
sveltos

Kubernetes addons

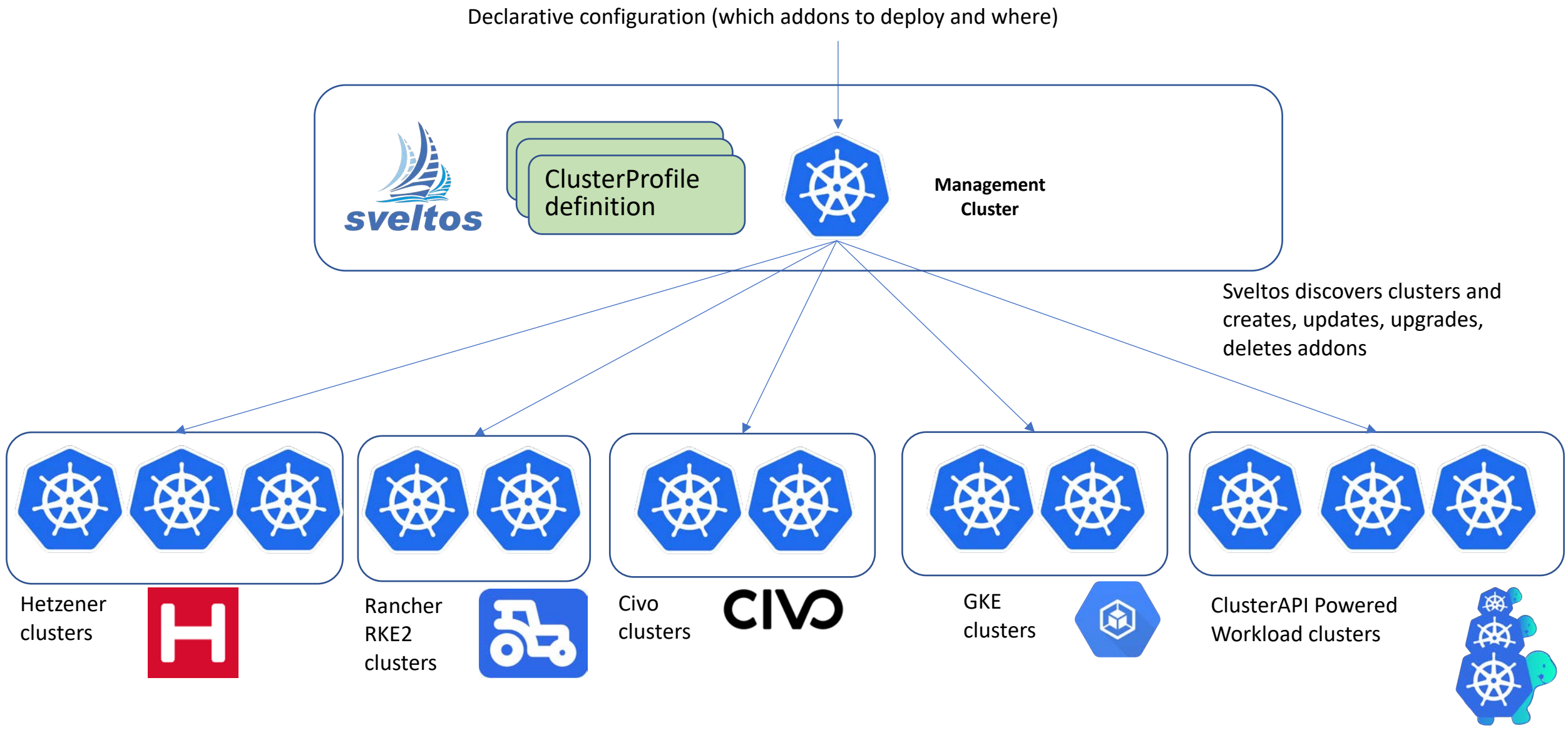
Kubernetes itself is not a complete solution. To build a production cluster, you need various additional addons.

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
Hetzener clusters
Rancher RKE2 clusters
Civo clusters
GKE clusters
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.

```yaml
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
```

