インチキ盆栽クラスタ

https://github.com/n4mlz/infra

発表者

- 川崎晃太朗
- 情報科学類4年
- リモートで発表します (声が聞こえにくかったらすみません)

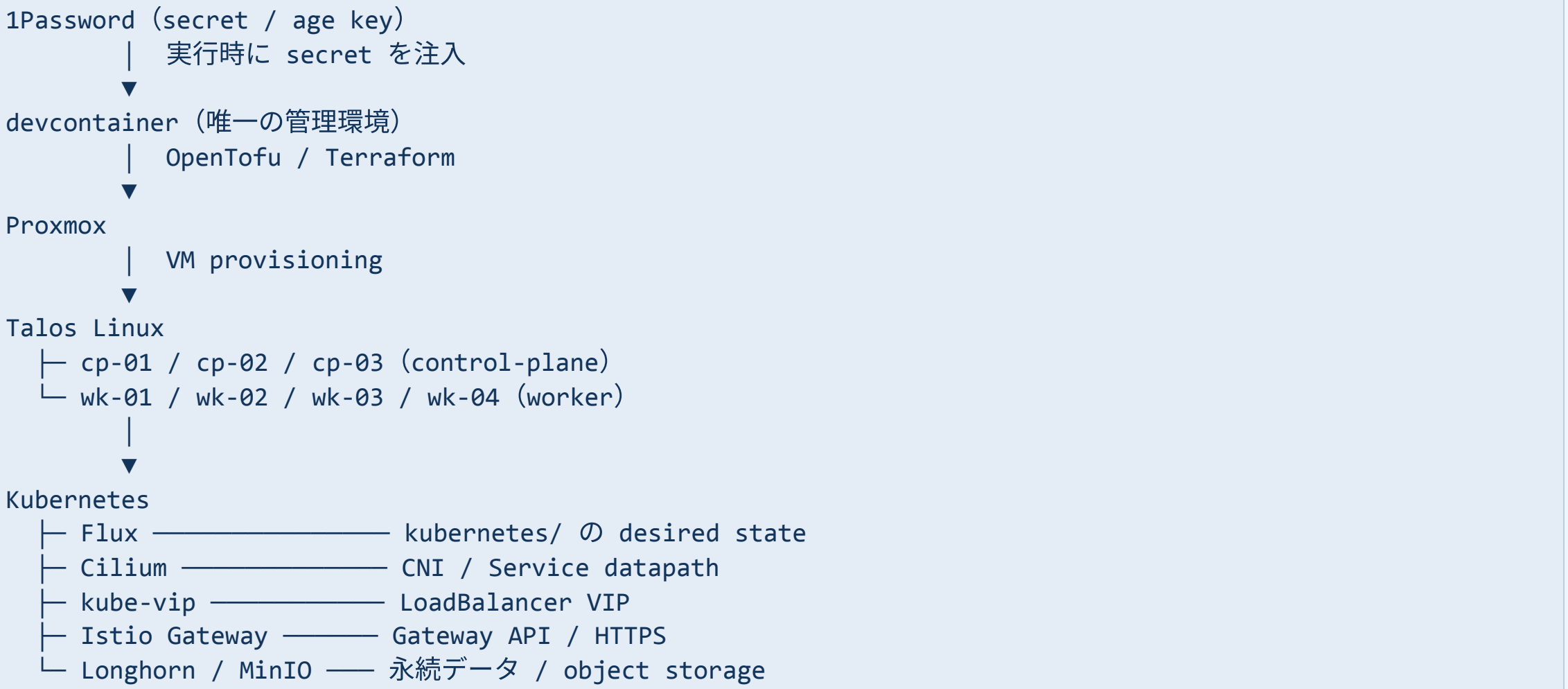
このリポジトリが担うもの

- 個人的なインフラを IaC と GitOps で管理するリポジトリ。
- このリポジトリは public にする前提で、secret は原則として 1Password または SOPS 暗号化ファイルに退避する。
- クラスタやアプリケーションの細かい手順は `docs/` と各ディレクトリの `README.md` に分けて記録する。

使用技術

領域	技術
VM provisioning	OpenTofu / Terraform, Proxmox
Kubernetes OS	Talos Linux, talhelper
GitOps	Flux
CNI / Service datapath	Cilium
LoadBalancer VIP	kube-vip, kube-vip-cloud-provider
Ingress	Gateway API, Istio Gateway
TLS / DNS	cert-manager, external-dns, Cloudflare
Storage	Longhorn, DirectPV, MinIO
Secret 管理	External Secrets Operator, 1Password, SOPS age
運用環境	devcontainer, Taskfile

クラスタの全体像


1Password（secret / age key）
実行時に secret を注入
devcontainer（唯一の管理環境）
OpenTofu / Terraform
Proxmox
VM provisioning
Talos Linux
cp-01 / cp-02 / cp-03（control-plane）
wk-01 / wk-02 / wk-03 / wk-04（worker）
Kubernetes
Flux ——— kubernetes/ の desired state
Cilium ——— CNI / Service datapath
kube-vip ——— LoadBalancer VIP
Istio Gateway ——— Gateway API / HTTPS
Longhorn / MinIO —— 永続データ / object storage

