**Foundation GCP**

**Projeto**

Padrão do plano de fundo

Descrição gerada automaticamente**oposta**

**Documentação**

**AMERICAS MED**

29 de setembro de 2025

Sumário

[**1. Identidade na Nuvem** 3](#_Toc205890993)

[**1.1 Domínios** 3](#_Toc205890995)

[**1.2 Grupos e Usuários** 3](#_Toc205890997)

[**1.3 Autenticação (SSO)** 3](#_Toc205891000)

[**2. Hierarquia de Recursos** 4](#_Toc205891001)

[**2.1 Serviços Compartilhados** 5](#_Toc205891003)

[**2.2 Controle de Custos** 6](#_Toc205891004)

[**3. Políticas Organizacionais** 6](#_Toc205891005)

[**3.1 Políticas Definidas** 7](#_Toc205891006)

[**4. Gerenciamento de Identidade e Acesso** 10](#_Toc205891007)

[**4.1 Políticas Definidas** 10](#_Toc205891008)

[**4.2 Funções e Permissões** 11](#_Toc205891009)

[**4.3 Grupos** 11](#_Toc205891010)

[**5. Conexões Externas** 12](#_Toc205891011)

[**6. Conexões Internas - VPC - Virtual Private Cloud** 13](#_Toc205891012)

[**6.1 Conectividade Privada** 15](#_Toc205891013)

[**7. IAC - Infra As Code - SSM & CloudBuild** 20](#_Toc205891014)

# **1. Identidade na Nuvem**

# O Google Cloud Identity é o produto utilizado para gerenciar usuários, grupos e configurações de segurança em todo o domínio para o G Suite e o Google Cloud Platform.

# **1.1 Domínios**

# O domínio "americasmed.com.br" foi configurado como domínio primário e o domínio "adhosp.com.br" configurado no serviço de dns da conta, para que seja resolvido no destino da dasa diagnósticos.

# **1.2 Grupos e Usuários**

# O provisionamento de Grupos e Usuários foi configurado para ser realizado através do Azure AD, sendo necessário primeiro sua criação no ambiente Microsoft, para então serem sincronizados para o Google Cloud Identity.

# O diagrama a seguir retrata a compreensão do fluxo de trabalho atual para gerenciamento de identidade do Azure AD para o Cloud Identity.

Uma imagem contendo Diagrama

O conteúdo gerado por IA pode estar incorreto.

# **1.3 Autenticação (SSO)**

De acordo com os requisitos da Americasmed, os usuários em sua organização precisarão de acesso ao Google Cloud Console, mas eles desejam continuar gerenciando suas identidades de usuário a partir do Azure AD, que atuaria como uma única fonte de verdade para autenticar seus usuários, portanto, atuaria como um Provedor de Identidade neste caso.

O SSO do Azure AD continuará a ser utilizado para realizar a autenticação do usuário quando eles tentarem fazer login no Google Cloud Console usando seu endereço de e-mail corporativo, eliminando a necessidade de criar uma senha separada para o usuário no Google Cloud ou sincronizar a senha daquele usuário do Azure AD.

Portanto, sempre que qualquer usuário da Americasmed tentar fazer login no Google Cloud Console, assim que ele digitar seu endereço de e-mail corporativo na página de login do GCP, será redirecionado para a página de login do Azure, onde será autenticado usando suas credenciais do Azure AD.

Observe que, quando os usuários tentam fazer login através do SSO, qualquer MFA configurada no lado do Azure AD será acionada e os usuários precisarão concluir a mesma para prosseguir, eliminando a necessidade de configurar qualquer outra forma de MFA/2FA no lado do GCP.

# **2. Hierarquia de Recursos**

# Todos os recursos da nuvem pertencem a um projeto do Google Cloud Platform. Os projetos formam a base para ativar e usar os serviços do Google Cloud Platform, incluindo gerenciamento de APIs, ativação de faturamento, adição e remoção de colaboradores e ativação de outros serviços do Google. Cada projeto é um compartimento separado e cada recurso pertence a exatamente um projeto. Os projetos podem ter vários proprietários e usuários. Cada projeto é gerenciado e faturado separadamente, embora vários projetos possam ter a mesma conta de faturamento.

Abaixo uma visão geral da estrutura criada, esta é hierarquia de recursos da fundação proposta para a Americasmed.

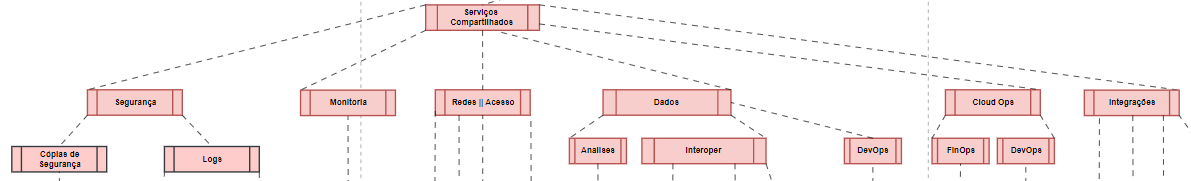
Diagrama

O conteúdo gerado por IA pode estar incorreto.

Ele contém recursos tanto atuais quanto novos e foi cuidadosamente discutido e planejado para permitir uma operação fácil para as equipes, mantendo a segurança como prioridade, seguindo os princípios de privilégio mínimo e segregação de grupos.

# **2.1 Serviços Compartilhados**

Esta área reúne serviços e recursos compartilhados e consumidos por equipes e colaboradores da Americasmed como um todo. Isso não significa que todos terão acesso a esses recursos, mas eles atendem a empresa horizontalmente com base em permissões.



Abaixo temos a seguinte estrutura detalhada:

* **Segurança:** Espaço projetado para administradores de segurança implantarem ferramentas corporativas para proteger a empresa, analisar ameaças e incidentes de segurança, bem como manter padrões de segurança, por exemplo, imagens douradas aprovadas para serem usadas pela empresa.
* **Cópias de Segurança:** Pasta destinada a backup de recursos de projeto.
* **Logs:** Contém buckets de log que servem como armazenamento para Log Sinks configurados em toda a empresa, divididos por ambiente e tipo de log. Também pode ser usado para exportar e analisar em detalhes, exportando para o BigQuery, por exemplo.
* **Monitoria:** Contém escopos de Monitoramento em Nuvem para ambientes de produção e não produção, agregando painéis de métricas, verificações de saúde e alertas.
* **Redes e acesso:** Abriga todos os recursos de rede, como VPCs, Cloud Routers, Cloud VPNs, etc. Esses recursos são compartilhados com projetos consumidores conforme necessário. Mais informações sobre isso na seção de redes.
* **Dados:** Unidade de negócios especial comum e compartilhada em toda a Americasmed para analisar, construir e testar insights a partir de dados antes de disponibilizá-los. A equipe de dados pode estender essas pastas juntamente com grupos IAM específicos para gestão.
* **CloudOps:** Uma equipe atualmente responsável por gerenciar recursos em Nuvem via IAC com uma pasta especial para desenvolver e manter recursos compartilhados em toda a Americasmed. Existem Kubernetes e Ferramentas, mas esses são apenas os começos para muitos mais que virão. Para cada pasta mencionada, há grupos de administração cruzada para gerenciar seus recursos. Confira a seção de gerenciamento de acesso para detalhes. Embora muito menos frequentes que as Verticais Digitais, novas pastas podem ser adicionadas sob Serviços Compartilhados caso a necessidade surja.

