflows. You can learn more about each case in Common Use Cases.
Using Compose is basically a three-step process:

1. Define your app's environment with a Dockerfile so it can be reproduced anywhere.
2. Define the services that make up your app in docker-compose.yml so they can be run together in an isolated environment.
3. Run docker-compose up and Compose starts and runs your entire app

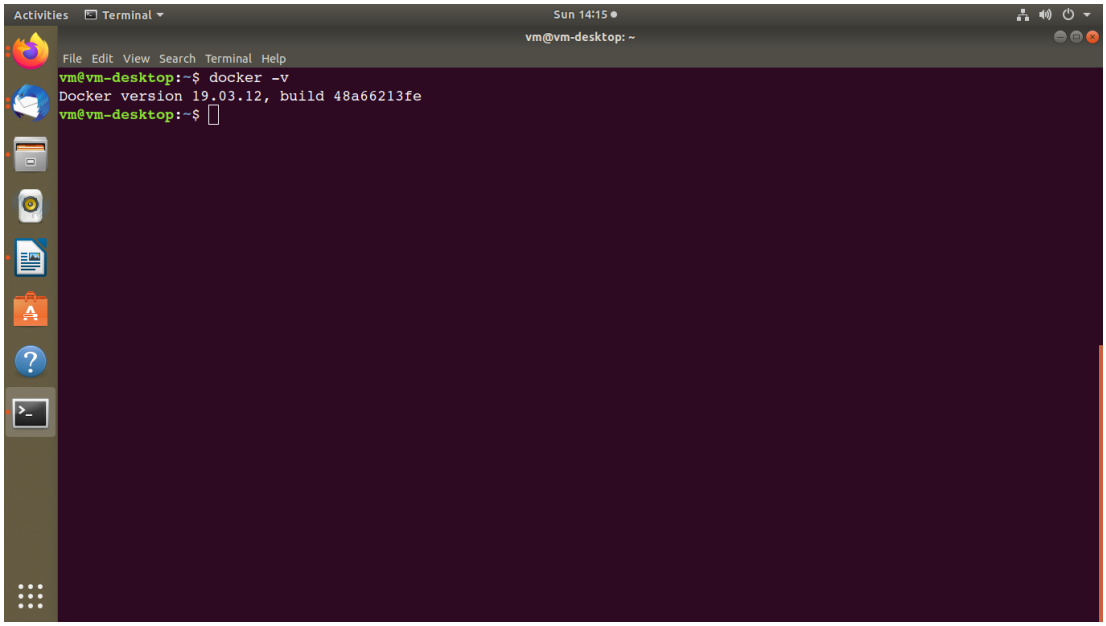
---

Docker Desktop

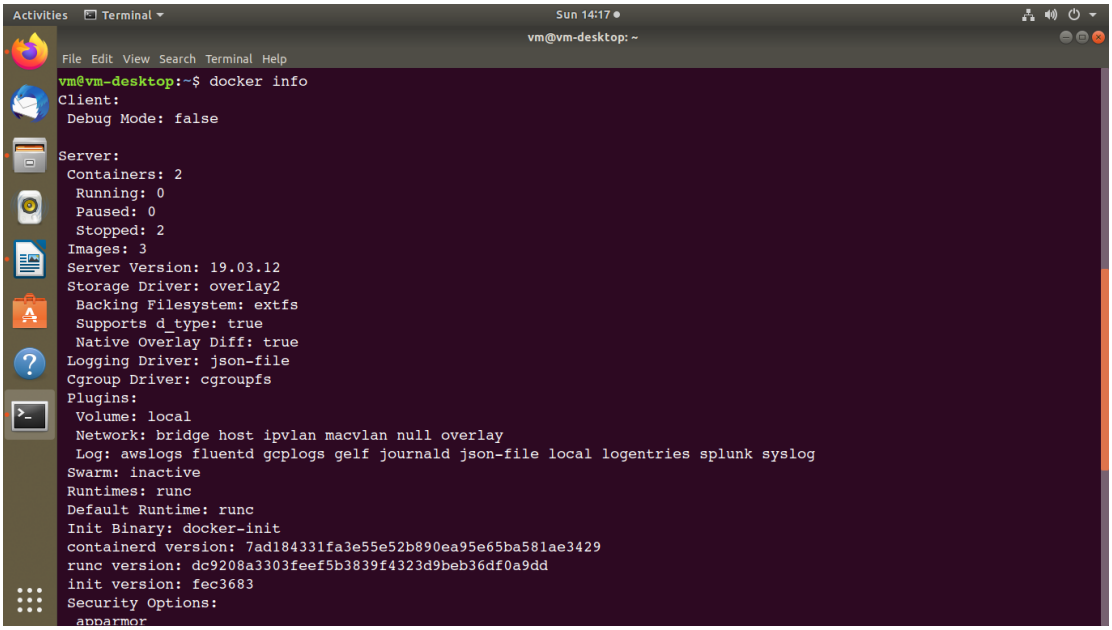
Docker Desktop is an easy-to-install application for your Mac or Windows environment that enables you to build and share containerized applications and microservices. Docker Desktop includes Docker Engine, Docker CLI client, Docker Compose, Notary, Kubernetes, and Credential Helper.

Docker Command Line (commands)

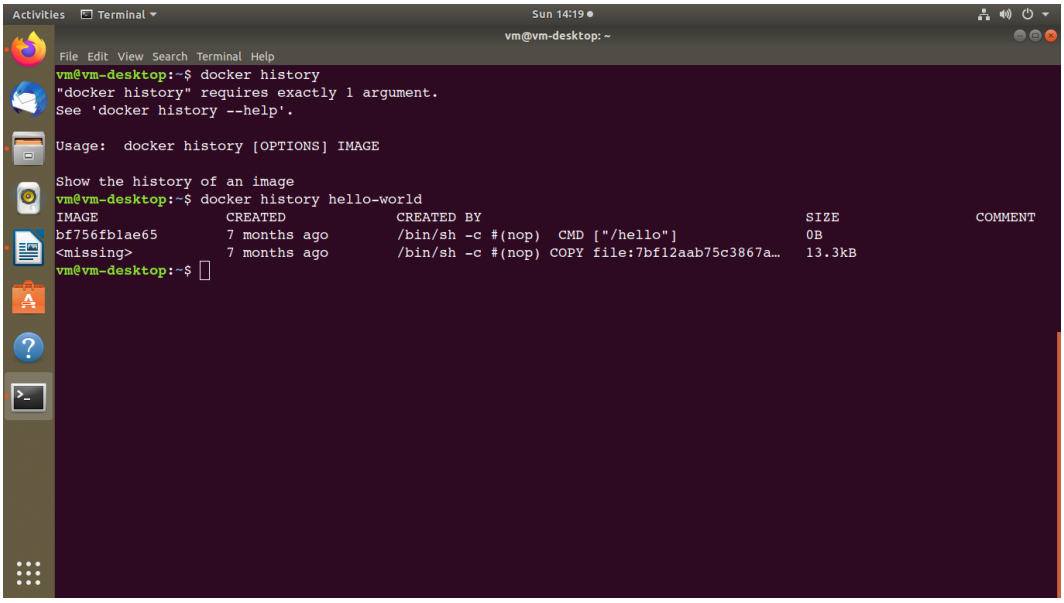
Ref: https://docs.docker.com/engine/reference/commandline/docker/
Commands
1) docker -v



2) docker info - Display system-wide information
docker info [OPTIONS]



3) docker history - Show the history of an image
docker history [OPTIONS] IMAGE



4) docker image - Manage images

a) docker ps - List containers

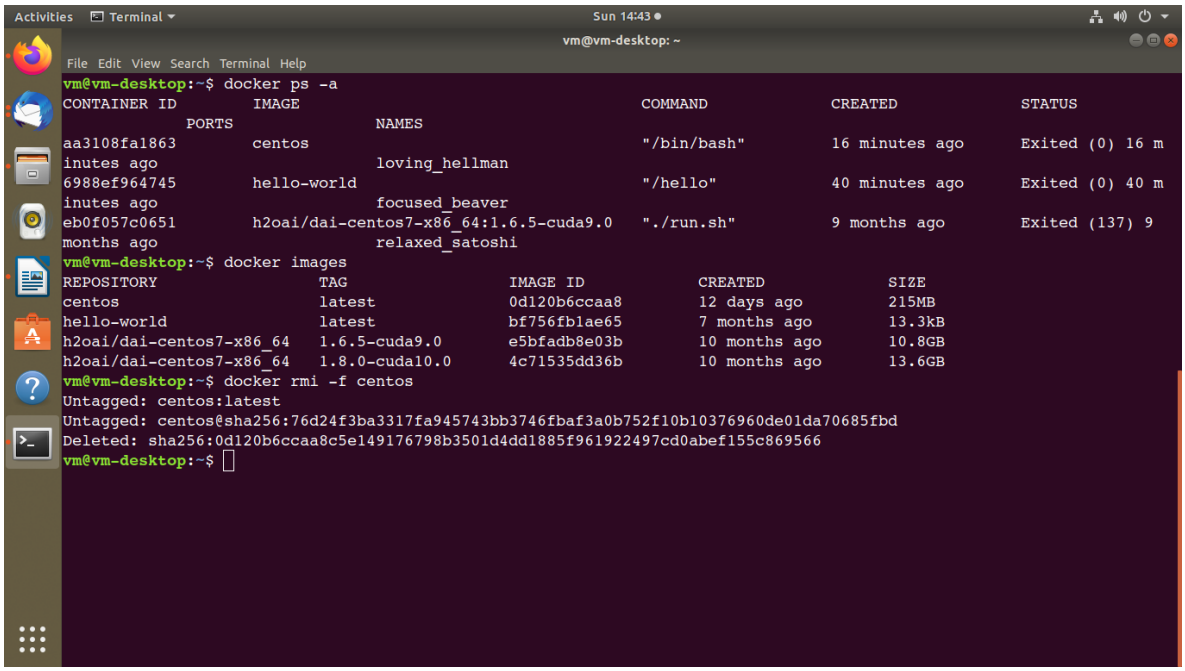
docker ps [OPTIONS]

b) docker images - List images

docker images [OPTIONS] [REPOSITORY[:TAG]]

c) docker rmi -Remove one or more images

docker rmi [OPTIONS] IMAGE [IMAGE...]



