

EECS 498 System Design of a Search Engine

Winter 2026

Lab 11: Google Cloud Platform Continued

GSI: Christopher Davis

IA: Rickey Dong

Nicole Hamilton

<https://web.eecs.umich.edu/~nham/nham@umich.edu>

Agenda

- Introduction
- Project SSH Keys
- Creating a VM Instance
- Building & Running *LinuxTinyServer*
- Configuring a systemd service

Agenda

- **Introduction**
- Project SSH Keys
- Creating a VM Instance
- Building & Running *LinuxTinyServer*
- Configuring a systemd service

Introduction

Presentations / demos are coming up!

See EdStem post #142

<https://edstem.org/us/courses/89542/discussion/7913989>

Review

Recall that we went through creating a demo VM in the last lab.

We will now do it again, but more concisely.

To save time, make sure you do it in the same project as last time!

console.cloud.google.com

Google Cloud MyExampleProject Search (/) for resources, docs, products, and more Search

Welcome, Christopher Davis

You've activated your free trial. Use any remaining credits by June 8, 2020.

\$39 out of \$300 credits used
Expires June 8, 2026

[Add people to your project](#)
[Set up budget alerts](#)
[Review product spend](#)
[See all credit usage](#)

Try Gemini Cloud Assist chat
(Tip: use Option G to open and close the chat)

Chat now →

Recommended based on your interest in General

Products

Create a VM Train and host ML Create a database Analyze and manage

This is your current project; make sure it's the right one.

Agenda

- Introduction
- **Project SSH Keys**
- Creating a VM Instance
- Building & Running *LinuxTinyServer*
- Configuring a systemd service

Adding Project SSH Keys

Instead of manually adding SSH keys to each VM, let's first add them globally to the project.

This allows us to SSH into any Linux VM instance in the project.

In the previous lab, we generated an SSH key at `~/ .ssh/gcp_key`

Again, copy the output of:

```
cat ~/ .ssh/gcp_key.pub
```

Welcome, Christopher

Open the sidebar here.

You've activated your full account

Use any remaining credits, then pay as you go.

\$39 out of \$300 credits
used

Expires June 8, 2026

You're working on project [MyExampleProject](#) ?

Number: 53193759241 ID: myexampleproject-492020

[Add people to your project](#)[Set up budget alerts](#)[Review product spend](#)[See all credit usage](#)

Try Gemini Cloud

Assist chat

(Tip: use Option G to open and
close the chat)

Chat now

Recommended based on your interest in [General](#) ▾

Products



Create a VM



Train and host ML



Create a database



Analyze and manage

The screenshot shows the Google Cloud console interface. On the left, the navigation menu is open. Under the 'Products' section, 'Compute Engine' is highlighted with a red circle and a red arrow pointing to it. A secondary menu is open for 'Compute Engine', and 'Metadata' is highlighted with a red circle and a red arrow pointing to it. A yellow text box on the right contains the instruction: 'Hover over **Compute Engine**, then scroll down and click on **Metadata**.' The background shows the main console area with a search bar, a 'View instances' button, and a 'Health' section.

Google Cloud

Cloud Hub >

Cloud overview >

Solutions >

Recently visited >

Favorite products

Favorite products appear here

Products

Billing ☆

IAM & Admin ☆ >

Marketplace ☆

APIs & Services ☆ >

Vertex AI ☆ >

Compute Engine ☆ >

Kubernetes Engine ☆ >

Cloud Storage ☆ >

Security ☆ >

View all products

VM Extension Manager

VM extension policies

VM Manager

Patch

OS policies

Bare Metal Solution

Servers

Networks

VRFs

Volumes

NFS Shares

Procurements

Maintenance Events

Settings

Metadata

Zones

Network endpoint groups

Preview Features **New**

Operations

Settings

console.cloud.google.com

Search

View instances

Is this page helpful? Refresh

Health

to view details for future incidents.

Service Health

daily trend

Instance Groups

Disks

1

Images

Reservations

0

UTC-4 8:40 AM 8:50 AM 9:00 AM 9:10 AM 9:20 AM 9:30 AM

20%

1 hour

View details in Observability

Open "https://console.cloud.google.com/compute/metadata?project=maia-489721" in a new tab

Google Cloud

eecs542

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

Search

Compute Engine

Disks

Storage Pools

Snapshots

Images

Async Replication

Consistency Groups

Instance groups

Instance groups

Health checks

VM Extension Manager

VM extension policies

VM Manager

Patch

OS policies

Bare Metal Solution

Marketplace

Release Notes

Metadata

Edit

Refresh

Learn

All instances in this project inherit these key-value pairs. [Learn more](#)

Metadata

SSH Keys

Go to **SSH Keys**.

Metadata

Compute Engine metadata lets you specify key-value pairs that are available to all VM instances in the project. A VM instance can query metadata as needed or it can wait to be notified of changes.

Add metadata

Equivalent REST

Google Cloud

eecs542

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

Search

13

Learn

Compute Engine

Disks

Storage Pools

Snapshots

Images

Async Replication

Consistency Groups

Instance groups

Instance groups

Health checks

VM Extension Manager

VM extension policies

VM Manager

Patch

OS policies

Bare Metal Solution

Marketplace

Release Notes

Metadata

Edit

Refresh

All instances in this project inherit these SSH keys. [Learn more](#)

Metadata

SSH Keys

Click **Add SSH key**.

SSH Keys

You can store SSH keys that can be used to connect to the VM instances of a project. Project-level keys are propagated to all VM instances that DO not have their own SSH keys.

Add SSH key

Equivalent REST

Google Cloud console interface showing the SSH Keys configuration page for project eecs542.

The left sidebar lists various services: Compute Engine, Disks, Storage Pools, Snapshots, Images, Async Replication, Consistency Groups, Instance groups, VM Extension Manager, VM Manager, Patch, OS policies, Bare Metal Solution, Marketplace, and Release Notes.

The main content area is titled "Metadata" and shows the "SSH Keys" tab. It displays a list of SSH keys with a table structure:

SSH key 1 *	
	

Below the table is a button labeled "+ Add item".