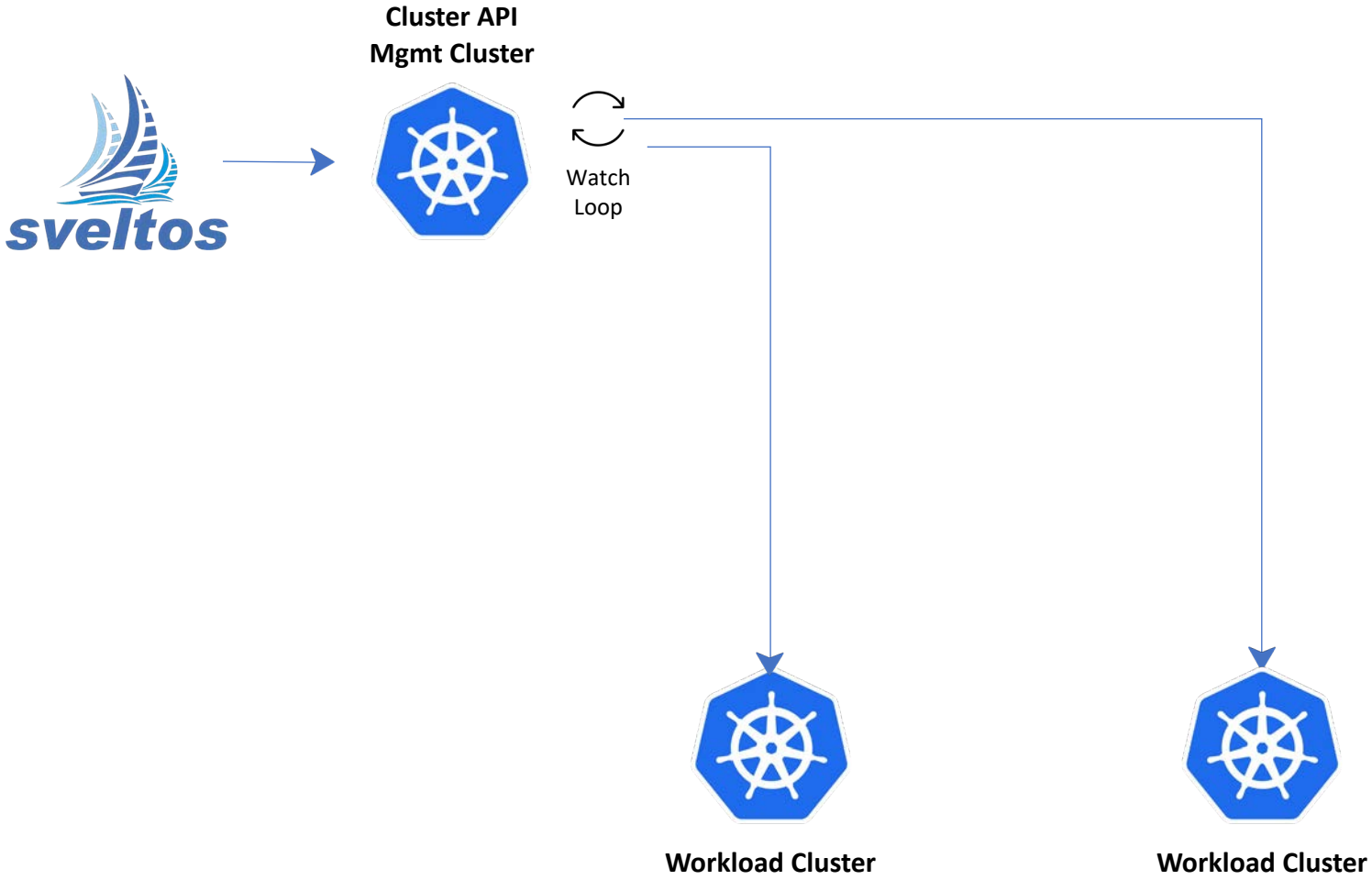
- ***clusterSelector:*** selects set of managed clusters;
- ***helmCharts***: list of helm charts to be deployed in the clusters matching clusterSelector;
- **kustomizationRefs**: : list of sources containing kustomization files. Resources will be deployed in the clusters matching clusterSelector;
- ***policyRefs***: list of ConfigMaps/Secrets containing the Kubernetes resources to be deployed in the clusters matching clusterSelector.

