
Setting Up a Virtual Machine on Google Cloud Platform (GCP)

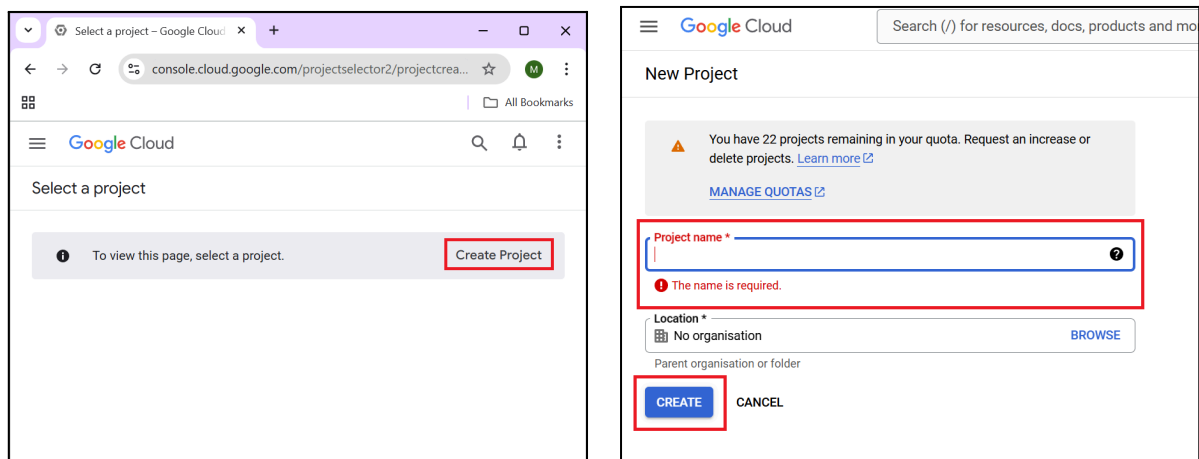
Step 1: Access the GCP Console

1. Navigate to <https://console.cloud.google.com/>.
2. Log in to your Google account and ensure your billing account is active.

Note: You are required to set up a billing account before using the Google Cloud Platform service (<https://cloud.google.com/billing/docs/how-to/manage-billing-account>)

Step 2: Create a Project

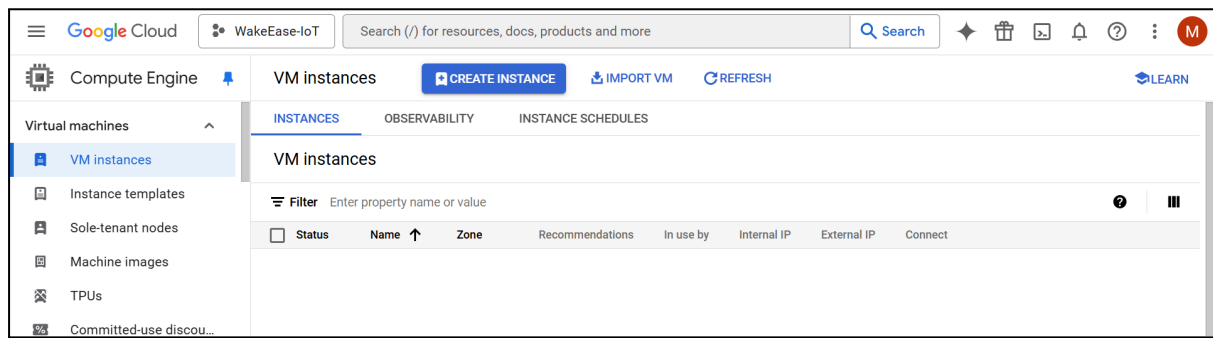
1. In the GCP console, click the **Create Project** > set the project name > click **CREATE**.