# **2.2 Controle de Custos**

Esta é uma área especial para administradores de faturamento controlarem as contas e também para análise, gráficos e relatórios sobre os gastos com o Google Cloud.

Gráfico, Gráfico de funil

O conteúdo gerado por IA pode estar incorreto.

# **3. Políticas Organizacionais**

As Políticas Organizacionais oferecem controle centralizado e programático sobre os recursos de nuvem da sua organização. Como administrador da política organizacional, você poderá configurar restrições em toda a sua hierarquia de recursos.

Uma política organizacional é uma configuração de restrições. Você, como administrador da política organizacional, define uma política organizacional e a aplica a organizações, pastas e projetos para impor as restrições a esse recurso e seus descendentes.

Para definir uma política organizacional, você escolhe uma restrição, que é um tipo particular de restrição contra um serviço do Google Cloud ou um grupo de serviços do Google Cloud. Você configura essa restrição com as restrições desejadas.

Os descendentes do nó da hierarquia de recursos alvo herdarão a política organizacional. Ao aplicar uma política organizacional no nó raiz da organização, você consegue efetivamente promover a aplicação dessa política organizacional e a configuração de restrições em toda a sua organização.

Uma restrição é um tipo de limitação contra um serviço do Google Cloud ou uma lista de serviços do Google Cloud. Pense na restrição como um projeto que define quais comportamentos são controlados. Este projeto é então aplicado a um nó da hierarquia de recursos como uma política organizacional, que implementa as regras definidas na restrição. O serviço do Google Cloud mapeado para essa restrição e associado a esse nó da hierarquia de recursos então aplicará as restrições configuradas dentro da política organizacional.

# **3.1 Políticas Definidas**

Durante as sessões de design, uma seleção de políticas para impor o padrão de segurança e governança foi escolhida e é aplicada no nível da pasta Americasmed.

**Observação:** Para esta fundação, as políticas organizacionais foram aplicadas apenas na nova pasta “Americasmed” e não impactaram nenhum recurso existente atualmente (representados como legado). Abaixo está a lista de políticas selecionadas:

Interface gráfica do usuário, Aplicativo

O conteúdo gerado por IA pode estar incorreto.

Tabela

O conteúdo gerado por IA pode estar incorreto.

Tabela

O conteúdo gerado por IA pode estar incorreto.

Interface gráfica do usuário, Texto

O conteúdo gerado por IA pode estar incorreto.

Interface gráfica do usuário, Texto, Aplicativo

O conteúdo gerado por IA pode estar incorreto.

Tabela

O conteúdo gerado por IA pode estar incorreto.

Interface gráfica do usuário, Texto

O conteúdo gerado por IA pode estar incorreto.

Observe que a tabela acima contém as políticas que estão sendo habilitadas no processo de fundação, no entanto, algumas políticas organizacionais foram discutidas e decididas para não serem habilitadas inicialmente, mas podem ser necessárias no futuro. Certifique-se de verificar a lista de todas as restrições disponíveis na documentação oficial. Esta lista é atualizada periodicamente e pode conter novas restrições que não estavam disponíveis durante a redação deste documento.

# **4. Gerenciamento de Identidade e Acesso**

(IAM) permite que você conceda acesso granular a recursos específicos do Google Cloud e ajuda a prevenir o acesso a outros recursos. O IAM permite que você adote o princípio de segurança do menor privilégio, que afirma que ninguém deve ter mais permissões do que realmente precisa. Você pode definir políticas de Gerenciamento de Identidade e Acesso (IAM) em diferentes níveis da hierarquia de recursos. Os recursos herdam políticas do recurso pai. A política efetiva para um recurso é a união da política definida naquele recurso e a política herdada de seu pai. Para mais detalhes sobre o IAM, consulte este guia.

# **4.1 Políticas Definidas**

Vínculos Lembre-se de que cada vínculo IAM é composto por quem (o principal), o que (ação a ser realizada) e onde (em qual recurso).

Veja o seguinte diagrama como exemplo:

Diagrama

O conteúdo gerado por IA pode estar incorreto.

O grupo de segurança (quem) terá permissões diferentes (o que) com base em qual pasta estamos olhando (onde). Ele nos permite conceder permissões administrativas completas na pasta de segurança, mas uma permissão limitada para gerenciar apenas regras de firewall na pasta de rede.

Você pode até conceder algumas permissões mais amplas no nível de Serviços Compartilhados para que as equipes de segurança visualizem recursos apenas em modo de visualização em todos os serviços, por exemplo.

Além do nível de pasta, você também pode conceder permissões no nível da organização (que se estenderão a todas as pastas e projetos) e no nível do projeto. Em alguns casos, você pode ir ainda mais longe e conceder permissão em um recurso específico usando condições IAM.

# **4.2 Funções e Permissões**

Você pode atribuir funções a sujeitos como usuários, grupos ou contas de serviço. Dentro dessas funções, geralmente há várias permissões que podem conceder funcionalidades muito específicas sobre algum produto ou serviço.

É uma boa prática usar funções pré-definidas criadas e mantidas pelo Google para conceder permissões de menor privilégio aos seus recursos. Embora as pré-definidas geralmente sejam suficientes, se você precisar de um controle mais específico sobre quais permissões conceder, pode criar sua própria função personalizada. Para qualquer um dos casos acima, pode ser confuso entender quais permissões ou até funções você precisa para uma determinada funcionalidade em um determinado produto. Para isso, mantenha os seguintes links de documentação sempre por perto para que você possa rapidamente consultar funções e permissões por produto:

* Entendendo funções Dicionário com todas as funções básicas e pré-definidas com rica descrição listando todas as permissões contidas nelas.
* Referência de permissões não tão úteis quanto o link acima, lista todas as permissões e também descreve quais funções pré-definidas as contêm. Pode ser útil para criar funções personalizadas.

