HashiTalks 2022

Tracy Holmes

Exploring HashiCorp Vault and ArgoCD the GitOps Way

February 17-18

hashi.co/hashitalks-2022

GitOps & Argo CD

What are these things?!

GitOps

https://opengitops.dev

- The system is described in a declarative manner.
- The definition of the system is versioned and audited.
- A software agent automatically pulls the Git state and matches the platform state.
- The state is continuously reconciled.

GitOps

A few benefits…

- Deploying faster and more often,
- Easier and quicker error handling and recovery
- Self-documenting deployments.
- Elimination of configuration drift.

Argo CD

https://argoproj.github.io/cd/

What & Why

Argo CD is a declarative, GitOps continuous delivery tool for Kubernetes.

Application definitions, configurations, and environments should be declarative and version controlled. Application deployment and lifecycle management should be automated, auditable, and easy to understand.

Secrets.

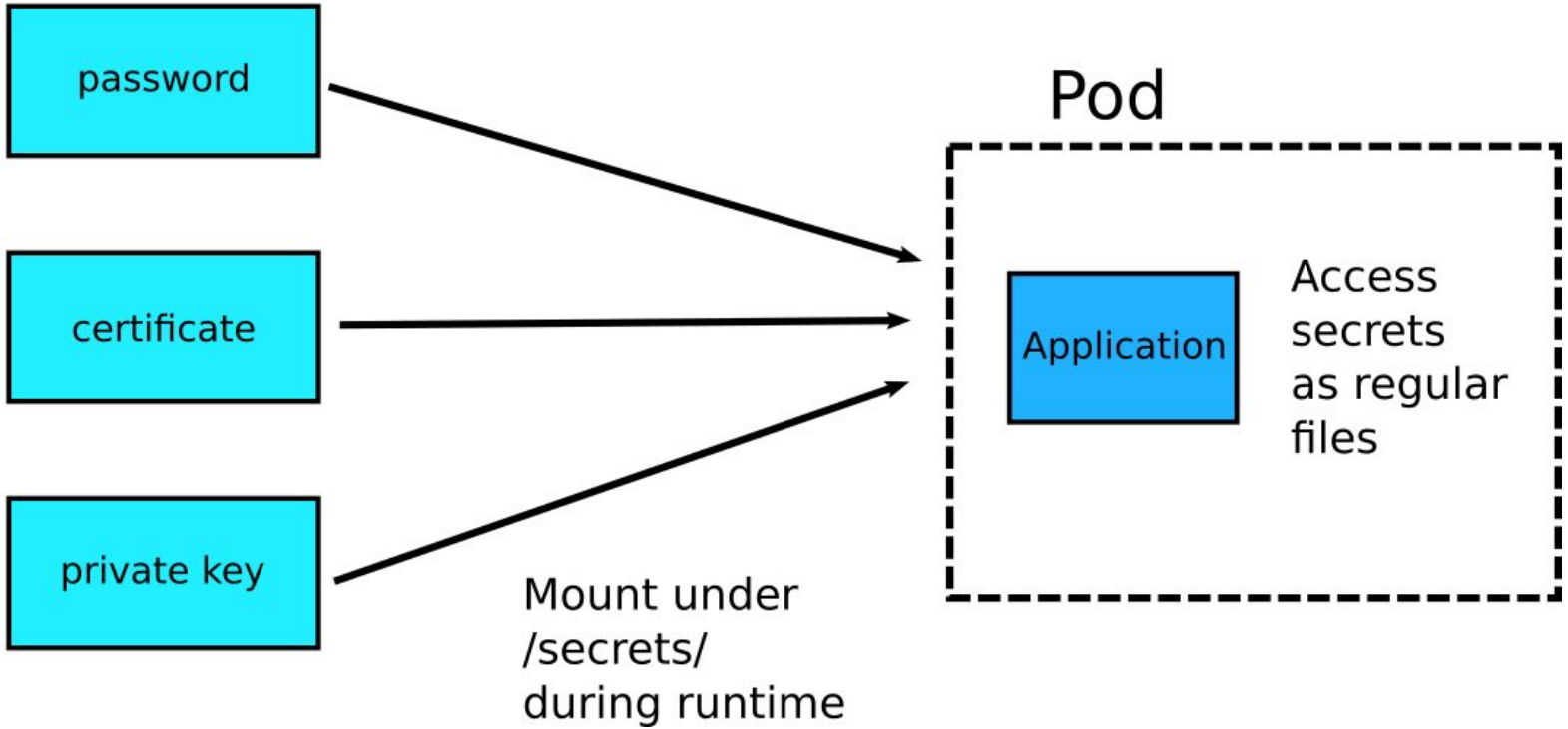
shhhhhhh…🤫

Why are secrets important?

