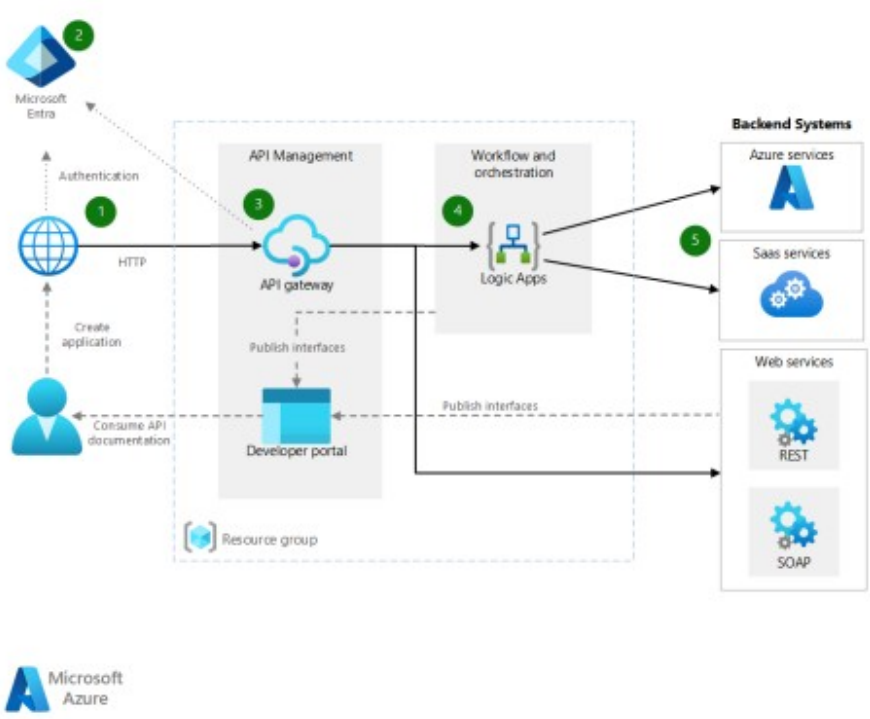
Relatório de análise da solução: tmp5blgp99k



Análise completa da solução atual

Modelo de cloud:
- Microsoft Azure

Lista com os componentes:
- Consumidor/Desenvolvedor de APIs (cliente externo)
- Microsoft Entra ID (antigo Azure AD) para autenticação/autorização
- Azure API Management (APIM)
  - API Gateway
  - Developer Portal (portal do desenvolvedor)
  - Políticas de API (rate limit, validação de JWT, transformação)
- Azure Logic Apps (Workflow and Orchestration)
- Sistemas de Backend
  - Azure services (ex.: Functions, Storage, Service Bus) [genérico no diagrama]
  - SaaS services (ex.: Salesforce, O365, etc.) [genérico no diagrama]
  - Web services REST
  - Web services SOAP
- Resource Group (limite de implantação)
- Fluxos/Conexões destacados no desenho
  - (1) Chamadas HTTP do cliente para o APIM
  - (2) Autenticação via Microsoft Entra
  - (3) Gateway do APIM publica/expõe interfaces e aplica políticas
  - (4) Lógica/orquestração via Logic Apps
  - (5) Chamadas do Logic Apps para backends (Azure, SaaS, REST, SOAP)

Interação entre os componentes:
- 1. O cliente registra a aplicação e obtém credenciais no Microsoft Entra; navega no Developer Portal do APIM para descobrir e subscrever APIs.
- 2. O cliente obtém um token (OAuth2/OIDC) do Microsoft Entra e o anexa às chamadas.
- 3. As requisições HTTP chegam ao API Gateway (APIM), onde políticas validam o JWT, aplicam rate limiting, CORS, transformação e roteamento.
- 4. O APIM encaminha a chamada para um workflow do Logic Apps para orquestração, integração, transformação de dados e automação de processos.
- 5. O Logic Apps chama os sistemas de backend (serviços Azure, SaaS e web services REST/SOAP), agrega respostas e retorna ao APIM, que responde ao cliente.
- Observação: O Developer Portal é usado para documentação, testes e gestão de chaves/subscrições. O Resource Group delimita os recursos do Azure envolvidos.

O que esse sistema faz:
- Fornece uma camada de gestão de APIs (façade) para expor serviços internos/externos com segurança e governança.
- Orquestra processos de negócio via Logic Apps, integrando múltiplos backends (Azure, SaaS, REST/SOAP).
- Centraliza autenticação/autorização via Microsoft Entra e simplifica o onboarding de desenvolvedores pelo Developer Portal.