A red arrow points to the trash icon, indicating the action to delete the existing key.

A yellow callout box contains the text: "Hit **Add item** and paste the SSH key here."

At the bottom, there are "Save" and "Cancel" buttons.

Google Cloud console interface showing the **SSH Keys** section under **Metadata** for project **eeecs542**.

The left sidebar lists various services: Compute Engine, Disks, Storage Pools, Snapshots, Images, Async Replication, Consistency Groups, Instance groups, VM Extension Manager, VM Manager, Patch, OS policies, Bare Metal Solution, Marketplace, and Release Notes.

The main content area displays the **SSH Keys** configuration. It includes a text input field labeled "SSH key 1 *" with the placeholder "Enter public SSH key". A red arrow points to the trash icon next to the input field, indicating the "Add item" action.

A yellow callout box states: **Hit *Add item* and paste the SSH key here.**

At the bottom, there are **Save** and **Cancel** buttons. A red arrow points to the **Save** button, which is circled in red. A yellow callout box states: **Then press *Save*.**

Agenda

- Introduction
- Project SSH Keys
- **Creating a VM Instance**
- Building & Running *LinuxTinyServer*
- Configuring a systemd service

Creating a VM Instance

We will create an instance for our web-server with these specs:

- Type: **e2-medium**
- OS: **Ubuntu 24.04 LTS (Minimal)**
- Boot Disk: **20GB** of **Standard Persist Disk**

Recall: last lab covered how to do this using the GCP web console.

Welcome, Christopher

Open the sidebar here.

You've activated your full account

Use any remaining credits, then pay as you go.



\$39 out of \$300 credits
used

Expires June 8, 2026

You're working on project [MyExampleProject](#) ?

Number: 53193759241 ID: myexampleproject-492020



[Add people to your project](#)

[Set up budget alerts](#)

[Review product spend](#)

[See all credit usage](#)

Try Gemini Cloud

Assist chat

(Tip: use Option G to open and
close the chat)

Chat now



Recommended based on your interest in [General](#) ▾

Products



Create a VM



Train and host ML



Create a database



Analyze and manage

The screenshot shows the Google Cloud console interface. On the left, the navigation menu is open. Under the 'Products' section, 'Compute Engine' is highlighted with a red circle and a red arrow pointing to it. A second red arrow points from 'VM instances' in the sub-menu to the main instruction box. The sub-menu for 'Compute Engine' includes options like 'Overview', 'Security risk overview', 'Virtual machines', 'VM instances', 'Instance templates', 'Sole-tenant nodes', 'Machine images', 'TPUs', 'Committed use discounts', 'Reservations', 'Migrate to Virtual Machines', 'Storage', 'Disks', 'Storage Pools', 'Snapshots', 'Images', 'Async Replication', 'Consistency Groups', 'Instance groups', and 'Health checks'. A yellow box with black text provides instructions: 'Hover over **Compute Engine**, then click on **VM instances**.' The background shows the main console area with a search bar and various project management options.

Google Cloud

Cloud Hub

Cloud overview

Solutions

Recently visited

Favorite products

Favorite products appear here

Products

Billing

IAM & Admin

Marketplace

APIs & Services

Vertex AI

Compute Engine

Kubernetes Engine

Cloud Storage

Security

Overview

Security risk overview

Virtual machines

VM instances

Instance templates

Sole-tenant nodes

Machine images

TPUs

Committed use discounts

Reservations

Migrate to Virtual Machines

Storage

Disks

Storage Pools

Snapshots

Images

Async Replication

Consistency Groups

Instance groups

Instance groups

Health checks

You're working on project **MyExampleProject** @

Number: 53193759241

[Add people to your project](#)

[Set up budget alerts](#)

[Review product spend](#)

[See all credit usage](#)

Chat now

our interest in **General**

Create a database

Analyze and manage

Open "https://console.cloud.google.com/compute/instances?authuser=0&project=myexampleproject-492020" in a new tab

Hover over *Compute Engine*, then click on *VM instances*.

Google Cloud console interface showing the VM instances page. The "Create instance" button is highlighted with a red circle and a red arrow pointing to it. A yellow box contains the text "Click *Create instance*."

VM instances

Instances Observability Instance schedules

VM instances

Filter Enter property name value

Status	Name ↑	Zone	Recommendations	In use by	Internal IP	External IP	Connect
--------	--------	------	-----------------	-----------	-------------	-------------	---------

Click *Create instance*.

VM Instances

Compute Engine lets you use virtual machines that run on Google's infrastructure. Create micro-VMs or larger instances running Debian, Windows, or other standard images. Create your first VM instance, import it using a migration service, or try the quickstart to build a sample app.

Google Cloud

MyExampleProject

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

Search

1

Create an instance

Create VM from...

Equivalent code

Machine configuration

e2-medium, us-central1

OS and storage

Debian GNU/Linux 12 (bookworm)

Data protection

Snapshot schedules

Networking

1 network interface

Observability

Install Ops Agent

Security

Advanced

Machine configuration

Name *

instance-20260401-203728

Region *

us-central1 (Iowa)

Region is permanent

Zone *

Any

Google will choose a zone on your behalf, maximizing VM obtainability. Zone is permanent.

Generate machine type suggestions with Gemini

New

Prompt

Describe your workload and requirements (like cost or performance). Generating machine type suggestions is free.

Low-cost web server

Low-volume database

AI/ML inference

General purpose

Compute optimized

Memory optimized

Storage optimized

GPUs

Machine types for common workloads, optimized for cost and flexibility

	Series	Description	vCPUs	Memory	CPU Platform
☐	C4	Consistently high performance	2 - 288		
☐	C4A	Arm-based consistently high performance	1 - 96		
☐	C4D	Consistently high performance	2 - 384	3 - 3,072 GB	AMD Turin
☐	N4	Flexible & cost-optimized	2 - 80	4 - 640 GB	Intel Emerald

Monthly estimate

\$25.46

That's about \$0.03 hourly

Pay for what you use: no upfront costs and per second billing

Item	Monthly estimate
2 vCPU + 4 GB memory	\$24.46
10 GB balanced persistent disk	\$1.00
Logging	Cost varies
Monitoring	Cost varies
Snapshot schedule	Cost varies
Total	\$25.46

Compute Engine pricing

Cloud Operations pricing

Create

Cancel

Equivalent code

You will get this menu.

