



PLATFORM ENGINEERING

Infrastructure as Code & Modern Secrets Management

Benefits of IaC • Pulumi with Go • External Secrets Operator • Security Best Practices

Joshua Haupt
jnh.dev
joshua@hauptj.com

A technical guide to declarative infrastructure and secure credential handling

Six benefits that change how teams ship infrastructure



Consistency & Repeatability

Identical environments every time — eliminate configuration drift between dev, staging, and prod.



Version Controlled

Infrastructure lives in Git: peer-reviewed, auditable, and easy to roll back to any prior state.



Speed & Automation

Provision full stacks in minutes and wire changes directly into CI/CD pipelines.



Scalability

Replicate infrastructure across regions and accounts with parameters, not copy-paste.



Disaster Recovery

Rebuild an entire environment from source, dramatically shrinking recovery time.



Cost Efficiency

Tear down unused resources on demand and remove costly manual provisioning errors.

Pulumi: Infrastructure in real programming languages



Real languages

Author infra in Go, Python, TypeScript, C#, or Java — with loops, functions, and unit tests.



Multi-cloud, one workflow

AWS, Azure, GCP, Kubernetes and 150+ providers behind a single consistent SDK.



Managed state

State stored in Pulumi Cloud or a self-managed backend (S3, Azure Blob, GCS).



Desired-state engine

You describe the outcome; Pulumi diffs, plans, and reconciles to it.

AT A GLANCE

5+

supported languages

150+

cloud & SaaS providers

1

unified deployment workflow

pulumi up — *preview, then deploy*

A program, not a template

```
main.go

package main

import (
    "github.com/pulumi/pulumi-aws/sdk/v6/go/aws/s3"
    "github.com/pulumi/pulumi/sdk/v3/go/pulumi"
)

func main() {
    pulumi.Run(func(ctx *pulumi.Context) error {
        // Provision an S3 bucket
        bucket, err := s3.NewBucketV2(ctx, "app-data",
            &s3.BucketV2Args{
                Tags: pulumi.StringMap{
                    "Env": pulumi.String("prod"),
                },
            })
        if err != nil {
            return err
        }
        ctx.Export("bucketName", bucket.ID())
        return nil
    })
}
```

Strongly typed

Resource args are Go structs — the compiler catches mistakes early.

Real error handling

Standard Go err checks; no bespoke DSL to learn.

