sveltos

Kubernetes addons

Kubernetes itself is not a complete solution. To build a production cluster, you need various additional addons.

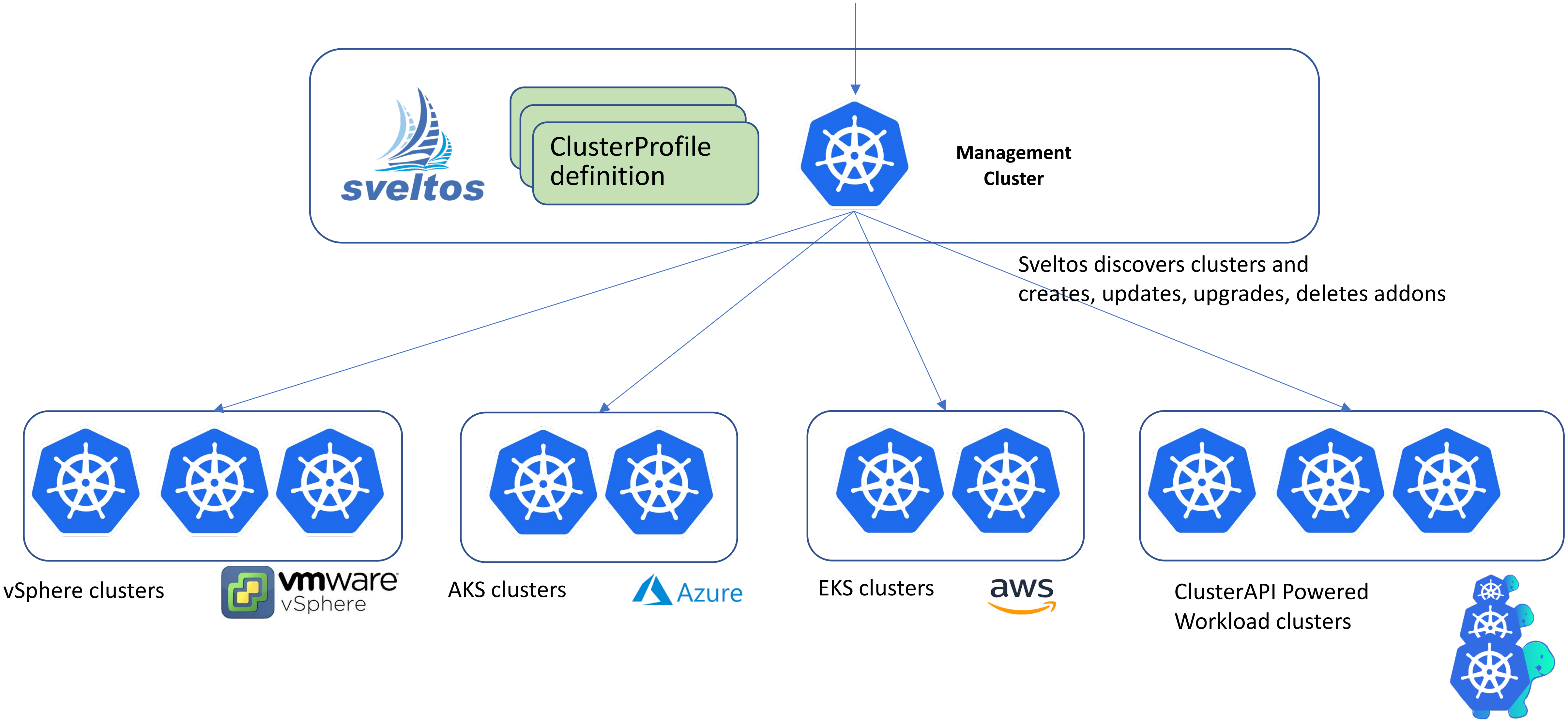
Sveltos wants to figure out the best way to install, manage and deliver cluster addons to tens of clusters.

The idea is simple:

1. from the management cluster, selects one or more clusters with a Kubernetes label selector;
2. lists which Kubernetes addons need to be deployed on such clusters.

Sveltos focuses not only on the ability to scale the number of clusters it can manage, but also to give visibility to exactly which addons are installed on each cluster.

Declarative configuration (which addons to deploy and where)



sveltos

ClusterProfile definition

Management Cluster

Sveltos discovers clusters and creates, updates, upgrades, deletes addons

vSphere clusters

vmware vSphere

AKS clusters

Azure

EKS clusters

aws

ClusterAPI Powered Workload clusters

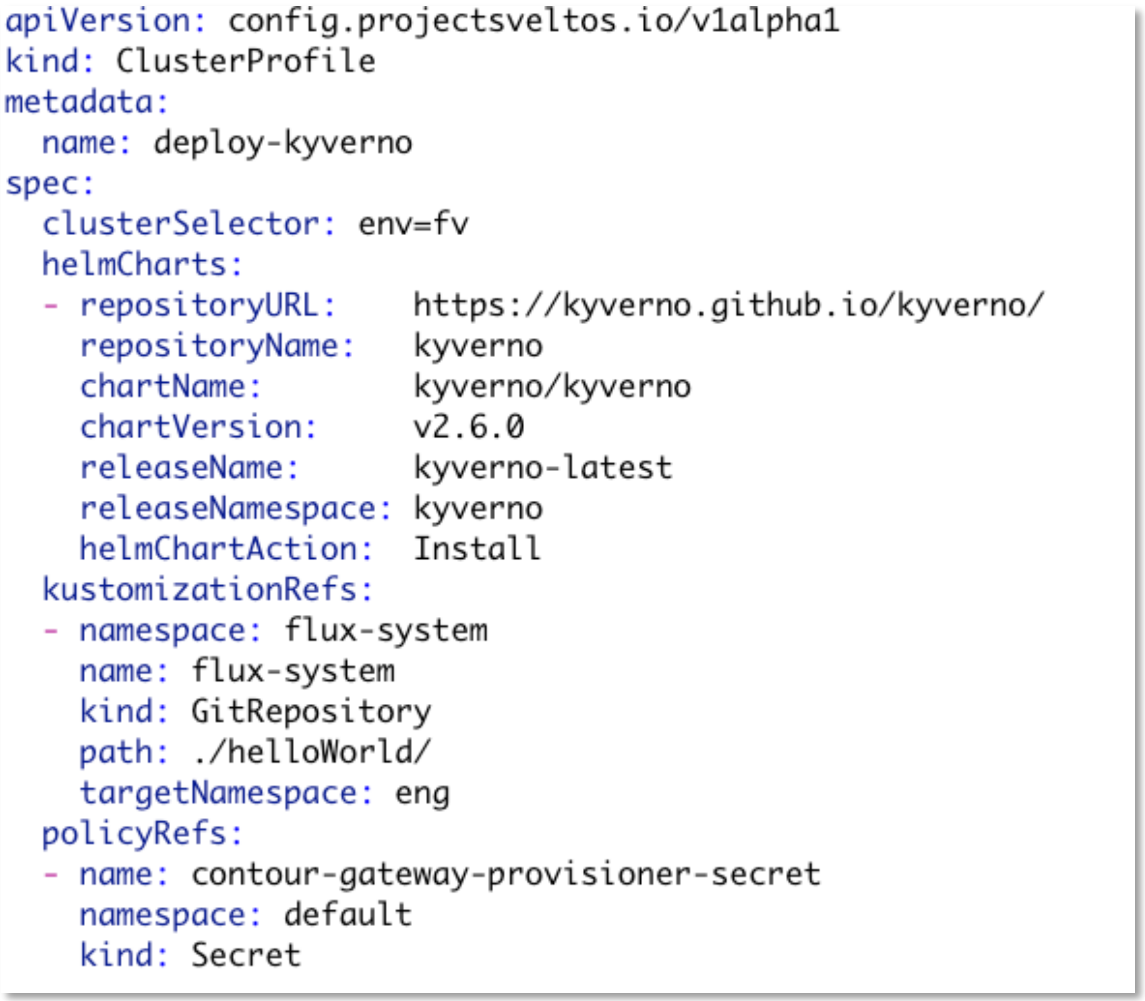
Other clusters can be easily registered with Sveltos

