



**UNIVERSIDADE DO SUL DE SANTA CATARINA**  
**FILIPPE COUTINHO DOS SANTOS**

**INFRAESTRUTURA COMO CÓDIGO:**  
**UTILIZAÇÃO DO TERRAFORM PARA DEPLOY DE INFRAESTRUTURA EM**  
**NUVEM**

Palhoça  
2020

FILIPPE COUTINHO DOS SANTOS

**INFRAESTRUTURA COMO CÓDIGO:  
UTILIZAÇÃO DO TERRAFORM PARA DEPLOY DE INFRAESTRUTURA EM  
NUVEM**

Relatório apresentado ao Curso **Tecnólogo em Gestão da Tecnologia da Informação**, da Universidade do Sul de Santa Catarina, como requisito parcial à aprovação na unidade de aprendizagem de Estudo de Caso.

Orientador: Roberto Fabiano Fernandes

Palhoça  
2020

FILIPPE COUTINHO DOS SANTOS

**INFRAESTRUTURA COMO CÓDIGO:  
UTILIZAÇÃO DO TERRAFORM PARA DEPLOY DE INFRAESTRUTURA EM  
NUVEM**

Este trabalho de pesquisa na modalidade de Estudo de Caso foi julgado adequado à obtenção do grau de Tecnólogo em Gestão da Tecnologia da Informação e aprovado, em sua forma final, pelo Curso Superior de Tecnologia em Gestão da Tecnologia da Informação, da Universidade do Sul de Santa Catarina.

Palhoça, 25 de maio de 2020.

---

Prof. e orientador Roberto Fabiano Fernandes, Ciência da computação, Especialista em Engenharia e Projetos de Software  
Universidade do Sul de Santa Catarina

## **AGRADECIMENTOS**

Ao meu pai, Benigno que me mostrou que tudo que queremos na vida só pode ser conquistado com suor e dedicação, à minha mãe Cibele que tanto me educou e batalhou para eu poder estar hoje fazendo este estudo, a minha esposa Manuela e ao meu filho Vicente por suportarem e me apoiarem em tantas horas de estudos mesmo que para isso exija abdicar de estar com eles essas horas.

## RESUMO

Este estudo de caso visa apresentar um exemplo prático e simples de como utilizar a ferramenta Terraform® para criar uma infraestrutura básica de servidor e rede virtual em nuvem pública e a levantar os próximos passos necessários para evoluir a utilização da ferramenta para melhor aproveitar todo seu potencial.

A utilização da ferramenta se justifica pela crescente necessidade de agilidade na criação e mantimento de ambientes em nuvem pública, ambientes estes que passam por frequentes transformações e exigem resposta rápidas e simples, bem como por seu baixo investimento para utilização.

Para atingir seu objetivo este trabalho utiliza o método explicativo, trazendo um exemplo prático de como utilizar a ferramenta, e colocando um cenário real de uma empresa a prova, onde foi possível comprovar a eficácia da ferramenta no dia a dia da operação da área de implantações de novos projetos desta empresa.

Após a aplicação do questionário, foi possível compilar todas as respostas e comprovar a eficácia da ferramenta nos tempos de execução da área de projetos da empresa, e definir um plano de ação para os próximos passos de evolução da ferramenta.

