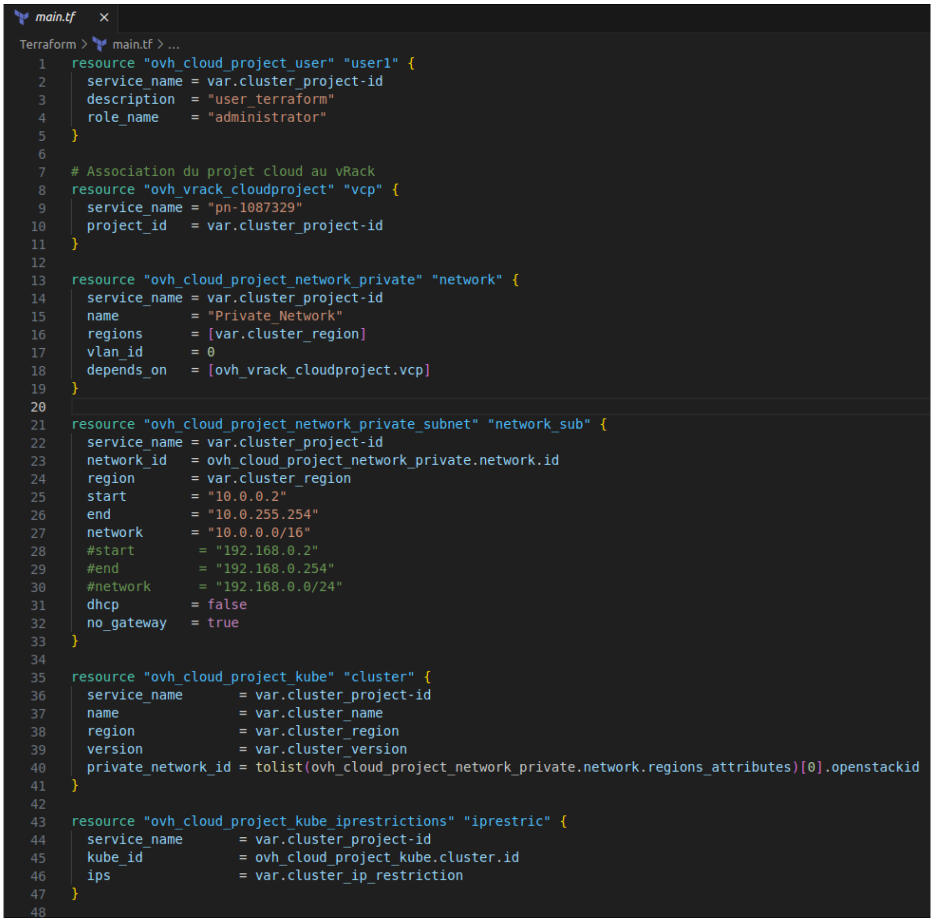
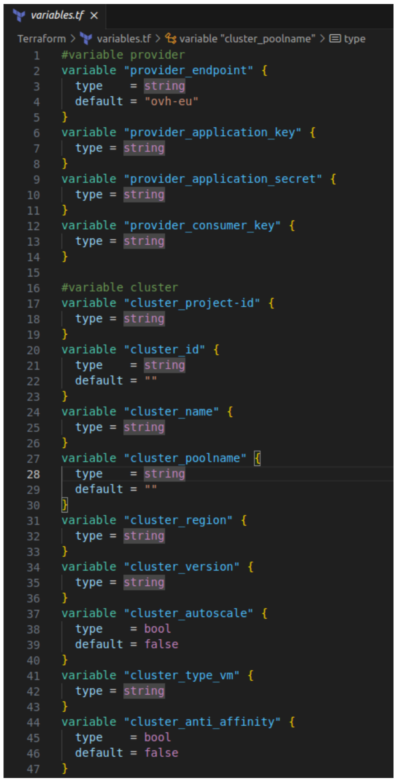
POC CLOUD