Built in support for ClusterAPI

ClusterProfile

ClusterProfile:

- CRD used to specify which add-ons need to be deployed in which cluster.



```
apiVersion: config.projectsveltos.io/v1alpha1
kind: ClusterProfile
metadata:
  name: deploy-kyverno
spec:
  clusterSelector: env=fv
  helmCharts:
  - repositoryURL:    https://kyverno.github.io/kyverno/
    repositoryName:   kyverno
    chartName:        kyverno/kyverno
    chartVersion:     v2.6.0
    releaseName:      kyverno-latest
    releaseNamespace: kyverno
    helmChartAction:  Install
  kustomizationRefs:
  - namespace: flux-system
    name: flux-system
    kind: GitRepository
    path: ./helloWorld/
    targetNamespace: eng
  policyRefs:
  - name: contour-gateway-provisioner-secret
    namespace: default
    kind: Secret
```

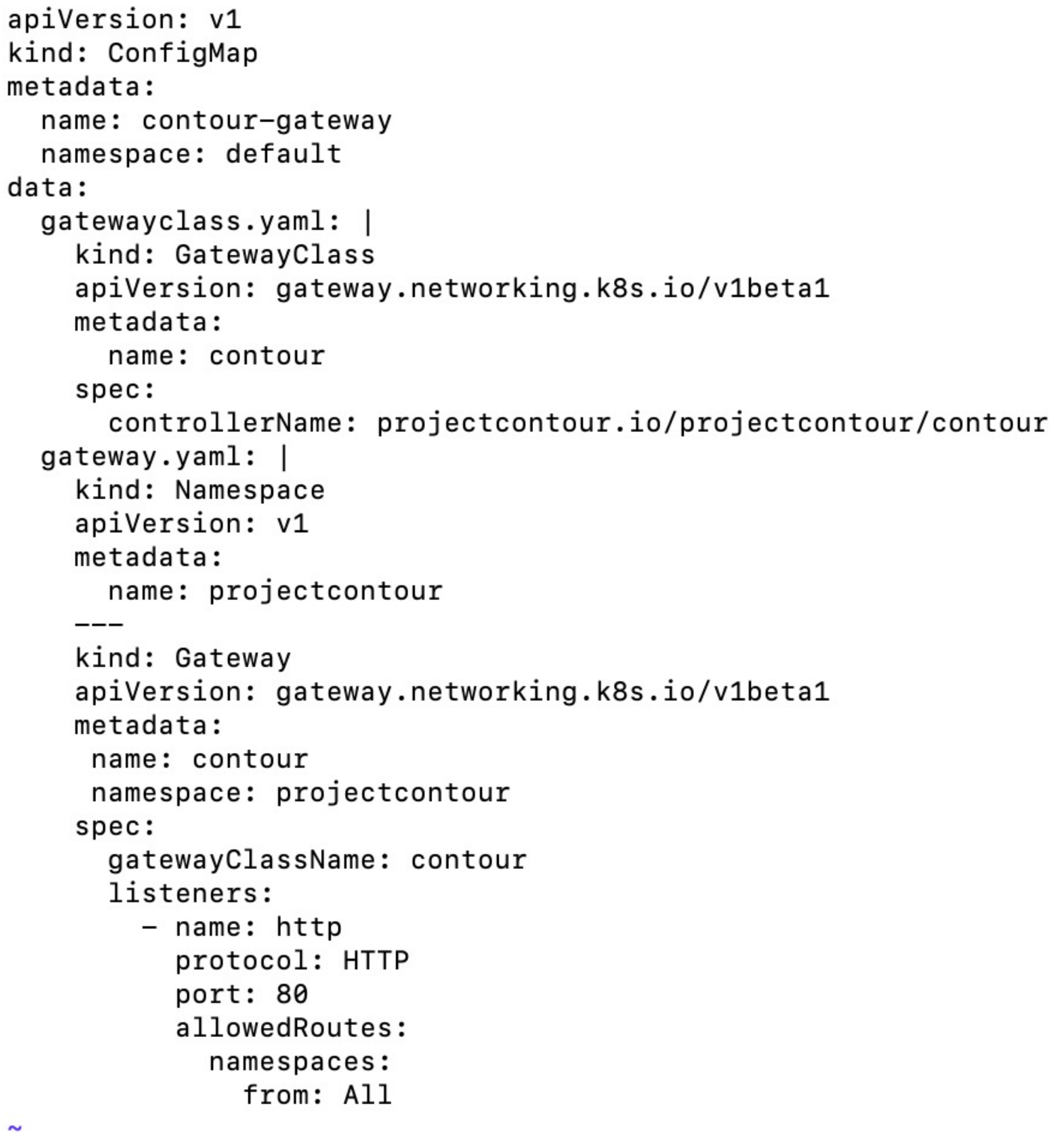
- ***clusterSelector:*** selects set of managed clusters;

- ***helmCharts***: list of helm charts to be deployed in the clusters matching clusterSelector;

- **kustomizationRefs**: : list of sources containing kustomization files. Resources will be deployed in the clusters matching clusterSelector;

- ***policyRefs***: list of ConfigMaps/Secrets containing the Kubernetes resources to be deployed in the clusters matching clusterSelector.

