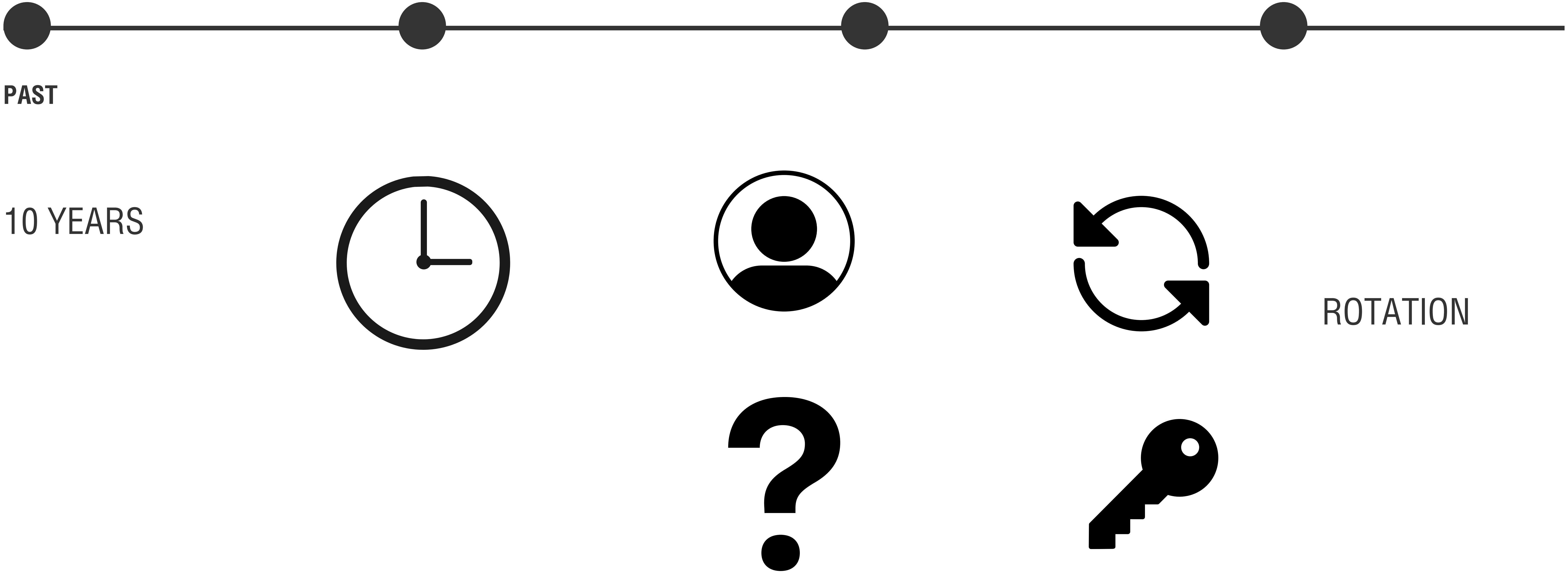
# WORKLOAD IDENTITY FEDERATION- GCP

**Laysa Uchoa**

# SERVICE ACCOUNT KEYS AS SECRETS

PAST

# SERVICE ACCOUNT KEYS AS SECRETS __



PAST

10 YEARS

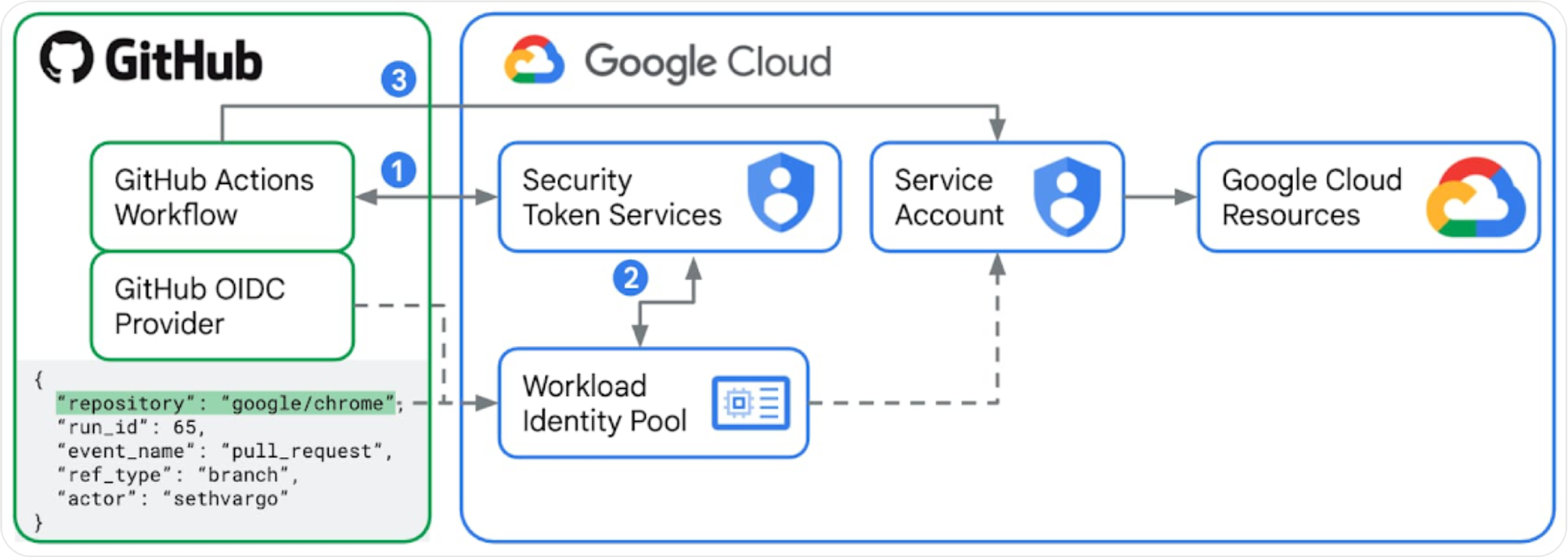
