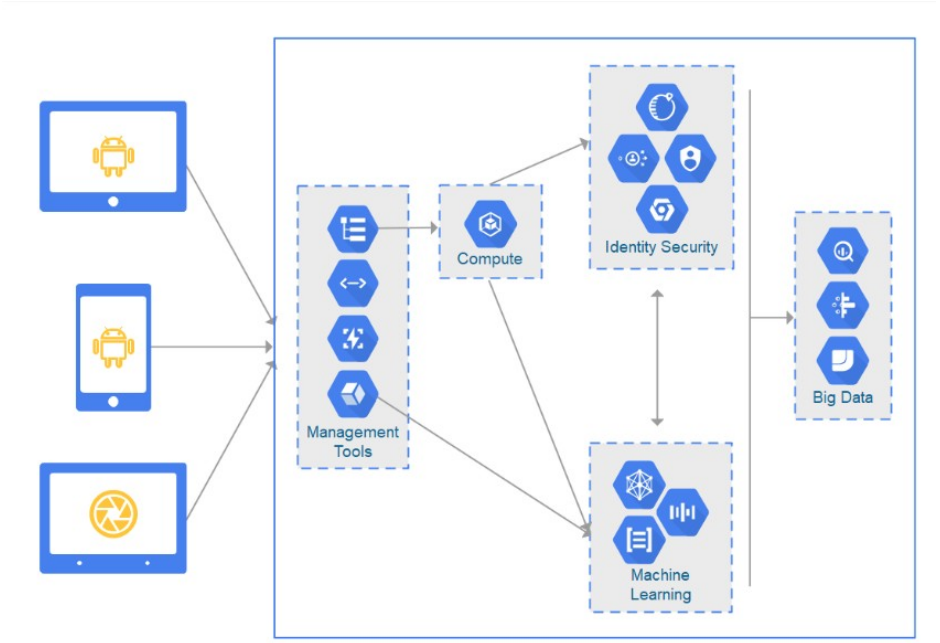
Relatório de análise da solução: tmpylq54dzo



Análise completa da solução atual

Modelo de cloud:
Google Cloud Platform (GCP)

Lista com os componentes:
- Dispositivos clientes:
  - Apps móveis Android (coleta de telemetria/uso) e tablet com câmera (captura de imagens/vídeo)
- Ingresso/Exposição de API (implícitos no diagrama GCP):
  - HTTPS Load Balancer + Cloud Endpoints/API Gateway
- Management Tools (conjunto de ícones hexagonais de “ferramentas”):
  - Repositório de código e CI/CD (Cloud Source Repositories, Cloud Build)
  - Orquestração de infraestrutura (Deployment Manager/Terraform)
  - Observabilidade (Cloud Monitoring, Cloud Logging)
  - Artefatos/Imagens (Artifact Registry)
- Compute (bloco “Compute”):
  - Camada de processamento/serviços: Compute Engine, Google Kubernetes Engine (GKE) e/ou Cloud Run/App Engine
- Identity & Security (bloco “Identity Security”):
  - IAM/Cloud Identity, Secret Manager, Cloud KMS, Identity-Aware Proxy (IAP), Security Command Center
- Big Data (bloco “Big Data”):
  - Pub/Sub (ingestão), Dataflow (ETL/stream), BigQuery (DW/analytics), Cloud Storage (data lake)
- Machine Learning (bloco “Machine Learning”):
  - Vertex AI (treino/serving), APIs de IA (ex.: Vision API para imagens)
- Rede/limites de confiança (implícitos):
  - VPC, Private Google Access, VPC Service Controls, Cloud NAT, Cloud Armor (WAF/Rate limiting)

Interação entre os componentes:
- Os dispositivos Android e o tablet com câmera enviam eventos/imagens via HTTPS para API Gateway/Cloud Endpoints no GCP.
- A camada de Compute expõe APIs, pré-processa dados e publica mensagens no Pub/Sub. Autenticação e autorização são mediadas por IAM/IAP; segredos e chaves via Secret Manager/KMS.
- Dataflow lê do Pub/Sub, transforma/valida dados e grava no BigQuery (dados estruturados) e Cloud Storage (blobs/imagens).
- Machine Learning (Vertex AI) consome dados do BigQuery/Cloud Storage para treino; modelos são versionados e servidos (online) em Vertex AI ou em serviços na camada Compute.

- Resultados de inferência e métricas retornam para BigQuery/Storage e são expostos por APIs na Compute.
- Management Tools provêem CI/CD, implantação em Compute (GKE/Cloud Run/GCE), observabilidade e governança.
- Identity & Security estabelece políticas de IAM, proteção de segredos, criptografia (CMEK via KMS) e postura de segurança em todo o fluxo.
- Big Data e Machine Learning trocam dados bidirecionalmente (treino, feature store, inferência), sob controle de IAM e VPC Service Controls.

O que esse sistema faz:
- Plataforma de ingestão e processamento de dados móveis e de imagens, com armazenamento analítico e treinamento/serving de modelos de Machine Learning. Provê APIs para coleta, análise e inferência em tempo real e em batch.

Vulnerabilidades e Solução para cada vulnerabilidade:
1) Tráfego de app para nuvem sem proteção robusta
- Risco: interceptação/manipulação de dados.
- Mitigação: HTTPS obrigatório com TLS 1.2+ e HSTS; opção por mTLS entre app e API; pinagem de certificado no app.