**Palavras-chave:** Nuvem Pública. Agilidade. Código. Terraform®.

## SUMÁRIO

<b>1 INTRODUÇÃO</b>	<b>6</b>
<b>2 TEMA</b>	<b>7</b>
<b>3 OBJETIVOS</b>	<b>8</b>
3.1 OBJETIVO GERAL	8
3.2 OBJETIVOS ESPECÍFICOS	8
<b>4 PROCEDIMENTOS METODOLÓGICOS</b>	<b>9</b>
4.1 CAMPO DE ESTUDO	9
4.2 INSTRUMENTOS DE COLETA DE DADOS	9
<b>5 APRESENTAÇÃO E ANÁLISE DA REALIDADE OBSERVADA</b>	<b>10</b>
5.1 CONTEXTO DA REALIDADE	10
5.2 DESCRIÇÃO E ANÁLISE DA REALIDADE OBSERVADA	10
<b>6 PROPOSTA DE SOLUÇÃO DA SITUAÇÃO-PROBLEMA</b>	<b>12</b>
6.1 PROPOSTA DE MELHORIA PARA A REALIDADE ESTUDADA	12
6.2 RESULTADOS ESPERADOS	12
6.3 VIABILIDADE DA PROPOSTA	13
<b>7 CONSIDERAÇÕES FINAIS</b>	<b>14</b>
<b>REFERÊNCIAS</b>	<b>15</b>
<b>APÊNDICE</b>	<b>16</b>

## 1 INTRODUÇÃO

A TI em 2020 é uma loucura, com a concretização dos conceitos de computação em nuvem estarem evoluindo exponencialmente, fabricantes com cerca de 300 serviços (só em nuvem) disponíveis para consumo, fez o mundo da TI virar de cabeça para baixo, com a facilidade de se experimentar tanto poder computacional, que até pouco tempo atrás estava restrito a um certo grupo de pessoas e regiões. Porém, um ponto permanece com o mesmo objetivo nas empresas, a necessidade de se fazer mais com menos e a necessidade de se ser ágil em uma época em que os dias podem ser o diferencial do sucesso para o fracasso.

A fim de mostrar a agilidade que a infraestrutura como código pode trazer para as organizações, elaborei este estudo para demonstrar que com muito pouco pode-se começar a se beneficiar da infraestrutura como código e que grandes investimentos financeiros não são essenciais, para começar a utilização e até mesmo para se evoluir numa utilização mais robusta.

Para tanto, foi criado um exemplo de deploy para exemplificar quão simples é a utilização da ferramenta, bem como, foram coletadas informações sobre a agilidade que a infraestrutura como código através da ferramenta Terraform® trouxe para uma empresa prestadora de serviços de TI para pequenas e médias empresas, ainda abordamos quais os próximos passos recomendados para que esta empresa possa evoluir ainda mais na agilidade das entregas de seus projetos, sem precisar elevar seu quadro de funcionários e também sem fazer grandes investimentos.

## 2 TEMA

O surgimento da computação em nuvem trouxe muita facilidade na criação da infraestrutura computacional para as empresas em geral, contudo para empresas que prestam serviços na área de infraestrutura em nuvem para outras empresas, pode não ser tão simples manter varias estruturas ativas e funcionais, bem como, traz um enorme trabalho repetitivo ficar criando estas estruturas base de forma manual, que na maioria das vezes são muito parecidas. Para isso, usamos o conceito de infraestrutura como código, onde são gerados códigos para criação de objetos em nuvem, que através de arquivos de configuração, podem gerar infraestruturas novas, ampliar uma infraestrutura existente ou retornar uma infraestrutura a um momento especifico, como para antes de uma atualização ou para atendimento de uma demanda sazonal que necessitava de mais poder computacional.

Infraestrutura como código mostrou-se uma excelente solução para livrar equipes de tarefas enfadonhas do cotidiano realizadas manualmente (CAVALCANTI, 2018).

O presente estudo deve ser compreendido como uma demonstração da utilização da ferramenta Terraform® para criar, uma infraestrutura básica em nuvem e como usar variáveis para reaproveitar este código. Para organizações que prestam serviço para diversas empresas neste campo de infraestrutura em nuvem, esse reaproveitamento de código traz agilidade e desonera a equipe, que não precisa ficar executando tarefas repetitivas na criação dos componentes destes ambientes, tornando prática a criação destes novos ambientes, apenas montando arquivos de configurações com as informações que cada componente do ambiente deve receber como por exemplo, na criação de uma máquina virtual em nuvem, bastaria informar o nome da mesma, a versão do Sistema Operacional e executar o modulo de criação, e não executar passo a passo a criação desta máquina virtual e todos seus componentes. O benefício social deste estudo é trazer informação sobre o assunto e a ferramenta escolhida, pois em um mundo cada vez mais ágil, quem dominar estes conceitos e ferramenta terá melhores oportunidades no mercado de trabalho atual.

