Prueba Técnica Administrador Cloud

Instalación de Prerrequisitos

Configuración de mi maquina

```
$ neofetch

#output
                   -`                    tux@mbpfan
                  .o+`                   ----------
                 `ooo/                   OS: Arch Linux x86_64
                `+oooo:                  Kernel: 6.16.10-arch1-1
               `+oooooo:                 Uptime: 1 hour, 25 mins
               -+oooooo+:                Packages: 988 (pacman), 6 (flatpak)
             `/:-:++oooo+:               Shell: bash 5.3.3
            `/++++/+++++++:              Resolution: 1920×1080, 1920×1080
           `/++++++++++++++:             WM: i3
          `/+++ooooooooooooo/`           Theme: Adwaita [GTK2], Adwaita-dark [G
TK3]
```

```
        ./ooosssso++osssssso+`        Icons: Adwaita [GTK2], Arc [GTK3]
       .oossssso-````/ossssss+`       Terminal: termite
      -osssssso.      :sssssssso.     Terminal Font: Monospace 9
     :osssssss/        osssso+++.     CPU: AMD Ryzen 5 3600 (12) @ 4.618GHz
    /ossssssss/        +ssssooo/-     GPU: NVIDIA GeForce GTX 1650 SUPER
  `/ossssso+/:-        -:/+osssso+-   Memory: 4967MiB / 31998MiB
 `+sso+:-`               `.-/+oso:
`++:.                      `-/+/
.`                          `/
```

Adicionar el repositorio oficial

```
$ curl -O https://dl.google.com/dl/cloudsdk/channels/rapid/downloads/google-cloud-cli-linux-x86_64.tar.gz
```

Instalar Google Cloud SDK

```
$ tar -xf google-cloud-cli-linux-x86_64.tar.gz
$ ./google-cloud-sdk/install.sh
```

Verificar gcloud

```
$ gcloud version

# output
Google Cloud SDK 541.0.0
bq 2.1.24
bundled-python3-unix 3.12.9
core 2025.09.29
gcloud-crc32c 1.0.0
gsutil 5.35
```

Configurar cuenta

```
$ gcloud init

# output
Welcome! This command will take you through the configuration of gcloud.

Your current configuration has been set to: [default]

You can skip diagnostics next time by using the following flag:
  gcloud init --skip-diagnostics

Network diagnostic detects and fixes local network connection issues.
Checking network connection...done.
Reachability Check passed.
Network diagnostic passed (1/1 checks passed).

You must sign in to continue. Would you like to sign in (Y/n)?

Your browser has been opened to visit:
```

Crear Proyecto

```
$ gcloud projects create efraintest01

# output
Create in progress for [https://cloudresourcemanager.googleapis.com/v1/projects/efraintest01].
Waiting for [operations/create_project.global.7534224932349503905] to finish...done.
Enabling service [cloudapis.googleapis.com] on project [efraintest01]...
Operation "operations/acat.p2-39556873005-dec9bd0c-3299-491a-a62d-102194541955" finished successfully.
```

Instalar Terraform

```
$ sudo pacman -S terraform
```

Verificar instalación de Terraform

```
$ terraform -version
Terraform v1.13.3
on linux_amd64
```

Instalar Visual Studio Code

```
$ flatpak install flathub com.visualstudio.code -y
Looking for matches…
Required runtime for com.visualstudio.code/x86_64/stable (runtime/org.freed
esktop.Sdk/x86_64/24.08) found in remote flathub
```

Instalar extensiones vscode para IaC

```
$ flatpak run com.visualstudio.code --install-extension hashicorp.terraform
$ flatpak run com.visualstudio.code --install-extension ms-vscode.azure-acc
ount
$ flatpak run com.visualstudio.code --install-extension googlecloudtools.clou
dcode
$ flatpak run com.visualstudio.code --install-extension ms-azuretools.vscode
-docker
$ flatpak run com.visualstudio.code --install-extension eamodio.gitlens
```

Escenario 1

Respuestas en el mismo documento.