リポジトリの構造

```
terraform/      Proxmox 上の Talos VM skeleton
talos/          Talos cluster topology と machine config 生成
kubernetes/     Flux が同期する Kubernetes desired state
docs/           設計、bootstrap 手順、runbook、security 方針
worklogs/       作業ログ
```

この構造で、インフラの構築・クラスタ設定・アプリケーション・運用知識を同じリポジトリから追跡できる。

再現可能な運用

- devcontainer による bootstrap 問題の排除、ツールや環境のバージョンの固定
  - 例えばクラスタを立てるために外から terraform を叩く VM などを用意する必要があるが、完全な IaC にこだわるならその VM を生成するための IaC... などニワトリ卵である
- taskfile による手続き操作の固定化 (宣言的に出来ない部分)
  - taskfile は他の taskfile に書かれた操作をサブコマンドとして親の taskfile から import できる (これは just でも出来る)
  - make や just と比べて見やすい

シークレットの扱い

- k8s 内の secret は全て 1password から生成 (external secret operator)
  - クラスタ内に Vault などを立てるとクラスタが死んだ時に全てが終わる
- クラスタ内 secret だけでなくそもそもクラスタを立てる為に使用する secret も 1password にある (詳細は後述)

公開リポジトリを中心にした構成

- リポジトリを出来る限り public にする運用
  - private な別リポジトリは用意していない
  - リポジトリの宣言的な IaC + 1password 内の secret のみでクラスタの構築が完了できる構成 (.gitignore は極力ビルドの残骸のみの最小限にする)
  - sops を利用して secret な情報の入ったファイルは暗号化して public にする
    - 例えば yaml ファイル内の形状を保ちながら value の部分だけ暗号化できるので暗号化後のファイルが何を表しているのかのみ理解できて良い

ノード基盤

- Kubernetes on Talos Linux on Proxmox
  - Talos Linux は Kubernetes のノードとして運用するための専用のディストリビューション
  - 他: Bottlerocket, CoreOS, Flatcar Container Linux

Talos の特徴

- Kubernetes on Talos Linux on Proxmox
  - 最大の特徴は完全なイミュータブル性 / talosctl (シェルがないため外から認証情報と IP アドレスと共に CLI で操作する。宣言的に設定を書いて CLI で appliy する)
  - ansible, kubespray, kubeadm の構成が丸ごといらなくなる
  - Talos は Terraform で立てる
  - qemu-agent を Talos に含めておくと初回起動時に DHCP から振られるアドレスを Proxmox から確認できて相性が良い

クラスタ構成と GitOps

- Kubernetes
  - 7ノード構成 (master 3 ノード、worker 4 ノード)
  - Flux による完全な GitOps (GitHub リポジトリを定期的にポーリングして自動で reconcile)

ネットワーク

- Kubernetes
  - ネットワーク
    - Cilium を利用
    - グローバル IP を節約する運用
      - ノードに private 用と public 用の NIC を2つ持っている
      - サービス用の IP pool にグローバル IP を4つ持っているが、クラスタは7ノード
      - kube-vip で NIC にグローバル IP を動的割り当てることでノード分の IP を必要とせずグローバル IP を節約し、IPAM は kube-vip-cloud-provider を用いることで Service に IP のアノテーションがいらないクラウドライクな運用に

サービス公開

- Kubernetes
  - 手軽にサービスを公開できる環境
    - Cloudflare と連携した cert-manager と external-dns により Let's Encrypt の証明書発行と DNS レコードの追加が reconcile によって自動で行われる
    - Istio Gateway によって新しくサービスを立てる時は公開するドメインとパスさえ書けば勝手にプロキシしてくれるので、上記も合わせると人間がほとんど意識することなく手軽に新規サービスの公開ができる

ストレージ

- Kubernetes
  - WIP: ストレージ基盤
    - Longhorn + SeaweedFS + CloudNativePG
    - Backblaze B2 を利用したクラウドへの定期バックアップ+リストアテストもやりたい

次の展望

- Kubernetes
  - WIP その他将来やりたいこと: 監視基盤/メトリクス/アラート、クラスタ内認証や RBAC、サービスメッシュ

まとめ

- 個人的なインフラを IaC と GitOps で管理するリポジトリ。
- Proxmox 上の Talos Linux に、master 3 ノード・worker 4 ノードの Kubernetes クラスタを構成。
- Terraform、Talos、Flux をつなぎ、VM から Kubernetes の desired state までを宣言的に管理。
- Cilium、kube-vip、Istio Gateway、cert-manager、external-dns により、限られたグローバル IP でサービスを公開。
- 1Password を source of truth とし、SOPS で暗号化した参照可能な構造を Git に置く。
- ストレージ、監視、認証/RBAC、サービスメッシュをこれから拡張していく。

https://github.com/n4mlz/infra