…and can I use them with GitOps

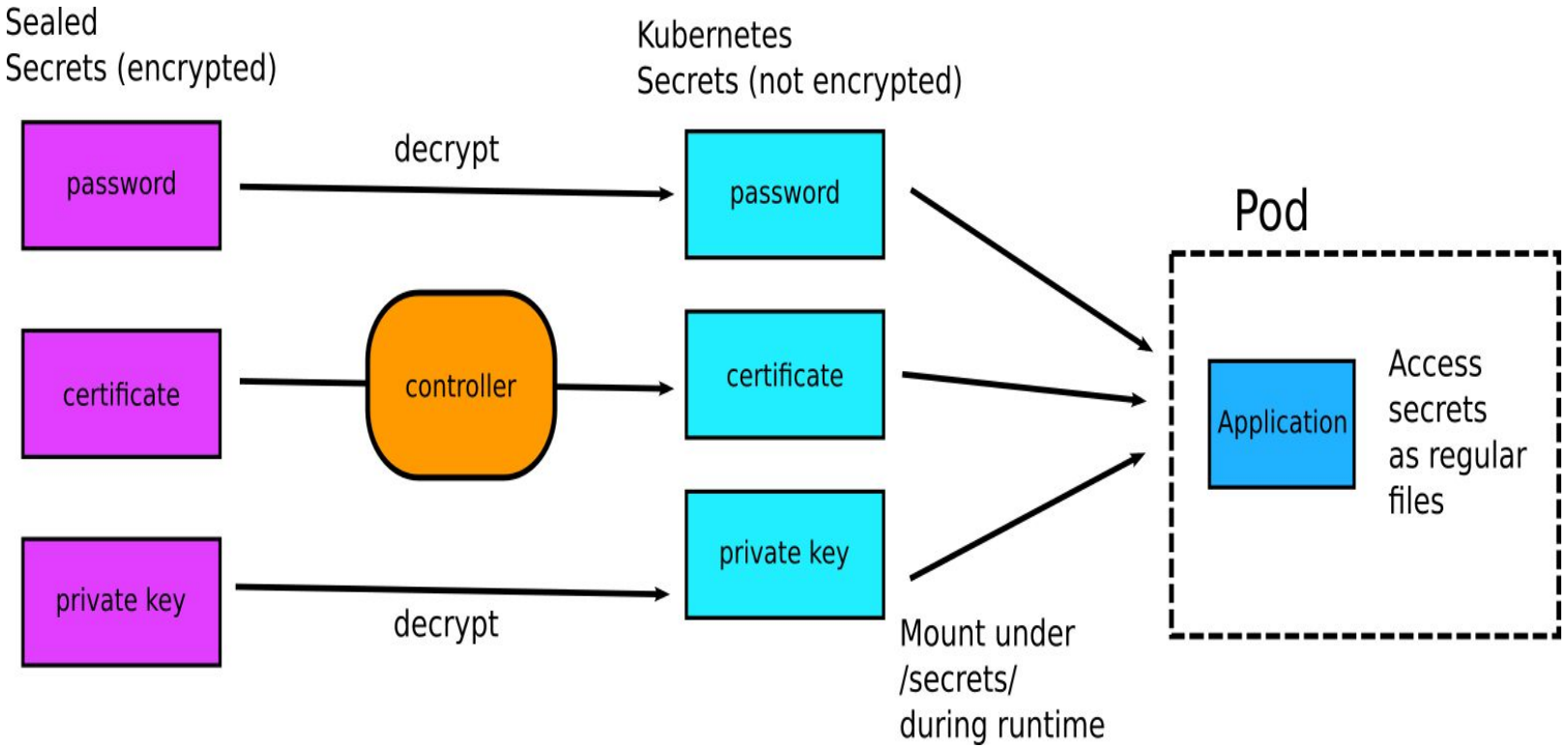
Tools I hear about

Kubernetes secrets (not encrypted)



Tools I hear about

Bitnami Sealed Secrets



Disadvantages

Vault.

Vault and GitOps.

What makes Vault not GitOps friendly?

Thoughts.

Vault is actually a controller.

All dynamic secrets in Vault are required to have a lease. A lease is required to force the consumer to check in routinely. So, it continuously checks the leases.

And you can configure it declaratively using the CLI.