ROTATION

# WORKLOAD IDENTITY

NO ROTATION NEEDED

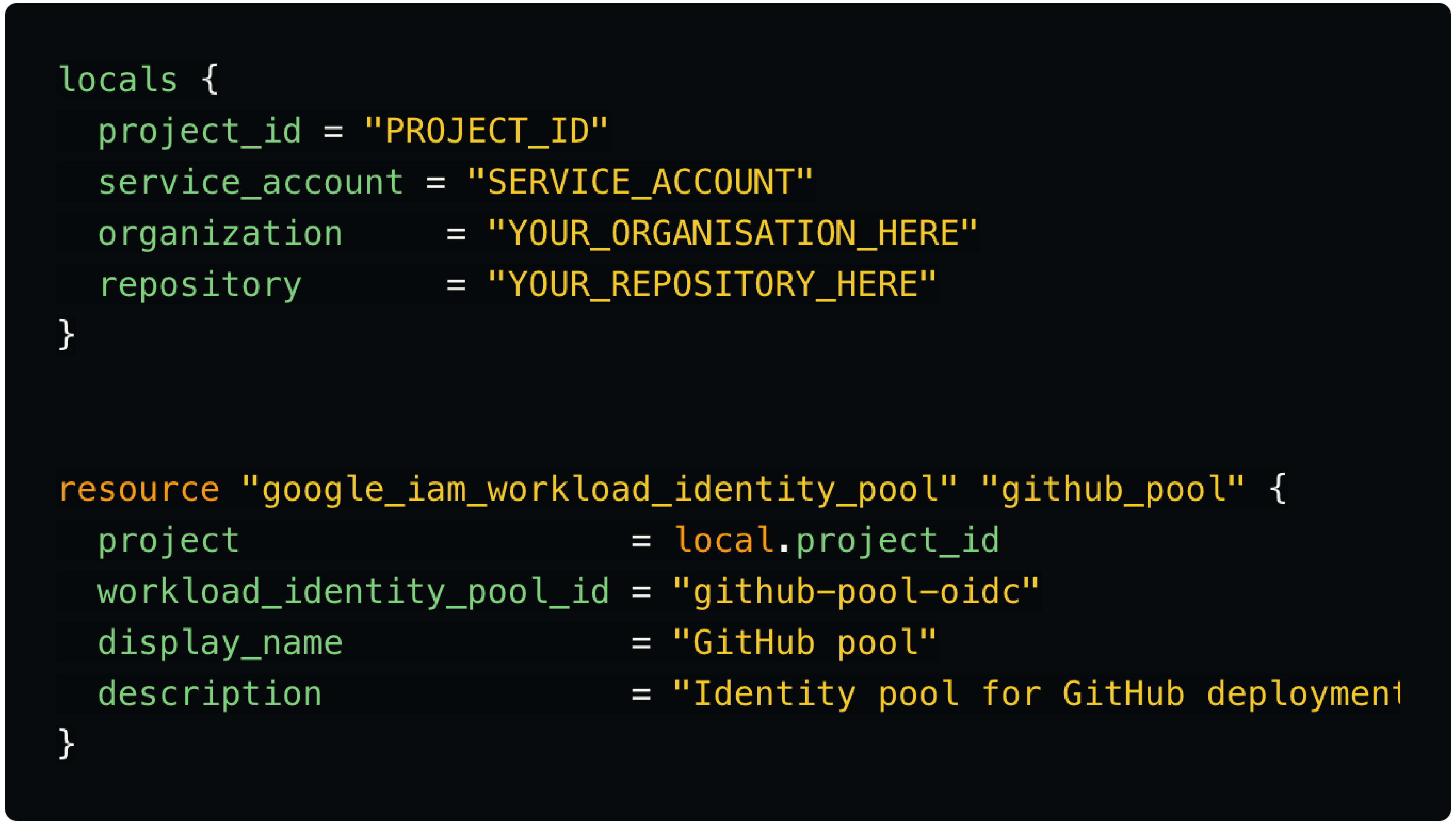
PRINCIPLE OF LEAST PRIVILEGE

# ARCHITECTURE __



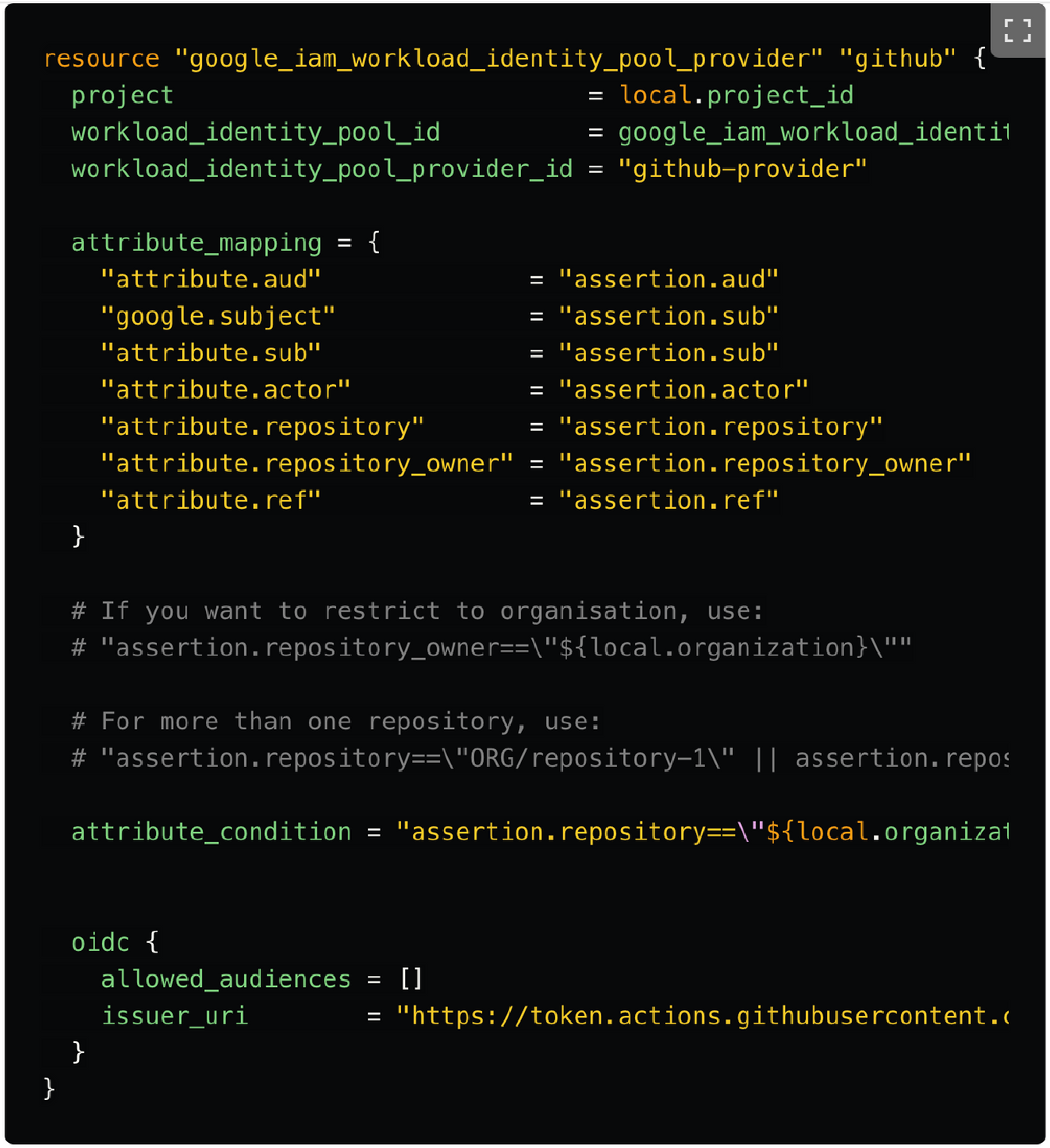
```
{