### **3 OBJETIVOS**

#### **3.1 OBJETIVO GERAL**

Demonstrar como a utilização do Terraform® para criar e gerenciar uma infraestrutura em cloud traz agilidade para as empresas.

#### **3.2 OBJETIVOS ESPECÍFICOS**

O presente estudo acadêmico visa demonstrar como utilizar o Terraform® para criar e manter atualizada sua infraestrutura em nuvem, para isso, vamos demonstrar como criar a infraestrutura básica com virtual networks, subnets e segmentação de rede em um ambiente de nuvem, aumentar essa infraestrutura básica com mais componentes como máquinas virtuais, trazer os benefícios de se criar módulos para reaproveitar códigos do Terraform®, bem como, evidenciar a melhora que sua aplicação pode trazer em um ambiente corporativo de prestação de serviços.

## 4 PROCEDIMENTOS METODOLÓGICOS

### 4.1 CAMPO DE ESTUDO

O estudo de caso baseia-se em um formato explicativo e o universo do estudo é a ferramenta Terraform® e como utilizá-la de uma forma coerente para otimizar os tempos de execução na criação de ambientes em nuvem. O estudo se baseia na implementação da ferramenta na área de implantação de ambientes em nuvem de uma empresa situada em Porto Alegre - RS, prestadora de serviço a várias pequenas e médias empresas no Brasil.

### 4.2 INSTRUMENTOS DE COLETA DE DADOS

Os instrumentos de coleta de dados adotados neste trabalho são descritos no quadro a seguir.

Quadro 1 – Instrumento de coleta de dados

<b>Instrumento de coleta de dados</b>	<b>Universo pesquisado</b>	<b>Finalidade do Instrumento</b>
<b>Entrevista</b>	Gerentes e técnicos da área de implantação da empresa.	Demonstrar a satisfação dos técnicos no uso da ferramenta e qual o grau de dificuldade na utilização da mesma.
<b>Dados arquivados</b>	Documentação do fabricante	Instruções de instalações da ferramenta.

Fonte: CAVALCANTI e MOREIRA (2008).

## 5 APRESENTAÇÃO E ANÁLISE DA REALIDADE OBSERVADA

Os arquivos de configuração que constam no apêndice I tem como objetivo criar uma infraestrutura básica no Azure®, plataforma de nuvem pública da Microsoft®. O arquivo Exemplo01.tf descreve a criação de um ambiente em nuvem composto por 1 grupo de recursos, 1 rede virtual, 2 subnets, 1 máquina virtual, 1 disco com sistema operacional Ubuntu®, 1 interface de rede, 1 ip público e 1 grupo de segurança que libera acesso da internet via ssh para máquina virtual.

Conforme roteiro de entrevista que consta no apêndice II, baseado na autora Carolina Ignaczk, a pergunta 1 trata de entender o cargo da pessoa entrevistada, a pergunta 2 trata-se de entender os tempos de execução de implantações de projetos antes da utilização do Terraform®, a pergunta 3 trata-se de entender os tempos de execução de implantações de projetos depois da utilização do Terraform® pela empresa, a pergunta 4 tem como finalidade entender a visão do entrevistado sobre alterações no ambiente utilizando a ferramenta Terraform® após a criação do mesmo, a pergunta 5 tem como objetivo entender a percepção da pessoa sobre a complexidade na utilização/adoção da ferramenta, a pergunta 6 tem como objetivo entender o grau de conhecimento da pessoa na ferramenta Terraform®, a pergunta 7 tem como objetivo entender o grau de satisfação dos entrevistados com a ferramenta e a pergunta 8 tenta entender quais pontos precisam ser explorados para melhoria.