Escenario 2

Crear Cuenta de Servicio

Crear cuenta de servicio terraform-service-account [terraform-sa]

```
$ gcloud iam service-accounts create terraform-sa \
  --display-name="Terraform Service Account"
Created service account [terraform-sa]
```

Asignar rol no-owner a la cuenta terraform-sa

```
$ export PROJECT_ID="efraintest01"
$  gcloud projects add-iam-policy-binding $PROJECT_ID \
  --member="serviceAccount:terraform-sa@${PROJECT_ID}.iam.gserviceacc
ount.com" \
  --role="roles/editor"

# output
Updated IAM policy for project [efraintest01].
bindings:
- members:
  - serviceAccount:terraform-sa@efraintest01.iam.gserviceaccount.com
  role: roles/editor
- members:
  - user:efraesco@gmail.com
  role: roles/owner
etag: BwZAW4NGTs0=
version: 1
```

Descargar la clave para Terraform

```
$ gcloud iam service-accounts keys create ~/terraform-key.json \
  --iam-account="terraform-sa@${PROJECT_ID}.iam.gserviceaccount.com"

# output
created key [74590b0d91daac7891574bfe07aa4a6b5b9222d5] of type [json]
as [/home/tux/terraform-key.json] for [terraform-sa@efraintest01.iam.gservic
eaccount.com]
```

Exportar la clave para Terraform

```
$ export GOOGLE_APPLICATION_CREDENTIALS="$HOME/terraform-key.jso
n"
```

Validar autenticación con el archivo de credenciales

```
$ gcloud auth activate-service-account --key-file=$GOOGLE_APPLICATION_
CREDENTIALS

# output
Activated service account credentials for: [terraform-sa@efraintest01.iam.gse
rviceaccount.com]


$ gcloud auth list

# output
                 Credentialed Accounts
ACTIVE  ACCOUNT
        efraesco@gmail.com
*       terraform-sa@efraintest01.iam.gserviceaccount.com

To set the active account, run:
    $ gcloud config set account `ACCOUNT`
```

Crear estructura del proyecto

```
gcptest/
├── main.tf
├── variables.tf
├── outputs.tf
├── terraform.tfvars
├── modules/
│   ├── network/
│   │   └── network.tf
│   └── compute/
│       └── compute.tf
```

Crear cada uno de los archivos en el directorio gcptest desde vscode.

```
$ cd gcptest/
$ flatpak run com.visualstudio.code .

# output
Warning: 'ms-enable-electron-run-as-node' is not in the list of known options, but still passed to Electron/Chromium.
```

variables.tf

```
variable "project_id" {
  description = "ID del proyecto GCP"
  type        = string
}

variable "region" {
  description = "Región para los recursos"
  type        = string
  default     = "us-central1"
```

```
}

variable "zone" {
  description = "Zona de despliegue"
  type        = string
  default     = "us-central1-a"
}

variable "network_name" {
  description = "Nombre de la red VPC"
  type        = string
  default     = "web-app-vpc"
}

variable "subnet_name" {
  description = "Nombre de la subred"
  type        = string
  default     = "web-app-subnet"
}

variable "instance_name" {
  description = "Nombre de la instancia de VM"
  type        = string
  default     = "web-server-1"
}

variable "machine_type" {
  description = "Tipo de máquina para la instancia"
  type        = string
  default     = "e2-medium"
}

variable "credentials_file" {
  description = "Ruta al archivo JSON de la cuenta de servicio"
```

```
  type        = string
}
```

terraform.tfvars

```
project_id       = "efraintest01"
credentials_file = "~/terraform-key.json"
region           = "us-central1"
zone             = "us-central1-a"
```

main.tf

```
terraform {
  required_providers {
    google = {
      source  = "hashicorp/google"
      version = "~> 6.0"
    }
  }
  required_version = ">= 1.7.0"
}

provider "google" {
  project     = var.project_id
  region      = var.region
  credentials = file(var.credentials_file)
}

# Módulo de red
module "network" {
  source       = "./modules/network"
  project_id   = var.project_id
  region       = var.region
  network_name = var.network_name
  subnet_name  = var.subnet_name
```

```
}

# Módulo de compute (instancia)
module "compute" {
  source        = "./modules/compute"
  project_id    = var.project_id
  zone          = var.zone
  instance_name = var.instance_name
  machine_type  = var.machine_type
  subnet_self_link = module.network.subnet_self_link
}
```

outputs.tf

```
output "vpc_name" {
  value = module.network.network_name
}

output "instance_external_ip" {
  value = module.compute.web_server_ip
}
```

