incl. **Demo** 🚀

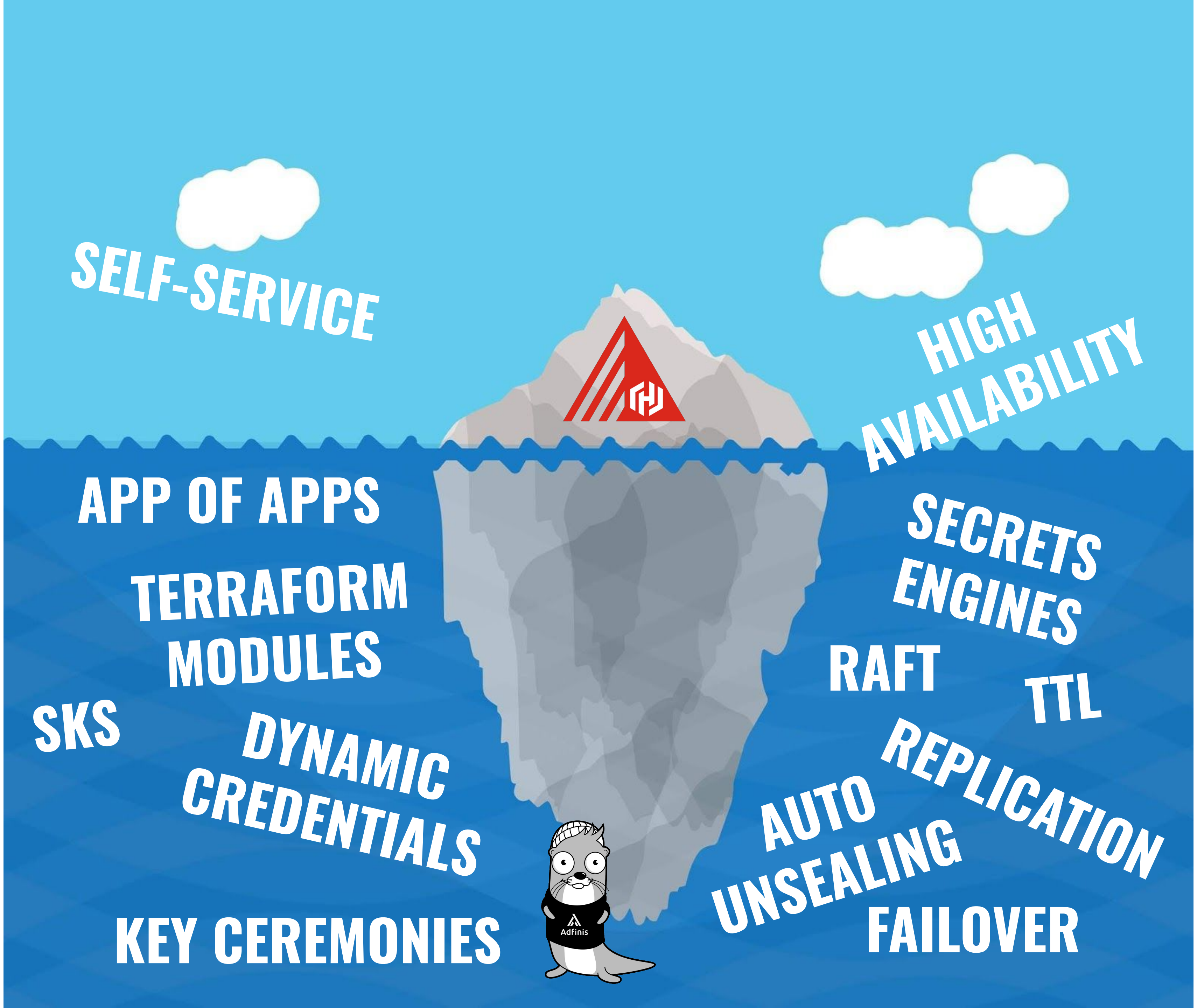
HashiCorp Vault Enterprise on Exoscale

HashiCorp User Group (HUG) Meetup
08.12.2022

Adfinis

Plan. Innovatively | **Build.** Sustainably | **Run.** Resiliently

Potential. Unlocked



SELF-SERVICE

HIGH AVAILABILITY

APP OF APPS

TERRAFORM MODULES

SKS

DYNAMIC CREDENTIALS

KEY CEREMONIES

SECRETS ENGINES

RAFT

TTL

REPLICATION

AUTO UNSEALING

FAILOVER

Andreas Gruhler

System Engineer

Adfinis

Zurich, Switzerland

