Deploying Node.js on Google Cloud Platform(GCP)

A Step-by-Step Guide

Lidia Bereketeab

Feb 7, 2024

Table of Content

Introduction

Objective:

- Demonstrate the process of setting up a Google Cloud Platform (GCP) account for a free trial.
- Deploy a Node.js server on Ubuntu hosted on GCP.
- Create an HTTP JSON API server in Node.js to retrieve the current time using JSON.

Overview:

- Step-by-step guide on setting up a GCP account for a free trial.
- Explanation of deploying a Node.js server on Ubuntu hosted on GCP.
- Demonstration of creating an HTTP JSON API server in Node.js for retrieving the current time using JSON.

Design

Identification of Needs:

- Recognizing the necessity to create a GCP account to access cloud services and resources.

Importance of Project Creation:

- Understanding the significance of project creation to efficiently organize and manage cloud resources.

Steps Involved in Deployment:

- Investigation of the steps involved in deploying a Node.js server on GCP, including setting up VM instances and configuring servers.

Design

Theoretical Comparison:

- Theoretical comparison of various cloud platforms to assess their suitability for hosting Node.js applications.

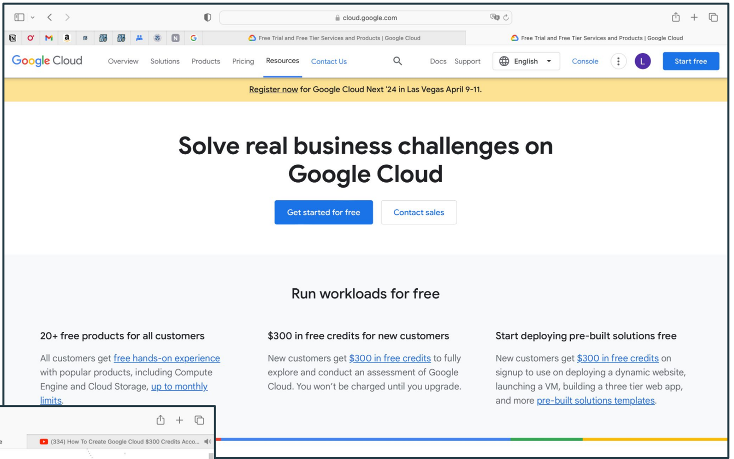
Selection of GCP:

- Selection of Google Cloud Platform (GCP) as the preferred platform based on its features, user-friendly interface, and comprehensive documentation.

Implementation - Setup

Setting up GCP free Trial Account

- Go to the following link: https://cloud.google.com/free?hl=en.
- Click on "Get started for free".
- Fill in your account information, including your Gmail address and location.


cloud.google.com
Free Trial and Free Tier Services and Products | Google Cloud
Free Trial and Free Tier Services and Products | Google Cloud
Google Cloud
Overview
Solutions
Products
Pricing
Resources
Contact Us
Docs
Support
English
Console
Start free

Register now for Google Cloud Next '24 in Las Vegas April 9-11.

Solve real business challenges on Google Cloud

Run workloads for free

20+ free products for all customers

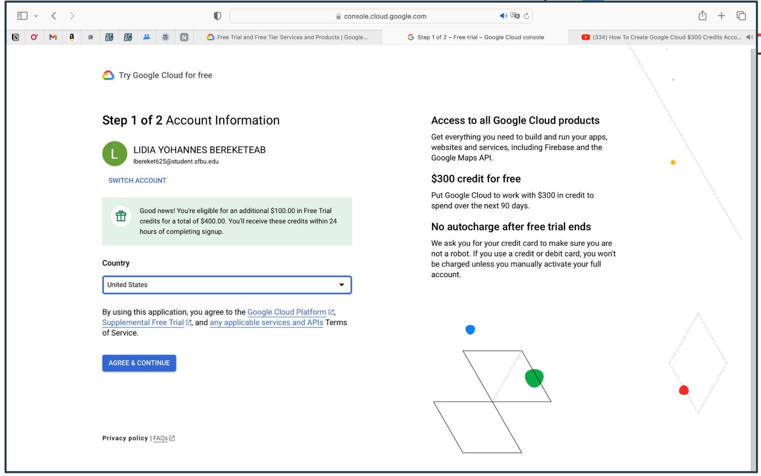
All customers get free hands-on experience with popular products, including Compute Engine and Cloud Storage, up to monthly limits.

$300 in free credits for new customers

New customers get $300 in free credits to fully explore and conduct an assessment of Google Cloud. You won't be charged until you upgrade.

Start deploying pre-built solutions free

New customers get $300 in free credits on signup to use on deploying a dynamic website, launching a VM, building a three tier web app, and more pre-built solutions templates.


console.cloud.google.com
Free Trial and Free Tier Services and Products | Google...
Step 1 of 2 – Free trial – Google Cloud console
(334) How To Create Google Cloud $300 Credits Acco...
Try Google Cloud for free

Step 1 of 2 Account Information

LIDIA YOHANNES BEREKETEAB
lbereket625@student.sfbu.edu

Good news! You're eligible for an additional $100.00 in Free Trial credits for a total of $400.00. You'll receive these credits within 24 hours of completing signup.

Country: United States

By using this application, you agree to the Google Cloud Platform, Supplemental Free Trial, and any applicable services and APIs Terms of Service.

Access to all Google Cloud products

Get everything you need to build and run your apps, websites and services, including Firebase and the Google Maps API.

$300 credit for free

Put Google Cloud to work with $300 in credit to spend over the next 90 days.

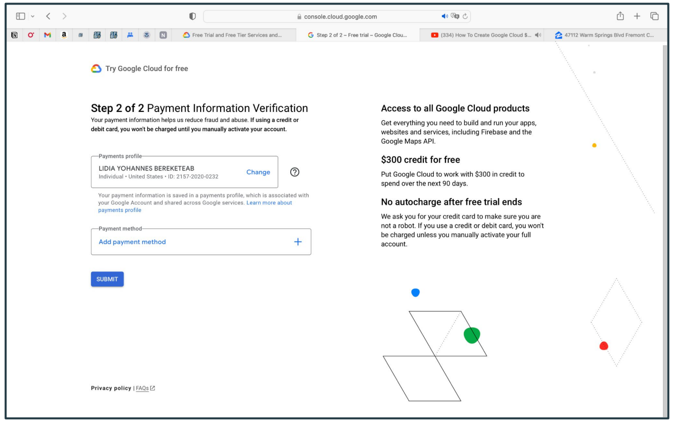
No autocharge after free trial ends

We ask you for your credit card to make sure you are not a robot. If you use a credit or debit card, you won't be charged unless you manually activate your full account.