2. Click the **ENABLE** button to enable the Compute Engine API for the project.

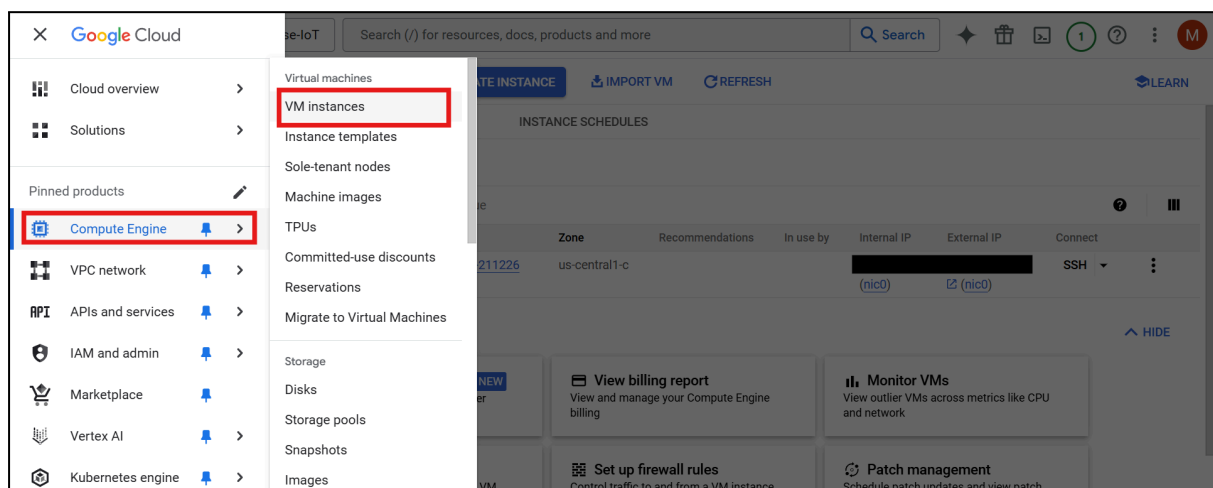


3. If the API is enabled, you will be redirected to the *VM Instances* page

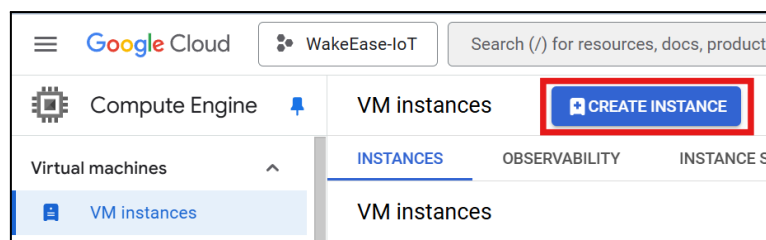


Step 3: Create a Virtual Machine (VM) Instance

1. If you are not redirected to the VM Instances page after enabling the API, manually navigate to it by selecting *Compute Engine* > *Virtual Machines* > *VM Instances* from the navigation menu.



2. Click the *Create Instance* button to start configuring a new VM.



Step 4: Configure the Virtual Machine

1. At **Machine Configuration** tab:

- Set the **Region** and **Zone** (You may use the default settings for all details here)

The screenshot shows the 'Machine configuration' tab in the Google Cloud console. The 'Name' field is set to 'instance-20250121-211226'. The 'Region' is set to 'us-central1 (Iowa)' and the 'Zone' is set to 'Any'. The 'General purpose' machine type is selected. The 'Monthly estimate' is shown as US\$25.46. The 'Operating system and storage' tab is also visible, showing 'Debian GNU/Linux 12 (bookworm)' as the image.

| Item | Monthly |
|----------------------|---------|
| 2 vCPU + 4 GB memory | |
| 10 GB balanced | |

2. At **OS and Storage** tab:

- Click the **CHANGE** button.

The screenshot shows the 'OS and storage' tab in the Google Cloud console. The 'Name' is 'instance-20250121-211226', 'Type' is 'New balanced persistent disk', 'Size' is '10 GB', 'Snapshot schedule' is 'No schedule selected', 'Licence type' is 'Free', and 'Image' is 'Debian GNU/Linux 12 (bookworm)'. The 'CHANGE' button is highlighted. The 'Monthly estimate' is shown as US\$25.46.

| Item | Monthly estimate |
|--------------------------------|------------------|
| 2 vCPU + 4 GB memory | US\$24.46 |
| 10 GB balanced persistent disk | US\$1.00 |
| Total | US\$25.46 |

- Set **Operating system** to **Ubuntu**, **version** to **Ubuntu 20.04 LTS** then click **SELECT**.