Vault and Version Control

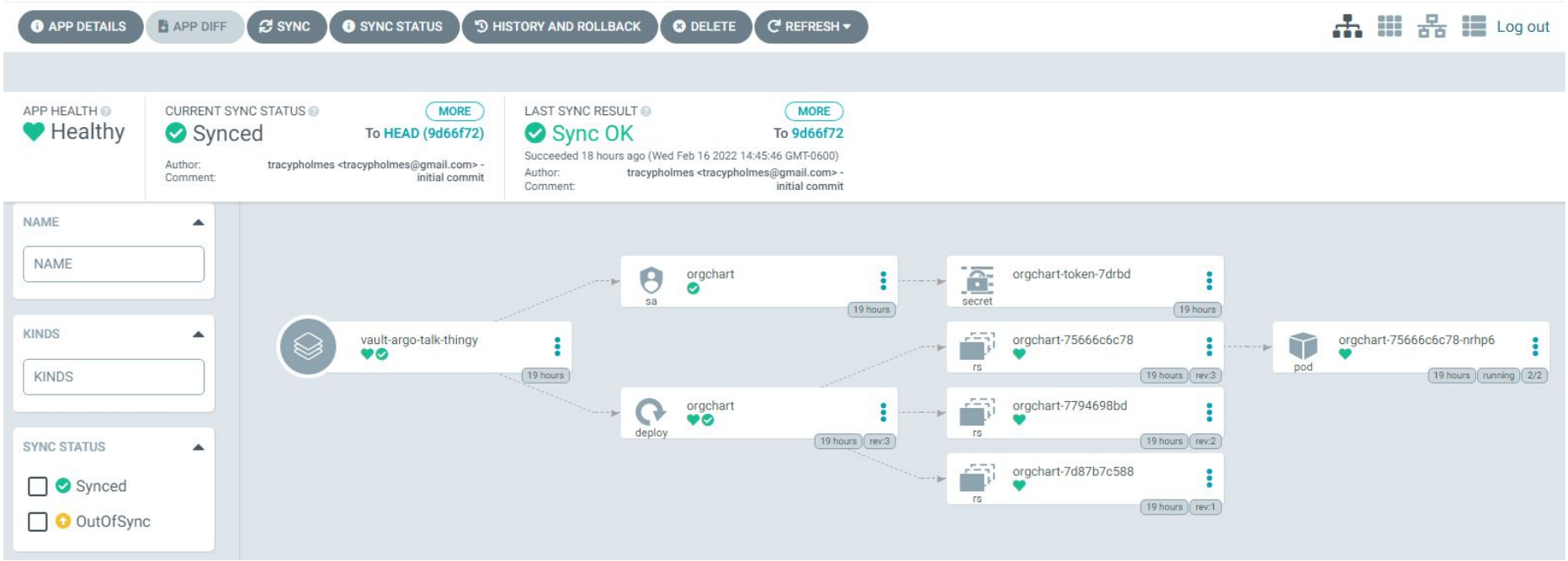
…and can I use them with GitOps

- **State Store**

A system for storing immutable versions of desired state declarations. This state store should provide access control and auditing on the changes to the Desired State. Git, from which GitOps derives its name, is the canonical example used as this state store but any other system that meets these criteria may be used. In all cases, these state stores must be properly configured and precautions must be taken to comply with requirements set out in the GitOps Principles.

https://github.com/open-gitops/documents/blob/v1.0.0/GLOSSARY.md#state-store

Vault and Kubernetes



```
template:
    metadata:
      annotations:
        vault.hashicorp.com/agent-inject: 'true'
        vault.hashicorp.com/role: 'orgchart'
        vault.hashicorp.com/agent-inject-secret-env: 'orgchart/data/database/config'
        vault.hashicorp.com/agent-inject-template-env: |
          {{- with secret "orgchart/data/database/config" -}}
          export DB_USERNAME={{ .Data.data.username }}
          export DB_PASSWORD={{ .Data.data.password }}
          {{- end }}
      labels:
        app: orgchart
```

Change…is good?

- Argo CD Vault plugin (works with Kubernetes secrets)
- Secrets Store CSI Driver (and the Vault provider for it)
- …something else out there I haven't had a chance to research

Sidenote

(and a neutral one at that!)

- Part of GitOps says configuration must be declarative. “you can’t configure Vault declaratively”. You can use Terraform for your configurations, OR the Vault Config Operator (which is a community tool)
- If you want to see what Vault Config Operator looks like in practice, check out this repo.
- It ALSO uses Argo Sync Wave to sync things that need to reconcile (seen here)

Why are secrets important?

…and can I use them with GitOps

Thank You

@tracypholmes
hugs@hashicorp.com | learn.hashicorp.com | discuss.hashicorp.com