100 Real-Time Terraform Use Cases

1. Provisioning an AWS EC2 Instance

```
provider "aws" {
  region = "us-west-2"
}  resource "aws_instance"
"example" {
  ami           = "ami-0c55b159cbfafe1f0"
  instance_type = "t2.micro"
}
```

Explanation: This configuration creates an EC2 instance using a specific Amazon Machine Image (AMI) and instance type.

2. Creating an AWS S3 Bucket

```
provider "aws" {
  region = "us-west-2"
}  resource "aws_s3_bucket"
"example" {
  bucket = "my-bucket"
  acl    = "private"
}
```

Explanation: This example sets up a private S3 bucket named "my-bucket".

3. Setting Up an AWS RDS Instance

```
provider "aws" {
  region = "us-west-2"
}  resource "aws_db_instance"
"example" {
  allocated_storage    = 20
  engine               = "mysql"
  engine_version       = "5.7"
  instance_class       = "db.t2.micro"
  name                 = "mydb"
  username             = "foo"
  password             = "bar"
  parameter_group_name = "default.mysql5.7"
}
```

Explanation: This example provisions a MySQL RDS instance with specific storage and instance class.

4. Deploying a Google Cloud Storage Bucket

```
provider "google" {
  project = "my-project-id"
  region = "us-central1"
}  resource "google_storage_bucket"
"example" { name  = "my-storage-bucket"
location    = "US" force_destroy = true
}
```

Explanation: Creates a Google Cloud Storage bucket with a specified name and location.

5. Creating an Azure Virtual Network

```
provider "azurerm" {
  features {}
}  resource "azurerm_virtual_network"
"example" { name  = "example-network"
  address_space   = ["10.0.0.0/16"] location
      = "East US"
  resource_group_name = azurerm_resource_group.example.name
}
```

Explanation: This example sets up a virtual network in Azure with a specified address space and location.

6. Provisioning an Azure Virtual Machine

```
provider "azurerm" {
  features {}
}  resource "azurerm_virtual_machine"
"example" { name  = "example-vm"
  location  = azurerm_resource_group.example.location
  resource_group_name = azurerm_resource_group.example.name
  network_interface_ids = [azurerm_network_interface.example.id]
  vm_size               = "Standard_DS1_v2"

  storage_os_disk {
    name              = "example-os-disk"
    caching           = "ReadWrite"
    create_option     = "FromImage"
    managed_disk_type = "Standard_LRS"
  }

  storage_image_reference {
    publisher = "Canonical"
    offer     = "UbuntuServer"
    sku       = "18.04-LTS"
    version   = "latest"
  }
 os_profile {
    computer_name    =    "examplevm"
    admin_username   =   "adminuser"
    admin_password = "P@ssw0rd1234!"
  } }
```

Explanation: Provisions a Linux virtual machine in Azure with specific VM size and OS disk.

7. Creating a Kubernetes Cluster with EKS

```
provider "aws" {
  region = "us-west-2"
}  module
"eks" {
  source          = "terraform-aws-modules/eks/aws"
  cluster_name    = "my-cluster"
  cluster_version = "1.17"
  subnets         = ["subnet-12345678", "subnet-87654321"]
  vpc_id          = "vpc-12345678"
}
```

Explanation: This module deploys an EKS (Elastic Kubernetes Service) cluster in AWS.

8. Deploying a Docker Container with ECS

```
provider "aws" {
  region = "us-west-2"
}  resource "aws_ecs_cluster"
"example" {
  name = "example-cluster"
}

resource "aws_ecs_task_definition" "example" {
  family    = "example"
  network_mode             = "bridge"
  requires_compatibilities = ["EC2"]
  cpu                      = "256"
  memory                   = "512"
  container_definitions    = <<DEFINITION
[
  {
    "name": "example",
    "image": "nginx",
    "memory": 512,
    "cpu": 256,
    "essential": true,
    "portMappings": [
      {
        "containerPort": 80,
        "hostPort": 80
      }
    ]
  }
]
DEFINITION
}
```

Explanation: Creates an ECS cluster and a task definition for a Docker container running NGINX.

9. Provisioning a Google Cloud SQL Instance

```
provider "google" {
  project = "my-project-id"
  region = "us-central1"
}  resource "google_sql_database_instance"
"example" { name  = "mysql-instance"
  database_version = "MYSQL_5_7"
  region           = "us-central1"
 settings {
    tier = "db-f1-micro"
  }
}
```

Explanation: Provisions a Google Cloud SQL instance running MySQL.

10. Creating an Azure Kubernetes Service (AKS) Cluster

```
provider "azurerm" {
  features {}
}  resource "azurerm_kubernetes_cluster"
"example" { name  = "example-aks"
  location  = azurerm_resource_group.example.location
  resource_group_name = azurerm_resource_group.example.name
  dns_prefix      = "exampleaks"

  default_node_pool {
    name       = "default"
    node_count = 1
    vm_size    = "Standard_DS2_v2"
  }
 identity {
    type = "SystemAssigned"
  }
}
```

Explanation: Creates an AKS cluster with a default node pool and system-assigned identity.

11. Provisioning a CloudFront Distribution

```
provider "aws" {
  region = "us-west-2"
}  resource "aws_cloudfront_distribution"
"example" {
  origin {
    domain_name = aws_s3_bucket.example.bucket_regional_domain_name
    origin_id = "S3-example"
  }

  enabled              = true
  is_ipv6_enabled      = true
  comment              = "Some comment"
  default_root_object = "index.html"
 default_cache_behavior {
    target_origin_id    = "S3-example"
    viewer_protocol_policy = "allow-all"
    allowed_methods     = ["GET", "HEAD", "OPTIONS"]
    cached_methods      = ["GET", "HEAD"]
    forwarded_values {
      query_string = false
      cookies {
        forward = "none"
      }
    }

    min_ttl                = 0
    default_ttl            = 3600
    max_ttl                = 86400
  }
 price_class = "PriceClass_100"
 restrictions {
    geo_restriction {
      restriction_type = "none"
    }
  }
 viewer_certificate {
    cloudfront_default_certificate = true
  }
}
```

Explanation: Creates a CloudFront distribution with an S3 bucket origin.

