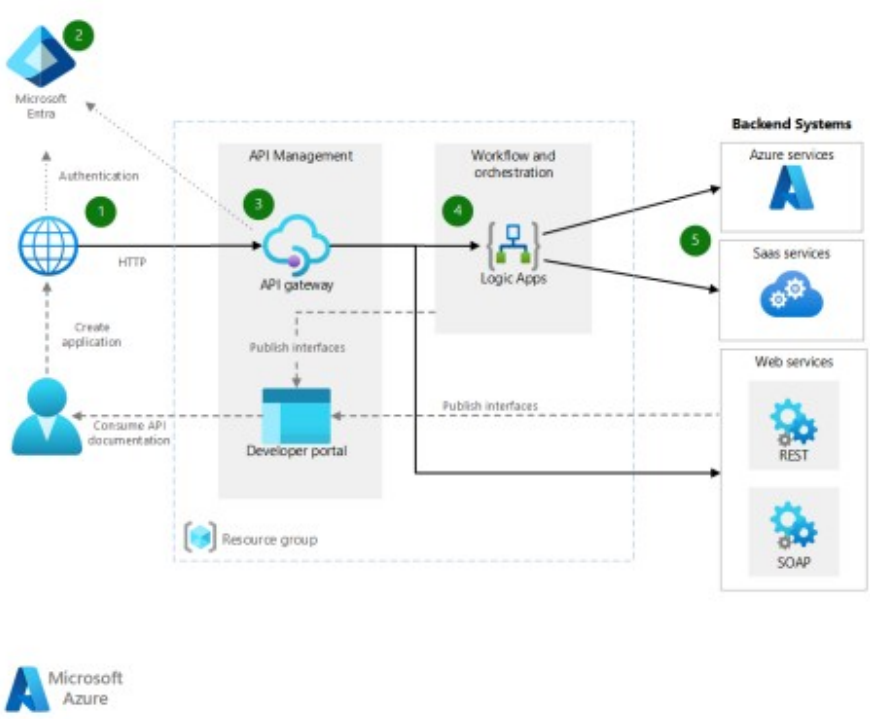
Relatório de análise da solução: tmpzooet9je



Análise completa da solução atual

Modelo de cloud:
- Microsoft Azure

Lista com os componentes:
- Identidade e acesso: Microsoft Entra ID (antigo Azure AD)
- Camada de entrada: API Management (APIM)
  - API Gateway
  - Developer Portal
- Orquestração: Azure Logic Apps
- Back-end systems:
  - Azure services (ex.: Functions, Storage, Service Bus, Cosmos DB)
  - SaaS services (ex.: Microsoft 365, Salesforce, ServiceNow, etc.)
  - Web services externos (REST e SOAP)
- Clientes/Desenvolvedores:
  - Aplicativos clientes que consomem as APIs via HTTPs
  - Desenvolvedores que registram apps e consultam documentação
- Limites de rede e gestão:
  - Resource Group (agrupa APIM e portal)
  - Conectividade pública/privada (Private Endpoints, VNET Integration) – implícitos
  - Observabilidade (Azure Monitor/Log Analytics, App Insights) – implícito

Interação entre os componentes:
- 1. O cliente chama as APIs publicadas via HTTPS no API Gateway do APIM.
- 2. O cliente obtém tokens (OAuth 2.0/OIDC) no Microsoft Entra ID; o APIM valida o token, escopos e claims.
- 3. O APIM aplica políticas (rate limiting, validação de schema, transformação, masking) e encaminha a chamada.
- 4. O APIM invoca Logic Apps para orquestrar fluxos, enriquecimento e integrações.
- 5. Logic Apps executa conectores e chama os back-ends:
  - Serviços Azure (por Managed Identity/Private Endpoint)
  - SaaS (via conectores OAuth)
  - Web services REST/SOAP externos (TLS, validação mútua opcional)
- Desenvolvedores usam o Developer Portal para:
  - Registrar aplicações, obter chaves/subscrições (se usado), ler documentação, testar APIs em sandbox.

O que esse sistema faz:
- Publica um front door de APIs sob um único gateway, autenticado via Microsoft Entra ID.
- Orquestra integrações e processos de negócio com Logic Apps.
- Expõe, de forma segura e governada, serviços internos (Azure), SaaS e serviços web externos (REST/SOAP) a consumidores internos/externos.
- Fornece portal para onboarding de desenvolvedores e gestão de subscrições das APIs.