  “repository”: “google/chrome”;
  “run_id”: 65,
  “event_name”: “pull_request”,
  “ref_type”: “branch”,
  “actor”: “sethvargo”
}
```

# CREATE A WORKLOAD IDENTITY POOL



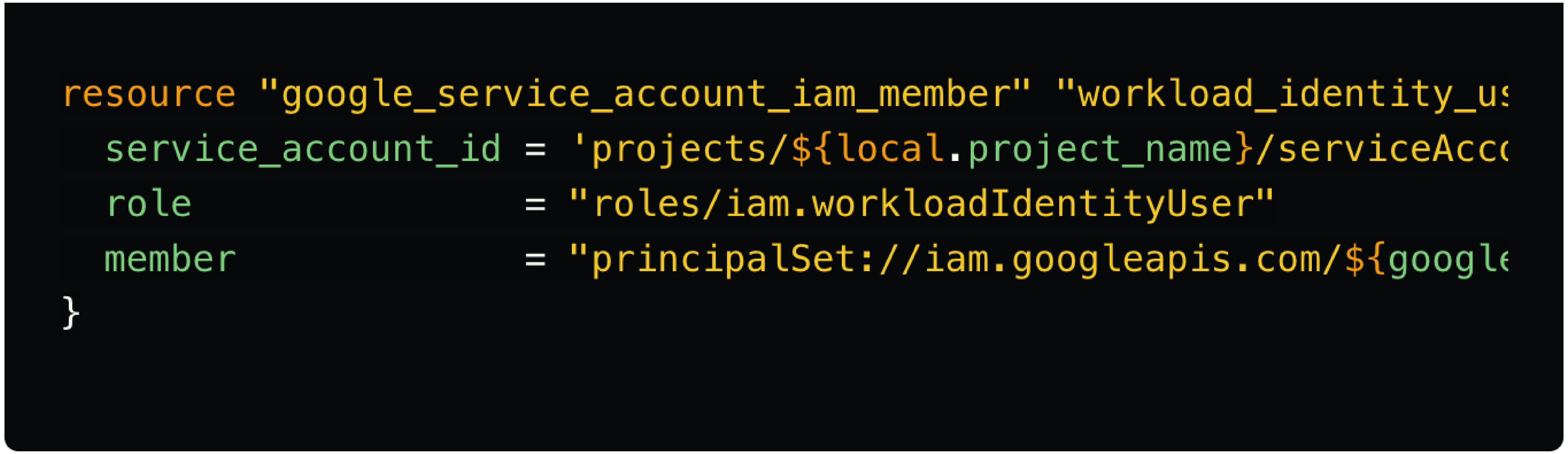
```
locals {
  project_id = "PROJECT_ID"
  service_account = "SERVICE_ACCOUNT"
  organization      = "YOUR_ORGANISATION_HERE"
  repository        = "YOUR_REPOSITORY_HERE"
}


resource "google_iam_workload_identity_pool" "github_pool" {
  project                   = local.project_id
  workload_identity_pool_id = "github-pool-oidc"
  display_name              = "GitHub pool"
  description               = "Identity pool for GitHub deployment
}
```