12. Creating an IAM Role and Policy in AWS

```
provider "aws" {
  region = "us-west-2"
}
resource "aws_iam_role" "example" {
  name = "example-role"
 assume_role_policy = jsonencode({
    Version = "2012-10-17"
    Statement = [
      {
        Action = "sts:AssumeRole"
        Effect = "Allow"
        Principal = {
          Service = "ec2.amazonaws.com"
        }
      },
    ]
  })
}  resource "aws_iam_policy"
"example" { name  = "example-policy"
  description = "A test policy"
  policy      = jsonencode({
    Version = "2012-10-17"
    Statement = [
      {
        Action = [
          "ec2:Describe*",
        ]
        Effect = "Allow"
        Resource = "*"
      },
    ]
  })
}  resource "aws_iam_role_policy_attachment"
"example" {
  role       = aws_iam_role.example.name policy_arn
  = aws_iam_policy.example.arn
}
```

Explanation: This example creates an IAM role and a policy, then attaches the policy to the role.

13. Creating a VPC in AWS

```
provider "aws" {
  region = "us-west-2"
}  resource "aws_vpc"
"example" {
  cidr_block = "10.0.0.0/16"
}

resource "aws_subnet" "example" {
  vpc_id            = aws_vpc.example.id
  cidr_block      = "10.0.1.0/24" availability_zone
  = "us-west-2a"
}
```

Explanation: Creates a VPC and a subnet within that VPC.

14. **Provisioning a Google Compute Engine

Instance**

```
provider "google" {
  project = "my-project-id"
  region = "us-central1"
}  resource "google_compute_instance"
"example" { name  = "example-instance"
  machine_type = "n1-standard-1"
  zone      = "us-central1-a"
 boot_disk {
    initialize_params {
      image = "debian-cloud/debian-9"
    }
  }
 network_interface {
    network = "default"

    access_config {
    }
  } }
```

Explanation: Provisions a Google Compute Engine instance with a specified machine type and Debian image.

15. Creating an Azure Storage Account

```
provider "azurerm" {
  features {}
}  resource "azurerm_storage_account"
"example" {
  name                     = "examplestorageacc"
  resource_group_name   = azurerm_resource_group.example.name
  location  = azurerm_resource_group.example.location
  account_tier             = "Standard"
  account_replication_type = "LRS"
}
```

Explanation: Creates an Azure Storage Account with locally-redundant storage.

16. Creating a Lambda Function in AWS

```
provider "aws" {
  region = "us-west-2"
}  resource "aws_lambda_function"
"example" {
  function_name = "example_lambda"
  role          = aws_iam_role.example.arn
  handler       = "index.handler"
  runtime       = "nodejs12.x"
  filename  = "lambda_function_payload.zip" }
```

Explanation: Creates an AWS Lambda function using a ZIP file containing the function's code.

17. Deploying a DigitalOcean Droplet

```
provider "digitalocean" {
  token = "your_api_token"
}  resource "digitalocean_droplet"
"example" {
  name = "example-droplet"
  region = "nyc3" size = "s-
  1vcpu-1gb" image =
  "ubuntu-20-04-x64"
}
```

Explanation: Provisions a DigitalOcean Droplet in the NYC3 region with Ubuntu 20.04.

18. Creating an AWS Elastic Beanstalk Application

```
provider "aws" {
  region = "us-west-2"
}  resource "aws_elastic_beanstalk_application"
"example" { name  = "example-app"
  description = "An example Elastic Beanstalk application"
}
```

Explanation: Sets up an Elastic Beanstalk application.

19. Provisioning a Google Kubernetes Engine (GKE) Cluster

```
provider "google" {
  project = "my-project-id"
  region = "us-central1"
}  resource "google_container_cluster"
"example" {
  name      = "example-gke-cluster"
  location = "us-central1"
  initial_node_count = 3

  node_config {
    machine_type = "n1-standard-1"
  }
}
```

Explanation: Creates a GKE cluster with three initial nodes of type `n1-standard-1`.

20. Setting Up an Azure SQL Database

```
provider "azurerm" {
  features {}
}  resource "azurerm_sql_server"
"example" {
  name                         = "example-sqlserver"
  resource_group_name          = azurerm_resource_group.example.name
  location                     = azurerm_resource_group.example.location
  version                      = "12.0"
  administrator_login          = "adminuser"
  administrator_login_password =
"H@Sh1CoR3!" }  resource
"azurerm_sql_database" "example" { name   =
"example-db"
  resource_group_name = azurerm_resource_group.example.name
  location  = azurerm_resource_group.example.location
  server_name         = azurerm_sql_server.example.name
  edition             = "Basic"
}
```

Explanation: Provisions an Azure SQL Server and a SQL database within it.

21. Deploying a Google Cloud Function

```
provider "google" {
  project = "my-project-id"
  region = "us-central1"
}  resource "google_cloudfunctions_function"
"example" { name  = "example-function"
  description = "An example Cloud Function" runtime   = "nodejs10"
  entry_point = "helloWorld" source_archive_bucket =
  google_storage_bucket.example.name source_archive_object =
  google_storage_bucket_object.example.name trigger_http = true
}
```

Explanation: Creates a Google Cloud Function triggered via HTTP.

22. Creating an AWS SNS Topic

```
provider "aws" {
  region = "us-west-2"
}  resource "aws_sns_topic"
"example" {
  name = "example-topic"
}
```

Explanation: Provisions an SNS topic for messaging in AWS.

23. Provisioning an AWS EFS File System

```
provider "aws" {
  region = "us-west-2"
}
resource "aws_efs_file_system" "example" {
  creation_token = "example-token"
}
```

Explanation: Sets up an Elastic File System (EFS) in AWS.

24. Creating a Google Cloud Pub/Sub Topic

```
provider "google" {
  project = "my-project-id"
  region = "us-central1"
}  resource "google_pubsub_topic"
"example" {
  name = "example-topic"
}
```

Explanation: Creates a Pub/Sub topic in Google Cloud.

25. Setting Up an Azure Load Balancer

```
provider "azurerm" {
  features {}
}  resource "azurerm_lb"
"example" {
  name                = "example-lb"
  location = azurerm_resource_group.example.location
  resource_group_name = azurerm_resource_group.example.name
  sku                 = "Basic"
  frontend_ip_configuration {
    name                 = "publicIPAddress"
    public_ip_address_id = azurerm_public_ip.example.id
  }
}
```

Explanation: Creates an Azure Load Balancer with a public IP address.