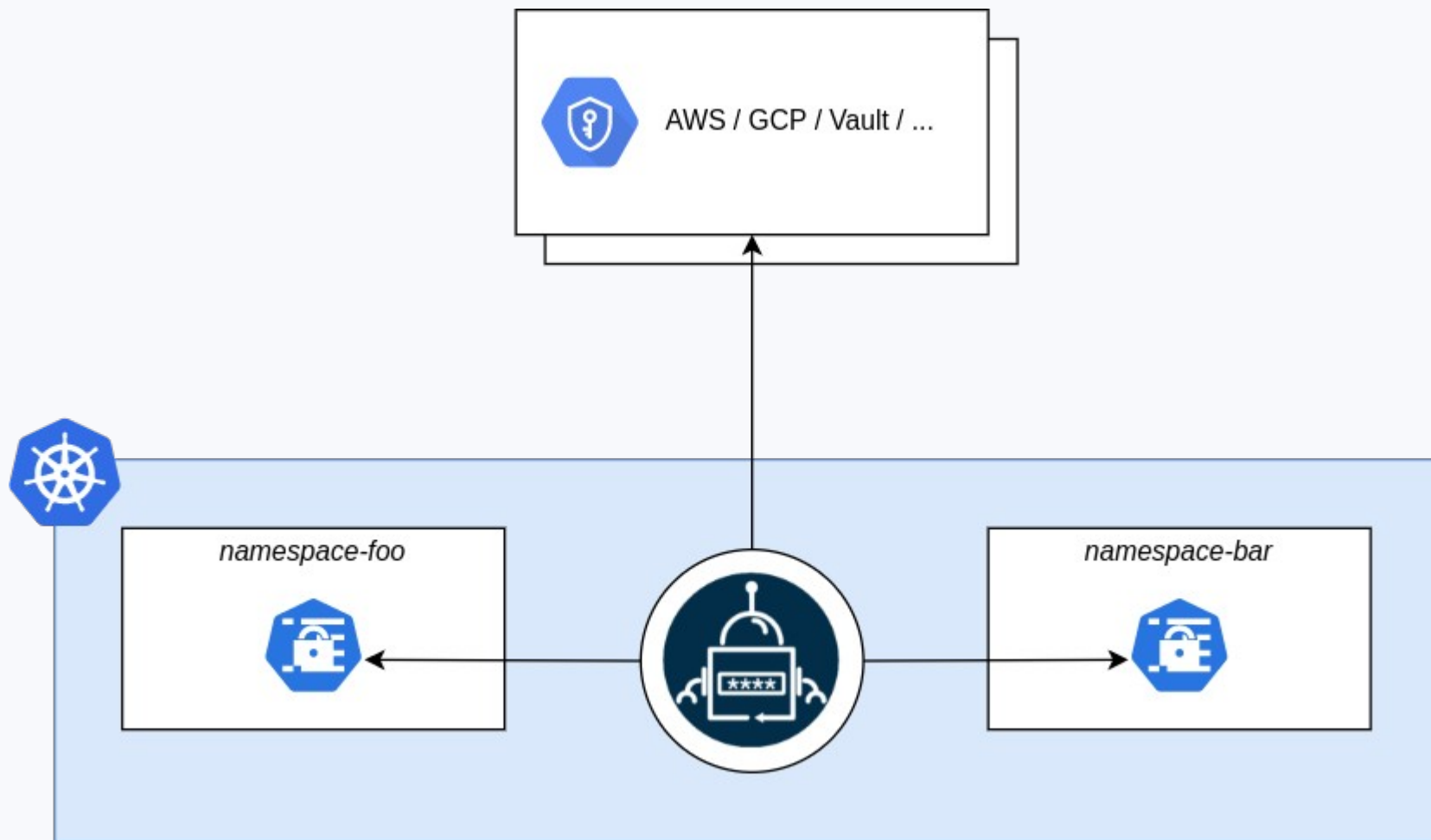
Stack outputs

ctx.Export surfaces values for other stacks or pipelines.

Pulumi vs. Terraform

	 Pulumi	 Terraform
Language	Go, Python, TypeScript, C#, Java, YAML	HCL — a purpose-built DSL
License	Apache 2.0 — fully open source	BSL 1.1 — source-available
Logic & reuse	Loops, functions, classes, packages	count, for_each, modules
Testing	Host-language unit tests (pytest, Go)	terraform test, Terratest
State	Pulumi Cloud or self-managed backends	State files + remote / HCP
Ecosystem	Newer; can bridge Terraform providers	Largest & most mature (2014)
<div>Pick Pulumi for real languages, in-code logic, and unit testing.</div>		<div>Pick Terraform for the broadest ecosystem and HCL simplicity.</div>

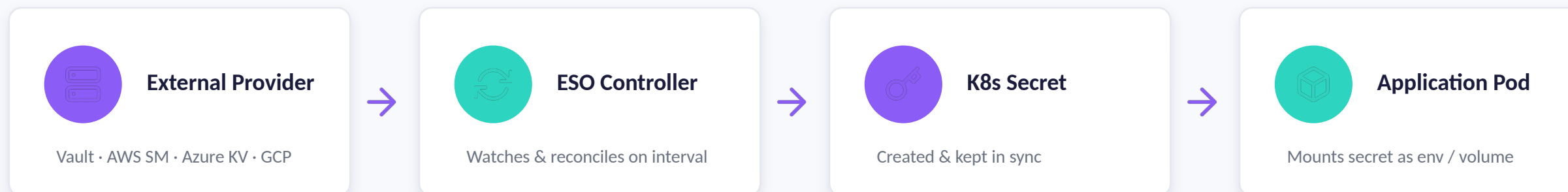
Note: Terraform is IBM-owned (2025); prefer open-source HCL? **OpenTofu** is its MPL-2.0, Linux Foundation fork.