# **4.3 Grupos**

Os grupos do Google podem ajudar você a gerenciar usuários em grande escala. Cada membro de um grupo do Google herda os papéis de Identity and Access Management (IAM) concedidos a esse grupo. Essa herança significa que você pode usar a adesão a um grupo para gerenciar os papéis dos usuários em vez de conceder papéis IAM a usuários individuais.

Os grupos da Americasmed foram criados por meio da ferramenta atual do Azure AD, seguindo os padrões e convenções de nomenclatura da empresa. Após a criação e sincronização com a Cloud Identity, os papéis foram concedidos a cada grupo respectivo de acordo com seus menores privilégios projetados para esta fundação.

Após a compreensão dos requisitos, o design recomendado para a Dasa é baseado nas seguintes categorias:

* **Administrativo**

Grupos altamente privilegiados para administrar a organização.

* **Cross**

Grupo altamente privilegiado dentro de sua própria área para gerenciar serviços compartilhados e controlar verticais de cima.

* **Verticais Digitais** (cada jornada tem sua própria cópia de cada grupo nesta seção)

Os grupos operacionais para colaboradores trabalharem em cada jornada dentro de sua área específica.

* **Contas de Serviço**

Ligadas a recursos para conceder papéis de menor privilégio e executar ações automatizadas.

# **5. Conexões Externas**

O Google oferece várias opções de conectividade para conectividade física em várias geografias. Redes privadas virtuais (VPNs) podem ser construídas em cima dessa camada física e o roteador em nuvem está disponível para gerenciar rotas dinâmicas usando BGP assim que essa conexão estiver configurada. A internet pública também é uma opção. O Interconnect em nuvem fornece conexões de baixa latência e alta disponibilidade que permitem transferir dados de forma confiável entre suas redes locais e as redes da Google Cloud Virtual Private Cloud (VPC). Além disso, as conexões de Interconnect fornecem comunicação de endereços IP internos, o que significa que os endereços IP internos são diretamente acessíveis de ambas as redes. O Interconnect em nuvem oferece duas opções para expandir sua rede local:

O Interconnect em nuvem fornece conexões de baixa latência e alta disponibilidade que permitem transferir dados de forma confiável entre suas redes locais e as redes da Google Cloud Virtual Private Cloud (VPC). Além disso, as conexões de Interconnect fornecem comunicação de endereços IP internos, o que significa que os endereços IP internos são diretamente acessíveis de ambas as redes.

O Interconnect em nuvem oferece duas opções para expandir sua rede local:

* **O Interconnect Dedicado**

Fornece uma conexão física direta entre sua rede local e a rede do Google.

* **O Interconnect de Parceiro**

Fornece conectividade entre suas redes locais e as redes VPC através de um provedor de serviços compatível.

No diagrama abaixo, evidencio o interconnect de parceiro que criamos para a conta da Americasmed, conectando redes locais ao VPC de produção, homologação e desenvolvimento da GCP. Essa conexão é composta por 1 link de 1 Gbps redundante, fornecendo 99,9% de SLA, criado na zona metropolitana de São Paulo (Instalações do parceiro Equinix) anexados apenas à região southamerica-east1. Embora tenhamos duas regiões disponíveis, uma em São Paulo/Brasil e a outra na região Central dos Estados Unidos, a comunicação dos Estados Unidos com os ambientes OnPremise e Demais clouds, se dará temporariamente através da região de SP na GCP, por uma questão de contratação (Em andamento), da segunda região de disponibilidade da Equinix (RJ), esta então se conectaria diretamente a região dos Estados Unidos no futuro.

Diagrama

O conteúdo gerado por IA pode estar incorreto.

# **6. Conexões Internas - VPC - Virtual Private Cloud**

A Nuvem Virtual Privada (VPC) fornece funcionalidade de rede para instâncias de máquinas virtuais (VM) do Compute Engine, clusters do Google Kubernetes Engine (GKE) e o ambiente flexível do App Engine. A VPC fornece rede para seus recursos e serviços baseados em nuvem que são globais, escaláveis e flexíveis. A nova arquitetura de rede foi projetada para separação de ambientes para fornecer melhor escalabilidade e maior segurança, por exemplo, não permitindo a comunicação entre os ambientes de desenvolvimento e produção, evitando erros e incidentes.

O diagrama a seguir retrata o modelo compartilhado de rede projetado:

Interface gráfica do usuário

O conteúdo gerado por IA pode estar incorreto.

No planejamento de rede acima foram criados 1 rede Hub (VPC Global) e 3 Spokes (VPC-PRD, VPC HML e VPC-DEV) com as seguintes características cada um:

* VPCs compartilhadas para estender suas sub-redes a todos os recursos das Americasmed através de projetos de serviço;
* O Controle centralizado sobre grupos de rede e segurança;
* Arquitetura de rede altamente segregada, segura e escalável conectada a um hub (VPC Global).
* Todas as VPCs estão divididas em projetos diferentes, elas não podem se comunicar entre si;
* Foi criada uma faixa de Acesso a Serviços Privados para alcançar recursos gerenciados pelo Google;
* Cada projeto de rede tem suas próprias regras de FW para acesso a redes específicas dos seus respectivos ambientes;
* Cada projeto de rede contem suas próprias configurações dos recursos de cloud-router, fw, nat-gw.

**Nota:** A VPC Peering não é transitiva, o que significa que não é possível alcançar recursos gerenciados pelo Google através de peering de Acesso a Serviços Privados ao alcançar as VPCs através de outro VPC Peering (Entre VPC Global e VPCs-Internas). Finalmente, o VPC Global das Americasmed atua como um projeto/VPC para hospedar serviços compartilhados entre todos os Spokes e também para fornecer conectividade entre redes novas e atuais, como VPC-PRD > OnPremises e/ou outros provedores de nuvem.

Abaixo trago também o planejamento e range de redes em uso e disponíveis, conforme solicitado o detalhamento pelo time de CloudOps:

Interface gráfica do usuário, Tabela

O conteúdo gerado por IA pode estar incorreto.

Tabela

O conteúdo gerado por IA pode estar incorreto.

Tabela

O conteúdo gerado por IA pode estar incorreto.

A conectividade entre a VPC-Global e as VPCs Internas, utilizam VPC Peering para desempenho máximo e todas as redes conectadas e não possuem intervalos CIDR sobrepostos. A conectividade entre a VPC-Global e as VPCs Internas estão compartilhadas para fornecer transitividade e controle sobre as rotas trocadas. Durante as discussões, foi definido que não há necessidade de inspeção de pacotes entre VPCs/premissas e que um firewall de rede é suficiente para garantir a segurança.