2) Autenticação fraca de clientes e chaves expostas no app
- Risco: spoofing de dispositivo/usuário e abuso de API.
- Mitigação: OAuth 2.0/OIDC, Identity-Aware Proxy, troca de API keys por tokens curtos; Play Integrity/DeviceCheck; rotação automática de credenciais; escopo mínimo.

3) Buckets/BigQuery expostos por erro de configuração
- Risco: vazamento de dados.
- Mitigação: IAM com princípio do menor privilégio; Uniform bucket-level access; prevenção de ACLs públicas; VPC Service Controls; Data Catalog + políticas de governança.

4) Tampering de payloads e injeções em APIs/ETL
- Risco: corrupção de dados, execução indevida.
- Mitigação: validação de esquema e tamanho; assinatura/HMAC do payload; sanitização; Cloud Armor (WAF/OWASP); verificação de conteúdo (antimalware).

5) Supply chain de build/deploy
- Risco: imagens maliciosas, backdoors.
- Mitigação: SLSA nível 2+; Cloud Build com provenance; Artifact Registry com scanning; Binary Authorization (enforcement de políticas) no GKE/Cloud Run.

6) Segredos em código ou imagem
- Risco: comprometimento lateral.
- Mitigação: Secret Manager; remoção de chaves estáticas; Workload Identity Federation; rotação/KMS; detecção preventiva (secret scanning em CI).

7) Privilégios excessivos em IAM e contas de serviço
- Risco: elevação de privilégio e movimento lateral.
- Mitigação: papéis customizados com menor privilégio; IAM Conditions; desabilitar chaves externas de SA; separação por projeto/ambiente.

8) DoS e floods em APIs/Pub/Sub
- Risco: indisponibilidade e custos.
- Mitigação: Cloud Armor (rate limiting, geo/IP rules), quotas por consumidor, backpressure no Dataflow, DLQ no Pub/Sub, autoscaling com limites, cache/CDN quando aplicável.

9) Exfiltração via egress/serviços externos
- Risco: perda de dados confidenciais.
- Mitigação: VPC Service Controls, Private Google Access, egress allowlist, DNS policy, Cloud DLP para monitorar saídas, Alertas no SCC.

10) Poisoning de dados/modelos e inferência reversa
- Risco: modelos enviesados/comprometidos; vazamento de dados pelo modelo.
- Mitigação: validação e versão de datasets; canary/rollback de modelos; detecção de drift; limitação de explicabilidade sensível; teste de membership inference.

11) Integridade de objetos e logs
- Risco: repúdio ou alteração de evidências.
- Mitigação: Log buckets com retenção e CMEK; sinks para BigQuery; Object Versioning + Retention Policy/Lock; assinaturas (Tink/KMS).

12) Configuração derivando ao longo do tempo
- Risco: deriva de segurança.
- Mitigação: Infra as Code (Terraform) + revisão; Policy as Code (Config Validator/OPA); Monitoramento contínuo no Security Command Center.

Gere um Relatório de Modelagem de Ameaças, baseado na metodologia STRIDE:
- Escopo e ativos
  - Escopo: apps móveis/câmeras, APIs em Compute, Pub/Sub→Dataflow→BigQuery/Storage, Vertex AI, IAM/KMS/Secret Manager, CI/CD.
  - Ativos: dados de usuários/imagens, modelos e features, credenciais/segredos, chaves KMS, logs/auditoria, pipelines CI/CD, imagens de container.
  - Limites de confiança: Dispositivo → Internet → GCP (VPC) → Serviços gerenciados; fronteiras de identidade (usuário/serviço); limites de dados (em trânsito/em repouso).

- DFD resumido
  1) App envia dados via HTTPS para API (Load Balancer + Endpoints).
  2) Serviço em Compute valida, autentica, publica no Pub/Sub e/ou escreve em Storage.
  3) Dataflow processa e grava no BigQuery/Storage.
  4) Vertex AI treina/serve e interage com BigQuery/Storage/Compute.
  5) Ferramentas de gestão implantam/monitoram; IAM/KMS regem acesso e criptografia.

- STRIDE
  - Spoofing
    - Ameaças: dispositivos falsos; roubo de tokens; impersonação de serviço.
    - Controles: OIDC/OAuth2 com IAP; atestação de dispositivo; mTLS; validação de JWT em Endpoints; Workload Identity Federation; no SA keys.
  - Tampering
    - Ameaças: alteração de payloads, imagens de container, configurações/IaC.
    - Controles: TLS + HMAC; Binary Authorization; provenance (SLSA); proteção de branches; assinaturas/Checksums; Object Versioning/Retention.
  - Repudiation
    - Ameaças: ações sem rastro; logs alterados.
    - Controles: Audit Logs habilitados para Admin/Data/Access; retenção imutável; sincronização de tempo; trilhas de aprovação em CI/CD.
  - Information Disclosure
    - Ameaças: buckets públicos; consulta indevida a BigQuery; vazamento em inferência.
    - Controles: IAM mínimo; Row/Column-level security; DLP para mascaramento; VPC SC; CMEK/CSEK; segregação por projeto/ambiente.
  - Denial of Service
    - Ameaças: floods HTTP; tópicos Pub/Sub saturados; jobs/consultas custosas.
    - Controles: Cloud Armor (WAF/rate); autoscaling com limites; DLQ e backoff; quotas; isolamento por região; orçamento/alertas.
  - Elevation of Privilege
    - Ameaças: exploração de container/VM; papéis amplos; exploração de metadados.
    - Controles: GKE/Compute hardening (Shielded VMs, GKE Sandbox), desabilitar legacy metadata; IAM Conditions; SCC + Alertas; patching contínuo.