Implementation: Setup

Setting up GCP free Trial Account

- Provide a payment method for verification purposes
- Answer some general questions to tailor the cloud experience to your desired solutions.
- Complete the final settings on the interface of the platform
- finish setting up your GCP account for the free trial.



console.cloud.google.com

Try Google Cloud for free

Step 2 of 2 Payment Information Verification

Your payment information helps us reduce fraud and abuse. **If using a credit or debit card, you won't be charged until you manually activate your account.**

Payments profile: LIDIA YOHANNES BEREKETEAB — Individual • United States • ID: 2157-2020-0232 — Change

Your payment information is saved in a payments profile, which is associated with your Google Account and shared across Google services. Learn more about payments profile

Payment method: Add payment method

SUBMIT

Privacy policy | FAQs

Access to all Google Cloud products

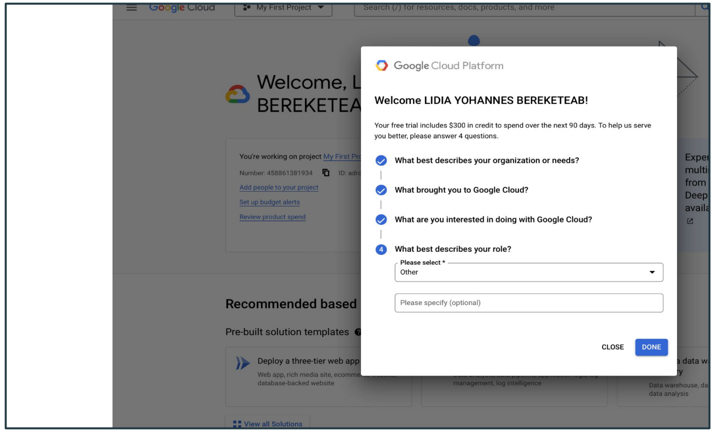
Get everything you need to build and run your apps, websites and services, including Firebase and the Google Maps API.

$300 credit for free

Put Google Cloud to work with $300 in credit to spend over the next 90 days.

No autocharge after free trial ends

We ask you for your credit card to make sure you are not a robot. If you use a credit or debit card, you won't be charged unless you manually activate your full account.



Google Cloud Platform

Welcome LIDIA YOHANNES BEREKETEAB!

Your free trial includes $300 in credit to spend over the next 90 days. To help us serve you better, please answer 4 questions.

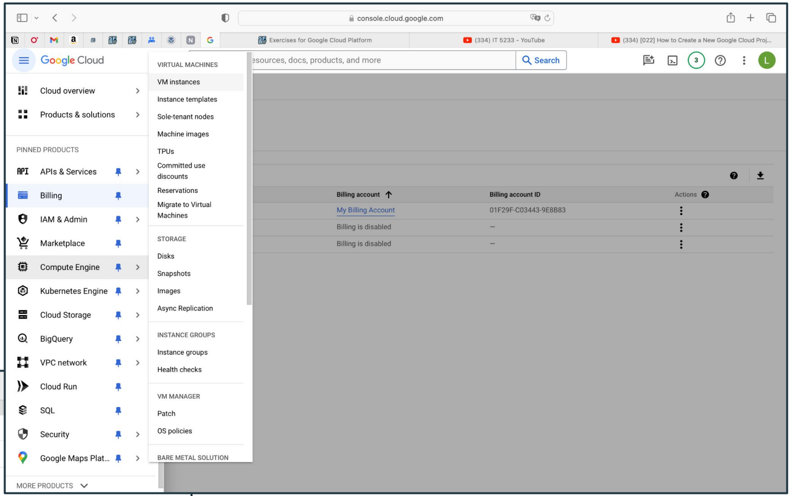
- ✓ What best describes your organization or needs?
- ✓ What brought you to Google Cloud?
- ✓ What are you interested in doing with Google Cloud?
- 4 What best describes your role?
  - Please select *: Other
  - Please specify (optional)

CLOSE | DONE

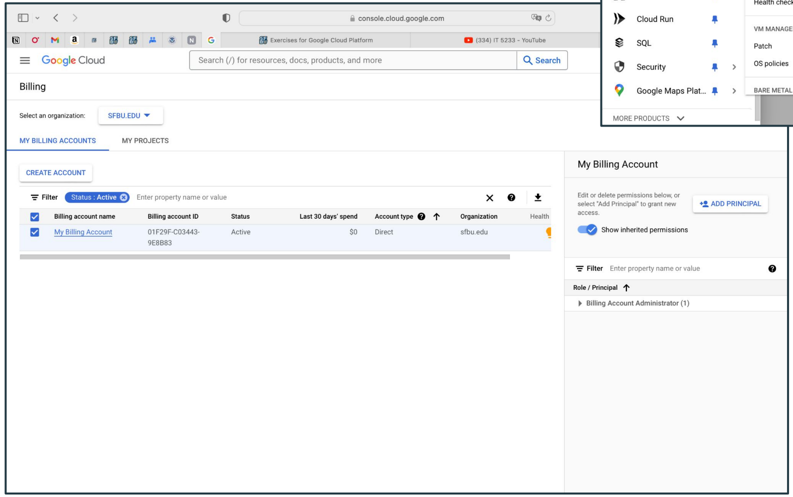
Implementation : Setup

Create Project in GCP

- Choose the billing account to which the project will be billed under.
- Choose your preferred service for the project. Example: "Compute Engine" -> "VM Instances" service



Billing account	Billing account ID	Actions
My Billing Account	01F29F-C03443-9E8B83	
Billing is disabled	–	
Billing is disabled	–	



Billing account name	Billing account ID	Status	Last 30 days' spend	Account type	Organization	Health
My Billing Account	01F29F-C03443-9E8B83	Active	$0	Direct	sfbu.edu	

My Billing Account

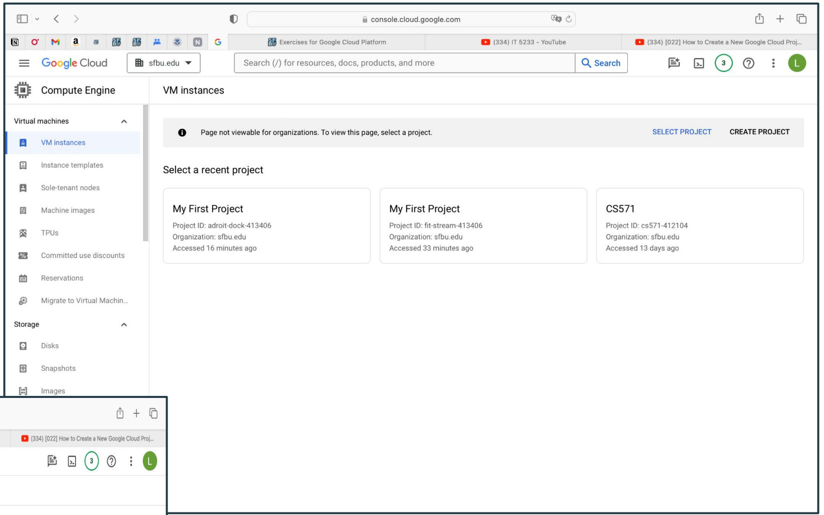
Edit or delete permissions below, or select "Add Principal" to grant new access.

Billing Account Administrator (1)

Implementation: Setup

Create Project in GCP

- Either choose an already created project “Select Project” or click on "create project".
- Name the project and finish the setup.



