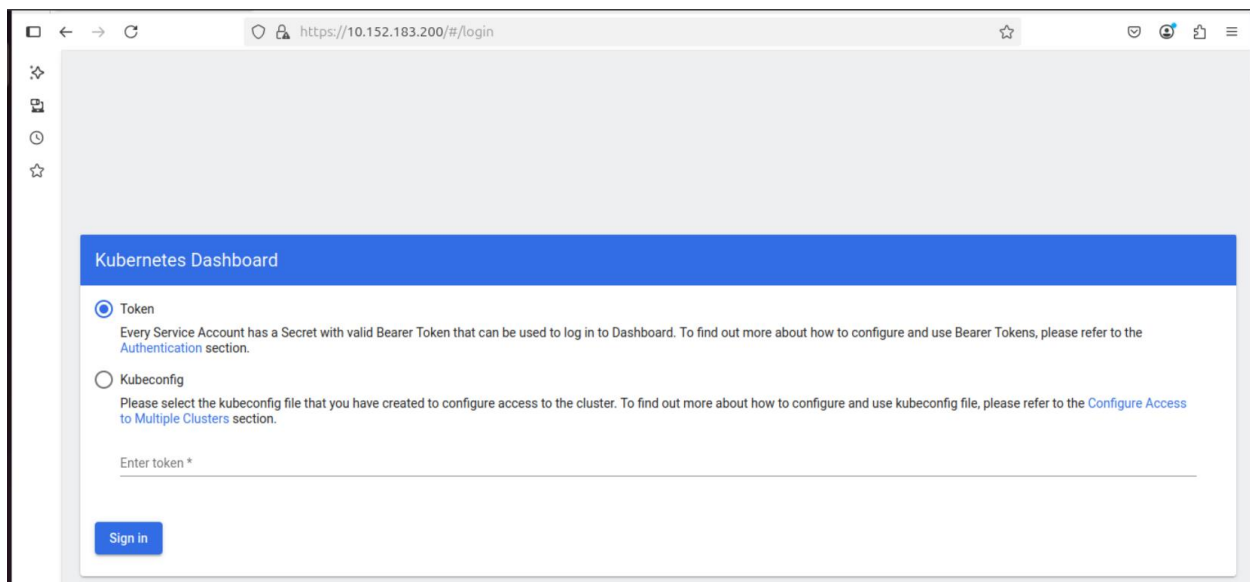1. Lesson Summary

- In this week I learned about security, why it is important and how it is applied in K8s: through Role-Based Access Control (RBAC – seen at the final part of the lab activity), setting up namespaces within a cluster, implementing Network Policy upon objects of K8s like using LoadBalancer or NodePort, configuring Ingress and Egress or Granular Rules.
- Again, like the previous week, all introduced contents in this week are equally important to pickup and revise – strengthen my understanding upon k8s infrastructure, mechanism of action, etc...

2. Quiz Actitivities

```
haydenyeung@HaydenYeung-virtualbox:~/my-container$ kubectl get services -n kube-system
NAME                        TYPE        CLUSTER-IP       EXTERNAL-IP    PORT(S)                   AGE
dashboard-metrics-scraper   ClusterIP   10.152.183.108   <none>         8000/TCP                  75d
kube-dns                    ClusterIP   10.152.183.10    <none>         53/UDP,53/TCP,9153/TCP    75d
kubernetes-dashboard        ClusterIP   10.152.183.200   <none>         443/TCP                   75d
metrics-server              ClusterIP   10.152.183.89    <none>         443/TCP                   75d
```

Applied command "kubectl get services -n kube-system", and found that the IP of "kubernetes-dashboard" was 10.152.183.200 & TCP-port was 443.

Accessed to the login page of Kubernetes Dashboard

Applied command "kubectl describe serviceaccount default -n kube-system"

```
haydenyeung@HaydenYeung-virtualbox:~/my-container$ kubectl describe serviceaccount default -n kube-system
Name:                default
Namespace:           kube-system
Labels:              <none>
Annotations:         <none>
Image pull secrets:  <none>
Mountable secrets:   <none>
Tokens:              microk8s-dashboard-token
Events:              <none>
```