Vulnerabilidades e Solução para cada vulnerabilidade:
- Autenticação fraca ou configuração incorreta (tokens sem escopos/roles adequados)
  - Solução: Aplicativos registrados no Entra com OAuth2/OIDC, escopos/roles obrigatórios, validação de JWT no APIM (issuer, audience, assinatura), uso de PKCE para apps públicos.
- Exposição de APIs a partir de endpoints públicos
  - Solução: APIM com VNet integration e Private Endpoints; Application Gateway/WAF na borda; IP allowlist; CORS restritivo; mTLS quando aplicável.
- Vazamento de segredos (chaves de subscrição, webhooks, conexões do Logic Apps)
  - Solução: Managed Identities para APIM/Logic Apps; segredos em Azure Key Vault com rotação; mascaramento de segredos em logs; desabilitar exibição de valores sensíveis.
- Falta de limitação de tráfego e proteção contra DoS
  - Solução: Políticas de rate limit/quotas no APIM; Azure DDoS Protection Standard; autoscaling de APIM/Logic Apps; caching onde aplicável; circuit breaker no Logic Apps.
- Validação insuficiente de entrada/payload
  - Solução: Políticas APIM de validação de esquema (OpenAPI), tamanho de payload, content-type; sanitização; antimalware em pontos de ingestão quando cabível.
- Exfiltração de dados via conectores do Logic Apps
  - Solução: Conectores aprovados com DLP; Private Link para serviços Azure/SaaS suportados; inspeção via Firewall; regras de saída restritivas.
- Logs com dados sensíveis e ausência de trilha de auditoria
  - Solução: Diagnósticos do APIM/Logic Apps no Log Analytics; mascarar PII; retenção e imutabilidade (Azure Storage com políticas WORM se necessário); sincronização de tempo.
- Escalonamento de privilégios por identidades gerenciadas superpermissivas
  - Solução: RBAC mínimo necessário; PIM (Privileged Identity Management); separação de funções; revisões periódicas de acesso; políticas de acesso condicional.
- Conexões inseguras a web services legados (SOAP/REST sem TLS forte)
  - Solução: TLS 1.2+ obrigatório; pinagem/cadeia de certificados confiável; rejeitar cifradores fracos; mensagens assinadas para webhooks.
- Riscos no Developer Portal (cadastro aberto, chaves expostas em exemplos)
  - Solução: Aprovação manual de subscrições; anonimização/masking de chaves nos exemplos; rate limit no portal de teste; CAPTCHA e políticas antiabuso.
- Integridade de mensagens e transformação incorreta
  - Solução: Políticas APIM de transformação com testes/CI; versionamento de APIs; contratos e testes de contrato; assinatura/verificação quando necessário.
- Resiliência e continuidade (falha regional, dependências externas)
  - Solução: APIM Premium multi-região, Logic Apps em zonas/regiões emparelhadas; replays idempotentes; DLQ (Service Bus) para integrações assíncronas; backups e DR.

Gere um Relatório de Modelagem de Ameaças, baseado na metodologia STRIDE:
- Escopo e fronteiras de confiança
  - Fronteira 1: Internet/Cliente → APIM (exposto publicamente ou via WAF).
  - Fronteira 2: APIM → Logic Apps (serviços gerenciados no Azure).
  - Fronteira 3: Logic Apps → Backends (Azure, SaaS, Web REST/SOAP; alguns externos à sua autoridade).
  - Armazenamentos implícitos: históricos de execução do Logic Apps, logs do APIM/Monitor.
- Ativos principais
  - Credenciais/tokens do Entra; chaves de subscrição; payloads de APIs; configurações de políticas APIM; conexões/segredos do Logic Apps; dados de negócio processados.
- STRIDE por fluxo e componente
  - S (Spoofing)
    - Risco: Falsificação de cliente ou serviço; tokens roubados; endpoints de backend falsos.
    - Mitigações: OAuth2/OIDC com validação de JWT no APIM; mTLS quando apropriado; Managed Identity em chamadas internas; pinagem/validação de certificado para saídas; Conditional Access.
  - T (Tampering)
    - Risco: Alteração de requisições/respostas; manipulação de políticas APIM; alteração de workflows.
    - Mitigações: TLS 1.2+; controles de mudança e CI/CD com validações; Azure Policy/Blueprints; RBAC mínimo; versionamento e aprovações de PR; assinaturas de mensagem para webhooks críticos.
  - R (Repudiation)
    - Risco: Atores negam ações sem trilha adequada.
    - Mitigações: Logging detalhado no APIM/Logic Apps com correlação (trace-id); retenção imutável;

