

# Introdução ao Terraform e Terragrunt

Apresentação técnica sobre uso de Terraform e Terragrunt para automação e multi-cloud em OCI



**Anderson Luiz Pancheniak**  
RM 358342



# Por que usar Terraform?

## 1 Adote Infraestrutura como Código (IaC) para gestão segura

Crie, altere e versiona infraestrutura de maneira segura e previsível, garantindo controle e rastreabilidade das mudanças.

## 2 Implemente multi-cloud com suporte a diversos provedores

Use Terraform para gerenciar infraestrutura em múltiplos provedores como OCI, AWS, Azure e GCP, garantindo flexibilidade e portabilidade.

## 3 Utilize abordagem declarativa para definir infraestrutura

Descreva o estado desejado da infraestrutura sem detalhar o passo a passo da execução, facilitando manutenção e previsibilidade.

## 4 Controle versões para gestão confiável de mudanças

Gerencie alterações na infraestrutura com versionamento, assegurando histórico e possibilidade de rollback seguro.

## 5 Automatize integrações com pipelines CI/CD

Facilite a entrega contínua integrando Terraform em pipelines CI/CD, promovendo automação e agilidade nas operações.

# Por que usar Terragrunt?

Principais vantagens e facilidades na gestão de infraestrutura com Terraform



# Principais Componentes OCI Utilizados - Visão Geral



# Detalhamento dos Componentes OCI - Computação e Rede

Recurso	Descrição
oci_identity_compartment	Compartimento para organizar recursos OCI por projeto ou ambiente
oci_core_virtual_network	Rede virtual isolada na nuvem para agrupar subnets e recursos
oci_core_subnet	Sub-rede dentro da VCN, pode ser pública ou privada
oci_core_internet_gateway	Permite tráfego da VCN para a internet
oci_core_nat_gateway	Permite que recursos privados acessem a internet sem IP público
oci_core_service_gateway	Permite acesso a serviços OCI (ex.: Object Storage) sem sair da rede
oci_core_route_table	Define regras de roteamento para tráfego dentro da VCN
oci_core_security_list	Firewall que controla entrada e saída de tráfego nas subnets
oci_core_instance	Instância de compute (VM) que roda Oracle Linux ou outro SO

# Benefícios do uso integrado: Terraform + Terragrunt

Vantagens técnicas e operacionais da combinação Terraform e Terragrunt

- **Garantir consistência entre ambientes**

Ambientes são criados de forma idêntica, reduzindo divergências e riscos de configuração inconsistente.

- **Aumentar segurança via versionamento**

Versionamento detalhado das mudanças minimiza erros manuais e garante rastreabilidade das alterações.

- **Automatizar processos com CI/CD**

Permite integração contínua com pipelines para criação e destruição rápida de infraestrutura, aumentando agilidade.

- **Promover visibilidade e controle com 'plan'**

O comando 'plan' exibe alterações antes da aplicação, permitindo análise prévia e maior controle operacional.

- **Facilitar escalabilidade e gestão multi-região**

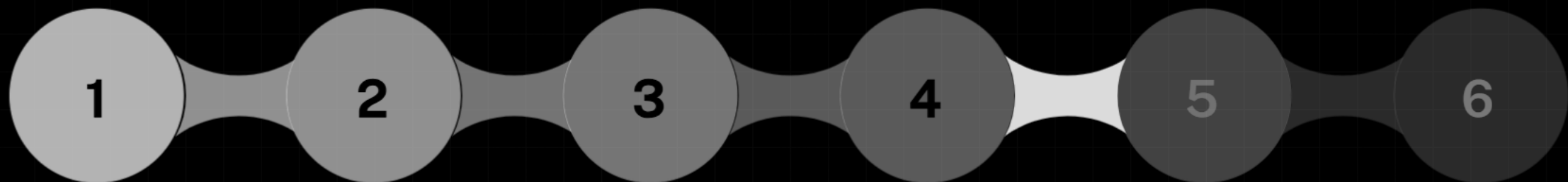
Replicação simples de ambientes e gerenciamento eficiente de múltiplas contas ou regiões para expansão.

- **Melhorar eficiência operacional e organização**

Reduz repetição de código e organiza configurações, otimizando o gerenciamento da infraestrutura como código.

# Componentes e Fluxo de Trabalho com Terragrunt

Visão geral da organização, execução e benefícios do Terragrunt em ambientes complexos



## **Estrutura hierárquica com terragrunt.hcl**

Organiza arquivos para incluir configurações comuns como variáveis e backends, facilitando a manutenção.

## **Gerenciamento de dependências entre módulos**

Permite controle e ordem correta na execução dos módulos interdependentes.

## **Execução apply-all**

Aplica todas as mudanças em sequência correta para provisionamento consistente.

## **Execução destroy-all**

Destrói recursos seguindo a ordem dependente para evitar falhas.

## **Execução plan-all**

Avalia mudanças em todos os módulos antes da aplicação para validação prévia.

## **Resultados e benefícios**

Oferece maior controle e automação focados em ambientes multi-cloud e multi-ambientes.

# Versões Utilizadas no Projeto

Versões específicas para garantir compatibilidade e estabilidade e otimização de custos em OCI

Ferramenta	Versão
Terraform	v1.5.7
Terragrunt	v0.55.11
OCI Provider	7.16.0



# Resumo dos Benefícios e Práticas Recomendadas

Visão geral dos principais ganhos e recomendações para uso eficiente do Terraform e Terragrunt

## Assegurar automação e consistência na infraestrutura

Terraform e Terragrunt garantem uma infraestrutura previsível e replicável, facilitando a automação de ambientes complexos.

## Simplificar o gerenciamento com módulos reutilizáveis

A utilização de módulos e hierarquias auxilia na manutenção eficiente e organização do código de infraestrutura.

## Garantir segurança e visibilidade das mudanças

Versionamento e execução de planos prévios minimizam riscos operacionais, aumentando a confiabilidade das alterações.

## Integrar processos CI/CD para acelerar entregas

A integração contínua e entrega contínua elimina processos manuais e acelera o ciclo de deploy de infraestrutura.

## Suportar escalabilidade multi-cloud e multi-ambiente

O uso do Terraform e Terragrunt permite gerenciar múltiplas regiões, contas e ambientes de forma escalável e flexível.

## Seguir recomendações para melhores práticas

Manter versões controladas e atualizadas, documentar hierarquias e dependências, além de monitorar planos antes de aplicar mudanças.



Vamos  
implementar  
automação  
avançada e  
fortalecer a  
governança para  
acelerar a entrega  
de infraestrutura