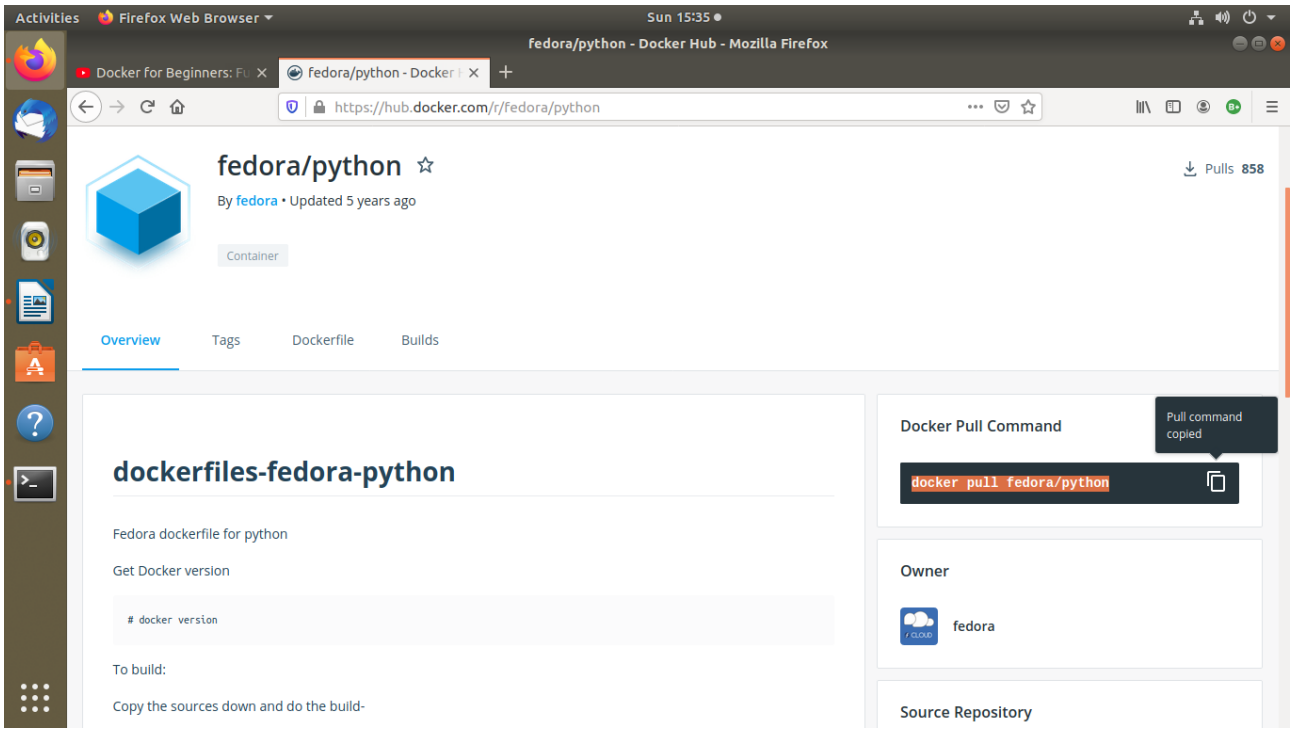
Install Tensorflow image or cupy image or any other image you want form the Docker hub. Here, I am pulling plain fedora image from docker.

Steps 1) login to docker hub account.

2) search “image name”in search box (I have searched for fedora/python image)

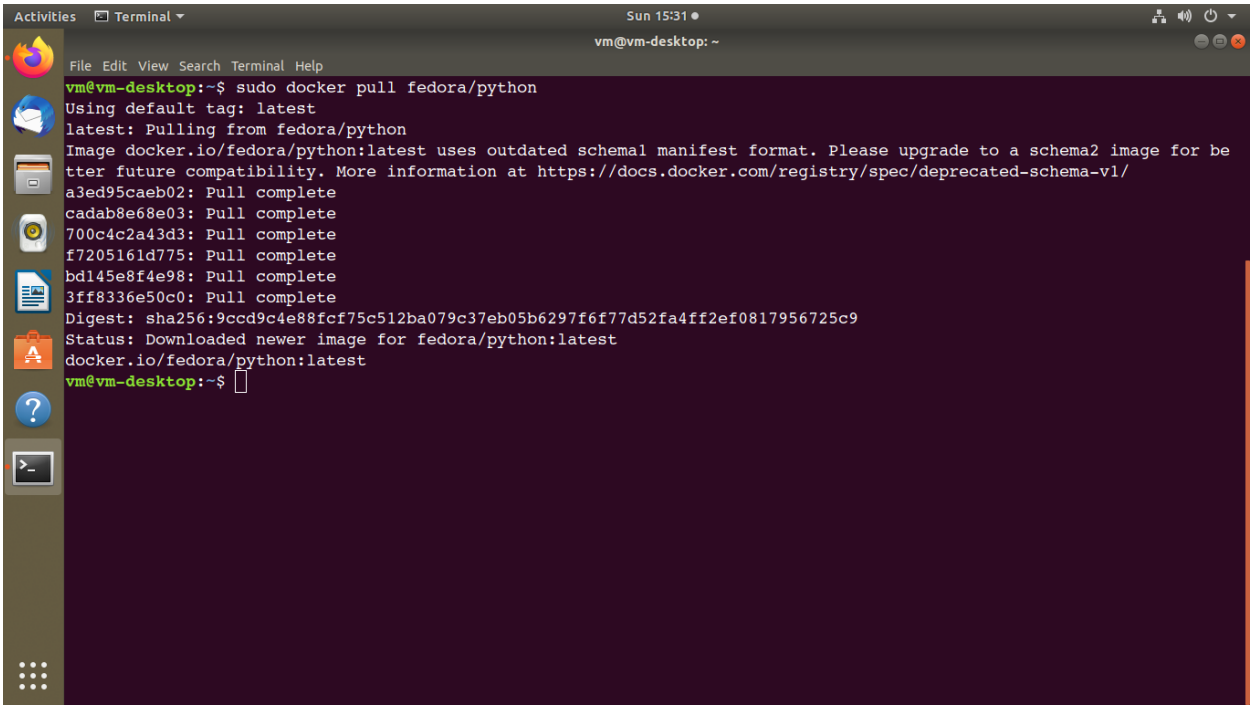
3) open the image link

4) copy the commnad



5) run the copied command in terminal

docker pull fedora/python



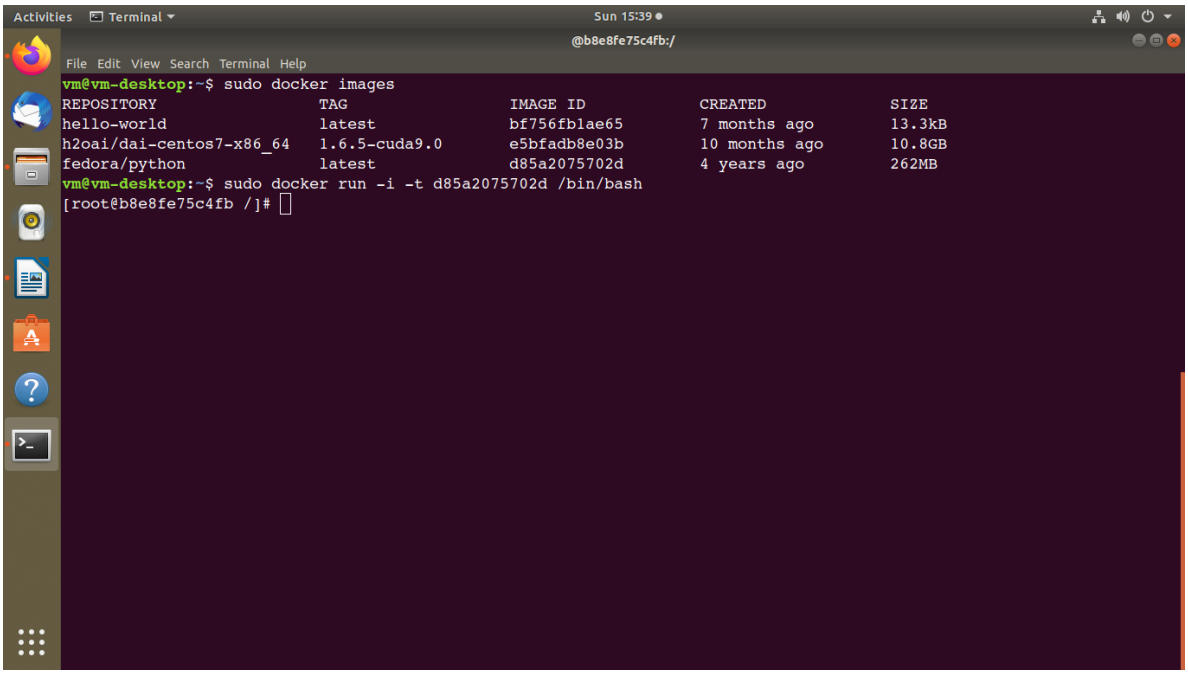
6) list the images by using command: sudo docker images

7) run the currently downloaded image using command:

sudo docker run -i -t d85a2075702d /bin/bash

-i => for input or interactive

-t => terminal



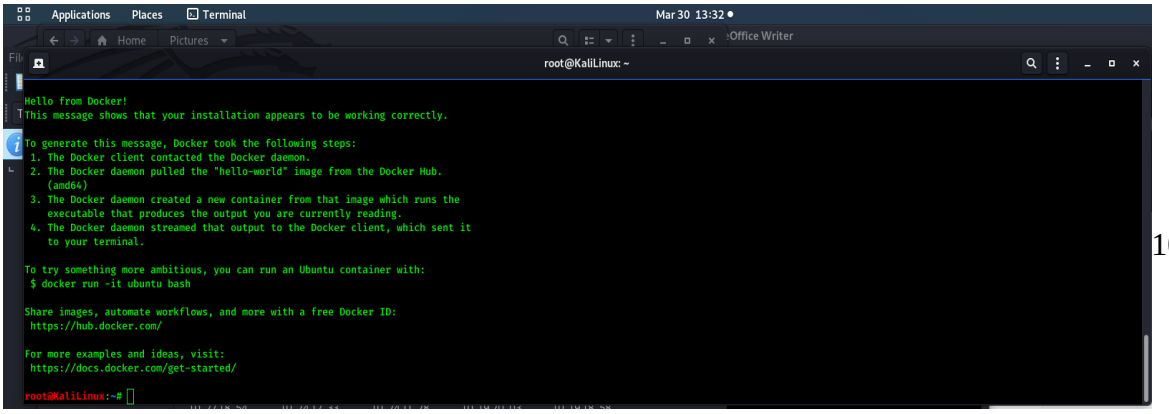
8) docker rm - Remove one or more containers

docker rm [OPTIONS] CONTAINER [CONTAINER...]