ConfigMap with YAML

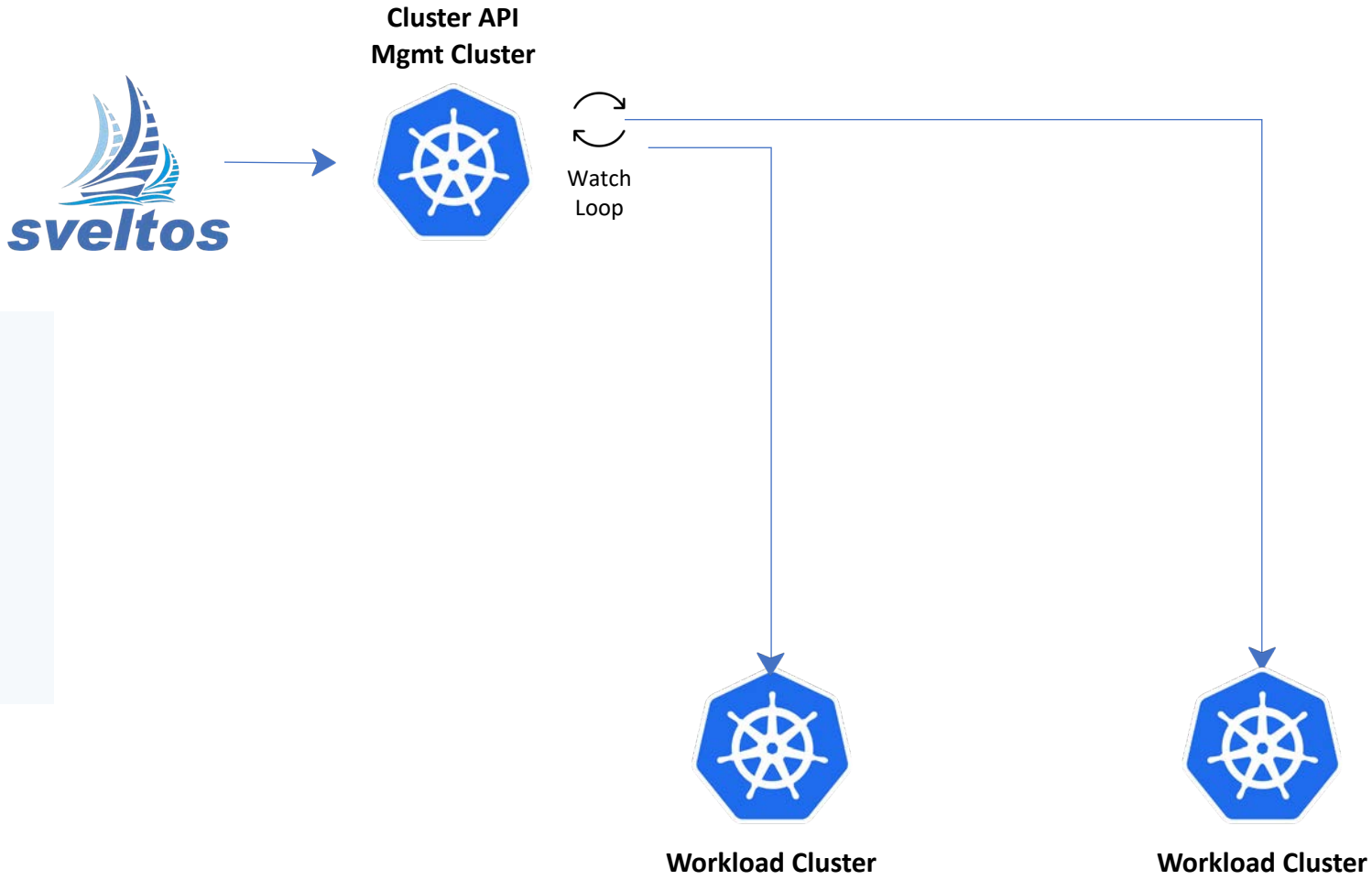
```yaml
apiVersion: v1
kind: ConfigMap
metadata:
  name: contour-gateway
  namespace: default
data:
  gatewayclass.yaml: |
    kind: GatewayClass
    apiVersion: gateway.networking.k8s.io/v1beta1
    metadata:
      name: contour
    spec:
      controllerName: projectcontour.io/projectcontour/contour
  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



Project Sveltos - Policy Driven Software Lifecycle Mgmt



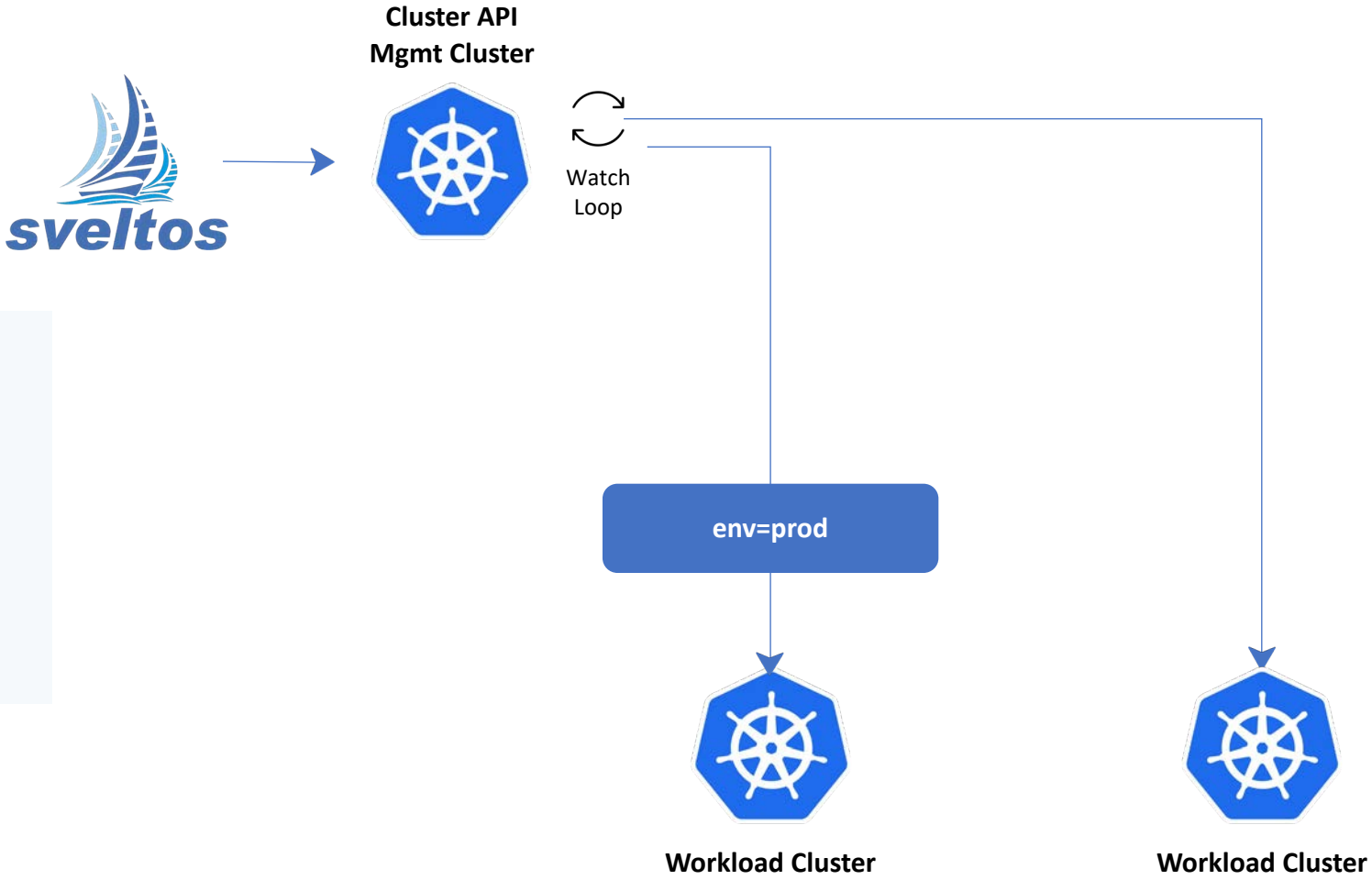
kubectl apply -f …

```yaml
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 - Policy Driven Software Lifecycle Mgmt

kubectl apply -f ...