- Riscos residuais e plano
  - Risco de supply chain residual: manter verificação de provenance e política BinAuthZ obrigatória; auditoria trimestral.
  - Risco de drift de dados/modelos: monitorar drift/qualidade; gatilhos de re-treino e revisão humana.
  - Risco de custo/DoS lógico: budgets com shutdown automático e rate limits por cliente.

- Métricas/SLOs de segurança
  - Cobertura de logs 100% em projetos de produção.
  - 0 segredos em repositórios; rotação de segredos ≤ 90 dias.
  - Bloqueio de imagens sem attestation: 100% em produção.
  - Bucket público não permitido por Org Policy (0 incidentes).

- Planos de resposta a incidentes
  - Playbooks para vazamento de dados, comprometimento de chaves, falhas de modelo.
  - Sinks de logs para SIEM e testes de simulação (tabletop) semestrais.

Additional resourcing needs
- Network Administrator: requisitos de VPC, firewall, DNS, conectividade privada e VPC Service Controls.
- Security Officer: alinhamento a políticas da autoridade, classificação de dados, aceitação de risco e resposta a incidentes.

- Cloud/DevOps: CI/CD, SLSA/Binary Authorization, IaC/Policy as Code, observabilidade e custos.
- Software Developer: desenho de APIs, segurança de chamadas, validação de payloads, versionamento e testes.
- Data/ML Engineer: pipelines de dados, governança (DLP/BigQuery), treinamento/serving no Vertex AI, monitoração de drift.
- Hardware/Mobile Developer: segurança do app/dispositivo, proteção de credenciais, atestação/anti-tampering.
- IAM/KMS Admin: desenho de papéis mínimos, chaves CMEK, rotação e segregação por ambiente.
- Compliance/Legal (quando aplicável): LGPD, retenção e requisitos contratuais de fornecedores.

Sugestões de melhorias

A seguir estão: (1) o diagrama Mermaid com os componentes, relações e mitigações; (2) um Terraform de referência (opinionado) para provisionar a base da solução na GCP com controles de segurança; e (3) o Relatório de Modelagem de Ameaças (STRIDE) solicitado.

Diagrama (Mermaid)
```mermaid
flowchart LR
  %% Dispositivos
  subgraph Cliente[Dispositivos clientes]
    A1[Apps Android\n(Telemetria/uso)]
    A2[Tablet c/ câmera\n(Imagens/Vídeo)]
  end

  %% Borda / Exposição
  subgraph Edge[Exposição/Entrada]
    LB[HTTPS Load Balancer\n+ Cloud Armor (WAF/Rate)]
    APIGW[Cloud Endpoints/API Gateway\n(opcional, OpenAPI/JWT)]
  end

  %% Rede / Confiança
  subgraph Net[Rede e limites]
    VPC[VPC + Firewall]
    PGA[Private Google Access]
    NAT[Cloud NAT]
    VPCSC[VPC Service Controls\n(perímetro de serviço)]
  end

  %% Identidade & Segurança
  subgraph Sec[Identity & Security]
    IAM[IAM/Cloud Identity]
    IAP[IAP (acesso a apps web/APIs)]
    SM[Secret Manager]
    KMS[Cloud KMS (CMEK)]
    SCC[Security Command Center]
  end

  %% Compute
  subgraph C[Compute]
    CR[Cloud Run / App Engine /\nGKE / Compute Engine\n(Serviços/API + pré-processamento)]
  end

  %% Big Data
  subgraph D[Big Data]
    PS[Pub/Sub]
    DF[Dataflow (ETL/stream)]
    BQ[BigQuery (DW/Analytics)]
    GCS[Cloud Storage (Data Lake/Blobs)]
  end

  %% ML
  subgraph ML[Machine Learning]
    VAI[Vertex AI (treino/serving)]
    VISION[Vision API\n(processamento de imagem)]
  end

  %% Gestão/DevSecOps
```

```
  subgraph Mgmt[Management/DevSecOps]
    SRC[Source Repos]
    CB[Cloud Build (SLSA/Provenance)]
    AR[Artifact Registry + Scanning]
    BA[Binary Authorization]
    MON[Cloud Monitoring/Logging]
    TF[Terraform/Policy as Code]
  end

  %% Fluxos principais
  A1-->|"HTTPS + TLS1.2+/mTLS\nPinagem de cert (1)"|LB
  A2-->|"HTTPS + TLS1.2+/mTLS\nPinagem de cert (1)"|LB
  LB-->|"JWT/OIDC + IAP\nQuotas/Rate (2,8)"|APIGW
  APIGW-->|"AuthZ IAM/IAP;\nSanitização/validação (2,4)"|CR
  CR-->|"Publica eventos"|PS
  CR-->|"Upload blobs"|GCS
  PS-->|"Ingestão stream"|DF
  DF-->|"Transforma/valida (4)"|BQ
  DF-->|"Grava blobs"|GCS
  VAI<-->|"Treino/Features/Inferência"|BQ
  VAI<-->|"Treino/Blobs/Artefatos"|GCS
  CR<-->|"Inferência online"|VAI
  CR-->|"Chamada a Vision API"|VISION

  %% Controles transversais
  CR---SM
  CR---KMS
  DF---SM
  DF---KMS
  D---KMS
  ML---KMS
  Edge-.->|WAF/OWASP, Rate limit (4,8)|LB
  VPC---PGA
  VPC---NAT
  VPCSC-.->|Perímetro de dados (3,9)|D
  SCC-.->|Postura/Alertas|Sec
  MON-.->|Logs/Métricas/Traces|Mgmt
  CB-.->|CI/CD + Scans (5,6)|AR
  AR-.->|Imagens assinadas|BA
  BA-.->|Enforcement em GKE/Run (5)|C
  TF-.->|IaC + Policy as Code (12)|Net
  IAM-.->|Princípio do menor privilégio (3,7)|Sec
  IAP-.->|Proteção de APIs (2)|C

  %% Mitigações por domínio (rótulos de apoio)
  M1[[Borda/Clientes:\n(1) TLS/HSTS/mTLS/Pinning\n(2) OIDC/IAP/Play Integrity\n(4)
WAF/Validação\n(8) Rate/Quotas]]
  M2[[Dados:\n(3) UBLE, Sem ACL pública,\nVPC SC, Governança/DLP\n(11)
Versioning+Retention+Assinaturas]]
  M3[[Supply chain:\n(5) SLSA/Provenance/Scanning/BinAuthZ\n(6) Secret Manager/WIF/Rotação]]
  M4[[Acesso/Privilégios:\n(7) Papéis mínimos/Conditions\nDesabilitar SA keys externas]]
  M5[[Resiliência/egress:\n(8) DLQ/Backpressure/Autoscaling\n(9) Egress allowlist/DNS policy]]
  M6[[ML:\n(10) Validação dataset, canary,\ndrift/membership tests]]
  M7[[Governança contínua:\n(12) IaC/Policy-as-Code\nSCC/Monitoramento]]

  M1-.->Edge
  M2-.->D
  M3-.->Mgmt
  M4-.->Sec
  M5-.->Net
  M6-.->ML
  M7-.->Mgmt
```