26. Provisioning a Google Cloud Armor Security Policy

```
provider "google" {
  project = "my-project-id"
  region = "us-central1"
}  resource "google_compute_security_policy"
"example" {
  name = "example-policy"
 rule {
    action = "allow"
    match {
      versioned_expr = "SRC_IPS_V1"
      config {
        src_ip_ranges = ["0.0.0.0/0"]
      }
    }
  }
}
```

Explanation: Creates a security policy in Google Cloud Armor.

27. Creating an AWS ElastiCache Cluster

```
provider "aws" {
  region = "us-west-2"
}  resource "aws_elasticache_cluster"
"example" { cluster_id  = "example-cluster"
  engine               = "redis"
  node_type            = "cache.t2.micro"
  num_cache_nodes      = 1
  parameter_group_name = "default.redis3.2"
}
```

Explanation: Provisions an ElastiCache cluster running Redis.

28. Deploying a Google Cloud Run Service

```
provider "google" {
  project = "my-project-id"
  region = "us-central1"
}  resource "google_cloud_run_service"
"example" {
  name      = "example-service"
  location = "us-central1"
 template {
    spec {
      containers {
        image = "gcr.io/cloudrun/hello"
      }
    }
  }
 traffic {
    percent         = 100
    latest_revision = true
  }
}
```

Explanation: Creates a Cloud Run service running a container image.

29. Setting Up an Azure DNS Zone

```
provider "azurerm" {
  features {}
}  resource "azurerm_dns_zone"
"example" { name  = "example.com"
  resource_group_name = azurerm_resource_group.example.name
}
```

Explanation: Creates a DNS zone in Azure.

30. Creating a Google Cloud Spanner Instance

```
provider "google" {
  project = "my-project-id"
  region = "us-central1"
}  resource "google_spanner_instance"
"example" { name  = "example-instance" config
      = "regional-us-central1"
  display_name = "Example Instance"
  node_count     = 1
  processing_units = 100
}
```

Explanation: Provisions a Spanner instance in Google Cloud.

31. Provisioning an AWS Route 53 Hosted Zone

```
provider "aws" {
  region = "us-west-2"
}  resource "aws_route53_zone"
"example" {
  name = "example.com"
}
```

Explanation: Creates a hosted zone in AWS Route 53.

32. Deploying a Google Cloud Dataflow Job

```
provider "google" {
  project = "my-project-id"
  region = "us-central1"
}  resource "google_dataflow_job"
"example" {
  name     = "example-job"
  template_gcs_path = "gs://dataflow-templates/latest/Word_Count"
  parameters = {
    inputFile = "gs://
 dataflow-
samples/shakespeare/kinglear.txt" output =
"gs://example-bucket/output"
  }
}
```

Explanation: Creates a Dataflow job to run a word count template.

33. Setting Up an Azure Application Gateway

```
provider "azurerm" {
  features {}
}  resource "azurerm_application_gateway"
"example" { name  = "example-app-gateway"
  location  = azurerm_resource_group.example.location
  resource_group_name = azurerm_resource_group.example.name
  sku {
    name     = "Standard_Small"
    tier     = "Standard"
    capacity = 2
  }
  gateway_ip_configuration {
    name    = "app-gateway-ip-config" subnet_id =
    azurerm_subnet.example.id
  }
  frontend_ip_configuration {
    name                 = "app-gateway-frontend-ip"
    public_ip_address_id = azurerm_public_ip.example.id
  }
  frontend_port
  {
    name = "app-gateway-frontend-port"
    port = 80
  }
  backend_address_pool {
    name = "app-gateway-backend-pool"
  }
  backend_http_settings {
    name                  = "app-gateway-http-settings"
    cookie_based_affinity = "Disabled"
    port                  = 80
    protocol              = "Http"
    request_timeout       = 20
  }
  http_listener {
    name                           = "app-gateway-http-listener"
    frontend_ip_configuration_name = "app-gateway-frontend-ip"
    frontend_port_name = "app-gateway-frontend-port"
    protocol                       = "Http"
  }
  url_path_map
  {
    name = "app-gateway-url-path-map"
    default_backend_address_pool_name = "app-gateway-backend-pool"
    default_backend_http_settings_name = "app-gateway-http-settings"
  }
}
```

Explanation: Creates an Azure Application Gateway with frontend and backend configurations.

34. Creating an AWS Glue Job

```
provider "aws" {
  region = "us-west-2"
}  resource "aws_glue_job" "example"
{ name      = "example-job" role_arn
= aws_iam_role.example.arn
 command {
    script_location = "s3://my-script-bucket/glue-script.py"
    python_version = "3"
  }
}
```

Explanation: Provisions an AWS Glue job for ETL operations.

35. Deploying a Google Cloud Composer Environment

```
provider "google" {
  project = "my-project-id"
  region = "us-central1"
}  resource "google_composer_environment"
"example" {
  name      = "example-environment"
  region = "us-central1"
  config {
    node_count = 3
    software_config {
      image_version = "composer-1.10.0-airflow-1.10.9"
    }
  }
}
```

Explanation: Creates a Cloud Composer environment for running Apache Airflow.

36. Setting Up an Azure Container Instance

```
provider "azurerm" {
  features {}
}  resource "azurerm_container_group"
"example" {
  name      = "example-container-group" location      =
  azurerm_resource_group.example.location resource_group_name
  = azurerm_resource_group.example.name
  os_type             = "Linux"
 container {
    name = "example-container" image
    = "nginx"
    cpu     = "0.5"
    memory = "1.5"
 ports {
      port     = 80
      protocol = "TCP"
    }
  }

  ip_address {
    ports {
      port     = 80
      protocol = "TCP"
    } type =
    "Public"
  } }
```

Explanation: Creates an Azure Container Instance with a public IP address running NGINX.