1) Déploiement Terraform

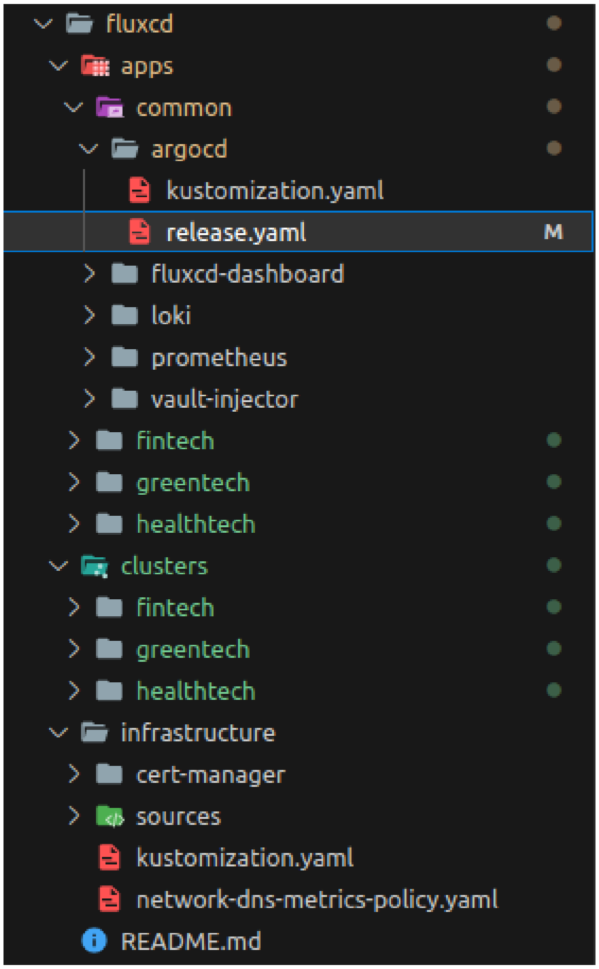


```
resource "ovh_cloud_project_user" "user1" {
  service_name = var.cluster_project-id
  description  = "user_terraform"
  role_name    = "administrator"
}

# Association du projet cloud au vRack
resource "ovh_vrack_cloudproject" "vcp" {
  service_name = "pn-1087329"
  project_id   = var.cluster_project-id
}

resource "ovh_cloud_project_network_private" "network" {
  service_name = var.cluster_project-id
  name         = "Private_Network"
  regions      = [var.cluster_region]
  vlan_id      = 0
  depends_on   = [ovh_vrack_cloudproject.vcp]
}

resource "ovh_cloud_project_network_private_subnet" "network_sub" {
  service_name = var.cluster_project-id
  network_id   = ovh_cloud_project_network_private.network.id
  region       = var.cluster_region
  start        = "10.0.0.2"
  end          = "10.0.255.254"
  network      = "10.0.0.0/16"
  #start         = "192.168.0.2"
  #end           = "192.168.0.254"
  #network       = "192.168.0.0/24"
  dhcp         = false
  no_gateway   = true
}

resource "ovh_cloud_project_kube" "cluster" {
  service_name       = var.cluster_project-id
  name               = var.cluster_name
  region             = var.cluster_region
  version            = var.cluster_version
  private_network_id = tolist(ovh_cloud_project_network_private.network.regions_attributes)[0].openstackid
}

resource "ovh_cloud_project_kube_iprestrictions" "iprestric" {
  service_name       = var.cluster_project-id
  kube_id            = ovh_cloud_project_kube.cluster.id
  ips                = var.cluster_ip_restriction
}