Vulnerabilidades e Solução para cada vulnerabilidade:
- Tráfego cliente → APIM
  - Risco: Downgrade/uso de protocolos fracos, roubo de tokens, ausência de limitação de taxa.
  - Mitigações: Forçar TLS 1.2+ e HSTS; usar OAuth 2.0/OIDC com PKCE; opcional mTLS; rate limiting/burst control; WAF frente ao APIM (Azure Front Door/WAF ou App Gateway WAF); validação de IP/Geo.
- Identidade (Microsoft Entra)
  - Risco: App registrations excessivamente permissivas, tokens longos, consent phishing.
  - Mitigações: Conditional Access, MFA; lifetimes curtos; Admin consent; revisões de consentimento; credenciais de app com certificado/Managed Identity; Identity Protection.
- API Management
  - Risco: Políticas mal configuradas (pass-through de cabeçalhos sensíveis), falha na validação de JWT/escopos, exposição de URLs internas, cache de dados sensíveis.
  - Mitigações: Políticas de validate-jwt, validate-content, limit-concurrency; remoção/normalização de headers; redaction em logs; desabilitar trace público; ocultar backends; private networking (VNET/Private Link); chaves por subscrição e RBAC.
- Developer Portal
  - Risco: Tomada de conta de conta, enumeração de usuários, divulgação indevida de docs/segredos em exemplos.
  - Mitigações: Autenticar via Entra ID/MFA; aprovar manualmente publishers; revisão de conteúdo; rate limit/logins; CAPTCHA; esconder APIs privadas.
- Logic Apps
  - Risco: Segredos em texto claro, histórico de execuções expondo PII, loops que causam DoS/custos, conectores com privilégios excessivos.
  - Mitigações: Managed Identity + Key Vault; Secure Inputs/Outputs habilitado; políticas de retry com backoff e circuit breaker; idempotência; RBAC mínimo necessário; Private Endpoints; Data Loss Prevention (DLP) em conectores.
- Back-ends Azure
  - Risco: Exposição pública indevida, SAS tokens fracos/sem expiração, falta de validação de schema.
  - Mitigações: Private Endpoints, firewall/IP restriction; rotação e tempo de vida mínimo para SAS; validação de schema; Defender for Cloud e PIM para operações.
- Web services REST/SOAP externos
  - Risco: TLS fraco, XXE em SOAP, falhas de certificação.
  - Mitigações: TLS 1.2+, pinning/validação de certificado; desabilitar XXE; validação de payload por schema; timeouts e retry com jitter.
- Observabilidade e logs
  - Risco: Vazamento de PII/segredos em logs; falta de detecção.
  - Mitigações: Data masking/redaction no APIM e Logic Apps; coleta central (Log Analytics); alertas (Defender for Cloud, Sentinel); retenção mínima necessária.
- Governança/RBAC
  - Risco: Privilégios excessivos, mudanças não auditadas.
  - Mitigações: RBAC mínimo, PIM para privilégios just-in-time; Azure Policy/Blueprints; CI/CD com revisões e aprovação; bloqueio de recursos críticos (resource locks).
- Supply chain
  - Risco: Conectores/SDKs vulneráveis, dependências de terceiros.
  - Mitigações: Inventário de dependências (SBOM), verificação de assinaturas, atualização contínua, revisão de conectores SaaS e escopos.

Gere um Relatório de Modelagem de Ameaças, baseado na metodologia STRIDE:
- Escopo e fronteiras de confiança
  - Fronteiras: Internet → APIM; APIM → Logic Apps; Logic Apps → Back-ends (Azure, SaaS, externos); Área administrativa (Dev Portal, Azure Portal).
  - Ativos: Endpoints de API, tokens OAuth/JWT, segredos/credenciais, dados de clientes/PII, configurações APIM/Logic Apps, logs/telemetria.
- S — Spoofing (Falsificação de identidade)
  - Ameaças: Uso de tokens roubados; spoof de cliente sem mTLS; impersonação de serviço externo.
  - Controles: OAuth2/OIDC com validate-jwt no APIM; mTLS onde aplicável; Managed Identity para chamadas a Azure; pinning/validação de certificado a terceiros; Conditional Access e MFA.
- T — Tampering (Adulteração)
  - Ameaças: Manipulação de requests/headers; alteração de políticas APIM; payloads SOAP com injeção/XXE.
  - Controles: TLS extremo a extremo; políticas de validação de conteúdo e schema; assinaturas de

mensagem onde exigido; CI/CD com code review e IaC; RBAC/PIM e activity logs imutáveis.
- R — Repudiation (Repudiação)
  - Ameaças: Usuários negam chamadas; mudanças administrativas sem trilha.
  - Controles: Logging com correlação (x-correlation-id), carimbo de tempo e identidade; retenção de logs; Azure AD audit logs; imutabilidade (Log Analytics, storage com WORM quando necessário).
- I — Information Disclosure (Divulgação)
  - Ameaças: Vazamento em logs/histórico de Logic Apps; APIs retornando dados além do necessário; endpoints públicos inadvertidos.
  - Controles: Data masking/redaction; Secure Inputs/Outputs em Logic Apps; validação de escopos/claims; princípios de menor privilégio; Private Endpoints/VNET; revisão de resposta e filtros de campos.
- D — Denial of Service (Negação de serviço)
  - Ameaças: Flood de requisições ao APIM; loops em orquestração; timeouts em cadeia.
  - Controles: Rate limiting e quotas no APIM; caching seletivo; retry com backoff e limites; circuit breaker; escalonamento automático e proteções de custo; WAF com regras de bot/DDOS Protection.
- E — Elevation of Privilege (Escalada de privilégios)
  - Ameaças: Chaves/senhas expostas; conectores com permissões amplas; admins permanentes.
  - Controles: Managed Identity e Key Vault; escopos mínimos nos conectores; RBAC por função; PIM para acesso just-in-time; varredura de segredos no repositório; revisão periódica de permissões.

Riscos priorizados e tratamento (exemplo rápido):
- Alto: Falha em validate-jwt no APIM → Bloqueio de chamadas sem token válido; testes automatizados de políticas; monitoramento de 401/403.
- Alto: Exposição pública de back-ends → Migrar para Private Endpoints, NSG/Firewall; validação de rota somente via APIM.
- Médio: PII em logs/histórico → Ativar redaction e Secure Inputs/Outputs; revisão de queries de Log Analytics; mascaramento no APIM.
- Médio: DoS por falta de limitação → Rate limit/quotas por subscrição; WAF com regras; circuit breaker e timeouts.
- Baixo: Tokens com vida longa → Reduzir TTL; rotação; monitorar uso anômalo (Identity Protection).

Recomendações de hardening:
- Padronizar políticas APIM (validate-jwt, rate-limit, quota, set-backend-service, mask) via reuso de fragments.
- Exigir Managed Identity em Logic Apps e conectores Azure; segredos no Key Vault.
- Integrar APIM e Logic Apps a VNET e usar Private Link para back-ends.
- Pipeline DevSecOps: IaC com validação (Bicep/Terraform), SAST/DAST, varredura de segredos, aprovações.
- Observabilidade: correlação ponta a ponta, dashboards e alertas (401/403 spikes, 5xx, latência, custo).

