Multitenant Clusters with Hierarchical Namespaces

Adrian Ludwin
Wednesday, August 19, 2020
aludwin@google.com

Google Cloud

Overview

Introduce the concept of **Hierarchical Namespaces**, explain how you can use them in your organization, and how you can help contribute.

Google Cloud

Topics

1. Why use multitenancy?
2. All about namespaces
3. Hierarchical Namespace Controller (HNC) “demo”
4. Advanced HNC topics
5. Next steps

Google Cloud

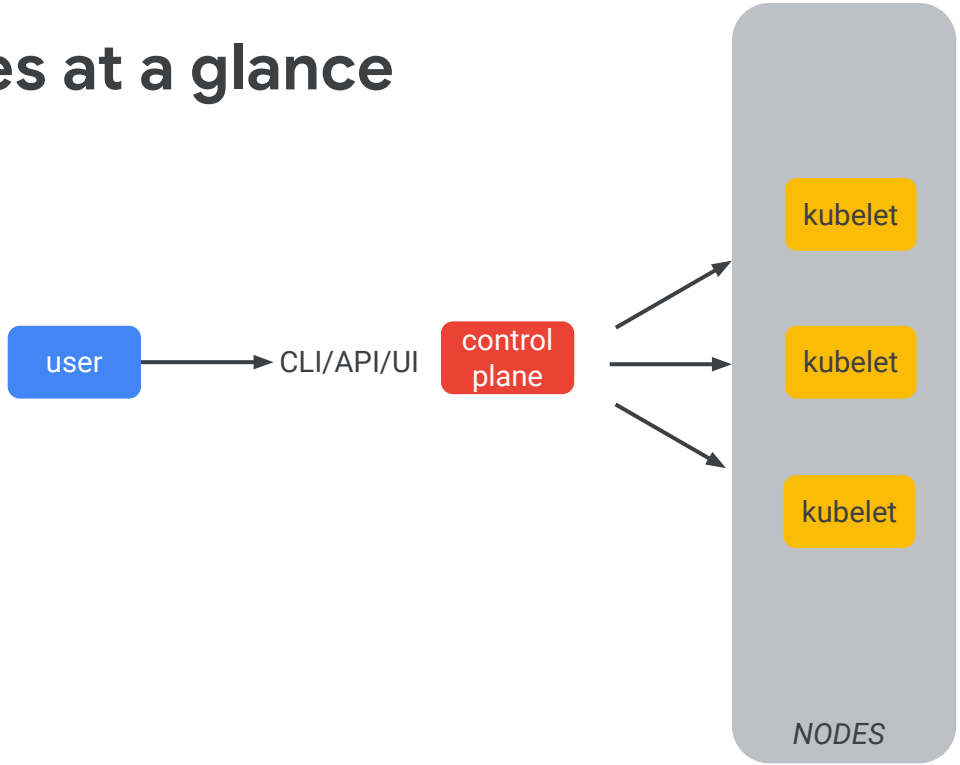
Why use multitenancy?

