

Cloud Monitoring: Qwik Start

Task 1: Create a Compute Engine instance

- **Instance Name:** lamp-1-vm
- **Region:** <REGION> (replace with your lab's assigned region)
- **Zone:** <ZONE>
- **Series:** E2
- **Machine type:** e2-medium
- **Boot Disk:** Debian GNU/Linux 12 (bookworm)
- **Firewall:** ✓ Allow HTTP traffic
- **Verify:** Green check in progress checker.

The screenshot shows the Google Cloud Platform dashboard for project "qwiklabs-gcp-01-0280f2bbea12". The dashboard includes sections for Project info, API APIs (Requests (requests/sec) chart), Google Cloud Platform status (All services normal), and Billing (Estimated charges USD \$0.00 for May 1 – 20, 2025). A "Go to project settings" link is also present.

The screenshot shows the Google Cloud Platform dashboard with a terminal window titled "CLOUD SHELL Terminal (qwiklabs-gcp-01-0280f2bbea12)" open. The terminal displays a welcome message and command history related to Cloud Shell setup.

```
Welcome to Cloud Shell! Type "help" to get started.  
Your Cloud Platform project in this session is set to qwiklabs-gcp-01-0280f2bbea12.  
Use `gcloud config set project [PROJECT ID]` to change to a different project.  
student_01_46e4ab78c8ad@cloudshell:~ (qwiklabs-gcp-01-0280f2bbea12)$
```



```
Welcome to Cloud Shell! Type "help" to get started.  
Your Cloud Platform project in this session is set to qwiklabs-gcp-01-0280f2bbea12.  
Use 'gcloud config set project [PROJECT_ID]' to change to a different project.  
student_01_46e4ab78c8ad@cloudshell:~ (qwiklabs-gcp-01-0280f2bbea12)$ gcloud config set compute/zone "europe-west1-c"  
export ZONE=$(gcloud config get compute/zone)  
  
gcloud config set compute/region "europe-west1"  
export REGION=$ (gcloud config get compute/region)  
Updated property [compute/region].  
Your active configuration is: [cloudshell-13525]  
Updated property [compute/region].  
Your active configuration is: [cloudshell-13525]  
student_01_46e4ab78c8ad@cloudshell:~ (qwiklabs-gcp-01-0280f2bbea12)$
```

VM instances [Create instance](#) [Import VM](#) [Refresh](#) [Learn](#)

Instances Observability Instance schedules

VM instances

Filter Enter property name or value [?](#) [☰](#)

<input type="checkbox"/> Status	Name ↑	Zone	Recommendations	In use by	Internal IP	External IP	Connect
<input checked="" type="checkbox"/>	lamp-1-vm	europe-west1-c			10.132.0.2 (nic0)	34.38.37.14 (nic0)	SSH ▼ ⋮

Related actions

[Explore protection summary](#) [New](#)
Identify gaps in data protection at no

[Monitor VMs](#)
View outlier VMs across metrics like CPU and network

[Explore VM logs](#)
View, search, analyze, and download VM instance logs

Task 2: Install Apache2 and PHP

SSH into your VM and run:

```
bash
sudo apt-get update
sudo apt-get install apache2 php7.0
```