# CONFIGURE THE WORKLOAD IDENTITY POOL PROVIDER



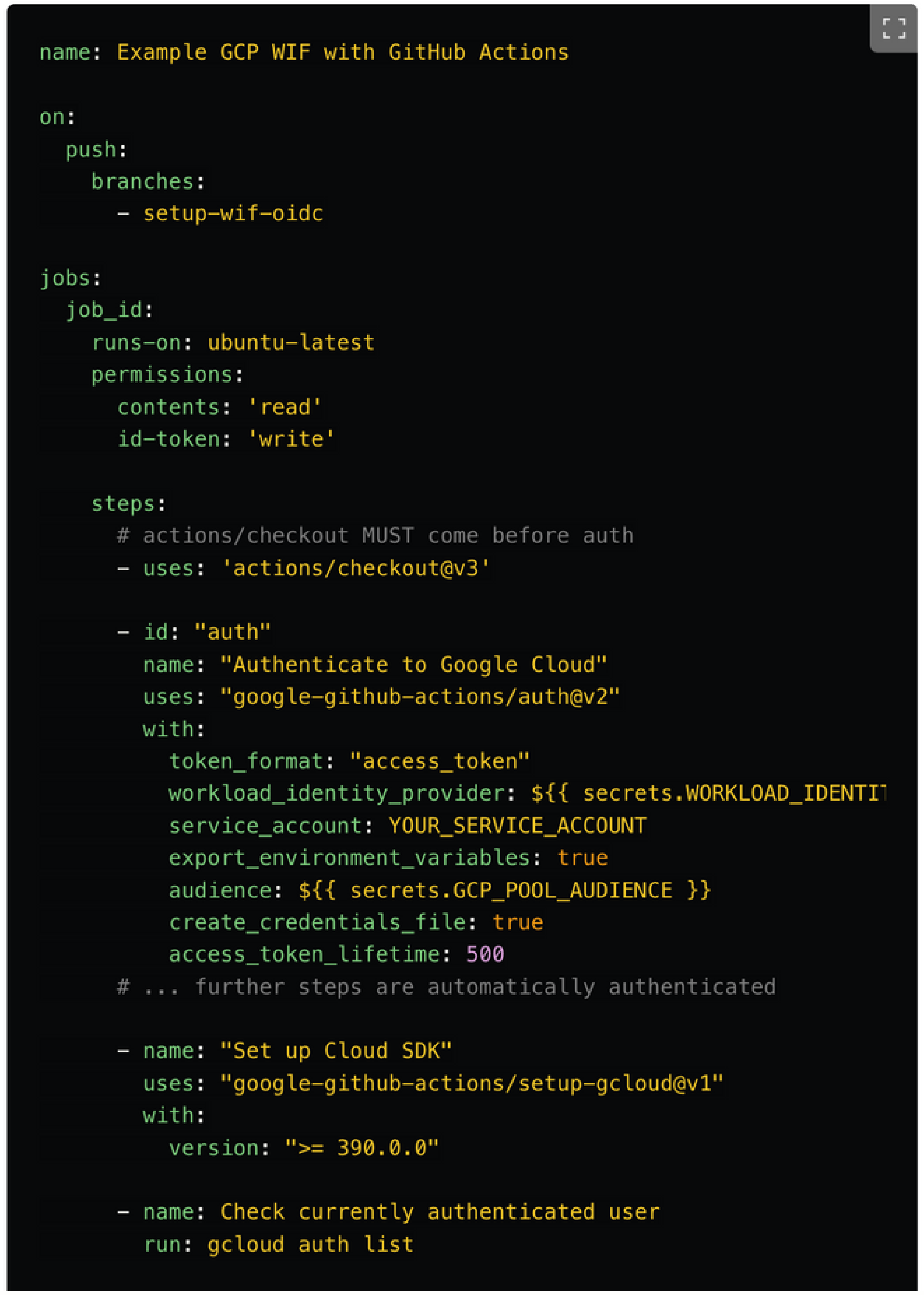
```
resource "google_iam_workload_identity_pool_provider" "github" {
  project                            = local.project_id
  workload_identity_pool_id          = google_iam_workload_identi
  workload_identity_pool_provider_id = "github-provider"

  attribute_mapping = {
    "attribute.aud"              = "assertion.aud"
    "google.subject"             = "assertion.sub"
    "attribute.sub"              = "assertion.sub"
    "attribute.actor"            = "assertion.actor"
    "attribute.repository"       = "assertion.repository"
    "attribute.repository_owner" = "assertion.repository_owner"
    "attribute.ref"              = "assertion.ref"
  }

  # If you want to restrict to organisation, use:
  # "assertion.repository_owner==\"${local.organization}\""

  # For more than one repository, use:
  # "assertion.repository==\"ORG/repository-1\" || assertion.repos

  attribute_condition = "assertion.repository==\"${local.organiza


  oidc {
    allowed_audiences = []
    issuer_uri        = "https://token.actions.githubusercontent.
  }
}
```