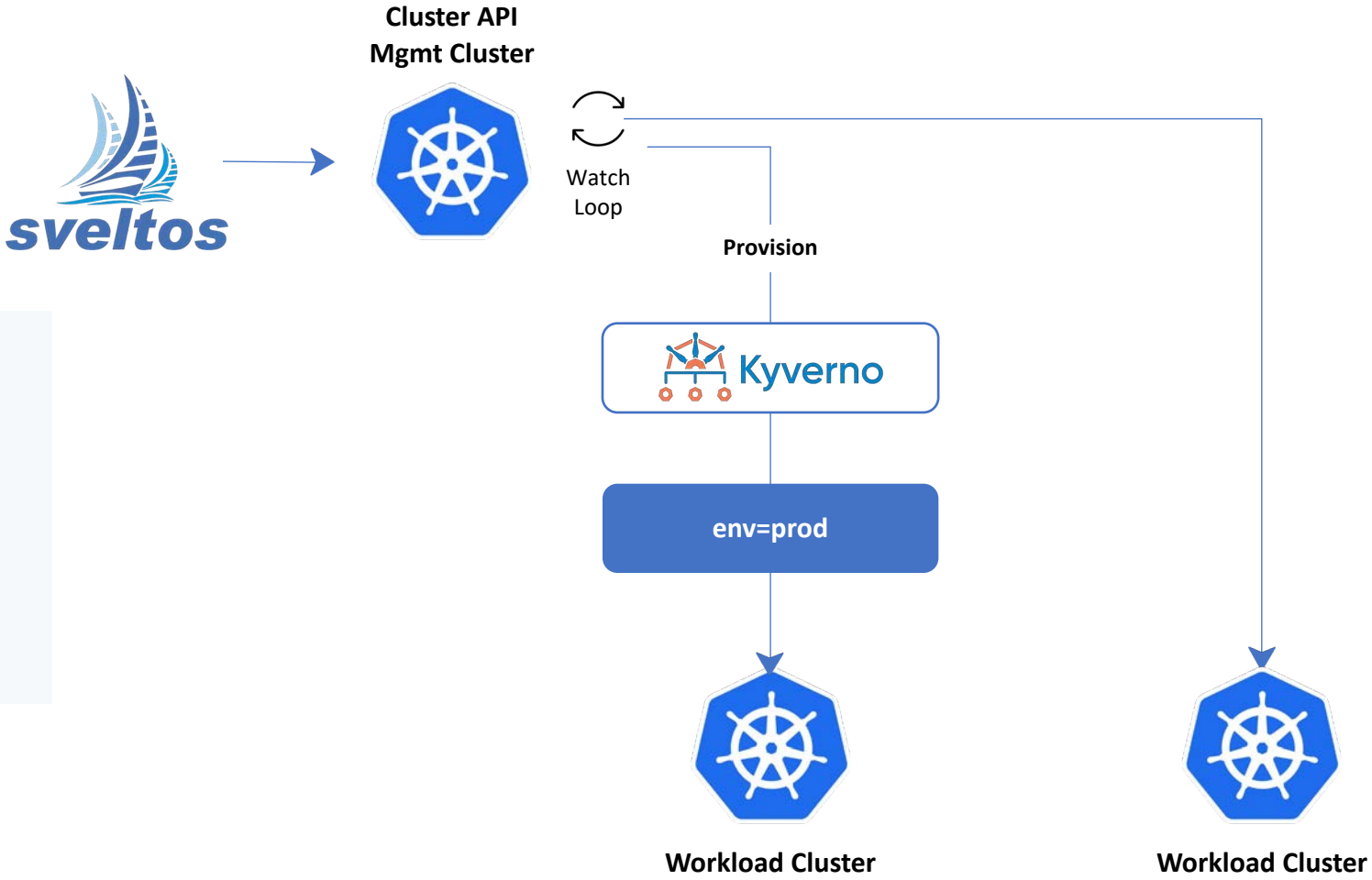
```yaml
apiVersion: config.projectsveltos.io/v1alpha1
kind: ClusterProfile
metadata:
  name: demo
spec:
  clusterSelector: env=prod
  syncMode: Continuous
  helmCharts:
  - repositoryURL: https://kyverno.github.io/kyverno/
    repositoryName: kyverno
    chartName: kyverno/kyverno
    chartVersion: v2.5.0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



Project Sveltos - Policy Driven Software Lifecycle Mgmt

kubectl apply -f ...

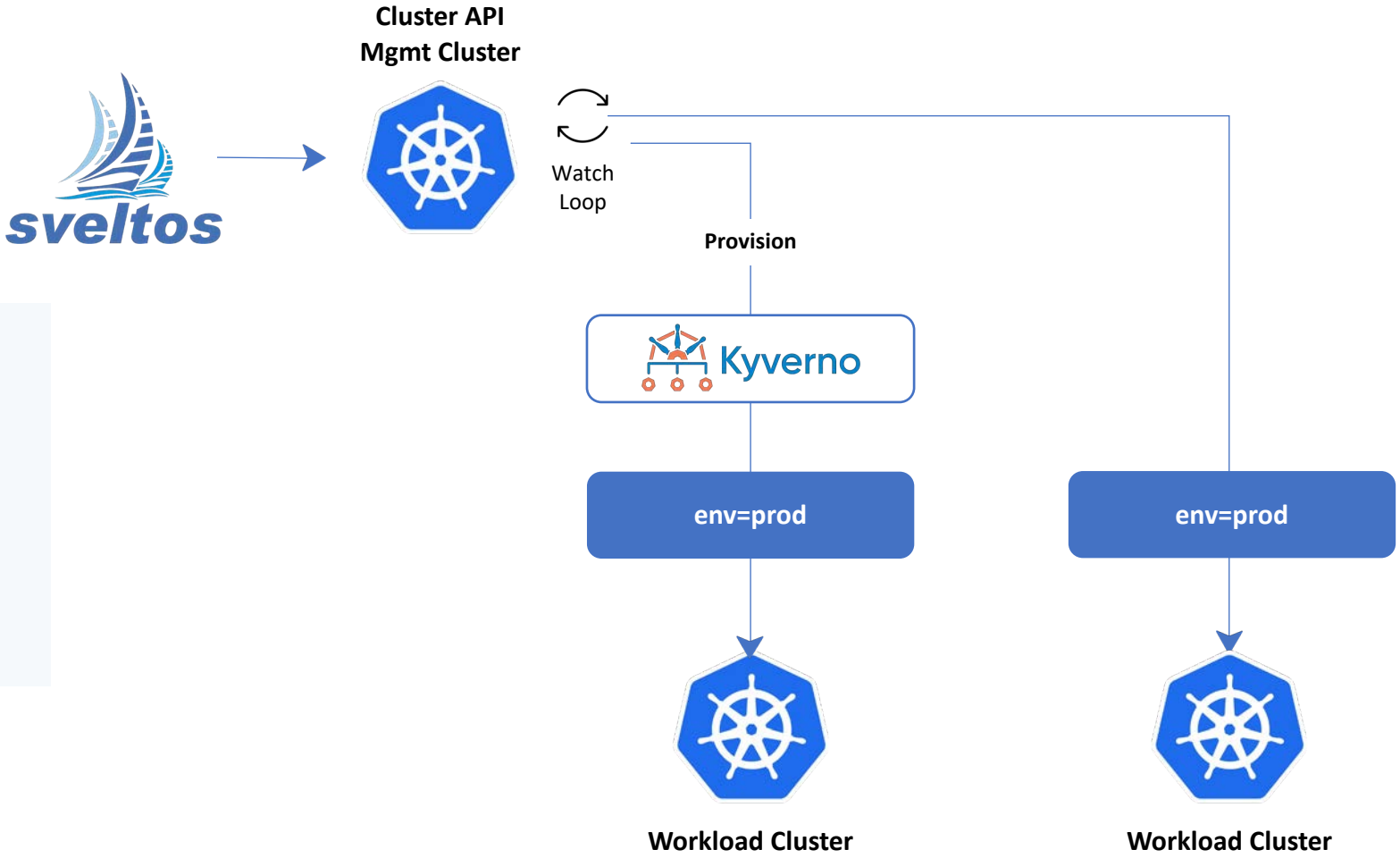
```yaml
apiVersion: config.projectsveltos.io/v1alpha1
kind: ClusterProfile
metadata:
  name: demo
spec:
  clusterSelector: env=prod
  syncMode: Continuous
  helmCharts:
  - repositoryURL: https://kyverno.github.io/kyverno/
    repositoryName: kyverno
    chartName: kyverno/kyverno
    chartVersion: v2.5.0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