relógios sincronizados; integração com SIEM (Microsoft Sentinel).
  - I (Information Disclosure)
    - Risco: Vazamento em trânsito/logs/históricos; exposição de CORS amplo; saída para SaaS fora de compliance.
    - Mitigações: TLS forte; mascaramento/redação em logs; DLP e classificação; CORS restrito; Private Link; criptografia de dados at-rest (chaves gerenciadas pelo cliente se necessário).
  - D (Denial of Service)
    - Risco: Saturação do APIM/Logic Apps; backends indisponíveis causando cascata.
    - Mitigações: Rate limit/quotas e burst control; DDoS Standard; cache; filas/bulkhead; timeouts e retries exponenciais; circuit breaker.
  - E (Elevation of Privilege)
    - Risco: Identidades gerenciadas com privilégios excessivos; abuso do portal do desenvolvedor; manipulação de políticas para ganhar acesso.
    - Mitigações: RBAC mínimo e PIM; separação de ambientes (dev/test/prod) e de funções; aprovação manual de subscrições; revisões de políticas APIM; varreduras de postura (Defender for Cloud).
- Suposições e dependências
  - O Microsoft Entra é a autoridade de identidade; TLS é obrigatório; conectividade a SaaS depende de conectores/credenciais aprovadas.
- Riscos residuais
  - Dependência de SaaS/terceiros; complexidade de orquestração; dados sensíveis em históricos do Logic Apps exigem governança estricta.
- Plano de tratamento (alto nível)
  - Curto prazo: habilitar validação de JWT e rate limiting no APIM; mover segredos para Key Vault; habilitar logs e mascaramento.
  - Médio prazo: VNet/Private Endpoints; WAF e DDoS; Managed Identities em todos os fluxos; CI/CD com testes de contrato.
  - Longo prazo: Multi-região para APIM/Logic Apps; SIEM com detecções customizadas; governança de dados e DLP; revisões periódicas de acesso e pen tests.

Additional resourcing needs:
- Network Administrator
  - Apoiar VNet integration, Private Endpoints, regras de NSG/Firewall, conectividade híbrida e DDoS/WAF.
- Security Officer
  - Validar aderência a padrões da autoridade; aprovar política de CORS, criptografia, DLP, retenção de logs e classificação de dados.
- Cloud/DevOps staff
  - Infra as Code (Bicep/Terraform) para APIM/Logic Apps/Networking; pipelines CI/CD com validações e promoção entre ambientes; integração com Key Vault.
- Software Developer
  - Definir contratos OpenAPI, versionamento, testes de contrato, políticas do APIM (transformação/validação); clientes com OAuth2/PKCE e práticas de segurança.
- Hardware Developer (se houver dispositivos/IoT clientes)
  - Proteção de credenciais em hardware; mTLS; hardening de firmware; rotação segura de chaves; proteção contra extração física.
- Identity & Access Management (opcional, mas recomendado)
  - Governança no Microsoft Entra, Conditional Access, PIM, revisão de acessos e definição de escopos/roles de aplicativos.
- Observability/Blue Team
  - Configurar Log Analytics/Sentinel, dashboards, alertas, regras de correlação, playbooks de resposta (Logic Apps) e testes de detecção.
- Legal/Compliance (quando há PII/PHI)
  - Avaliar bases legais, retenção, localização de dados, contratos com SaaS e DPIA/LIA quando aplicável.

Observações finais:
- O desenho reflete um padrão de integração API-led típico no Azure: APIM como fachada, Logic Apps como orquestrador e múltiplos backends. Para produção, priorize isolamento de rede, identidades gerenciadas, validação de entrada, limitação de tráfego e observabilidade centralizada.

Sugestões de melhorias

A seguir estão: (1) diagrama Mermaid da arquitetura com fluxos e controles de segurança; (2) um script Terraform de referência para provisionar a base da solução no Azure (APIM em VNet interna, Logic Apps Standard, Key Vault, Log Analytics, DDoS, diagnósticos e políticas APIM); e (3) Relatório de Modelagem de Ameaças (STRIDE).