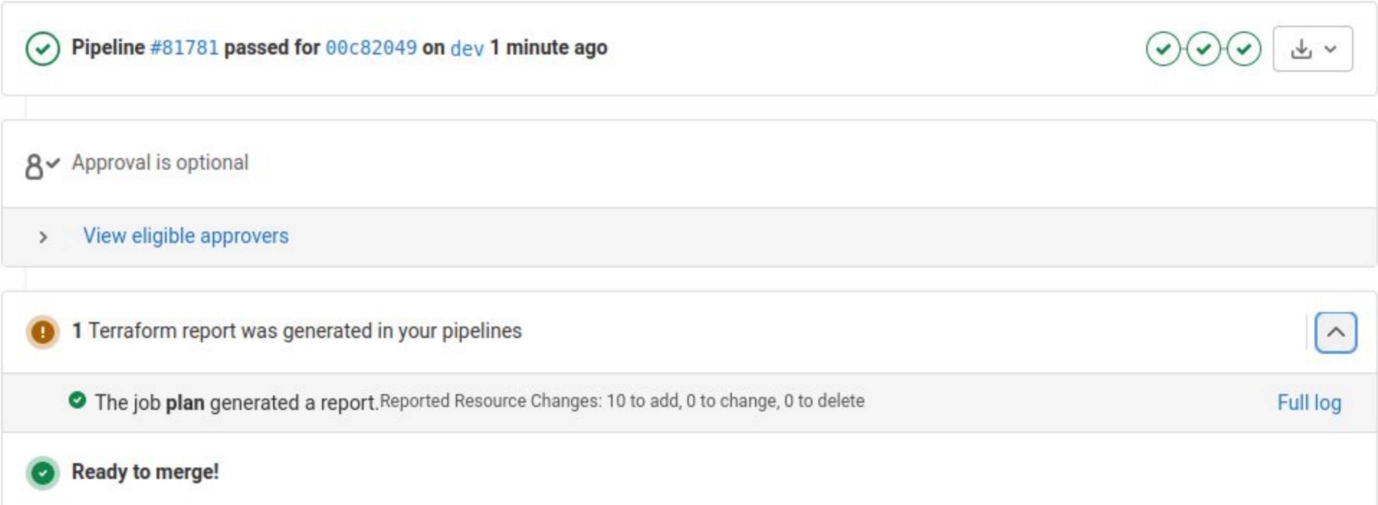
andreas.gruhler@adfinis.com

github.com/in0rdr

Terraform in GitLab Pipelines

- Exoscale API key in the CI/CD variables as the inputs
- Terraform Docker image by GitLab with batteries included (e.g., jq)
- Terraform integration in merge requests with Terraform plan widget
- Kubeconfig as the final output artifact for download

⚠️ Ensure permissions for CI/CD input variables, Terraform state and sensitive Kubeconfig pipeline artifacts