Terraform (baseline seguro e opinado)
- Observações importantes:
  - Ajuste variáveis project_id, region, bq_location, domain e, se desejar IAP e orçamento, iap_oauth_client_id/secret e billing_account_id.
  - O Load Balancer HTTPS com Cloud Armor protege um backend Cloud Run (ingresso via LB). IAP

é opcional (requer OAuth client).
  - Buckets e BigQuery usam CMEK (KMS) e políticas de retenção e versionamento.
  - Pub/Sub tem DLQ; quotas e rate limiting são endereçados via Cloud Armor e API rate/quotas (no app/API Gateway).
  - Alguns controles org-wide (VPC SC, SCC profundo, Org Policies) exigem nível de organização; incluímos exemplos opcionais.

```hcl
terraform {
  required_version = ">= 1.5.0"
  required_providers {
    google = {
      source  = "hashicorp/google"
      version = ">= 5.30.0"
    }
    google-beta = {
      source  = "hashicorp/google-beta"
      version = ">= 5.30.0"
    }
  }
}

provider "google" {
  project = var.project_id
  region  = var.region
}

provider "google-beta" {
  project = var.project_id
  region  = var.region
}

data "google_project" "current" {}

##############################
# APIs necessárias
##############################
resource "google_project_service" "services" {
  for_each = toset([
    "compute.googleapis.com",
    "run.googleapis.com",
    "iam.googleapis.com",
    "cloudkms.googleapis.com",
    "secretmanager.googleapis.com",
    "pubsub.googleapis.com",
    "bigquery.googleapis.com",
    "bigquerydatatransfer.googleapis.com",
    "storage.googleapis.com",
    "logging.googleapis.com",
    "monitoring.googleapis.com",
    "artifactregistry.googleapis.com",
    "cloudbuild.googleapis.com",
    "iap.googleapis.com",
    "aiplatform.googleapis.com",
    "dataflow.googleapis.com",
    "dlp.googleapis.com",
    "servicemanagement.googleapis.com",
    "servicecontrol.googleapis.com",
    "binaryauthorization.googleapis.com",
    "accesscontextmanager.googleapis.com"
  ])
  project            = var.project_id
  service            = each.value
  disable_on_destroy  = false
}

##############################
# Rede: VPC, Subnet, NAT, PGA
##############################
```

```
resource "google_compute_network" "vpc" {
  name                    = "main-vpc"
  auto_create_subnetworks = false
}

resource "google_compute_subnetwork" "subnet" {
  name                  = "main-subnet"
  ip_cidr_range         = "10.10.0.0/20"
  region                = var.region
  network               = google_compute_network.vpc.id
  private_ip_google_access = true
}

resource "google_compute_router" "router" {
  name    = "main-router"
  region  = var.region
  network = google_compute_network.vpc.id
}

resource "google_compute_router_nat" "nat" {
  name                               = "main-nat"
  router                             = google_compute_router.router.name
  region                             = var.region
  nat_ip_allocate_option             = "AUTO_ONLY"
  source_subnetwork_ip_ranges_to_nat = "LIST_OF_SUBNETWORKS"
  subnetwork {
    name                    = google_compute_subnetwork.subnet.name
    source_ip_ranges_to_nat = ["ALL_IP_RANGES"]
  }
}

##############################
# KMS (CMEK) para dados
##############################
resource "google_kms_key_ring" "kr" {
  name     = "data-kr"
  location = var.kms_location
}

resource "google_kms_crypto_key" "ck" {
  name            = "data-key"
  key_ring        = google_kms_key_ring.kr.id
  rotation_period = "7776000s" # 90 dias
  lifecycle {
    prevent_destroy = true
  }
}

##############################
# Buckets Storage (Data Lake) com segurança
##############################
resource "google_storage_bucket" "data_lake" {
  name                        = "${var.project_id}-datalake"
  location                    = var.storage_location
  uniform_bucket_level_access = true
  public_access_prevention    = "enforced"
  force_destroy               = false

  versioning {
    enabled = true
  }

  retention_policy {
    retention_period = 60 * 60 * 24 * var.storage_retention_days
    is_locked        = false
  }

  encryption {
    default_kms_key_name = google_kms_crypto_key.ck.id
```

```
  }

  lifecycle_rule {
    action { type = "Delete" }
    condition { age = var.storage_delete_after_days }
  }
}

##############################
# BigQuery Dataset com CMEK
##############################
resource "google_bigquery_dataset" "analytics" {
  dataset_id                  = "analytics"
  location                    = var.bq_location
  delete_contents_on_destroy  = false
  default_encryption_configuration {
    kms_key_name = google_kms_crypto_key.ck.id
  }
}

##############################
# Pub/Sub com DLQ
##############################
resource "google_pubsub_topic" "telemetry" {
  name = "telemetry"
  message_storage_policy {
    allowed_persistence_regions = [var.region]
  }
  kms_key_name = google_kms_crypto_key.ck.id
}

resource "google_pubsub_topic" "telemetry_dlq" {
  name = "telemetry-dlq"
  message_storage_policy {
    allowed_persistence_regions = [var.region]
  }
  kms_key_name = google_kms_crypto_key.ck.id
}

resource "google_pubsub_subscription" "telemetry_sub" {
  name  = "telemetry-sub"
  topic = google_pubsub_topic.telemetry.name

  ack_deadline_seconds = 30

  dead_letter_policy {
    dead_letter_topic     = google_pubsub_topic.telemetry_dlq.id
    max_delivery_attempts = 5
  }

  retry_policy {
    minimum_backoff = "10s"
    maximum_backoff = "600s"
  }

  enable_message_ordering = false
}

##############################
# Secret Manager (exemplo)
##############################
resource "google_secret_manager_secret" "app_config" {
  secret_id = "app-config"
  replication {
    user_managed {
      replicas { location = var.region }
    }
  }
}
```

```
resource "google_secret_manager_secret_version" "app_config_v1" {
  secret      = google_secret_manager_secret.app_config.id
  secret_data = base64encode("{}") # substitua pelo conteúdo codificado
}

##############################
# Artifact Registry (Docker)
##############################
resource "google_artifact_registry_repository" "repo" {
  location      = var.region
  repository_id = "containers"
  description   = "Repositório de imagens"
  format        = "DOCKER"
}

##############################
# Cloud Run (API) — backend
##############################
resource "google_service_account" "app_sa" {
  account_id   = "app-sa"
  display_name = "App Runtime SA"
}

resource "google_cloud_run_v2_service" "api" {
  name     = "mobile-api"
  location = var.region
  ingress  = "INGRESS_TRAFFIC_INTERNAL_ONLY" # protegido por LB via NEG serverless

  template {
    service_account = google_service_account.app_sa.email
    scaling {
      min_instance_count = 0
      max_instance_count = 10
    }
    containers {
      image = var.app_image # ex: "us-docker.pkg.dev/cloudrun/container/hello"
      env {
        name  = "PUBSUB_TOPIC"
        value = google_pubsub_topic.telemetry.name
      }
    }
  }

  depends_on = [google_project_service.services]
}

# Permitir que o Load Balancer invoque o Cloud Run (serverless service agent)
resource "google_cloud_run_v2_service_iam_member" "run_invoker_lb" {
  name     = google_cloud_run_v2_service.api.name
  location = var.region
  role     = "roles/run.invoker"
  member   =
"serviceAccount:service-${data.google_project.current.number}@serverless-robot-prod.iam.gservicea
ccount.com"
}

# Permissões mínimas da SA do app
resource "google_project_iam_member" "app_sa_pubsub_pub" {
  role   = "roles/pubsub.publisher"
  member = "serviceAccount:${google_service_account.app_sa.email}"
}
resource "google_project_iam_member" "app_sa_secret_access" {
  role   = "roles/secretmanager.secretAccessor"
  member = "serviceAccount:${google_service_account.app_sa.email}"
}
resource "google_kms_crypto_key_iam_member" "app_sa_kms_decrypt" {
  crypto_key_id = google_kms_crypto_key.ck.id
  role          = "roles/cloudkms.cryptoKeyDecrypter"
```

```
  member       = "serviceAccount:${google_service_account.app_sa.email}"
}

##############################
# Serverless NEG + HTTPS LB + Cloud Armor (WAF/Rate) + TLS 1.2+
##############################
resource "google_compute_region_network_endpoint_group" "serverless_neg" {
  name                  = "cr-neg"
  network_endpoint_type = "SERVERLESS"
  region                = var.region

  cloud_run {
    service = google_cloud_run_v2_service.api.name
  }
}

resource "google_compute_security_policy" "armor" {
  name        = "armor-waf-rate"
  description = "WAF OWASP + Rate limiting"

  # Rate limit global por IP
  rule {
    priority = 1
    action   = "rate_based_ban"