Diagrama Mermaid

```mermaid
flowchart LR
  %% Legend:
  %% (1) Cliente -> APIM
  %% (2) Autenticação no Entra ID
  %% (3) Políticas no APIM (JWT/RateLimit/CORS/Transformação)
  %% (4) APIM -> Logic Apps (orquestração)
  %% (5) Logic Apps -> Backends (Azure/SaaS/REST/SOAP)

  %% Nós externos
  C[Cliente Externo<br/>(Consumidor/Desenvolvedor de APIs)]
  subgraph IdP[Microsoft Entra ID (Azure AD)]
    ENTRA[Authority OAuth2/OIDC]
  end

  %% Assinatura Azure
  subgraph SUB[Assinatura Azure]
    direction TB
    RG[[Resource Group]]
    subgraph NET[VNet + NSG + DDoS Standard]
      direction TB

      %% Borda opcional
      WAF[(Application Gateway/WAF - opcional)]

      %% APIM
      subgraph APIM[Azure API Management]
        direction TB
        G[API Gateway (privado na VNet)]
        DP[Developer Portal]
        POL[Políticas: validação de JWT, rate-limit/quota,<br/>CORS restritivo, transformação, roteamento,<br/>limpeza de headers, IP allowlist, caching opcional]
      end

      %% Orquestrador
      LA[Azure Logic Apps (Standard) - Orquestração/Workflow<br/>VNet Integration, Managed Identity]

      %% Backends
      subgraph BE[Sistemas de Backend]
        direction LR
        BEAZ[Azure services (Functions/Storage/Service Bus)...<br/>(Private Link quando aplicável)]
        BESAAS[SaaS (Salesforce, O365, etc.)<br/>Conectores aprovados/DLP]
        BEREST[Web services REST<br/>TLS 1.2+, pinagem opcional]
        BESOAP[Web services SOAP<br/>TLS 1.2+, políticas de contrato]
      end

      %% Suporte/Segurança/Observabilidade
      KV[Azure Key Vault<br/>Segredos/Conexões (rotacionados)]
      LOG[Log Analytics / Monitor / Sentinel]
    end
  end

  %% Fluxos principais
  C -->|1. HTTP (descoberta/testes)| DP
  C -->|1. HTTP (chamada API)| WAF
  WAF --> G
  C -.->|1. Subscrever APIs / gestão de chaves| DP
  C -->|2. OAuth2/OIDC: obter token| ENTRA

  ENTRA -->|2. Emite token (aud/escopos/roles)| C
  G -->|3. Valida JWT (issuer/audience/assinatura),<br/>aplica políticas| POL
  POL -->|4. Chama workflow| LA

  LA -->|5a. Azure (Private Link/MIs)| BEAZ
  LA -->|5b. SaaS (conectores aprovados/DLP)| BESAAS
  LA -->|5c. REST (TLS forte)| BEREST
  LA -->|5d. SOAP (TLS/contratos)| BESOAP
  LA -->|Agrega resposta| G
```

```
  G -->|Resposta| C

  %% Identidades gerenciadas e segredos
  G -.->|Managed Identity (leitura de segredos de backend)| KV
  LA -.->|Managed Identity (conexões)| KV

  %% Observabilidade
  G -->|Diagnósticos (Gateway/Requests/Audit)| LOG
  LA -->|Execução/Telemetria| LOG
  KV -->|AuditEvent| LOG

  %% Notas de endurecimento
  note right of G
    Correções aplicadas:
    - Validação JWT (issuer/audience/assinatura)
    - Rate-limit/quotas por assinatura/IP
    - CORS restritivo
    - IP allowlist e mTLS (quando aplicável)
    - Transformação e remoção de dados sensíveis
    - Cache e proteção contra burst
  end

  note bottom of NET
    Isolamento de rede:
    - APIM em VNet (Internal) com NSG
    - DDoS Protection Standard
    - Private Link/Endpoints para backends Azure
    - Regras de saída restritivas e Firewall
  end

  note left of DP
    Developer Portal:
    - Cadastro controlado e aprovação de subscrições
    - Chaves mascaradas nos exemplos
    - Rate limit no console de testes/CAPTCHA
  end
```