External Secrets Operator

External Secrets Operator (ESO)

A Kubernetes operator that syncs secrets from external providers into native K8s Secrets — so credentials never live in your manifests or Git.



SecretStore

Defines how to authenticate to one external provider (namespaced), or ClusterSecretStore cluster-wide.

ExternalSecret

Declares which remote keys to fetch and how to map them into a target Kubernetes Secret.

Auto-refresh

Re-syncs on a configurable interval, so rotated upstream secrets propagate automatically.

The ExternalSecret resource

externalsecret.yaml

```
apiVersion: external-secrets.io/v1
kind: ExternalSecret
metadata:
  name: db-credentials
spec:
  refreshInterval: 1h
  secretStoreRef:
    name: vault-backend
    kind: SecretStore
  target:
    name: db-secret
    creationPolicy: Owner
  data:
    - secretKey: password
      remoteRef:
        key: prod/db
        property: password
  dataFrom:
    - extract:
        key: prod/db/config
```

secretStoreRef

Selects the SecretStore or ClusterSecretStore to authenticate against.

refreshInterval

How often ESO re-reads the provider; set 0 to sync only once.

target

The Secret to manage; creationPolicy is Owner, Merge, or None.

data / dataFrom

Map individual keys, or bulk-extract every key found at a path.

Custom Resource Definitions ESO provides

Core kinds live in the [external-secrets.io](#) group; a second [generators.external-secrets.io](#) group mints dynamic values.



ExternalSecret

Namespaced

Declares which secrets to fetch and how to assemble the target Kubernetes Secret.



SecretStore

Namespaced

Defines how to authenticate and connect to a single external provider.



ClusterSecretStore

Cluster-scoped

A shared provider connection that any namespace can reference.



ClusterExternalSecret

Cluster-scoped

Templates ExternalSecrets across many namespaces via a namespace selector.



PushSecret

Namespaced

Reverses the flow — pushes Secret values back to a provider (also ClusterPushSecret).



Generators

`generators.external-secrets.io`

Dynamically mint values: Password, UUID, VaultDynamicSecret, ECR tokens, and more.

Security best practices for secrets



Never commit secrets

Keep credentials out of Git; enforce with .gitignore and automated secret scanning in CI.



Encrypt everywhere

Encrypt secrets at rest and in transit (TLS); prefer envelope encryption with a managed KMS.



Least privilege

Scope IAM and RBAC narrowly — grant each workload only the exact secrets it needs.



Rotate & expire

Rotate regularly and favor short-lived, dynamically issued credentials over static ones.



Centralize management

Use a dedicated secrets manager (Vault, AWS SM) as the single source of truth.



Audit & monitor

Log every access, alert on anomalies, and keep secrets out of application logs.

KEY TAKEAWAYS

Codify infrastructure, externalize secrets



laC brings consistency, auditability, and speed to every environment.



Pulumi lets you express infrastructure in Go with real types and tests.



ESO keeps secrets in a managed store and syncs them into Kubernetes.



Treat secrets as first-class: encrypt, scope, rotate, centralize, and audit.



Declarative infrastructure + **disciplined secrets management** = *resilient, secure platforms.*

TRY IT YOURSELF

Explore the demo repo

Everything from these slides, assembled into one runnable project you can clone, deploy, and adapt.



A Pulumi program in Go that provisions the Kubernetes resources



External Secrets Operator wired to a store and synced into the cluster



Manifests and ExternalSecret examples ready to apply



<https://github.com/HauptJ/Pulumi-Go-K8s-ESO-Demo>



Scan to open

github.com/HauptJ/Pulumi-Go-K8s-ESO-Demo



References

1. External Secrets Operator - <https://external-secrets.io/latest/>
2. Anthropic Claude - <https://claude.ai/>