```


variables.tf ×
Terraform > variables.tf > variable "cluster_poolname" > type

```
#variable provider
variable "provider_endpoint" {
  type    = string
  default = "ovh-eu"
}
variable "provider_application_key" {
  type = string
}
variable "provider_application_secret" {
  type = string
}
variable "provider_consumer_key" {
  type = string
}

#variable cluster
variable "cluster_project-id" {
  type = string
}
variable "cluster_id" {
  type    = string
  default = ""
}
variable "cluster_name" {
  type = string
}
variable "cluster_poolname" {
  type    = string
  default = ""
}
variable "cluster_region" {
  type = string
}
variable "cluster_version" {
  type = string
}
variable "cluster_autoscale" {
  type    = bool
  default = false
}
variable "cluster_type_vm" {
  type = string
}
variable "cluster_anti_affinity" {
  type    = bool
  default = false
}
```

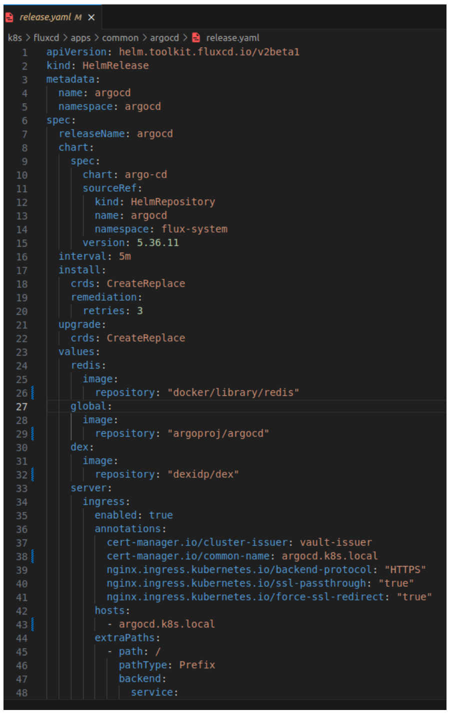
2) Architecture FluxCD



```
fluxcd
  apps
    common
      argocd
        kustomization.yaml
        release.yaml                      M
      fluxcd-dashboard
      loki
      prometheus
      vault-injector
    fintech
    greentech
    healthtech
  clusters
    fintech
    greentech
    healthtech
  infrastructure
    cert-manager
    sources
    kustomization.yaml
    network-dns-metrics-policy.yaml
  README.md
```

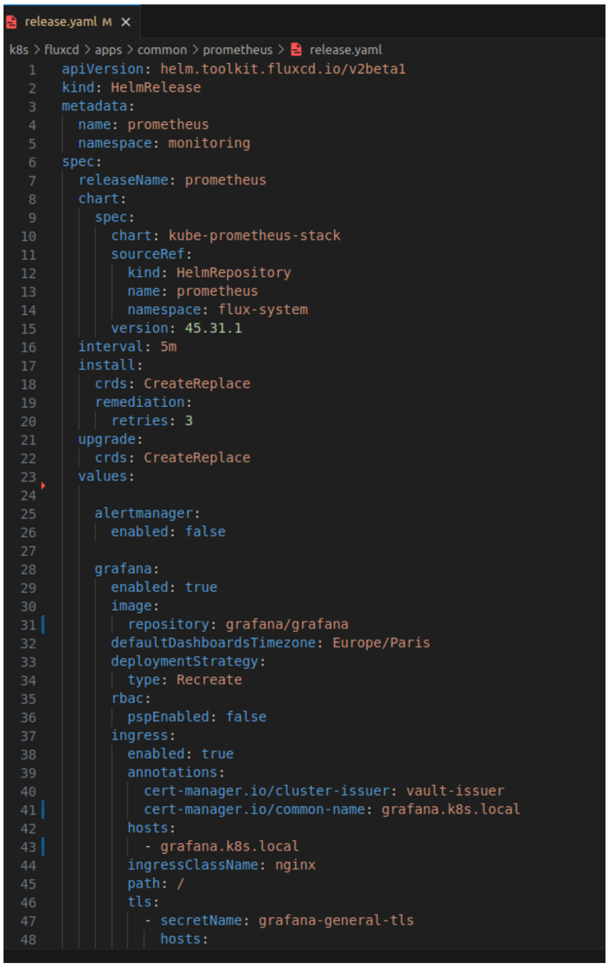
3) Déploiement infrastructure via Flux et Helm