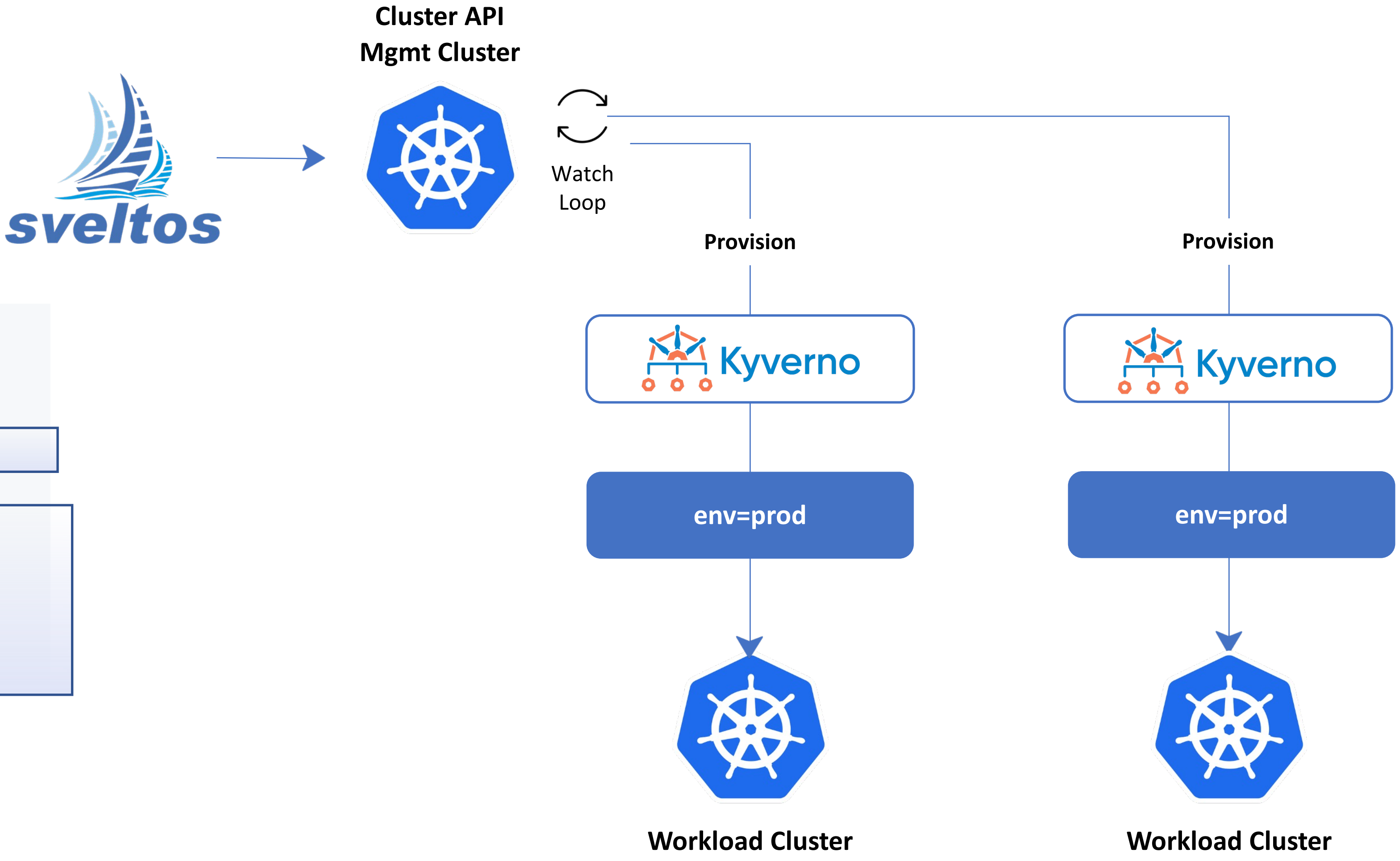
ConfigMap with YAML



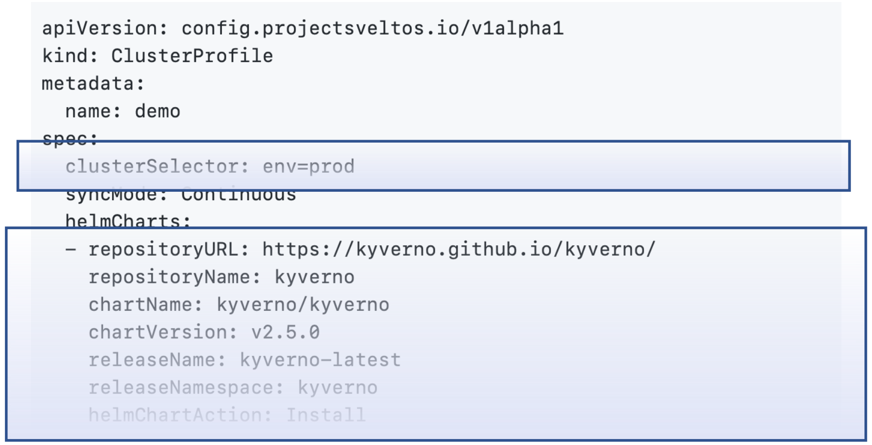
```
apiVersion: v1
kind: ConfigMap
metadata:
  name: contour-gateway
  namespace: default
data:
  gatewayclass.yaml: |
    kind: GatewayClass
    apiVersion: gateway.networking.k8s.io/v1beta1
    metadata:
      name: contour
    spec:
      controllerName: projectcontour.io/projectcontour/contour
  gateway.yaml: |
    kind: Namespace
    apiVersion: v1
    metadata:
      name: projectcontour
    ---
    kind: Gateway
    apiVersion: gateway.networking.k8s.io/v1beta1
    metadata:
     name: contour
     namespace: projectcontour
    spec:
      gatewayClassName: contour
      listeners:
        - name: http
          protocol: HTTP
          port: 80
          allowedRoutes:
            namespaces:
              from: All
~
```