Additional resourcing needs:
- Network Administrator: validar VNET, sub-redes, NSG, Private Endpoints, rota só via APIM.
- Security Officer: garantir conformidade com políticas da autoridade; revisão STRIDE e plano de tratamento de risco.
- Cloud/DevOps: definir IaC para APIM/Logic Apps/Key Vault; pipelines; políticas Azure (Policy/Blueprints).
- Software Developer: padrões de API (OpenAPI), contratos, autenticação, versionamento, testes de contrato.
- Hardware Developer: necessário apenas se houver dispositivos/IoT integrando-se às APIs; garantir proteção de credenciais no hardware.
- Identity/IAM Specialist: modelagem de apps no Entra, Conditional Access, PIM, governança de consent.
- Data Protection Officer (DPO): avaliação de PII, minimização de dados, retenção e base legal.
- Penetration Tester/AppSec: testes de APIs (OWASP API Top 10), SOAP/REST, avaliação de políticas do APIM.
- Threat Intelligence/Blue Team: regras no Sentinel, casos de uso de detecção, simulações.
- FinOps: limites e alertas de custo, proteção contra loops e bursts.
- Product Owner/Business Owner: definição de escopo de dados e SLAs; priorização de riscos.
- QA/Testing: testes funcionais e de resiliência (timeouts, retries, circuit breaker).

Observações finais:
- Priorize fechar as fronteiras de rede (Private Link) e a validação de identidade no APIM.
- Documente o modelo de dados e publique OpenAPI versionado no portal.
- Revise a cada mudança material e, no mínimo, trimestralmente, o STRIDE e os controles associados.

Sugestões de melhorias

Segue a entrega em três partes:
- Diagrama Mermaid da arquitetura com os pontos de segurança/correções
- Relatório de Modelagem de Ameaças (STRIDE), consolidado e acionável
- Script Terraform inicial (end-to-end) para provisionamento da solução no Azure

1) Diagrama Mermaid

```mermaid
flowchart LR
  %% Fronteiras e trust boundaries
  subgraph Internet
    C[Clientes (Apps)]:::user
    Dev[Desenvolvedores]:::user
  end

  subgraph Identity[Microsoft Entra ID]
    AAD[Autorização OAuth2/OIDC
    - Conditional Access
    - MFA
    - Admin consent
    - Tokens de curta duração]:::control
  end

  subgraph Edge[Camada de Entrada]
    AFD[Azure Front Door + WAF
    - TLS 1.2+
    - HSTS
    - Regras OWASP/Bot
    - Geo/IP filter
    - mTLS (opcional)]:::control
    APIM[API Management (Gateway + Dev Portal)
    - validate-jwt/escopos
    - rate-limit/quotas
    - validação de schema
    - mascaramento/redaction
    - remoção de headers sensíveis
    - RBAC/Keys por subscrição
    - Diagnostics -> Log Analytics]:::apim
  end

  subgraph VNET[Virtual Network (Subnets: apim, integration, private-endpoints)]
    subgraph Orq[Orquestração]
      LA[Logic Apps Standard
      - Managed Identity
      - Secure Inputs/Outputs
      - Retry+backoff/circuit breaker
      - VNET Integration]:::orq
    end

    subgraph AzureBackends[Back-ends Azure]
      FUNC[Azure Functions]:::svc
      STG[Storage (Private Endpoint, SAS mínimo)]:::svc
      SB[Service Bus (Private Endpoint)]:::svc
      COS[Cosmos DB (Private Endpoint)]:::svc
      KV[Key Vault (MI+RBAC, segredos)]:::control
    end

    DNS[Private DNS Zones]:::net
  end

  subgraph SaaS[SaaS]
    M365[Microsoft 365]:::ext
    SF[Salesforce]:::ext
    SN[ServiceNow]:::ext
  end

  subgraph External[Web Externos]
    REST[REST (TLS 1.2+, pinning)]:::ext
    SOAP[SOAP (XXE desabilitado)]:::ext
  end
```

```
  MON[Observabilidade
  - Log Analytics
  - App Insights
  - Defender for Cloud
  - Sentinel/Alertas]:::control

  %% Fluxos
  C -->|HTTPS TLS1.2+, PKCE, mTLS opcional| AFD -->|WAF/Proteção| APIM
  Dev -->|SSO Entra ID| APIM
  APIM -->|validate-jwt/escopos/claims| AAD
  APIM -->|Políticas (rate limit, schema, mask)| LA
  APIM -.-> MON
  LA -->|Managed Identity + Private DNS| STG
  LA -->|Managed Identity| FUNC
  LA -->|Managed Identity + Private DNS| SB
  LA -->|Managed Identity + Private DNS| COS
  LA -->|OAuth Escopos mínimos| M365
  LA -->|OAuth Escopos mínimos| SF
  LA -->|OAuth Escopos mínimos| SN
  LA -->|TLS 1.2+, pinning e timeouts| REST
  LA -->|TLS 1.2+, XXE off, schema validation| SOAP
  KV <--> LA
  DNS --- LA
  STG --- DNS
  SB --- DNS
  COS --- DNS
  APIM --- DNS
  MON --- APIM
  MON --- LA

  classDef user fill:#e8f3ff,stroke:#6aa9ff;
  classDef apim fill:#fff4e6,stroke:#ff8c1a;
  classDef orq fill:#f0fff4,stroke:#19a974;
  classDef svc fill:#f5f5f5,stroke:#999;
  classDef ext fill:#f7f2ff,stroke:#9a7cec;
  classDef control fill:#eefaf7,stroke:#3fb589;
  classDef net fill:#eef2f7,stroke:#7d8ca0;
```