Terraform (referência)
- Observações:
  - Este código cria a base segura: RG, VNet, DDoS, APIM Premium em modo Internal na VNet, Logic Apps Standard com VNet integration, Key Vault com purge protection, Log Analytics e diagnósticos. Inclui exemplo de API no APIM com políticas (JWT, rate-limit, CORS) e uso de Key Vault para esconder o URL do webhook/trigger do Logic App (placeholder).
  - Para testes, você pode trocar o SKU do APIM para Developer; para produção, mantenha Premium para VNet/Internal. O Logic App não publica um workflow aqui; após publicar o workflow HTTP trigger, substitua o segredo KV “logicapp-trigger-url”.
  - Application Gateway/WAF e Private Endpoints adicionais são recomendados, mas omitidos aqui para manter o exemplo enxuto.

```hcl
terraform {
  required_version = ">= 1.5.0"
  required_providers {
    azurerm = {
      source  = "hashicorp/azurerm"
      version = ">= 3.100.0"
    }
    random = {
      source  = "hashicorp/random"
      version = ">= 3.5.1"
    }
  }
}

provider "azurerm" {
  features {}
}
```

```
data "azurerm_client_config" "current" {}

#####################
# Variáveis padrão #
#####################
variable "prefix" {
  description = "Prefixo de nomenclatura"
  type        = string
  default     = "apim-la"
}

variable "location" {
  description = "Região Azure"
  type        = string
  default     = "eastus"
}

variable "apim_sku" {
  description = "SKU do APIM (Premium_1 recomendado para VNet Internal)"
  type        = string
  default     = "Premium_1"
}

variable "allowed_cors_origins" {
  description = "Origens permitidas (CORS) no APIM"
  type        = list(string)
  default     = ["https://example.com"]
}

variable "apim_jwt_audience" {
  description = "Audience esperado nos tokens (ex: api://seu-app-id ou App ID URI)"
  type        = string
  default     = "api://change-me-audience"
}

#########################
# Resource Group/Logs  #
#########################
resource "azurerm_resource_group" "rg" {
  name     = "${var.prefix}-rg"
  location = var.location
  tags     = { env = "prod", solution = "apim-logicapps" }
}

resource "azurerm_log_analytics_workspace" "law" {
  name                = "${var.prefix}-law"
  location            = azurerm_resource_group.rg.location
  resource_group_name = azurerm_resource_group.rg.name
  sku                 = "PerGB2018"
  retention_in_days   = 30
  tags                = azurerm_resource_group.rg.tags
}

#####################
# Rede e Proteções  #
#####################
resource "azurerm_network_ddos_protection_plan" "ddos" {
  name                = "${var.prefix}-ddos"
  location            = azurerm_resource_group.rg.location
  resource_group_name = azurerm_resource_group.rg.name
  tags                = azurerm_resource_group.rg.tags
}

resource "azurerm_virtual_network" "vnet" {
  name                = "${var.prefix}-vnet"
  address_space       = ["10.10.0.0/16"]
  location            = azurerm_resource_group.rg.location
  resource_group_name = azurerm_resource_group.rg.name
```

```
  ddos_protection_plan {
    enable = true
    id     = azurerm_network_ddos_protection_plan.ddos.id
  }

  tags = azurerm_resource_group.rg.tags
}

resource "azurerm_subnet" "apim" {
  name                 = "snet-apim"
  resource_group_name  = azurerm_resource_group.rg.name
  virtual_network_name = azurerm_virtual_network.vnet.name
  address_prefixes     = ["10.10.1.0/24"]
}

resource "azurerm_subnet" "logic" {
  name                 = "snet-logic"
  resource_group_name  = azurerm_resource_group.rg.name
  virtual_network_name = azurerm_virtual_network.vnet.name
  address_prefixes     = ["10.10.2.0/24"]
  delegation {
    name = "delegation-appsvc"
    service_delegation {
      name = "Microsoft.Web/serverFarms"
      actions = [
        "Microsoft.Network/virtualNetworks/subnets/join/action",
      ]
    }
  }
}

#################
# Key Vault     #
#################
resource "random_string" "kv" {
  length  = 6
  lower   = true
  upper   = false
  number  = true
  special = false
}

resource "azurerm_key_vault" "kv" {
  name                        = "${var.prefix}-kv-${random_string.kv.result}"
  location                    = azurerm_resource_group.rg.location
  resource_group_name         = azurerm_resource_group.rg.name
  tenant_id                   = data.azurerm_client_config.current.tenant_id
  sku_name                    = "standard"
  soft_delete_retention_days  = 90