9) Docker container logs

Usage: docker container logs [OPTIONS] CONTAINER

Fetch the logs of a container

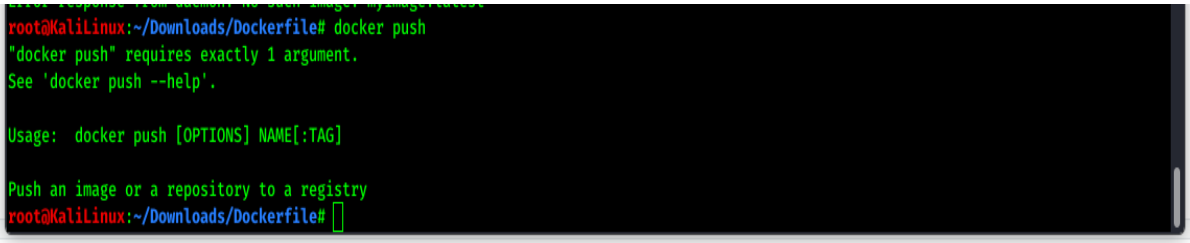


10)

Docker push

Usage: docker push [OPTIONS] NAME[:TAG]

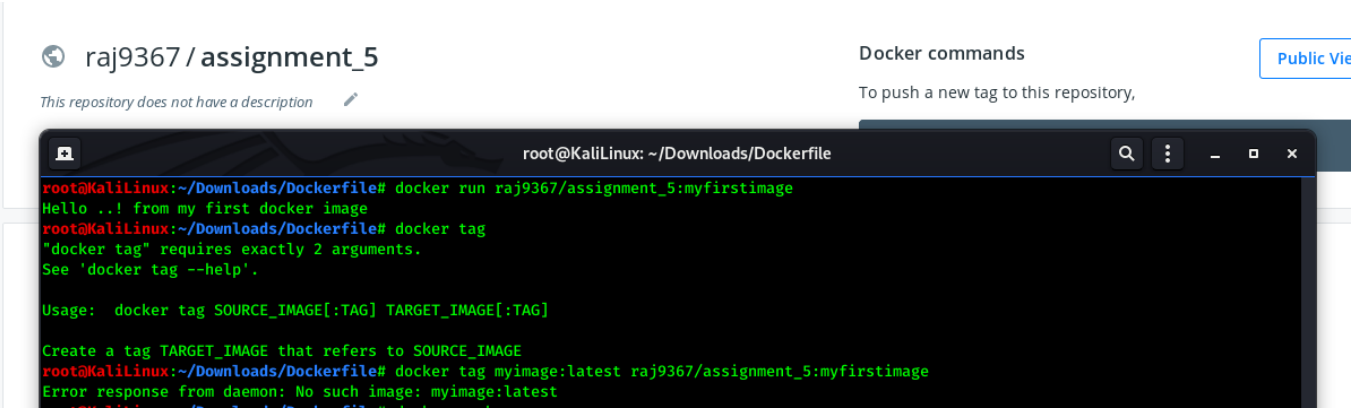
Push an image or a repository to a registry



11) Docker tag

Usage: docker tag SOURCE_IMAGE[:TAG] TARGET_IMAGE[:TAG]

Create a tag TARGET_IMAGE that refers to SOURCE_IMAGE

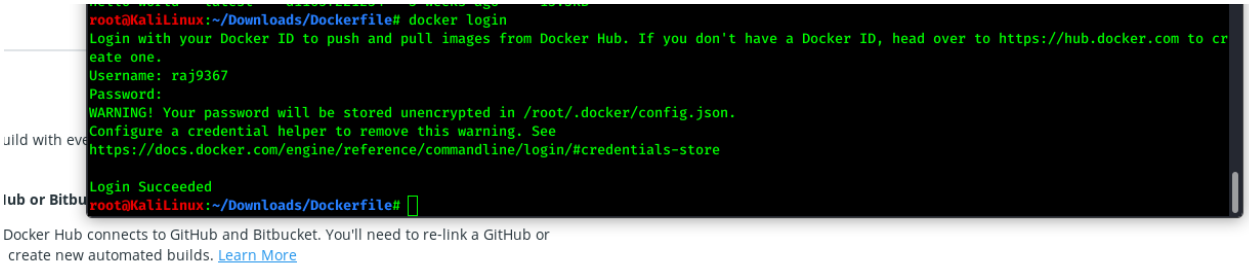


10) Docker login

Usage: docker login [OPTIONS] [SERVER]

Log in to a Docker registry.

If no server is specified, the default is defined by the daemon.



Docker Hub

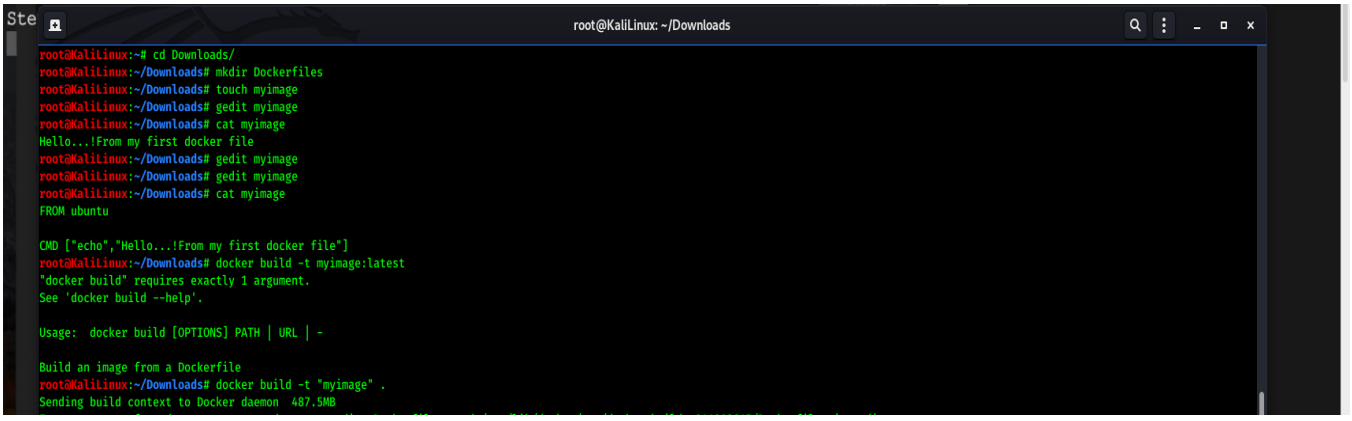
Ref: https://docs.docker.com/docker-hub/
Docker Hub is a service provided by Docker for finding and sharing container images with your team. It provides the following major features:

- Repositories: Push and pull container images.
- Teams & Organizations: Manage access to private repositories of container images.
- Official Images: Pull and use high-quality container images provided by Docker.
- Publisher Images: Pull and use high- quality container images provided by external vendors.
- Builds: Automatically build container images from GitHub and Bitbucket and push them to Docker Hub.
- Webhooks: Trigger actions after a successful push to a repository to integrate Docker Hub with other services.

Step 1: Sign up for Docker Hub

Start by creating an account.

Sterp 2) write own docker file and push it.

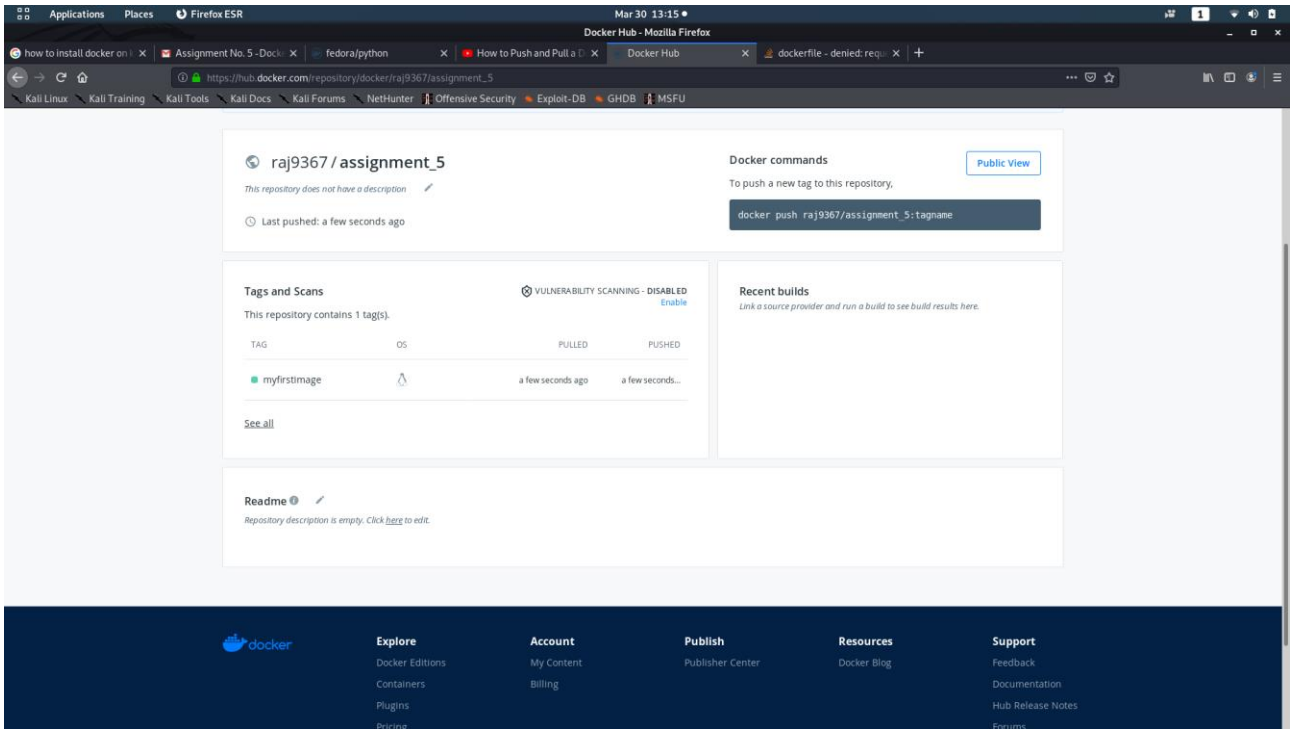


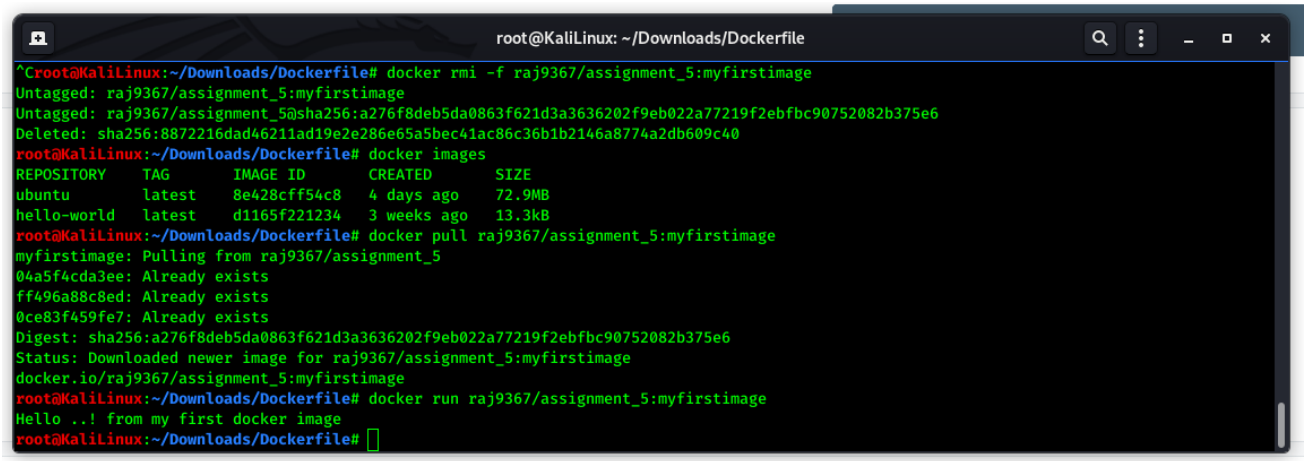
```
root@KaliLinux:~/Downloads/Dockerfile# docker tag myimage:latest raj9367/assignment_5:myfirstimage
root@KaliLinux:~/Downloads/Dockerfile# docker push raj9367/assignment_5:myfirstimage
The push refers to repository [docker.io/raj9367/assignment_5]
d4dfaa212623: Pushed
cba97cc5811c: Pushed
0c78fac124da: Pushed
myfirstimage: digest: sha256:a276f8deb5da0863f621d3a3636202f9eb022a77219f2ebfbc90752082b375e6 size: 943
root@KaliLinux:~/Downloads/Dockerfile#
```

Enable

Link a source provider and run a build to see build results here.