Project Sveltos - Policy Driven Software Lifecycle Mgmt

kubectl apply -f ...

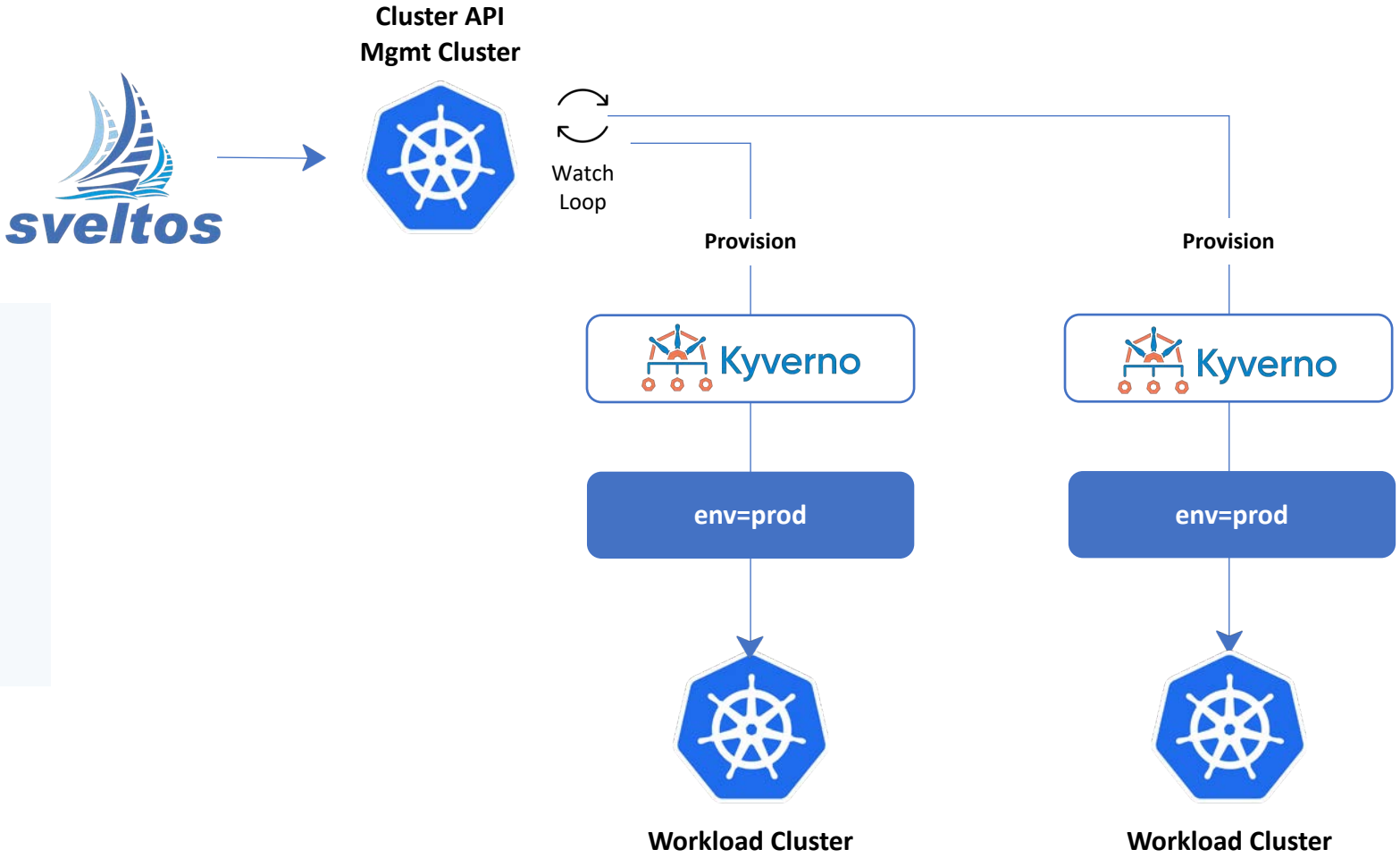
```yaml
apiVersion: config.projectsveltos.io/v1alpha1
kind: ClusterProfile
metadata:
  name: demo
spec:
  clusterSelector: env=prod
  syncMode: Continuous
  helmCharts:
  - repositoryURL: https://kyverno.github.io/kyverno/
    repositoryName: kyverno
    chartName: kyverno/kyverno
    chartVersion: v2.5.0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



Project Sveltos - Policy Driven Software Lifecycle Mgmt

kubectl apply -f ...