# CONFIGURE IAM ROLES



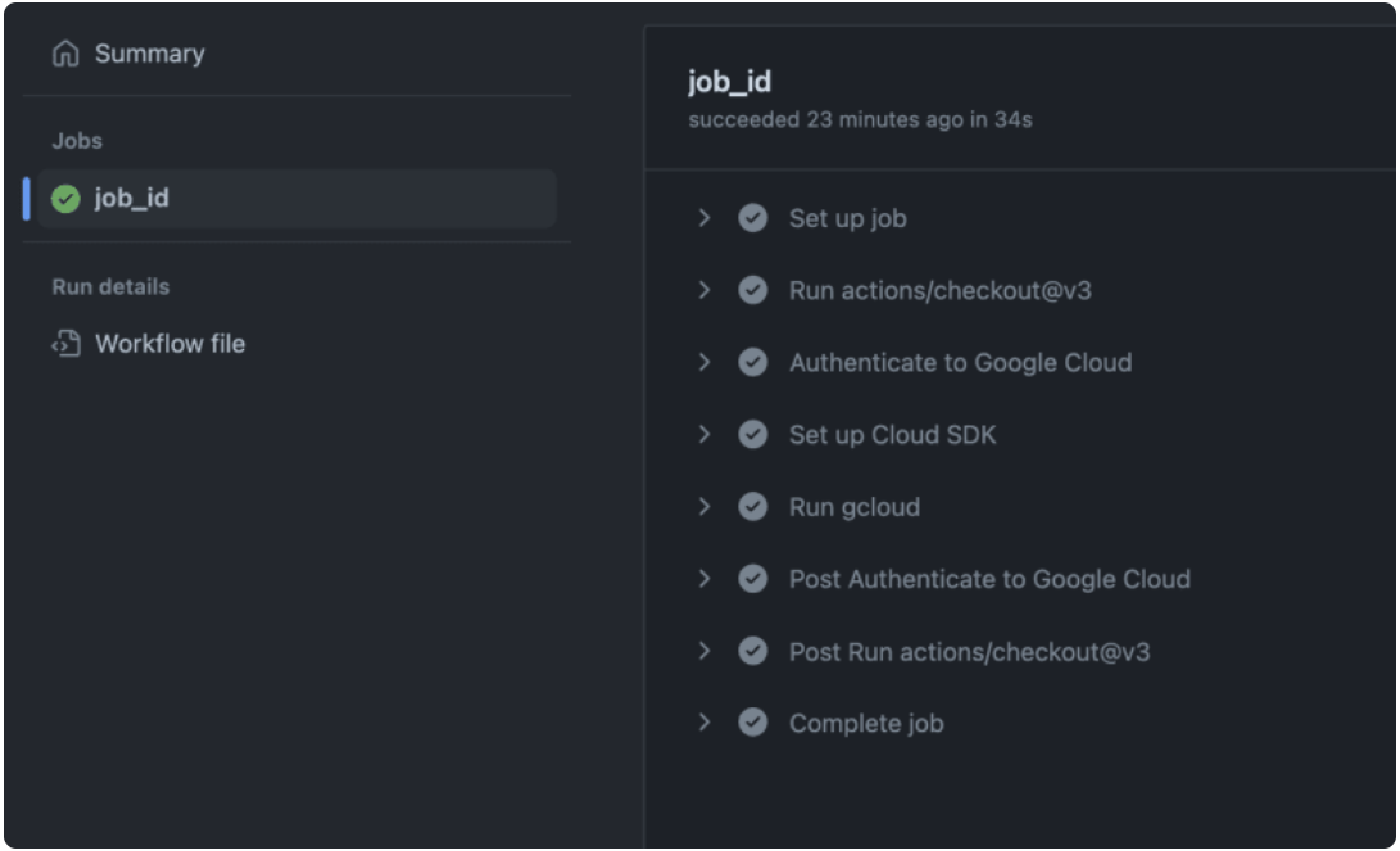
```
resource "google_service_account_iam_member" "workload_identity_us
  service_account_id = 'projects/${local.project_name}/serviceAcco
  role               = "roles/iam.workloadIdentityUser"
  member             = "principalSet://iam.googleapis.com/${google
}
```

# GITHUB SETUP



```yaml
name: Example GCP WIF with GitHub Actions

on:
  push:
    branches:
      - setup-wif-oidc

jobs:
  job_id:
    runs-on: ubuntu-latest
    permissions:
      contents: 'read'
      id-token: 'write'

    steps:
      # actions/checkout MUST come before auth
      - uses: 'actions/checkout@v3'

      - id: "auth"
        name: "Authenticate to Google Cloud"
        uses: "google-github-actions/auth@v2"
        with:
          token_format: "access_token"
          workload_identity_provider: ${{ secrets.WORKLOAD_IDENTIT
          service_account: YOUR_SERVICE_ACCOUNT
          export_environment_variables: true
          audience: ${{ secrets.GCP_POOL_AUDIENCE }}
          create_credentials_file: true
          access_token_lifetime: 500
      # ... further steps are automatically authenticated

      - name: "Set up Cloud SDK"
        uses: "google-github-actions/setup-gcloud@v1"
        with:
          version: ">= 390.0.0"

      - name: Check currently authenticated user
        run: gcloud auth list
```