: of the most recent tags here.





Assignment No 6

Title: Setup Single Node Kubernetes Cluster with Minikube.

Theory:

- **What is Kubernetes?**

Kubernetes is a portable, extensible, open-source platform for managing containerized workloads and services, that facilitates both declarative configuration and automation. It has a large, rapidly growing ecosystem. Kubernetes services, support, and tools are widely available.

The name Kubernetes originates from Greek, meaning helmsman or pilot. Google open-sourced the Kubernetes project in 2014. Kubernetes combines over 15 years of Google's experience running production workloads at scale with best-of-breed ideas and practices from the community

- **Why you need Kubernetes and what it can do**

Containers are a good way to bundle and run your applications. In a production environment, you need to manage the containers that run the applications and ensure that there is no downtime. For example, if a container goes down, another container needs to start. Wouldn't it be easier if this behavior was handled by a system?

That's how Kubernetes comes to the rescue! Kubernetes provides you with a framework to run distributed systems resiliently. It takes care of scaling and failover for your application, provides deployment patterns, and more. For example, Kubernetes can easily manage a canary deployment for your system.

Kubernetes provides you with:

- **Service discovery and load balancing**
- **Storage orchestration**

- **Automated rollouts and rollbacks**
- **Automatic bin packing**
- **Self-healing**
- **Secret and configuration management**

Kubernetes Components

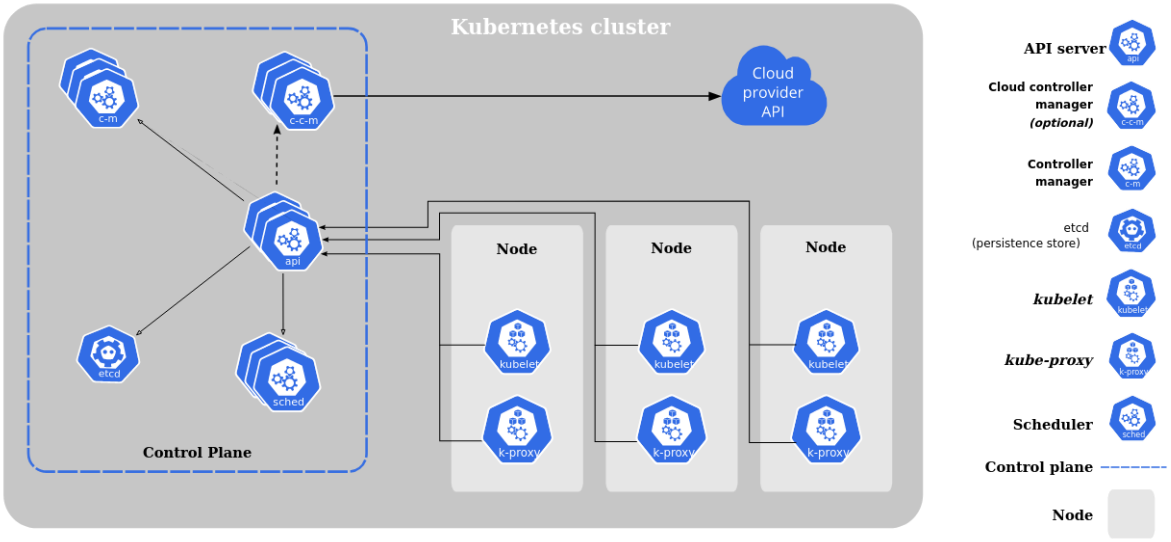
When you deploy Kubernetes, you get a cluster.

A Kubernetes cluster consists of a set of worker machines, called nodes, that run containerized applications. Every cluster has at least one worker node.

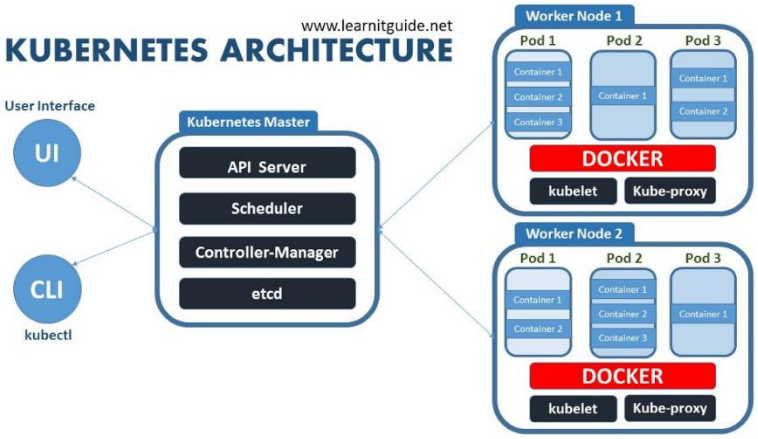
The worker node(s) host the Pods that are the components of the application workload. The control plane manages the worker nodes and the Pods in the cluster. In production environments, the control plane usually runs across multiple computers and a cluster usually runs multiple nodes, providing fault-tolerance and high availability.

This document outlines the various components you need to have a complete and working Kubernetes cluster.

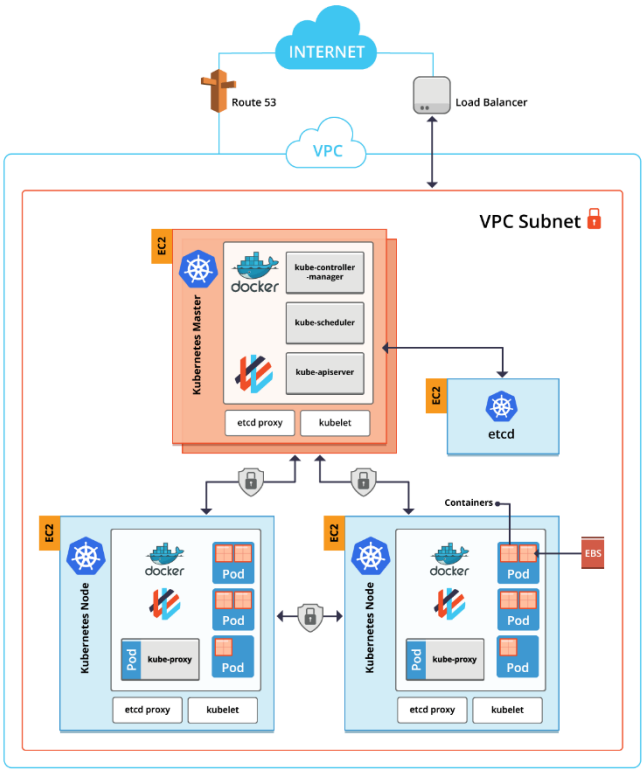
Here's the diagram of a Kubernetes cluster with all the components tied together.



- **Kubernetes Architecture**



Kubernetes Architecture on AWS:



Control Plane Components

The control plane's components make global decisions about the cluster (for example, scheduling), as well as detecting and responding to cluster events (for example, starting up a new pod when a deployment's replicas field is unsatisfied).

Control plane components can be run on any machine in the cluster. However, for simplicity, set up scripts typically start all control plane components on the same machine, and do not run user containers on this machine. See Building High-Availability Clusters for an example multi-master-VM setup.

kube-apiserver

The API server is a component of the Kubernetes control plane that exposes the Kubernetes API. The API server is the front end for the Kubernetes control plane.

The main implementation of a Kubernetes API server is kube-apiserver. kube-apiserver is designed to scale horizontally—that is, it scales by deploying more instances. You can run several instances of kube-apiserver and balance traffic between those instances.

etcd

Consistent and highly-available key value store used as Kubernetes' backing store for all cluster data.

If your Kubernetes cluster uses etcd as its backing store, make sure you have a back up plan for those data.

You can find in-depth information about etcd in the official documentation.

kube-scheduler

Control plane component that watches for newly created Pods with no assigned node, and selects a node for them to run on.

Factors taken into account for scheduling decisions include: individual and collective resource requirements, hardware/software/policy constraints, affinity and anti-affinity specifications, data locality, inter-workload interference, and deadlines.

kube-controller-manager

Control Plane component that runs controller processes.

Logically, each controller is a separate process, but to reduce complexity, they are all compiled into a single binary and run in a single process.

These controllers include:

- Node controller: Responsible for noticing and responding when nodes go down.
- Replication controller: Responsible for maintaining the correct number of pods for every replication controller object in the system.
- Endpoints controller: Populates the Endpoints object (that is, joins Services & Pods).
- Service Account & Token controllers: Create default accounts and API access tokens for new namespaces.

cloud-controller-manager

A Kubernetes control plane component that embeds cloud-specific control logic. The cloud controller manager lets you link your cluster into your cloud provider's API, and separates out the components that interact with that cloud platform from components that just interact with your cluster.

The cloud-controller-manager only runs controllers that are specific to your cloud provider. If you are running Kubernetes on your own premises, or in a learning environment inside your own PC, the cluster does not have a cloud controller manager.

As with the kube-controller-manager, the cloud-controller-manager combines several logically independent control loops into a single binary that you run as a single process. You can scale horizontally (run more than one copy) to improve performance or to help tolerate failures.

The following controllers can have cloud provider dependencies:

- Node controller: For checking the cloud provider to determine if a node has been deleted in the cloud after it stops responding
- Route controller: For setting up routes in the underlying cloud infrastructure
- Service controller: For creating, updating and deleting cloud provider load balancers

Node Components

Node components run on every node, maintaining running pods and providing the Kubernetes runtime environment.

kubelet

An agent that runs on each node in the cluster. It makes sure that containers are running in a Pod.

The kubelet takes a set of PodSpecs that are provided through various mechanisms and ensures that the containers described in those PodSpecs are running and healthy. The kubelet doesn't manage containers which were not created by Kubernetes.

kube-proxy

kube-proxy is a network proxy that runs on each node in your cluster, implementing part of the Kubernetes Service concept.

kube-proxy maintains network rules on nodes. These network rules allow network communication to your Pods from network sessions inside or outside of your cluster.

kube-proxy uses the operating system packet filtering layer if there is one and it's available. Otherwise, kube-proxy forwards the traffic itself.