VM instances

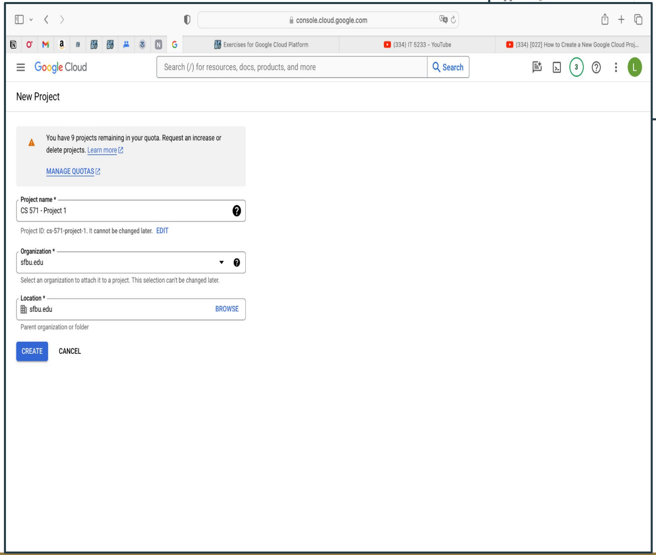
Page not viewable for organizations. To view this page, select a project.

Select a recent project

My First Project
Project ID: adroit-dock-413406
Organization: sfbu.edu
Accessed 16 minutes ago

My First Project
Project ID: fit-stream-413406
Organization: sfbu.edu
Accessed 33 minutes ago

CS571
Project ID: cs571-412104
Organization: sfbu.edu
Accessed 13 days ago


console.cloud.google.com
Exercises for Google Cloud Platform
(334) IT 5233 - YouTube
(334) [022] How to Create a New Google Cloud Proj...
Google Cloud
Search (/) for resources, docs, products, and more
Search
MANAGE QUOTAS
EDIT
BROWSE
CREATE
CANCEL

New Project

You have 9 projects remaining in your quota. Request an increase or delete projects. Learn more

Project name *
CS 571 - Project 1

Project ID: cs-571-project-1. It cannot be changed later.

Organization *
sfbu.edu

Select an organization to attach it to a project. This selection can't be changed later.

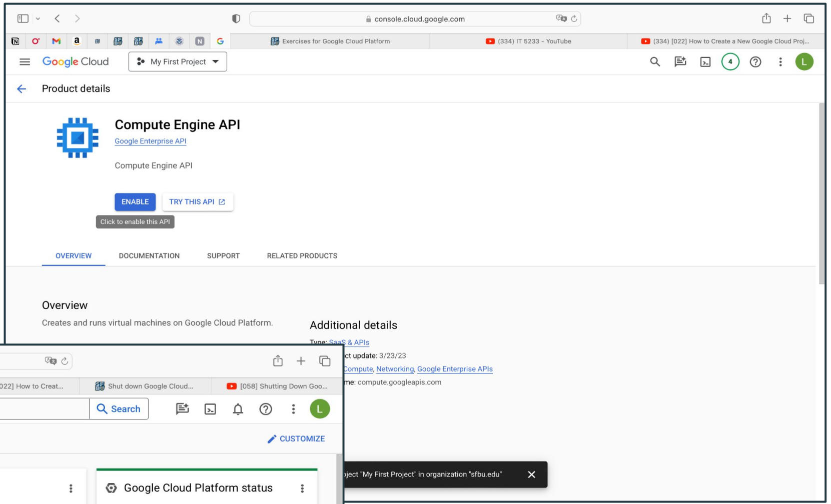
Location *
sfbu.edu

Parent organization or folder

Implementation: Setup

Create Project in GCP

- Enable the Engine API.
- Review the project.



console.cloud.google.com

Exercises for Google Cloud Platform | (334) IT 5233 - YouTube | (334) [022] How to Create a New Google Cloud Proj...

Google Cloud | My First Project

Product details

Compute Engine API

Google Enterprise API

Compute Engine API

ENABLE | TRY THIS API

Click to enable this API

OVERVIEW | DOCUMENTATION | SUPPORT | RELATED PRODUCTS

Overview

Creates and runs virtual machines on Google Cloud Platform.

Additional details

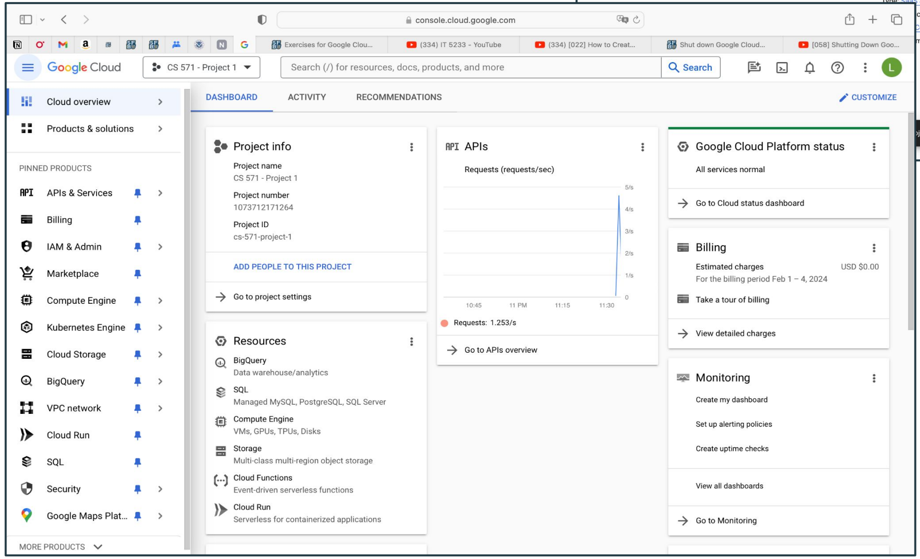
Type: SaaS & APIs

ct update: 3/23/23

Compute, Networking, Google Enterprise APIs

me: compute.googleapis.com

oject "My First Project" in organization "sfbu.edu"



console.cloud.google.com

Exercises for Google Clou... | (334) IT 5233 - YouTube | (334) [022] How to Creat... | Shut down Google Cloud... | [058] Shutting Down Goo...

Google Cloud | CS 571 - Project 1 | Search (/) for resources, docs, products, and more | Search

Project info

Project name
CS 571 - Project 1

Project number
1073712171264

Project ID
cs-571-project-1

ADD PEOPLE TO THIS PROJECT

Go to project settings

APIs

Requests (requests/sec)