Google Cloud

MyExampleProject

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

Search

Create an instance

Create VM from...

Equivalent code

Machine configuration

e2-medium, us-central1

OS and storage

Debian GNU/Linux 12 (bookworm)

Data protection

Snapshot schedules

Networking

1 network interface

Observability

Install Ops Agent

Security

Advanced

General purpose

Compute optimized

Memory optimized

Storage optimized

Machine types for common workloads, optimized for cost and flexibility

	Series ?	Description	vCPUs ?	Memory ?	CPU Platfor
☐	C4	Consistently high performance	2 - 288	4 - 2,232 GB	Intel Emerald
☐	C4A	Arm-based consistently high performance	1 - 96	2 - 768 GB	Google Axion
☐	C4D	Consistently high performance	2 - 384	3 - 3,072 GB	AMD Turin
☐	N4	Flexible & cost-optimized	2 - 80	4 - 640 GB	Intel Emerald
☐	N4A	Arm-based flexibility & cost optimization	1 - 64	2 - 512 GB	Google Axion
☐	N4D	Flexible & cost-optimized	2 - 96	4 - 768 GB	AMD Turin
☐	C3	Consistently high performance	4 - 192	8 - 1,536 GB	Intel Sapphire
☐	C3D	Consistently high performance			
☒	E2	Low cost, day-to-day computing			
☐	N2	Balanced price & performance			
☐	N2D	Balanced price & performance	2 - 224	2 - 896 GB	AMD Milan
☐	T2A	Scale-out workloads	1 - 48	4 - 192 GB	Ampere Altra
☐	T2D	Scale-out workloads	1 - 60	4 - 240 GB	AMD Milan
☐	N1	Balanced price & performance	0.25 - 96	0.6 - 624 GB	Intel Haswell

That's about \$0.03 hourly

Pay for what you use: no upfront costs and per second billing

Item	Monthly estimate
2 vCPU + 4 GB memory	\$24.46
10 GB balanced persistent disk	\$1.00
Logging	Cost varies
Monitoring	Cost varies
Snapshot schedule	Cost varies
Total	\$25.46

[Compute Engine pricing](#)
[Cloud Operations pricing](#)
[Less](#)

Machine type

Choose a machine type with preset amounts of vCPUs and memory that suit most workloads. Or, you can create a

Create

Cancel

Equivalent code

Scroll down to see all the series.

We want series E2.

Google Cloud

MyExampleProject

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

Search

Create an instance

Create VM from...

Equivalent code

Machine configuration

e2-medium, us-central1

OS and storage

Debian GNU/Linux 12 (bookworm)

Data protection

Snapshot schedules

Networking

1 network interface

Observability

Install Ops Agent

Security

Advanced

C3D

Consistently high performance

E2

Low cost, day-to-day computing

N2

Balanced price & performance

N2D

Balanced price & performance

T2A

Scale-out workloads

T2D

Scale-out workloads

N1

Balanced price & performance

Machine type

Choose a machine type with preset amounts of vCPUs and memory that suit most workloads. Or, you can create a custom machine for your workload's particular needs. [Learn more](#)

Preset

Custom

e2-medium (2 vCPU, 1 core, 4 GB memory)

vCPU

1-2 vCPU (1 shared core)

Memory

4 GB

Advanced configurations

Provisioning model

VM provisioning model

Standard

Determines how long your VMs will run. Choice is permanent

We want series E2.

Monthly estimate

\$25.46

That's about \$0.03 hourly

Pay for what you use: no upfront costs and per second billing

Item	Monthly estimate
2 vCPU + 4 GB memory	\$24.46
10 GB balanced persistent disk	\$1.00
Logging	Cost varies
Monitoring	Cost varies
Snapshot schedule	Cost varies
Total	\$25.46

[Compute Engine pricing](#)

[Cloud Operations pricing](#)

Choose e2-medium

Create

Cancel

Equivalent code

Google Cloud

MyExampleProject

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

Search

Create an instance

Create VM from...

Equivalent code

Machine configuration

e2-standard-2 (2 vCPU, 8 GB memory)

OS and storage

Debian GNU/Linux 12 (bookworm)

Data protection

Snapshot schedules

Networking

1 network interface

Observability

Install Ops Agent

Security

Advanced

C3D

Consistently high performance

4 - 360

8 - 2,880 GB

AMD Genoa

E2

Low cost, day-to-day computing

0.25 - 32

1 - 128 GB

Intel Broadwell

N2

Balance

scade

N2D

Balance

lan

T2A

Scale-optimized

Altra

T2D

Scale-optimized

lan

N1

Balance

swell

Machine type

Choose a machine type with preset amounts of vCPUs and memory that suit most workloads. Or, you can create a custom machine for your workload's particular needs. [Learn more](#)

Preset

Custom

e2-standard-2 (2 vCPU, 1 core, 8 GB memory)

vCPU

2 (1 core)

Memory

8 GB

Advanced configurations

Provisioning model

VM provisioning model

Standard

Determines how long your VMs will run. Choice is permanent

Create

Cancel

Equivalent code

Monthly estimate

\$49.92

That's about \$0.07 hourly

Pay for what you use: no upfront costs and per second billing

Item	Monthly estimate
2 vCPU + 8 GB memory	\$48.92
10 GB balanced persistent disk	\$1.00
Logging	Cost varies
Monitoring	Cost varies
Snapshot schedule	Cost varies
Total	\$49.92

[Compute Engine pricing](#)

[Cloud Operations pricing](#)

Less

Google Cloud

MyExampleProject

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

Search

Create an instance

Create VM from...