Container runtime

The container runtime is the software that is responsible for running containers.

Kubernetes supports several container runtimes: Docker, containerd, CRI-O, and any implementation of the Kubernetes CRI (Container Runtime Interface).

Pods

Pods are the smallest deployable units of computing that you can create and manage in Kubernetes.

A *Pod* (as in a pod of whales or pea pod) is a group of one or more containers, with shared storage/network resources, and a specification for how to run the containers. A Pod's contents are always co-located and co-scheduled, and run in a shared context. A Pod models an application-specific "logical host": it contains one or more application containers which are relatively tightly coupled. In non-cloud contexts, applications executed on the same physical or virtual machine are analogous to cloud applications executed on the same logical host.

Using Pods

Usually you don't need to create Pods directly, even singleton Pods. Instead, create them using workload resources such as Deployment or Job. If your Pods need to track state, consider the StatefulSet resource.

Pods in a Kubernetes cluster are used in two main ways:

- **Pods that run a single container**. The "one-container-per-Pod" model is the most common Kubernetes use case; in this case, you can think of a Pod as a wrapper around a single container; Kubernetes manages Pods rather than managing the containers directly.

- **Pods that run multiple containers that need to work together**. A Pod can encapsulate an application composed of multiple co-located containers that are tightly coupled and need to share resources. These co-located containers form a single cohesive unit of service—for example, one container serving data stored in a shared volume to the public, while a separate *sidecar* container refreshes or updates those files. The Pod wraps these containers, storage resources, and an ephemeral network identity together as a single unit.

Service

An abstract way to expose an application running on a set of Pods as a network service.

With Kubernetes you don't need to modify your application to use an unfamiliar service discovery mechanism. Kubernetes gives Pods their own IP addresses and a single DNS name for a set of Pods, and can load-balance across them.

Ingress

FEATURE STATE: Kubernetes v1.19 [stable]

An API object that manages external access to the services in a cluster, typically HTTP.

Ingress may provide load balancing, SSL termination and name-based virtual hosting.

Terminology

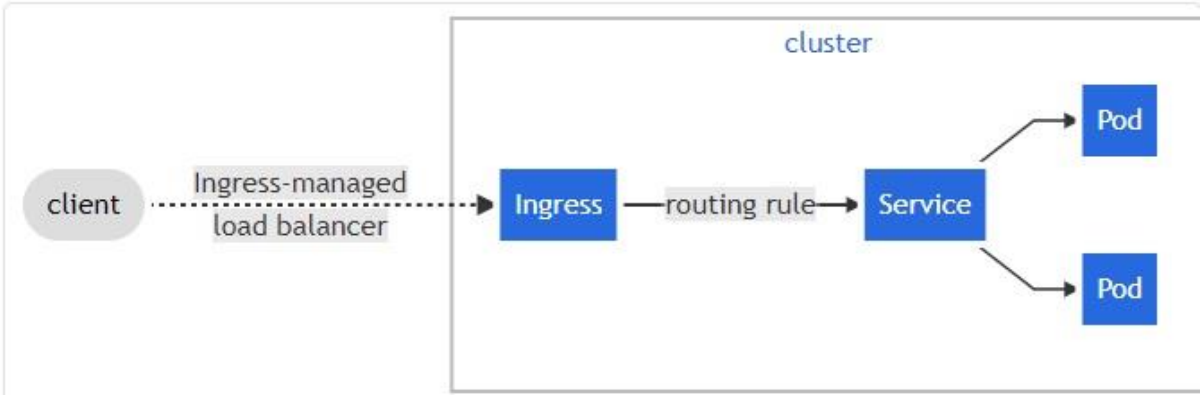
For clarity, this guide defines the following terms:

- Node: A worker machine in Kubernetes, part of a cluster.
- Cluster: A set of Nodes that run containerized applications managed by Kubernetes. For this example, and in most common Kubernetes deployments, nodes in the cluster are not part of the public internet.
- Edge router: A router that enforces the firewall policy for your cluster. This could be a gateway managed by a cloud provider or a physical piece of hardware.
- Cluster network: A set of links, logical or physical, that facilitate communication within a cluster according to the Kubernetes networking model.
- Service: A Kubernetes Service that identifies a set of Pods using label selectors. Unless mentioned otherwise, Services are assumed to have virtual IPs only routable within the cluster network.

What is Ingress?

Ingress exposes HTTP and HTTPS routes from outside the cluster to services within the cluster. Traffic routing is controlled by rules defined on the Ingress resource.

Here is a simple example where an Ingress sends all its traffic to one Service:



An Ingress may be configured to give Services externally-reachable URLs, load balance traffic, terminate SSL / TLS, and offer name-based virtual hosting. An Ingress controller is responsible for fulfilling the Ingress, usually with a load balancer, though it may also configure your edge router or additional frontends to help handle the traffic.

An Ingress does not expose arbitrary ports or protocols. Exposing services other than HTTP and HTTPS to the internet typically uses a service of type Service.Type=NodePort or Service.Type=LoadBalancer.

Try Kubernetes:

There are many ways to install Kubernetes.

1) Manual means that you manually set up everything from the networking bits over downloading, configuring and launching components such as etcd or kube-apiserver. You've got full control over what goes where, but it might take you some time and it's error-prone. The ultimate reference in this area is Kelsey Hightower's Kubernetes The Hard Way (KTHW).

2) Installer are CLI tools that leverage templates and/or automation tools such as Terraform and Ansible. You typically have a lot control over what is going on, but less than in the manual approach. Examples include:

- OpenCredo's Kubernetes from scratch to AWS with Terraform and Ansible
- kubeadm (bare metal installation, a building block for other installers)
- kops, mainly AWS (AWS also provide service for kubernetes: EKS – Elastic Kubernets Service)
- kubicorn
- kubespray (Ansible-based, both bare-metal and cloud)
- OpenShift advanced install based on Ansible

3) Hosted effectively means little to no installation effort on your end. For example:

- Azure Container Service (ACS) with Kubernetes
- Google Container Engine (GKE)
- IBM Bluemix Container Service
- OpenShift Online (OSO)

4) Minikube or Minishift – Virtualized Environemnt for Kubernetes

Setup Single Node Kubernetes Cluster with Minikube

1) Installing Minikube -> https://kubernetes.io/docs/tasks/tools/install-minikube/

Install Virtualbox latest edition.

2) Setting up Minikube on virtualbox -> https://kubernetes.io/docs/tasks/tools/install-minikube/

You will need to keep the minikube in the PATH both on Windows/Linux

Use the following instruction to setup single node Kubernetes cluster

minikube start --driver=virtualbox

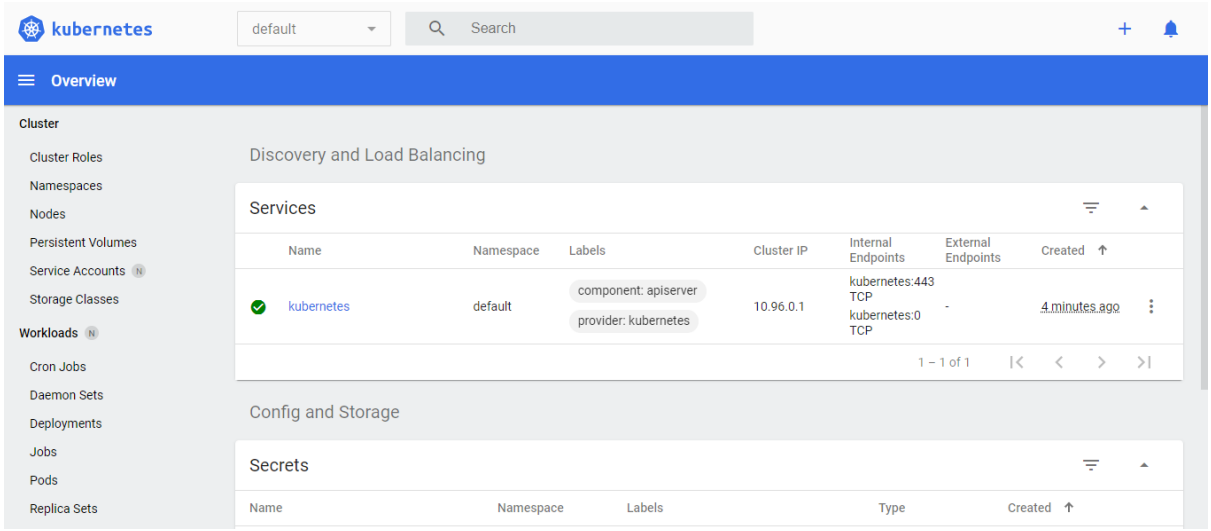
You can also set custom configuration like

minikube start --driver=virtualbox --cpus=2 --memory=4096m

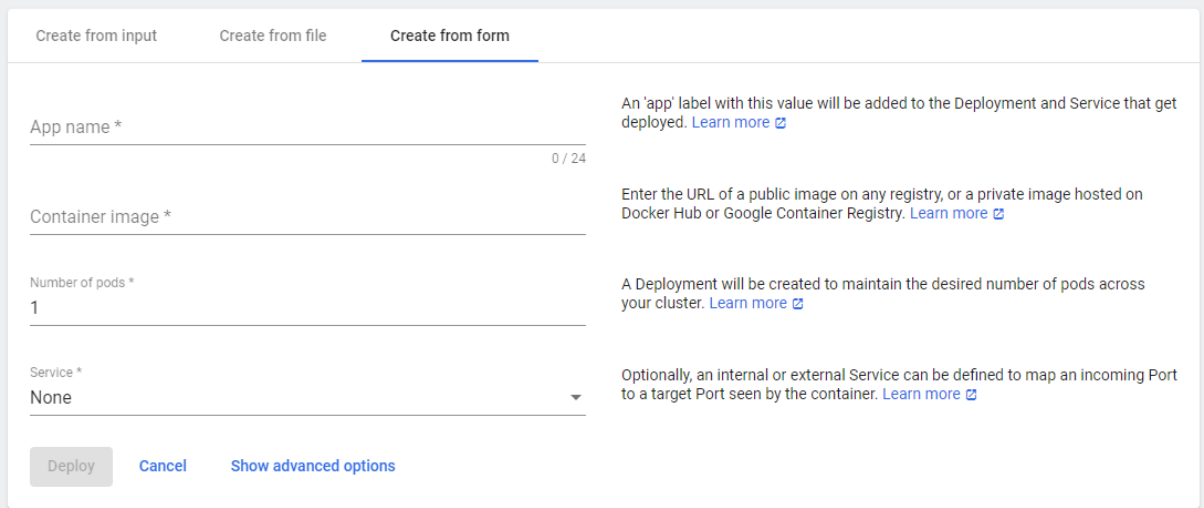
3) Open the minikube dashboard with following command. It will take 2-10 mins depending on your bandwidth.