- Data can contain one or more resources;
- Both YAML or JSON can be used

Project Sveltos - Policy Driven Software Lifecycle Mgmt

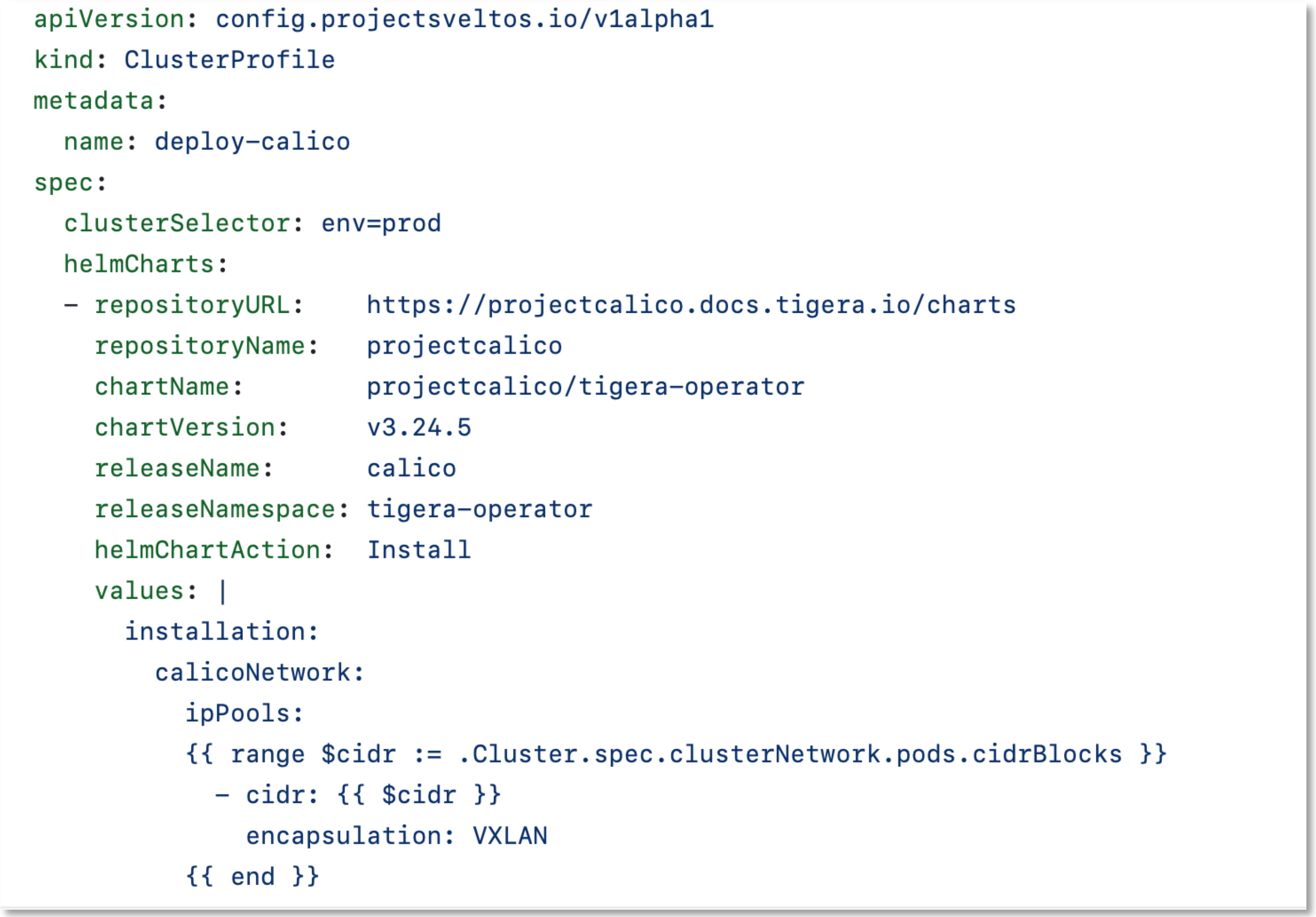


kubectl apply -f …



```
apiVersion: config.projectsveltos.io/v1alpha1
kind: ClusterProfile
metadata:
  name: demo
spec:
  clusterSelector: env=prod
  syncMode: Continuous
  helmCharts:
  - repositoryURL: https://kyverno.github.io/kyverno/
    repositoryName: kyverno
    chartName: kyverno/kyverno
    chartVersion: v2.5.0
    releaseName: kyverno-latest
    releaseNamespace: kyverno
    helmChartAction: Install
```

Project Sveltos - Templates



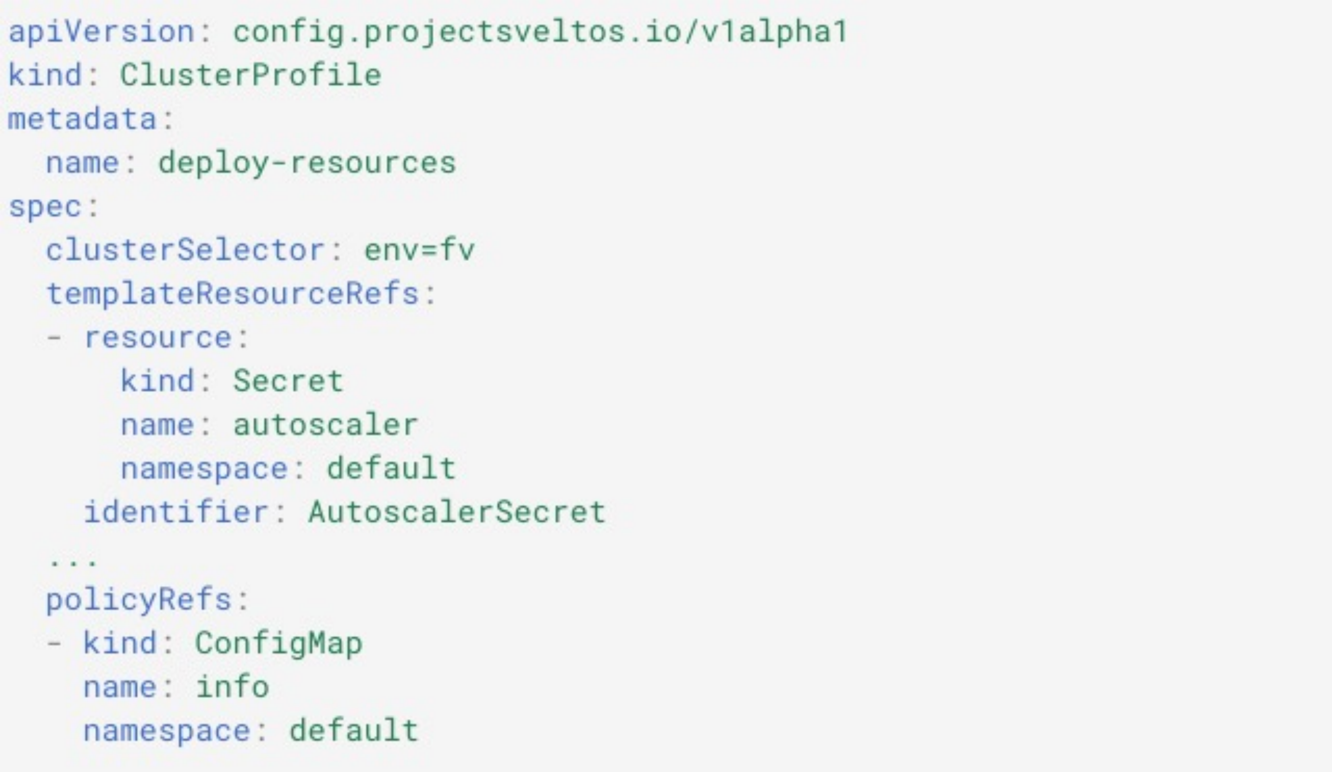
```yaml
apiVersion: config.projectsveltos.io/v1alpha1
kind: ClusterProfile
metadata:
  name: deploy-calico
spec:
  clusterSelector: env=prod
  helmCharts:
  - repositoryURL:    https://projectcalico.docs.tigera.io/charts
    repositoryName:   projectcalico
    chartName:        projectcalico/tigera-operator
    chartVersion:     v3.24.5
    releaseName:      calico
    releaseNamespace: tigera-operator
    helmChartAction:  Install
    values: |
      installation:
        calicoNetwork:
          ipPools:
          {{ range $cidr := .Cluster.spec.clusterNetwork.pods.cidrBlocks }}
            - cidr: {{ $cidr }}
              encapsulation: VXLAN
          {{ end }}
```

Can fetch data from management Cluster.