Pipeline #81781 passed for 00c82049 on dev 1 minute ago

Approval is optional

View eligible approvers

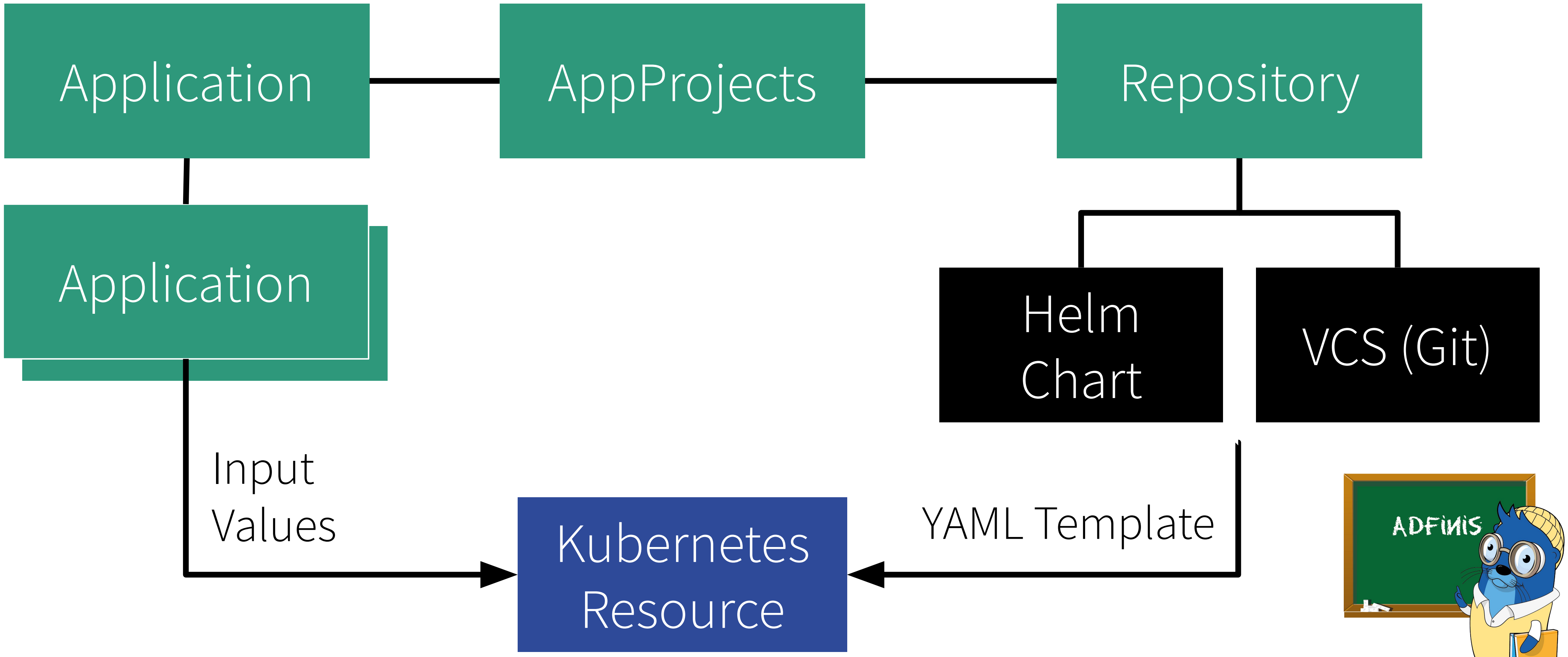
1 Terraform report was generated in your pipelines

The job **plan** generated a report. Reported Resource Changes: 10 to add, 0 to change, 0 to delete

Full log

Ready to merge!