Google Cloud

What companies care about

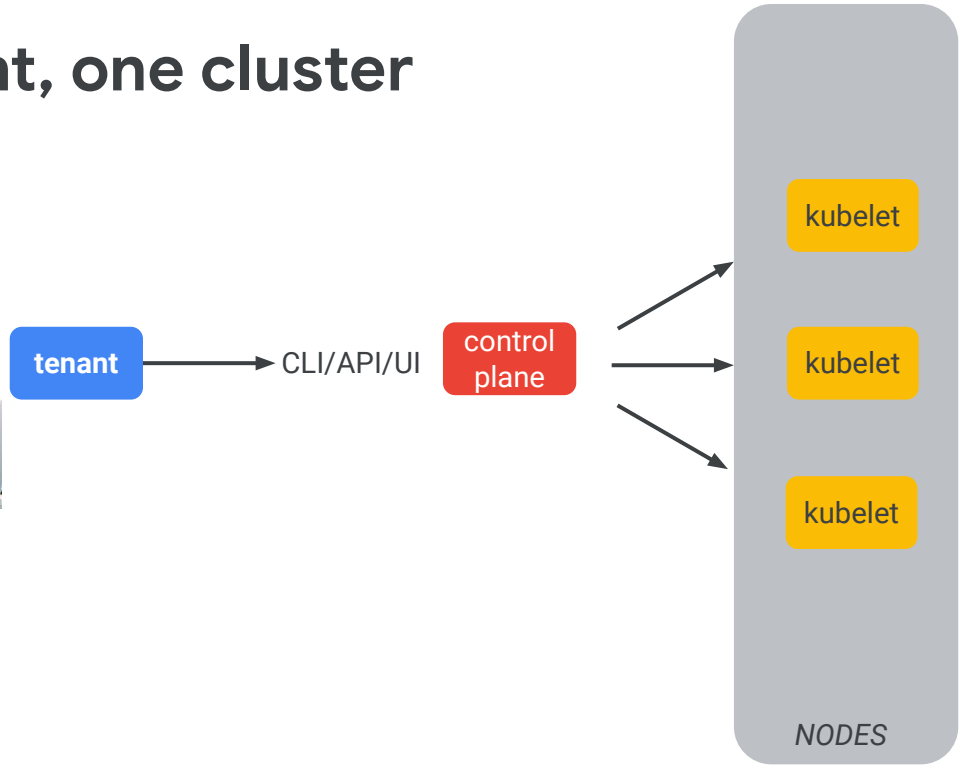
Google Cloud

Kubernetes at a glance



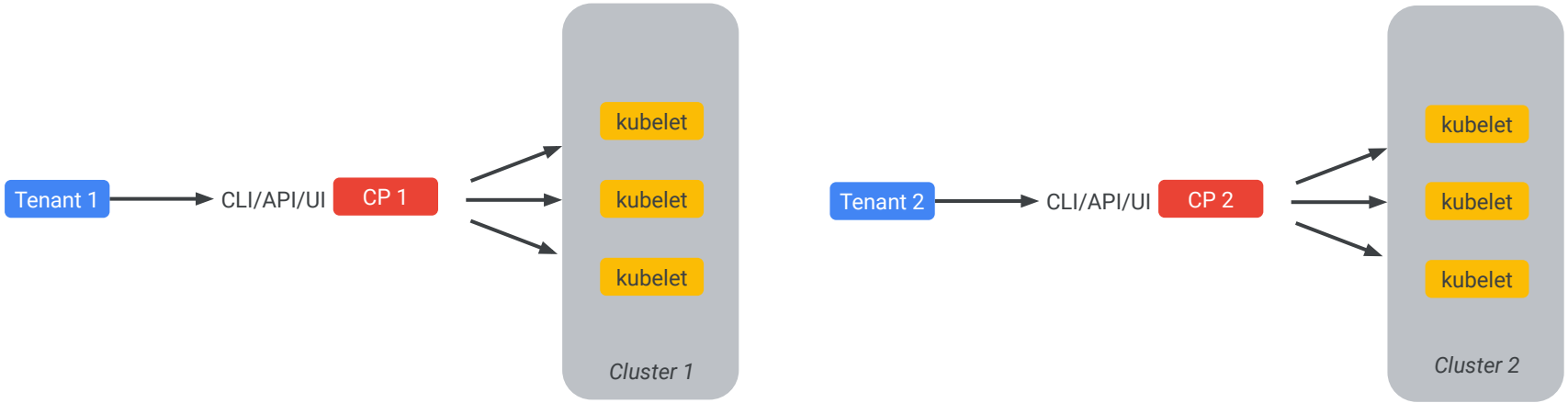
Google Cloud

One tenant, one cluster



Google Cloud

Multiple tenants, multiple clusters?



Google Cloud

Kubesprawl: how does this scale?