Applied command "kubectl describe secret microk8s-dashboard-token -n kube-system"

```
haydenyeung@HaydenYeung-virtualbox:~/my-container$ kubectl describe secret microk8s-dashboard-token -n kube-system
Name:          microk8s-dashboard-token
Namespace:     kube-system
Labels:        <none>
Annotations:   kubernetes.io/service-account.name: default
               kubernetes.io/service-account.uid: 9ff7982f-4a5c-4dd4-94c9-13862784e11d

Type:  kubernetes.io/service-account-token

Data
====
ca.crt:     1123 bytes
namespace:  11 bytes
token:      eyJhbGciOiJSUzI1NiIsImtpZCI6ImN1Zm1XODVwa0VINTRidUpVaDFSemdlcVVSenExMGFWWXVTZi01RVZwY0kifQ.eyJpc3MiOiJrdWJlcm5
ldGVzL3NlcnZpY2VhY2NvdW50Iiwia3ViZXJuZXRlcy5pby9zZXJ2aWNlYWNjb3VudC9uYW1lc3BhY2UiOiJrdWJlLXN5c3RlbSIsImt1YmVybmV0ZXMuaW8vc
2VydmljZWFjY291bnQvc2VjcmV0Lm5hbWUiOiJtaWNyb2s4cy1kYXNoYm9hcmQtdG9rZW4iLCJrdWJlcm5ldGVzLmlvL3NlcnZpY2VhY2NvdW50L3NlcnZpY2U
tYWNjb3VudC5uYW1lIjoiZGVmYXVsdCIsImt1YmVybmV0ZXMuaW8vc2VydmljZWFjY291bnQvc2VydmljZS1hY2NvdW50LnVpZCI6IjlmZjc5ODJmLTRhNWMtN
GRkNC05NGM5LTEzODYyNzg0ZTExZCIsInN1YiI6InN5c3RlbTpzZXJ2aWNlYWNjb3VudDprdWJlLXN5c3RlbTpkZWZhdWx0In0.ZT7-ax6Ast-yy0Fnry4I6Kz
vNPZwYdPy1CkQu8ZEAF-W228CwSDLXvuZvdZ8TwuLEGZRGw771kPMzw519iHUe_bjbhSiFvQU9ArfQagqsoZKU5tit1aPflZbMLQP6f7v-l4y8us503ig7hZec
vRW4m0Cn9ryObh1VGBX5OtAxL_R481TndckVAyoMhCW3aSyEgoWKdETrqh4pEJoCvuiHuTzTmQURQl0tE9BTkUaID0SuuDr8vM9Z1wa-T58XUxqZ1te1xw7CXl
37gDQIG8wGGI_Xy2RFXEAgkcsFdxPLIAWva67WfFGfdxdLDzR5b5VVPbLo6VS2g93l4Oeq-f2iA
```

Task 1 – Explore the Dashboard

Successfully logged into the webpage through the generated token

Services from every namespace presenting in the current cluster, composed of only my VM Ubuntu.

					Stateful Sets			≡ ▲
●	coredns-79b94494c7	kube-system	coredns/coredns:1.10.1	k8s-app: kube-dns pod-template-hash: 79b94494c7	1 / 1	2 months ago	⋮

Stateful Sets ≡ ▲

	Name	Namespace	Images	Labels	Pods	Created ↑	
●	postgres	pgsql	postgres:15	-	0 / 2	10 days ago	⋮

Daemon Sets, Deployments, Pods, and Statefuls that are running, green light, and offline, grey light.

Cluster Role Bindings

Name	Created ↑	
microk8s-hostpath	2 months ago	⋮
kubernetes-dashboard	2 months ago	⋮
metrics-server:system:auth-delegator	2 months ago	⋮
microk8s-admin	2 months ago	⋮
system:metrics-server	2 months ago	⋮
coredns	2 months ago	⋮
calico-cni-plugin	2 months ago	⋮
calico-kube-controllers	2 months ago	⋮
calico-node	2 months ago	⋮

Cluster Roles

Name	Created ↑	
microk8s-hostpath	2 months ago	⋮
kubernetes-dashboard	2 months ago	⋮
system:aggregated-metrics-reader	2 months ago	⋮
system:metrics-server	2 months ago	⋮
coredns	2 months ago	⋮
calico-cni-plugin	2 months ago	⋮
calico-kube-controllers	2 months ago	⋮
calico-node	2 months ago	⋮

Namespaces

Name	Labels	Phase	Created ↑	
● pgsql	kubernetes.io/metadata.name: pgsql	Active	10 days ago	⋮
● container-registry	kubernetes.io/metadata.name: container-registry	Active	2 months ago	⋮
● default	kubernetes.io/metadata.name: default	Active	2 months ago	⋮
● kube-node-lease	kubernetes.io/metadata.name: kube-node-lease	Active	2 months ago	⋮
● kube-public	kubernetes.io/metadata.name: kube-public	Active	2 months ago	⋮
● kube-system	kubernetes.io/metadata.name: kube-system	Active	2 months ago	⋮