The screenshot shows the 'OS and storage' tab in the Google Cloud console. The 'Operating system' is set to 'Ubuntu' and the 'Version' is set to 'Ubuntu 20.04 LTS'. The 'Boot disk type' is set to 'Balanced persistent disk'. The 'Size (GB)' is set to '10'. The 'SELECT' button is highlighted.

| Operating system | Version | Boot disk type | Size (GB) |
|------------------|------------------|--------------------------|-----------|
| Ubuntu | Ubuntu 20.04 LTS | Balanced persistent disk | 10 |

3. At **Networking** tab:

- Under **Firewall**, check the boxes for:
 - **Allow HTTP traffic**
 - **Allow HTTPS traffic**

Google Cloud | WakeEase-IoT | Search (/) for resources, docs, products and more

Create an instance | CREATE VM FROM...

Machine configuration: e2-medium, us-central1

OS and storage: Ubuntu 20.04 LTS

Networking: 2 firewall rules, 1 network interface

Observability

Security

Advanced

Networking

Firewall ⓘ
Add tags and firewall rules to allow specific network traffic from the Internet

- ☒ Allow HTTP traffic
- ☒ Allow HTTPS traffic
- ☐ Allow load balancer health checks

Network tags

Hostname

Set a custom hostname for this instance or leave it default. Choice is permanent

IP forwarding ⓘ
☐ Enable

CREATE CANCEL EQUIVALENT CODE

Monthly estimate
US\$25.46
That's about US\$0.03 hourly
Pay for what you use: No upfront costs and per-second billing

| Item | Monthly estimate |
|--------------------------------|------------------|
| 2 vCPU + 4 GB memory | US\$24.46 |
| 10 GB balanced persistent disk | US\$1.00 |
| Total | US\$25.46 |

[Compute Engine pricing](#)
[LESS](#)

4. Lastly, click the **CREATE** button to complete the VM creation.

Observability

Security

Advanced

Network tags

Hostname

Set a custom hostname for this instance or leave it default. Choice is permanent

IP forwarding ⓘ
☐ Enable

CREATE CANCEL EQUIVALENT CODE

5. Once completed, you will see the newly created instance in the list.

Google Cloud | WakeEase-IoT | Search (/) for resources, docs, products and more

Compute Engine | VM instances | CREATE INSTANCE | IMPORT VM | REFRESH | LEARN

Virtual machines

VM instances

Instance templates

Sole-tenant nodes

Machine images

INSTANCES OBSERVABILITY INSTANCE SCHEDULES

VM instances

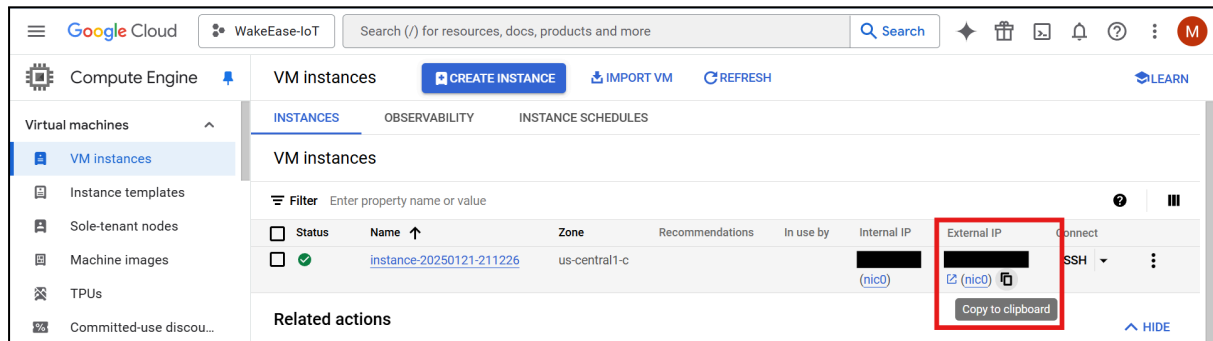
Filter Enter property name or value

| Status | Name ↑ | Zone | Recommendations | In use by | Internal IP | External IP | Connect |
|-------------------------------------|--|---------------|-----------------|-----------|-------------|-------------|---------|
| <input checked="" type="checkbox"/> | instance-20250121-211226 | us-central1-c | | | | | SSH |