modules/network/network.tf

```
resource "google_compute_network" "vpc_network" {
  name                    = var.network_name
  auto_create_subnetworks = false
}

resource "google_compute_subnetwork" "vpc_subnet" {
  name          = var.subnet_name
  ip_cidr_range = "10.0.0.0/24"
  region        = var.region
  network       = google_compute_network.vpc_network.id
}
```

```
resource "google_compute_firewall" "allow_https" {
  name    = "allow-https"
  network = google_compute_network.vpc_network.name

  allow {
    protocol = "tcp"
    ports    = ["443"]
  }

  source_ranges = ["0.0.0.0/0"]
  target_tags   = ["web-server"]
}

output "network_name" {
  description = "Nombre de la red VPC creada"
  value       = google_compute_network.vpc_network.name
}

output "subnet_name" {
  description = "Nombre de la subred creada"
  value       = google_compute_subnetwork.vpc_subnet.name
}

output "subnet_self_link" {
  description = "Self link de la subred"
  value       = google_compute_subnetwork.vpc_subnet.self_link
}
```

modules/network/variables.tf

```
variable "project_id" {}
variable "region" {}
```

```
variable "network_name" {}
variable "subnet_name" {}
```

modules/compute/compute.tf

```
resource "google_compute_instance" "web_server" {
  name         = var.instance_name
  machine_type = var.machine_type
  zone         = var.zone

  tags = ["web-server"]

  boot_disk {
    initialize_params {
      image = "ubuntu-os-cloud/ubuntu-2204-lts"
    }
  }

  network_interface {
    subnetwork = var.subnet_self_link

    access_config {
      # Esto asigna una IP pública automáticamente
    }
  }

  metadata = {
    ssh-keys = "tux:${file("~/.ssh/id_rsa.pub")}"
  }
}
```

modules/compute/variables.tf

```
variable "project_id" {}
variable "zone" {}
```

```
variable "instance_name" {}
variable "machine_type" {}
variable "subnet_self_link" {}
```

Terraform init

```
$ cd gcptest/
$ ls
 main.tf     'Prueba técnica administrador cloud.docx'
 modules      terraform.tfvasrs
 outputs.tf   variables.tf
$ terraform init

# output
Initializing the backend...
Initializing modules...
- compute in modules/compute
- network in modules/network
Initializing provider plugins...
- Finding hashicorp/google versions matching "~> 6.0"...
- Installing hashicorp/google v6.50.0...
- Installed hashicorp/google v6.50.0 (signed by HashiCorp)
Terraform has created a lock file .terraform.lock.hcl to record the provider
selections it made above. Include this file in your version control repository
so that Terraform can guarantee to make the same selections by default when
you run "terraform init" in the future.

Terraform has been successfully initialized!

You may now begin working with Terraform. Try running "terraform plan" to s
ee
any changes that are required for your infrastructure. All Terraform commands
should now work.

If you ever set or change modules or backend configuration for Terraform,
```

rerun this command to reinitialize your working directory. If you forget, other commands will detect it and remind you to do so if necessary.

Terraform validate

```
$ terraform validate

# output
Success! The configuration is valid.
```