37. Provisioning an AWS CodeBuild Project

```
provider "aws" {
  region = "us-west-2"
}
resource "aws_codebuild_project" "example" {
  name      = "example-codebuild"
  description = "Example CodeBuild project" service_role
  = aws_iam_role.example.arn
 artifacts {
    type = "NO_ARTIFACTS"
  }
  environment {

    compute_type                = "BUILD_GENERAL1_SMALL"
    image                       = "aws/codebuild/standard:4.0"
    type                        = "LINUX_CONTAINER"
    environment_variable    {
    name    =    "EXAMPLE_ENV"
    value = "example-value" }

  }
  source {

    type            = "GITHUB"
    location        = "https://github.com/hashicorp/terraform"
    buildspec       = "buildspec.yml"
  }
}
```

Explanation: Creates a CodeBuild project that builds a GitHub repository.

38. Creating a Google Cloud Storage Transfer Service Job

```
provider "google" {
  project = "my-project-id"
  region = "us-central1"
}  resource "google_storage_transfer_job"
"example" {
  description = "Transfer job from AWS to Google Cloud"
  project     = "my-project-id"
  transfer_spec {
    gcs_data_sink {
      bucket_name = google_storage_bucket.example.name
    }
    aws_s3_data_source            {
      bucket_name = "source-bucket"
      aws_access_key {
        access_key_id   = "your-access-key" secret_access_key
        = "your-secret-key"
      }
    }
  }
  schedule
  {
    schedule_start_date {
      year = 2021
      month = 12
      day = 31
    }
  }
  status = "ENABLED"
}
```

Explanation: Creates a Storage Transfer Service job to transfer data from an AWS S3 bucket to a Google Cloud Storage bucket.

39. Provisioning an AWS OpsWorks Stack

```
provider "aws" {
  region = "us-west-2"
}  resource "aws_opsworks_stack"
"example" { name  = "example-stack"
  service_role_arn = aws_iam_role.example.arn
  default_instance_profile_arn =
  aws_iam_instance_profile.example.arn
```

```
 configuration_manager {
    name    = "Chef" version
    = "12"
  }
 custom_json = <<EOF
{
  "opsworks": {
    "stack": {
      "instance_config": {
        "instance_count": 2
      }
    }
  } }
EOF
}
```

Explanation: Provisions an OpsWorks stack using Chef for configuration management.

40. Deploying a Google Cloud VPN Tunnel

```
provider "google" {
  project = "my-project-id"
  region = "us-central1"
}  resource "google_compute_vpn_tunnel"
"example" { name  = "example-vpn-tunnel"
  region                = "us-central1"
  target_vpn_gateway    =
google_compute_target_vpn_gateway.example.self_link
  peer_ip               = "35.192.0.2"
  shared_secret         = "my-shared-secret"
  ike_version           = 2
}  resource "google_compute_target_vpn_gateway"
"example" {
  name      = "example-target-vpn-gateway" network
  = google_compute_network.example.self_link region
  = "us-central1"
}
```

Explanation: Creates a VPN tunnel in Google Cloud, connecting to an external IP.

41. Setting Up an Azure Synapse Analytics Workspace

```
provider "azurerm" {
  features {}
}  resource "azurerm_synapse_workspace"
"example" {
  name                 = "example-synapse-workspace"
  resource_group_name = azurerm_resource_group.example.name
  location  = azurerm_resource_group.example.location
  storage_data_lake_gen2_filesystem_id =
azurerm_storage_data_lake_gen2_filesystem.example.id
  sql_administrator_login     = "sqladminuser"
  sql_administrator_login_password   = "H@Sh1CoR3!" identity {
    type = "SystemAssigned"
  } }
```

Explanation: Creates an Azure Synapse Analytics Workspace with system-assigned managed identity.

42. Provisioning a Google Cloud Memorystore Instance

```
provider "google" {
  project = "my-project-id"
  region = "us-central1"
}  resource "google_redis_instance"
"example" { name  = "example-instance"
  memory_size_gb = 1
  tier           = "BASIC"
}
```

Explanation: Creates a Google Cloud Memorystore instance for Redis.

43. Creating an AWS CloudWatch Log Group

```
provider "aws" {
  region = "us-west-2"
}  resource "aws_cloudwatch_log_group"
"example" { name  = "example-log-group"
  retention_in_days = 14
}
```

Explanation: Provisions a CloudWatch Log Group with a 14-day retention policy.

44. Deploying a Google Cloud NAT Gateway

```
provider "google" {
  project = "my-project-id"
  region = "us-central1"
} resource "google_compute_router"

"example" { name  = "example-router"

  network = google_compute_network.example.self_link
  region = "us-central1"
}  resource "google_compute_router_nat"
"example" { name  =

 "example-nat" router =
  google_compute_router.example.name region =
  "us-central1"

  nat_ip_allocate_option = "AUTO_ONLY"
  source_subnetwork_ip_ranges_to_nat = "ALL_SUBNETWORKS_ALL_IP_RANGES"
}
```

Explanation: Creates a NAT Gateway using a Google Compute Router.

45. Provisioning an AWS Glue Crawler

```
provider "aws" {
  region = "us-west-2"
}  resource "aws_glue_crawler"
"example" { name  = "example-crawler"
  database_name = "example-db"
  role      = aws_iam_role.example.arn s3_target {
    path = "s3://example-bucket"
  }
}
```

Explanation: Sets up a Glue Crawler to catalog data stored in S3.

46. Creating a Google Cloud Bigtable Instance

```
provider "google" {
  project = "my-project-id"
  region = "us-central1"
}  resource "google_bigtable_instance"
"example" { name  = "example-bigtable-
instance" cluster_id    = "example-cluster"
cluster_zone = "us-central1-a"
cluster_num_nodes = 3 cluster_storage_type =
"SSD"
}
```

Explanation: Creates a Bigtable instance with an SSD storage cluster.

47. Setting Up an Azure Redis Cache

```
provider "azurerm" {
  features {}
}  resource "azurerm_redis_cache"
"example" { name  = "example-redis-cache"
  location  = azurerm_resource_group.example.location
  resource_group_name = azurerm_resource_group.example.name
  capacity            = 1
  family              = "C"
  sku_name            = "Standard"
}
```

Explanation: Provisions an Azure Redis Cache.

48. Creating an AWS ElasticSearch Domain

```
provider "aws" {
  region = "us-west-2"
}  resource "aws_elasticsearch_domain"
"example" { domain_name       = "example-
domain"
  elasticsearch_version = "7.10" cluster_config
  {
    instance_type = "m5.large.elasticsearch"
  }
}
```

Explanation: Creates an ElasticSearch domain in AWS.