  purge_protection_enabled    = true
  public_network_access_enabled = true

  network_acls {
    default_action = "Allow"
    bypass         = "AzureServices"
  }

  tags = azurerm_resource_group.rg.tags
}

# Placeholder a ser substituído pelo URL real do trigger do Logic App (SAS, se aplicável)
resource "azurerm_key_vault_secret" "logicapp_trigger_url" {
  name         = "logicapp-trigger-url"
  value        = "https://replace.me/logicapp/trigger?code=SAS_OR_USE_AAD"
  key_vault_id = azurerm_key_vault.kv.id
}

##########################
# Logic Apps (Standard) #
```

```
#########################
resource "azurerm_storage_account" "la" {
  name                     = "${replace(var.prefix, "-", "")}sa${random_string.kv.result}"
  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
  tags                     = azurerm_resource_group.rg.tags
}

resource "azurerm_service_plan" "la" {
  name                = "${var.prefix}-asp"
  resource_group_name = azurerm_resource_group.rg.name
  location            = azurerm_resource_group.rg.location
  os_type             = "Linux"
  sku_name            = "WS1"
  tags                = azurerm_resource_group.rg.tags
}

resource "azurerm_logic_app_standard" "la" {
  name                       = "${var.prefix}-la"
  resource_group_name        = azurerm_resource_group.rg.name
  location                   = azurerm_resource_group.rg.location
  app_service_plan_id        = azurerm_service_plan.la.id
  storage_account_name       = azurerm_storage_account.la.name
  storage_account_access_key = azurerm_storage_account.la.primary_access_key
  sku_name                   = "WS1"
  virtual_network_subnet_id  = azurerm_subnet.logic.id

  https_only = true
  identity {
    type = "SystemAssigned"
  }

  site_config {
    minimum_tls_version = "1.2"
    use_32_bit_worker   = false
    ftps_state          = "Disabled"
  }

  tags = azurerm_resource_group.rg.tags
}

###############################
# API Management (APIM)     #
###############################
resource "azurerm_api_management" "apim" {
  name                = "${var.prefix}-apim"
  location            = azurerm_resource_group.rg.location
  resource_group_name = azurerm_resource_group.rg.name

  publisher_name  = "API Team"
  publisher_email = "api-team@example.com"

  sku_name = var.apim_sku

  virtual_network_type = "Internal"
  virtual_network_configuration {
    subnet_id = azurerm_subnet.apim.id
  }

  identity {
    type = "SystemAssigned"
  }

  tags = azurerm_resource_group.rg.tags
}
```

```
# Conceder APIM e Logic App acesso a segredos no Key Vault (mínimo necessário)
resource "azurerm_key_vault_access_policy" "kv_apim" {
  key_vault_id = azurerm_key_vault.kv.id
  tenant_id    = data.azurerm_client_config.current.tenant_id
  object_id    = azurerm_api_management.apim.identity[0].principal_id

  secret_permissions = ["Get", "List"]
}

resource "azurerm_key_vault_access_policy" "kv_la" {
  key_vault_id = azurerm_key_vault.kv.id
  tenant_id    = data.azurerm_client_config.current.tenant_id
  object_id    = azurerm_logic_app_standard.la.identity[0].principal_id

  secret_permissions = ["Get", "List"]
}

# Named Value no APIM apontando para o segredo do KV (evita vazar o trigger URL)
resource "azurerm_api_management_named_value" "logicapp_trigger_url" {
  name                = "logicapp-trigger-url"
  resource_group_name = azurerm_resource_group.rg.name
  api_management_name = azurerm_api_management.apim.name
  display_name        = "LOGICAPP_TRIGGER_URL"
  secret              = true

  key_vault {
    secret_identifier = azurerm_key_vault_secret.logicapp_trigger_url.id
    # Usa identidade gerenciada do APIM (SystemAssigned)
  }

  depends_on = [azurerm_key_vault_access_policy.kv_apim]
}

# API de fachada no APIM
resource "azurerm_api_management_api" "orchestration" {
  name                = "orchestration-api"
  resource_group_name = azurerm_resource_group.rg.name
  api_management_name = azurerm_api_management.apim.name
  revision            = "1"
  display_name        = "Orchestration API"
  path                = "orchestrations"
  protocols           = ["https"]
  subscription_required = true
  tags                = ["logicapps","facade"]
}

# Políticas de segurança (JWT, CORS, Rate-limit, roteamento para Logic App)
resource "azurerm_api_management_api_policy" "orchestration_policy" {
  resource_group_name = azurerm_resource_group.rg.name
  api_management_name = azurerm_api_management.apim.name