Terraform plan

```
$ terraform plan

# output

Terraform used the selected providers to generate the following execution pla
n. Resource actions are indicated with the following
symbols:
  + create

Terraform will perform the following actions:

  # module.compute.google_compute_instance.web_server will be created
  + resource "google_compute_instance" "web_server" {
      + can_ip_forward       = false
      + cpu_platform         = (known after apply)
      + creation_timestamp   = (known after apply)
      + current_status       = (known after apply)
      + deletion_protection  = false
      + effective_labels     = {
          + "goog-terraform-provisioned" = "true"
        }
      + id                   = (known after apply)
```

```
      + instance_id         = (known after apply)
      + label_fingerprint   = (known after apply)
      + machine_type        = "e2-medium"
      + metadata            = {
          + "ssh-keys" = <<EOT
              tux:ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAACAQDLYydQW7Wd
nxQeR1lbI2OpBLcgByD5EfYngWsGaaEdYvWeWT4yVEr0URLOGPUGFY8VK6U
pXZlXqA5HCE15laGV/kTxf7PwT2zTd5w1zSrCX+/dc3A88gcNFi8JKeUzZvPf12
W7NJkJ3Qz81KPTxkgprC8G/PExtkDeeWhaRjxExtorxOqeRVPw5wvAOF2fVjrc
ZgpH+SPDLWBriphD1YKEgcUvR3mO0w7gN1ClNMwjGLaqsWq45k+UNqC7kk
gZ9X+VBF1qsCliC4tfUinOiyuSKS34L13Nysnked8jm38N15Bqqw24hAUPGsOP
MdbBmqKG+QAAsmqCbCKEqS2xFwMIyYZNC2dYU/gXgmnt4LbOQBMkPJ+jq
zoRqyrBsz5LKwQHwGEWf8oU1wA195MeD1i/iRD7+mbXT6u1Nk7UbownBISSB
1/tExMYiEEjQu7HfabIK1ZL7lCRHUHoblxRxBWuEwPFjN4nxsrFfXTREXwRlitJNd
AINVjC/bWwK/cJBKVpioTXqmSzegnTuBbMBPd9q4vvhDM3RrFpOmFxF0xfsS
aivW9r2i3Nmfm3W+WFtjx4bQpk9d9TPZ+uI8QWVF+fA+c38mJnmL/+6znxrw
2LpW38H+z3DWM4rowC2BOjCrQSn9pzpcZGfu0C5fsyNCbi6VfRnVd9gV/uqq
eR75SmBw== tux@efraintest01
              EOT
          }
      + metadata_fingerprint = (known after apply)
      + min_cpu_platform     = (known after apply)
      + name                 = "web-server-1"
      + project              = "efraintest01"
      + self_link            = (known after apply)
      + tags                 = [
          + "web-server",
        ]
      + tags_fingerprint     = (known after apply)
      + terraform_labels     = {
          + "goog-terraform-provisioned" = "true"
        }
      + zone                 = "us-central1-a"

      + boot_disk {
          + auto_delete                = true
```

```
      + device_name              = (known after apply)
      + disk_encryption_key_sha256 = (known after apply)
      + guest_os_features        = (known after apply)
      + kms_key_self_link        = (known after apply)
      + mode                     = "READ_WRITE"
      + source                   = (known after apply)

      + initialize_params {
          + architecture         = (known after apply)
          + image                = "ubuntu-os-cloud/ubuntu-2204-lts"
          + labels               = (known after apply)
          + provisioned_iops     = (known after apply)
          + provisioned_throughput = (known after apply)
          + resource_policies    = (known after apply)
          + size                 = (known after apply)
          + snapshot             = (known after apply)
          + type                 = (known after apply)
        }
    }

  + confidential_instance_config (known after apply)

  + guest_accelerator (known after apply)

  + network_interface {
      + internal_ipv6_prefix_length = (known after apply)
      + ipv6_access_type           = (known after apply)
      + ipv6_address               = (known after apply)
      + name                       = (known after apply)
      + network                    = (known after apply)
      + network_attachment         = (known after apply)
      + network_ip                 = (known after apply)
      + stack_type                 = (known after apply)
      + subnetwork                 = (known after apply)
      + subnetwork_project         = (known after apply)
```

```
      + access_config {
          + nat_ip       = (known after apply)
          + network_tier = (known after apply)
        }
    }

  + reservation_affinity (known after apply)

  + scheduling (known after apply)
}

# module.network.google_compute_firewall.allow_https will be created
+ resource "google_compute_firewall" "allow_https" {
  + creation_timestamp = (known after apply)
  + description        = "Allow HTTPS traffic from anywhere"
  + destination_ranges = (known after apply)
  + direction          = "INGRESS"
  + enable_logging     = (known after apply)
  + id                 = (known after apply)
  + name               = "allow-https"
  + network            = "web-app-vpc"
  + priority           = 1000
  + project            = "efraintest01"
  + self_link          = (known after apply)
  + source_ranges      = [
      + "0.0.0.0/0",
    ]

  + allow {
      + ports    = [
          + "443",
        ]
      + protocol = "tcp"
    }
}
```

```
# module.network.google_compute_firewall.allow_ssh will be created
+ resource "google_compute_firewall" "allow_ssh" {
    + creation_timestamp = (known after apply)
    + description        = "Allow SSH access to instances with tag ssh-server"
    + destination_ranges = (known after apply)
    + direction          = "INGRESS"
    + enable_logging     = (known after apply)
    + id                 = (known after apply)
    + name               = "allow-ssh-webapp"
    + network            = "web-app-vpc"
    + priority           = 1000
    + project            = "efraintest01"
    + self_link          = (known after apply)
    + source_ranges      = [
        + "0.0.0.0/0",
      ]
    + target_tags        = [
        + "ssh-server",
      ]

    + allow {
        + ports    = [
            + "22",
          ]
        + protocol = "tcp"
      }
  }

# module.network.google_compute_network.web_app_vpc will be created
+ resource "google_compute_network" "web_app_vpc" {
    + auto_create_subnetworks                 = false
    + bgp_always_compare_med                  = (known after apply)
    + bgp_best_path_selection_mode            = (known after apply)
    + bgp_inter_region_cost                   = (known after apply)
    + delete_default_routes_on_create         = false
    + gateway_ipv4                            = (known after apply)
```

```
      + id                                     = (known after apply)
      + internal_ipv6_range                    = (known after apply)
      + mtu                                    = (known after apply)
      + name                                   = "web-app-vpc"
      + network_firewall_policy_enforcement_order = "AFTER_CLASSIC_FIREWA
LL"
      + network_id                             = (known after apply)
      + numeric_id                             = (known after apply)
      + project                                = "efraintest01"
      + routing_mode                           = (known after apply)
      + self_link                              = (known after apply)
    }

  # module.network.google_compute_subnetwork.web_app_subnet will be cre
ated
  + resource "google_compute_subnetwork" "web_app_subnet" {
      + creation_timestamp         = (known after apply)
      + enable_flow_logs           = (known after apply)
      + external_ipv6_prefix       = (known after apply)
      + fingerprint                = (known after apply)
      + gateway_address            = (known after apply)
      + id                         = (known after apply)
      + internal_ipv6_prefix       = (known after apply)
      + ip_cidr_range              = "10.0.0.0/24"
      + ipv6_cidr_range            = (known after apply)
      + ipv6_gce_endpoint          = (known after apply)
      + name                       = "web-app-subnet"
      + network                    = (known after apply)
      + private_ip_google_access   = (known after apply)
      + private_ipv6_google_access = (known after apply)
      + project                    = "efraintest01"
      + purpose                    = (known after apply)
      + region                     = "us-central1"
      + self_link                  = (known after apply)
      + stack_type                 = (known after apply)
      + state                      = (known after apply)
```

```
      + subnetwork_id             = (known after apply)

      + secondary_ip_range (known after apply)
    }

Plan: 5 to add, 0 to change, 0 to destroy.

Changes to Outputs:
  + instance_external_ip = (known after apply)
  + vpc_name             = "web-app-vpc"
```

