Terraform - Example sheet

White boxes are examples. Red boxes are examples with more real example information!

Contents

General

```
provider "google" {
  project = "your-gcp-project-id"
  region  = "europe-north1"
}
```

Compute engine

```
resource "google_compute_instance" "example" {
  name         = "example-instance"
  machine_type = "e2-medium"
  zone         = "europe-north1-a"

  boot_disk {
    initialize_params {
      image = "debian-9-stretch-v20191210"
    }
  }

  network_interface {
    network = "default"
    access_config {
      // Allocate a public IP address
    }
  }
}
```

```
resource "google_compute_instance" "web-server" {
  name         = "web-server-instance"
  machine_type = "e2-medium"
  zone         = "europe-north1-a"

  boot_disk {
    initialize_params {
      image = "ubuntu-2004-focal-v20211216"
    }
  }

  network_interface {
    network = "default"
    access_config {
      // Allocate a public IP address
    }
  }

  metadata = {
    startup-script = "#! /bin/bash\n apt-get update\n apt-get install -y 
apache2\n systemctl start apache2\n"
  }
}
```

Storage Bucket

```
resource "google_storage_bucket" "example_bucket" {
  name     = "my-example-bucket"
  location = "EUROPE-NORTH1"

  website {
    main_page_suffix = "index.html"
    not_found_page   = "404.html"
  }
}
```

```
resource "google_storage_bucket" "landing_page_bucket" {
  name     = "my-landing-page-bucket-xyz123"
  location = "EUROPE-NORTH1"

  website {
    main_page_suffix = "index.html"
    not_found_page   = "404.html"
  }
}

resource "google_storage_object" "index_html" {
  name   = "index.html"
  bucket = google_storage_bucket.landing_page_bucket.name
  content = <<EOF
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Welcome to My Landing Page</title>
</head>
<body>
  <h1>Welcome to My Landing Page</h1>
  <p>This is a simple static website hosted on Google Cloud Storage using
Terraform!</p>
</body>
</html>
EOF
}
```

VPC

```
resource "google_compute_network" "default" {
  name = "default-vpc"
}

resource "google_compute_subnetwork" "default" {
  name          = "default-subnet"
  network       = google_compute_network.default.name
  region        = "europe-north1"
  ip_cidr_range = "10.0.0.0/24"
}
```

```
resource "google_compute_network" "default" {
  name = "my-vpc-network"
}

resource "google_compute_subnetwork" "default" {
  name          = "my-subnet"
  network       = google_compute_network.default.name
  region        = "europe-north1"
  ip_cidr_range = "10.0.0.0/24"
}
```

Cloud SQL

```
resource "google_sql_database_instance" "example" {
  name   = "example-db-instance"
  region = "europe-north1"

  database_version = "MYSQL_5_7"
  tier             = "db-f1-micro"

  root_password = "your-password"

  settings {
    ip_configuration {
      authorized_networks {
        name  = "example-network"
        value = "0.0.0.0/0"
      }
    }
  }
}
```

```
resource "google_sql_database_instance" "my-db-instance" {
  name   = "my-db-instance"
  region = "europe-north1"

  database_version = "MYSQL_5_7"
  tier             = "db-f1-micro"

  root_password = "SuperSecurePassword123!"

  settings {
    ip_configuration {
      authorized_networks {
        name  = "allowed-network"
        value = "0.0.0.0/0"
      }
    }
  }
}

resource "google_sql_database" "my_database" {
  name     = "mydatabase"
  instance = google_sql_database_instance.my-db-instance.name
}
```

IAM Role Binding

```
resource "google_project_iam_member" "example" {
  project = "your-gcp-project-id"
  role    = "roles/viewer"
  member  = "user:example-user@example.com"
}
```

```
resource "google_project_iam_member" "viewer_binding" {
  project = "my-gcp-project-id"
  role    = "roles/viewer"
  member  = "user:john.doe@example.com"
}
```

Load Balancer

```
resource "google_compute_backend_service" "example" {
  name        = "example-backend"
  protocol    = "HTTP"
  port_name   = "http"
  backends {
    group = google_compute_instance_group.example.self_link
  }
}

resource "google_compute_url_map" "example" {
  name = "example-url-map"

  default_url_redirect {
    https_redirect = false
    strip_query    = false
    prefix_redirect = "http://example.com"
  }
}

resource "google_compute_target_http_proxy" "example" {
  name    = "example-http-proxy"
  url_map = google_compute_url_map.example.id
}

resource "google_compute_global_forwarding_rule" "example" {
  name        = "example-http-rule"
  target      = google_compute_target_http_proxy.example.id
  port_range  = "80"
  ip_address  = google_compute_global_address.example.address
}
```

```
resource "google_compute_backend_service" "web-backend" {
  name        = "web-backend-service"
  protocol    = "HTTP"
  port_name   = "http"
  backends {
    group = google_compute_instance_group.web-server-group.self_link
  }
}

resource "google_compute_url_map" "web-url-map" {
  name = "web-url-map"

  default_url_redirect {
    https_redirect = false
    strip_query    = false
    prefix_redirect = "http://mywebsite.com"
  }
}

resource "google_compute_target_http_proxy" "http-proxy" {
  name    = "http-proxy"
  url_map = google_compute_url_map.web-url-map.id
}

resource "google_compute_global_forwarding_rule" "http-forwarding-rule" {
  name        = "http-forwarding-rule"
  target      = google_compute_target_http_proxy.http-proxy.id
  port_range  = "80"
  ip_address  = google_compute_global_address.web-ip.address
}
```