    match {
      versioned_expr = "SRC_IPS_V1"
      config {
        src_ip_ranges = ["*"]
      }
    }
    rate_limit_options {
      rate_limit_threshold {
        count       = 300
        interval_sec = 60
      }
      ban_threshold {
        count       = 900
        interval_sec = 300
      }
      ban_duration_sec = 600
      enforce_on_key   = "IP"
      conform_action   = "allow"
      exceed_action    = "deny(429)"
    }
    description = "Rate limiting por IP"
  }

  # Exemplo de regra OWASP CRS (SQLi)
  rule {
    priority = 1000
    action   = "deny(403)"
    match {
      expr {
        expression = "evaluatePreconfiguredWaf('owasp-crs-v030001-id942100-sqli')"
      }
    }
    description = "Bloqueio SQLi"
  }

  # Exemplo XSS
  rule {
    priority = 1001
    action   = "deny(403)"
    match {
      expr {
        expression = "evaluatePreconfiguredWaf('owasp-crs-v030001-id941100-xss')"
      }
    }
    description = "Bloqueio XSS"
```

```
  }
}

resource "google_compute_backend_service" "lb_backend" {
  name                  = "cr-backend"
  protocol              = "HTTP"
  timeout_sec           = 30
  security_policy       = google_compute_security_policy.armor.id
  load_balancing_scheme = "EXTERNAL_MANAGED"

  backend {
    group = google_compute_region_network_endpoint_group.serverless_neg.id
  }
}

# Política TLS (forçar TLS 1.2+)
resource "google_compute_ssl_policy" "tls_policy" {
  name            = "tls12-modern"
  min_tls_version = "TLS_1_2"
  profile         = "MODERN"
}

# IP global
resource "google_compute_global_address" "lb_ip" {
  name = "api-lb-ip"
}

# Certificado gerenciado (exige validação DNS do domínio)
resource "google_compute_managed_ssl_certificate" "cert" {
  name = "api-cert"
  managed {
    domains = [var.domain]
  }
}

resource "google_compute_url_map" "urlmap" {
  name            = "api-urlmap"
  default_service = google_compute_backend_service.lb_backend.id
}

resource "google_compute_target_https_proxy" "https_proxy" {
  name             = "api-https-proxy"
  url_map          = google_compute_url_map.urlmap.id
  ssl_certificates = [google_compute_managed_ssl_certificate.cert.id]
  ssl_policy       = google_compute_ssl_policy.tls_policy.id
}

resource "google_compute_global_forwarding_rule" "fr_https" {
  name                  = "api-forwarding-rule-https"
  target                = google_compute_target_https_proxy.https_proxy.id
  port_range            = "443"
  ip_protocol           = "TCP"
  load_balancing_scheme = "EXTERNAL_MANAGED"
  ip_address            = google_compute_global_address.lb_ip.id
}

# IAP opcional para o Backend Service (requer OAuth)
resource "google_iap_web_backend_service" "iap" {
  count          = var.enable_iap ? 1 : 0
  project        = var.project_id
  backend_service = google_compute_backend_service.lb_backend.name
  iap {
    enabled              = true
    oauth2_client_id     = var.iap_oauth_client_id
    oauth2_client_secret = var.iap_oauth_client_secret
  }
}

############################
```

```
# Dataflow (SA + permissões mínimas)
##############################
resource "google_service_account" "dataflow_sa" {
  account_id   = "dataflow-sa"
  display_name = "Dataflow SA"
}

resource "google_project_iam_member" "df_worker" {
  role   = "roles/dataflow.worker"
  member = "serviceAccount:${google_service_account.dataflow_sa.email}"
}
resource "google_project_iam_member" "df_pubsub_sub" {
  role   = "roles/pubsub.subscriber"
  member = "serviceAccount:${google_service_account.dataflow_sa.email}"
}
resource "google_project_iam_member" "df_bq_editor" {
  role   = "roles/bigquery.dataEditor"
  member = "serviceAccount:${google_service_account.dataflow_sa.email}"
}
resource "google_project_iam_member" "df_gcs_writer" {
  role   = "roles/storage.objectUser"
  member = "serviceAccount:${google_service_account.dataflow_sa.email}"
}
resource "google_kms_crypto_key_iam_member" "df_kms_decrypt" {
  crypto_key_id = google_kms_crypto_key.ck.id
  role          = "roles/cloudkms.cryptoKeyDecrypter"
  member        = "serviceAccount:${google_service_account.dataflow_sa.email}"
}

##############################
# Vertex AI Endpoint (placeholder)
##############################
resource "google_vertex_ai_endpoint" "endpoint" {
  name     = "online-inference"
  region   = var.region
  display_name = "Online Inference"
}

##############################
# Logging: sink de auditoria -> BigQuery
##############################
resource "google_logging_project_sink" "audit_to_bq" {
  name        = "audit-logs-to-bq"
  destination =
"bigquery.googleapis.com/projects/${var.project_id}/datasets/${google_bigquery_dataset.analytics.dat
aset_id}"

  # Todos os logs de auditoria; ajuste conforme necessário
  filter = "logName:\"cloudaudit.googleapis.com\""

  unique_writer_identity = true
}

resource "google_bigquery_dataset_iam_member" "sink_writer" {
  dataset_id = google_bigquery_dataset.analytics.dataset_id
  role       = "roles/bigquery.dataEditor"
  member     = google_logging_project_sink.audit_to_bq.writer_identity
}

##############################
# Org/Project Policies (mitigações 3,6,7,12)
##############################
resource "google_project_organization_policy" "no_public_buckets" {
  project = var.project_id
  constraint = "constraints/storage.publicAccessPrevention"
  boolean_policy {
    enforced = true
  }
}
```

```
resource "google_project_organization_policy" "disable_sa_key_creation" {
  project    = var.project_id
  constraint = "constraints/iam.disableServiceAccountKeyCreation"
  boolean_policy { enforced = true }
}

##############################
# Budget (DoS lógico/custos) — opcional
##############################
resource "google_billing_budget_budget" "budget" {
  count             = var.billing_account_id == "" ? 0 : 1
  billing_account   = var.billing_account_id
  display_name      = "prod-budget"
  amount {
    specified_amount {
      currency_code = "USD"
      units         = var.monthly_budget_usd
    }
  }
  budget_filter {
    projects = ["projects/${var.project_id}"]
  }
  threshold_rules {
    threshold_percent = 0.5
  }
  threshold_rules {
    threshold_percent = 0.8
  }
  threshold_rules {
    threshold_percent = 1.0
  }
}

##############################
# Variáveis
##############################
variable "project_id" {
  description = "ID do projeto GCP"
  type        = string
}

variable "region" {
  description = "Região padrão (ex.: us-central1)"
  type        = string
}

variable "storage_location" {
  description = "Local do bucket (ex.: US ou regionais)"
  type        = string
  default     = "US"
}

variable "bq_location" {
  description = "Local do BigQuery (ex.: US, EU)"
  type        = string
  default     = "US"
}

variable "kms_location" {
  description = "Local do KMS (ex.: us)"
  type        = string
  default     = "us"
}

variable "app_image" {
  description = "Imagem do container da API (Cloud Run)"
  type        = string
  default     = "us-docker.pkg.dev/cloudrun/container/hello"
```

```
}

variable "domain" {
  description = "Domínio público para o certificado TLS gerenciado (ex.: api.example.com)"
  type        = string
}

variable "enable_iap" {
  description = "Habilita IAP no Backend Service do LB"
  type        = bool
  default     = false
}

variable "iap_oauth_client_id" {
  description = "OAuth Client ID para IAP"
  type        = string
  default     = ""
}

variable "iap_oauth_client_secret" {
  description = "OAuth Client Secret para IAP"
  type        = string
  default     = ""
  sensitive   = true
}

variable "billing_account_id" {
  description = "Billing Account ID para orçamento (opcional)"
  type        = string
  default     = ""
}

variable "monthly_budget_usd" {
  description = "Orçamento mensal (USD)"
  type        = number
  default     = 1000
}

variable "storage_retention_days" {
  description = "Dias de retenção do bucket"
  type        = number
  default     = 365
}

variable "storage_delete_after_days" {
  description = "Regra de deleção (lifecycle) após N dias (opcional)"
  type        = number
  default     = 0
}

##############################
# Saídas úteis
##############################
output "https_endpoint_ip" {
  value       = google_compute_global_address.lb_ip.address
  description = "IP público do Load Balancer HTTPS"
}

output "cloud_run_service_url" {
  value       = google_cloud_run_v2_service.api.uri
  description = "URL interna do Cloud Run"
}

output "bucket_datalake" {
  value = google_storage_bucket.data_lake.url
}

output "bigquery_dataset" {
  value = google_bigquery_dataset.analytics.dataset_id
```