Terraform apply

```
```

Ingresar al servidor por ssh

```
$ gcloud compute ssh --zone "us-central1-a" "web-server-1" --project "efrain
test01"

# output
Updating project ssh metadata...⠧Updated [https://www.googleapis.com/com
pute/v1/projects/efraintest01].
Updating project ssh metadata...done.
Waiting for SSH key to propagate.
Warning: Permanently added 'compute.4285412229480452973' (ED25519) to
the list of known hosts.
Welcome to Ubuntu 22.04.5 LTS (GNU/Linux 6.8.0-1041-gcp x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:     https://landscape.canonical.com
 * Support:        https://ubuntu.com/pro

 System information as of Sat Oct  4 23:28:47 UTC 2025
```

```
  System load:  0.82              Processes:             119
  Usage of /:   22.5% of 9.51GB   Users logged in:       0
  Memory usage: 7%                IPv4 address for ens4: 10.0.0.2
  Swap usage:   0%


Expanded Security Maintenance for Applications is not enabled.

0 updates can be applied immediately.

Enable ESM Apps to receive additional future security updates.
See https://ubuntu.com/esm or run: sudo pro status

New release '24.04.3 LTS' available.
Run 'do-release-upgrade' to upgrade to it.


Last login: Sat Oct  4 23:28:49 2025 from 190.158.28.104
```