Notas de segurança incorporadas no desenho:
- Borda pública protegida por Azure Front Door Standard/Premium com WAF (TLS 1.2+, HSTS, regras OWASP, Bot protection, Geo/IP; mTLS opcional).
- APIM com políticas padrão reutilizáveis: validate-jwt, validate-content, rate-limit/quotas, remoção de headers sensíveis e masking/redaction; Diagnostics para Log Analytics; gateway integrado à VNET (subnet dedicada).
- Logic Apps Standard com Managed Identity, Secure Inputs/Outputs, VNET Integration e padrões de resiliência (retry/backoff/circuit breaker, idempotência).
- Back-ends Azure expostos via Private Endpoints + Private DNS Zones; Key Vault para segredos/identidades; RBAC mínimo necessário.
- Observabilidade centralizada (Log Analytics, App Insights), alertas e detecção (Defender for Cloud, Sentinel).

2) Relatório de Modelagem de Ameaças (STRIDE)

Escopo e fronteiras de confiança
- Fronteiras:
  - Internet → Front Door/WAF → APIM (borda pública)
  - APIM → Logic Apps (orquestração em VNET)
  - Logic Apps → Back-ends (Azure via Private Endpoints, SaaS via OAuth, externos REST/SOAP com TLS)
  - Área administrativa (Dev Portal, Azure Portal, pipelines CI/CD)
- Ativos:
  - Endpoints de API, definições OpenAPI
  - Tokens OAuth/JWT, segredos (Key Vault), Managed Identities
  - Dados de cliente/PII, payloads de integração
  - Configurações/políticas do APIM e Logic Apps
  - Logs/telemetria, dashboards/alertas

S — Spoofing
- Ameaças: uso de tokens roubados; cliente sem mTLS; impersonação de serviços externos.
- Controles:
  - OAuth2/OIDC com validate-jwt no APIM (issuer/audience/escopos/claims)
  - PKCE para SPAs/mobile; Conditional Access e MFA no Entra ID
  - mTLS opcional no APIM/Front Door; certificate pinning para saídas
  - Managed Identity para chamadas a serviços Azure

T — Tampering
- Ameaças: manipulação de requests/headers; alteração indevida de políticas APIM; XXE/injeção em SOAP.
- Controles:
  - TLS fim a fim e HSTS; validação de conteúdo/schema no APIM
  - CI/CD com IaC (Terraform), revisões e approvals; RBAC/PIM
  - Desabilitar XXE; assinatura de mensagens onde aplicável

R — Repudiation
- Ameaças: negação de chamadas; mudanças sem trilha.
- Controles:
  - Logging com correlação (x-correlation-id), timestamps e identidade no APIM/Logic Apps
  - Retenção adequada e imutabilidade onde necessário
  - Azure AD audit logs; Activity Logs e Diagnostics centralizados

I — Information Disclosure
- Ameaças: vazamento em logs/históricos; APIs com over-sharing; endpoints públicos inadvertidos.
- Controles:
  - Redaction/masking no APIM; Secure Inputs/Outputs em Logic Apps
  - Menor privilégio e escopos mínimos; revisão de contratos de API
  - Private Endpoints/VNET; revisão periódica de visibilidade do Dev Portal

D — Denial of Service
- Ameaças: flood no gateway; loops de orquestração; timeouts em cadeia.
- Controles:
  - Rate limit/quotas/caching seletivo no APIM
  - Retry com backoff/jitter, circuit breaker e timeouts; limites de custo
  - WAF com regras anti-bot e Azure DDoS Protection (se necessário); autoscaling

E — Elevation of Privilege
- Ameaças: segredos expostos; conectores excessivos; admins permanentes.
- Controles:
  - Managed Identity + Key Vault; varredura de segredos
  - Escopos mínimos em conectores; RBAC por função; PIM para JIT
  - Revisões periódicas de permissões e consent

Riscos priorizados e tratamento
- Alto: Falha em validate-jwt no APIM → Bloquear chamadas sem token válido; testes de políticas; monitorar 401/403.
- Alto: Back-ends expostos publicamente → Migrar para Private Endpoints + Private DNS; permitir tráfego só via VNET/APIM.
- Médio: PII em logs/histórico → Redaction e Secure Inputs/Outputs; revisar consultas/retentiva.
- Médio: DoS sem limitação → Rate limit/quotas por subscrição; WAF; circuit breaker/timeouts no fluxo.
- Baixo: Tokens longos → Reduzir TTL/rotação; monitorar anomalias com Identity Protection.

Recomendações de hardening
- Padronizar fragments de políticas no APIM (validate-jwt, rate-limit/quota, set-backend-service, masking).
- Exigir Managed Identity em Logic Apps e conectores; segredos apenas no Key Vault.
- Integrar APIM e Logic Apps à VNET e usar Private Link para back-ends; Private DNS Zones.
- DevSecOps: IaC (Terraform) com validações, SAST/DAST, varredura de segredos, aprovações obrigatórias.
- Observabilidade: correlação ponta a ponta; dashboards e alertas (401/403 spikes, 5xx, latência e custo).