Baseado nas respostas das entrevistas que constam no apêndice III, temos informações para mostrar como a ferramenta auxiliou a empresa na melhora das entregas da sua área de implantação, bem como a satisfação dos entrevistados com o uso do Terraform®.

### 5.1 CONTEXTO DA REALIDADE

O estudo foi executado em uma empresa do Rio Grande do Sul, prestadora de serviços de TI para pequenas e médias empresas de todo o Brasil, o estudo se baseia em demonstrar como criar uma infraestrutura básica de rede utilizando a ferramenta Terraform® e levantar a melhora na performance na implantação de projetos antes e depois da implantação da ferramenta Terraform® como ponto centralizador de deploy de ambientes em nuvem pública. Foram entrevistados 1 gerente, 1 coordenador e 2 técnicos da área de implantação de projetos.

### 5.2 DESCRIÇÃO E ANÁLISE DA REALIDADE OBSERVADA

Para executar a criação do ambiente, você precisa fazer o download da ferramenta Terraform® através dos procedimentos descritos na seção Download Terraform®, disponível no Site Terraform (2020), após a ferramenta estar instalada, criar dois arquivos chamados “exemplo01.tf” e “variables.tf” no seu computador e conforme instruções descritas na seção de Introduction do Site Terraform (2020), executar seu primeiro deploy com Terraform®. Tomando como base os exemplos citados neste estudo, podemos compreender um pouco do funcionamento da ferramenta e como ela pode trazer agilidade no deploy de múltiplos ambiente, reaproveitando o código já criado e substituindo apenas o arquivo de variáveis.

Apresento agora com base nas respostas do questionário como a ferramenta trouxe para a empresa melhoria no processo de implantação de seus novos clientes, podemos constatar que o nível gerencial teve muita pouca interação com a ferramenta, a coordenação tem ciência de como a ferramenta funciona, porem quase não utiliza e os técnicos são as pessoas que tem mais contato com a ferramenta Terraform®.

Com base nos tempos de implantação antes e depois da ferramenta, foi possível constatar que os cargos gerenciais se basearam mais nas ferramentas de controle para responder os questionamentos e os técnicos em sua experiencia de utilização. Houve um grande ganho após a implementação da ferramenta, os tempos caíram consideravelmente, chegando em alguns casos até a 80%, mas no mínimo 40% para as novas implementações. Para as alterações em ambientes já existentes o ganho também foi constatado, principalmente por não haver mais a necessidade de se entender todo ambiente que já existia para a alteração.

Sobre a complexidade da ferramenta, para o nível gerencial ficou evidente que não foi tão simples o entendimento dela, o que já era de se esperar devido ao nível gerencial não atuar na parte técnica. Para a coordenação o entendimento foi maior, mesmo assim não tão profundo, pois utilizar a ferramenta no dia-a-dia não é uma função do coordenador. Já para a área técnica, pode se perceber que o entendimento da ferramenta foi mais simples, devido a estas pessoas estarem atuando na área técnica e implantações e pelo fato da ferramenta se basear em uma linguagem já compreendida pelos técnicos.

De uma forma geral, a implantação da ferramenta trouxe um ganho em produtividade e celeridade na implantação dos projetos da área estudada, todos os entrevistados se mostraram satisfeitos com a implantação da nova ferramenta, mas pontos de melhoria puderam ser constatados.

## 6 PROPOSTA DE SOLUÇÃO DA SITUAÇÃO-PROBLEMA

### 6.1 PROPOSTA DE MELHORIA PARA A REALIDADE ESTUDADA

Conforme foi possível constatar, a utilização da ferramenta traz benefícios e agilidade no deploy de infraestrutura em nuvem pública, porém de configuração citado neste estudo é básico, bem como a implantação executada na empresa de TI do Rio grande do Sul em forma de POC.

Agora que a ferramenta mostrou seu poder e os resultados que pode trazer para a área de implantações, uma implantação formal da ferramenta deve ocorrer, nesta implantação deve-se aprofundar a criação de módulos bem como a lógica de iterações dos deploys, trazendo a possibilidade de executar um mesmo módulo várias vezes consecutivas afim de criar um ambiente maior e mais complexo do que o citado no exemplo.