Argo CD: Apps and Projects



Application

AppProjects

Repository

Application

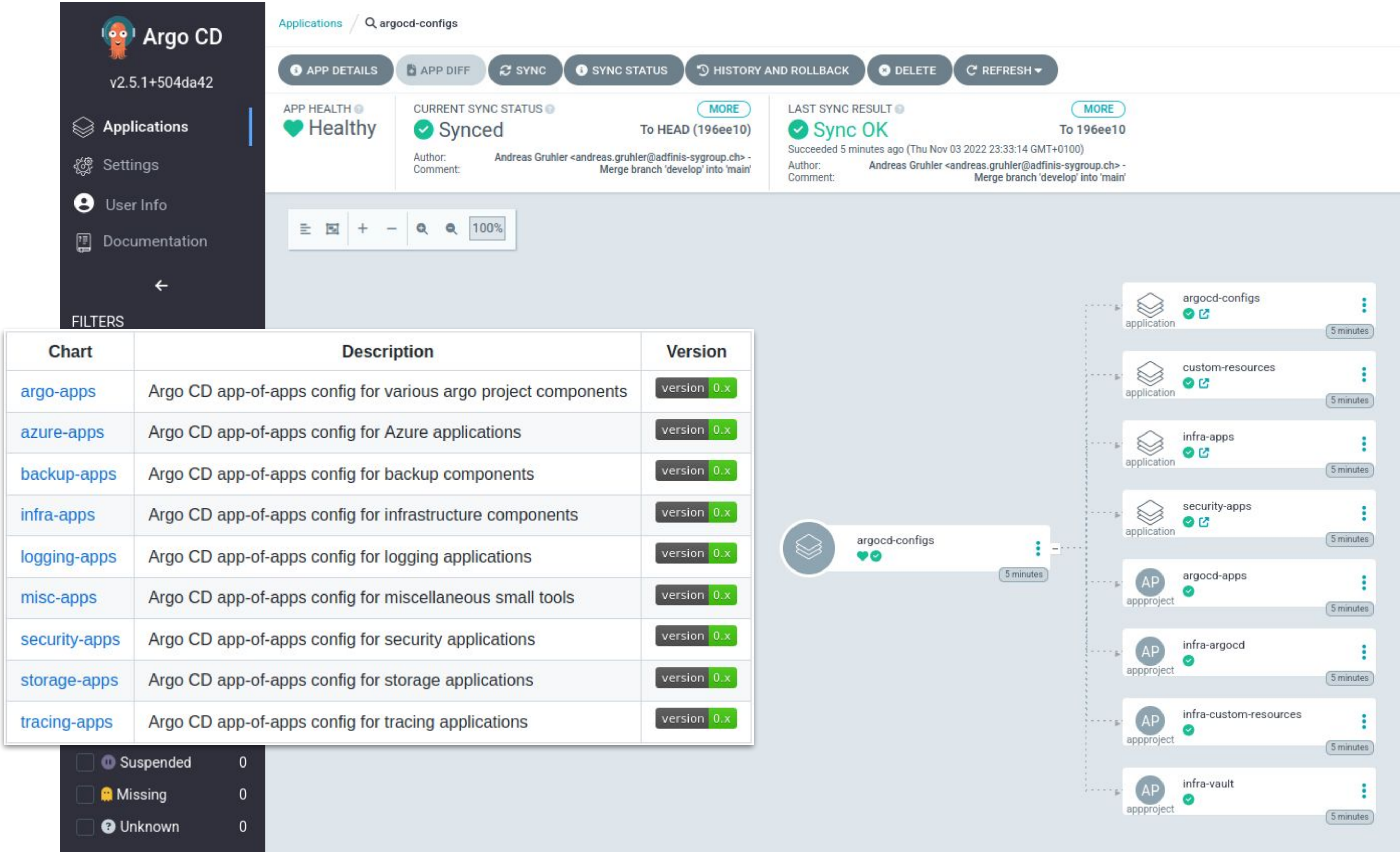
Helm Chart

VCS (Git)

Input Values

Kubernetes Resource

YAML Template



APP HEALTH: Healthy

CURRENT SYNC STATUS: Synced To HEAD (196ee10)
Author: Andreas Gruhler <andreas.gruhler@adfinis-sygroup.ch> -
Comment: Merge branch 'develop' into 'main'