Requests: 1.253/s

Go to APIs overview

Google Cloud Platform status

All services normal

Go to Cloud status dashboard

Billing

Estimated charges USD $0.00

For the billing period Feb 1 – 4, 2024

Take a tour of billing

View detailed charges

Resources

BigQuery
Data warehouse/analytics

SQL
Managed MySQL, PostgreSQL, SQL Server

Compute Engine
VMs, GPUs, TPUs, Disks

Storage
Multi-class multi-region object storage

Cloud Functions
Event-driven serverless functions

Cloud Run
Serverless for containerized applications

Monitoring

Create my dashboard

Set up alerting policies

Create uptime checks

View all dashboards

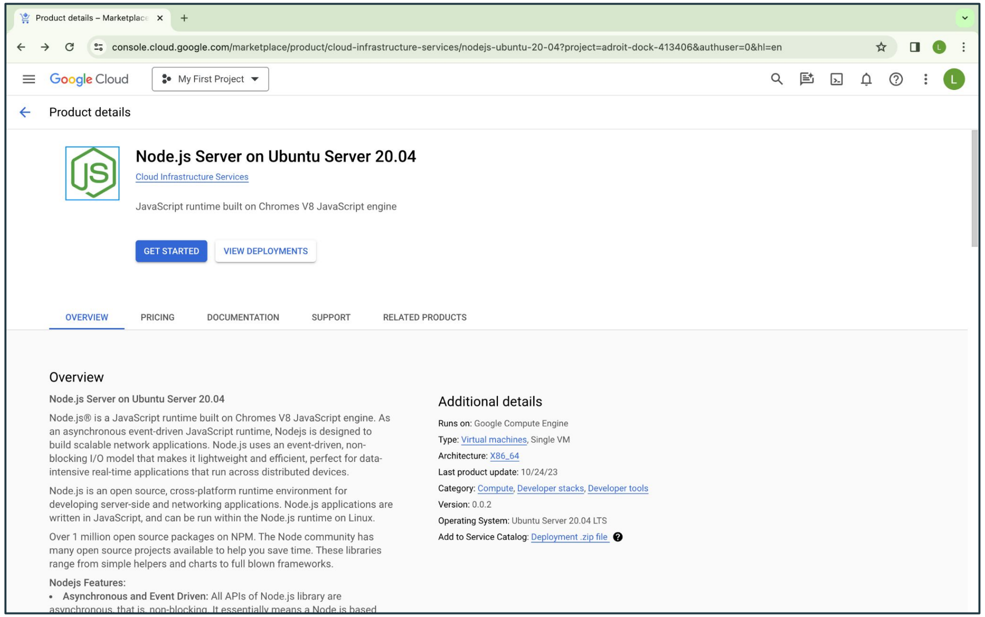
Go to Monitoring

Implementation

Setup Node.js server on Ubuntu in GCP

Accessing Google Cloud Platform Marketplace

- Navigate to the Google Cloud Platform Marketplace : https://console.cloud.google.com/marketplace/product/cloud-infrastructure-services/nodejs-ubuntu-20-04
- Access the listing for Ubuntu with pre-installed Node.js.



console.cloud.google.com/marketplace/product/cloud-infrastructure-services/nodejs-ubuntu-20-04?project=adroit-dock-413406&authuser=0&hl=en

Google Cloud

Product details

Node.js Server on Ubuntu Server 20.04

Cloud Infrastructure Services

JavaScript runtime built on Chromes V8 JavaScript engine

Overview

Node.js Server on Ubuntu Server 20.04

Node.js® is a JavaScript runtime built on Chromes V8 JavaScript engine. As an asynchronous event-driven JavaScript runtime, Nodejs is designed to build scalable network applications. Node.js uses an event-driven, non-blocking I/O model that makes it lightweight and efficient, perfect for data-intensive real-time applications that run across distributed devices.

Node.js is an open source, cross-platform runtime environment for developing server-side and networking applications. Node.js applications are written in JavaScript, and can be run within the Node.js runtime on Linux.

Over 1 million open source packages on NPM. The Node community has many open source projects available to help you save time. These libraries range from simple helpers and charts to full blown frameworks.

Nodejs Features:

- **Asynchronous and Event Driven:** All APIs of Node.js library are asynchronous, that is, non-blocking. It essentially means a Node.js based

Additional details

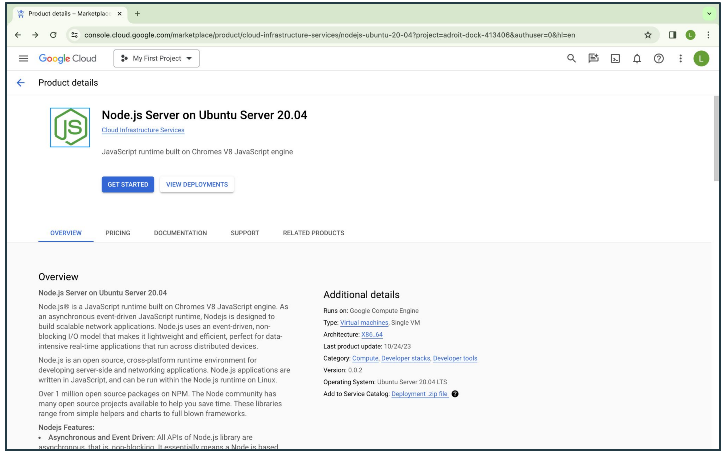
Runs on: Google Compute Engine
Type: Virtual machines, Single VM
Architecture: X86_64
Last product update: 10/24/23
Category: Compute, Developer stacks, Developer tools
Version: 0.0.2
Operating System: Ubuntu Server 20.04 LTS
Add to Service Catalog: Deployment .zip file

Implementation

Setup Node.js server on Ubuntu in GCP

Launching Node Js server on Ubuntu in GCP

- Click on “Get started “ or “Launch”
- Customize the virtual machine by providing a deployment name, selecting the region or zone, choosing the machine type, and specifying disk space.



Product details – Marketplace
console.cloud.google.com/marketplace/product/cloud-infrastructure-services/nodejs-ubuntu-20-04?project=adroit-dock-413406&authuser=0&hl=en
Google Cloud
My First Project
GET STARTED
VIEW DEPLOYMENTS
OVERVIEW
PRICING
DOCUMENTATION
SUPPORT
RELATED PRODUCTS

Product details

Node.js Server on Ubuntu Server 20.04

Cloud Infrastructure Services

JavaScript runtime built on Chromes V8 JavaScript engine

Overview

Node.js Server on Ubuntu Server 20.04

Node.js® is a JavaScript runtime built on Chromes V8 JavaScript engine. As an asynchronous event-driven JavaScript runtime, Nodejs is designed to build scalable network applications. Node.js uses an event-driven, non-blocking I/O model that makes it lightweight and efficient, perfect for data-intensive real-time applications that run across distributed devices.

Node.js is an open source, cross-platform runtime environment for developing server-side and networking applications. Node.js applications are written in JavaScript, and can be run within the Node.js runtime on Linux.

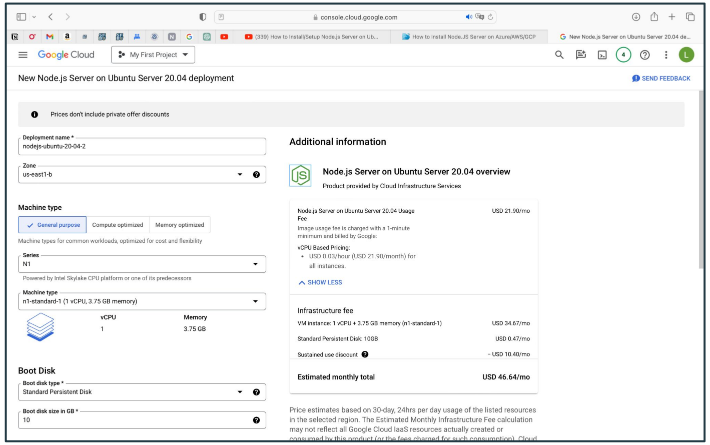
Over 1 million open source packages on NPM. The Node community has many open source projects available to help you save time. These libraries range from simple helpers and charts to full blown frameworks.