a. ArgoCD



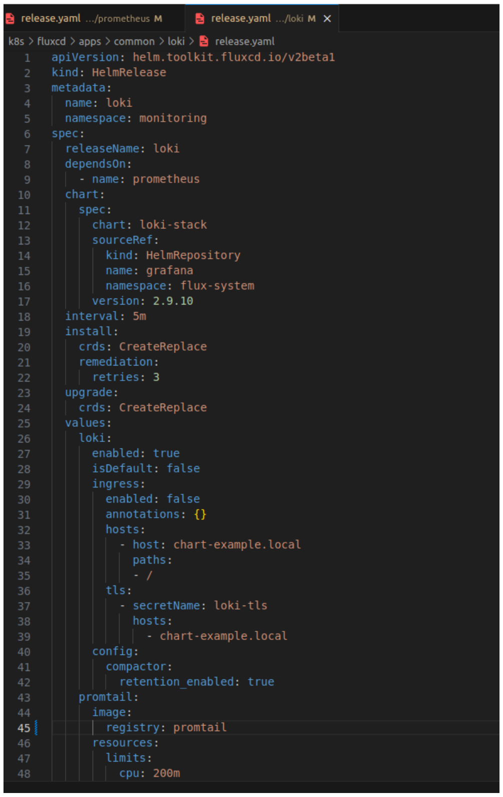
```yaml
apiVersion: helm.toolkit.fluxcd.io/v2beta1
kind: HelmRelease
metadata:
  name: argocd
  namespace: argocd
spec:
  releaseName: argocd
  chart:
    spec:
      chart: argo-cd
      sourceRef:
        kind: HelmRepository
        name: argocd
        namespace: flux-system
      version: 5.36.11
  interval: 5m
  install:
    crds: CreateReplace
    remediation:
      retries: 3
  upgrade:
    crds: CreateReplace
  values:
    redis:
      image:
        repository: "docker/library/redis"
    global:
      image:
        repository: "argoproj/argocd"
    dex:
      image:
        repository: "dexidp/dex"
    server:
      ingress:
        enabled: true
        annotations:
          cert-manager.io/cluster-issuer: vault-issuer
          cert-manager.io/common-name: argocd.k8s.local
          nginx.ingress.kubernetes.io/backend-protocol: "HTTPS"
          nginx.ingress.kubernetes.io/ssl-passthrough: "true"
          nginx.ingress.kubernetes.io/force-ssl-redirect: "true"
        hosts:
          - argocd.k8s.local
        extraPaths:
          - path: /
            pathType: Prefix
            backend:
              service:
```