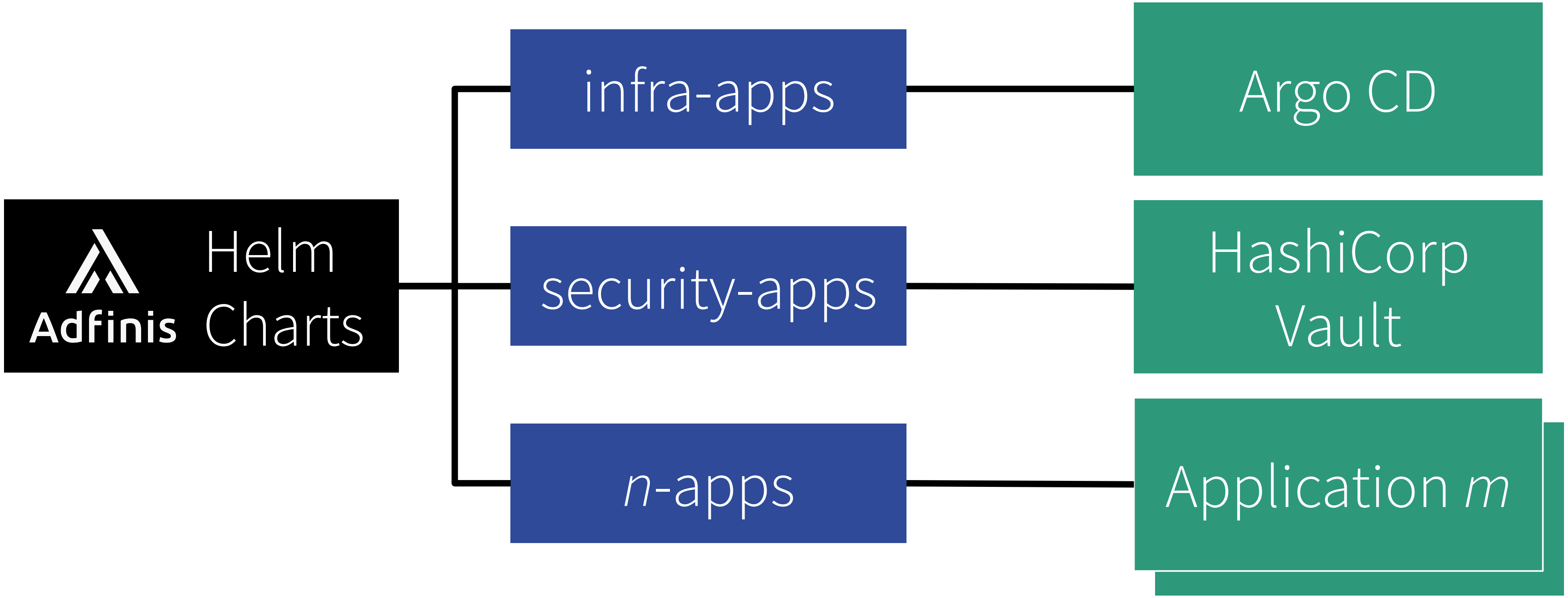
LAST SYNC RESULT: Sync OK To 196ee10
Succeeded 5 minutes ago (Thu Nov 03 2022 23:33:14 GMT+0100)
Author: Andreas Gruhler <andreas.gruhler@adfinis-sygroup.ch> -
Comment: Merge branch 'develop' into 'main'

Chart	Description	Version
argo-apps	Argo CD app-of-apps config for various argo project components	version 0.x
azure-apps	Argo CD app-of-apps config for Azure applications	version 0.x
backup-apps	Argo CD app-of-apps config for backup components	version 0.x
infra-apps	Argo CD app-of-apps config for infrastructure components	version 0.x
logging-apps	Argo CD app-of-apps config for logging applications	version 0.x
misc-apps	Argo CD app-of-apps config for miscellaneous small tools	version 0.x
security-apps	Argo CD app-of-apps config for security applications	version 0.x
storage-apps	Argo CD app-of-apps config for storage applications	version 0.x
tracing-apps	Argo CD app-of-apps config for tracing applications	version 0.x

Argo CD “App of Apps”

- Declaratively specify one Argo CD app that consisting of other apps
- Each Application can contain several Helm Charts
- The Applications can be used to organize Charts thematically
- Adfinis example: https://github.com/adfinis/helm-charts