Firewall Rule

```
resource "google_compute_firewall" "http" {
  name    = "allow-http"
  network = "default"

  allow {
    protocol = "tcp"
    ports    = ["80"]
  }

  source_ranges = ["0.0.0.0/0"]
}
```

```
resource "google_compute_firewall" "allow-http" {
  name    = "allow-http-rule"
  network = "default"

  allow {
    protocol = "tcp"
    ports    = ["80"]
  }

  source_ranges = ["0.0.0.0/0"]
}
```

Identity Aware Proxy

```
resource "google_iap_web_backend_service" "example" {
  name = "example-iap-backend"
  backend_service = google_compute_backend_service.example.id
}

resource "google_iap_web" "example" {
  app_engine = google_iap_web_backend_service.example.id
}
```

```
resource "google_iap_web_backend_service" "iap-backend-service" {
  name = "iap-backend-service"
  backend_service = google_compute_backend_service.web-backend.id
}

resource "google_iap_web" "iap-web" {
  app_engine = google_iap_web_backend_service.iap-backend-service.id
}
```

VPN

```
resource "google_compute_vpn_gateway" "example" {
  name    = "example-vpn-gateway"
  network = google_compute_network.default.name
  region  = "europe-north1"
}

resource "google_compute_vpn_tunnel" "example" {
  name              = "example-vpn-tunnel"
  vpn_gateway       = google_compute_vpn_gateway.example.id
  peer_ip           = "peer-vpn-ip"
  shared_secret     = "your-shared-secret"
  region            = "europe-north1"
  target_vpn_gateway = google_compute_vpn_gateway.example.id
}
```

```
resource "google_compute_vpn_gateway" "vpn-gateway" {
  name    = "vpn-gateway"
  network = google_compute_network.default.name
  region  = "europe-north1"
}

resource "google_compute_vpn_tunnel" "vpn-tunnel" {
  name              = "vpn-tunnel"
  vpn_gateway       = google_compute_vpn_gateway.vpn-gateway.id
  peer_ip           = "203.0.113.1"
  shared_secret     = "SuperSecureSecret!"
  region            = "europe-north1"
  target_vpn_gateway = google_compute_vpn_gateway.vpn-gateway.id
}
```

Full Example

VPC network with a SQL database and a compute engine web server. Fire wall rule configured to allow HTTP to the web server.

```
provider "google" {
  project = "my-gcp-project-id"
  region  = "europe-north1"
}

# VPC and Subnet Setup
resource "google_compute_network" "default" {
  name = "my-vpc-network"
}

resource "google_compute_subnetwork" "default" {
  name          = "my-subnet"
  network       = google_compute_network.default.name
  region        = "europe-north1"
  ip_cidr_range = "10.0.0.0/24"
}

# Cloud SQL Instance
resource "google_sql_database_instance" "my-db-instance" {
  name   = "my-db-instance"
  region = "europe-north1"

  database_version = "MYSQL_5_7"
  tier             = "db-f1-micro"

  root_password = "SuperSecurePassword123!"

  settings {
    ip_configuration {
      authorized_networks {
        name  = "allowed-network"
        value = "0.0.0.0/0"
      }
    }
  }
}

# Cloud SQL Database
resource "google_sql_database" "my_database" {
  name     = "mydatabase"
  instance = google_sql_database_instance.my-db-instance.name
}

# Firewall Rule to allow HTTP access to the Compute Engine
resource "google_compute_firewall" "allow-http" {
  name    = "allow-http-rule"
  network = google_compute_network.default.name

  allow {
    protocol = "tcp"
    ports    = ["80"]
  }

  source_ranges = ["0.0.0.0/0"]
}

# Compute Engine Instance
resource "google_compute_instance" "web-server" {
  name         = "web-server-instance"
  machine_type = "e2-medium"
  zone         = "europe-north1-a"

  boot_disk {
    initialize_params {
      image = "ubuntu-2004-focal-v20211216"
    }
  }

  network_interface {
    network = google_compute_network.default.name
    subnetwork = google_compute_subnetwork.default.name
    access_config {
      // Allocate a public IP address
    }
  }

  metadata = {
    startup-script = <<-EOF
      #!/bin/bash
      apt-get update
      apt-get install -y apache2
      apt-get install -y mysql-client
      systemctl start apache2
      systemctl enable apache2

      # Configure the application to connect to the Cloud SQL database
      DB_HOST=${google_sql_database_instance.my-db-
instance.ip_address[0].ip_address}
      DB_USER=root
      DB_PASSWORD=SuperSecurePassword123!
      echo "Database Host: $DB_HOST" > /var/www/html/index.html
    EOF
  }

  tags = ["web-server"]
}

# Output for Cloud SQL connection details
output "db_host" {
  value = google_sql_database_instance.my-db-instance.ip_address[0].ip_address
}

output "compute_engine_ip" {
  value = google_compute_instance.web-
server.network_interface[0].access_config[0].nat_ip
}
```