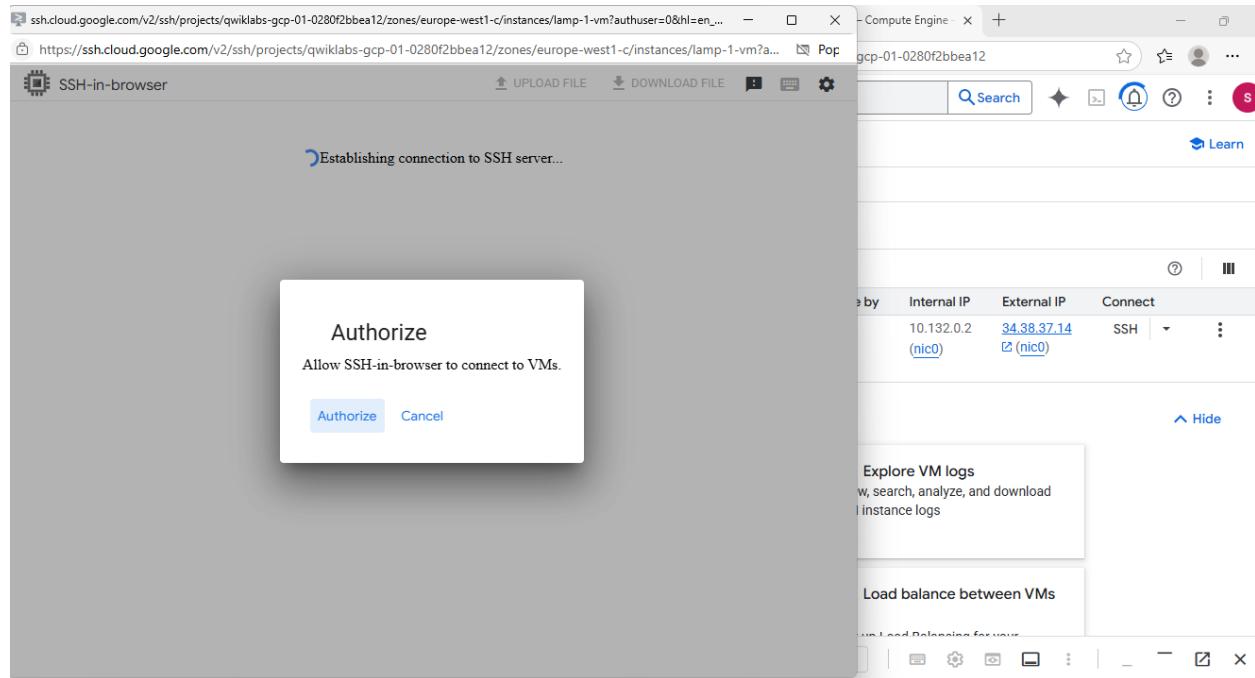
If php7.0 fails:

```
bash  
sudo apt-get install php5
```

Then:

```
bash  
sudo service apache2 restart
```

- Open your **External IP** in a browser to check the **Apache2 default page**.
- Enable the External IP column if it's hidden.



```
*****
Get:1 https://download.docker.com/linux/ubuntu noble InRelease [48.8 kB]
Get:2 https://cli.github.com/packages stable InRelease [3,917 B]
Get:3 https://packages.cloud.google.com/apt/gcsfuse-noble InRelease [1,227 B]
Get:4 https://packages.cloud.google.com/apt/cloud-sdk InRelease [1,618 B]
Get:5 https://cli.github.com/packages/stable/main amd64 Packages [345 B]
Get:6 https://apt.postgresql.org/pub/repos/apt/noble-pgdg InRelease [129 kB]
Get:7 http://security.ubuntu.com/ubuntu noble-security InRelease [126 kB]
Hit:8 http://archive.ubuntu.com/ubuntu noble InRelease
Get:9 http://archive.ubuntu.com/ubuntu noble-updates InRelease [126 kB]
Get:10 https://packages.cloud.google.com/apt/cloud-sdk/main amd64 Packages [3,813 kB]
Get:11 https://apt.postgresql.org/pub/repos/apt/noble-pgdg/main amd64 Packages [528 kB]
Get:12 https://packages.cloud.google.com/apt/cloud-sdk/main all Packages [1,713 kB]
Get:13 https://ppa.launchpadcontent.net/dotnet/backports/ubuntu noble InRelease [24.1 kB]
Get:14 http://security.ubuntu.com/ubuntu noble-security/main amd64 Packages [1,077 kB]
Get:15 http://archive.ubuntu.com/ubuntu noble-backports InRelease [126 kB]
Get:16 https://ppa.launchpadcontent.net/dotnet/backports/ubuntu noble/main amd64 Packages [6,491 B]
Get:17 http://archive.ubuntu.com/ubuntu noble-updates/universe amd64 Packages [1,382 kB]
Get:18 http://security.ubuntu.com/ubuntu noble-security/universe amd64 Packages [1,085 kB]
Get:19 http://security.ubuntu.com/ubuntu noble-security/restricted amd64 Packages [1,421 kB]
Get:20 http://archive.ubuntu.com/ubuntu noble-updates/main amd64 Packages [1,396 kB]
Get:21 http://archive.ubuntu.com/ubuntu noble-updates/restricted amd64 Packages [1,465 kB]
Get:22 http://archive.ubuntu.com/ubuntu noble-backports/main amd64 Packages [48.0 kB]
Get:23 http://archive.ubuntu.com/ubuntu noble-backports/universe amd64 Packages [31.8 kB]
Fetched 14.6 MB in 3s (5,368 kB/s)
Reading package lists... Done
```

```

student_01 46e4ab78c8ad@cloudshell:~ (qwiklabs-gcp-01-0280f2bbea12)$ sudo apt-get install apache2 php7.0
*****
You are running apt-get inside of Cloud Shell. Note that your Cloud Shell
machine is ephemeral and no system-wide change will persist beyond session end.

To suppress this warning, create an empty ~/.cloudshell/no-apt-get-warning file.
The command will automatically proceed in 5 seconds or on any key.

Visit https://cloud.google.com/shell/help for more information.
*****