Nodejs Features:

- **Asynchronous and Event Driven:** All APIs of Node.js library are asynchronous, that is, non-blocking. It essentially means a Node.js based

Additional details

Runs on: Google Compute Engine
Type: Virtual machines, Single VM
Architecture: X86_64
Last product update: 10/24/23
Category: Compute, Developer stacks, Developer tools
Version: 0.0.2
Operating System: Ubuntu Server 20.04 LTS
Add to Service Catalog: Deployment .zip file



console.cloud.google.com
(339) How to Install/Setup Node.js Server on Ub...
How to Install Node.JS Server on Azure/AWS/GCP
New Node.js Server on Ubuntu Server 20.04 de...
Google Cloud
My First Project
SEND FEEDBACK
General purpose
Compute optimized
Memory optimized
SHOW LESS

New Node.js Server on Ubuntu Server 20.04 deployment

Prices don't include private offer discounts

Deployment name *: nodejs-ubuntu-20-04-2

Zone: us-east1-b

Machine type

Machine types for common workloads, optimized for cost and flexibility

Series: N1

Powered by Intel Skylake CPU platform or one of its predecessors

Machine type: n1-standard-1 (1 vCPU, 3.75 GB memory)

vCPU	Memory
1	3.75 GB

Boot Disk

Boot disk type *: Standard Persistent Disk

Boot disk size in GB *: 10

Additional information

Node.js Server on Ubuntu Server 20.04 overview

Product provided by Cloud Infrastructure Services

Node.js Server on Ubuntu Server 20.04 Usage Fee	USD 21.90/mo

Image usage fee is charged with a 1-minute minimum and billed by Google:

vCPU Based Pricing:

- USD 0.03/hour (USD 21.90/month) for all instances.

Infrastructure fee

VM instance: 1 vCPU + 3.75 GB memory (n1-standard-1)	USD 34.67/mo
Standard Persistent Disk: 10GB	USD 0.47/mo
Sustained use discount	– USD 10.40/mo
Estimated monthly total	**USD 46.64/mo**

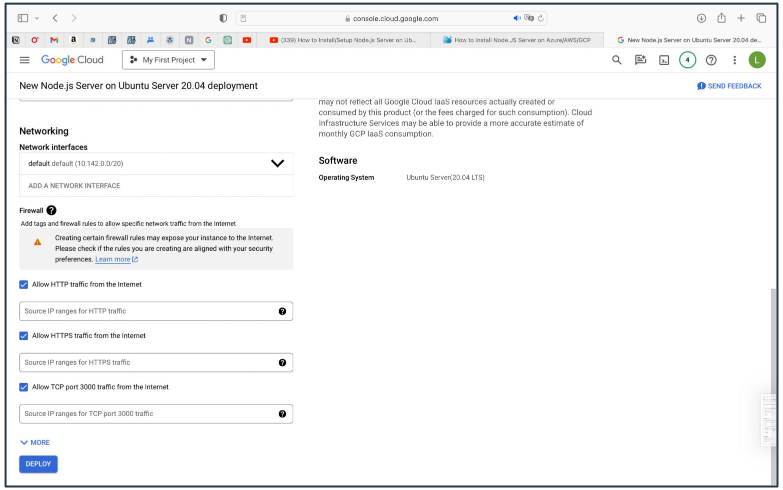
Price estimates based on 30-day, 24hrs per day usage of the listed resources in the selected region. The Estimated Monthly Infrastructure Fee calculation may not reflect all Google Cloud IaaS resources actually created or consumed by this product (or the fees charged for such consumption). Cloud

Implementation

Setup Node.js server on Ubuntu in GCP

Setup Network and Deploy

- Leave the network settings as default.
- Accept the agreement and click on "Deploy" to start the deployment process.
- Waiting for Deployment Completion


console.cloud.google.com
(339) How to Install/Setup Node.js Server on Ub...
How to Install Node.JS Server on Azure/AWS/GCP
New Node.js Server on Ubuntu Server 20.04 de...
Google Cloud
My First Project
SEND FEEDBACK

New Node.js Server on Ubuntu Server 20.04 deployment

may not reflect all Google Cloud IaaS resources actually created or consumed by this product (or the fees charged for such consumption). Cloud Infrastructure Services may be able to provide a more accurate estimate of monthly GCP IaaS consumption.

Software

Operating System: Ubuntu Server(20.04 LTS)

Networking

Network interfaces

default default (10.142.0.0/20)

ADD A NETWORK INTERFACE

Firewall

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

Creating certain firewall rules may expose your instance to the Internet. Please check if the rules you are creating are aligned with your security preferences. Learn more

- [x] Allow HTTP traffic from the Internet

Source IP ranges for HTTP traffic

- [x] Allow HTTPS traffic from the Internet

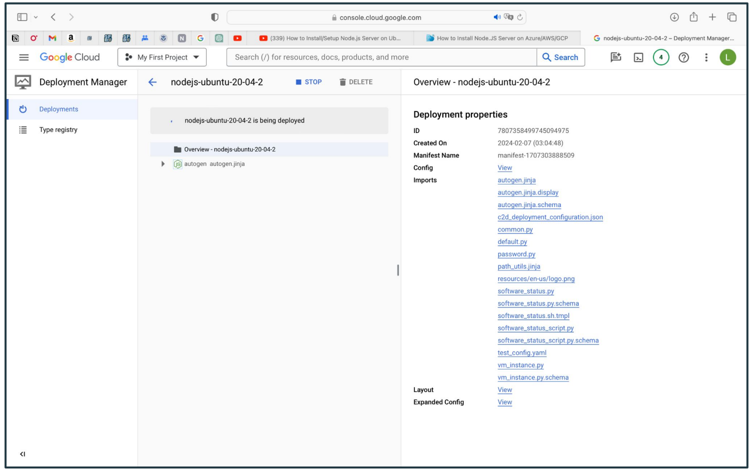
Source IP ranges for HTTPS traffic

- [x] Allow TCP port 3000 traffic from the Internet

Source IP ranges for TCP port 3000 traffic

MORE

DEPLOY


console.cloud.google.com
(339) How to Install/Setup Node.js Server on Ub...
How to Install Node.JS Server on Azure/AWS/GCP
nodejs-ubuntu-20-04-2 – Deployment Manager...
Google Cloud
My First Project
Search (/) for resources, docs, products, and more
Search
Deployment Manager
Deployments
Type registry
STOP
DELETE