49. Deploying a Google Cloud VPC Network

```
provider "google" {
  project = "my-project-id"
  region = "us-central1"
}

resource "google_compute_network" "example" { name
      = "example-network"
  auto_create_subnetworks = true
}
```

Explanation: Creates a VPC network in Google Cloud.

50. Setting Up an Azure Bastion Host

```
provider "azurerm" {
  features {}
}  resource "azurerm_bastion_host"
"example" { name  = "example-bastion"
  location  = azurerm_resource_group.example.location
  resource_group_name = azurerm_resource_group.example.name
  dns_name            = "example-bastion"
  sku                 = "Basic"
  ip_configuration {
    name                 = "configuration"
    subnet_id            = azurerm_subnet.example.id
    public_ip_address_id = azurerm_public_ip.example.id
  }
}
```

Explanation: Provisions an Azure Bastion Host for secure RDP and SSH access.

51. Creating an AWS SSM Parameter Store Parameter

```
provider "aws" {
  region = "us-west-2"
}  resource "aws_ssm_parameter"
"example" {
  name  =  "example-parameter"
  type  =  "String"  value  =
  "example-value"
}
```

Explanation: Creates an SSM Parameter Store parameter to store configuration data.

52. Deploying a Google Cloud Interconnect

```
provider "google" {
  project = "my-project-id"
  region = "us-central1"
}  resource "google_compute_interconnect"
"example" { name  = "example-interconnect"
  location           = "us-central1"
  interconnect_type = "DEDICATED" link_type     =
  "LINK_TYPE_ETHERNET_10G_LR"
  requested_link_count = 1 administrative_status
  = "ACTIVE"
}
```

Explanation: Sets up a dedicated Interconnect connection in Google Cloud.

53. Creating an AWS CodePipeline

```
provider "aws" {
  region = "us-west-2"
}  resource "aws_codepipeline"
"example" { name  = "example-pipeline"
role_arn = aws_iam_role.example.arn
```

```
  artifact_store { location =
    aws_s3_bucket.example.bucket type
      = "S3"
  }
 stage {
    name = "Source"
    action {

      name             = "Source"
      category         = "Source"
      owner            = "AWS"
      provider         = "S3"
      version          = "1"
      output_artifacts = ["source_output"]
      configuration = {
        S3Bucket = aws_s3_bucket.example.bucket
        S3ObjectKey = "source.zip"
      }
    }
  }
 stage {
    name = "Deploy"

    action {

      name             = "Deploy"
      category         = "Deploy"
      owner            = "AWS"
      provider         = "CodeDeploy"
      version          = "1"
      input_artifacts = ["source_output"]
      configuration = {
        ApplicationName = aws_codedeploy_app.example.name
        DeploymentGroupName = aws_codedeploy_deployment_group.example.name
      }
    }
  } }
```

Explanation: Provisions a CodePipeline to automate deployment using CodeDeploy.

54. Setting Up an Azure Logic App

```
provider "azurerm" {
  features {}
}  resource "azurerm_logic_app_workflow"
"example" { name  = "example-logic-app"
  location  = azurerm_resource_group.example.location
  resource_group_name = azurerm_resource_group.example.name

  definition = <<DEFINITION
{
  "definition": {
    "$schema":
"https://schema.management.azure.com/providers/Microsoft.Logic/schemas/201
6
-06-01/workflowdefinition.json#",
    "contentVersion": "1.0.0.0",
    "triggers": {
      "manual": {
        "type": "Request",
        "kind": "http",
        "inputs": {
          "schema": {}
        }
      }
    },
    "actions": {
      "response": {
        "type": "Response",
        "inputs": {
          "statusCode": 200,
          "body": "Hello, World!"
        }
      }
    }
  }
}
DEFINITION
}
```

Explanation: Creates an Azure Logic App with a simple HTTP request trigger and response action.

55. Creating a Google Cloud SQL User

```
provider "google" {
  project = "my-project-id"
  region = "us-central1"
}  resource "google_sql_user"
"example" {
  instance =
  google_sql_database_instance.example.name name
      = "example-user" password = "example-password"
}
```

Explanation: Creates a user for a Google Cloud SQL instance.

56. Provisioning an AWS WAF WebACL

```
provider "aws" {
  region = "us-west-2"
}  resource "aws_wafv2_web_acl"
"example" { name  = "example-web-acl"
  scope       = "REGIONAL"
  description = "Example Web ACL"
  default_action {
    allow {}
  }
 rule {
    name     = "example-rule"
    priority = 1
    action {
      block {}
    }
    statement
    {
      byte_match_statement {
        search_string = "bad-bot"
        field_to_match {
          single_header {
            name = "User-Agent"
          }
        }
        positional_constraint = "CONTAINS"
        text_transformations {
          priority = 1
          type     = "NONE"
        }
      }
    }
    visibility_config {
      cloudwatch_metrics_enabled = true
      metric_name                = "example-rule"
      sampled_requests_enabled = true
    }
  }
 visibility_config {
    cloudwatch_metrics_enabled = true
    metric_name                = "example-web-acl"
    sampled_requests_enabled = true
  }
}
```

Explanation: Creates an AWS WAF Web ACL with a rule to block requests with a specific User-Agent header.

57. Deploying a Google Cloud Filestore Instance

```
provider "google" {
  project = "my-project-id"
  region = "us-central1"
}  resource "google_filestore_instance"
"example" {
  name       = "example-filestore"
  tier       = "STANDARD"
  file_shares {
    capacity_gb = 1024
    name        = "example-share"
  }
  networks
  {
    network = "default"
  }
}
```

Explanation: Provisions a Google Cloud Filestore instance with a 1 TB file share.

58. Creating an AWS Kinesis Stream

```
provider "aws" {
  region = "us-west-2"
}

resource "aws_kinesis_stream" "example" {
  name      = "example-stream"
  shard_count = 1
}
```

Explanation: Provisions an

AWS Kinesis Stream with one shard.

59. Setting Up an Azure Data Factory

```
provider "azurerm" {
  features {}
}
resource "azurerm_data_factory" "example" {
  name                = "example-data-factory"
  location  = azurerm_resource_group.example.location
  resource_group_name = azurerm_resource_group.example.name
}
```

Explanation: Creates an Azure Data Factory instance.