```

The browser window shows the Apache2 Debian Default Page. It features a red header bar with the text "It works!". Below it, there is descriptive text about the page's purpose and configuration. A "Configuration Overview" section details the layout of configuration files in /etc/apache2/. At the bottom, there is a code block showing the directory structure of /etc/apache2/.

Monitoring Setup

1. Create a Monitoring Metrics Scope

- a. Go to: \equiv > **Observability** > **Monitoring** \rightarrow it sets up automatically.

2. Install Monitoring & Logging Agents (in SSH):

```

bash
curl -sSO https://dl.google.com/cloudagents/add-google-cloud-ops-agent-repo.sh
sudo bash add-google-cloud-ops-agent-repo.sh --also-install
sudo systemctl status google-cloud-ops-agent "*"
# (Press `q` to exit)
sudo apt-get update

```

Task 3: Create an Uptime Check

- \equiv > **Monitoring** > **Uptime checks** > **Create**

- **Protocol:** HTTP
- **Resource Type:** Instance
- **Instance:** lamp-1-vm
- **Frequency:** 1 minute
- **Title:** Lamp Uptime Check
- Test → Create

Wait until green checks appear after restart.

The screenshot shows two views of the Google Cloud Observability Monitoring interface.

Top View (Overview Page):

- Left Sidebar:** Includes sections for Overview, Dashboards, Application monitoring, Explore (Metrics explorer, Logs explorer, Log analytics, Trace explorer), Detect (Alerting, Observability Scopes, Release Notes), and Metrics explorer.
- Right Content Area:**
 - Alerts:** 0 (View alerts)
 - Services with SLO out of budget:** 0 (View services)
 - Failed uptime checks:** 0 (View checks)
 - VMs with Ops Agent:** 1 of 1 (No legacy agents. View VMs)
- Bottom Buttons:** Predefined (Last 1 hour, IST), Search, and Customize.

Bottom View (Uptime checks Page):

- Left Sidebar:** Same as the Overview page.
- Right Content Area:**
 - Uptime checks:** + Create uptime check
 - Try the new Synthetic monitoring capability! (New):** Proactively monitor your application's critical user journeys by simulating user behavior. Detect and alert on failures, and receive detailed reports for debugging. [Learn more.](#)
 - Create Synthetic Monitor** button and [View Documentation](#) link.
 - Table:**

Display Name	Asia Pacific	Europe	North America	South America	Policies
Lamp Uptime Check	✓	✓	✓	✓	0

Task 4: Create an Alerting Policy

- \equiv > **Monitoring** > **Alerting** > +Create Policy
- Select Metric:
 - Uncheck “Active”
 - Filter: Network traffic
 - Select: VM instance > Interface > Network traffic
- Set:
 - **Threshold:** Above 500
 - **Retest Window:** 1 min
- Add **Notification Channel:**
 - Email → add, confirm, refresh
- Set Alert Name: **Inbound Traffic Alert**
- Add optional documentation/message
- Create Policy

Task 5: Create Dashboard & Charts

\equiv > **Monitoring** > **Dashboards** > +Create Custom Dashboard

- Name: Cloud Monitoring LAMP Qwik Start Dashboard

Add Widget 1:

- Line chart
- Title: CPU Load
- Metric: VM Instance > Cpu > CPU load (1m)

Add Widget 2:

- Line chart
- Title: Received Packets
- Metric: VM Instance > Instance > Received Packets

Task 6: View Logs

☰ > Logging > Logs Explorer

- Filter by:
 - **Resource:** VM Instance
 - **Instance:** lamp-1-vm

Test logs by:

1. Opening Compute Engine in new tab
2. Stopping the VM → watch Logs
3. Starting the VM again → watch startup logs

Task 7: Review Uptime & Alerts

- ☰ > Monitoring > **Uptime Checks**
 - Look for **Lamp Uptime Check** status
 - Reload until it turns green after VM restart
- ☰ > Monitoring > **Alerting**
 - Check for incidents/events
- Check your **email** for alert notification from Cloud Monitoring.