Equivalent code

Machine configuration

e2-standard-2, us-central1

OS and storage

Debian GNU/Linux 12 (bookworm)

Data protection

Snapshot schedules

Networking

1 network interface

Observability

Install Ops Agent

Security

Advanced

Operating system and storage

Name

instance-20260401-203728

Type

New balanced persistent disk

Size

10 GB

Snapshot schedule

default-schedule-1

License type

Free

Image

Debian GNU/Linux

Change

Additional disks

+ Add new disk

+ Attach existing disk

+ Add local SSD

Container

Deprecated

Deploy a container image to this VM instance

Deploy container

Monthly estimate

\$49.92

That's about \$0.07 hourly

Pay for what you use: no upfront costs and per second billing

Item	Monthly estimate
GB memory	\$48.92
enced persistent	\$1.00
	Cost varies
	Cost varies
Snapshot schedule	Cost varies
Total	\$49.92

[Compute Engine pricing](#)

[Cloud Operations pricing](#)

Less

Create

Cancel

Equivalent code

Click here to change the OS and boot disk type.

Google Cloud

MyExampleProject

Create an instance

Create VM from...

Machine configuration

e2-standard-2, us-central1

OS and storage

Debian GNU/Linux 12 (bookworm)

Data protection

Snapshot schedules

Networking

1 network interface

Observability

Install Ops Agent

Security

Advanced

Operating system

Name

Type

Size

Snapshot schedule ?

License type ?

Image

Change

Additional disks

+ Add new disk

Container

Deprecate

Deploy a container image to this instance

Deploy container

Create

Cancel

Boot disk

Select an image or snapshot to create a boot disk; or attach an existing disk. Can't find what you're looking for? Explore hundreds of VM solutions in [Marketplace](#)

Public images

Custom images

Snapshots

Archive Snapshots

Existing Disks

Operating system

Filter ubuntu

Ubuntu

Ubuntu Accelerator Optimized

Ubuntu Pro

Compare disk types

Size (GB) *
10
Provision between 10 and 65536 GB

Show advanced configuration

Select

Cancel

Let's use Ubuntu for the demo. Scroll down or filter and select it.

Google Cloud console.cloud.google.com

Machine configuration
e2-medium, us-central1

OS and storage
Debian GNU/Linux 12 (bookworm)

Data protection
Snapshot schedules

Networking
1 network interface

Observability
Install Ops Agent

Security

Advanced

Operating system

Name
Type
Size
Snapshot schedule
License type
Image

Additional disks
+ Add new disk

Container
Deploy a container image to this instance

Boot disk

Select an image or snapshot to create a boot disk; or attach an existing disk. Can't find what you're looking for? Explore hundreds of VM solutions in [Marketplace](#)

Public images Custom images Snapshots Archive Snapshots Existing Disks

Operating system
Ubuntu

Version *

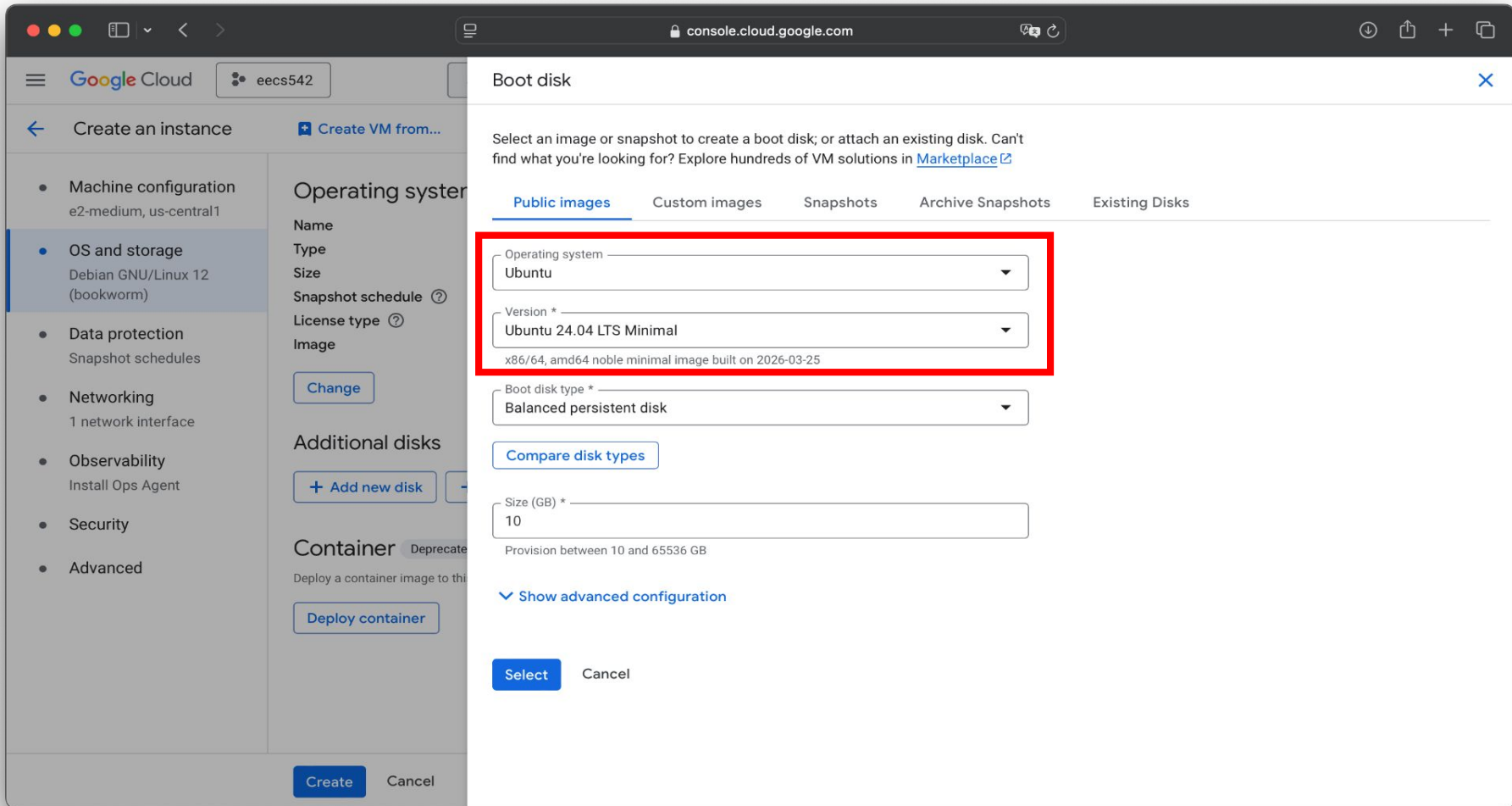
Filter Type to filter

- Ubuntu 25.10 Minimal
Arm64, arm64 questing minimal image built on 2026-03-20
- Ubuntu 25.10 Minimal
x86/64, amd64 questing minimal image built on 2026-03-20
- Ubuntu 24.04 LTS Minimal
Arm64, arm64 noble minimal image built on 2026-03-25
- Ubuntu 24.04 LTS Minimal
x86/64, amd64 noble minimal image built on 2026-03-25
- Ubuntu 22.04 LTS Minimal
x86/64, amd64 jammy minimal image built on 2026-04-02
- Ubuntu 22.04 LTS Minimal