Google Cloud

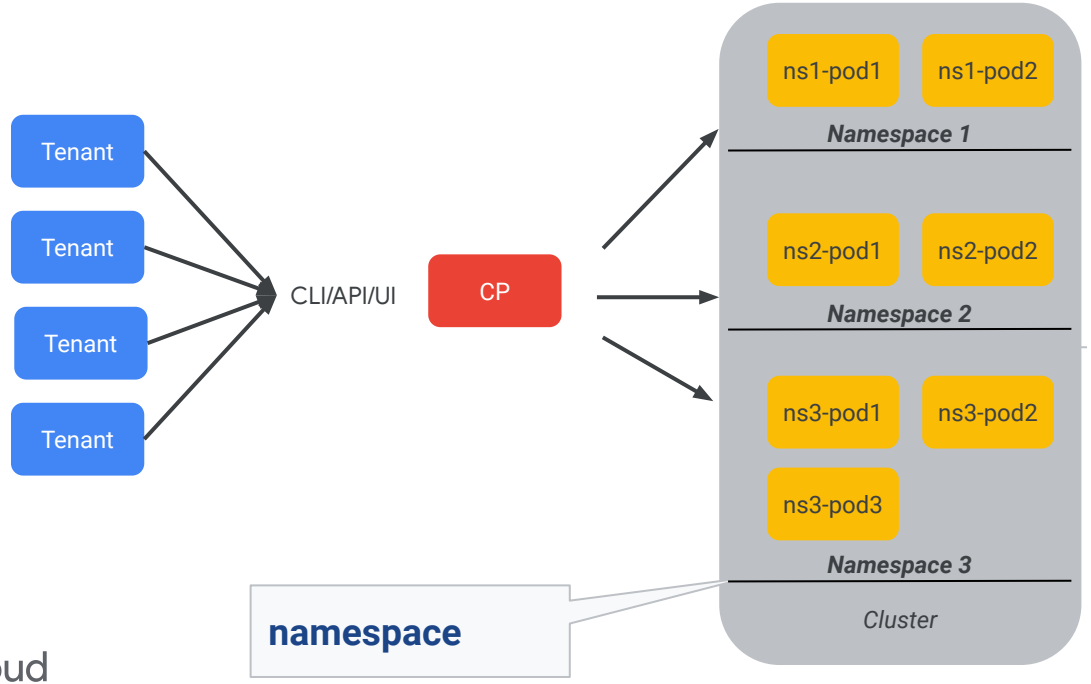
About wg-multitenancy

The Multitenancy Working Group was formed to categorize and solve multitenancy problems in the Kubernetes ecosystem. Current projects include HNC (this presentation), Virtual Clusters and the multitenancy benchmark project.

There's more at the end of this presentation, but TL;DR: github.com/kubernetes-sigs/multi-tenancy

Google Cloud

Alternative: many tenants, one cluster



Google Cloud

All about namespaces

Google Cloud

Namespaces

Namespaces are the primary unit of tenancy in Kubernetes.

By themselves, they don't do much except organize other objects - but almost all policies require or support namespaces by default.

Google Cloud

Some security features *require* namespaces

Service accounts and Secrets are freely usable within a namespace

- *Anyone* with permission to deploy a pod in a namespace can use *any* Secret or run as *any* SA
- This is why it's best practice to segregate workloads and teams in different namespaces if their secrets/SAs are sensitive

Note: namespaces *only* isolate the control plane, not the data plane

- A malicious workload that escapes its container can attack anything else in the cluster
- Use sandboxing (e.g. gVisor, Kata) to defend the data plane

Other features provide *support* for namespaces

RBAC works best at the namespace level:

- Only way to scope creation
- Least brittle way to scope other operations

Also applies to most other policies:

- Resource quotas and limit ranges only apply to namespaces
- Network policies can be more finely targeted but use namespace boundaries by default
    - Caveat: requires labels, which are not secure

What if you want policies *across* namespaces?

Usually, you need a tool and source-of-truth *outside* of Kubernetes:

- Flux, Argo, GKE Config Sync, Anthos Config Management

Alternatively, some in-cluster solutions add “accounts” or “tenants”

- Kiosk or the Tenant CRD (another wg-multitenancy project)

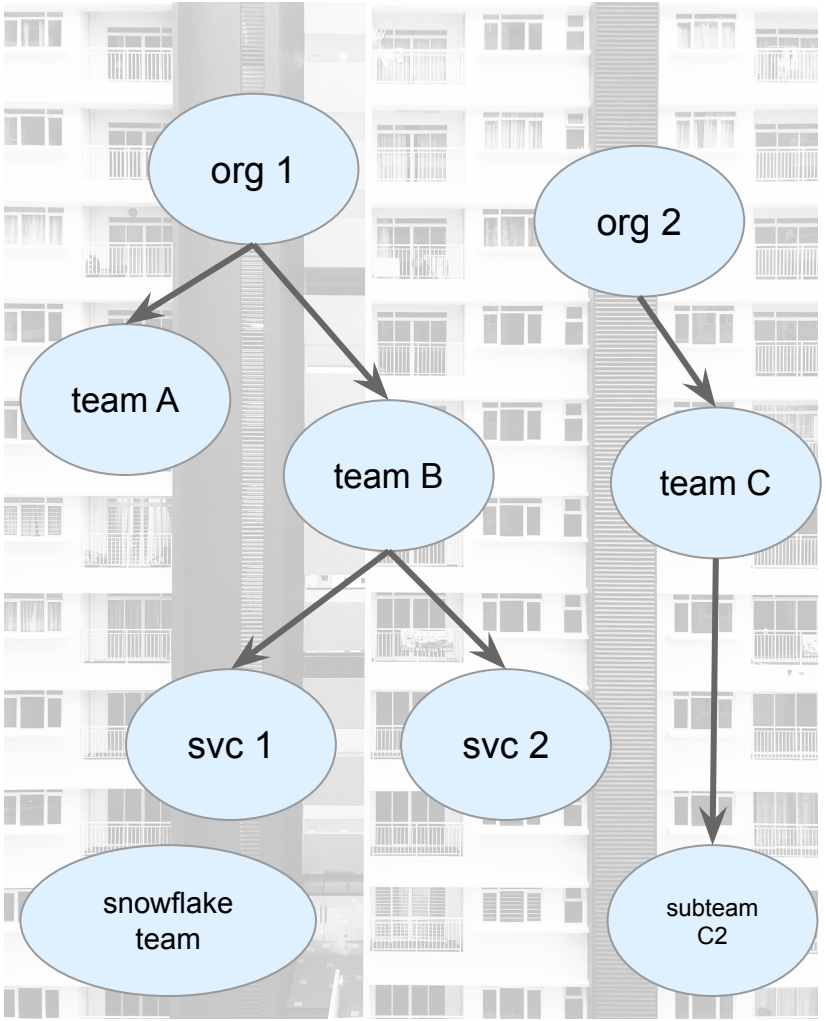
We felt there was a need for a solution that:

- Was fully Kubernetes-native (i.e. no dependencies on Git)
- Extended existing concepts rather than add new ones

Hierarchical namespaces

An incubating OSS standard to express *ownership,* which allows for admin delegation and cascading policies.

Hierarchical Namespaces are provided by the [Hierarchical Namespace Controller (HNC)].



Google Cloud

Properties of hierarchical namespaces

Entirely Kubernetes-native, but compatible with existing Gitops tools (e.g. Flux).

Builds on regular Kubernetes namespaces, plus:

- Delegated subnamespace creation without cluster privileges
- Cascading policies, secrets, configmaps, etc.
- Trusted labels for policy application (e.g. Network Policies)
- Easy to extend and integrate
    - Including to build higher-level abstractions like “tenants” if desired

Hierarchical Namespace Controller (HNC) “demo”

Google Cloud

```
aludwin@aludwin0:~$ k edit configmanagement config-management
```

```
# Please edit the object below. Lines beginning with a '#' will be ignored,
# and an empty file will abort the edit. If an error occurs while saving this file will be
# reopened with the relevant failures.
#
apiVersion: configmanagement.gke.io/v1
kind: ConfigManagement
metadata:
  annotations:
    kubectl.kubernetes.io/last-applied-configuration: |
      {"apiVersion":"configmanagement.gke.io/v1","kind":"ConfigManagement","metadata":{"annotations":{},"name":"c
onfig-management"},"spec":{"hierarchyController":{"enabled":true}}}
  creationTimestamp: "2020-07-27T17:52:42Z"
  finalizers:
  - operator.configmanagement.gke.io
  generation: 8
  name: config-management
  resourceVersion: "4116137"
  selfLink: /apis/configmanagement.gke.io/v1/configmanagements/config-management
  uid: 6f5c720a-edc7-4065-a5bf-81efd8a9ec45
spec:
  hierarchyController:
    enabled: true
  ApplicationController: null
  ConfigSyncDisableFSWatcher: false
  ConfigSyncLogLevel: 0
-- INSERT --                                                              21,19
```

```
aludwin@aludwin0:~$ k edit configmanagement config-management
configmanagement.configmanagement.gke.io/config-management edited
aludwin@aludwin0:~$
```

```
aludwin@aludwin0:~$ k edit configmanagement config-management
configmanagement.configmanagement.gke.io/config-management edited
aludwin@aludwin0:~$ k create ns normal-parent
```

Setup

```
aludwin@aludwin0:~$ k edit configmanagement config-management
configmanagement.configmanagement.gke.io/config-management edited
aludwin@aludwin0:~$ k create ns normal-parent
namespace/normal-parent created
aludwin@aludwin0:~$
```

```
aludwin@aludwin0:~$ k edit configmanagement config-management
configmanagement.configmanagement.gke.io/config-management edited
aludwin@aludwin0:~$ k create ns normal-parent
namespace/normal-parent created
aludwin@aludwin0:~$ k create ns normal-child
```

Setup

```
aludwin@aludwin0:~$ k edit configmanagement config-management
configmanagement.configmanagement.gke.io/config-management edited
aludwin@aludwin0:~$ k create ns normal-parent
namespace/normal-parent created
aludwin@aludwin0:~$ k create ns normal-child
namespace/normal-child created
aludwin@aludwin0:~$
```