**Nota sobre as rotas:** As rotas são uma característica fundamental de cada VPC para garantir que os pacotes sejam entregues aos destinos corretos. Existem vários tipos de rotas, como rotas de sub-rede, geradas pelo sistema, estáticas e dinâmicas.

É importante lembrar sobre conectividade na arquitetura:

* **Emparelhamento VPC entre o VPC-Global e as VPCs Internas:**

Nem todas as rotas de sub-rede são trocadas automaticamente entre as VPCs envolvidas e a exportação/importação de rotas personalizadas precisou ser habilitada, além de configurar regras internas e externas de firewall (Equinix / OnPremises / Outras Clouds), para permitir efetivamente que a comunicação ocorra.

# **6.1 Conectividade Privada**

As opções de acesso privado são formas de recursos dentro de uma VPC acessarem APIs do Google sem um IP externo e mantendo o tráfego dentro da rede do Google.

Além do Acesso a Serviços Privados, temos mais 2 opções especificamente para APIs do Google:

* **Acesso Privado ao Google Instâncias de VM**

Que possuem apenas endereços IP internos (sem endereços IP externos) podem usar o Acesso Privado ao Google. Elas podem alcançar os endereços IP externos das APIs e serviços do Google. O endereço IP de origem do pacote pode ser o endereço IP interno principal da interface de rede ou um endereço em um intervalo de IPs alias que é atribuído à interface:

Interface gráfica do usuário, Aplicativo

O conteúdo gerado por IA pode estar incorreto.

Esse recurso pode (e foi decidido que) ser habilitado em cada sub-rede e alguns requisitos precisam ser atendidos:

* O Acesso Privado ao Google não tem efeito em instâncias com endereços IP externos.
* Sua rede deve ter rotas apropriadas para os intervalos de IP de destino usados pelas APIs e serviços do Google. Essas rotas devem usar o gateway de internet padrão como próximo salto.

Se você desativar o Acesso Privado ao Google, as instâncias de VM não poderão mais acessar as APIs e serviços do Google; elas poderão apenas enviar tráfego dentro da rede VPC.

* **Conexão de Serviço Privado**

Outra opção é o Private Service Connect, que permite o consumo privado de serviços entre redes VPC que pertencem a diferentes grupos, equipes, projetos ou organizações. O PSC também permite o acesso privado às APIs do Google.

Para este design, um PSC será configurado em cada VPC:

Diagrama

O conteúdo gerado por IA pode estar incorreto.

* **Endpoints**

Foram configurados alguns PSCs para acessar as APIs do Google em cada VPC do ambiente das Americasmed. A configuração permite o acesso a todas as APIs do pacote private.googleapis.com.

Diagrama

O conteúdo gerado por IA pode estar incorreto.

Na tabela abaixo, todos os endpoints PSC por VPC:

* **Endpoints do Private Service Connect**

Os endpoints para VPCs conectadas são necessários porque esses endpoints não são acessíveis através do VPC Peering:

* + **SP**

VPC-PRD: 10.37.41.10

VPC-HML: 10.37.81.10

VPC-DEV: 10.37.121.10

* + **US**

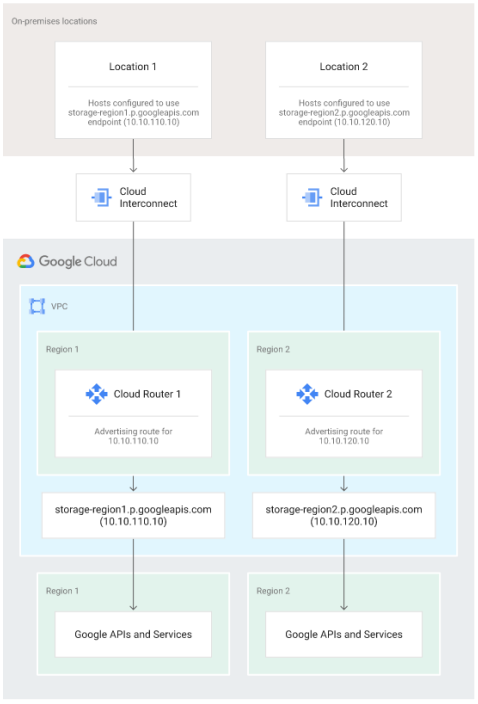
VPC-PRD: 10.37.171.10

VPC-HML: 10.37.211.10

VPC-DEV: 10.37.251.10

O acesso a partir de redes locais também é possível configurando um anúncio de rota customizada do Cloud Router para anunciar rotas para o endpoint do Private Service Connect na sessão BGP que gerencia rotas para a anexação VLAN.

Confira o diagrama de referência para ilustração:



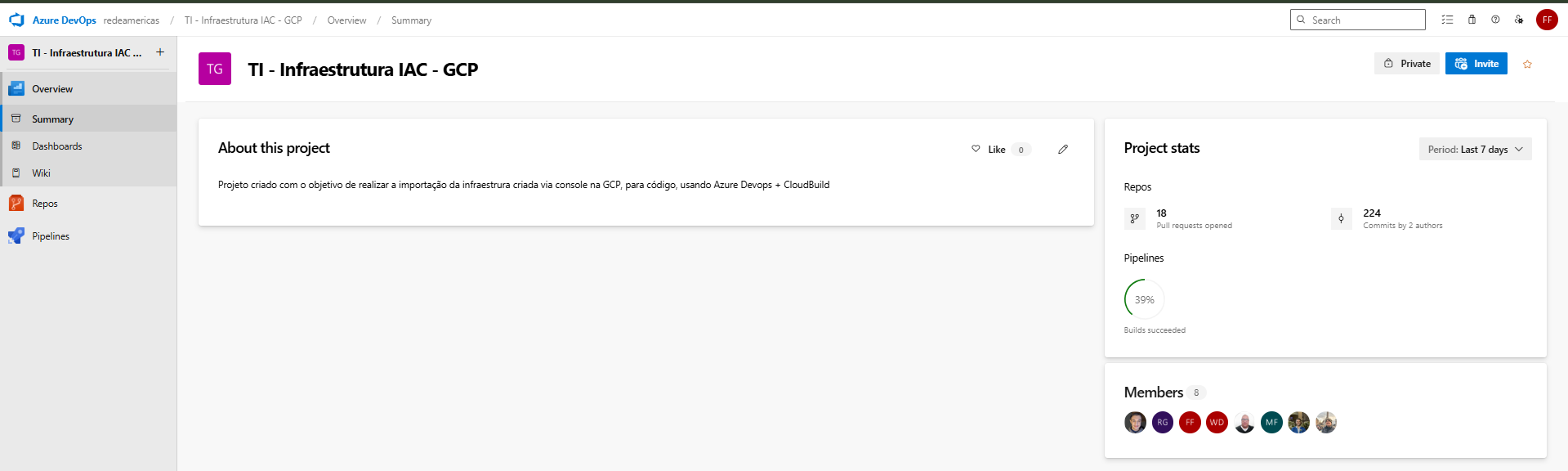
**Nota:** Ao enviar tráfego para APIs e serviços do Google, não direcione o tráfego através de uma VM de salto ou através de um balanceador de carga TCP/UDP interno de salto. Em vez disso, direcione esse tráfego através de um gateway de internet padrão de salto. Isso inclui o tráfego para APIs e serviços do Google enviados por meio do Acesso Privado do Google.