Currently fetched by default:

1. Cluster instance
2. SveltosCluster instance
3. Infrastructure Provider instance
4. KubeadmControlPlane instance

Project Sveltos - Templates



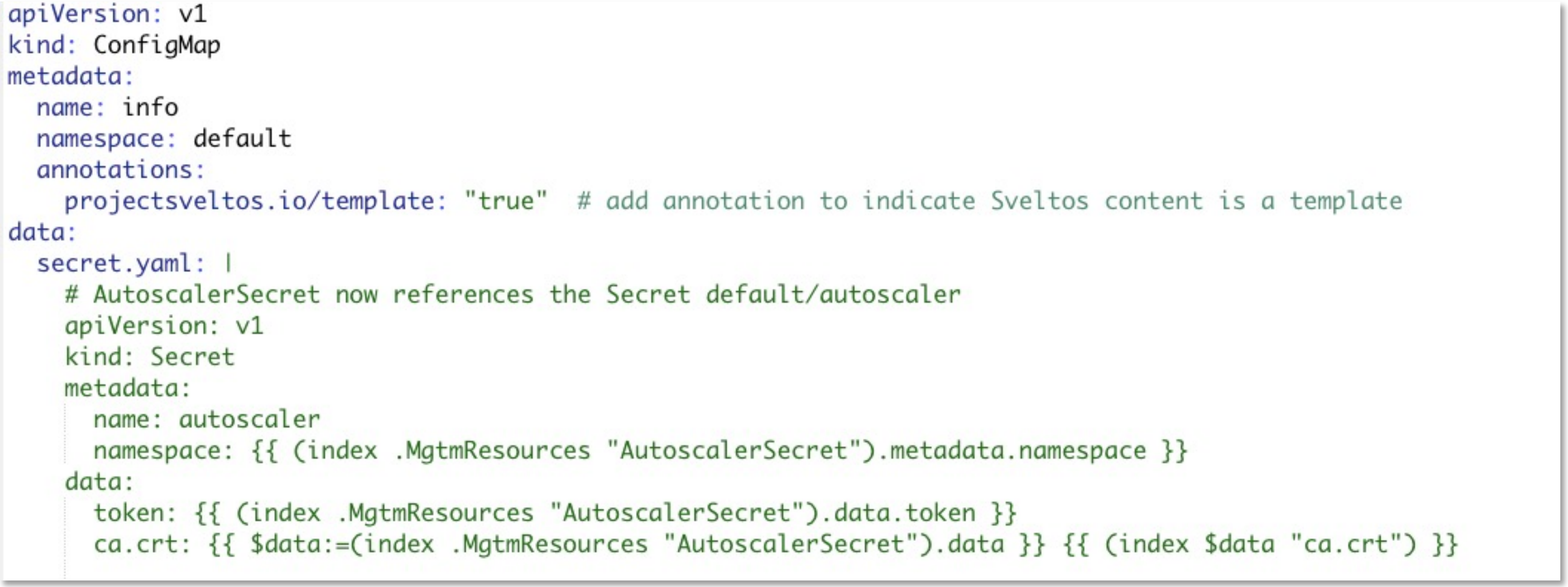
```yaml
apiVersion: config.projectsveltos.io/v1alpha1
kind: ClusterProfile
metadata:
  name: deploy-resources
spec:
  clusterSelector: env=fv
  templateResourceRefs:
  - resource:
      kind: Secret
      name: autoscaler
      namespace: default
    identifier: AutoscalerSecret
  ...
  policyRefs:
  - kind: ConfigMap
    name: info
    namespace: default
```

Sveltos can be instructed to fetch any resource from management cluster

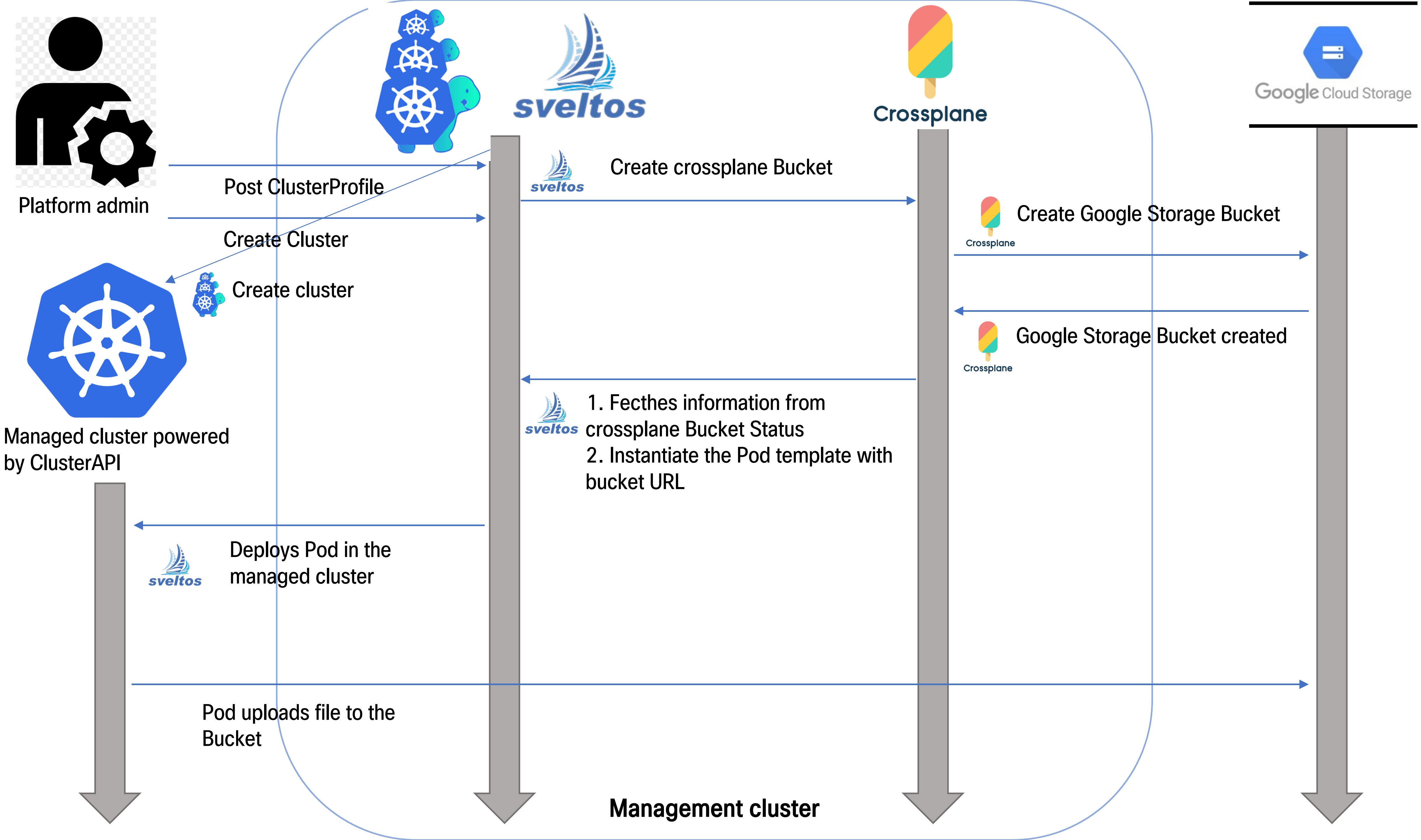
Following YAML instructs Sveltos to fetch the Secret instance *autoscaler* in the namespace *default* and make it available to the template with the keyword AutoscalerSecret

Sveltos does not have all the necessary permissions to fetch resources from the management cluster by default.
Therefore, when using *templateResourceRefs*, you need to provide Sveltos with the correct RBACs.