Select Cancel

Create Cancel

We will use the x86/64
Ubuntu 24.04 LTS
Minimal.



Google Cloud console.cloud.google.com

eeecs542

Create an instance

Operating system

- Machine configuration
e2-medium, us-central1
- OS and storage**
Debian GNU/Linux 12 (bookworm)
- Data protection
Snapshot schedules
- Networking
1 network interface
- Observability
Install Ops Agent
- Security
- Advanced

Additional disks

+ Add new disk

Container

Deploy a container image to this instance

Deploy container

Create Cancel

Boot disk

Select an image or snapshot to create a boot disk; or attach an existing disk. Can't find what you're looking for? Explore hundreds of VM solutions in [Marketplace](#)

Public images Custom images Snapshots Archive Snapshots Existing Disks

Operating system
Ubuntu

Version *
Ubuntu 24.04 LTS Minimal

x86_64, amd64 noble minimal image built on 2026-03-25

Boot disk type *
Balanced persistent disk

Compare disk types

Size (GB) *
10

Provision between 10 and 65536 GB

Show advanced configuration

Select Cancel

Now change the boot disk to **20 GB** of **Standard Persistent Disk**.

Google Cloud

MyExampleProject

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

Search

1

Create an instance

Create VM from...

Equivalent code

Machine configuration

e2-standard-2, us-central1

OS and storage

Ubuntu 25.10 Minimal

Data protection

Snapshot schedules

Networking

1 network interface

Observability

Install Ops Agent

Security

Advanced

Networking

Firewall

Select firewall rules to allow specific network traffic from the Internet. Firewall rules will only be applied when an instance is created.

☐ Allow HTTP traffic

☐ Allow HTTPS traffic

☐ Allow Load Balancer Health Checks

Network tags

Network tags

Hostname

Hostname

Set a custom hostname for this instance

IP forwarding

☐ Enable

Network performance configuration

Network bandwidth

☐ Enable per VM Tier_1 networking performance

Create

Cancel

Equivalent code

Monthly estimate

\$49.32

That's about \$0.07 hourly

Pay for what you use: no upfront costs and per second billing

Item	Monthly estimate
2 vCPU + 8 GB memory	\$48.92
10 GB standard persistent disk	\$0.40
Logging	Cost varies
Monitoring	Cost varies
Snapshot schedule	Cost varies
Total	\$49.32

Now go to the **Networking** tab, and scroll down until you see **Firewall** settings.

Google Cloud

MyExampleProject

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

Search

Create an instance

Create VM from...

Equivalent code

Machine configuration

e2-medium, us-central1-b

OS and storage

Ubuntu 24.04 LTS Minimal

Data protection

Snapshot schedules

Networking

1 firewall rule, 1 network interface

Observability

Install Ops Agent

Security

Advanced

Networking

Firewall

Select firewall rules to allow specific network traffic from the Internet. Firewall rules will only be applied when an instance is created.

☒ Allow HTTP traffic

☐ Allow HTTPS traffic

☐ Allow Load Balancer Health Checks

Network tags

Network tags

http-server

Hostname

Hostname

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

IP forwarding

☐ Enable

Network performance configuration

Network bandwidth

☐ Enable per VM Tier_1 networking performance

Monthly estimate

\$26.46

That's about \$0.04 hourly

Pay for what you use: no upfront costs and per second billing

Item	Monthly estimate
2 vCPU + 4 GB memory	\$24.46
20 GB balanced persistent disk	\$2.00
Logging	Cost varies
Monitoring	Cost varies
Snapshot schedule	Cost varies
Total	\$26.46

[Compute Engine pricing](#)

[Cloud Operations pricing](#)

Less

Create

Cancel

Equivalent code

Check **Allow HTTP traffic.**

Google Cloud

MyExampleProject

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

Search

5

Create an instance

Create VM from...

Equivalent code

Machine configuration

e2-medium, us-central1-b

OS and storage

Ubuntu 24.04 LTS Minimal

Data protection

Snapshot schedules

Networking

1 firewall rule, 1 network interface

Observability

Install Ops Agent

Security

Advanced

Networking

Firewall

Network tags

Hostname

IP forwarding

Network performance configuration

Allow HTTP traffic

Allow HTTPS traffic

Allow Load Balancer Health Checks

http-server

Hostname

Enable

Enable per VM T...

Now hit **Create**.

Monthly estimate

\$26.46

That's about \$0.04 hourly

Pay for what you use: no upfront costs and per second billing

Item	Monthly estimate
2 vCPU + 4 GB memory	\$24.46
20 GB balanced persistent disk	\$2.00
Logging	Cost varies
Monitoring	Cost varies
Snapshot schedule	Cost varies
Total	\$26.46

Compute Engine pricing

Cloud Operations pricing

Less

Create

Cancel

Equivalent code

Agenda

- Introduction
- Project SSH Keys
- Creating a VM Instance
- **Building & Running *LinuxTinyServer***
- Configuring a systemd service

Setting Up the VM Environment

Let's set up our VM environment just like we did for *LinuxGetSsl*.

You can:

- SSH from the web console directly
- Add a local SSH key to the instance like in the previous lab
- Or, SSH using **gcloud**, which automatically handles the local keys

Setting Up the VM Environment

Let's set up our VM environment just like we did for *LinuxGetSsl*.