Setup

```
aludwin@aludwin0:~$ k edit configmanagement config-management
configmanagement.configmanagement.gke.io/config-management edited
aludwin@aludwin0:~$ k create ns normal-parent
namespace/normal-parent created
aludwin@aludwin0:~$ k create ns normal-child
namespace/normal-child created
aludwin@aludwin0:~$ k get ns
```

Setup

```
aludwin@aludwin0:~$ k edit configmanagement config-management
configmanagement.configmanagement.gke.io/config-management edited
aludwin@aludwin0:~$ k create ns normal-parent
namespace/normal-parent created
aludwin@aludwin0:~$ k create ns normal-child
namespace/normal-child created
aludwin@aludwin0:~$ k get ns
NAME                          STATUS   AGE
config-management-system      Active   8m23s
default                       Active   10d
hnc-system                    Active   83s
kube-node-lease               Active   10d
kube-public                   Active   10d
kube-system                   Active   10d
normal-child                  Active   4s
normal-parent                 Active   10s
aludwin@aludwin0:~$
```

Setup

```
aludwin@aludwin0:~$ k edit configmanagement config-management
configmanagement.configmanagement.gke.io/config-management edited
aludwin@aludwin0:~$ k create ns normal-parent
namespace/normal-parent created
aludwin@aludwin0:~$ k create ns normal-child
namespace/normal-child created
aludwin@aludwin0:~$ k get ns
NAME                          STATUS   AGE
config-management-system      Active   8m23s
default                       Active   10d
hnc-system                    Active   83s
kube-node-lease               Active   10d
kube-public                   Active   10d
kube-system                   Active   10d
normal-child                  Active   4s
normal-parent                 Active   10s
aludwin@aludwin0:~$ k hns tree -A
```

```
aludwin@aludwin0:~$ k edit configmanagement config-management
configmanagement.configmanagement.gke.io/config-management edited
aludwin@aludwin0:~$ k create ns normal-parent
namespace/normal-parent created
aludwin@aludwin0:~$ k create ns normal-child
namespace/normal-child created
aludwin@aludwin0:~$ k get ns
NAME                          STATUS   AGE
config-management-system      Active   8m23s
default                       Active   10d
hnc-system                    Active   83s
kube-node-lease               Active   10d
kube-public                   Active   10d
kube-system                   Active   10d
normal-child                  Active   4s
normal-parent                 Active   10s
aludwin@aludwin0:~$ k hns tree -A
config-management-system
default
hnc-system
kube-node-lease
kube-public
kube-system
normal-child
normal-parent
aludwin@aludwin0:~$
```

```
aludwin@aludwin0:~$ k hns set normal-child --parent normal-parent
```

Full namespace hierarchy

```
aludwin@aludwin0:~$ k hns set normal-child --parent normal-parent
Setting the parent of normal-child to normal-parent
Succesfully updated 1 property of the hierarchical configuration of normal-child
aludwin@aludwin0:~$
```

Full namespace hierarchy

```
aludwin@aludwin0:~$ k hns set normal-child --parent normal-parent
Setting the parent of normal-child to normal-parent
Succesfully updated 1 property of the hierarchical configuration of normal-child
aludwin@aludwin0:~$ k hns tree -A
```

Full namespace hierarchy

```
aludwin@aludwin0:~$ k hns set normal-child --parent normal-parent
Setting the parent of normal-child to normal-parent
Succesfully updated 1 property of the hierarchical configuration of normal-child
aludwin@aludwin0:~$ k hns tree -A
config-management-system
default
hnc-system
kube-node-lease
kube-public
kube-system
normal-parent
└── normal-child
aludwin@aludwin0:~$
```

Full namespace hierarchy

```
aludwin@aludwin0:~$ k -n normal-parent create role parent-sre --verb=update --resource=deployments
```

Policy propagation

```
aludwin@aludwin0:~$ k -n normal-parent create role parent-sre --verb=update --resource=deployments
```

Policy propagation

```
aludwin@aludwin0:~$ k -n normal-parent create role parent-sre --verb=update --resource=deployments
role.rbac.authorization.k8s.io/parent-sre created
aludwin@aludwin0:~$
```

Policy propagation

```
aludwin@aludwin0:~$ k -n normal-parent create role parent-sre --verb=update --resource=deployments
role.rbac.authorization.k8s.io/parent-sre created
aludwin@aludwin0:~$ k get -n normal-child -oyaml role | head -n10
```