Opinionated Adfinis “App of Apps” Umbrella Charts



Adfinis Helm Charts

infra-apps

security-apps

n-apps

Argo CD

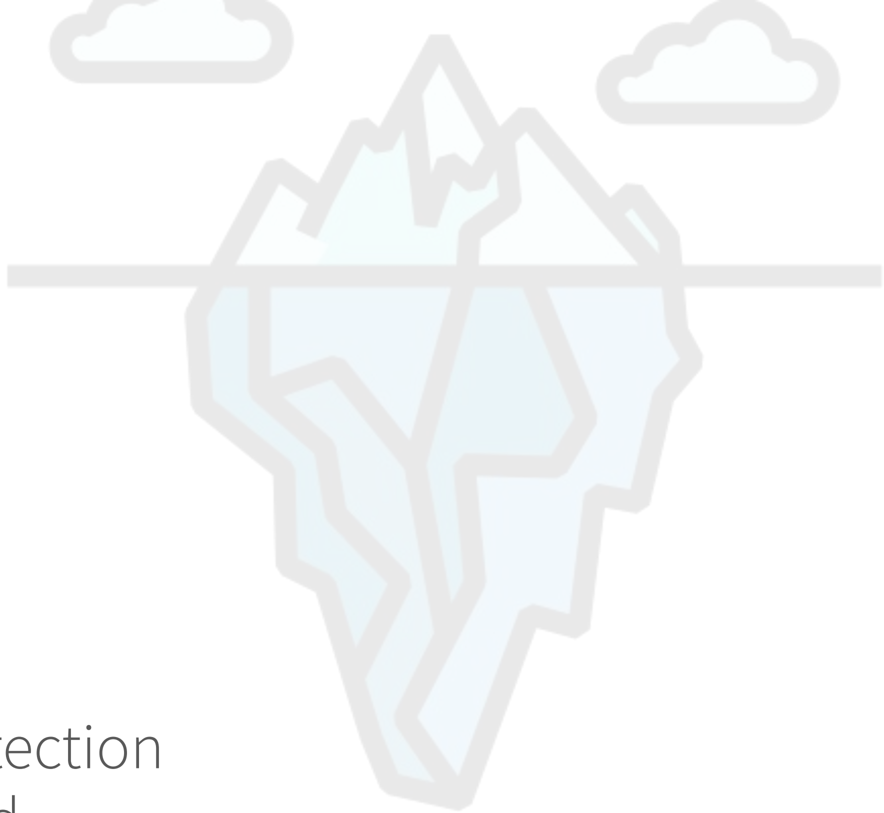
HashiCorp Vault

Application *m*

Benefits and Consequences of Umbrella Charts

Aspect	👍 Good	👎 Bad
Centralized repository	A central repository presents a single point for any configuration change	Maintenance is key. Otherwise, this translates to a single point of failure and a dependency on the Chart provider
Documentation	Documentation trail (changelogs) and ease of navigation, add your own docs	Confusion with opinionated organisation of Umbrella chart and upstream Charts
Review process	Consolidated 4-eye review from trusted sources	Reviews from the upstream Chart only
Testing ✌️	Confidence and reliability through (end-to-end) tests of the different categories (in union), organisation of the changes in a way that “works well together”	Isolated tests through maintainers of upstream Charts only
Life cycle management	Control the lifecycle of target revisions, structured rollout of changes with 4-eye principle and changelogs	No centralised lifecycle control for the upstream Charts
Communication	Consolidated communication of bugs and changes	Watch different upstream feeds for changes
Opinionated structure	Get used to a certain structure, same organisation on different customer systems	All team members need to agree and get comfortable with conventions and assumptions