minikube dashboard

4) The dashboard will open in your default browser



5) On the right side corner you will see a symbol for +, click on it, and go to -> Create from form



6) Enter the details as below

App Name: nginx

Container image: nginx

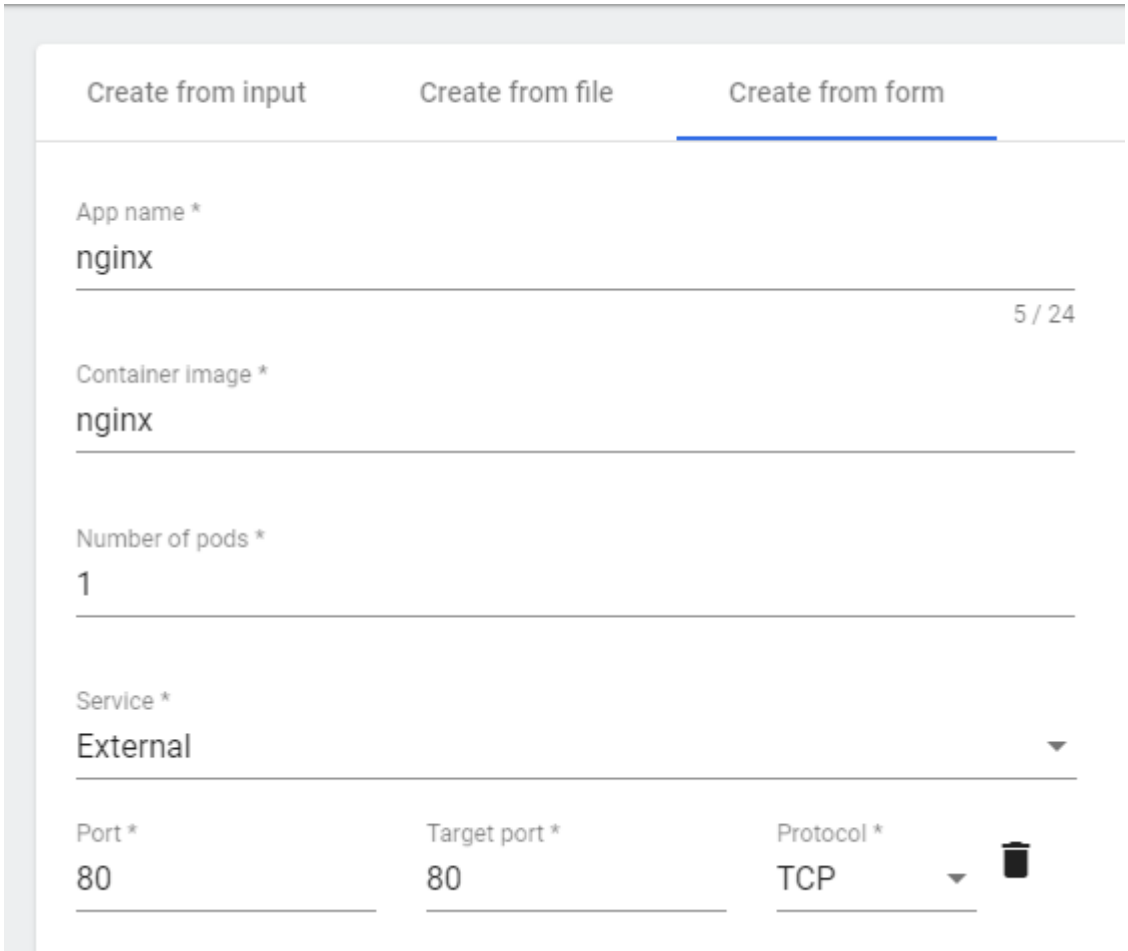
Number of pods: 1

Service: External

Port: 80

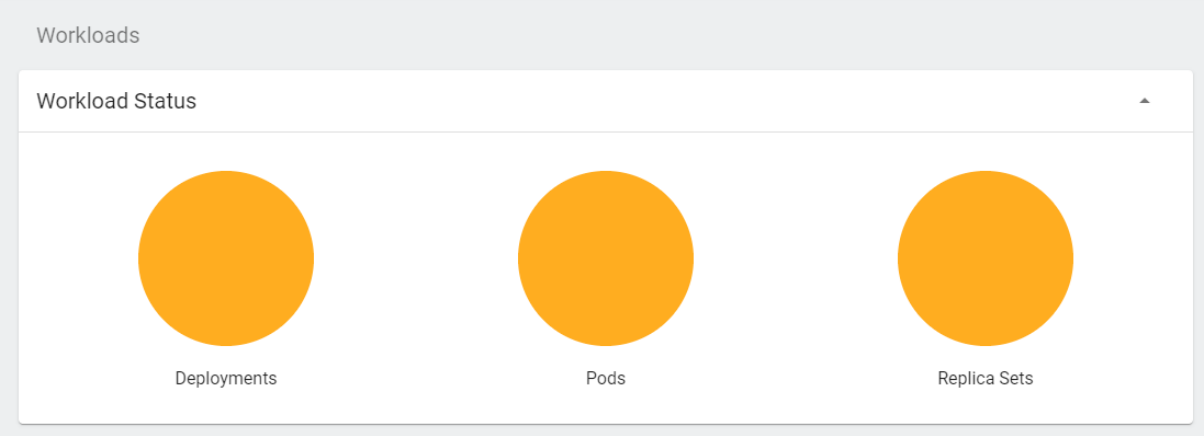
Target Port: 80

Protocol: TCP

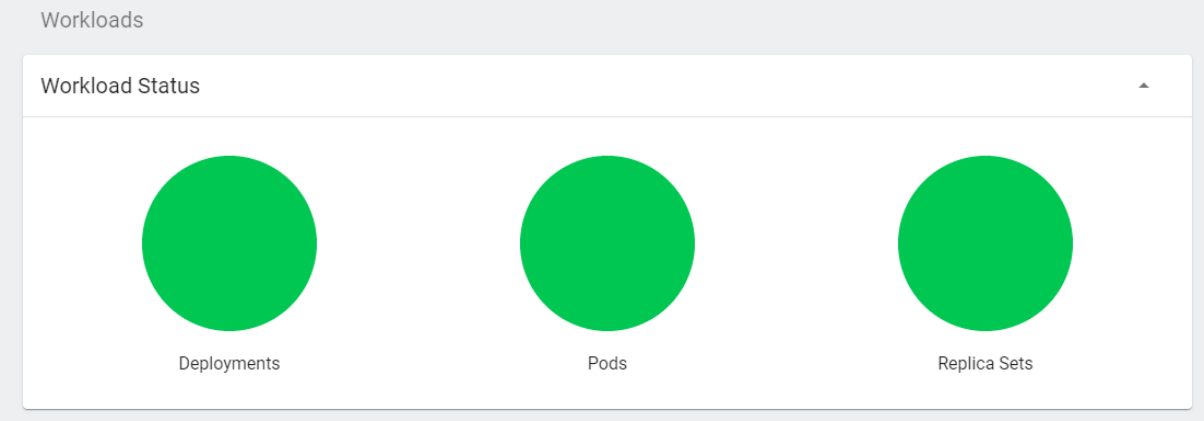


Click Deploy

7) Now you will see something like this,



Wait for it to turn green.

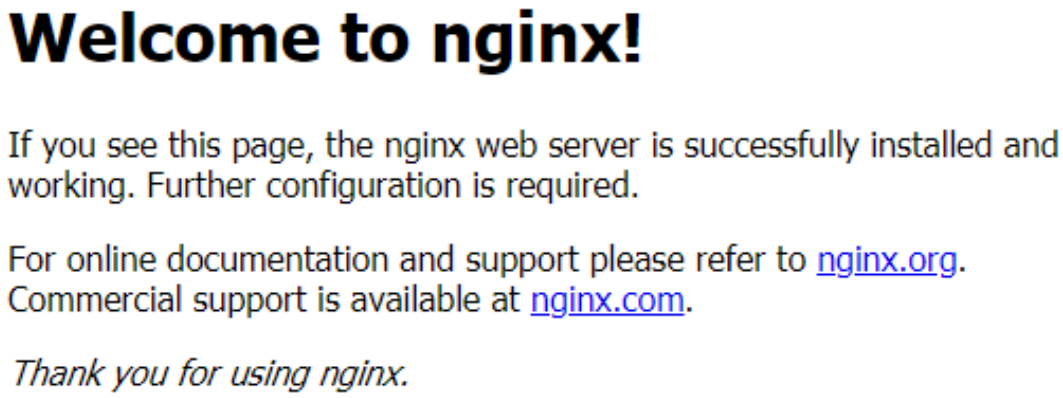


You will see the details for deployment below.

8) Access the nginx application in your browser with following command.

minikube service nginx

The nginx default page will open in browser and you will see the service details as well.