}
```

Notas de implantação e hardening
- HSTS/headers de segurança: preferencialmente adicionados na aplicação (Cloud Run) e/ou via CDN/LB header actions (se aplicável).
- mTLS/Pinagem: implementados no app móvel e, se necessário, em camada de Apigee/API Gateway para mTLS gerenciado.
- API Gateway/Endpoints: este baseline expõe Cloud Run via HTTPS LB com Cloud Armor. Caso prefira API Gateway/Endpoints, mantenha Cloud Armor no LB frontal ou use Apigee para WAF nativo + mTLS.
- Binary Authorization: efetivo para GKE; em Cloud Run utilize escaneamento (AR) e políticas de deploy no Cloud Build; só permitir imagens do Artifact Registry confiável.
- VPC Service Controls: requer nível de organização. Planeje perímetro envolvendo BigQuery, GCS, Pub/Sub, Vertex AI e restrições de egress ao perímetro.

Relatório de Modelagem de Ameaças (STRIDE)

Escopo e ativos
- Escopo: apps móveis/câmeras; APIs/serviços em Compute (Cloud Run/GKE/GCE); Edge (HTTPS LB + Cloud Armor + IAP/API Gateway/Endpoints); Pub/Sub→Dataflow→BigQuery/Storage; Vertex AI e Vision API; IAM/KMS/Secret Manager; CI/CD/Artifact Registry/Monitoring/Logging; Rede (VPC, NAT, PGA, VPC SC).
- Ativos: dados de usuários/imagens; datasets e features; modelos/artefatos; credenciais/segredos; chaves KMS; logs/auditoria; pipelines CI/CD; imagens de contêiner.
- Limites de confiança: Dispositivo → Internet → LB/GCP → VPC/serviços gerenciados; fronteiras de identidade (usuário/serviço); dados em trânsito/em repouso.

DFD resumido
1) App envia dados via HTTPS para LB (+Cloud Armor) e API Gateway/Endpoints ou diretamente à API no Compute (protegida por IAP).
2) Serviços em Compute autenticam/autorizam, validam e publicam no Pub/Sub e/ou escrevem no Storage.
3) Dataflow lê do Pub/Sub, transforma/valida e grava no BigQuery/Storage.
4) Vertex AI treina/serve, interagindo com BigQuery/Storage/Compute.
5) Ferramentas de gestão fazem CI/CD, observabilidade; IAM/KMS/Secret Manager e políticas governam acesso e criptografia.

STRIDE
- Spoofing
  - Ameaças: dispositivos falsos; roubo de tokens; impersonação de serviços.
  - Controles: OAuth2/OIDC + IAP; Play Integrity/DeviceCheck; mTLS e pinagem de certificado; validação de JWT no gateway/Endpoints; Workload Identity Federation; proibir chaves externas de SAs.
- Tampering
  - Ameaças: payloads malformados; imagens contaminadas; alterações em IaC.
  - Controles: TLS1.2+/HMAC/validação de esquema e tamanho; sanitização; Cloud Armor (WAF/OWASP); antivirus/antimalware de conteúdo; SLSA + provenance; Binary Authorization (GKE); Object Versioning/Retention; revisão de IaC.
- Repudiation
  - Ameaças: falta de trilha; alteração de logs.
  - Controles: Audit Logs (Admin/Data/Access) habilitados; sinks para BigQuery; retenção imutável (Lock); sincronização de tempo; aprovações em CI/CD.
- Information Disclosure
  - Ameaças: buckets públicos; acesso indevido a BigQuery; vazamento por inferência.
  - Controles: IAM mínimo + Conditions; UBLE + publicAccessPrevention; VPC Service Controls; DLP/masking; CMEK/rotação; segregação por projeto/ambiente; RLS/CLS em BQ.
- Denial of Service
  - Ameaças: floods HTTP; saturação Pub/Sub; jobs/consultas custosas.
  - Controles: Cloud Armor (rate/geo/IP/WAF); quotas/limites por cliente; backpressure/retry no Dataflow; DLQ; autoscaling com limites; budgets/alertas.
- Elevation of Privilege
  - Ameaças: exploração em containers/VM; papéis excessivos; metadata server.
  - Controles: hardening (Shielded VMs, GKE Sandbox); desabilitar metadados legados; papéis mínimos/custom; SCC + alertas; patching contínuo.

Vulnerabilidades e respectivas correções (mapeamento prático)
1) Tráfego app→nuvem: TLS 1.2+ via LB, HSTS na app/LB; mTLS opcional; pinagem no app.
2) Autenticação fraca/chaves no app: OIDC/IAP; tokens curtos; Play Integrity; rotação; remover API
```