60. Provisioning a Google Cloud IAM Service Account

```
provider "google" {
  project = "my-project-id"
  region = "us-central1"
}  resource "google_service_account"
"example" {
  account_id = "example-account" display_name =
  "Example Service Account"
}
```

Explanation: Creates a Google Cloud IAM service account.

61. Creating an AWS CloudFormation Stack

```
provider "aws" {
  region = "us-west-2"
}  resource "aws_cloudformation_stack"
"example" { name  = "example-stack"
  template_body = file("cloudformation_template.json")
}
```

Explanation: Provisions an AWS CloudFormation stack using a JSON template.

62. Deploying a Google Cloud IAM Policy Binding

```
provider "google" {
  project = "my-project-id"
  region = "us-central1"
}  resource "google_project_iam_binding"
"example" {
  project = "my-project-id"
  role      = "roles/editor"
  members = [
   "serviceAccount:example-account@my-project-id.iam.gserviceaccount.com",
  ] }
```

Explanation: Creates a policy binding to grant the editor role to a service account in Google Cloud.

63. Setting Up an Azure Key Vault

```
provider "azurerm" {
  features {}
}  resource "azurerm_key_vault"
"example" {
  name                = "example-key-vault"
  location  = azurerm_resource_group.example.location
  resource_group_name = azurerm_resource_group.example.name
  tenant_id       = "your-tenant-id"
  sku_name            = "standard"
  access_policy {
    tenant_id = "your-tenant-id"
    object_id = "your-object-id"
    key_permissions = [
      "get",
    ]
  }
}
```

Explanation: Provisions an Azure Key Vault with access policies.

64. Creating a Google Cloud Build Trigger

```
provider "google" {
  project = "my-project-id"
  region = "us-central1"
}  resource "google_cloudbuild_trigger"
"example" {
  filename = "cloudbuild.yaml" trigger_template
  {
    project_id = "my-project-id"
    branch_name     =     "main"
    repo_name = "example-repo"
  } }
```

Explanation: Sets up a Google Cloud Build trigger to start builds on changes to the main branch of a repository.

65. Provisioning an AWS SageMaker Notebook Instance

```
provider "aws" {
  region = "us-west-2"
} resource "aws_sagemaker_notebook_instance"
"example" {
  name                 = "example-notebook"
  instance_type        = "ml.t2.medium"
  role_arn             = aws_iam_role.example.arn
  subnet_id            = aws_subnet.example.id
  security_group_ids   = [aws_security_group.example.id]
}
```

Explanation: Creates a SageMaker Notebook Instance for machine learning development.

66. Creating a Google Cloud SQL SSL Cert

```
provider "google" {
  project = "my-project-id"
  region = "us-central1"
}  resource "google_sql_ssl_cert"
"example" {
  instance = google_sql_database_instance.example.name
  common_name = "example-cert"
}
```

Explanation: Provisions an SSL certificate for a Google Cloud SQL instance.

67. Setting Up an Azure HDInsight Cluster

```
provider "azurerm" {
  features {}
}

resource "azurerm_hdinsight_hadoop_cluster" "example" {
  name      = "example-hdinsight"
  location  = azurerm_resource_group.example.location
  resource_group_name = azurerm_resource_group.example.name
  cluster_version       = "3.6"
  tier                = "Standard"
  gateway {
    enabled    = true
    username   = "adminuser"
    password   = "H@Sh1CoR3!"
  }
  storage_accounts {
    storage_account_key =
azurerm_storage_account.example.primary_access_key
    storage_container_id = azurerm_storage_container.example.id
  }
  roles
  {
    head_node {
      vm_size = "Standard_D3_V2"
    }
    worker_node
    {
      vm_size = "Standard_D3_V2"
      target_instance_count = 3
    }
  }
}
```

Explanation: Creates an Azure HDInsight Hadoop Cluster with specified configurations.

68. Deploying a Google Cloud BigQuery Dataset

```
provider "google" {
  project = "my-project-id"
  region = "us-central1"
}

resource "google_bigquery_dataset" "example" {
  dataset_id     =     "example_dataset"
  friendly_name   =   "Example   Dataset"
  description = "A dataset for examples"
  location = "US"
}
```

Explanation: Provisions a BigQuery dataset in Google Cloud.

69. Creating an AWS IAM Group and User

```
provider "aws" {
  region = "us-west-2"
}

resource "aws_iam_group" "example" {
  name = "example-group"
}  resource "aws_iam_user"
"example" {
  name = "example-user"
}  resource "aws_iam_group_membership"
"example" {
  name  =  "example-group-membership"
  users = [aws_iam_user.example.name]
  group = aws_iam_group.example.name
}
```

Explanation: Creates an IAM group, a user, and adds the user to the group.

70. Setting Up an Azure Event Hub

```
provider "azurerm" {
  features {}
}  resource "azurerm_eventhub_namespace"
"example" {
  name                = "example-eventhub-namespace"
  location  = azurerm_resource_group.example.location
  resource_group_name = azurerm_resource_group.example.name
  sku                 = "Standard"
}  resource "azurerm_eventhub"
"example" {
  name                = "example-eventhub"
  namespace_name      = azurerm_eventhub_namespace.example.name
  resource_group_name = azurerm_resource_group.example.name
  partition_count       = 2
  message_retention = 1
}
```

Explanation: Provisions an Event Hub namespace and an Event Hub.

71. Deploying a Google Cloud Endpoints API

```
provider "google" {
  project = "my-project-id"
  region = "us-central1"
}  resource "google_endpoints_service"
"example" {
  name      = "example-api.endpoints.my-project-id.cloud.goog"
  openapi_config = file("openapi.yaml")
}
```

Explanation: Creates an API managed by Google Cloud Endpoints using an OpenAPI configuration.

72. Creating an AWS RDS Snapshot

```
provider "aws" {
  region = "us-west-2"
}  resource "aws_db_snapshot"
"example" {
  db_instance_identifier = aws_db_instance.example.id
  db_snapshot_identifier = "example-snapshot"
}
```

Explanation: Creates a snapshot of an RDS instance.