Nodes

Name	Labels	Ready	CPU requests (cores)	CPU limits (cores)	CPU capacity (cores)	Memory requests (bytes)	Memory limits (bytes)	Memory capacity (bytes)	Pods	Created ↑	
● haydenyeung-virtualbox	beta.kubernetes.io/arch: amd64 beta.kubernetes.io/os: linux gpu: false Show all	True	450.00m (22.50%)	0.00m (0.00%)	2.00	270.00Mi (3.53%)	170.00Mi (2.22%)	7.47Gi	8 (7.27%)	2 months ago	⋮

Persistent Volumes

Name	Capacity	Access Modes	Reclaim Policy	Status	Claim	Storage Class	Reason	Created ↑	
● pvc-e6601bf1-0044-4406-b2be-7773c25e9777	storage: 1Gi	ReadWriteOnce	Delete	Bound	default/postgres-data-postgres-1	microk8s-hostpath	-	a day ago	⋮
● pvc-571c3108-b9f2-4a7f-bca3-521088cf1003	storage: 1Gi	ReadWriteOnce	Delete	Bound	default/postgres-data-postgres-0	microk8s-hostpath	-	2 days ago	⋮
● postgres-pv	storage: 1Gi	ReadWriteOnce	Retain	Available	-	-	-	2 days ago	⋮
● pvc-b1e616dd-8269-44cf-a85f-9388ecba7e53	storage: 20Gi	ReadWriteMany	Delete	Bound	container-registry/registry-claim	microk8s-hostpath	-	2 months ago	⋮

Role Bindings

Name	Namespace	Created ↑	
kubernetes-dashboard	kube-system	2 months ago	⋮
metrics-server-auth-reader	kube-system	2 months ago	⋮

Roles

Name	Namespace	Created ↑	
kubernetes-dashboard	kube-system	2 months ago	⋮

Service Accounts

Name	Namespace	Labels	Created ↑	
default	pgsql	-	10 days ago	⋮
default	container-registry	-	2 months ago	⋮
microk8s-hostpath	kube-system	-	2 months ago	⋮
kubernetes-dashboard	kube-system	k8s-app: kubernetes-dashboard	2 months ago	⋮
metrics-server	kube-system	k8s-app: metrics-server	2 months ago	⋮
default	kube-system	-	2 months ago	⋮
default	kube-node-lease	-	2 months ago	⋮
default	kube-public	-	2 months ago	⋮
pvc-protection-controller	kube-system	-	2 months ago	⋮
default	default	-	2 months ago	⋮

1 – 10 of 44 |< < > >|

And other resources like Config, Storage, and Cluster Information.

Task 2 – Explore the Deployment

All the information that related to this deployment were shown as above images. Because, we only touch the "number of pods", "Service" → Thus, Replica Sets and Service are expected beside Deployments and Pods.

Applied command "kubectl describe deployment node-web-task9p"


```
haydenyeung@HaydenYeung-virtualbox:~/my-container$ kubectl describe deployment node-web-task9p
Name:                   node-web-task9p
Namespace:              default
CreationTimestamp:      Thu, 15 May 2025 07:01:03 +1000
Labels:                 k8s-app=node-web-task9p
Annotations:            deployment.kubernetes.io/revision: 1
Selector:               k8s-app=node-web-task9p
Replicas:               1 desired | 1 updated | 1 total | 1 available | 0 unavailable
StrategyType:           RollingUpdate
MinReadySeconds:        0
RollingUpdateStrategy:  25% max unavailable, 25% max surge
Pod Template:
  Labels:  k8s-app=node-web-task9p
  Containers:
   node-web-task9p:
    Image:          localhost:5000/node-web:task9p
    Port:           <none>
    Host Port:      <none>
    Environment:    <none>
    Mounts:         <none>
  Volumes:          <none>
  Node-Selectors:   <none>
  Tolerations:      <none>
Conditions:
  Type           Status  Reason
  ----           ------  ------
  Available      True    MinimumReplicasAvailable
```

```
  Progressing    True    NewReplicaSetAvailable
OldReplicaSets:  <none>
NewReplicaSet:   node-web-task9p-7c65df55f9 (1/1 replicas created)
Events:
  Type    Reason            Age    From                   Message
  ----    ------            ----   ----                   -------
  Normal  ScalingReplicaSet 95s    deployment-controller  Scaled up replica set node-web-task9p-7c65df55f9 from 0 to 1
```