  api_name            = azurerm_api_management_api.orchestration.name

  xml_content = <<POLICY
<policies>
  <inbound>
    <base />
    <!-- CORS restritivo -->
    <cors allow-credentials="false">
      <allowed-origins>
        ${join("\n        ", [for o in var.allowed_cors_origins : "<origin>" ~ o ~ "</origin>"])}
      </allowed-origins>
      <allowed-methods>
        <method>GET</method>
        <method>POST</method>
      </allowed-methods>
      <allowed-headers>
        <header>authorization</header>
        <header>content-type</header>
```

```
        <header>ocp-apim-subscription-key</header>
      </allowed-headers>
      <expose-headers>
        <header>request-id</header>
      </expose-headers>
    </cors>

    <!-- Rate limiting por chave de subscrição -->
    <rate-limit-by-key calls="60" renewal-period="60" increment-condition="@((bool)true)"
counter-key="@(context.Subscription?.Key ?? context.Request.IpAddress)" />

    <!-- Validação de JWT -->
    <validate-jwt header-name="Authorization" failed-validation-httpcode="401"
failed-validation-error-message="Invalid or missing token" require-scheme="Bearer">
      <openid-config
url="https://login.microsoftonline.com/${data.azurerm_client_config.current.tenant_id}/v2.0/.well-kno
wn/openid-configuration" />
      <audiences>
        <audience>${var.apim_jwt_audience}</audience>
      </audiences>
      <!-- Exigir ao menos um escopo/role -->
      <required-claims>
        <claim name="scp">
          <value>api.read</value>
        </claim>
      </required-claims>
    </validate-jwt>

    <!-- Encaminhar para Logic App (URL mantido no Key Vault) -->
    <set-backend-service base-url="{{LOGICAPP_TRIGGER_URL}}" />

    <!-- Opcional: remover cabeçalhos sensíveis da requisição -->
    <set-header name="x-powered-by" exists-action="delete" />
    <set-header name="Server" exists-action="delete" />
  </inbound>
  <backend>
    <base />
  </backend>
  <outbound>
    <base />
    <!-- Sanitização de headers de resposta -->
    <set-header name="Set-Cookie" exists-action="delete" />
    <set-header name="X-AspNet-Version" exists-action="delete" />
  </outbound>
  <on-error>
    <base />
  </on-error>
</policies>
POLICY
}

##########################
# Diagnósticos (Logs)   #
##########################
resource "azurerm_monitor_diagnostic_setting" "apim_diag" {
  name                       = "${var.prefix}-apim-diag"
  target_resource_id         = azurerm_api_management.apim.id
  log_analytics_workspace_id = azurerm_log_analytics_workspace.law.id

  log {
    category = "GatewayLogs"
    enabled  = true
  }
  log {
    category = "GatewayRequests"
    enabled  = true
  }
  log {
    category = "AuditLogs"
```

```
    enabled  = true
  }
  metric {
    category = "AllMetrics"
    enabled  = true
  }
}

resource "azurerm_monitor_diagnostic_setting" "la_diag" {
  name                       = "${var.prefix}-la-diag"
  target_resource_id         = azurerm_logic_app_standard.la.id
  log_analytics_workspace_id = azurerm_log_analytics_workspace.law.id

  # Categorias típicas de App Service/Logic App Standard
  log { category = "AppServiceHTTPLogs"      enabled = true }
  log { category = "AppServiceConsoleLogs"   enabled = true }
  log { category = "AppServiceAppLogs"       enabled = true }
  log { category = "AppServiceAuditLogs"     enabled = true }
  log { category = "AppServiceIPSecAuditLogs" enabled = true }
  log { category = "AppServicePlatformLogs"  enabled = true }

  metric {
    category = "AllMetrics"
    enabled  = true
  }
}

resource "azurerm_monitor_diagnostic_setting" "kv_diag" {
  name                       = "${var.prefix}-kv-diag"
  target_resource_id         = azurerm_key_vault.kv.id
  log_analytics_workspace_id = azurerm_log_analytics_workspace.law.id

  log { category = "AuditEvent" enabled = true }

  metric {
    category = "AllMetrics"
    enabled  = true
  }
}

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

output "apim_name" {
  value = azurerm_api_management.apim.name
}

output "apim_private_ips" {
  description = "IPs privados do gateway APIM (modo Internal)"
  value       = azurerm_api_management.apim.private_ip_addresses
}

output "logic_app_name" {
  value = azurerm_logic_app_standard.la.name
}

output "key_vault_name" {
  value = azurerm_key_vault.kv.name
}

output "log_analytics_workspace_id" {
  value = azurerm_log_analytics_workspace.law.id
}
```