HashiCorp Vault Enterprise



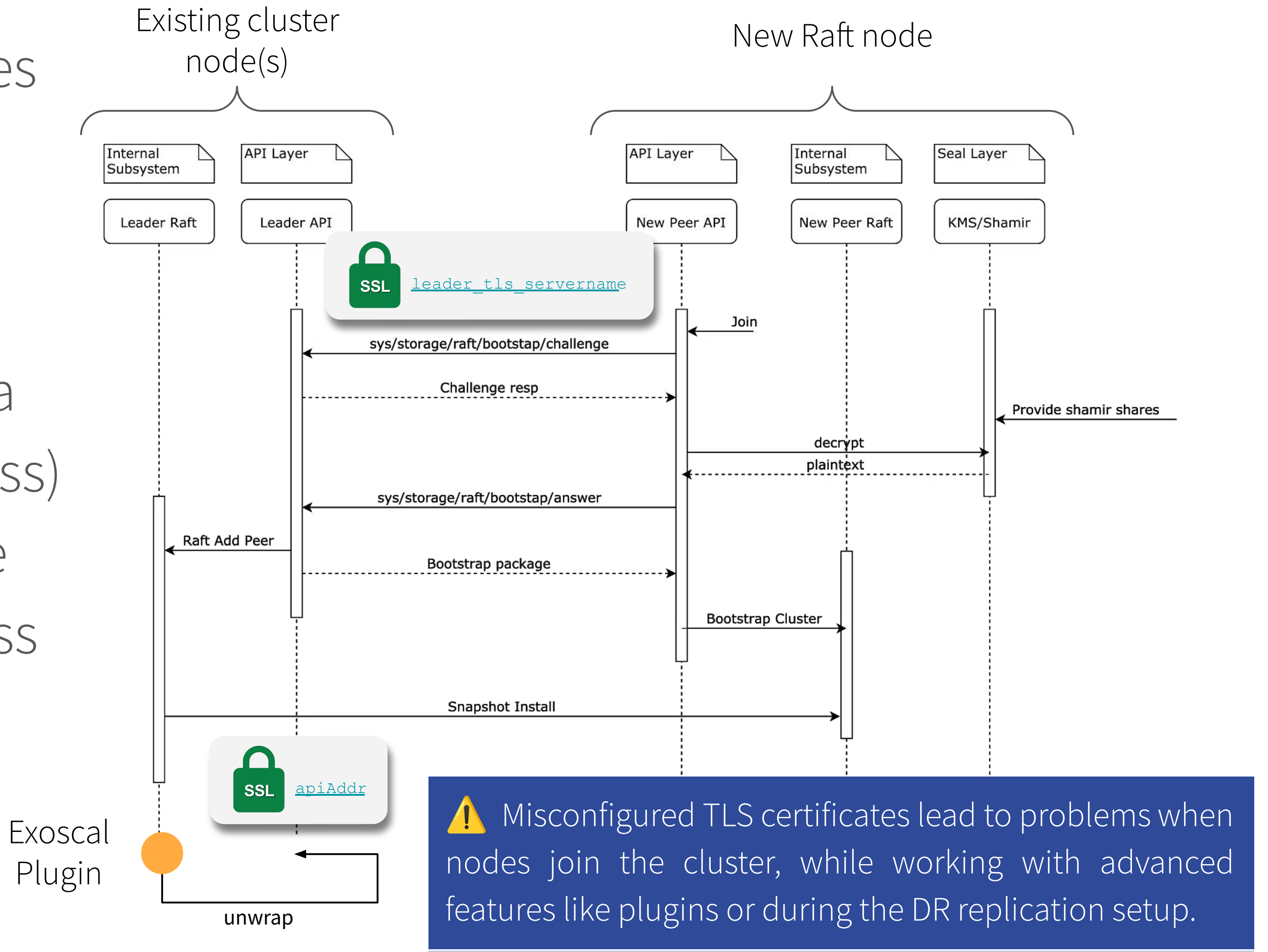
The “tip of the HashiCorp Vault Enterprise Iceberg”:

- Disaster recovery replication
- Multi-tenancy with namespaces
- Automated integrated storage snapshots
- Enterprise secrets engines for Advanced Data Protection
- HSM integrations for unsealing, Seal wrapping and entropy augmentation
- Performance standby nodes
- …