Actualizar el sistema

```
tux@web-server-1:~$ sudo apt update -y

# output

Hit:1 http://us-central1.gce.archive.ubuntu.com/ubuntu jammy InRelease
Get:2 http://us-central1.gce.archive.ubuntu.com/ubuntu jammy-updates InRel
ease [128 kB]
Get:3 http://us-central1.gce.archive.ubuntu.com/ubuntu jammy-backports InR
elease [127 kB]
Get:4 http://security.ubuntu.com/ubuntu jammy-security InRelease [129 kB]
Get:5 http://us-central1.gce.archive.ubuntu.com/ubuntu jammy/universe amd6
4 Packages [14.1 MB]
Get:6 http://security.ubuntu.com/ubuntu jammy-security/main amd64 c-n-f M
etadata [13.9 kB]
```

```
Get:7 http://security.ubuntu.com/ubuntu jammy-security/universe amd64 Pac
kages [997 kB]
Get:8 http://us-central1.gce.archive.ubuntu.com/ubuntu jammy/universe Trans
lation-en [5652 kB]
Get:9 http://security.ubuntu.com/ubuntu jammy-security/universe Translation-
en [219 kB]
Get:10 http://security.ubuntu.com/ubuntu jammy-security/universe amd64 c-n
-f Metadata [22.1 kB]
Get:11 http://security.ubuntu.com/ubuntu jammy-security/multiverse amd64 P
ackages [56.9 kB]
Get:12 http://security.ubuntu.com/ubuntu jammy-security/multiverse Translati
on-en [11.9 kB]
Get:13 http://security.ubuntu.com/ubuntu jammy-security/multiverse amd64 c
-n-f Metadata [520 B]
Get:14 http://us-central1.gce.archive.ubuntu.com/ubuntu jammy/universe amd
64 c-n-f Metadata [286 kB]
Get:15 http://us-central1.gce.archive.ubuntu.com/ubuntu jammy/multiverse a
md64 Packages [217 kB]
Get:16 http://us-central1.gce.archive.ubuntu.com/ubuntu jammy/multiverse Tr
anslation-en [112 kB]
Get:17 http://us-central1.gce.archive.ubuntu.com/ubuntu jammy/multiverse am
d64 c-n-f Metadata [8372 B]
Get:18 http://us-central1.gce.archive.ubuntu.com/ubuntu jammy-updates/main
amd64 Packages [2978 kB]
Get:19 http://us-central1.gce.archive.ubuntu.com/ubuntu jammy-updates/main
Translation-en [464 kB]
Get:20 http://us-central1.gce.archive.ubuntu.com/ubuntu jammy-updates/mai
n amd64 c-n-f Metadata [19.0 kB]
Get:21 http://us-central1.gce.archive.ubuntu.com/ubuntu jammy-updates/restr
icted Translation-en [859 kB]
Get:22 http://us-central1.gce.archive.ubuntu.com/ubuntu jammy-updates/univ
erse amd64 Packages [1230 kB]
Get:23 http://us-central1.gce.archive.ubuntu.com/ubuntu jammy-updates/univ
erse Translation-en [306 kB]
Get:24 http://us-central1.gce.archive.ubuntu.com/ubuntu jammy-updates/univ
erse amd64 c-n-f Metadata [29.6 kB]
```

```
Get:25 http://us-central1.gce.archive.ubuntu.com/ubuntu jammy-updates/mult
iverse amd64 Packages [57.6 kB]
Get:26 http://us-central1.gce.archive.ubuntu.com/ubuntu jammy-updates/mult
iverse Translation-en [13.2 kB]
Get:27 http://us-central1.gce.archive.ubuntu.com/ubuntu jammy-updates/mult
iverse amd64 c-n-f Metadata [600 B]
Get:28 http://us-central1.gce.archive.ubuntu.com/ubuntu jammy-backports/m
ain amd64 Packages [69.4 kB]
Get:29 http://us-central1.gce.archive.ubuntu.com/ubuntu jammy-backports/m
ain Translation-en [11.5 kB]
Get:30 http://us-central1.gce.archive.ubuntu.com/ubuntu jammy-backports/m
ain amd64 c-n-f Metadata [412 B]
Get:31 http://us-central1.gce.archive.ubuntu.com/ubuntu jammy-backports/re
stricted amd64 c-n-f Metadata [116 B]
Get:32 http://us-central1.gce.archive.ubuntu.com/ubuntu jammy-backports/u
niverse amd64 Packages [30.1 kB]
Get:33 http://us-central1.gce.archive.ubuntu.com/ubuntu jammy-backports/u
niverse Translation-en [16.6 kB]
Get:34 http://us-central1.gce.archive.ubuntu.com/ubuntu jammy-backports/u
niverse amd64 c-n-f Metadata [672 B]
Get:35 http://us-central1.gce.archive.ubuntu.com/ubuntu jammy-backports/m
ultiverse amd64 c-n-f Metadata [116 B]
Fetched 28.2 MB in 5s (5207 kB/s)
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
All packages are up to date.
```

Instalar nginx

```
$ sudo apt install -y nginx

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