```yaml
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

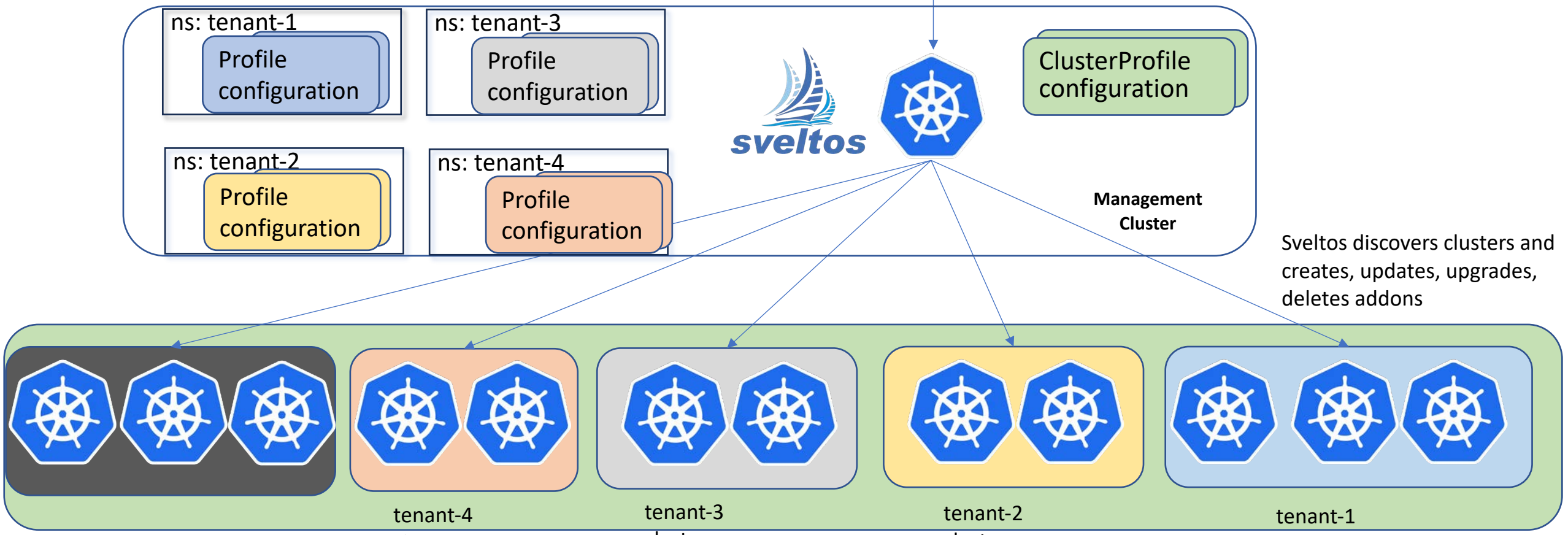
Following YAML instructs Sveltos to fetch the Secret instance *autoscaler* in the namespace *default* and make it available to the template with the keyword AutoscalerSecret

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.

Profile can match
clusters in same namespace only
Tenant admins maintain those

Declarative configuration (which add-ons to deploy and where)

ClusterProfile can match
clusters across all namespaces
Platform admin maintains those



Project Sveltos – Multi-tenancy

Roles:

- **Platform admin:**
  - Creates/upgrade/deletes managed clusters;
  - Has cluster-admin access to all managed clusters;
  - Manages tenants by assigning clusters and/or namespaces in shared clusters.

- **Tenant admin**:
  - Has admin access to the clusters/namespaces assigned to it by the platform admin;
  - Manages tenant applications in such clusters/namespaces from the management cluster.

Project Sveltos – Multi-tenancy

- **RoleRequest:**
  - CRD introduced by Sveltos to allow platform admin to grant permissions to tenant admins;

```yaml
apiVersion: lib.projectsveltos.io/v1alpha1
kind: RoleRequest
metadata:
  name: full-access
spec:
  clusterSelector: dep=eng
  admin: eng
  roleRefs:
  - name: full-access
    namespace: default
    kind: ConfigMap
```

- ***ClusterSelector****:* selects set of managed clusters;
- ***Admin***: tenant admin to whom permissions are granted in the selected managed clusters;
- ***RoleRefs***: list of ConfigMaps/Secrets containing the Kubernetes ClusterRoles/Roles with admin RBACs.

Project Sveltos – Multi-tenancy

When a ClusterProfile is created by a tenant admin, Sveltos expects following label is present:

- projectsveltos.io/admin-name: <admin>

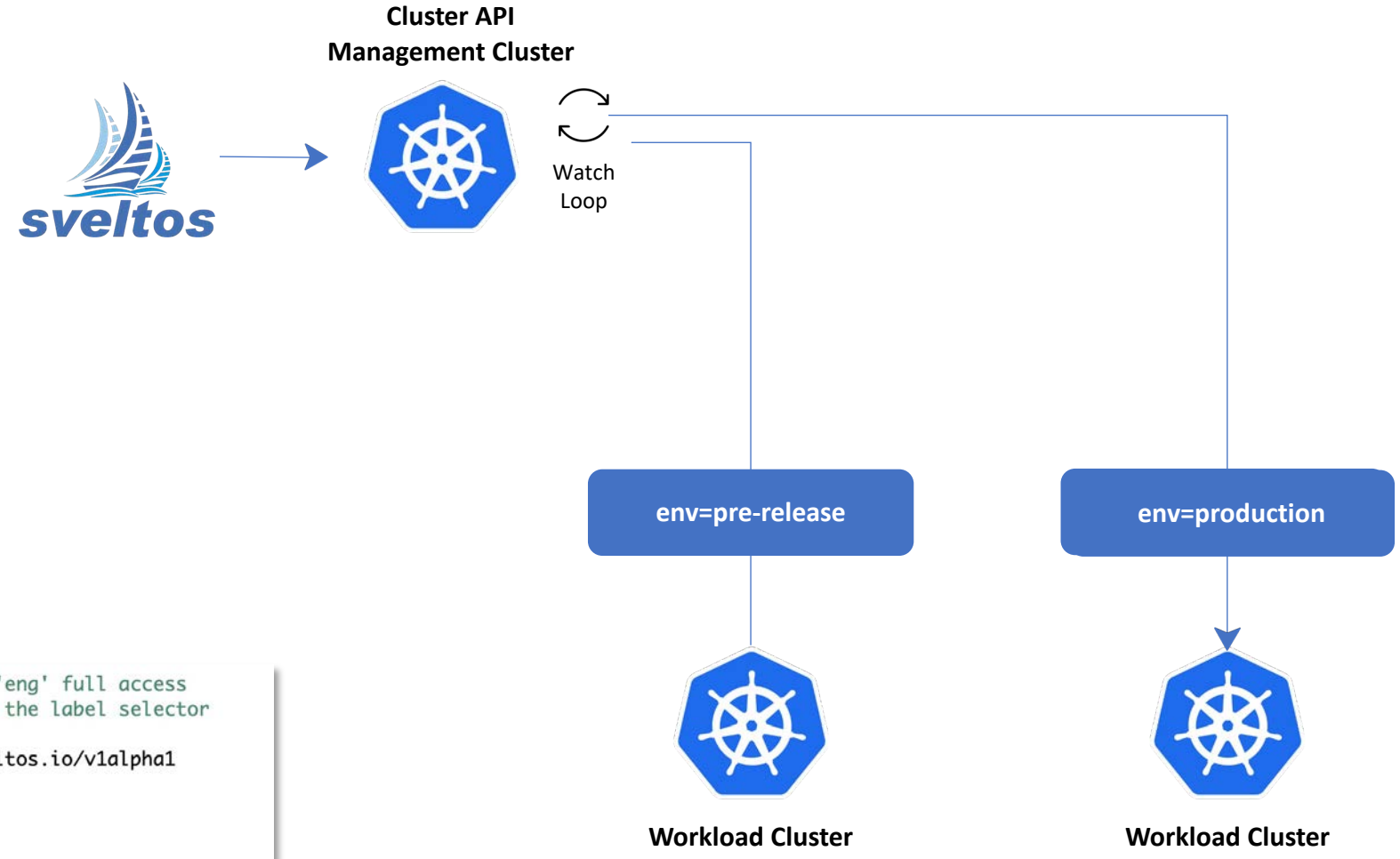
<admin> must match the RoleRequest.Spec.Admin