Project Sveltos - Templates



```yaml
apiVersion: v1
kind: ConfigMap
metadata:
  name: info
  namespace: default
  annotations:
    projectsveltos.io/template: "true"  # add annotation to indicate Sveltos content is a template
data:
  secret.yaml: |
    # AutoscalerSecret now references the Secret default/autoscaler
    apiVersion: v1
    kind: Secret
    metadata:
      name: autoscaler
      namespace: {{ (index .MgtmResources "AutoscalerSecret").metadata.namespace }}
    data:
      token: {{ (index .MgtmResources "AutoscalerSecret").data.token }}
      ca.crt: {{ $data:=(index .MgtmResources "AutoscalerSecret").data }} {{ (index $data "ca.crt") }}
```



Platform admin

Post ClusterProfile

Create Cluster

Create cluster

Managed cluster powered by ClusterAPI

Create crossplane Bucket

Create Google Storage Bucket

Google Storage Bucket created

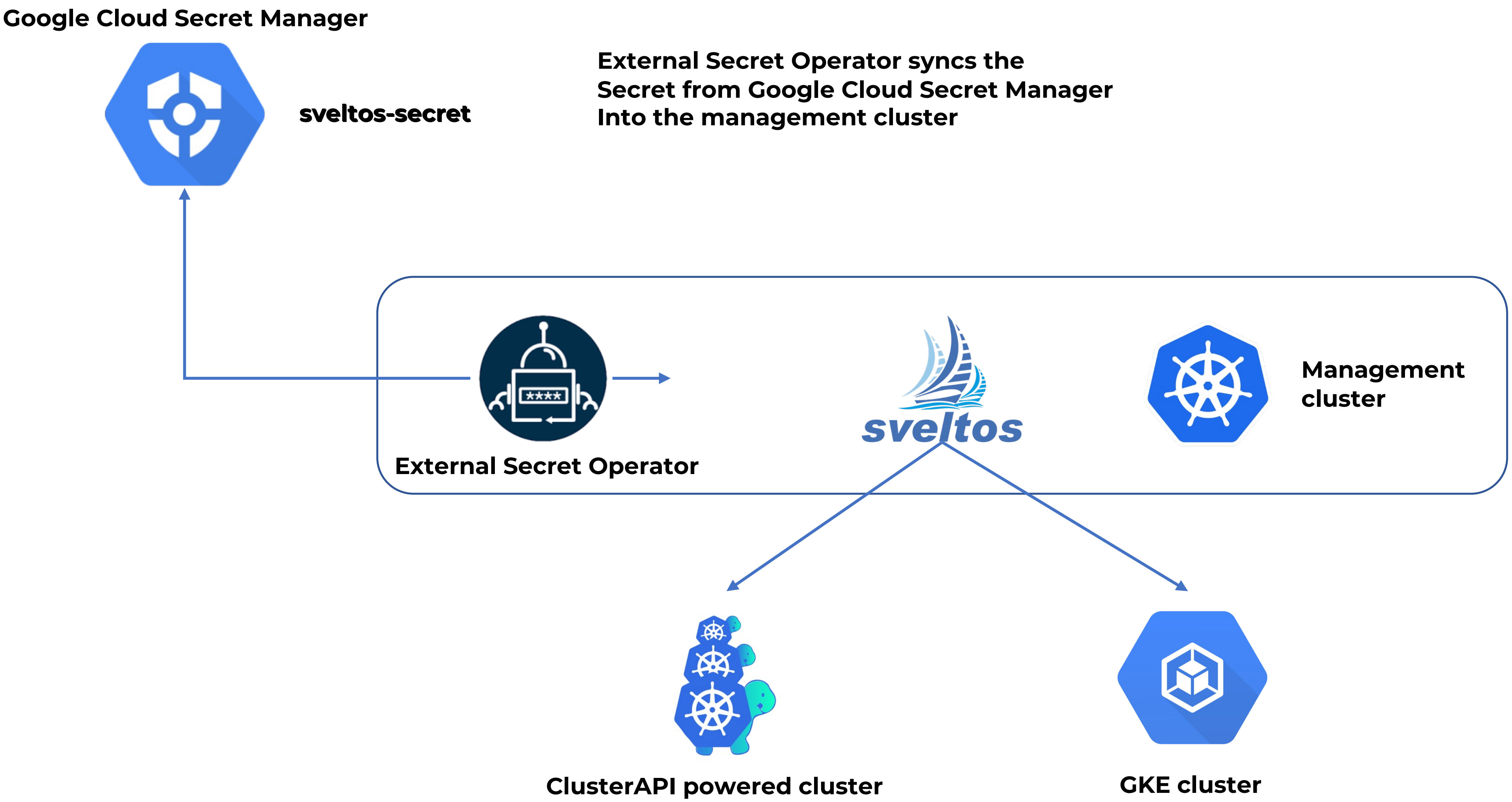
1. Fecthes information from crossplane Bucket Status
2. Instantiate the Pod template with bucket URL

Deploys Pod in the managed cluster

Pod uploads file to the Bucket

Management cluster

External Secret Management Integration



Google Cloud Secret Manager

sveltos-secret

External Secret Operator syncs the
Secret from Google Cloud Secret Manager
Into the management cluster