3) Script Terraform (Azure)
Observações:
- Exemplo opinativo, pronto para subir um esqueleto funcional com: RG, VNET/subnets, APIM (Developer SKU) com identidade gerenciada, Front Door Standard + WAF, Logic Apps Standard com integração a VNET, Key Vault, Storage/Service Bus/Cosmos DB com Private Endpoints, Log

Analytics + App Insights, Azure AD App Registration para OIDC.
- Personalize nomes, tags, SKU/custos e domínios.
- Requer providers: azurerm >= 3.x e azuread >= 2.x.

main.tf
```hcl
terraform {
  required_version = ">= 1.5.0"
  required_providers {
    azurerm = {
      source  = "hashicorp/azurerm"
      version = ">= 3.106.0"
    }
    azuread = {
      source  = "hashicorp/azuread"
      version = ">= 2.48.0"
    }
    random = {
      source  = "hashicorp/random"
      version = ">= 3.5.0"
    }
  }
}

provider "azurerm" {
  features {}
}

provider "azuread" {}

locals {
  name_prefix = var.name_prefix
  location    = var.location
  tags = merge(
    {
      environment = var.environment
      owner       = var.owner
      costCenter  = var.cost_center
      workload    = "api-integration-hub"
    },
    var.tags
  )
}

resource "azurerm_resource_group" "rg" {
  name     = "${local.name_prefix}-rg"
  location = local.location
  tags     = local.tags
}

# Log Analytics + Application Insights (workspace-based)
resource "azurerm_log_analytics_workspace" "law" {
  name                = "${local.name_prefix}-law"
  location            = azurerm_resource_group.rg.location
  resource_group_name = azurerm_resource_group.rg.name
  sku                 = "PerGB2018"
  retention_in_days   = 30
  tags                = local.tags
}

resource "azurerm_application_insights" "appi" {
  name                = "${local.name_prefix}-appi"
  location            = azurerm_resource_group.rg.location
  resource_group_name = azurerm_resource_group.rg.name
  workspace_id        = azurerm_log_analytics_workspace.law.id
  application_type    = "web"
  tags                = local.tags
}
```

```
# Virtual Network e Subnets
resource "azurerm_virtual_network" "vnet" {
  name                = "${local.name_prefix}-vnet"
  location            = azurerm_resource_group.rg.location
  resource_group_name = azurerm_resource_group.rg.name
  address_space       = ["10.20.0.0/16"]
  tags                = local.tags
}

resource "azurerm_subnet" "snet_apim" {
  name                 = "${local.name_prefix}-snet-apim"
  resource_group_name  = azurerm_resource_group.rg.name
  virtual_network_name = azurerm_virtual_network.vnet.name
  address_prefixes     = ["10.20.1.0/24"]
  delegations {
    name = "apim-delegation"
    service_delegation {
      name = "Microsoft.ApiManagement/service"
    }
  }
}

resource "azurerm_subnet" "snet_integration" {
  name                 = "${local.name_prefix}-snet-int"
  resource_group_name  = azurerm_resource_group.rg.name
  virtual_network_name = azurerm_virtual_network.vnet.name
  address_prefixes     = ["10.20.2.0/24"]
  service_endpoints    = ["Microsoft.Storage"]
}

resource "azurerm_subnet" "snet_priv_endpoints" {
  name                 = "${local.name_prefix}-snet-pe"
  resource_group_name  = azurerm_resource_group.rg.name
  virtual_network_name = azurerm_virtual_network.vnet.name
  address_prefixes     = ["10.20.10.0/24"]
  private_endpoint_network_policies_enabled = false
}

# Private DNS Zones e links (Storage, Service Bus, Cosmos)
resource "azurerm_private_dns_zone" "pdz_blob" {
  name                = "privatelink.blob.core.windows.net"
  resource_group_name = azurerm_resource_group.rg.name
  tags                = local.tags
}
resource "azurerm_private_dns_zone" "pdz_sb" {
  name                = "privatelink.servicebus.windows.net"
  resource_group_name = azurerm_resource_group.rg.name
  tags                = local.tags
}
resource "azurerm_private_dns_zone" "pdz_cosmos" {
  name                = "privatelink.documents.azure.com"
  resource_group_name = azurerm_resource_group.rg.name
  tags                = local.tags
}

resource "azurerm_private_dns_zone_virtual_network_link" "link_blob" {
  name                  = "${local.name_prefix}-blob-link"
  resource_group_name   = azurerm_resource_group.rg.name
  private_dns_zone_name = azurerm_private_dns_zone.pdz_blob.name
  virtual_network_id    = azurerm_virtual_network.vnet.id
  registration_enabled  = false
}

resource "azurerm_private_dns_zone_virtual_network_link" "link_sb" {
  name                  = "${local.name_prefix}-sb-link"
  resource_group_name   = azurerm_resource_group.rg.name
  private_dns_zone_name = azurerm_private_dns_zone.pdz_sb.name
  virtual_network_id    = azurerm_virtual_network.vnet.id
  registration_enabled  = false
```

```
}

resource "azurerm_private_dns_zone_virtual_network_link" "link_cosmos" {
  name                  = "${local.name_prefix}-cosmos-link"
  resource_group_name   = azurerm_resource_group.rg.name
  private_dns_zone_name = azurerm_private_dns_zone.pdz_cosmos.name
  virtual_network_id    = azurerm_virtual_network.vnet.id
  registration_enabled  = false
}

# Key Vault
resource "azurerm_key_vault" "kv" {
  name                       = "${replace(local.name_prefix, "-", "")}kv"
  location                   = azurerm_resource_group.rg.location
  resource_group_name        = azurerm_resource_group.rg.name
  tenant_id                  = data.azuread_client_config.current.tenant_id
  sku_name                   = "standard"
  purge_protection_enabled   = true
  soft_delete_retention_days = 14
  enable_rbac_authorization  = true
  tags                       = local.tags
}

data "azuread_client_config" "current" {}

# Azure AD App Registration (OIDC para validar JWT no APIM)
resource "azuread_application" "api_oidc" {
  display_name = "${local.name_prefix}-api"
  sign_in_audience = "AzureADMyOrg"

  api {
    requested_access_token_version = 2
    oauth2_permission_scope {
      admin_consent_description = "Permite acesso às APIs do gateway"
      admin_consent_display_name = "Acessar APIs"
      enabled = true
      id = uuidv5("dns", "scope-access-${local.name_prefix}")
      type = "User"
      value = "api.access"
    }
  }
}

resource "azuread_service_principal" "api_oidc_sp" {
  application_id = azuread_application.api_oidc.application_id
}

# APIM (Developer SKU) com VNET External e Identity
resource "azurerm_api_management" "apim" {
  name                = "${local.name_prefix}-apim"
  location            = azurerm_resource_group.rg.location
  resource_group_name = azurerm_resource_group.rg.name
  publisher_name      = var.publisher_name
  publisher_email     = var.publisher_email
  sku_name            = "Developer_1"

  identity {
    type = "SystemAssigned"
  }

  virtual_network_type = "External"
  virtual_network_configuration {
    subnet_id = azurerm_subnet.snet_apim.id