Keys estáticas.
3) Exposição de buckets/BQ: UBLE + publicAccessPrevention; IAM mínimo; VPC SC; governança (Data Catalog).
4) Tampering/injeções: validação de esquema/tamanho; HMAC; sanitização; WAF OWASP; varredura de malware.
5) Supply chain: SLSA nível 2+; Cloud Build provenance; AR scanning; Binary Authorization (GKE/Cloud Run policies).
6) Segredos em código: Secret Manager; remoção de chaves estáticas; WIF; rotação/KMS; secret scanning no CI.
7) Privilégios excessivos: papéis custom/mínimos; IAM Conditions; bloquear chaves externas SA; segregação por ambiente.
8) DoS/flood: Cloud Armor rate/geo; quotas; backpressure/ DLQ; autoscaling com limites; CDN/cache se aplicável.
9) Exfiltração por egress: VPC SC; Private Google Access; egress allowlist; DNS policy; DLP/alertas.
10) Poisoning/modelo: versionar/validar datasets; canary/rollback; drift detection; avaliar membership inference; limitar exposição sensível.
11) Integridade de objetos/logs: retention/lock; versioning; assinaturas (Tink/KMS); sinks de logs para BQ/SIEM.
12) Deriva de configuração: IaC (Terraform) + revisão; Policy as Code (Config Validator/OPA); SCC monitorando.