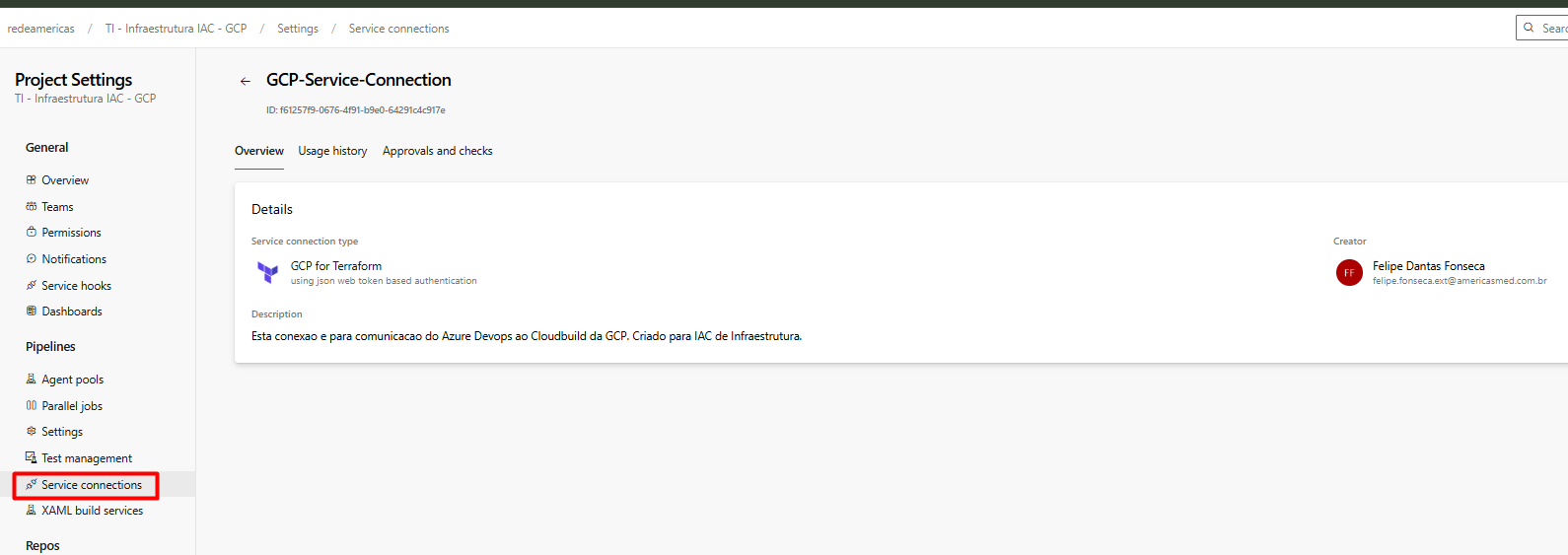
# **7. IAC - Infra As Code - Terraform, Azure Devops & CloudBuild**

Neste projeto o cliente define o repositório do Azure Devops como repositório de código oficial e o Cloud Build como serviço para execução dos deploys de infraestrutura na GCP.

Dentro do Azure Devops temos o projeto “TI-Infraestrutura IAC – GCP”, que contém todos os membros, configurações de conexão com o Cloud Provider GCP, o repositório com o código de infraestrutura dos recursos criados na GCP, além da pipeline com as informações relativas às execuções de deploy aos ambientes de DEV / HML e PRD.

Abaixo seguem as evidências da estrutura configurada:





Interface gráfica do usuário, Texto, Aplicativo, Email

O conteúdo gerado por IA pode estar incorreto.

Interface gráfica do usuário, Texto, Aplicativo, Email

O conteúdo gerado por IA pode estar incorreto.

Interface gráfica do usuário, Texto, Aplicativo, Email

O conteúdo gerado por IA pode estar incorreto.

Interface gráfica do usuário, Texto, Aplicativo

O conteúdo gerado por IA pode estar incorreto.

Interface gráfica do usuário, Aplicativo, Teams

O conteúdo gerado por IA pode estar incorreto.

Texto

O conteúdo gerado por IA pode estar incorreto.

Texto

O conteúdo gerado por IA pode estar incorreto.

Texto

O conteúdo gerado por IA pode estar incorreto.

Texto

O conteúdo gerado por IA pode estar incorreto.

Abaixo, evidencio o acionamento do CloudBuild para a execução do deploy, logo após a aprovação da fase do Apply no Pipeline do Azure Devops. Aqui eu consigo ver o que esta sendo Adicionado, Alterado ou Excluído na GCP nos relatórios de log, simplesmente clicando em cima do link do Build criado durante esta execução.

Tabela

O conteúdo gerado por IA pode estar incorreto.

Interface gráfica do usuário, Texto, Aplicativo, Email

O conteúdo gerado por IA pode estar incorreto.

# **7.1 Repositório e Estrutura dos TFs - Visão Geral**

Este repositório implementa a infraestrutura fundamental de múltiplos projetos GCP ( DEV, HML, PRD ) usando Terraform de forma altamente modular. O processo de execução \*\*não roda Terraform direto no agente do Azure DevOps\*\*; em vez disso, cada estágio ( Plan / Apply ) dispara um \*\*Cloud Build\*\* que executa um script orquestrador (**`.cloudbuild/terraform\_runner.sh`**). Esse script percorre uma lista ordenada de diretórios ("módulos" autocontenidos) definida em arquivos **`deploy\_order\_\*.txt`**, inicializa cada módulo, faz \*plan/apply\* e lida automaticamente com:

* Backend remoto em \*\*GCS\*\* ( State por módulo + Ambiente );
* Força de unlock de states travados;
* Auto-import de recursos já existentes ( "already exists" ) por módulo + ambiente);
* Tolerância a dependências entre módulos via ordem explícita;
* Heurísticas avançadas de dedução de IDs para **`terraform import`**;
* Geração de lock multi-plataforma do **`.terraform.lock.hcl`**;
* Cada ambiente possui árvore própria sob **`iac-infra-coetec/`**:

**`americas-devops-core-cicd/` ( Projeto de apoio / Ferramentas )**

**`xamericas-network-dev/`**

**`xamericas-network-hml/`**

**`xamericas-network-prd/`**

# **7.2 Repositório e Estrutura dos TFs - Estrutura de Diretórios (Padrão de Módulo)**

# Dentro de cada projeto / ambiente existem pastas por tipo de recurso GCP ( Ex.: **`networks/`, `subnetworks/`, `firewall/`, `iam/`, `kms/`, `logging/`, `cloudsql/`**, etc). Dentro de cada tipo há subpastas de escopo geográfico :

* **`global`** → Recursos de âmbito global
* **`<região>`** → Ex.: **`southamerica-east1`, `us-central1`**

Cada subpasta ( Ex.: **`xamericas-network-dev/networks/southamerica-east1/`**) é um \*\*módulo independente\*\* contendo arquivos **`.tf`** com:

* **`provider.tf`** ( Backend + provider google com **`project`** + **`region`** / **`global`**)
* Recursos Terraform importados (**`resource "google\_\*" ...`**) – muitos nomes seguem padrão **`tfer--...`** indicando origem de importação via Terraformer originalmente
* Possíveis **`variables.tf`, `outputs.tf`, `\_standard\_variables.tf`**

Backend / State **:** O backend é sempre **`gcs`**. Ex. **:**

terraform {

  backend "gcs" {

    bucket = "americasmed-iac-bucket"

    prefix = "infra/state/<projeto>/<módulo>"

  }

}

O **`prefix`** diferencia ambiente + caminho lógico ( Ex.: **`infra/state/xamericas-network-dev/gcs-global`**). Assim, \*\* Cada diretório tem state isolado \*\* minimizando blast radius e facilitando reprocessamento granular.

# **7.3 Repositório e Estrutura dos TFs - Estrutura de Diretórios ( Arquivos de Ordem de Deploy )**

Os arquivos: **`deploy\_order\_dev.txt`, `deploy\_order\_hml.txt`, `deploy\_order\_prd.txt`,** eles listam, linha a linha, os diretórios ( Módulos ) a serem processados. Linhas iniciadas por **`#`** são comentários.

A ordem impõe dependências lógicas:

* **\*\* Bootstrap do backend \*\* (`gcs/\*`) –** Garante buckets para state e artefatos;
* **\*\* Fundamentos \*\* –** **`project`, `iam`, `kms`, `logging`, `monitoring`;**
* **\*\* Rede base \*\* –** **`networks`, `subnetworks`, `routers`, `routes`, `firewall`;**
* **\*\* Conectividade \*\* –** VPN / Interconnect / Tunnels;
* **\*\* Addresses / globalAddresses \*\* –** ( Após rede );
* **\*\* Compute base \*\* –** Imagens, disks, policies, reservations;
* **\*\* Instâncias & escalonamento \*\* –** Templates, instances, grupos, autoscalers;
* **\*\* Load Balancing \*\* –** Health checks, NEGs, backend services, URL maps, proxies, forwarding rules, SSL, security policies;
* **\*\* Serviços gerenciados \*\* –** Cloud Build, Cloud Functions, Cloud SQL, BigQuery, DataProc, MemoryStore, Pub/Sub, Scheduler, etc;
* **\*\* Ambiente PRD \*\* –** Inclui também **`gke/\*`** em pontos específicos.

Comentários no arquivo indicam módulos desativados ou tratados de forma especial ( Ex.: duplicados PSC, dependências vazias, etc ).

# **7.4 Repositório e Estrutura dos TFs - Estrutura de Diretórios (Pipelines Azure Devops - `azure-pipelines.yml` )**

O Fluxo de Stages :

* **\*\* Detect \*\* –** Determina se houve mudanças em diretórios de cada ambiente comparando com commit anterior. Expõe variáveis **`CHANGED\_DEV`, `CHANGED\_HML`, `CHANGED\_PRD`**
* **\*\* Dev / Hml / Prd (paralelos condicionais) \*\* –**

Job **`Plan\_<Env>`** → Dispara Cloud Build com **`\_TF\_ACTION=plan`**

Job **`Approve\_<Env>`** → `Manual Validation` ( Janela até 24h )

Job **`Apply\_<Env>`** → Dispara Cloud Build com **`\_TF\_ACTION=apply -auto-approve`**

Parâmetros de Força :

* **`forceDev`, `forceHml`, `forcePrd` (boolean) –** Permitem executar mesmo sem diffs detectados.

Substituições enviadas ao Cloud Build :

* Incluem **`\_ORDER\_FILE`, `\_PROJECT\_ID`, `\_TARGET\_PROJECT\_PATH`, `\_TF\_ACTION`** e flags de desbloqueio **(`\_TF\_AUTO\_FORCE\_UNLOCK`, etc.)**

Autenticação GCP :

O pipeline usa variável secreta Base64 **(`GCP\_SERVICE\_KEY\_BASE64`)** para gerar **`gcp-key.json`** e autenticar **`gcloud auth activate-service-account`** antes de chamar **`gcloud builds submit`**

# **7.5 Repositório e Estrutura dos TFs - Estrutura de Diretórios ( Cloud Build - `.cloudbuild/cloudbuild.yaml` )**

Define :

* Versão Terraform **(`\_TERRAFORM\_VERSION`)**
* Container de execução **`hashicorp/terraform:<version>`**
* Volume de cache compartilhado de plugins **(`/root/.terraform.d/plugin-cache`)**
* Invoca **`./.cloudbuild/terraform\_runner.sh`** após exportar variáveis de ambiente, como a da conta de serviço “Devops” usada para backend do CloudBuild :

**Service Account: `devops-service@americas-devops-core-cicd.iam.gserviceaccount.com`**

# **7.6 Repositório e Estrutura dos TFs - Estrutura de Diretórios ( Script Orquestrador - `terraform\_runner.sh` )**

Responsável por todo o ciclo de cada módulo listado no **`\_ORDER\_FILE`**e suas principais capacidades são :