Platform admin can use an admission controller (Sveltos provides one based on Kyverno)

```yaml
apiVersion: kyverno.io/v1
kind: ClusterPolicy
metadata:
  name: add-labels
  annotations:
    policies.kyverno.io/title: Add Labels
spec:
  validationFailureAction: enforce
  background: false
  rules:
  - name: add-labels
    match:
      resources:
        kinds:
        - ClusterProfile
    mutate:
      patchStrategicMerge:
        metadata:
          labels:
            projectsveltos.io/admin-name:
                "{{serviceAccountName}}"
```

- For instance, if tenant admins are represented by ServiceAccounts in the management cluster, this Kyverno ClusterPolicy adds proper label

Project Sveltos – Multi-tenancy



platform admin: kubectl apply -f ...

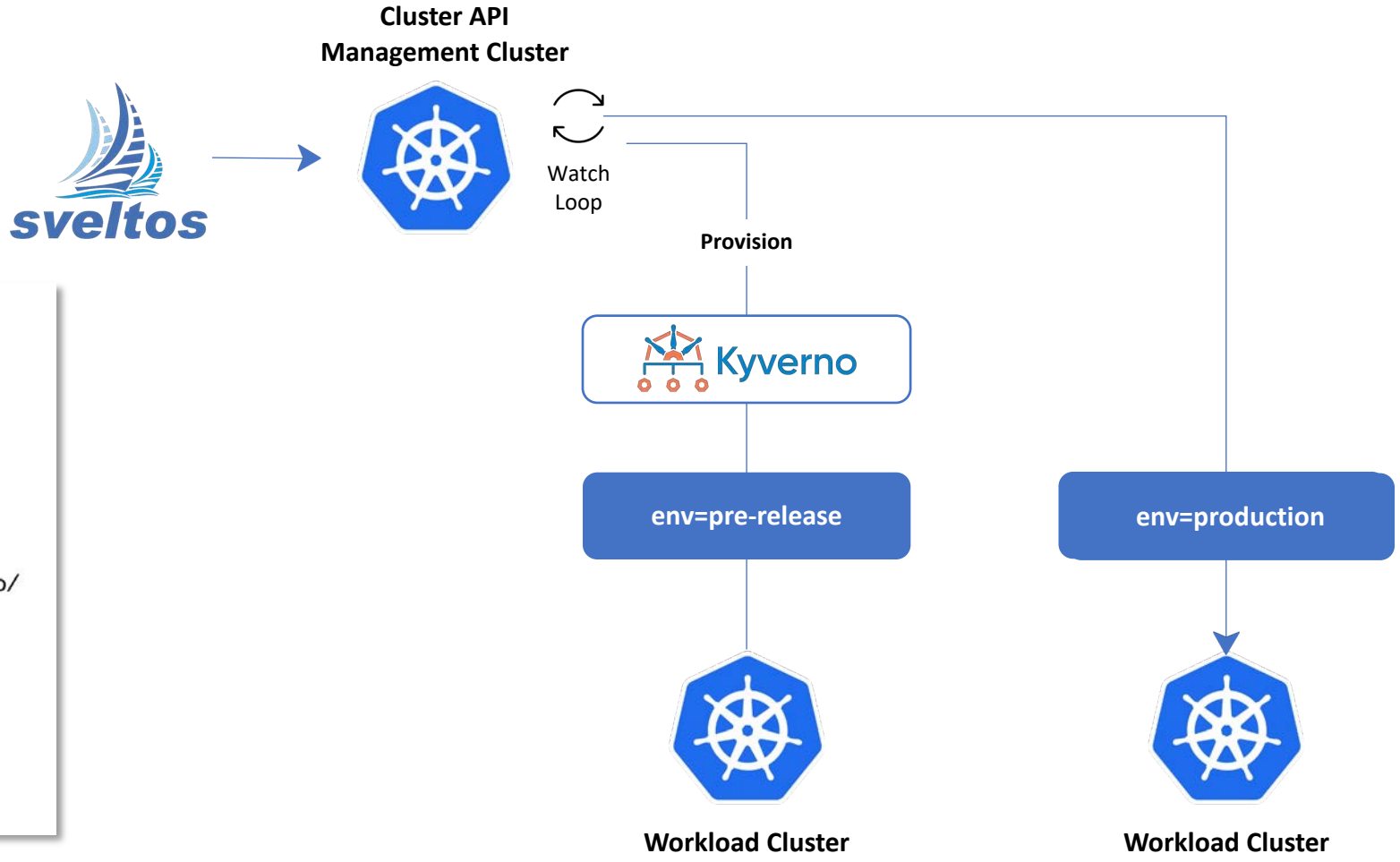
```yaml
# ConfigMap contains a ClusterRole which gives
# full access
apiVersion: v1
kind: ConfigMap
metadata:
  name: full-access
  namespace: default
data:
  role.yaml: |
    apiVersion: rbac.authorization.k8s.io/v1
    kind: ClusterRole
    metadata:
      name: eng-full-access
    rules:
    - apiGroups: ["*"]
      resources: ["*"]
      verbs: ["*"]
```

```yaml
# RoleRequest gives admin 'eng' full access
# to all clusters matching the label selector
# env=pre-release
apiVersion: lib.projectsveltos.io/v1alpha1
kind: RoleRequest
metadata:
  name: eng-full-access
spec:
  clusterSelector: env=pre-release
  admin: eng
  roleRefs:
  - name: full-access
    namespace: default
    kind: ConfigMap
```