Em nosso exemplo criamos um recurso de cada como forma de ilustrar o que a ferramenta pode fazer, como próximos passos devemos modular cada item criado no exemplo, tornando estes módulos passíveis de repetição em tempo de execução, assim, ao invés de criar apenas um recurso de cada tipo, os módulos poderiam ser chamados em vários pontos do deploy, procedendo com a criação de vários recursos diferentes, porém usando o mesmo template.

Outro ponto de melhoria é a utilização de um orquestrador para os deploys, mesmo a ferramenta Terraform® podendo ser executada de qualquer computador, para uma operação escalar e para manter os níveis de segurança, recomenda-se a utilização de um orquestrador como o Azure DevOps® por exemplo, com esta ferramenta é possível executar múltiplos deploys simultâneos em ambientes distintos sem expor/distribuir as chaves de acesso destes ambiente para várias pessoas/computadores, uma vez que estas senhas ficam salvas em um cofre central e são utilizadas em tempo de deploy apenas.

### 6.2 RESULTADOS ESPERADOS

Esperamos com a implantação de módulos trazer um melhor aproveitamento de código para criação de objetos do mesmo tipo, desta forma não precisamos ficar duplicando código, chamamos o mesmo módulo várias vezes passando arquivos de configurações distintos, ou até um único arquivo de configuração com vários objetos diferentes referenciados.

Referente aos orquestrador, esperamos que a implantação do mesmo traga o benefício da escalabilidade e segurança, uma vez que podemos executar múltiplos deploys simul-

tâneos, manter todas as chaves de acesso armazenadas em um único lugar central sem expor as mesmas, e ainda será possível ter uma visibilidade central de todos os deploys executados, bem como, uma regra de autorização para os mesmos, onde um técnico pode submeter um deploy de criação, mas o mesmo não será executado até que o coordenador ou outro técnico com mais experiência valide e aprove a solicitação.

### 6.3 VIABILIDADE DA PROPOSTA

Para a implementação dos módulos e do orquestrador estimamos que será necessário investir cerca de 360 horas, aproximadamente 2 meses de trabalho ao qual, 50 horas estão previstas para levantamentos dos objetos mais comuns criados dentro da nuvem publica para clientes da empresa, 200 horas destinadas a criação dos módulos, 80 horas previstas para a configuração do orquestrador e 30 horas para testes e gestão do projeto.

Para a execução técnica, a empresa já possui o recurso com o conhecimento necessário na ferramenta Terraform®, portanto o investimento será em forma de tempo deste profissional, que já foi contratado para executar esta evolução na plataforma, o investimento financeiro previsto com a plataforma de orquestração é zero para até 5 deploys simultâneos durante 22 dias no mês, essa previsão se dá devido a ferramenta Azure DevOps® ser gratuita para até 5 usuários básicos e possuir 1800 horas de execução de agentes de deploy gratuitos por mês.

Item Porposto	Semana 1	Semana 2	Semana 3	Semana 4	Semana 5	Semana 6	Semana 7	Semana 8	Semana 9
Levantamento de objetos	X	X							
Criação dos módulos		X	X	X	X	X			
Ferramenta de orquestração							X	X	
Gestão do Projeto									X
Investimento financeiro	N/A								

## 7 CONSIDERAÇÕES FINAIS

Este estudo de caso demonstrou como utilizar o Terraform® para criação de itens básicos em uma estrutura de nuvem pública, bem como abordou a importância de se evoluir na utilização da ferramenta com o incremento de módulos e de um orquestrador, que trará melhor agilidade e segurança para uma empresa que presta serviços de TI em nuvem pública para outras empresas.

Ainda foi possível demonstrar como a agilidade da ferramenta, trouxe grandes ganhos em tempos no deploy de infraestrutura em nuvem pública, ficando evidenciado reduções de até 80% do tempo que levava para determinadas tarefas serem executadas, bem como, a padronização que a ferramenta traz a criação de objetos em nuvem pública.