You can:

- SSH from the web console directly
- Add a local SSH key to the instance like in the previous lab
- Or, SSH using **gcloud** CLI, which automatically handles the local keys

Since we already set up a global SSH key for the project, we can just run:

```
ssh -i ~/.ssh/gcp_key <username>@<vm_external_ip>
```

Setting Up the VM Environment (review)

Now we set up the environment exactly as with *LinuxGetSsl*.

In particular, run these commands in the SSH terminal on the VM:

```
sudo apt update && sudo apt upgrade -y
```

```
sudo apt install -y build-essential cmake git gdb
```

Then depending on if you want **vim** or **nano**:

```
sudo apt install -y vim
```

or

```
sudo apt install -y nano
```

Getting Code onto GCP

Now, we just need to get our *LinuxTinyServer* code onto GCP.

Just like last lab, there are several ways to do this. Namely:

- Setting up Github credentials on GCP (or vice versa)
- Using `scp` or `rsync` with a local SSH key
- Using `gcloud`

Getting Code onto GCP (scp)

For example, with this tutorial's naming, I would run:

```
scp -r -i ~/.ssh/gcp_key \  
    ~/local/path/to/my/code \  
    <username>@<vm_ip>:~/LinuxTinyServer
```

Getting HTML onto GCP

Since this is a web-server, we need some HTML to serve.

Luckily, we already have some from the HTMLParser assignment, such as the `NYTimes.html` file.

Let's put them in a `~/html` directory. On the remote VM, run:

```
mkdir ~/html
```

Getting HTML onto GCP

For every local HTML file `example.html` that you want to copy, run this locally:

```
scp -i ~/.ssh/gcp_key \  
    ~/local/path/to/my/code \  
    <username>@<vm_ip>:~/html/
```

Note the trailing slash in `~/html/`

Getting HTML onto GCP

For every local HTML file `example.html` that you want to copy, run this locally:

```
scp -i ~/.ssh/gcp_key \  
    ~/local/path/to/my/code \  
    <username>@<vm_ip>:~/html/
```

Note the trailing slash in `~/html/`

You can also get new webpages with curl; e.g., on the remote VM you can do:

```
curl -L -o ~/html/linux.html "https://en.wikipedia.org/wiki/Linux"
```

Building & Running *LinuxTinyServer*

Now, you can build and run *LinuxTinyServer* as you usually would.

For instance:

```
cd ~/LinuxTinyServer
```

```
g++ -pthread LinuxTinyServer.cpp -o LinuxTinyServer
```

```
sudo ./LinuxTinyServer 80 ~/html
```

Connecting to LinuxTinyServer

We can then connect to the LinuxTinyServer as usual, except we specify the external IP of the VM instance instead of localhost.

I.e., you can enter the following into your browser:

```
http://{vm_ip_address}/{file_name}.html
```

Connecting to LinuxTinyServer

We can then connect to the LinuxTinyServer as usual, except we specify the external IP of the VM instance instead of localhost.

I.e., you can enter the following into your browser:

```
http://{vm_ip_address}/{file_name}.html
```

Alternatively, fetch with curl (might have to use `--http0.9` flag):

```
curl -v http://{vm_ip_address}/{file_name}.html
```



U.S. INTERNATIONAL CANADA ESPAÑOL 中文

Thursday, April 9, 2026

Today's Paper

The New York Times

World U.S. Politics N.Y. Business Opinion Tech Science Health Sports Arts Books Style Food Travel Magazine T Magazine Real Estate Video



Listen to 'The Daily'
Impeached, again.



The Book Review Podcast
Charles Yu on his National Book Award-winning
novel, "Interior Chinatown."



Listen to 'The Argument'
Jane Coaston loves Section 230. Got a problem with
that?

LIVE

Fractures in G.O.P. Widen After Impeachment Vote

- Some House Republicans have called on Representative Liz Cheney to resign from her leadership post after she voted to impeach President Trump.
- Ms. Cheney was one of 10 Republicans to break with the party on the vote, setting up a messy internal feud that could define its future.
- President-elect Joe Biden has stayed largely removed from the fray of the impeachment and kept his focus on the pandemic and the economy. Here's the latest.



Erin Schaff/The New York Times

Rare Pentagon Mission: Armed Troops
and not active-duty. But officials know such e

'Suspicious' Visits to Capitol Eyed as Dragnet Snare 2 Police Officers

Democratic lawmakers
demanded answers about
visitors to the Capitol a day
before the riot.

Here's NYTimes.html on my demo server.

`http://{ip_addr}/NYTimes.html`

Agenda

- Introduction
- Project SSH Keys
- Creating a VM Instance
- Building & Running *LinuxTinyServer*
- **Configuring a systemd service**

Reliably Run a Long-Running Service

We can now run our *LinuxTinyServer* and serve incoming requests.

However, merely running it from the terminal means it may die whenever our SSH session ends.

Using something like **tmux** can help, but it is not a permanent solution:

- Has session overhead
- Not an OS first-class citizen
- No automatic restarts or reliability

The “real” solution here is to use a **systemd** service.

Configuring systemd

We can define a service via a `.service` file in the `systemd` directory.

This is an INI file (sort of like YAML) that defines the config for the service.

For instance, it can control:

- Automatic restarts
- Starting service on boot
- User/Group permissions
- Executable to run (and its argument)
- Environment variables
- Working directory (where the executable is run)

Making a New Service

We will create `/etc/systemd/system/linuxtinyserver.service`

You can find an example `linuxtinyserver.service` file on Canvas in Files, under Labs.

Download this file, and then edit it to replace all instances of `<username>` with the username you used to SSH into the VM.

Adding Service to systemd

Now, we just need to put this file into the `/etc/systemd/system/` directory.

You can do this by copy/pasting with `vim` or `nano`, or by using `scp` / `nano`.