Project Sveltos – Multi-tenancy

eng admin: kubectl apply -f …

```yaml
apiVersion: config.projectsveltos.io/v1alpha1
kind: ClusterProfile
metadata:
  name: deploy-kyverno
  labels:
    projectsveltos.io/admin-name: eng
spec:
  clusterSelector: env=pre-release
  syncMode: Continuous
  helmCharts:
  - repositoryURL:    https://kyverno.github.io/kyverno/
    repositoryName:   kyverno
    chartName:        kyverno/kyverno
    chartVersion:     v2.6.0
    releaseName:      kyverno-latest
    releaseNamespace: kyverno
    helmChartAction:  Install
```

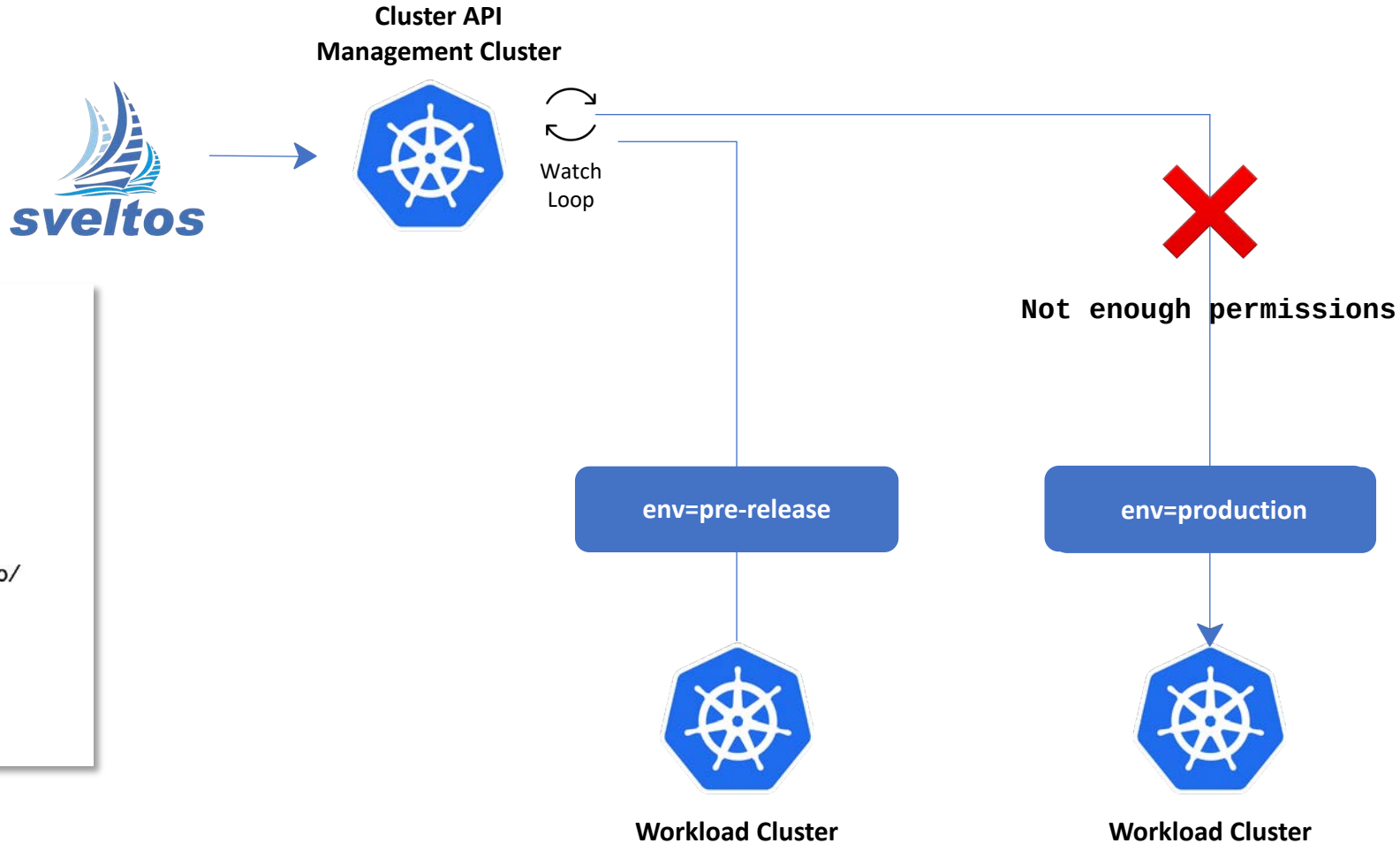


Admin eng gets full access to cluster

Project Sveltos – Multi-tenancy

eng admin: kubectl apply -f …

```yaml
apiVersion: config.projectsveltos.io/v1alpha1
kind: ClusterProfile
metadata:
  name: deploy-kyverno
  labels:
    projectsveltos.io/admin-name: eng
spec:
  clusterSelector: env=production
  syncMode: Continuous
  helmCharts:
  - repositoryURL:    https://kyverno.github.io/kyverno/
    repositoryName:   kyverno
    chartName:        kyverno/kyverno
    chartVersion:     v2.6.0
    releaseName:      kyverno-latest
    releaseNamespace: kyverno
    helmChartAction:  Install
```



Admin eng gets full access to cluster