nodejs-ubuntu-20-04-2

nodejs-ubuntu-20-04-2 is being deployed

Overview - nodejs-ubuntu-20-04-2

autogen autogen.jinja

Overview - nodejs-ubuntu-20-04-2

Deployment properties

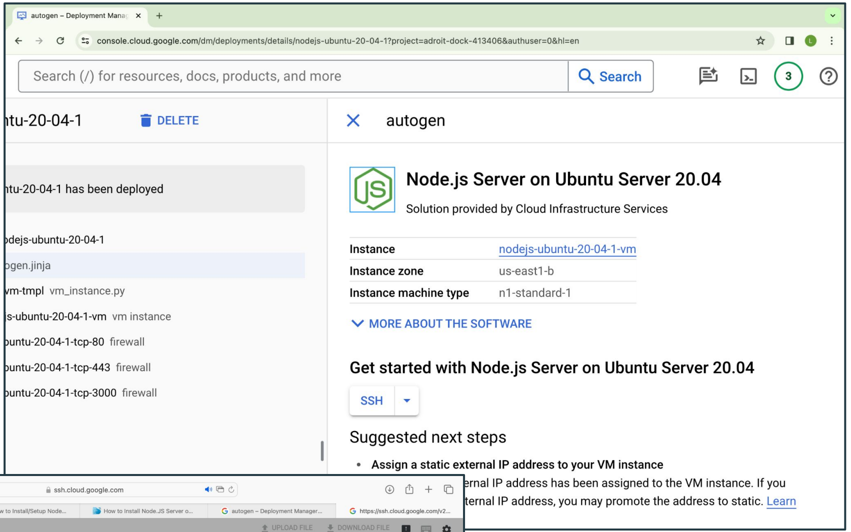
ID	7807358499745094975
Created On	2024-02-07 (03:04:48)
Manifest Name	manifest-1707303888509
Config	View
Imports	autogen.jinja autogen.jinja.display autogen.jinja.schema c2d_deployment_configuration.json common.py default.py password.py path_utils.jinja resources/en-us/logo.png software_status.py software_status.py.schema software_status.sh.tmpl software_status_script.py software_status_script.py.schema test_config.yaml vm_instance.py vm_instance.py.schema
Layout	View
Expanded Config	View

Implementation

Setup Node.js server on Ubuntu in GCP

Opening SSH Terminal

- Once deployment is complete, open the SSH terminal for the virtual machine.
- Authorize for SSH in-browser to connect with VMs



autogen – Deployment Manag

console.cloud.google.com/dm/deployments/details/nodejs-ubuntu-20-04-1?project=adroit-dock-413406&authuser=0&hl=en

Search (/) for resources, docs, products, and more | Search

ntu-20-04-1 | DELETE

ntu-20-04-1 has been deployed

odejs-ubuntu-20-04-1
ogen.jinja
vm-tmpl vm_instance.py
s-ubuntu-20-04-1-vm vm instance
buntu-20-04-1-tcp-80 firewall
buntu-20-04-1-tcp-443 firewall
buntu-20-04-1-tcp-3000 firewall

autogen

Node.js Server on Ubuntu Server 20.04

Solution provided by Cloud Infrastructure Services

Instance	nodejs-ubuntu-20-04-1-vm
Instance zone	us-east1-b
Instance machine type	n1-standard-1

MORE ABOUT THE SOFTWARE

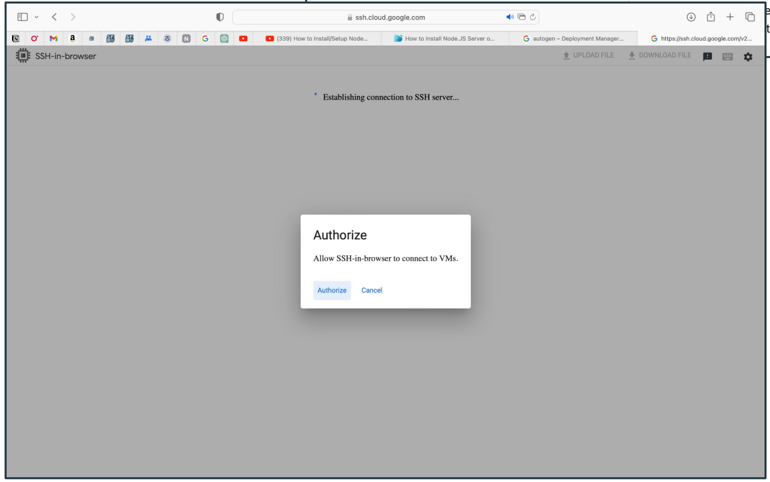
Get started with Node.js Server on Ubuntu Server 20.04

SSH

Suggested next steps

- Assign a static external IP address to your VM instance

ernal IP address has been assigned to the VM instance. If you ternal IP address, you may promote the address to static. Learn



ssh.cloud.google.com

(339) How to Install/Setup Node... | How to Install Node.JS Server o... | autogen – Deployment Manager... | https://ssh.cloud.google.com/v2...

SSH-in-browser | UPLOAD FILE | DOWNLOAD FILE

Establishing connection to SSH server...

Authorize

Allow SSH-in-browser to connect to VMs.

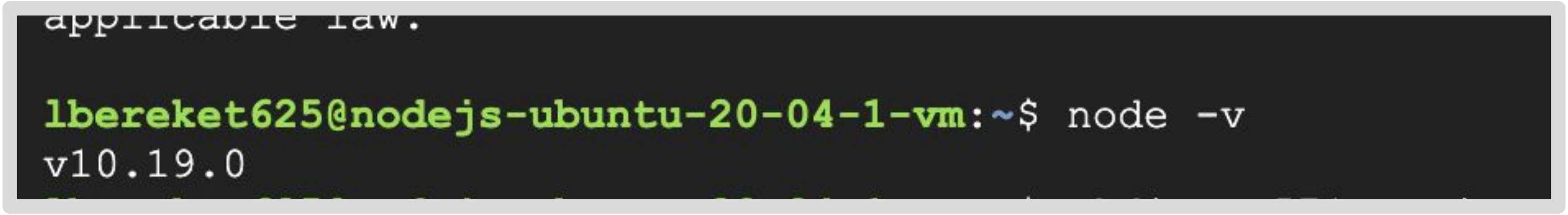
Authorize | Cancel

Implementation

Setup Node.js server on Ubuntu in GCP

Verifying Node.js Installation

Verify the Node.js installation by running the command `node -v` in the SSH terminal.



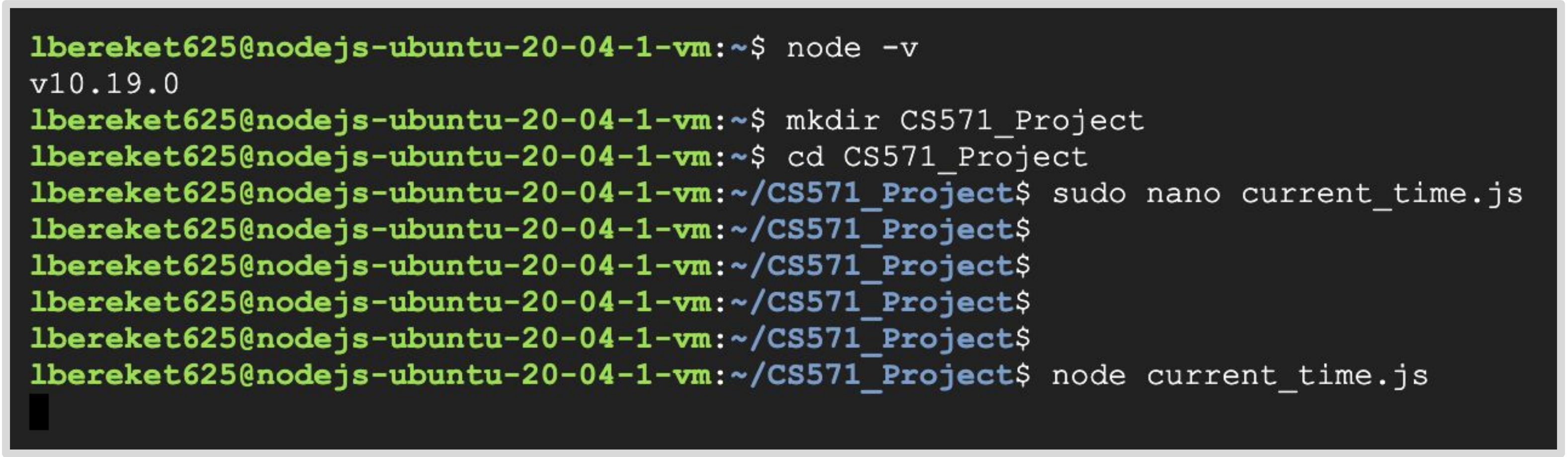
```
applicable law.

lbereket625@nodejs-ubuntu-20-04-1-vm:~$ node -v
v10.19.0
```

Test

Create an HTTP JSON API server in Node.js to retrieve the current time using JSON.