Policy propagation

```
aludwin@aludwin0:~$ k -n normal-parent create role parent-sre --verb=update --resource=deployments
role.rbac.authorization.k8s.io/parent-sre created
aludwin@aludwin0:~$ k get -n normal-child -oyaml role | head -n10
apiVersion: v1
items:
- apiVersion: rbac.authorization.k8s.io/v1
  kind: Role
  metadata:
    creationTimestamp: "2020-07-27T18:08:01Z"
    labels:
      hnc.x-k8s.io/inheritedFrom: normal-parent
    name: parent-sre
    namespace: normal-child
aludwin@aludwin0:~$
```

Policy propagation

```
aludwin@aludwin0:~$ k hns create sub-child -n normal-parent
```

Subnamespace hierarchy

```
aludwin@aludwin0:~$ k hns create sub-child -n normal-parent
Successfully created "sub-child" subnamespace anchor in "normal-parent" namespace
aludwin@aludwin0:~$
```

Subnamespace hierarchy

```
aludwin@aludwin0:~$ k hns create sub-child -n normal-parent
Successfully created "sub-child" subnamespace anchor in "normal-parent" namespace
aludwin@aludwin0:~$ k hns tree normal-parent
```

Subnamespace hierarchy

```
aludwin@aludwin0:~$ k hns create sub-child -n normal-parent
Successfully created "sub-child" subnamespace anchor in "normal-parent" namespace
aludwin@aludwin0:~$ k hns tree normal-parent
normal-parent
├── normal-child
└── sub-child
aludwin@aludwin0:~$
```

Subnamespace hierarchy

```
aludwin@aludwin0:~$ k describe ns sub-child
```

Extension: tree labels

```
aludwin@aludwin0:~$ k describe ns sub-child
Name:         sub-child
Labels:       normal-parent.tree.hnc.x-k8s.io/depth=1
              sub-child.tree.hnc.x-k8s.io/depth=0
Annotations:  hnc.x-k8s.io/subnamespaceOf: normal-parent
Status:       Active

Resource Quotas
 Name:                         gke-resource-quotas
 Resource                      Used  Hard
 --------                      ---   ---
 count/ingresses.extensions    0     100
 count/jobs.batch              0     5k
 pods                          0     1500
 services                      0     500

No resource limits.
aludwin@aludwin0:~$
```

Extension: tree labels

```
aludwin@aludwin0:~$ tree ~/git/hnc-gitops/policies/hnc-acm-1/namespaces/
```

Extension: external hierarchy

```
aludwin@aludwin0:~$ tree ~/git/hnc-gitops/policies/hnc-acm-1/namespaces/
/usr/local/google/home/aludwin/git/hnc-gitops/policies/hnc-acm-1/namespaces/
└── acme-org
    ├── eng
    │   ├── eng-sre-rolebinding.yaml
    │   ├── np-allow-eng.yaml
    │   ├── team-a
    │   │   ├── namespace.yaml
    │   │   └── team-a-rolebinding.yaml
    │   └── team-b
    │       ├── namespace.yaml
    │       └── team-b-rolebinding.yaml
    ├── np-deny.yaml
    ├── sre-rolebinding.yaml
    └── team-c
        ├── namespace.yaml
        └── team-c-rolebinding.yaml

5 directories, 10 files
aludwin@aludwin0:~$
```

Extension: external hierarchy

```
aludwin@aludwin0:~$ tree ~/git/hnc-gitops/policies/hnc-acm-1/namespaces/
/usr/local/google/home/aludwin/git/hnc-gitops/policies/hnc-acm-1/namespaces/
└── acme-org
    ├── eng
    │   ├── eng-sre-rolebinding.yaml
    │   ├── np-allow-eng.yaml
    │   ├── team-a
    │   │   ├── namespace.yaml
    │   │   └── team-a-rolebinding.yaml
    │   └── team-b
    │       ├── namespace.yaml
    │       └── team-b-rolebinding.yaml
    ├── np-deny.yaml
    ├── sre-rolebinding.yaml
    └── team-c
        ├── namespace.yaml
        └── team-c-rolebinding.yaml

5 directories, 10 files
aludwin@aludwin0:~$ k get ns | grep team
```

Extension: external hierarchy

```
aludwin@aludwin0:~$ tree ~/git/hnc-gitops/policies/hnc-acm-1/namespaces/
/usr/local/google/home/aludwin/git/hnc-gitops/policies/hnc-acm-1/namespaces/
└── acme-org
    ├── eng
    │   ├── eng-sre-rolebinding.yaml
    │   ├── np-allow-eng.yaml
    │   ├── team-a
    │   │   ├── namespace.yaml
    │   │   └── team-a-rolebinding.yaml
    │   └── team-b
    │       ├── namespace.yaml
    │       └── team-b-rolebinding.yaml
    ├── np-deny.yaml
    ├── sre-rolebinding.yaml
    └── team-c
        ├── namespace.yaml
        └── team-c-rolebinding.yaml

5 directories, 10 files
aludwin@aludwin0:~$ k get ns | grep team
team-a                    Active   99s
team-b                    Active   100s
team-c                    Active   99s
aludwin@aludwin0:~$
```