Como operar após o deploy
- Publique um workflow no Logic Apps Standard com um gatilho HTTP. Se usar URL com SAS, atualize o segredo logicapp-trigger-url no Key Vault. Idealmente, habilite Azure AD Authentication no Logic App (Easy Auth) e troque a política do APIM para usar authentication-managed-identity ao chamar o backend, eliminando SAS.
- Configure aprovação manual de subscrições no APIM; crie Products e atribua quotas.
- Integre serviços Azure via Private Link quando aplicável; restrinja saída do Logic App por NSG/Firewall.
- Opcional: Coloque um Application Gateway WAF com IP público na borda e publique o APIM Internal via backend pool privado.

Relatório de Modelagem de Ameaças (STRIDE)
- Escopo e fronteiras de confiança
  - Fronteira 1: Internet/Cliente → APIM (via WAF opcional).
  - Fronteira 2: APIM → Logic Apps (serviços gerenciados no Azure, rede privada).
  - Fronteira 3: Logic Apps → Backends (Azure/SaaS/REST/SOAP; alguns fora da autoridade).
  - Armazenamentos: históricos de execução do Logic Apps; logs/diagnósticos APIM/Monitor; Key Vault (segredos).
- Ativos principais
  - Credenciais/tokens do Entra; chaves de subscrição do APIM; payloads e metadados das APIs; políticas de APIM; segredos/conexões do Logic Apps; dados de negócio.
- STRIDE por fluxo/componente
  - S (Spoofing)
    - Riscos: Falsificação de cliente/serviço; tokens roubados; endpoints falsos.
    - Mitigações: OAuth2/OIDC; validação JWT no APIM (issuer/audience/assinatura); Conditional Access; mTLS quando aplicável; Managed Identity para chamadas internas; pinagem/validação de certificados em saídas.
  - T (Tampering)
    - Riscos: Alteração de requisições/respostas; manipulação de políticas APIM; alteração de workflows.
    - Mitigações: TLS 1.2+ end-to-end; CI/CD com revisão de PR para políticas e workflows; RBAC mínimo no Azure; Azure Policy/Blueprints; versionamento de APIs; assinatura de mensagens para webhooks críticos.
  - R (Repudiation)
    - Riscos: Ausência de rastreabilidade e correlação.
    - Mitigações: Diagnósticos detalhados no APIM/Logic Apps com correlação (trace-id); retenção/imutabilidade (WORM) quando necessário; sincronização de tempo; integração com SIEM (Microsoft Sentinel).
  - I (Information Disclosure)
    - Riscos: Vazamento em trânsito, logs ou conectores; CORS amplo; exfiltração para SaaS.
    - Mitigações: TLS forte; mascaramento/redação em logs; CORS restrito; DLP e classificação; Private Link; criptografia at-rest (CMK se necessário); segredos no Key Vault com rotação.
  - D (Denial of Service)
    - Riscos: Saturação do APIM/Logic Apps; cascata de indisponibilidades.
    - Mitigações: Rate-limit/quota e burst control no APIM; Azure DDoS Protection Standard; caching; timeouts/retries exponenciais; circuit breaker/bulkhead; autoscaling onde suportado.
  - E (Elevation of Privilege)
    - Riscos: Identidades gerenciadas superpermissivas; abuso do Developer Portal; manipulação de políticas.
    - Mitigações: RBAC mínimo/PIM; separação de ambientes e funções; aprovação manual de subscrições; revisão periódica de políticas; varreduras de postura (Defender for Cloud).
- Suposições e dependências
  - Microsoft Entra como IdP; TLS obrigatório; conectividade a SaaS via conectores aprovados; governança de dados ativa.
- Riscos residuais
  - Dependências de terceiros/SaaS; complexidade de orquestração; dados sensíveis nos históricos do Logic Apps exigem governança estrita.
- Plano de tratamento
  - Curto prazo: habilitar validação de JWT e rate limiting no APIM; mover segredos para Key Vault; ativar diagnósticos com mascaramento de PII.
  - Médio prazo: VNet/Private Endpoints; WAF e DDoS; Managed Identities ponta-a-ponta; CI/CD com testes de contrato.
  - Longo prazo: APIM/Logic Apps multi-região; SIEM com regras customizadas; DLP/classificação de dados; revisões periódicas de acesso e pentests.

Recursos humanos adicionais
- Network Administrator: VNet/Private Endpoints/NSG/Firewall, conectividade híbrida, DDoS/WAF.
- Security Officer: políticas CORS/criptografia/DLP/logs, conformidade.
- Cloud/DevOps: IaC (Terraform), pipelines CI/CD, integração com Key Vault.

- Software Developer: contratos OpenAPI, versionamento, políticas APIM (transformação/validação), clientes OAuth2/PKCE.
- IAM: Governança no Entra, Conditional Access, PIM, revisão de escopos/roles.
- Observability/Blue Team: Log Analytics/Sentinel, dashboards/alertas, playbooks (Logic Apps).
- Legal/Compliance: bases legais, retenção/localização, contratos com SaaS, DPIA/LIA.

Observações finais
- O padrão proposto cobre os controles essenciais: isolamento de rede (APIM Internal), identidades gerenciadas, validação/limitação de tráfego, armazenamento seguro de segredos e observabilidade centralizada. Para produção, complemente com WAF na borda, Private Link consistente aos backends, políticas de saída restritivas e automação de governança e segurança via CI/CD.