Identificou-se ainda que para a empresa em questão, o investimento financeiro para melhorar o uso da ferramenta dentro da empresa era muito baixo, e já havia sido feito pela empresa com a contratação do profissional que montou o estudo, bastante apenas dedicação de tempo para concretizar esta evolução.

## REFERÊNCIAS

CAVALCANTE, Marcelo. Infraestrutura como Código com Terraform. **Blog Marcelo Cavalcante**, 2018. Disponível em: <https://blog.marcelocavalcante.net/infraestrutura-como-c%C3%B3digo-com-terraform/>. Acesso em: 20 mar. 2020.

Site Terraform. Download Terraform. **HASICORP**, 2020 Disponível em: <https://www.terraform.io/downloads.html>. Acesso em: 25 mai. 2020.

Site Terraform. Introduction. **HASICORP**, 2020. Disponível em: [https://learn.hashicorp.com/terraform/azure/intro\\_az](https://learn.hashicorp.com/terraform/azure/intro_az). Acesso em: 25 mai. 2020.

IGNACZUK, Carolina. Guia de como fazer pesquisa de satisfação do cliente: tudo o que você precisa saber. **Movidesk**, 2019. Disponível em: <https://conteudo.movidesk.com/como-fazer-pesquisa-satisfacao-cliente/>. Acesso em: 10 mai. 2020

CAVALCANTI, Marcelo e MOREIRA, Enzo. **Metodologia de estudo de caso**: livro didático. 3. ed. rev. e atual. Palhoça: Unisul Virtual, 2008. 170 p.

LOHN, Joel Irineu. **Metodologia para elaboração e aplicação de projetos**: livro didático. 2 ed. rev. e atual. Palhoça: Unisul Virtual, 2005. 100 p.

RAUEN, Fábio José. **Roteiros de investigação científica**. Tubarão: Unisul, 2002.

## APÊNDICE I

### ARQUIVO “EXEMPLO01.TF”

```

# Configura credenciais de acesso ao Azure
provider "azurerm" {
  version = "~>1.32.0"
  subscription_id = "coloque aqui o id da subscrição"
  client_id      = "coloque aqui o client id do service principal"
  client_secret  = "coloque aqui o client secret do service principal "
  tenant_id     = "coloque aqui o tenant id"
}

# Cria o resource group "rsg001"
resource "azurerm_resource_group" "rsg001" {
  name     = var.rsg001-name
  location = var.location
  tags    = var.tags
}

# Cria a rede virtual "vnet001"
resource "azurerm_virtual_network" "vnet001" {
  name            = var.vnet001-name
  address_space   = ["10.0.0.0/16"]
  location        = var.location
  resource_group_name = azurerm_resource_group.rsg001.name
  tags           = var.tags
}

# Cria a subnet 01 "vnet001-snt01"
resource "azurerm_subnet" "vnet001-snt01" {
  name                = var.vnet001-snt01-name
  resource_group_name = azurerm_resource_group.rsg001.name
  virtual_network_name = azurerm_virtual_network.vnet001.name
  address_prefix      = "10.0.1.0/24"
}

# Cria a subnet 02 "vnet001-snt02"
resource "azurerm_subnet" "vnet001-snt02" {
  name                = var.vnet001-snt02-name
  resource_group_name = azurerm_resource_group.rsg001.name
  virtual_network_name = azurerm_virtual_network.vnet001.name
  address_prefix      = "10.0.2.0/24"
}

# Cria o ip público "vm001-pip01" para a "vm001"
resource "azurerm_public_ip" "vm001-pip01" {
  name      = var.vm001-pip01-name
  location  = var.location

```

```
resource_group_name = azurerm_resource_group.rsg001.name
allocation_method = "Static"
```

```
tags = var.tags
```

```
}
```

```
# Cria o grupo de segurança de rede "vm001-nsg01" para a "vm001" liberando conexão ssh da internet.
```

```
resource "azurerm_network_security_group" "vm001-nsg01" {
  name          = var.vm001-nsg-name
  location      = var.location
  resource_group_name = azurerm_resource_group.rsg001.name
  tags = var.tags
```