Trial licenses can be requested at https://vaultproject.io/trial

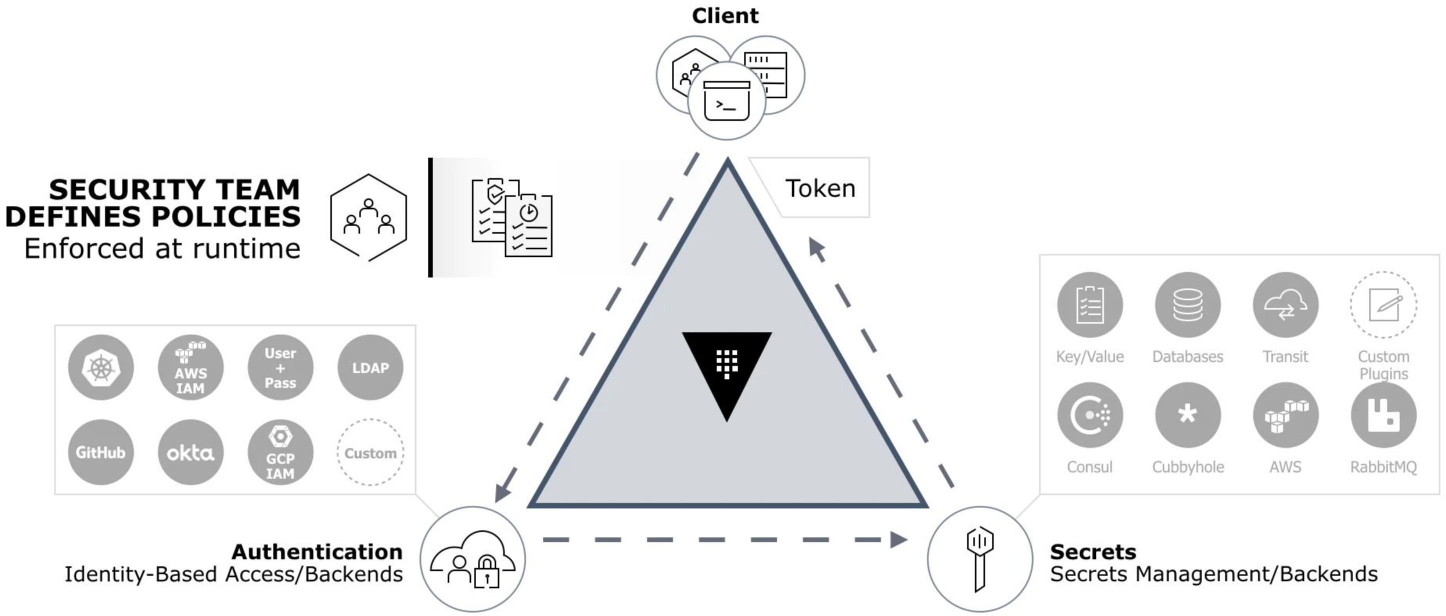
Vault Integrated Storage and TLS

- HashiCorp Vault Enterprise requires a (HA) Raft or Consul storage backend
- Proper TLS certificates are a prerequisites for joining nodes to a Vault cluster ([bootstrapping](#) process)
- The cluster nodes join through the internal Kubernetes service address for Vault



⚠️ Misconfigured TLS certificates lead to problems when nodes join the cluster, while working with advanced features like plugins or during the DR replication setup.

Dynamic Exoscale IAM Credentials with HashiCorp Vault



- Plug & play: Secret engine vs. auth backend plugins
- The plugin binary is injected into Vault server Pods through init containers
- The ⌛ TTL of the dynamic API credentials are managed by leases
- Access to the secrets engines is secured by policies (principle of least privilege)

Destroy the Cluster

- Destroy the Kubernetes objects (Argo CD) and Exoscale components managed by the Cloud Controller Manager (CCM) first

Code and Limitations

https://github.com/adfinis/sks-vault-demo

Limitations of the Demo Setup:

- No auto-unsealing
- No persistent data storage and no audit logs
- No Ingress: Vault API not exposed outside of the Kubernetes cluster
- No identity management and Vault policy
- Self-signed TLS certificate

Stay in Touch

/adfinis

/adfinis

adfinis.com

info@adfinis.com

/adfinis