[WIP] Repositório Infra

Este repositório gerencia a infraestrutura de VMs e clusters Kubernetes do projeto DICOM no vSphere/vCenter. As VMs são provisionadas com **Terraform/Terragrunt** e configuradas com **Ansible**. Nenhuma VM deve ser criada ou modificada manualmente fora deste repositório.

Sumário:

- Repositório e Pipelines
- Exemplos de Operações
- Variáveis de CI
- Cuidados Importantes

- Estrutura e Automações

Estrutura e Automações

Estrutura do repositório

A estrutura do repositório segue o seguinte formato:

```
infra/
├── clusters/           # Um diretório por cluster Kubernetes
│   └── admin/          # Cluster admin (Terraform + Terragrunt)
├── modules/            # Módulos Terraform reutilizáveis
│   ├── vsphere-vm/     # Provisionamento de VM no vSphere
│   ├── k3s-cluster/    # Cluster k3s completo
│   ├── k3s-server/     # Nó servidor k3s
│   └── k3s-agent/      # Nó agente k3s
├── ansible/            # Playbooks de pós-configuração
│   ├── site.yml        # Configura clusters k8s (addons, discos, etc)
│   └── edge.yml        # Configura VMs de borda (nginx, firewall)
└── .gitlab-ci.yml      # Pipeline de CI
```

Cada diretório em `clusters/` é um módulo Terragrunt que referencia os módulos reutilizáveis de `modules/`. O estado do terraform está sendo salvo no próprio GitLab. Desse modo, todos os comandos do terraform devem ser executados diretamente pelo GitLab CI ou utilizando o estado remoto do GitLab.

Nomenclatura de branches neste repositório

Prefixo	Exemplo
`feat/`	`feat/cluster-portal`
`upgrade/`	`upgrade/k3s-1.32`, `upgrade/terraform-1.10`
`fix/`	`fix/rede-admin`, `fix/disco-node1`

Prefixo	Exemplo
`resize/`	`resize/admin-node1-memoria`, `resize/portal-node2-cpu`
`docs/`	`docs/tutorial-novo-cluster`

Pipeline de CI

O pipeline possui quatro stages executados em sequência:

```
validate ──► plan ──► apply ──► configure
```

Stage	Quando roda	O que faz
`validate`	Todo push e todo MR	Valida a sintaxe dos arquivos Terraform
`plan`	Todo push e todo MR	Mostra o que seria criado, alterado ou destruído
`apply`	Somente na `main`, **manual**	Aplica as mudanças de infraestrutura no vSphere
`configure`	Somente na `main`, automático após apply	Roda Ansible para configurar k8s e VMs de borda

O `apply` **nunca roda sozinho** — exige um clique manual de quem aprovou o MR. O `configure` roda automaticamente em seguida, se o `apply` for bem-sucedido.

Variáveis de CI necessárias

Configuradas no grupo `dicom` do GitLab (Settings → CI/CD → Variables):

Variável	Descrição
`VSPHERE_PASSWORD`	Senha do usuário de serviço no vSphere
`ANSIBLE_SSH_PUBLIC_KEY`	Chave pública SSH injetada nas VMs pelo Terraform
`ANSIBLE_SSH_PRIVATE_KEY`	Chave privada SSH usada pelo Ansible (em base64)
`CERT_MANAGER_TSIG_SECRET`	Chave TSIG para desafios DNS01 do cert-manager (Obsoleto: está sendo movido para repositório `stacks`)