73. Setting Up an Azure SQL Managed Instance

```
provider "azurerm" {
  features {}
}

resource "azurerm_sql_managed_instance" "example" {
  name          = "example-sqlmi"
  location = azurerm_resource_group.example.location resource_group_name =
  azurerm_resource_group.example.name
  administrator_login          = "adminuser"
  administrator_login_password = "H@Sh1CoR3!"
  sku_name                     = "GP_Gen5_2"
  storage_size_in_gb           = 128
  subnet_id                    = azurerm_subnet.example.id
  public_data_endpoint_enabled = false
}
```

Explanation: Provisions an Azure SQL Managed Instance with specified configurations.

74. Provisioning a Google Cloud Logging Metric

```
provider "google" {
  project = "my-project-id"
  region = "us-central1"
}

resource "google_logging_metric" "example" {
  name        = "example-metric"
  description = "A metric to count occurrences of a specific log" filter
      = "resource.type=\"gce_instance\" AND logName=\"projects/my-
project-id/logs/example-log\""
  metric_descriptor {
    metric_kind = "DELTA"
    value_type = "INT64"
    unit        = "1"
  }
}
```

Explanation: Creates a Cloud Logging metric to count specific log entries.

75. Creating an AWS Elastic File System Mount Target

```
provider "aws" {
  region = "us-west-2"
}  resource "aws_efs_mount_target" "example" {
file_system_id = aws_efs_file_system.example.id
subnet_id   = aws_subnet.example.id
  security_groups = [

aws_security_group.example.id]
}
```

Explanation: Provisions an EFS mount target in a specified subnet with a security group.

76. Setting Up an Azure Cosmos DB Account

```
provider "azurerm" {
  features {}
}  resource "azurerm_cosmosdb_account"
"example" {
  name      = "example-cosmosdb-account" location     =
  azurerm_resource_group.example.location resource_group_name
  = azurerm_resource_group.example.name
  offer_type          = "Standard"
  kind                = "GlobalDocumentDB"
  consistency_policy {
    consistency_level = "Session"
  }
}
```

Explanation: Provisions an Azure Cosmos DB account with session consistency.

77. Deploying a Google Cloud Scheduler Job

```
provider "google" {
  project = "my-project-id"
  region = "us-central1"
}  resource "google_cloud_scheduler_job"
"example" {
  name      = "example-scheduler-job"
  schedule = "*/5 * * * *"
  http_target {
    uri     = "https://example.com/cron" http_method =
    "POST"
  }
}
```

Explanation: Creates a Cloud Scheduler job to invoke an HTTP endpoint every 5 minutes.

78. Creating an AWS Auto Scaling Group

```
provider "aws" {
  region = "us-west-2"
}  resource "aws_autoscaling_group"
"example" {
  availability_zones = ["us-west-2a"]
  desired_capacity      = 2 max_size
      = 3 min_size      = 1
  launch_configuration = aws_launch_configuration.example.id
}
```

Explanation: Provisions an Auto Scaling group with a specified launch configuration.

79. Setting Up an Azure Data Lake Storage Gen2

```
provider "azurerm" {
  features {}
}  resource "azurerm_storage_account"
"example" {
  name                     = "examplestorageacc"
  resource_group_name   = azurerm_resource_group.example.name
  location  = azurerm_resource_group.example.location
  account_tier             = "Standard"
  account_replication_type = "LRS"
  is_hns_enabled           = true
}
```

Explanation: Creates a Storage Account with hierarchical namespace enabled for Data Lake Storage Gen2.

80. Provisioning a Google Cloud DNS Managed Zone

```
provider "google" {
  project = "my-project-id"
  region = "us-central1"
}

resource "google_dns_managed_zone" "example" {
  name      = "example-zone"
  dns_name = "example.com."
}
```

Explanation: Creates a managed DNS zone in Google Cloud DNS.

81. Creating an AWS SES Email Identity

```
provider "aws" {
  region = "us-west-2"
}  resource "aws_ses_domain_identity"
"example" {
  domain = "example.com"
}
```

Explanation: Provisions an SES email identity for a domain.

82. Setting Up an Azure Stream Analytics Job

```
provider "azurerm" {
  features {}
}  resource "azurerm_stream_analytics_job"
"example" {
  name      = "example-stream-analytics-job" location       =
  azurerm_resource_group.example.location resource_group_name
  = azurerm_resource_group.example.name
  streaming_units       = 3
  output {
    name    = "example-output" data_type
    = "AzureBlob"
    resource_id = azurerm_storage_account.example.id
    storage_container      =      "output-container"
    path_pattern = "{date}/{time}"
  }
}
```

Explanation: Creates an Azure Stream Analytics job with output to a Blob storage container.

83. Deploying a Google Cloud DNS Record Set

```
provider "google" {
  project = "my-project-id"
  region = "us-central1"
}

resource "google_dns_record_set" "example" {
  name  =  "www.example.com."  type  =  "A"  ttl  =  300
  managed_zone = google_dns_managed_zone.example.name
  rrdatas = ["192.0.2.1"]
}
```

Explanation: Creates a DNS A record for a domain in a managed zone.

84. Creating an AWS Batch Compute Environment

```
provider "aws" {
  region = "us-west-2"
}

resource "aws_batch_compute_environment" "example" {
  compute_environment_name = "example-batch-environment"
  service_role    = aws_iam_role.example.arn
  type                       = "MANAGED"
  compute_resources {
    instance_role = aws_iam_instance_profile.example.arn
    instance_types = ["m4.large"]
    max_vcpus       = 16
    min_vcpus       = 0
    security_group_ids = [aws_security_group.example.id]
    subnets         = [aws_subnet.example.id]
    type            = "EC2"
  } }
```

Explanation: Provisions an AWS Batch compute environment with specified instance types and compute resources.

85. Setting Up an Azure Function App

```
provider "azurerm" {
  features {}
}  resource "azurerm_function_app"
"example" {
  name                = "example-function-app"
  location  = azurerm_resource_group.example.location
  resource_group_name = azurerm_resource_group.example.name
  app_service_plan_id = azurerm_app_service_plan.example.id
  storage_account_name = azurerm_storage_account.example.name
  storage_account_access_key =
azurerm_storage_account.example.primary_access_key
}
```

Explanation: Creates an Azure Function App with a specified App Service plan and storage account.