* Carregamento de variáveis e auto-dedução de **`TF\_VAR\_gcp\_project\_id`, `TF\_VAR\_network\_name`**,região por diretório;
* Suporte a arquivo **`.tfvars`** autodetectado **(`envs/<env>.tfvars`, `<project>.tfvars`, `default.tfvars`);**
* Execução ordenada do **Plan / Apply** com até **3** tentativas por módulo **( Retries com atraso );**
* **\*\* Auto unlock \*\*** de state travado **(force-unlock) + fallback** via remoção do objeto **`.tflock`** no bucket se configurado**;**
* **\*\* Auto-import avançado \*\*** ao detectar erro 409 **`already exists`,** tenta deduzir IDs de recursos e importar automaticamente antes de repetir o **Apply;**
* Dedução heurística para diversos tipos **(`google\_compute\_\*`, `google\_dns\_managed\_zone`, `url\_map`, proxies, certificados, etc.);**
* Geração opcional de lock multi-plataforma para **`.terraform.lock.hcl` (`terraform providers lock`);**
* Filtros que pulam módulos redundantes **(** Ex.: **`globalForwardingRules`** não globais**, `globalAddresses`** regionais **);**
* Tratamento de **NAT auto-addresses (** Não criáveis manualmente **)** via import**;**
* Sanitização de **IDs import (** Remove duplicações de **self\_link**, normalizando Region / Zone **);**
* Modo de **import declarativo** via **CSV (`\_IMPORT\_MAP`),** com formato: **`module\_path;terraform\_address;id`.**

Variáveis / Flags Importantes ( Definíveis via substituições do Cloud Build ou export no shell local **) :**

* **`\_ORDER\_FILE` –** Arquivo de ordem **( Default: `deploy\_order\_dev.txt`);**
* **`\_TARGET\_PROJECT\_PATH` –** Raiz do ambiente a filtrar **( Ex.: `iac-infra-coetec/xamericas-network-dev`);**
* **`\_TF\_ACTION` –** Ex.: **`plan -input=false`** ou **`apply -auto-approve -input=false`;**
* **`\_TFVARS\_FILE` – var-file** explícito **( Override do autodetect );**
* **`\_TF\_LOCK\_TIMEOUT` –** Timeout de lock Terraform **( Ex.: `10s`);**
* **`\_TF\_AUTO\_FORCE\_UNLOCK`** / **`\_TF\_FORCE\_UNLOCK\_ANY`** / **`\_TF\_FORCE\_UNLOCK\_MAX`** / **`\_TF\_AUTO\_FORCE\_UNLOCK\_GSUTIL` –** Política de desbloqueio;
* **`\_AUTO\_IMPORT`** / **`\_AUTO\_IMPORT\_TYPES`** / **`\_AUTO\_IMPORT\_EAGER`** / **`\_AUTO\_IMPORT\_DEBUG` –** Controle de auto-import;
* **`\_DEDUCE\_ID\_GENERIC` –** Ativa heurística genérica de ID;
* **`\_NETWORK\_NAME` –** Força nome da VPC ( Senão deduz por projeto );
* **`\_IMPERSONATE\_SA` –** Impersonação de Service Account **(** Gera token e exporta **`GOOGLE\_OAUTH\_ACCESS\_TOKEN`);**
* **`\_IMPORT\_MAP` –** Caminho CSV de imports declarativos.

Fluxo Simplificado Interno :

* Ler **`\_ORDER\_FILE`** linha a linha**;**
* Para cada módulo: validar diretório, detectar região, **`terraform init` (** Reconfigure + upgrade **);**
* Ajustar lock **multi-plataforma, se necessário;**
* Executar **`Terraform Plan | Apply`** com **Retries;**
* Em erro 409: **Auto-import e Retry;**
* Em lock preso: **Force-unlock (** Respeitando políticas **);**
* **Prosseguir até o final e imprimir sucesso.**

# **7.7 Repositório e Estrutura dos TFs - Estrutura de Diretórios ( Execução Local - Desenvolvedores / Debug )**

Pré-Requisitos :

* **`gcloud`** Autenticado com acesso ao projeto e bucket de state**;**
* **`terraform` (** Ideal: mesma versão do **`\_TERRAFORM\_VERSION`** especificado – atualmente 1.3.9 **);**
* Permissões adequadas na **GCP** para **CRUD** dos recursos **(** Ou ao menos leitura para Plan **)**.

Passos :

**```bash**

1. **Clone do repositório**

cd ~/git/americasmed/TI\ -\ Infraestrutura\ IAC\ -\ GCP

1. **Autenticar gcloud ( Se necessário )**

gcloud auth login

**# Se usar ADC para Terraform**

gcloud auth application-default login

1. **( Opcional ) Selecionar projeto base para bootstrap de APIs compartilhadas**

gcloud config set project americas-devops-core-cicd

1. **Executar apenas um plan para DEV**

env PROJECT\_ID=xamericas-network-dev \

\_ORDER\_FILE=deploy\_order\_dev.txt \

\_TARGET\_PROJECT\_PATH=iac-infra-coetec/xamericas-network-dev \

\_TF\_ACTION="plan -input=false" \

bash .cloudbuild/terraform\_runner.sh

1. **Executar apply controlado para um único módulo ( Ex: Sub-networks Região SP )**

cd iac-infra-coetec/xamericas-network-dev/subnetworks/southamerica-east1

terraform init -reconfigure

terraform plan

**# Revisar**

terraform apply -auto-approve

Dicas :

* Para aplicar **\*\* Todos os módulos \*\*** mantenha na raiz e deixe o runner percorrer o arquivo de ordem;
* Ajuste **`\_AUTO\_IMPORT=0`** Se quiser bloquear import automático ( Ex.: auditorias );
* Use **`\_AUTO\_IMPORT\_DEBUG=1`** para obter logs detalhados de dedução de Ids;
* Para forçar re-download de providers: **export `\_TF\_CLEAN\_LOCAL=1`** antes de executar.

# **7.8 Repositório e Estrutura dos TFs - Estrutura de Diretórios ( Adicionando Novo Módulo )**

Adicionando um Novo Módulo :

* Criar pasta no ambiente correto seguindo padrão : **`iac-infra-coetec/<projeto>/<tipo>/<global|região>`;**
* Incluir **`provider.tf`** com backend consistente **(** Mesmo bucket e prefix exclusivo **):**

   ```hcl

   terraform {

     backend "gcs" {

       bucket = "americasmed-iac-bucket"

       prefix = "infra/state/<projeto>/<tipo>-<escopo>"

     }

     required\_providers {

       google = { source = "hashicorp/google" version = ">= 4.70.0" }

     }

   }

   provider "google" {

     project = "<projeto>"

     region  = "<global|região>"

   }

   ```

* Declarar variáveis necessárias **(** Ou reutilizar variáveis padrão **`\_standard\_variables.tf`** se aplicável **);**
* Criar recursos **`resource "google\_\*"`** com nomes claros **( Evite manter prefixo `tfer--` em novos recursos );**
* Adicionar a linha do diretório no **`deploy\_order\_<env>.txt`** respeitando dependências;
* Executar **`plan` ( Cloud Build ou local )** e revisar **diff;**
* Ajustar **`iam`**, **`kms`** ou outras dependências se o módulo referenciar outputs não existentes**.**