  }

  protocols {
    enable_http2 = true
  }
```

```
  policy {
    xml_content = <<POLICY
<policies>
  <inbound>
    <base />
    <!-- Forçar correlação -->
    <set-header name="x-correlation-id" exists-action="override">
      <value>@(Guid.NewGuid().ToString())</value>
    </set-header>

    <!-- HSTS na resposta será aplicado em outbound -->
    <!-- Validação de JWT (OIDC) -->
    <validate-jwt header-name="Authorization" failed-validation-httpcode="401"
require-scheme="Bearer">
      <openid-config
url="https://login.microsoftonline.com/${data.azuread_client_config.current.tenant_id}/v2.0/.well-kno
wn/openid-configuration" />
      <audiences>
        <audience>${azuread_application.api_oidc.application_id}</audience>
      </audiences>
      <required-claims>
        <claim name="scp">
          <value>api.access</value>
        </claim>
      </required-claims>
    </validate-jwt>

    <!-- Rate limiting (exemplo global; refine por produto/subscrição) -->
    <rate-limit calls="100" renewal-period="60" />

    <!-- Remover headers sensíveis de entrada -->
    <set-header name="x-forwarded-for" exists-action="delete" />
    <set-header name="x-azure-fdid" exists-action="delete" />
  </inbound>

  <backend>
    <base />
    <!-- Retry com backoff -->
    <retry condition="@(context.Response == null || (context.Response.StatusCode >= 500 &&
context.Response.StatusCode != 501 && context.Response.StatusCode != 505))"
           count="3"
           interval="1"
           max-interval="5"
           delta="1" />
  </backend>

  <outbound>
    <base />
    <!-- HSTS -->
    <set-header name="Strict-Transport-Security" exists-action="override">
      <value>max-age=63072000; includeSubDomains; preload</value>
    </set-header>
    <!-- Redaction de headers -->
    <set-header name="Set-Cookie" exists-action="delete" />
  </outbound>

  <on-error>
    <base />
    <return-response>
      <set-status code="502" reason="Bad Gateway" />
      <set-header name="Content-Type" exists-action="override">
        <value>application/json</value>
      </set-header>
      <set-body>{
        "error":"Upstream error",
        "correlationId":"@(context.Request.Headers.GetValueOrDefault("x-correlation-id",""))"
      }</set-body>
    </return-response>
  </on-error>
```

```
</policies>
POLICY
  }

  tags = local.tags
}

# Logger do APIM para App Insights
resource "azurerm_api_management_logger" "apim_ai_logger" {
  name                = "appinsights"
  api_management_name = azurerm_api_management.apim.name
  resource_group_name = azurerm_resource_group.rg.name

  application_insights {
    instrumentation_key = azurerm_application_insights.appi.instrumentation_key
  }
}

# Diagnostics do APIM para Log Analytics
resource "azurerm_monitor_diagnostic_setting" "diag_apim" {
  name                       = "${local.name_prefix}-apim-diag"
  target_resource_id         = azurerm_api_management.apim.id
  log_analytics_workspace_id = azurerm_log_analytics_workspace.law.id

  enabled_log {
    category = "GatewayLogs"
  }
  enabled_log {
    category = "WebSocketConnectionLogs"
  }
  enabled_log {
    category = "AuditLogs"
  }
  metric {
    category = "AllMetrics"
  }
}

# App Service Plan para Logic Apps Standard
resource "azurerm_service_plan" "asp_la" {
  name                = "${local.name_prefix}-asp-la"
  location            = azurerm_resource_group.rg.location
  resource_group_name = azurerm_resource_group.rg.name
  os_type             = "Windows"
  sku_name            = "WS1" # Ajuste conforme necessidade/custos
  tags                = local.tags
}

# Storage para Logic Apps Standard (state/workflows)
resource "azurerm_storage_account" "st_la" {
  name                     = lower(replace("${local.name_prefix}stla", "-", ""))
  resource_group_name      = azurerm_resource_group.rg.name
  location                 = azurerm_resource_group.rg.location
  account_tier             = "Standard"
  account_replication_type = "LRS"

  min_tls_version          = "TLS1_2"
  allow_nested_items_to_be_public = false

  tags = local.tags
}

# Logic Apps Standard com Managed Identity e VNET Integration
resource "azurerm_logic_app_standard" "la" {
  name                       = "${local.name_prefix}-la"
  location                   = azurerm_resource_group.rg.location
  resource_group_name        = azurerm_resource_group.rg.name
  app_service_plan_id        = azurerm_service_plan.asp_la.id
  storage_account_name       = azurerm_storage_account.st_la.name
```

```
  storage_account_access_key = azurerm_storage_account.st_la.primary_access_key
  virtual_network_subnet_id  = azurerm_subnet.snet_integration.id
  https_only                 = true

  identity {
    type = "SystemAssigned"
  }

  app_settings = {
    WEBSITE_RUN_FROM_PACKAGE = "0"
    APPINSIGHTS_INSTRUMENTATIONKEY =
azurerm_application_insights.appi.instrumentation_key
    WORKFLOWS_TENANT_ID = data.azuread_client_config.current.tenant_id
  }

  tags = local.tags
}

# Key Vault access (RBAC habilitado - conceder roles)
data "azurerm_subscription" "current" {}

resource "azurerm_role_assignment" "kv_la_secrets_user" {
  scope              = azurerm_key_vault.kv.id
  role_definition_name = "Key Vault Secrets User"
  principal_id       = azurerm_logic_app_standard.la.identity[0].principal_id
}

resource "azurerm_key_vault_secret" "example_api_secret" {
  name         = "SampleSecret"
  value        = random_password.sample_secret.result
  key_vault_id = azurerm_key_vault.kv.id
}

resource "random_password" "sample_secret" {
  length  = 32
  special = true
}

# Back-ends Azure com Private Endpoints
resource "azurerm_storage_account" "stg" {
  name                     = lower(replace("${local.name_prefix}stg", "-", ""))
  resource_group_name      = azurerm_resource_group.rg.name
  location                 = azurerm_resource_group.rg.location
  account_tier             = "Standard"
  account_replication_type = "LRS"
  min_tls_version          = "TLS1_2"
  allow_blob_public_access = false
  blob_properties {
    delete_retention_policy {
      days = 7
    }
  }
  network_rules {
    default_action             = "Deny"
    bypass                     = ["AzureServices"]
    ip_rules                   = []
    virtual_network_subnet_ids = []
  }
  tags = local.tags
}

resource "azurerm_private_endpoint" "pe_stg_blob" {
  name                = "${local.name_prefix}-pe-stg-blob"
  location            = azurerm_resource_group.rg.location
  resource_group_name = azurerm_resource_group.rg.name
  subnet_id           = azurerm_subnet.snet_priv_endpoints.id

  private_service_connection {
    name                           = "blob"
```

```
    private_connection_resource_id = azurerm_storage_account.stg.id