Step 5: Note the External IP Address

1. At the VM instance list, locate the **External IP Address** of your VM.

Note: The external IP address changes whenever the VM instance is restarted.



2. Copy the external IP address and update it in the following files:

- **WakeEase.ino** (line 11)

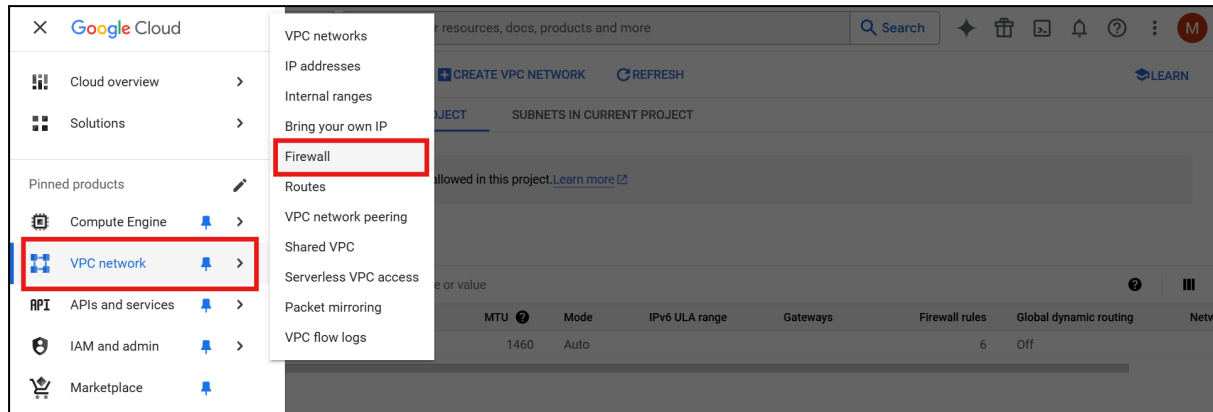
```
WakeEase.ino
1  #include <PubSubClient.h>
2  #include <WiFi.h>
3  #include <time.h>
4  #include <ESP32Servo.h>
5
6  // Wi-Fi Credentials
7  const char* ssid = "yourwifi"; // Replace with your Wi-Fi SSID
8  const char* password = "yourwifipassword"; // Replace with your Wi-Fi password
9
10 // MQTT Credentials
11 const char* MQTT_SERVER = "yourExternalIP"; // Replace with your VM External IP
12 const int MQTT_PORT = 1883; // Non-TLS communication port
13 WiFiClient espClient;
14 PubSubClient client(espClient);
```

- **WakeEase_script.py** (line 11)