- **References for further study:**

1) https://kubernetes.io/docs/home/

2) you need to learn Kubernetes RIGHT NOW!! - Networkchuck

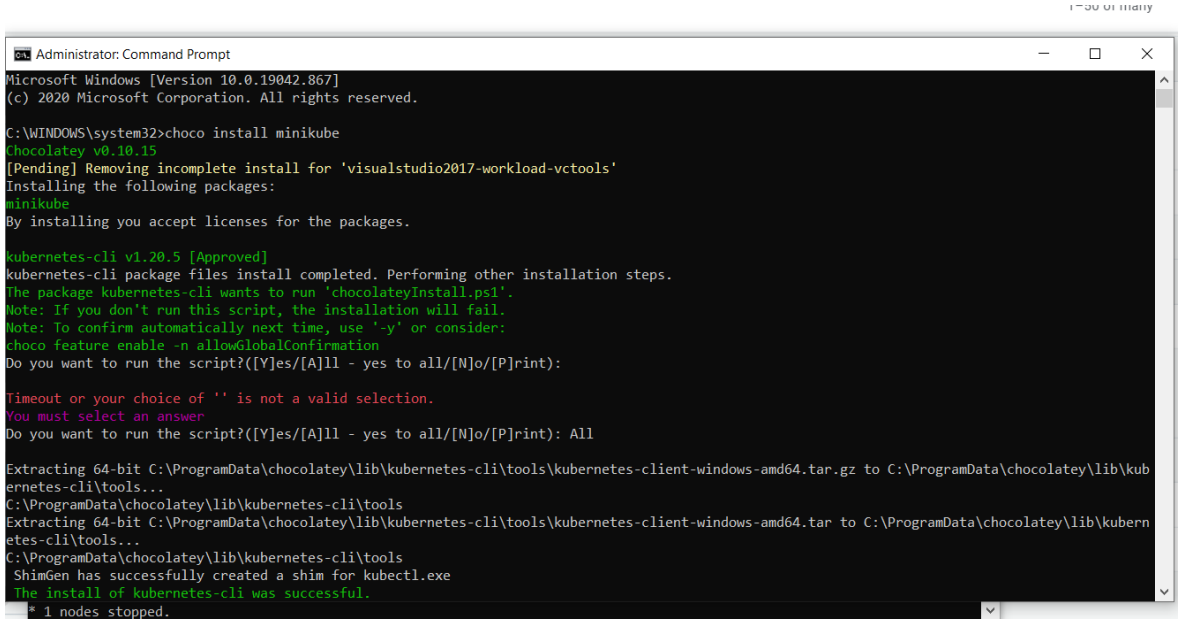
3) techworld with Nana

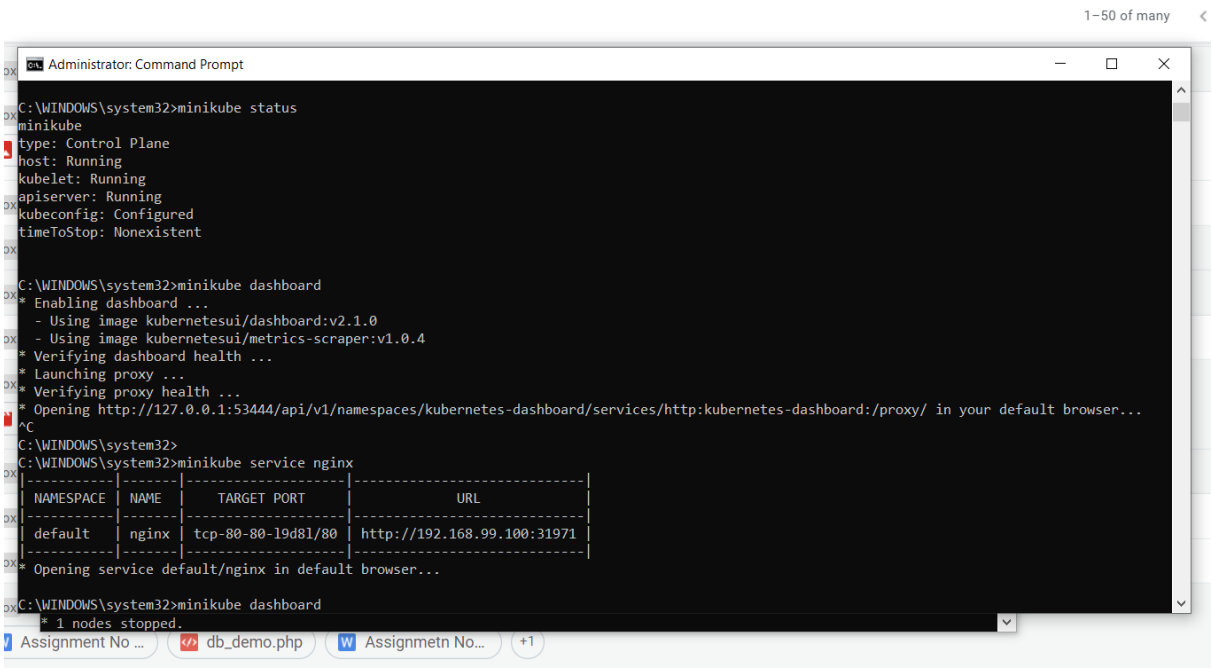
4) Kubernetes Concepts Explained in 9 minutes! - KodeKloud

5) Kubernetes Tutorial for Beginners | Kubernetes Tutorial | Intellipaat

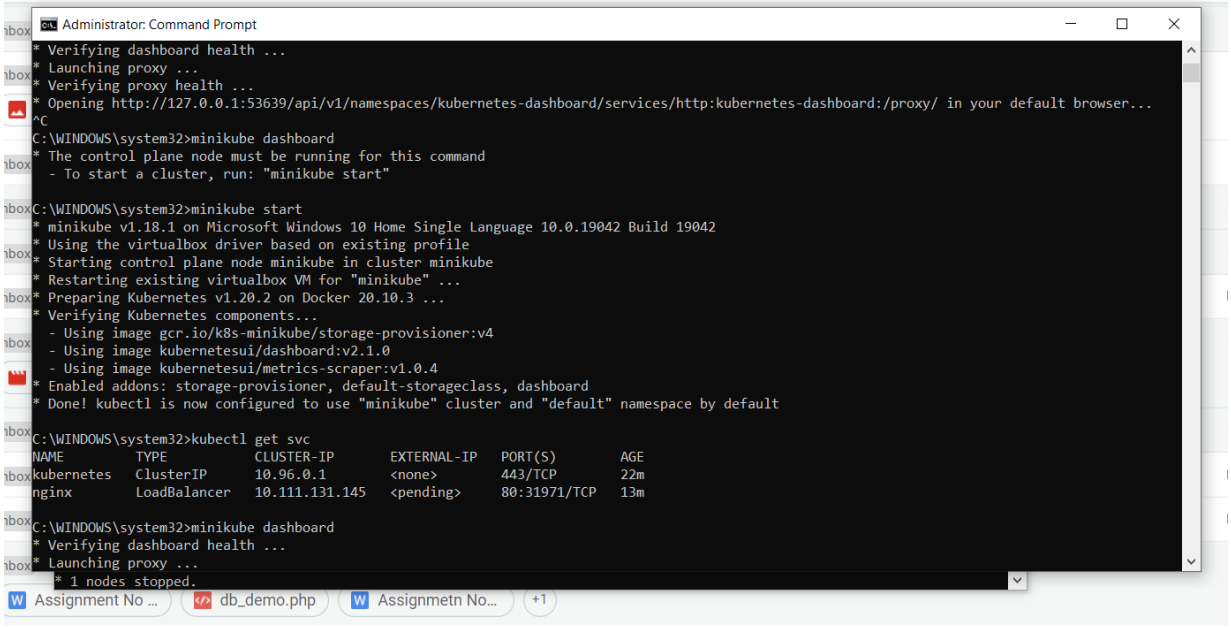
6) Install Kubernetes | Setup Kubernetes Step by Step | Kubernetes Training | Intellipaat

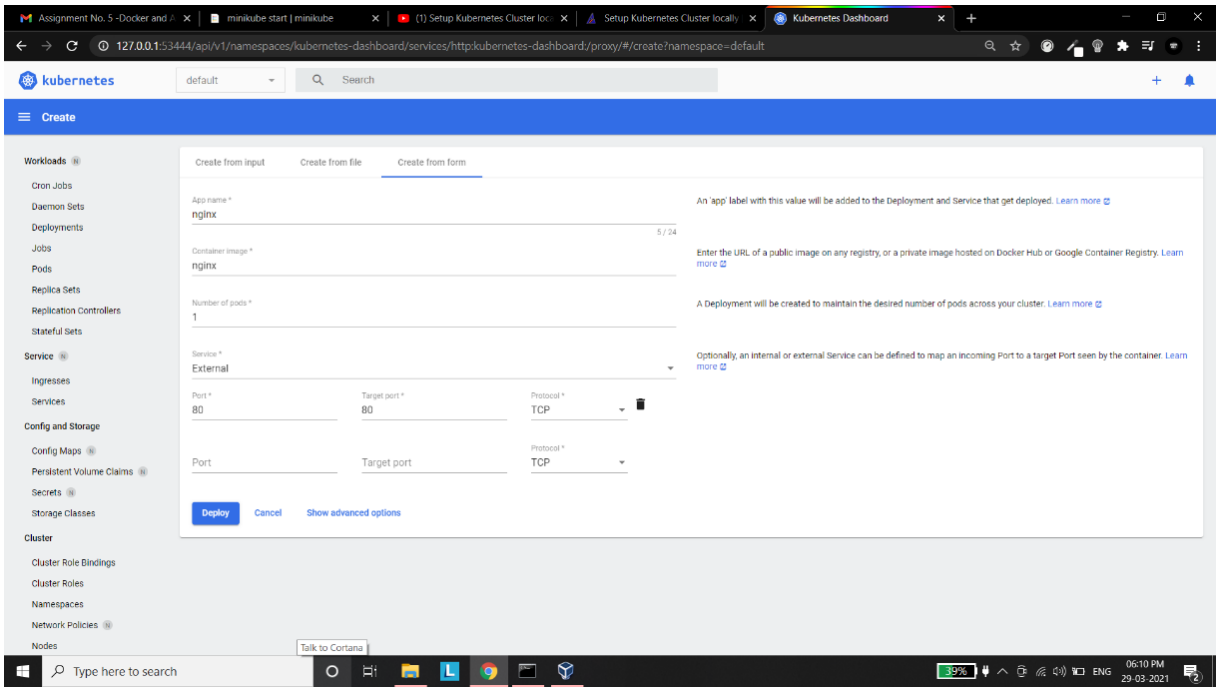
Output :