Riscos residuais e plano
- Supply chain residual: manter verificação de provenance e BinAuthZ obrigatório (ambientes críticos); auditoria trimestral de pipelines e imagens.
- Drift de dados/modelos: monitorar qualidade e drift; acionadores de re-treino; revisão humana antes de promoção.
- Custo/DoS lógico: budgets com thresholds/alerta; rate limiting e quotas por cliente; tuning de autoscaling.

Métricas/SLOs de segurança
- Cobertura de logs 100% (Admin/Data/Access) em produção.
- 0 segredos em repositórios; rotação de segredos ≤ 90 dias.
- Imagens sem attestation bloqueadas: 100% em produção (GKE/BinAuthZ).
- Nenhum bucket público permitido por Org Policy.

Planos de resposta a incidentes
- Playbooks: vazamento de dados; comprometimento de chaves; falhas de modelo/deriva.
- Exportação de logs para SIEM e exercícios de simulação semestrais (tabletop).
- Runbooks para isolamento de perímetro (VPC SC), revogação/rotação de chaves KMS/segredos, e bloqueio de endpoints no Cloud Armor.

Recursos adicionais necessários
- Network Administrator: VPC, firewall, DNS, conectividade privada e VPC SC.
- Security Officer: políticas, classificação de dados, aceitação de risco e IR.
- Cloud/DevOps: CI/CD, SLSA/BinAuthZ, IaC/Policy, observabilidade/custos.
- Software Developer: desenho de APIs, segurança, validação de payloads, versionamento/testes.
- Data/ML Engineer: pipelines, governança (DLP/BQ), Vertex AI, monitoramento de drift.
- Hardware/Mobile Developer: segurança do app/dispositivo, proteção de credenciais, atestação/anti-tampering.
- IAM/KMS Admin: papéis mínimos, chaves CMEK, rotação e segregação por ambiente.
- Compliance/Legal: LGPD, retenção e requisitos contratuais.