    subresource_names              = ["blob"]
    is_manual_connection           = false
  }

  private_dns_zone_group {
    name                 = "blob-zone"
    private_dns_zone_ids = [azurerm_private_dns_zone.pdz_blob.id]
  }

  tags = local.tags
}

resource "azurerm_servicebus_namespace" "sb" {
  name                = "${local.name_prefix}-sb"
  location            = azurerm_resource_group.rg.location
  resource_group_name = azurerm_resource_group.rg.name
  sku                 = "Standard"
  minimum_tls_version = "1.2"

  network_rules {
    default_action  = "Deny"
    trusted_services_allowed = true
  }

  tags = local.tags
}

resource "azurerm_private_endpoint" "pe_sb" {
  name                = "${local.name_prefix}-pe-sb"
  location            = azurerm_resource_group.rg.location
  resource_group_name = azurerm_resource_group.rg.name
  subnet_id           = azurerm_subnet.snet_priv_endpoints.id

  private_service_connection {
    name                           = "sb-conn"
    private_connection_resource_id = azurerm_servicebus_namespace.sb.id
    subresource_names              = ["namespace"]
    is_manual_connection           = false
  }

  private_dns_zone_group {
    name                 = "sb-zone"
    private_dns_zone_ids = [azurerm_private_dns_zone.pdz_sb.id]
  }

  tags = local.tags
}

resource "azurerm_cosmosdb_account" "cosmos" {
  name                = "${local.name_prefix}-cosmos"
  location            = azurerm_resource_group.rg.location
  resource_group_name = azurerm_resource_group.rg.name
  offer_type          = "Standard"
  kind                = "GlobalDocumentDB"

  consistency_policy {
    consistency_level = "Session"
  }

  enable_free_tier = true
  enable_automatic_failover = false

  capabilities {
    name = "EnableServerless"
  }

  is_virtual_network_filter_enabled = true
  public_network_access_enabled     = false
```

```
  tags = local.tags
}

resource "azurerm_private_endpoint" "pe_cosmos" {
  name                = "${local.name_prefix}-pe-cosmos"
  location            = azurerm_resource_group.rg.location
  resource_group_name = azurerm_resource_group.rg.name
  subnet_id           = azurerm_subnet.snet_priv_endpoints.id

  private_service_connection {
    name                           = "cosmos-conn"
    private_connection_resource_id = azurerm_cosmosdb_account.cosmos.id
    subresource_names              = ["Sql"]
    is_manual_connection           = false
  }

  private_dns_zone_group {
    name                 = "cosmos-zone"
    private_dns_zone_ids = [azurerm_private_dns_zone.pdz_cosmos.id]
  }

  tags = local.tags
}

# Front Door Standard/Premium com WAF apontando para o APIM público
resource "azurerm_cdn_frontdoor_profile" "fd" {
  name                = "${local.name_prefix}-fd"
  resource_group_name = azurerm_resource_group.rg.name
  sku_name            = "Standard_AzureFrontDoor"
  tags                = local.tags
}

resource "azurerm_cdn_frontdoor_endpoint" "fd_ep" {
  name                     = "${local.name_prefix}-fde"
  cdn_frontdoor_profile_id = azurerm_cdn_frontdoor_profile.fd.id
  tags                     = local.tags
}

# Política WAF (OWASP + bot), personalize regras conforme necessidade
resource "azurerm_cdn_frontdoor_firewall_policy" "fd_waf" {
  name                = "${local.name_prefix}-waf"
  resource_group_name = azurerm_resource_group.rg.name
  sku_name            = azurerm_cdn_frontdoor_profile.fd.sku_name
  mode                = "Prevention"

  managed_rule {
    type    = "DefaultRuleSet"
    version = "2.1"
  }

  managed_rule {
    type    = "BotProtection"
    version = "1.0"
  }

  tags = local.tags
}

# Origin Group
resource "azurerm_cdn_frontdoor_origin_group" "fd_og" {
  name                     = "${local.name_prefix}-og"
  cdn_frontdoor_profile_id = azurerm_cdn_frontdoor_profile.fd.id

  load_balancing {}
  health_probe {
    protocol = "Https"
    path     = "/status-0123456789abcdef" # Ajuste para health endpoint do APIM (ou API de status)
    interval_in_seconds = 60
```

```
  }

  tags = local.tags
}

# Origin: APIM hostname público (o Front Door fará proxy com Host header do APIM)
resource "azurerm_cdn_frontdoor_origin" "fd_origin_apim" {
  name                          = "apim-origin"
  cdn_frontdoor_origin_group_id = azurerm_cdn_frontdoor_origin_group.fd_og.id

  enabled             = true
  host_name           = azurerm_api_management.apim.gateway_url # ex: https://{name}.azure-api.net
  http_port           = 80
  https_port          = 443
  origin_host_header  = replace(azurerm_api_management.apim.gateway_url, "https://", "")
  priority            = 1
  weight              = 1000

  certificate_name_check_enabled = true
}

# Rota Front Door + associação WAF
resource "azurerm_cdn_frontdoor_route" "fd_route" {
  name                          = "${local.name_prefix}-route"
  cdn_frontdoor_endpoint_id     = azurerm_cdn_frontdoor_endpoint.fd_ep.id
  cdn_frontdoor_origin_group_id = azurerm_cdn_frontdoor_origin_group.fd_og.id
  cdn_frontdoor_origin_ids      = [azurerm_cdn_frontdoor_origin.fd_origin_apim.id]
  supported_protocols           = ["Https"]
  https_redirect_enabled        = true
  forwarding_protocol           = "HttpsOnly"
  patterns_to_match             = ["/*"]
  link_to_default_domain        = true
  cdn_frontdoor_custom_domain_ids = []

  cache {
    query_string_caching_behavior = "IgnoreQueryString"
  }

  rule_sets = []
}

resource "azurerm_cdn_frontdoor_endpoint_waf_association" "fd_waf_assoc" {
  cdn_frontdoor_endpoint_id = azurerm_cdn_frontdoor_endpoint.fd_ep.id
  firewall_policy_id        = azurerm_cdn_frontdoor_firewall_policy.fd_waf.id
}

# Diagnostics diversos
resource "azurerm_monitor_diagnostic_setting" "diag_la" {
  name                       = "${local.name_prefix}-la-diag"
  target_resource_id         = azurerm_logic_app_standard.la.id
  log_analytics_workspace_id = azurerm_log_analytics_workspace.law.id

  enabled_log { category = "WorkflowRuntime" }
  metric { category = "AllMetrics" }
}

resource "azurerm_monitor_diagnostic_setting" "diag_kv" {
  name                       = "${local.name_prefix}-kv-diag"
  target_resource_id         = azurerm_key_vault.kv.id
  log_analytics_workspace_id = azurerm_log_analytics_workspace.law.id

  enabled_log { category = "AuditEvent" }
  metric { category = "AllMetrics" }
}

# Saídas
output "resource_group" {
  value = azurerm_resource_group.rg.name
}
```

```
output "apim_gateway_url" {
  value = azurerm_api_management.apim.gateway_url
}

output "front_door_endpoint" {
  value = azurerm_cdn_frontdoor_endpoint.fd_ep.host_name
}

output "logic_app_standard_url_hint" {
  value =
"https://${azurerm_logic_app_standard.la.default_hostname}/workflows/{workflowName}/triggers/{tr
iggerName}/invoke?api-version=2019-05-01&sig={key}"
}
```