External Secret Operator

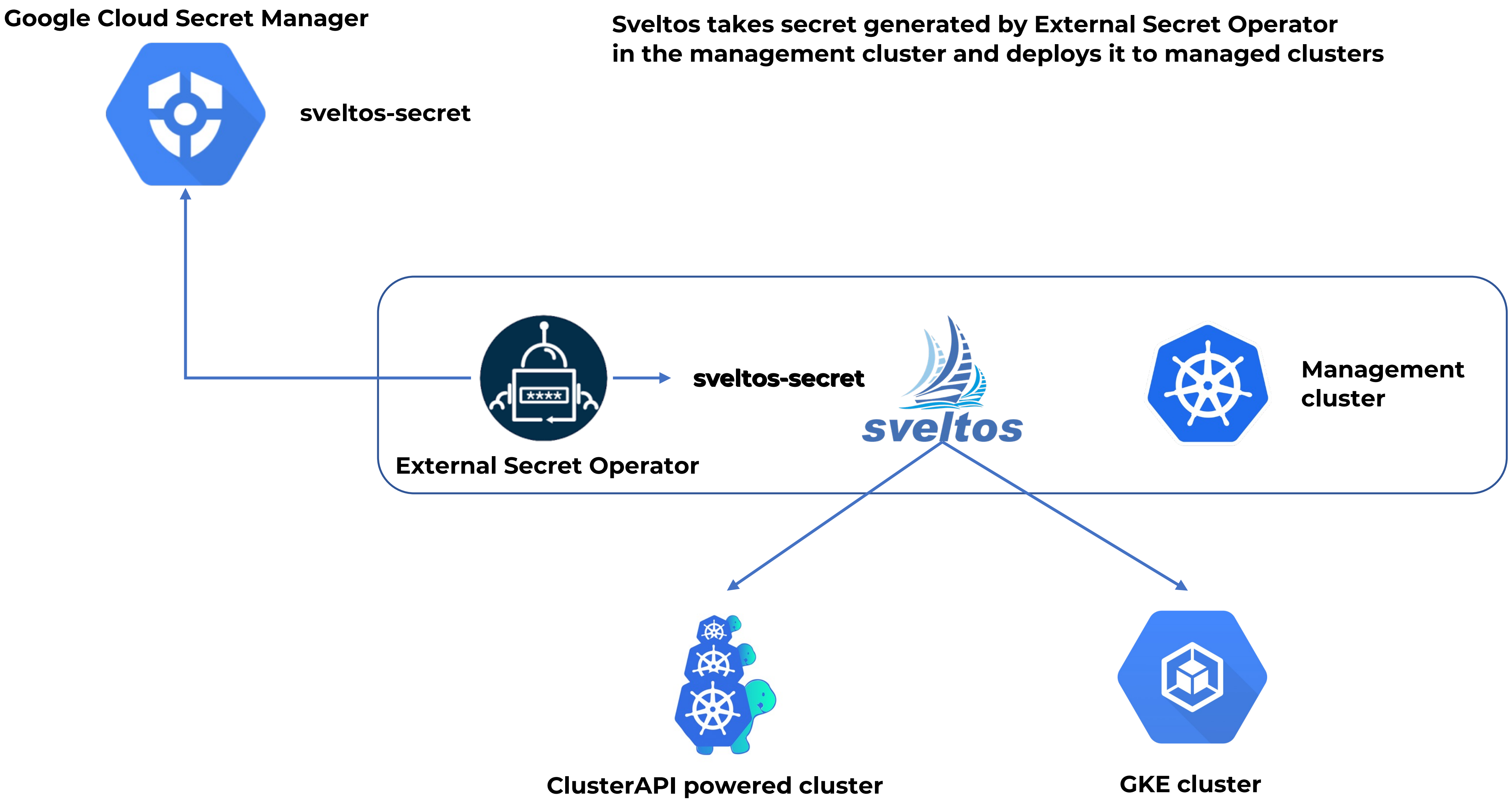
Management
cluster

ClusterAPI powered cluster

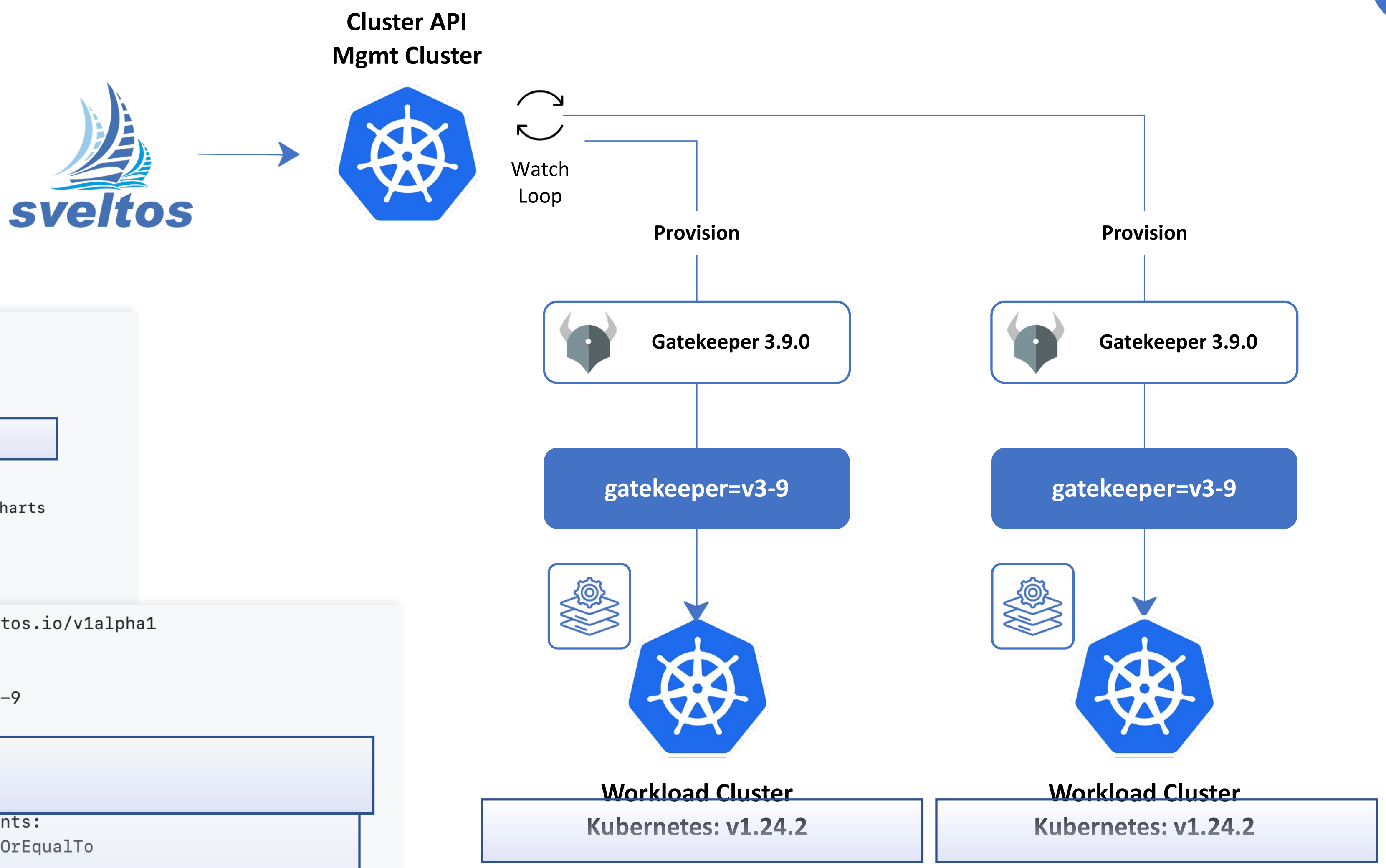
GKE cluster

External Secret Management Integration

Sveltos takes secret generated by External Secret Operator
in the management cluster and deploys it to managed clusters