# **7.9 Repositório e Estrutura dos TFs - Estrutura de Diretórios ( Convenções / Boas Práticas )**

Abaixo uma lista de boas práticas para a estrutura de TFs dos projetos :

* **\*\* Isolamento de State \*\* - Um state por módulo →** Recuperação rápida e mitigação de corrupção;
* **\*\* Nomes de Recursos \*\* - Preferir padrão descritivo →** `<contexto>-<env>-<região>-<função>` (ex.: `spk-dev-sp-private-1`);
* **\*\* Rotas & Firewall \*\* - Separar regras por escopo para facilitar auditoria;**
* **\*\* APIs de Projeto \*\* - Mantidas em módulo → `project/global`** através de **`google\_project\_service`;**
* **\*\* Evitar Dependências Implícitas \*\* - Preferir outputs + `data.terraform\_remote\_state` (** Já observado em **`subnetworks`** usando **`network\_self\_link`).**

# **7.10 Repositório e Estrutura dos TFs - Estrutura de Diretórios ( Importando Recursos Existentes )**

O Processo de Importação pode ser Automático :

**Ao detectar erro 409 ( Resource Already Exists ), o runner tenta :**

* Extrair endereço Terraform **(`with google\_\*.<name>,`);**
* Deduzir ID canônico **( `projects/<proj>/regions/<region>/...` ou global );**
* Executar **`terraform import`;**
* Repetir **`apply`.**

Declarativo ( CSV ) :

**Criar arquivo CSV ( Ex.: `imports\_templates/imports\_dev\_core.csv` ) com :**

**```csv**

**<module\_path>;<terraform\_address>;<resource\_id>**

**```**

Exemplo :

**```csv**

**iac-infra-coetec/xamericas-network-dev/networks/southamerica-east1;google\_compute\_network.tfer--vpc-dev;projects/xamericas-network-dev/global/networks/vpc-dev**

**```**

Executar com **`\_IMPORT\_MAP=<caminho>`** e opcionalmente **`\_AUTO\_IMPORT\_EAGER=1`.**

Ou importação Manual **Dentro do módulo :**

**```bash**

**terraform init -reconfigure**

**terraform import google\_compute\_network.vpc projects/xamericas-network-dev/global/networks/vpc-dev**

**terraform plan**

**```**

# **7.11 Repositório e Estrutura dos TFs - Estrutura de Diretórios (IAM & Segurança)**

* **Chave de Service Account -** Usada pelo pipeline fica em variável segura **Base64. \*\*Nunca\*\* Versionar `gcp-key.json`;**
* **Avaliar redução de permissões do SA - ( `devops-service` )** ao mínimo necessário **(** Principais: Cloud Build, Storage ( State ), Compute, Networking, IAM read / limited write conforme necessidade )**;**
* **Usar `force-unlock` com cautela -** Remover lock manualmente via **`gsutil rm`** apenas quando confirmado que não há execução ativa**.**

# **7.12 Repositório e Estrutura dos TFs - Estrutura de Diretórios ( Troubleshooting )**

**| Situação | Causa Provável | Ação |**

* **`Error acquiring the state lock` -** Execução anterior abortou **( Lock órfão )** | Esperar timeout ou permitir **auto-force-unlock;** usar **`terraform force-unlock <ID>`;**
* **`already exists` -** Recurso criado fora do Terraform / import pendente **| Auto-import** deve resolver; senão fazer import manual;
* **`invalid\_grant / invalid\_rapt` -** Sessão gcloud expirada ou impersonação sem token Creator;
* **`gcloud auth login --update-adc`** e verificar role **`serviceAccountTokenCreator`;**
* **`Unsupported attribute`** de remote state **-** Módulo dependente rodou antes do produtor | Garantir ordem no **`deploy\_order\_<env>.txt`;**
* **Checksum mismatch provider | Cache** compartilhado corrompido **|** Script purga e reinicia**;** local: apagar **`~/.terraform.d/plugin-cache`.**

# **7.13 Repositório e Estrutura dos TFs - Estrutura de Diretórios ( Atualização de Providers / Terraform )**

* Preferir atualizar versão no **`cloudbuild.yaml`** e alinhar localmente;
* Gerar lock **multi-plataforma automaticamente ( Já habilitado )** para evitar **drift** entre agentes**.**

# **7.14 Repositório e Estrutura dos TFs - Estrutura de Diretórios ( Fluxo de Contribuição - Pull Request )**

* Criar branch feature;
* Alterar / adicionar módulos e posicionar no arquivo de ordem;
* Executar **`plan`** local ( Ou Cloud Build forçado: Parâmetro **`force\*`**);
* Revisar diffs e impactos;
* Abrir PR → Ao merge em **`main`**, pipeline executará automaticamente.

# **7.15 Repositório e Estrutura dos TFs - Estrutura de Diretórios ( Execução Seletiva / Forçada )**

* Usar parâmetros **`forceDev`, `forceHml`, `forcePrd`** ao iniciar pipeline manualmente para forçar Plan / Apply mesmo sem mudanças detectadas.

# **7.16 Repositório e Estrutura dos TFs - Estrutura de Diretórios ( Boas Práticas Adicionais Recomendadas )**

* Introduzir validação de formato ( Ex.: **`tflint`, `terraform fmt -check`)** em estágio futuro;
* Publicar artefatos de **`plan`** ( Ex.: salvar **`plan.out`**) para auditoria antes da aprovação manual;
* Adicionar notificação ( **`notifyUsers`** ) no **`ManualValidation`** para responsáveis de cada ambiente;
* Criar **`envs/<env>.tfvars`** padronizados para consolidar variáveis futuras.

# **7.17 Repositório e Estrutura dos TFs - Estrutura de Diretórios ( Glossário Rápido )**

* **\*\* Módulo ( Neste contexto ) \*\* -** Diretório isolado com backend e recursos próprios (não é necessariamente um módulo reutilizável Terraform);
* **\*\* State isolado \*\* -** Cada módulo tem seu próprio arquivo de estado remoto em GCS;
* **\*\* Auto-import \*\* -** Processo automatizado de executar **`terraform import`** antes de novo **`apply`** em caso de conflito de existência.

# **7.18 Repositório e Estrutura dos TFs - Estrutura de Diretórios ( Resumo )**

Este design privilegia Segurança, Granularidade e Resiliência de execução: Pequenos estados, ordem explícita, tolerância a recursos preexistentes e automação robusta de recuperação de travas e imports. Amplia a confiabilidade do provisionamento multi-ambiente e reduz esforço manual.