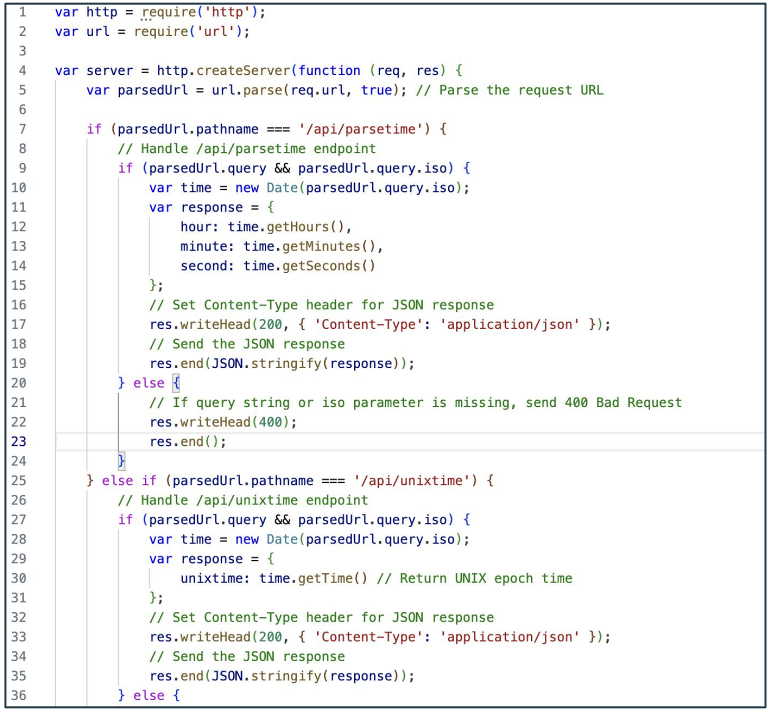
Creating Node.js File in SSH: Steps to Generate current_time.js

- Create a directory/folder by typing mkdir CS571_Project in the SSH
- Change path to the folder - cd CS571_project
- Create nodejs file using a text editor - current_time.js - sudo nano current_time.js

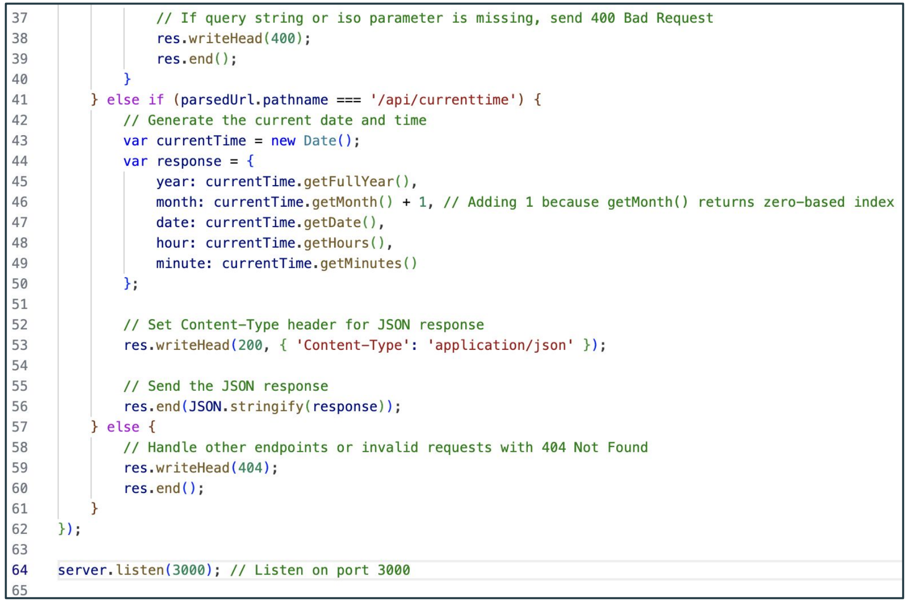


```
lbereket625@nodejs-ubuntu-20-04-1-vm:~$ node -v
v10.19.0
lbereket625@nodejs-ubuntu-20-04-1-vm:~$ mkdir CS571_Project
lbereket625@nodejs-ubuntu-20-04-1-vm:~$ cd CS571_Project
lbereket625@nodejs-ubuntu-20-04-1-vm:~/CS571_Project$ sudo nano current_time.js
lbereket625@nodejs-ubuntu-20-04-1-vm:~/CS571_Project$
lbereket625@nodejs-ubuntu-20-04-1-vm:~/CS571_Project$
lbereket625@nodejs-ubuntu-20-04-1-vm:~/CS571_Project$
lbereket625@nodejs-ubuntu-20-04-1-vm:~/CS571_Project$
lbereket625@nodejs-ubuntu-20-04-1-vm:~/CS571_Project$ node current_time.js
```

Test: current_time.js Code



```
var http = require('http');
var url = require('url');

var server = http.createServer(function (req, res) {
    var parsedUrl = url.parse(req.url, true); // Parse the request URL

    if (parsedUrl.pathname === '/api/parsetime') {
        // Handle /api/parsetime endpoint
        if (parsedUrl.query && parsedUrl.query.iso) {
            var time = new Date(parsedUrl.query.iso);
            var response = {
                hour: time.getHours(),
                minute: time.getMinutes(),
                second: time.getSeconds()
            };
            // Set Content-Type header for JSON response
            res.writeHead(200, { 'Content-Type': 'application/json' });
            // Send the JSON response
            res.end(JSON.stringify(response));
        } else {
            // If query string or iso parameter is missing, send 400 Bad Request
            res.writeHead(400);
            res.end();
        }
    } else if (parsedUrl.pathname === '/api/unixtime') {
        // Handle /api/unixtime endpoint
        if (parsedUrl.query && parsedUrl.query.iso) {
            var time = new Date(parsedUrl.query.iso);
            var response = {
                unixtime: time.getTime() // Return UNIX epoch time
            };
            // Set Content-Type header for JSON response
            res.writeHead(200, { 'Content-Type': 'application/json' });
            // Send the JSON response
            res.end(JSON.stringify(response));
        } else {
```