b. Prometheus/Grafana



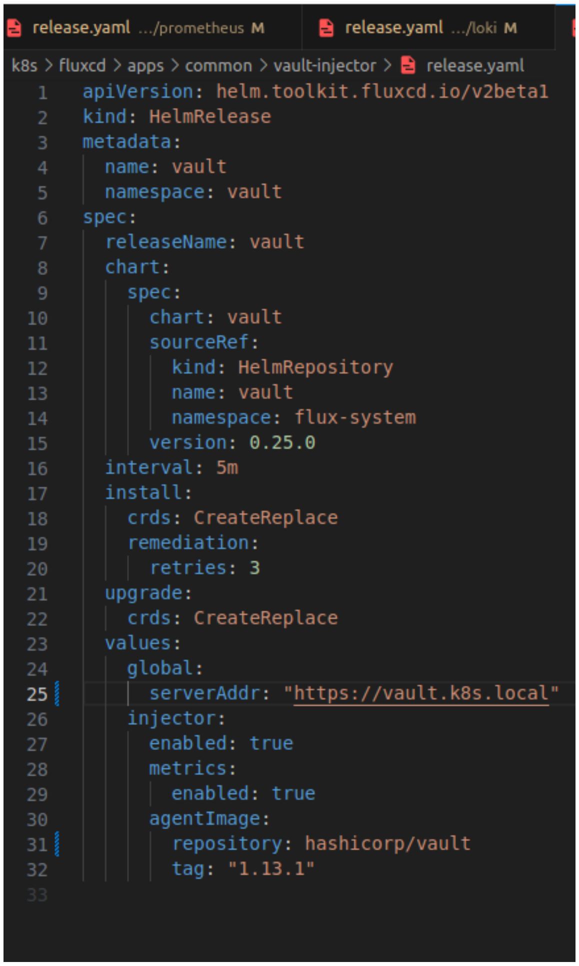
```yaml
apiVersion: helm.toolkit.fluxcd.io/v2beta1
kind: HelmRelease
metadata:
  name: prometheus
  namespace: monitoring
spec:
  releaseName: prometheus
  chart:
    spec:
      chart: kube-prometheus-stack
      sourceRef:
        kind: HelmRepository
        name: prometheus
        namespace: flux-system
      version: 45.31.1
  interval: 5m
  install:
    crds: CreateReplace
    remediation:
      retries: 3
  upgrade:
    crds: CreateReplace
  values:

    alertmanager:
      enabled: false

    grafana:
      enabled: true
      image:
        repository: grafana/grafana
      defaultDashboardsTimezone: Europe/Paris
      deploymentStrategy:
        type: Recreate
      rbac:
        pspEnabled: false
      ingress:
        enabled: true
        annotations:
          cert-manager.io/cluster-issuer: vault-issuer
          cert-manager.io/common-name: grafana.k8s.local
        hosts:
          - grafana.k8s.local
        ingressClassName: nginx
        path: /
        tls:
          - secretName: grafana-general-tls
            hosts:
```

c. Loki


release.yaml .../prometheus M
release.yaml .../loki M
k8s > fluxcd > apps > common > loki > release.yaml

```yaml
apiVersion: helm.toolkit.fluxcd.io/v2beta1
kind: HelmRelease
metadata:
  name: loki
  namespace: monitoring
spec:
  releaseName: loki
  dependsOn:
    - name: prometheus
  chart:
    spec:
      chart: loki-stack
      sourceRef:
        kind: HelmRepository
        name: grafana
        namespace: flux-system
      version: 2.9.10
  interval: 5m
  install:
    crds: CreateReplace
    remediation:
      retries: 3
  upgrade:
    crds: CreateReplace
  values:
    loki:
      enabled: true
      isDefault: false
      ingress:
        enabled: false
        annotations: {}
        hosts:
          - host: chart-example.local
            paths:
            - /
        tls:
          - secretName: loki-tls
            hosts:
              - chart-example.local
      config:
        compactor:
          retention_enabled: true
    promtail:
      image:
        registry: promtail
      resources:
        limits:
          cpu: 200m
```

d. Vault


release.yaml .../prometheus M
release.yaml .../loki M
k8s > fluxcd > apps > common > vault-injector > release.yaml

```yaml
apiVersion: helm.toolkit.fluxcd.io/v2beta1
kind: HelmRelease
metadata:
  name: vault
  namespace: vault
spec:
  releaseName: vault
  chart:
    spec:
      chart: vault
      sourceRef:
        kind: HelmRepository
        name: vault
        namespace: flux-system
      version: 0.25.0
  interval: 5m
  install:
    crds: CreateReplace
    remediation:
      retries: 3
  upgrade:
    crds: CreateReplace
  values:
    global:
      serverAddr: "https://vault.k8s.local"
    injector:
      enabled: true
      metrics:
        enabled: true
      agentImage:
        repository: hashicorp/vault
        tag: "1.13.1"
```