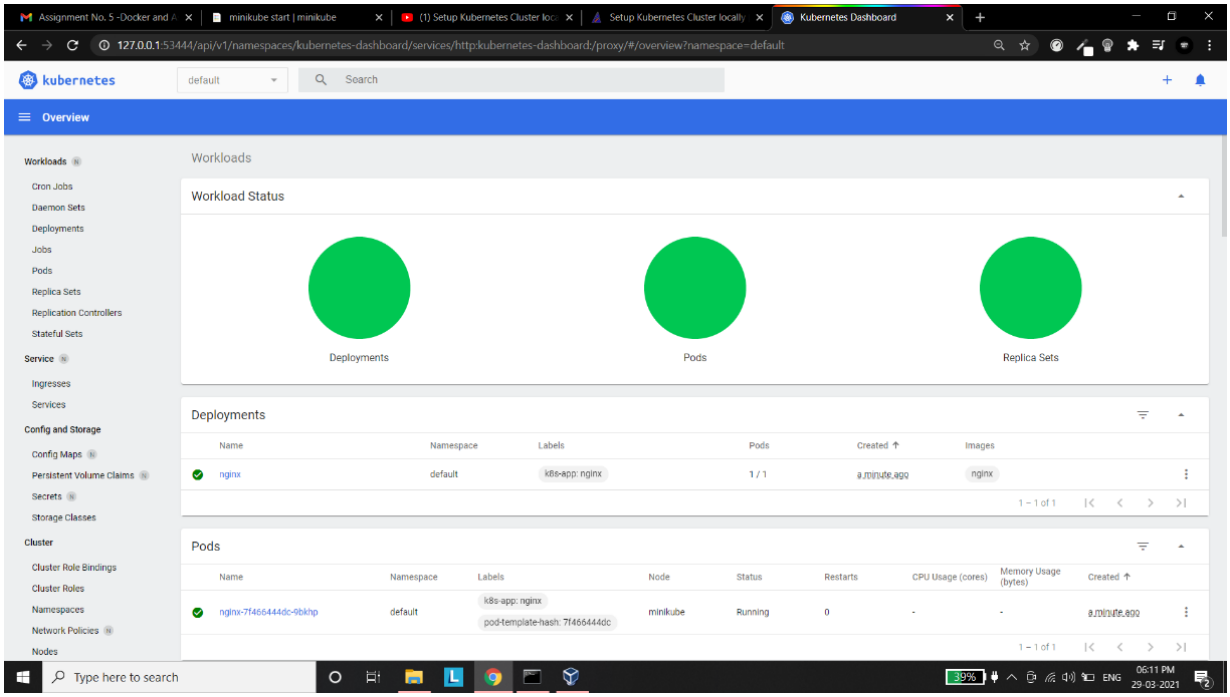

Administrator: Command Prompt
Microsoft Windows [Version 10.0.19042.867]
(c) 2020 Microsoft Corporation. All rights reserved.
C:\WINDOWS\system32>choco install minikube
Chocolatey v0.10.15
[Pending] Removing incomplete install for 'visualstudio2017-workload-vctools'
Installing the following packages:
minikube
By installing you accept licenses for the packages.
kubernetes-cli v1.20.5 [Approved]
kubernetes-cli package files install completed. Performing other installation steps.
The package kubernetes-cli wants to run 'chocolateyInstall.ps1'.
Note: If you don't run this script, the installation will fail.
Note: To confirm automatically next time, use '-y' or consider:
choco feature enable -n allowGlobalConfirmation
Do you want to run the script?([Y]es/[A]ll - yes to all/[N]o/[P]rint):
Timeout or your choice of '' is not a valid selection.
You must select an answer
Do you want to run the script?([Y]es/[A]ll - yes to all/[N]o/[P]rint): All
Extracting 64-bit C:\ProgramData\chocolatey\lib\kubernetes-cli\tools\kubernetes-client-windows-amd64.tar.gz to C:\ProgramData\chocolatey\lib\kubernetes-cli\tools...
C:\ProgramData\chocolatey\lib\kubernetes-cli\tools
Extracting 64-bit C:\ProgramData\chocolatey\lib\kubernetes-cli\tools\kubernetes-client-windows-amd64.tar to C:\ProgramData\chocolatey\lib\kubernetes-cli\tools...
C:\ProgramData\chocolatey\lib\kubernetes-cli\tools
ShimGen has successfully created a shim for kubectl.exe
The install of kubernetes-cli was successful.
* 1 nodes stopped.

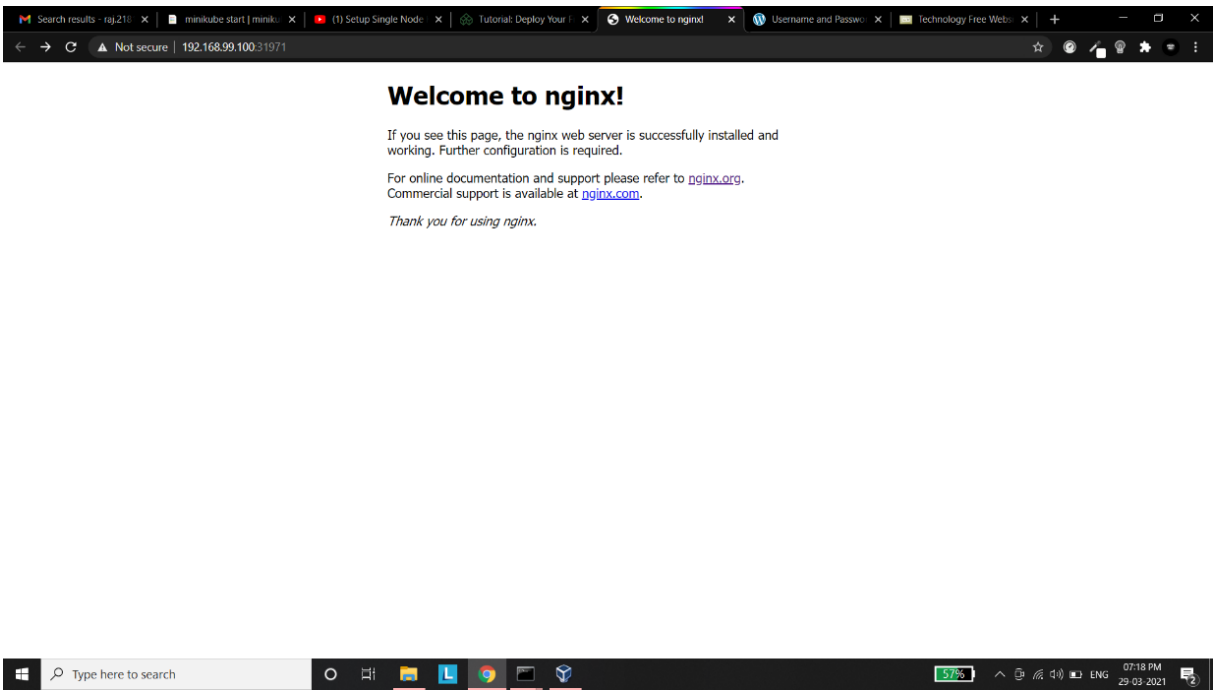

1–50 of many
Administrator: Command Prompt
C:\WINDOWS\system32>minikube status
minikube
type: Control Plane
host: Running
kubelet: Running
apiserver: Running
kubeconfig: Configured
timeToStop: Nonexistent
C:\WINDOWS\system32>minikube dashboard
* Enabling dashboard ...
- Using image kubernetesui/dashboard:v2.1.0
- Using image kubernetesui/metrics-scraper:v1.0.4
* Verifying dashboard health ...
* Launching proxy ...
* Verifying proxy health ...
* Opening http://127.0.0.1:53444/api/v1/namespaces/kubernetes-dashboard/services/http:kubernetes-dashboard:/proxy/ in your default browser...
^C
C:\WINDOWS\system32>
C:\WINDOWS\system32>minikube service nginx
| NAMESPACE | NAME | TARGET PORT | URL |
| default | nginx | tcp-80-80-19d81/80 | http://192.168.99.100:31971 |
* Opening service default/nginx in default browser...
C:\WINDOWS\system32>minikube dashboard
* 1 nodes stopped.
Assignment No ...
db_demo.php
Assignmetn No...
+1

1–50 of many


Administrator: Command Prompt
* Verifying dashboard health ...
* Launching proxy ...
* Verifying proxy health ...
* Opening http://127.0.0.1:53639/api/v1/namespaces/kubernetes-dashboard/services/http:kubernetes-dashboard:/proxy/ in your default browser...
^C
C:\WINDOWS\system32>minikube dashboard
* The control plane node must be running for this command
- To start a cluster, run: "minikube start"
C:\WINDOWS\system32>minikube start
* minikube v1.18.1 on Microsoft Windows 10 Home Single Language 10.0.19042 Build 19042
* Using the virtualbox driver based on existing profile
* Starting control plane node minikube in cluster minikube
* Restarting existing virtualbox VM for "minikube" ...
* Preparing Kubernetes v1.20.2 on Docker 20.10.3 ...
* Verifying Kubernetes components...
- Using image gcr.io/k8s-minikube/storage-provisioner:v4
- Using image kubernetesui/dashboard:v2.1.0
- Using image kubernetesui/metrics-scraper:v1.0.4
* Enabled addons: storage-provisioner, default-storageclass, dashboard
* Done! kubectl is now configured to use "minikube" cluster and "default" namespace by default
C:\WINDOWS\system32>kubectl get svc
NAME TYPE CLUSTER-IP EXTERNAL-IP PORT(S) AGE
kubernetes ClusterIP 10.96.0.1 <none> 443/TCP 22m
nginx LoadBalancer 10.111.131.145 <pending> 80:31971/TCP 13m
C:\WINDOWS\system32>minikube dashboard
* Verifying dashboard health ...
* Launching proxy ...
* 1 nodes stopped.
Assignment No ...
db_demo.php
Assignmetn No...
+1


Assignment No. 5 -Docker and
minikube start | minikube
(1) Setup Kubernetes Cluster loc
Setup Kubernetes Cluster locally
Kubernetes Dashboard
127.0.0.1:53444/api/v1/namespaces/kubernetes-dashboard/services/http:kubernetes-dashboard:/proxy/#/create?namespace=default
kubernetes
default
Search
Create
Workloads
Cron Jobs
Daemon Sets
Deployments
Jobs
Pods
Replica Sets
Replication Controllers
Stateful Sets
Service
Ingresses
Services
Config and Storage
Config Maps
Persistent Volume Claims
Secrets
Storage Classes
Cluster
Cluster Role Bindings
Cluster Roles
Namespaces
Network Policies
Nodes
Create from input
Create from file
Create from form
App name *
nginx
An 'app' label with this value will be added to the Deployment and Service that get deployed. Learn more
Container image *
nginx
Enter the URL of a public image on any registry, or a private image hosted on Docker Hub or Google Container Registry. Learn more
Number of pods *
1
A Deployment will be created to maintain the desired number of pods across your cluster. Learn more
Service *
External
Optionally, an internal or external Service can be defined to map an incoming Port to a target Port seen by the container. Learn more
Port *
80
Target port *
80
Protocol *
TCP
Port
Target port
Protocol *
TCP
Deploy
Cancel
Show advanced options
Talk to Cortana
Type here to search
39%
ENG
06:10 PM
29-03-2021


Assignment No. 5 -Docker and
minikube start | minikube
(1) Setup Kubernetes Cluster loc
Setup Kubernetes Cluster locally
Kubernetes Dashboard
127.0.0.1:53444/api/v1/namespaces/kubernetes-dashboard/services/http:kubernetes-dashboard:/proxy/#/overview?namespace=default
kubernetes
default
Search
Overview
Workloads
Cron Jobs
Daemon Sets
Deployments
Jobs
Pods
Replica Sets
Replication Controllers
Stateful Sets
Service
Ingresses
Services
Config and Storage
Config Maps
Persistent Volume Claims
Secrets
Storage Classes
Cluster
Cluster Role Bindings
Cluster Roles
Namespaces
Network Policies
Nodes
Workloads
Workload Status
Deployments
Pods
Replica Sets
Deployments
Name
Namespace
Labels
Pods
Created
Images
nginx
default
k8s-app: nginx
1 / 1
a minute ago
nginx
1 – 1 of 1
Pods
Name
Namespace
Labels
Node
Status
Restarts
CPU Usage (cores)
Memory Usage (bytes)
Created
nginx-7f466444dc-9bkhp
default
k8s-app: nginx
pod-template-hash: 7f466444dc
minikube
Running
0
-
-
a minute ago
1 – 1 of 1
Type here to search
89%
ENG
06:11 PM
29-03-2021


Search results - raj.218
minikube start | minikube
(1) Setup Single Node
Tutorial: Deploy Your
Welcome to nginx!
Username and Passwo
Technology Free Webs
Not secure | 192.168.99.100:31971
Welcome to nginx!
If you see this page, the nginx web server is successfully installed and working. Further configuration is required.
For online documentation and support please refer to nginx.org.
Commercial support is available at nginx.com.
Thank you for using nginx.
Type here to search
57%
ENG
07:18 PM
29-03-2021