Project Sveltos - Policy Driven Software Lifecycle Mgmt

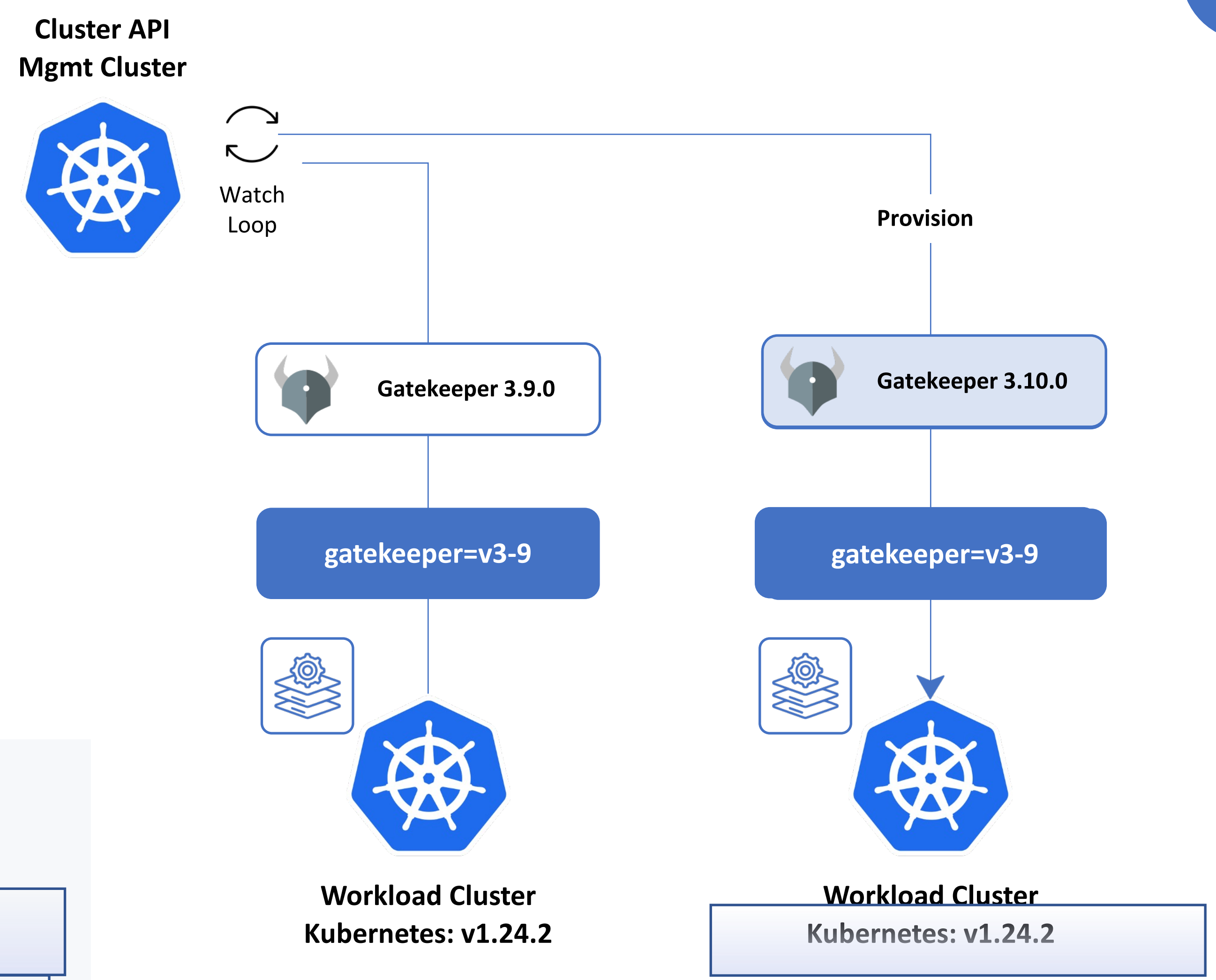


kubectl apply -f …

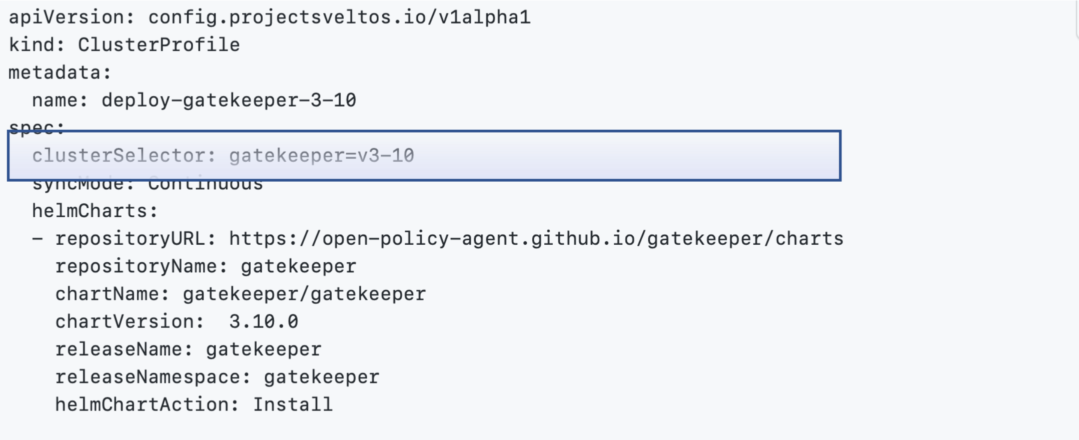
```
apiVersion: config.projectsveltos.io/v1alpha1
kind: ClusterProfile
metadata:
  name: deploy-gatekeeper-3-9
spec:
  clusterSelector: gatekeeper=v3-9
  syncMode: Continuous
  helmCharts:
  - repositoryURL: https://open-policy-agent.github.io/gatekeeper/charts
    repositoryName: gatekeeper
    chartName: gatekeeper/gatekeeper
    chartVersion:  3.9.0
    releaseName: gatekeeper
    releaseNamespace: gatekeeper
    helmChartAction: Install
```

```
apiVersion: lib.projectsveltos.io/v1alpha1
kind: Classifier
metadata:
  name: deploy-gatekeeper-3-9
spec:
  classifierLabels:
  - key: gatekeeper
    value: v3-9
  kubernetesVersionConstraints:
  - comparison: GreaterThanOrEqualTo
    version: 1.24.0
  - comparison: LessThan
    version: 1.25.0
```

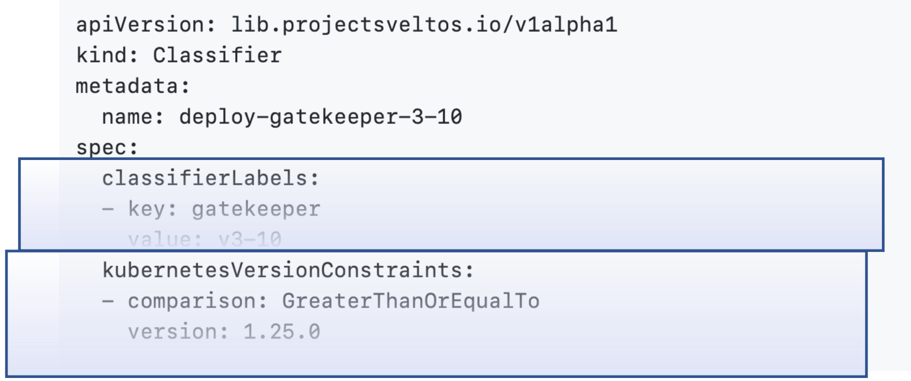
Project Sveltos - Policy Driven Software Lifecycle Mgmt



kubectl apply -f ...



```
apiVersion: config.projectsveltos.io/v1alpha1
kind: ClusterProfile
metadata:
  name: deploy-gatekeeper-3-10
spec:
  clusterSelector: gatekeeper=v3-10
  syncMode: Continuous
  helmCharts:
  - repositoryURL: https://open-policy-agent.github.io/gatekeeper/charts
    repositoryName: gatekeeper
    chartName: gatekeeper/gatekeeper
    chartVersion:  3.10.0
    releaseName: gatekeeper
    releaseNamespace: gatekeeper
    helmChartAction: Install
```



```
apiVersion: lib.projectsveltos.io/v1alpha1
kind: Classifier
metadata:
  name: deploy-gatekeeper-3-10
spec:
  classifierLabels:
  - key: gatekeeper
    value: v3-10
  kubernetesVersionConstraints:
  - comparison: GreaterThanOrEqualTo
    version: 1.25.0
```

Project Sveltos - References

Github: https://github.com/projectsveltos
Documentation: https://projectsveltos.github.io/sveltos/
Slack: @Projectsveltos
Linkedin: https://www.linkedin.com/in/gianlucamardente/