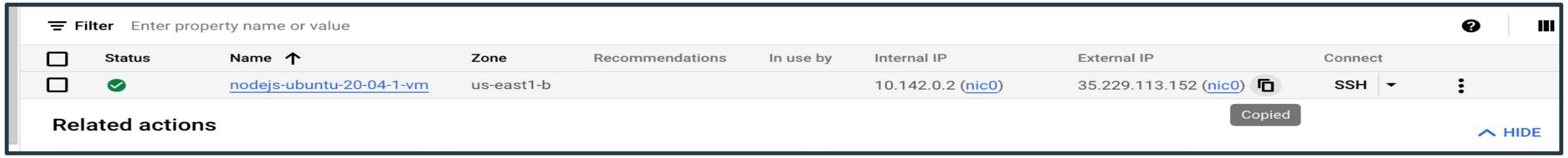


```
            // If query string or iso parameter is missing, send 400 Bad Request
            res.writeHead(400);
            res.end();
        }
    } else if (parsedUrl.pathname === '/api/currenttime') {
        // Generate the current date and time
        var currentTime = new Date();
        var response = {
            year: currentTime.getFullYear(),
            month: currentTime.getMonth() + 1, // Adding 1 because getMonth() returns zero-based index
            date: currentTime.getDate(),
            hour: currentTime.getHours(),
            minute: currentTime.getMinutes()
        };

        // Set Content-Type header for JSON response
        res.writeHead(200, { 'Content-Type': 'application/json' });

        // Send the JSON response
        res.end(JSON.stringify(response));
    } else {
        // Handle other endpoints or invalid requests with 404 Not Found
        res.writeHead(404);
        res.end();
    }
});

server.listen(3000); // Listen on port 3000

```

Test: Create an HTTP JSON API server in Node.js to retrieve the current time using JSON.

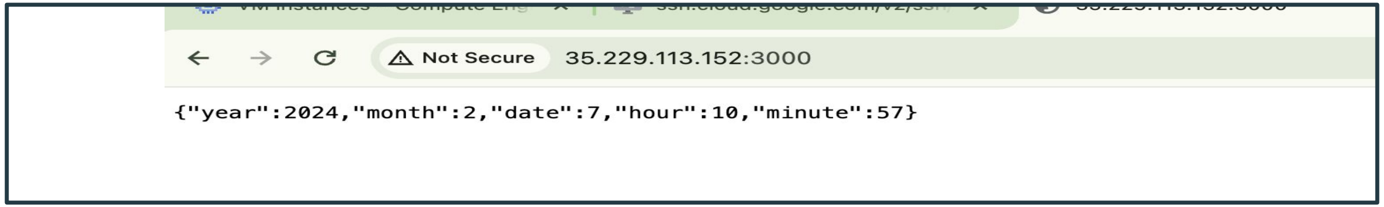
Writing Node.js Server Code for Current Time Display: Steps and Shortcuts

- Execute `node current_time.js` in SSH.
- Server starts on port 3000.
- Copy external IP from GCP.
- Append IP with ":3000" in browser.
- Hit Enter to access server.
- View response in browser window.
- Stop server with Ctrl + C in terminal.



Status	Name ↑	Zone	Recommendations	In use by	Internal IP	External IP	Connect
✓	nodejs-ubuntu-20-04-1-vm	us-east1-b			10.142.0.2 (nic0)	35.229.113.152 (nic0)	SSH

OUTPUT:

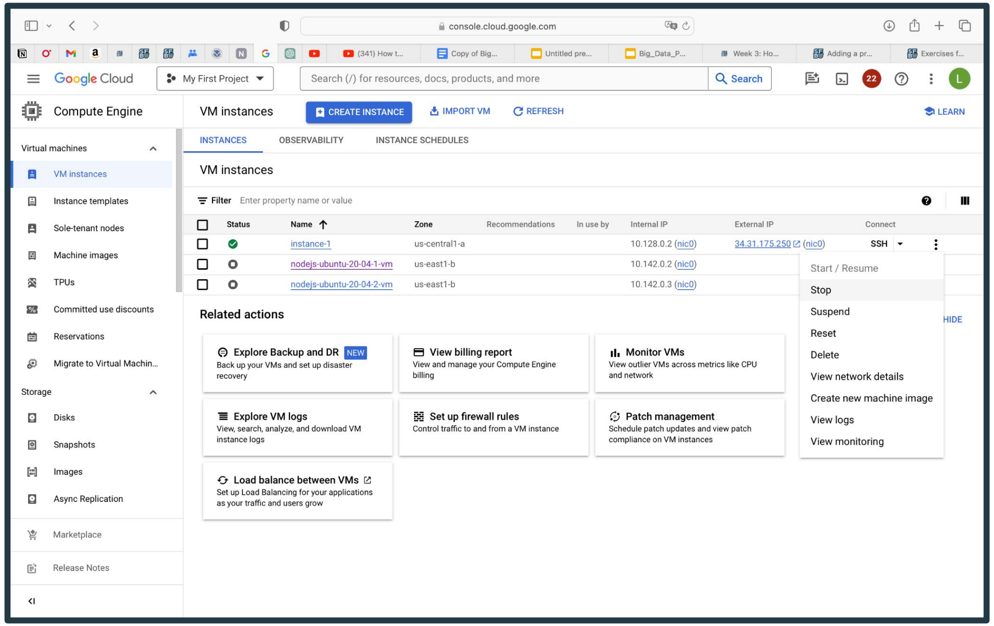


```
{"year":2024,"month":2,"date":7,"hour":10,"minute":57}
```

Final Setting :

Shut/stop VM IN GCP

- Go to Google Cloud Console.
- Select project.
- Click "Compute Engine".
- Choose VM instance.
- Click "Stop" button.
- Confirm action.
- Wait for shutdown.



VM instances

Status	Name	Zone	Recommendations	In use by	Internal IP	External IP	Connect
✓	instance-1	us-central1-a			10.128.0.2 (nic0)	34.31.175.250 (nic0)	SSH
●	nodejs-ubuntu-20-04-1-vm	us-east1-b			10.142.0.2 (nic0)		
●	nodejs-ubuntu-20-04-2-vm	us-east1-b			10.142.0.3 (nic0)		

Related actions

Explore Backup and DR NEW
Back up your VMs and set up disaster recovery

View billing report
View and manage your Compute Engine billing

Monitor VMs
View outlier VMs across metrics like CPU and network

Explore VM logs
View, search, analyze, and download VM instance logs

Set up firewall rules
Control traffic to and from a VM instance

Patch management
Schedule patch updates and view patch compliance on VM instances

Load balance between VMs
Set up Load Balancing for your applications as your traffic and users grow

Enhancement Ideas

- Integrate monitoring tools to track server performance and resource utilization.
- Add logging functionality to track server activities and debug issues.
- Introduce authentication and authorization mechanisms for secure server access.
- Implement rate limiting to prevent abuse or excessive resource usage

Conclusion

- Deploying current_time.js on GCP showcases ease and efficiency of cloud infrastructure for Node.js apps.
- GCP's Compute Engine simplifies setup, offering scalability and resource management.
- Performance monitoring enhances server reliability and optimization.
- Considerations such as pricing and service integrations are crucial when selecting a cloud provider.
- Overall, GCP streamlines Node.js deployment, optimizing development and delivery processes.

References

- Geewax, J. J. (. (2018). Google Cloud Platform in Action. United States: Manning.
- https://www.youtube.com/watch?v=cju2NqPEcp4&t=39s
- https://cloud.google.com/docs/overview
- https://joecreager.com/learnyounode-lesson-10-time-server/
- https://leejjon.medium.com/let-a-node-js-backend-consume-and-produce-json-b206e8043fbf

THANK YOU!