Extension: external hierarchy

```
aludwin@aludwin0:~$ k describe ns team-b | head -n20
```

Extension: external hierarchy

```
aludwin@aludwin0:~$ k describe ns team-b | head -n20
Name:         team-b
Labels:       acme-org.tree.hnc.x-k8s.io/depth=2
              app.kubernetes.io/managed-by=configmanagement.gke.io
              config-sync-root.tree.hnc.x-k8s.io/depth=3
              eng.tree.hnc.x-k8s.io/depth=1
              team-b.tree.hnc.x-k8s.io/depth=0
Annotations:  configmanagement.gke.io/declared-config:
                {"apiVersion":"v1","kind":"Namespace","metadata":{"annotations":{"configmanagement.gke.io/managed
":"enabled","configmanagement.gke.io/sour...
              configmanagement.gke.io/managed: enabled
              configmanagement.gke.io/source-path: namespaces/acme-org/eng/team-b/namespace.yaml
              configmanagement.gke.io/token: b8640420399fbdd352a3885e5f156f2a94d2a7e4
              hnc.x-k8s.io/managedBy: configmanagement.gke.io
Status:       Active

Resource Quotas
 Name:                     gke-resource-quotas
 Resource                  Used  Hard
 --------                  ---   ---
 count/ingresses.extensions  0     100
 count/jobs.batch            0     5k
aludwin@aludwin0:~$
```

Extension: external hierarchy

```
aludwin@aludwin0:~$ k hns create svc1 -n team-b
```

Extension: Stackdriver logs

```
aludwin@aludwin0:~$ k hns create svc1 -n team-b
Successfully created "svc1" subnamespace anchor in "team-b" namespace
aludwin@aludwin0:~$
```

Extension: Stackdriver logs

```
aludwin@aludwin0:~$ k hns create svc1 -n team-b
Successfully created "svc1" subnamespace anchor in "team-b" namespace
aludwin@aludwin0:~$ k run websvr -n svc1 --image=nginx --restart=Never --expose --port 80
```

Extension: Stackdriver logs

```
aludwin@aludwin0:~$ k hns create svc1 -n team-b
Successfully created "svc1" subnamespace anchor in "team-b" namespace
aludwin@aludwin0:~$ k run websvr -n svc1 --image=nginx --restart=Never --expose --port 80
service/websvr created
pod/websvr created
aludwin@aludwin0:~$
```

Extension: Stackdriver logs

```
aludwin@aludwin0:~$ k hns create svc1 -n team-b
Successfully created "svc1" subnamespace anchor in "team-b" namespace
aludwin@aludwin0:~$ k run websvr -n svc1 --image=nginx --restart=Never --expose --port 80
service/websvr created
pod/websvr created
aludwin@aludwin0:~$ k describe po websvr -n svc1 | head -n15
```

Extension: Stackdriver logs

```
aludwin@aludwin0:~$ k hns create svc1 -n team-b
Successfully created "svc1" subnamespace anchor in "team-b" namespace
aludwin@aludwin0:~$ k run websvr -n svc1 --image=nginx --restart=Never --expose --port 80
service/websvr created
pod/websvr created
aludwin@aludwin0:~$ k describe po websvr -n svc1 | head -n15
Name:         websvr
Namespace:    svc1
Priority:     0
Node:         gke-hnc-reg2-default-pool-40c02cc3-slvt/10.128.0.29
Start Time:   Mon, 27 Jul 2020 16:08:42 -0400
Labels:       acme-org.tree.hnc.x-k8s.io/depth=3
              config-sync-root.tree.hnc.x-k8s.io/depth=4
              eng.tree.hnc.x-k8s.io/depth=2
              run=websvr
              svc1.tree.hnc.x-k8s.io/depth=0
              team-b.tree.hnc.x-k8s.io/depth=1
Annotations:  <none>
Status:       Running
IP:           10.16.2.204
IPs:
aludwin@aludwin0:~$
```