```
Reading state information... Done
The following additional packages will be installed:
  fontconfig-config fonts-dejavu-core libdeflate0 libfontconfig1 libgd3 libjbig0 l
ibjpeg-turbo8 libjpeg8 libnginx-mod-http-geoip2
  libnginx-mod-http-image-filter libnginx-mod-http-xslt-filter libnginx-mod-ma
il libnginx-mod-stream libnginx-mod-stream-geoip2
  libtiff5 libwebp7 libxpm4 nginx-common nginx-core
Suggested packages:
  libgd-tools fcgiwrap nginx-doc ssl-cert
The following NEW packages will be installed:
  fontconfig-config fonts-dejavu-core libdeflate0 libfontconfig1 libgd3 libjbig0 l
ibjpeg-turbo8 libjpeg8 libnginx-mod-http-geoip2
  libnginx-mod-http-image-filter libnginx-mod-http-xslt-filter libnginx-mod-ma
il libnginx-mod-stream libnginx-mod-stream-geoip2
  libtiff5 libwebp7 libxpm4 nginx nginx-common nginx-core
0 upgraded, 20 newly installed, 0 to remove and 0 not upgraded.
Need to get 2694 kB of archives.
After this operation, 8346 kB of additional disk space will be used.
```

Crear el archivo index.html

```
$ cd /var/www/html
tux@web-server-1:/var/www/html$ ls -ali
total 12
258413 drwxr-xr-x 2 root root 4096 Oct  4 23:38 .
258412 drwxr-xr-x 3 root root 4096 Oct  4 23:38 ..
258142 -rw-r--r-- 1 root root  612 Oct  4 23:38 index.nginx-debian.html
tux@web-server-1:/var/www/html$ sudo vi index.html

$ cat index.html

# output
<title> Welcome! </title>
</head>
<body>
```

```
<h1> "¡Bienvenido a la prueba técnica de GCP!" </h1>
</body>
</html>
```

Acceder desde cualquier parte

Crear un certificado para el servicio ngnix

```
sudo mkdir -p /etc/nginx/ssl
sudo openssl req -x509 -nodes -days 365 \
  -newkey rsa:2048 \
  -keyout /etc/nginx/ssl/nginx.key \
  -out /etc/nginx/ssl/nginx.crt \
  -subj "/CN=136.112.215.32"
```

Relacionar el certificado en el servicio

```
$ sudo vi /etc/nginx/sites-available/default

$ cat /etc/nginx/sites-available/default | grep -v "#"

server {
listen 443 ssl;

ssl_certificate     /etc/nginx/ssl/nginx.crt;
ssl_certificate_key /etc/nginx/ssl/nginx.key;


    root /var/www/html;

    index index.html index.htm index.nginx-debian.html;

    server_name _;
```

```
    location / {
        try_files $uri $uri/ =404;
    }


}
```

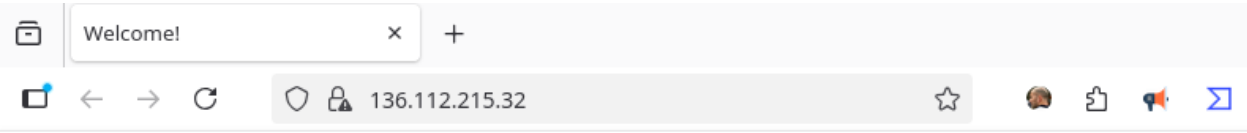
Reiniciar el servicio de ngnix

```
tux@web-server-1:~$ sudo systemctl restart nginx.service
```

Probar el acceso externo al servidor web

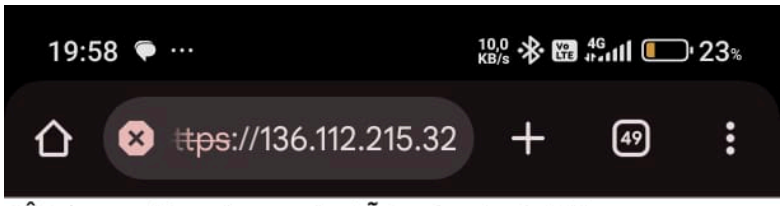
```
$ curl -k https://136.112.215.32
<title> Welcome! </title>
</head>
<body>
<h1> "¡Bienvenido a la prueba técnica de GCP!" </h1>
</body>
</html>
```

Prueba desde el navegador en Desktop



"Â¡Bienvenido a la prueba tÃ©cnica de GCP!"

Prueba desde el móvil



"Â¡Bienvenido a la prueba tÃ©cnica de GCP!"