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

```
# Cria a interface de rede "vm001-nic01" para a "vm001"
```

```
# Associa o ip público "vm001-pip01" e o grupo de segurança de rede "vm001-nsg01"
```

```
resource "azurerm_network_interface" "vm001-nic01" {
  name          = var.vm001-nic01-name
  location      = var.location
  resource_group_name = azurerm_resource_group.rsg001.name
  network_security_group_id = azurerm_network_security_group.vm001-nsg01.id
```

```
  ip_configuration {
    name          = "ipconfig"
    subnet_id     = azurerm_subnet.vnet001-snt01.id
    private_ip_address_allocation = "dynamic"
    public_ip_address_id = azurerm_public_ip.vm001-pip01.id
  }
}
```

```
tags = var.tags
```

```
}
```

```
# Cria a vm linux "vm001" com sistema operacional ubuntu, vinculando todos os recursos criados anteriormente.
```

```
resource "azurerm_virtual_machine" "vm001" {
  name          = var.vm001_name
  location      = var.location
```

```
resource_group_name = azure_rm_resource_group.rsg001.name
network_interface_ids = [azure_rm_network_interface.vm001-nic01.id]
vm_size              = "Standard_DS1_v2"

storage_os_disk {
  name           = var.vm001-osd-name
  caching        = "ReadWrite"
  create_option  = "FromImage"
  managed_disk_type = "Standard_LRS"
}

storage_image_reference {
  publisher = "Canonical"
  offer     = "UbuntuServer"
  sku       = lookup(var.sku, var.location)
  version   = "latest"
}

os_profile {
  computer_name = var.vm001_name
  admin_username = var.admin_username
  admin_password = var.admin_password
}

os_profile_linux_config {
  disable_password_authentication = false
}

tags = var.tags
}
```

## ARQUIVO "VARIABLES.TF"

# variável que define a localização onde os recursos serão criados.

```
variable "location" {  
  type = string  
  description = "default location"  
  default = "EastUS2"  
}
```

# variável que define o nome do usuário admin das vm's

```
variable "admin_username" {  
  type = string  
  description = "Administrator user name for virtual machine"  
  default = "admin.empresa"  
}
```

# Variável que define a senha do usuário admin das vm's

```
variable "admin_password" {  
  type = string  
  description = "Password must meet Azure complexity requirements"  
  default = "08179501-5d8e-4473-816b-b5791971740d"  
}
```

# variável que define o nome da vm001

```
variable "vm001-name" {  
  type = string  
  description = "name for vm"  
  default = "vm001"  
}
```

# variável que define o nome do grupo de recursos da vm001

```
variable "rsg001-name" {  
  type = string  
  description = "nome do grupo de recursos da vm 001"  
  default = "rsg001"  
}
```

# variável que define o nome da rede virtual da vm001

```
variable "vnet001-name" {  
  type = string  
  description = "nome da rede virtual da vm 001"  
  default = "vnet001"  
}
```

# variável que define o nome da subnet01 da rede virtual vnet001

```
variable "vnet001-snt01-name" {  
  type = string  
  description = "nome da subnet 001"  
  default = "vnet001-snt01"  
}
```

```
# variável que define o nome da subnet02 da rede virtual vnet001
variable "vnet001-snt02-name" {
    type = string
    description = "nome da subnet 002"
    default = "vnet001-snt02"
}

# variável que define o nome do ip público da vm001
variable "vm001-pip01-name" {
    type = string
    description = "nome do ip público da vm 001"
    default = "vm001-pip01"
}

# variável que define o nome do grupo de segurança de rede da vm001
variable "vm001-nsg-name" {
    type = string
    description = "nome do grupo de segurança de rede da vm001"
    default = "vm001-nsg"
}

# variável que define o nome da interface de rede da vm001
variable "vm001-nic01-name" {
    type = string
    description = "nome da interface de rede da vm001"
    default = "vm001-nic01"
}

# variável que define o nome da disco do sistema operacional da vm001
variable "vm001-osd-name" {
    type = string
    description = "nome do disco do sistema operacional da vm001"
    default = "vm001-osd"
}

# Variável que define as tags padrões dos objetos criados.
variable "tags" {
    type = map
    description = "nome e valores das tags a serem inseridos nos objetos criados."
    default = {
        Environment = "d1"
        Department = "it"
        Description = "ambiente de teste"
        BusinessUnit = "infra"
    }
}
```