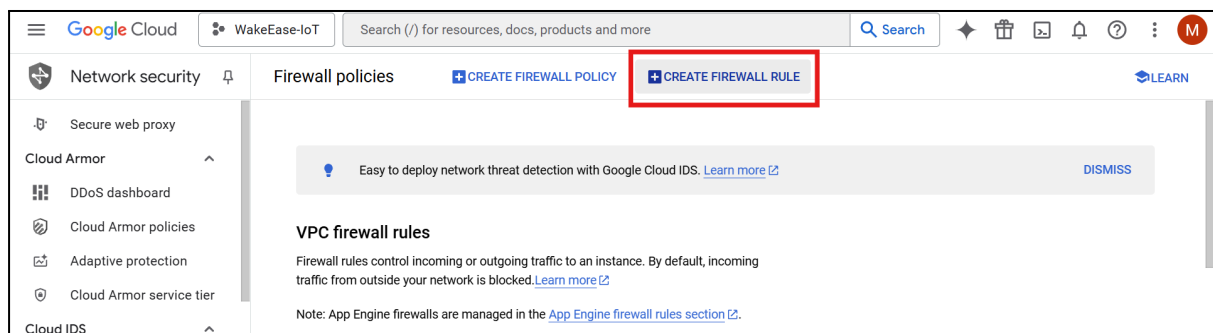
```
wakeEase_script.py > [?] mqtt_broker_address
1  import pymongo
2  import paho.mqtt.client as mqtt
3  from datetime import datetime
4  import pytz # To handle timezone conversions
5
6  # MongoDB configuration
7  mongo_client = pymongo.MongoClient("mongodb://localhost:27017/")
8  db = mongo_client["WakeEase"]
9
10 # MQTT configuration
11 mqtt_broker_address = "yourExternalIP" # Replace with your VM instance external IP address
12 mqtt_topics = [("notification", 0), ("duration", 0), ("response_time", 0)] # List of topics
```

Step 6: Configure a VPC Firewall Rule

1. Navigate to *VPC Network > Firewall* in the navigation menu.



2. Click the *CREATE FIREWALL RULE* button at the top.



3. Configure the firewall rule as follows:

- **Name:** allow-mqtt-1883
- **Description:** Allow traffic on port 1883.
- **Logs:** Off.
- **Network:** default.
- **Priority:** 1000.
- **Direction of Traffic:** Ingress.
- **Action on Match:** Allow.
- **Targets:** All instances in the network.
- **Source Filter:** IPv4 ranges.
- **Source IPv4 Ranges:** 0.0.0.0/0.
- **Second Source Filter:** None.
- **Destination Filter:** None.
- **Protocols and Ports:** TCP port 1883.

Google Cloud WakeEase-IoT Search (/) for resources, docs, products and more

Network security > Create a firewall rule

Firewall rules control incoming or outgoing traffic to an instance. By default, incoming traffic from outside your network is blocked. [Learn more](#)

Name *
allow-mqtt-1883
Lowercase letters, numbers, hyphens allowed

Description
Allow traffic on port 1883

Protocols
Separate multiple protocols by commas, e.g. AH, ICMP
tcp:1883

4. Click the **CREATE** button to save the firewall rule.

Cloud IDS
IDS dashboard
IDS endpoints
IDS threats

Protocols
Separate multiple protocols by commas, e.g. AH, ICMP
tcp:1883

DISABLE RULE
CREATE CANCEL

5. Ensure the new VPC firewall rule appears in the list.

Google Cloud WakeEase-IoT Search (/) for resources, docs, products and more

Network security > Firewall policies

CREATE FIREWALL POLICY CREATE FIREWALL RULE

| Name | Type | Targets | Filters | Protocols/ports | Action | Priority | Network |
|------------------------|---------|--------------|------------|------------------------------------|--------|----------|---------|
| allow-mqtt-1883 | Ingress | Apply to all | IP ranges: | tcp:1883 | Allow | 1000 | default |
| default-allow-http | Ingress | http-server | IP ranges: | tcp:80 | Allow | 1000 | default |
| default-allow-https | Ingress | https- | IP ranges: | tcp:443 | Allow | 1000 | default |
| default-allow-icmp | Ingress | Apply to all | IP ranges: | icmp | Allow | 65534 | default |
| default-allow-internal | Ingress | Apply to all | IP ranges: | tcp:0-65535 udp:0-65535 icmp | Allow | 65534 | default |
| default-allow-rdp | Ingress | Apply to all | IP ranges: | tcp:3389 | Allow | 65534 | default |
| default-allow-ssh | Ingress | Apply to all | IP ranges: | tcp:22 | Allow | 65534 | default |

6. Your VM instance is now fully configured and ready for the next steps.

***Note:** Setting the source IP range to 0.0.0.0/0 allows access from anywhere and is not recommended for production environments.*

Step 7: Access the VM via SSH

1. Click the **SSH** button next to your VM instance to launch the SSH-in-browser.