For instance, on a local terminal:

```
scp -i ~/.ssh/gcp_key \  
    ~/local/path/to/linuxtinyserver.service \  
    <username>@<vm_ip>:~/LinuxTinyServer/
```

And then on the remote VM instance:

```
sudo cp ~/LinuxTinyServer/linuxtinyserver.service \  
    /etc/systemd/system/
```

Service Permissions

Make sure the service file has the proper ownership and permissions:

```
sudo chown root:root /etc/systemd/system/linuxtinyserver.service
```

```
sudo chmod 644 /etc/systemd/system/linuxtinyserver.service
```

[Unit]

Description=Linux Tiny Server C++ Bi

After=network-online.target # Ensur

Wants=network-online.target

Here's an overview of the file.

[Service]

Type=simple

User=<username> # Run service as yo

Group=<username>

Restart=always # Automatically restart the service if it crashes

RestartSec=3 # Wait 3 seconds before restarting

StandardOutput=journal

StandardError=journal

TimeoutStopSec=30

Note: this is condensed to fit on the slide and not technically valid; comments in INI have to be on their own lines.

The absolute path to your executable

ExecStart=/home/<username>/LinuxTinyServer/LinuxTinyServer 80 /home/<username>/html

The directory from which the binary is run

WorkingDirectory=/home/<username>/LinuxTinyServer

[Install]

WantedBy=multi-user.target # Starts the service in standard multi-user boot mode

Explaining the Service File

As you can see, there are a lot of options on this service file, and even more you can set.

We will briefly go over each of the specified attributes in the example file.

There are a lot of other fields and options you can set. I'd recommend reading more documentation on the later.

E.g.,

https://docs.redhat.com/en/documentation/red_hat_enterprise_linux/9/html/using_systemd_unit_files_to_customize_and_optimize_your_system/assembly_working-with-systemd-unit-files_working-with-systemd

[Unit]

Description=Linux Tiny Server C++ Binary

After=network-online.target # Ensures the network is up before starting the service

Wants=network-online.target

The ***Description*** is a simple plaintext description of the service.

[Service]

Type=simple

User=<username> # Run service as your user rather than root (recommended for security)

Group=<username>

Restart=always # Automatically restart the service if it crashes

RestartSec=3 # Wait 3 seconds before restarting

StandardOutput=journal

StandardError=journal

TimeoutStopSec=30

The absolute path to your executable

ExecStart=/home/<username>/LinuxTinyServer/LinuxTinyServer 80 /home/<username>/html

The directory from which the binary is run

WorkingDirectory=/home/<username>/LinuxTinyServer

[Install]

WantedBy=multi-user.target # Starts the service in standard multi-user boot mode

```
[Unit]
Description=Linux Tiny Server C++ Binary
After=network-online.target # Ensures the network is up before starting the service
Wants=network-online.target
```

```
[Service]
Type=simple
User=<username> # Run service as your user (root by default)
Group=<username>
Restart=always # Automatically restart the service
RestartSec=3 # Wait 3 seconds before restarting
StandardOutput=journal
StandardError=journal
TimeoutStopSec=30
```

The ***After*** and ***Wants*** line here indicate that the service should wait for **network-online.target**

I.e., it waits for the network to be online before starting the service

```
# The absolute path to your executable
ExecStart=/home/<username>/LinuxTinyServer/LinuxTinyServer 80 /home/<username>/html
```

```
# The directory from which the binary is run
WorkingDirectory=/home/<username>/LinuxTinyServer
```

```
[Install]
WantedBy=multi-user.target # Starts the service in standard multi-user boot mode
```

```
[Unit]
Description=Linux Tiny Server C++ Binary
After=network-online.target # Ensures the network is up before starting the service
Wants=network-online.target
```

```
[Service]
```

```
Type=simple
```

```
User=<username> # Run service as your user rather than root (recommended for security)
Group=<username>
Restart=always # Automatically restart the service
RestartSec=3 # Wait 3 seconds before restarting
StandardOutput=journal
StandardError=journal
TimeoutStopSec=30
```

```
# The absolute path to your executable
```

```
ExecStart=/home/<username>/LinuxTinyServer
```

```
# The directory from which the binary is run
```

```
WorkingDirectory=/home/<username>/LinuxTinyServer
```

```
[Install]
```

```
WantedBy=multi-user.target # Starts the service in standard multi-user boot mode
```

The **Type** here indicates how the service acts: is it a daemon, does it fork, etc.

simple is the default and means a simple process that does not need to handle forking or acting as a daemon.

```
[Unit]
Description=Linux Tiny Server C++ Binary
After=network-online.target # Ensures the network is up before starting the service
Wants=network-online.target

[Service]
Type=simple
User=<username> # Run service as your user rather than root (recommended for security)
Group=<username>
Restart=always # Automatically restart the service if it crashes
RestartSec=3 # Wait 3 seconds before restarting
StandardOutput=journal
StandardError=journal
TimeoutStopSec=30

# The absolute path to your executable
ExecStart=/home/<username>/LinuxTinyServer

# The directory from which the binary is run
WorkingDirectory=/home/<username>/LinuxTinyServer

[Install]
WantedBy=multi-user.target # Starts the service at boot
```

User and **Group** specify the user/group permissions the service will have.

For brevity, we will just set them to your username on the VM.

For a real service, it would be better to set a custom User and Group.

```
[Unit]
Description=Linux Tiny Server C++ Binary
After=network-online.target # Ensures the network is up before starting the service
Wants=network-online.target

[Service]
Type=simple
User=<username> # Run service as your user rather than root (recommended for security)
Group=<username>
Restart=always # Automatically restart the service if it crashes
RestartSec=3 # Wait 3 seconds before restarting
StandardOutput=journal
StandardError=journal
TimeoutStopSec=30

# The absolute path to your executable
ExecStart=/home/<username>/LinuxTinyServer

# The directory from which the binary is run
WorkingDirectory=/home/<username>/LinuxTinyServer

[Install]
WantedBy=multi-user.target # Starts the service at boot
```

Restart controls how the service restarts.

always means it always restarts, but there is also an **on-failure** option, which only restarts for non-zero exit statuses

RestartSec is how many second between restarts.

```
[Unit]
Description=Linux Tiny Server C++ Binary
After=network-online.target # Ensures
Wants=network-online.target
```

