

Trabalho prático

- **Aluno:** Guilherme Rodriguez Vicentin
- **E-mail:** vicentingr@gmail.com
- **Link do projeto:** <https://github.com/gvicentin/inf0500/inf0550>

1. Introdução

Para a realização da atividade prática da disciplina de **Computação em Nuvem I**, o provedor de Cloud escolhido foi a Microsoft Azure.

Utilizando o e-mail institucional da Unicamp é possível adquirir *U\$ 100* de crédito com expiração após um ano.

Todo o código e os arquivos necessários para reproduzir o laboratório aqui apresentado, pode ser encontrado no GitHub da disciplina.

2. Configuração do sistema

Para criar e configurar as VMs de forma automatizada, iremos utilizar o HashiCorp Terraform.

Usando Infrastructure as Code (IaC), é possível experimentar mais e reduzir os custos, uma vez que podemos criar e remover os recursos da cloud sem intervenção manual. Além disso não precisaremos manualmente configurar e rodar os testes de benchmark.

2.1. Configurando a Autenticação

Primeiramente iremos realizar o login na Azure utilizando a CLI, assim poderemos autenticar o provider do Terraform posteriormente. Abaixo segue os passos tomados:

1. Instalar a Azure CLI.
2. Rodar o comando `az login`. Uma nova janela será aberta no navegador para ser realizado o login com uma conta Microsoft.

Agora, criaremos um par de chaves SSH diretamente na Azure e salvaremos localmente a chave privada.

1. `ssh_key_name`: Escolhemos aleatoriamente um nome para o recurso.
2. `ssh_public_key_gen`: Inicializamos a ação que criará um par de chaves.
3. `ssh_public_key`: Fazemos o **bind** entre a ação que cria uma chave e o recurso em si.
4. `private_key_file`: Salvamos o resultado da chave privada em um arquivo local.

Arquivo `infra/ssh.tf`:

```

resource "random_pet" "ssh_key_name" {
  prefix      = "ssh"
  separator   = ""
}

resource "azapi_resource_action" "ssh_public_key_gen" {
  type          = "Microsoft.Compute/sshPublicKeys@2022-11-01"
  resource_id   = azapi_resource.ssh_public_key.id
  action        = "generateKeyPair"
  method        = "POST"

  response_export_values = ["publicKey", "privateKey"]
}

resource "azapi_resource" "ssh_public_key" {
  type          = "Microsoft.Compute/sshPublicKeys@2022-11-01"
  name          = random_pet.ssh_key_name.id
  location      = azurerm_resource_group.rg.location
  parent_id     = azurerm_resource_group.rg.id
}

resource "local_sensitive_file" "private_key_file" {
  content        = azapi_resource_action.ssh_public_key_gen.output.privateKey
  filename       = "${path.module}/id_azure"
  file_permission = "0600"
}

```

2.2. Criando recursos de Rede

Antes de criarmos nossas VMs, precisamos criar alguns recursos de rede como Virtual Network e Subnet.

1. Virtual Network com um CIDR block primário de classe A (10.0.0.0/16).
2. Subnet utilizando o CIDR 10.0.1/24 (256 hosts máximos).

Arquivo infra/main.tf:

```

# Create virtual network
resource "azurerm_virtual_network" "my_terraform_network" {
  name          = "myVnet"
  address_space = ["10.0.0.0/16"]
  location      = azurerm_resource_group.rg.location
  resource_group_name = azurerm_resource_group.rg.name
}

# Create subnet
resource "azurerm_subnet" "my_terraform_subnet" {

```

```

    name                = "mySubnet"
    resource_group_name = azurerm_resource_group.rg.name
    virtual_network_name = azurerm_virtual_network.my_terraform_network.name
    address_prefixes     = ["10.0.1.0/24"]
}

```

2.3. Criando as VMs

Para cada VM, criaremos os seguintes recursos:

1. IP público que será utilizado para acessar a máquina utilizando SSH.
2. Security Group, onde serão especificadas as regras de Firewall a nível da instância (liberar porta 22 de qualquer IP).
3. NIC (Interface de rede) que será atribuída a instância.
4. Bind entre o Security Group (criado no item 2) e a interface de rede (criada no item 3).
5. Virtual Machine propriamente dita. Os recursos criados nos itens anteriores serão utilizados aqui.

Arquivo `infra/main.tf`:

```

# Create public IPs
resource "azurerm_public_ip" "my_terraform_public_ip" {
    name                = "myPublicIP"
    location            = azurerm_resource_group.rg.location
    resource_group_name = azurerm_resource_group.rg.name
    allocation_method   = "Dynamic"
}

# Create Network Security Group and rule
resource "azurerm_network_security_group" "my_terraform_nsg" {
    name                = "myNetworkSecurityGroup"
    location            = azurerm_resource_group.rg.location
    resource_group_name = azurerm_resource_group.rg.name

    security_rule {
        name                = "SSH"
        priority            = 1001
        direction           = "Inbound"
        access              = "Allow"
        protocol            = "Tcp"
        source_port_range   = "*"
        destination_port_range = "22"
        source_address_prefix = "*"
        destination_address_prefix = "*"
    }
}

```

```

# Create network interface
resource "azurerm_network_interface" "my_terraform_nic" {
  name                = "myNIC"
  location            = azurerm_resource_group.rg.location
  resource_group_name = azurerm_resource_group.rg.name

  ip_configuration {
    name                        = "my_nic_configuration"
    subnet_id                 = azurerm_subnet.my_terraform_subnet.id
    private_ip_address_allocation = "Dynamic"
    public_ip_address_id       = azurerm_public_ip.my_terraform_public_ip.id
  }
}

# Connect the security group to the network interface
resource "azurerm_network_interface_security_group_association" "example" {
  network_interface_id = azurerm_network_interface.my_terraform_nic.id
  network_security_group_id = azurerm_network_security_group.my_terraform_nsg.id
}

# Create virtual machine
resource "azurerm_linux_virtual_machine" "my_terraform_vm" {
  name                = "myVM"
  location            = azurerm_resource_group.rg.location
  resource_group_name = azurerm_resource_group.rg.name
  network_interface_ids = [azurerm_network_interface.my_terraform_nic.id]
  size                = "Standard_DS1_v2"

  os_disk {
    name                = "myOsDisk"
    caching             = "ReadWrite"
    storage_account_type = "Premium_LRS"
  }

  source_image_reference {
    publisher = "Canonical"
    offer     = "0001-com-ubuntu-server-jammy"
    sku       = "22_04-lts-gen2"
    version   = "latest"
  }

  computer_name = "hostname"
  admin_username = var.username

  admin_ssh_key {

```

```
    username    = var.username
    public_key = azapi_resource_action.ssh_public_key_gen.output.publicKey
  }

  boot_diagnostics {
    storage_account_uri = azurerm_storage_account.my_storage_account.primary_blob_endpoint
  }
}
```

3. Benchmarks escolhidos

4. Execução dos benchmarks

5. Resultados e discussão

6. Conclusão