## APÊNDICE II

Roteiro das questões executadas.

1 - Qual seu cargo dentro da empresa?

2 - Qual o tempo médio de implementação de um projeto em nuvem antes da ferramenta terraform®?

3 - Qual o tempo médio de implementação de um projeto em nuvem depois da ferramenta terraform®?

4 - Qual seu ponto de vista sobre alterações no ambiente em nuvem utilizando a ferramenta terraform®?

5 - Qual sua percepção da complexidade da ferramenta?

6 - Qual seu grau de utilização da ferramenta?

A - Só ouvi falar

B - Fiz um teste

C - Uso muito pouco no dia-a-dia

D - Uso todos os dias

7 - Qual seu grau de satisfação com a utilização da ferramenta?

A - Alto

B - Médio

c - Baixo

8 - Em quais pontos você acha que podemos melhorar na utilização da ferramenta?

### APÊNDICE III

Respostas:

Gerente:

- 1 – Gerente de operações
- 2 – De 8 a 16 horas conforme dados da ferramenta de apontamento de horas
- 3 – De 4 a 10 horas conforme dados da ferramenta de apontamento de horas
- 4 – O tempo das alterações ficou mais rápido conforme dados da ferramenta de apontamento de horas.
- 5 – Não utilizei a ferramenta, mas o retorno da equipe técnica foi muito bom.
- 6 – A
- 7 – A
- 8 – Analisar as necessidades dos técnicos para poder melhorar a utilização da ferramenta dentro da empresa.

Coordenador:

- 1 – Coordenador de operações.
- 2 – Algo em torno de 8 e 16 horas.
- 3 – Algo em torno de 4 e 10 horas.
- 4 – Ficou mais ágil.
- 5 – Parece complexa a utilização para quem não está mais no meio técnico.
- 6 – B
- 7 – A (Satisfeito pela utilização da equipe.)
- 8 – Aumentar o uso dela para todos nossos projetos de nuvem pública.

Técnico 1

- 1 – Consultor Técnico Jr.
- 2 – Entre 8 e 12 Horas
- 3 – Entre 2 a 8 horas
- 4 – Ficou muito mais fácil, como já temos todo o ambiente descrito em YAML, fica mais simples a expansão deste ambiente com novos recursos.
- 5 – Exige um grande estudo para se iniciar a utilizar, mas depois de entender o funcionamento dela, toda as atividades de criação de recursos na nuvem passam a ser rápidas e simples.
- 6 – D
- 7 – A
- 8 – Avanço na criação de módulos para melhorar o deploy de ambientes mais complexos.

Técnico 2

- 1 – Consultor Técnico Senior
- 2 – Para ambiente mais simples, de 2 a 6 horas e ambientes mais complexos de 8 a 20 horas.
- 3 – Para ambiente mais simples de 1 a 3 horas e ambientes mais complexos de 4 a 10 horas.
- 4 – Muito mais prático e fácil de se executar.
- 5 – Eu já utilizo a ferramenta a algum tempo, a implantação dela dentro da empresa foi puxada por mim, então posso dizer que a ferramenta não é complexa, mas exige estudo e dedicação para sua boa utilização.
- 6 – D
- 7 – A
- 8 – Difundir para toda a equipe a utilização da ferramenta e iniciar a implementação de módulos e a automação do processo de execução da ferramenta através do Azure DevOps.