Extension: Stackdriver logs

```
aludwin@aludwin0:~$ k hns create svc1 -n team-b
Successfully created "svc1" subnamespace anchor in "team-b" namespace
aludwin@aludwin0:~$ k run websvr -n svc1 --image=nginx --restart=Never --expose --port 80
service/websvr created
pod/websvr created
aludwin@aludwin0:~$ k describe po websvr -n svc1 | head -n15
Name:         websvr
Namespace:    svc1
Priority:     0
Node:         gke-hnc-reg2-default-pool-40c02cc3-slvt/10.128.0.29
Start Time:   Mon, 27 Jul 2020 16:08:42 -0400
Labels:       acme-org.tree.hnc.x-k8s.io/depth=3
              config-sync-root.tree.hnc.x-k8s.io/depth=4
              eng.tree.hnc.x-k8s.io/depth=2
              run=websvr
              svc1.tree.hnc.x-k8s.io/depth=0
              team-b.tree.hnc.x-k8s.io/depth=1
Annotations:  <none>
Status:       Running
IP:           10.16.2.204
IPs:
aludwin@aludwin0:~$
```

Extension: Stackdriver logs

```
aludwin@aludwin0:~$ gcloud logging read 'labels.k8s-pod/eng_tree_hnc_x-k8s_io/depth!='' AND resource.type=k8s_con
tainer" | head -n23
```

Extension: Stackdriver logs

```
aludwin@aludwin0:~$ gcloud logging read "labels.k8s-pod/eng_tree_hnc_x-k8s_io/depth!='' AND resource.type=k8s_con
tainer" | head -n23
---
insertId: zobc220a86lvzb6xr
labels:
  k8s-pod/acme-org_tree_hnc_x-k8s_io/depth: '3'
  k8s-pod/config-sync-root_tree_hnc_x-k8s_io/depth: '4'
  k8s-pod/eng_tree_hnc_x-k8s_io/depth: '2'
  k8s-pod/run: websvr
  k8s-pod/svc1_tree_hnc_x-k8s_io/depth: '0'
  k8s-pod/team-b_tree_hnc_x-k8s_io/depth: '1'
logName: projects/aludwin-1/logs/stdout
receiveTimestamp: '2020-07-28T01:41:07.096565692Z'
resource:
  labels:
    cluster_name: hnc-reg2
    container_name: websvr
    location: us-central1-c
    namespace_name: svc1
    pod_name: websvr
    project_id: aludwin-1
  type: k8s_container
severity: INFO
textPayload: |
  /docker-entrypoint.sh: Configuration complete; ready for start up
aludwin@aludwin0:~$
```

Extension: Stackdriver logs

Advanced topics

04

Google Cloud

Other features of HNC

- Authorization checks before modifying the hierarchy
- Cascading deletion of subnamespaces
  - And safeties to prevent you from doing this accidentally
- Monitoring options
  - Metrics via OpenCensus
  - Status reporting in namespaced and cluster-wide objects
- Uninstallation support
  - Ensure your data isn’t deleted if you uninstall HNC

Emerging best practices

In dev clusters or simple prod environments:

- Give teams control over their own namespace hierarchy

In more complex, multicluster production environments:

- Safely distribute Secrets among related namespaces
- Allow teams to select their own CD tooling (e.g. Gitops)
- Restrict tools' service accounts to a namespace subtree

In summary: extend HNC's trusted base to create higher-level tools.

Next steps

Google Cloud

Getting hierarchical namespaces

Simple addon to any Kubernetes 1.15+ cluster:

- **OSS:** follow easy installation from our Github releases
  - github.com/kubernetes-sigs/multi-tenancy/incubator/hnc
  - Or search for “Hierarchical namespace controller”
- **GKE/Anthos:** enable Hierarchy Controller in Config Sync/ACM
  - Hierarchy Controller includes GCP-specific integrations

Follow the user guide and demos to get started.

Seeking contributors

We welcome contributors who are interested in features such as:

- Exceptions
  - Allow certain policies to be overridden
  - Create subnamespaces with default policies (self-serve)
- Per-subtree configuration
- Namespaced CRDs and admission webhooks
- Hierarchical resource quota
- Improved productionization (e.g. Prometheus support)

Plus testing and documentation help is always welcome!

Join the multitenancy working group

The multitenancy working group (wg-multitenancy) oversees:

- Hierarchical Namespaces
- Virtual Clusters and the Tenant CRD
- Multitenancy benchmarking (i.e. conformance)

Leadership: Tasha Drew (VMWare) and Sanjeev Rampal (Cisco).

We meet every second Tuesday - join us at
github.com/kubernetes-sigs/multi-tenancy.

Thanks!

Google Cloud