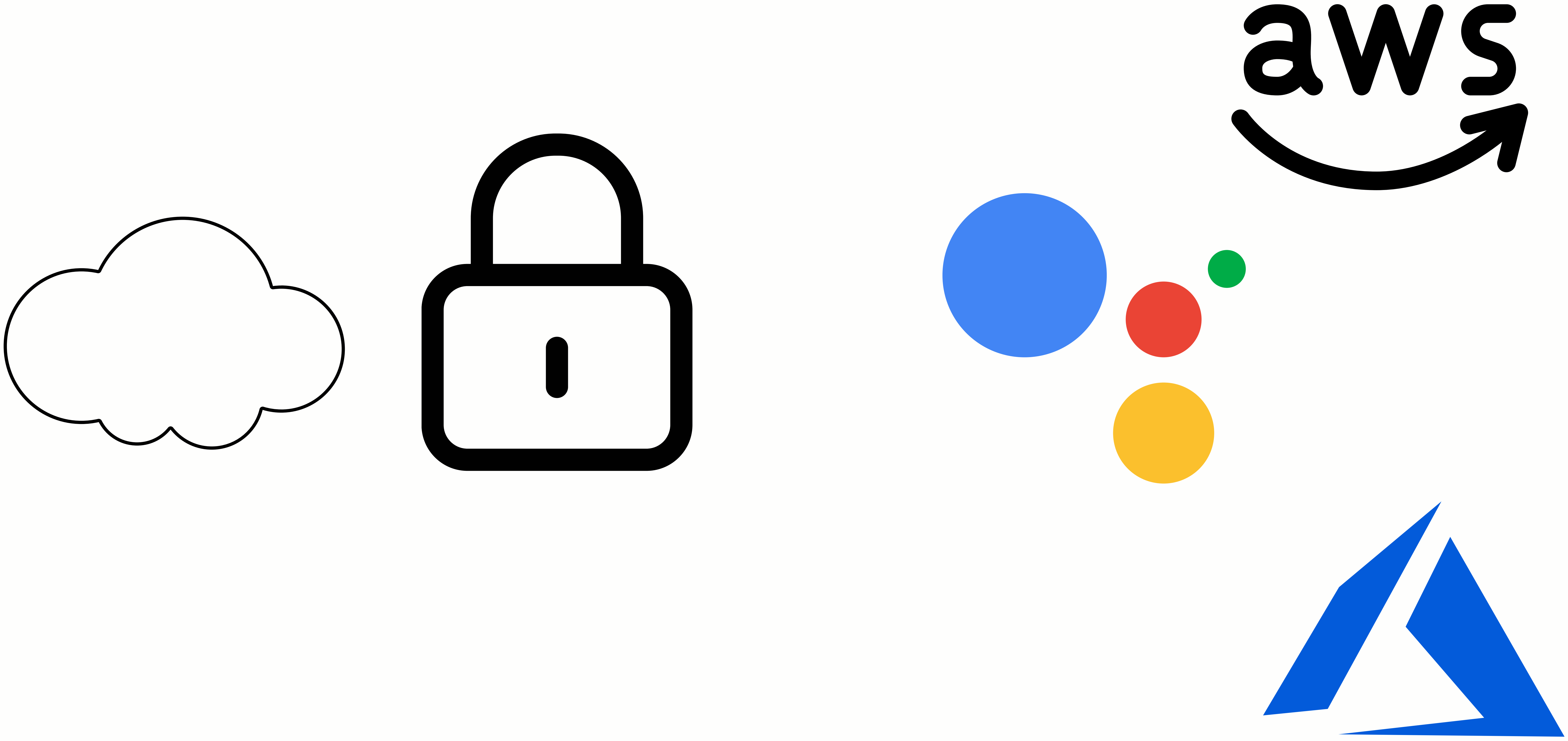
# RUN PIPELINE __



job_id

succeeded 23 minutes ago in 34s

- ✓ Set up job
- ✓ Run actions/checkout@v3
- ✓ Authenticate to Google Cloud
- ✓ Set up Cloud SDK
- ✓ Run gcloud
- ✓ Post Authenticate to Google Cloud
- ✓ Post Run actions/checkout@v3
- ✓ Complete job


aws

# THANK YOU!

END _

**Laysa Uchoa**