variables.tf
```hcl
variable "name_prefix" {
  description = "Prefixo curto para nomear recursos (ex: acme-api)"
  type        = string
}

variable "location" {
  description = "Região do Azure (ex: brazilsouth, eastus)"
  type        = string
  default     = "eastus"
}

variable "publisher_name" {
  description = "Nome do publisher (APIM)"
  type        = string
}

variable "publisher_email" {
  description = "Email do publisher (APIM)"
  type        = string
}

variable "environment" {
  description = "Ambiente (dev, qa, prod)"
  type        = string
  default     = "dev"
}

variable "owner" {
  description = "Owner do workload"
  type        = string
  default     = "team-platform"
}

variable "cost_center" {
  description = "Centro de custo"
  type        = string
  default     = "0000"
}

variable "tags" {
  description = "Tags adicionais"
  type        = map(string)
  default     = {}
}
```

Como continuar após o deploy
- Configure as APIs no APIM importando seus contratos OpenAPI e anexe fragments de políticas padronizadas (validate-jwt, rate-limit/quotas, validate-content, set-backend-service, masking/redaction).
- Publique o Developer Portal com SSO via Entra ID e revisão de conteúdo. O “Try it” pode usar o

Authorization Server no APIM apontando para o App Registration criado.
- Crie os workflows no Logic Apps Standard (com MI e Secure Inputs/Outputs) e conectores com escopos mínimos; acesse segredos no Key Vault.
- Complete a privatização dos back-ends restantes (Functions, etc.) e teste a resolução via Private DNS.
- Ative alertas no Log Analytics/App Insights (spikes 401/403, 5xx, latência, custo).
- Opcional: habilite Azure DDoS Protection (para workloads em VNET que exponham IP público) e bloqueios de recurso (resource locks) em RGs críticos.

Limitações/ajustes esperados
- O exemplo de rota Front Door usa o hostname público do APIM; para máxima segurança, considere FD Premium com Private Link para APIM em modo Internal ou APIM + App Gateway WAF com VNET restrita.
- Logic Apps Standard foi provisionado sem um workflow exemplo (implante seus workflows via CI/CD).
- Políticas APIM devem ser refinadas por produto/API/rota e configuradas como fragments reusáveis no seu repositório.