86. Creating a Google Cloud Bigtable Table

```
provider "google" {
  project = "my-project-id"
  region = "us-central1"
}  resource "google_bigtable_table"
"example" { name  = "example-table"
  instance_name = google_bigtable_instance.example.name
  column_family {
    family = "cf1"
  }
}
```

Explanation: Provisions a Bigtable table with a specified column family.

87. Setting Up an AWS Secrets Manager Secret

```
provider "aws" {
  region = "us-west-2"
}  resource "aws_secretsmanager_secret"
"example" {
  name = "example-secret"
}  resource "aws_secretsmanager_secret_version"
"example" {
  secret_id      = aws_secretsmanager_secret.example.id
  secret_string =
"{\"username\":\"example_user\",\"password\":\"example_password\"}"
}
```

Explanation: Creates a Secrets Manager secret and a version with secret data.

88. Deploying a Google Cloud VPC Peering Connection

```
provider "google" {
  project = "my-project-id"
  region = "us-central1"
}  resource "google_compute_network_peering" "example"
{ name = "example-peering"
  network = google_compute_network.example.self_link
  peer_network = google_compute_network.peer_network.self_link
}
```

Explanation: Provisions a VPC peering connection between two networks.

89. Creating an AWS Redshift Cluster

```
provider "aws" {
  region = "us-west-2"
}  resource "aws_redshift_cluster"
"example" {
  cluster_identifier = "example-cluster"
  database_name      = "exampledb"
  master_username    = "admin"
  master_password    = "example-password"
  node_type          = "dc2.large"
  cluster_type       = "single-node"
}
```

Explanation: Creates a Redshift cluster with a specified database and credentials.

90. Setting Up an Azure SignalR Service

```
provider "azurerm" {
  features {}
}  resource "azurerm_signalr_service"
"example" { name  = "example-signalr"
  location  = azurerm_resource_group.example.location
  resource_group_name = azurerm_resource_group.example.name
  sku {
    name     = "Standard_S1"
    capacity = 1
  } }
```

Explanation: Provisions an Azure SignalR Service with a specified SKU and capacity.

91. Deploying a Google Cloud Storage Notification

```
provider "google" {
  project = "my-project-id"
  region = "us-central1"
}  resource "google_storage_notification"
"example" { bucket =
google_storage_bucket.example.name topic =
google_pubsub_topic.example.name event_types =
["OBJECT_FINALIZE"]
}
```

Explanation: Creates a Cloud Storage notification to trigger a Pub/Sub topic on object finalize events.

92. Creating an AWS Step Functions State Machine

```
provider "aws" {
  region = "us-west-2"
}  resource "aws_sfn_state_machine"
"example" {
  name      = "example-state-machine"
  role_arn = aws_iam_role.example.arn
  definition = <<DEFINITION
{
  "Comment": "A Hello World example of the Amazon States Language using a
Pass state",
  "StartAt": "HelloWorld",
  "States": {
    "HelloWorld": {
      "Type": "Pass",
      "Result": "Hello, World!",
      "End": true
```

```
 }
  }
}
DEFINITION
}
```

Explanation: Provisions a Step Functions state machine with a simple pass state.

93. Setting Up an Azure Log Analytics Workspace

```
provider "azurerm" {
  features {}
}  resource "azurerm_log_analytics_workspace"
"example" { name  = "example-log-analytics"
  location  = azurerm_resource_group.example.location
  resource_group_name = azurerm_resource_group.example.name
  sku                 = "PerGB2018"
}
```

Explanation: Creates an Azure Log Analytics Workspace with a specified SKU.

94. Provisioning a Google Cloud IAM Custom Role

```
provider "google" {
  project = "my-project-id"
  region = "us-central1"
}
resource "google_project_iam_custom_role" "example" {
  role_id   = "exampleCustomRole" title
      = "Example Custom Role"
  description = "A custom role with limited permissions"
  permissions = [
    "storage.objects.get",
    "storage.objects.list",
  ] }
```

Explanation: Creates a custom IAM role with specified permissions in Google Cloud.

95. Creating an AWS GuardDuty Detector

```
provider "aws" {
  region = "us-west-2"
}  resource "aws_guardduty_detector"
"example" {
  enable = true
}
```

Explanation: Provisions an AWS GuardDuty detector to monitor malicious activities.

96. Setting Up an Azure Kubernetes Service (AKS) Node Pool

```
provider "azurerm" {
  features {}
}  resource "azurerm_kubernetes_cluster_node_pool"
"example" { name  = "example-nodepool"
  kubernetes_cluster_id = azurerm_kubernetes_cluster.example.id
  vm_size = "Standard_DS2_v2" node_count = 3
}
```

Explanation: Creates a node pool for an existing AKS cluster.

97. Deploying a Google Cloud Logging Sink

```
provider "google" {
  project = "my-project-id"
  region = "us-central1"
}  resource "google_logging_project_sink"
"example" { name  = "example-sink"
  destination = "storage.googleapis.com/example-bucket"
  filter      = "logName=\"projects/my-project-id/logs/example-log\""
}
```

Explanation: Creates a logging sink to export logs to a Cloud Storage bucket.

98. Creating an AWS MQ Broker

```
provider "aws" {
  region = "us-west-2"
}
resource "aws_mq_broker" "example" {
  broker_name   =   "example-broker"
  engine_type      =      "ActiveMQ"
  engine_version      =     "5.15.6"
  host_instance_type = "mq.t2.micro"
  publicly_accessible = true users {
    username = "admin"
    password = "example-password"
  }
}
```

Explanation: Provisions an MQ broker with ActiveMQ engine.

99. Setting Up an Azure Traffic Manager Profile

```
provider "azurerm" {
  features {}
}  resource "azurerm_traffic_manager_profile"
"example" { name  = "example-traffic-manager"
  resource_group_name = azurerm_resource_group.example.name
  location            = "global"
  profile_status  = "Enabled" traffic_routing_method =
  "Performance"
}
```

Explanation: Creates an Azure Traffic Manager profile to distribute traffic based on performance.

100. Deploying a Google Cloud Memorystore for Memcached Instance

```
provider "google" {
  project = "my-project-id"
  region = "us-central1"
}  resource "google_memcache_instance"
"example" { name  = "example-memcached"
  node_count      = 1
  node_config {
    cpu_count = 1
    memory_size_mb = 1024
  } }
```

Explanation: Provisions a Memorystore for Memcached instance with specified CPU and memory configurations.