This controls where **stdout** and **stderr** get sent.

```
[Service]
Type=simple
User=<username> # Run service as your
Group=<username>
Restart=always # Automatically restart the service if it crashes
RestartSec=3 # Wait 3 seconds before restarting
StandardOutput=journal
StandardError=journal
TimeoutStopSec=30
```

The default is the **systemd** journal. I set it here to be explicit.

```
# The absolute path to your executable
ExecStart=/home/<username>/LinuxTinyServer/LinuxTinyServer 80 /home/<username>/html

# The directory from which the binary is run
WorkingDirectory=/home/<username>/LinuxTinyServer

[Install]
WantedBy=multi-user.target # Starts the service in standard multi-user boot mode
```

```
[Unit]
Description=Linux Tiny Server C++ Binary
After=network-online.target # Ensures the network is up before starting the service
Wants=network-online.target

[Service]
Type=simple
User=<username> # Run service as your user rather than root (recommended for security)
Group=<username>
Restart=always # Automatically restart the service if it crashes
RestartSec=3 # Wait 3 seconds before restarting
StandardOutput=journal
StandardError=journal
TimeoutStopSec=30

# The absolute path to your executable
ExecStart=/home/<username>/LinuxTinyServer/LinuxTinyServer 80 /home/<username>/html

# The directory from which the binary is run
WorkingDirectory=/home/<username>/LinuxTinyServer

[Install]
WantedBy=multi-user.target # Starts the service in standard multi-user boot mode
```

This is the time to stop the service.

```
[Unit]
Description=Linux Tiny Server C++ Binary
After=network-online.target # Ensures the network is up before starting the service
Wants=network-online.target

[Service]
Type=simple
User=<username> # Run service as your user rather than root (recommended for security)
Group=<username>
Restart=always # Automatically restart the service
RestartSec=3 # Wait 3 seconds before restarting
StandardOutput=journal
StandardError=journal
TimeoutStopSec=30

# The absolute path to your executable
ExecStart=/home/<username>/LinuxTinyServer/LinuxTinyServer 80 /home/<username>/html

# The directory from which the binary is run
WorkingDirectory=/home/<username>/LinuxTinyServer

[Install]
WantedBy=multi-user.target # Starts the service in standard multi-user boot mode
```

ExecStart is the absolute path to the executable, along with any arguments for the program.

```
[Unit]
Description=Linux Tiny Server C++ Binary
After=network-online.target # Ensures the network is up before starting the service
Wants=network-online.target

[Service]
Type=simple
User=<username> # Run service as your user rather than root (recommended for security)
Group=<username>
Restart=always # Automatically restart the service if it crashes
RestartSec=3 # Wait 3 seconds before restarting
StandardOutput=journal
StandardError=journal
TimeoutStopSec=30

# The absolute path to your executable
ExecStart=/home/<username>/LinuxTinyServer

# The directory from which the binary is run
WorkingDirectory=/home/<username>/LinuxTinyServer

[Install]
WantedBy=multi-user.target # Starts the service in standard multi-user boot mode
```

WorkingDirectory is the directory from which the executable will be run.

```
[Unit]
Description=Linux Tiny Server C++ Binary
After=network-online.target # Ensures the network is up before starting the service
Wants=network-online.target

[Service]
Type=simple
User=<username> # Run service as your user rather than root (recommended for security)
Group=<username>
Restart=always # Automatically restart the service if it crashes
RestartSec=3 # Wait 3 seconds before restarting
StandardOutput=journal
StandardError=journal
TimeoutStopSec=30

# The absolute path to your executable
ExecStart=/home/<username>/LinuxTinyServer

# The directory from which the binary is run
WorkingDirectory=/home/<username>/LinuxTinyServer

[Install]
WantedBy=multi-user.target # Starts
```

What this line does is make the service start automatically during bootup.

Specifically, during the multi-user (non-graphical, text mode) state.

Activating the Service

First, ensure the binary is executable:

```
chmod +x ~/LinuxTinyServer/LinuxTinyServer
```

Since port 80 is protected, we need to give *LinuxTinyServer* permission to use it. You can do that with this command, replacing USERNAME with your username:

```
sudo setcap 'cap_net_bind_service=+ep' \  
    /home/USERNAME/LinuxTinyServer/LinuxTinyServer
```

You can check the permissions with:

```
getcap /home/USERNAME/LinuxTinyServer/LinuxTinyServer
```

Load and Start the Service

To run the service, do the following commands on the VM:

```
sudo systemctl daemon-reload
```

```
sudo systemctl enable linuxtinyserver
```

```
sudo systemctl start linuxtinyserver
```

```
sudo systemctl status linuxtinyserver
```

```
christopherdavis@demo-server:~$ sudo systemctl status linuxtinyserver
• linuxtinyserver.service - Linux Tiny Server C++ Binary
   Loaded: loaded (/etc/systemd/system/linuxtinyserver.service;
   enabled; preset: enabled)
   Active: active (running) since Fri 2026-04-10 16:01:36 UTC; 13s
   ago
     Main PID: 7795 (LinuxTinyServer)
       Tasks: 1 (limit: 4598)
      Memory: 292.0K (peak: 460.0K)
         CPU: 4ms
        CGroup: /system.slice/linuxtinyserver.service
                └─7795
                /home/christopherdavis/LinuxTinyServer/LinuxTinyServer 80
                /home/christopherdavis/html

Apr 10 16:01:36 demo-server systemd[1]: linuxtinyserver.service:
Scheduled restart job, restart counter is at 96.
Apr 10 16:01:36 demo-server systemd[1]: Started
linuxtinyserver.service - Linux Tiny Server C++ Binary.
```

Summary

Your service should be active!

It now runs independently of SSH sessions, and will automatically restart if the process dies or the VM reboots.

To stop your service (and prevent it from restarting on boot), run this:

```
sudo systemctl disable --now linuxtinyserver
```

To re-enable it, run the four commands on the previous step.

Summary

Remember to Suspend or Stop your VM!

In either case—since we created a systemd service—our web-server will reboot.

Next Steps

We will also release a guide on creating custom images, instance templates, and CI/CD with GitHub!