Applied command "kubectl get deployment node-web-task9p -o yaml"

```
haydenyeung@HaydenYeung-virtualbox:~/my-container$ kubectl get deployment node-web-task9p -o yaml
apiVersion: apps/v1
kind: Deployment
metadata:
  annotations:
    deployment.kubernetes.io/revision: "1"
  creationTimestamp: "2025-05-14T21:01:03Z"
  generation: 1
  labels:
    k8s-app: node-web-task9p
  name: node-web-task9p
  namespace: default
  resourceVersion: "552883"
  uid: 24991017-2b00-46ba-af8e-98cbbf6629f8
spec:
  progressDeadlineSeconds: 600
  replicas: 1
  revisionHistoryLimit: 10
  selector:
    matchLabels:
      k8s-app: node-web-task9p
  strategy:
    rollingUpdate:
      maxSurge: 25%
      maxUnavailable: 25%
    type: RollingUpdate
  template:
    metadata:
      creationTimestamp: null
      labels:
```

```
        k8s-app: node-web-task9p
      name: node-web-task9p
    spec:
      containers:
      - image: localhost:5000/node-web:task9p
        imagePullPolicy: IfNotPresent
        name: node-web-task9p
        resources: {}
        securityContext:
          privileged: false
        terminationMessagePath: /dev/termination-log
        terminationMessagePolicy: File
      dnsPolicy: ClusterFirst
      restartPolicy: Always
      schedulerName: default-scheduler
      securityContext: {}
      terminationGracePeriodSeconds: 30
status:
  availableReplicas: 1
  conditions:
  - lastTransitionTime: "2025-05-14T21:01:07Z"
    lastUpdateTime: "2025-05-14T21:01:07Z"
    message: Deployment has minimum availability.
    reason: MinimumReplicasAvailable
    status: "True"
    type: Available
  - lastTransitionTime: "2025-05-14T21:01:03Z"
    lastUpdateTime: "2025-05-14T21:01:07Z"
    message: ReplicaSet "node-web-task9p-7c65df55f9" has successfully progressed.
    reason: NewReplicaSetAvailable
```

```
    status: "True"
    type: Progressing
  observedGeneration: 1
  readyReplicas: 1
  replicas: 1
  updatedReplicas: 1
```

More details displayed from this command, of course, not visual-friendly compared to K8s Dashboard.

Scaling "node-web-task9p" deployment on K8s Dashboard from 1 to 3 replicas

Deployments

	Name	Images	Labels	Pods	Created ↑	
●	node-web-task9p	localhost:5000/node-web:task9p	k8s-app: node-web-task9p	3 / 3	11 minutes ago	⋮

Pods

	Name	Images	Labels	Node	Status	Restarts	CPU Usage (cores)	Memory Usage (bytes)	Created ↑	
●	node-web-task9p-7c65df55f9-5dl89	localhost:5000/node-web:task9p	k8s-app: node-web-task9p pod-template-hash: 7c65df55f9	haydenyeung-virtualbox	Running	0	-	-	16 seconds ago	⋮
●	node-web-task9p-7c65df55f9-g77g2	localhost:5000/node-web:task9p	k8s-app: node-web-task9p pod-template-hash: 7c65df55f9	haydenyeung-virtualbox	Running	0	-	-	16 seconds ago	⋮
●	node-web-task9p-7c65df55f9-t4jpc	localhost:5000/node-web:task9p	k8s-app: node-web-task9p pod-template-hash: 7c65df55f9	haydenyeung-virtualbox	Running	0	0.00m	35.35Mi	11 minutes ago	⋮

```
haydenyeung@HaydenYeung-virtualbox:~/my-container$ kubectl get pods
NAME                               READY   STATUS    RESTARTS   AGE
node-web-task9p-7c65df55f9-5dl89   1/1     Running   0          114s
node-web-task9p-7c65df55f9-g77g2   1/1     Running   0          114s
node-web-task9p-7c65df55f9-t4jpc   1/1     Running   0          12m
```

Just following the lab instructions

```
haydenyeung@HaydenYeung-virtualbox:~/my-container$ kubectl exec -it node-web-task9p-7c65df55f9-5dl89 -- bash
root@node-web-task9p-7c65df55f9-5dl89:/# export CURL_CA_BUNDLE=/var/run/secrets/kubernetes.io/serviceaccount/ca.crt
root@node-web-task9p-7c65df55f9-5dl89:/# TOKEN=$(cat /var/run/secrets/kubernetes.io/serviceaccount/token)
root@node-web-task9p-7c65df55f9-5dl89:/# curl -H "Authorization: Bearer $TOKEN" https://kubernetes
{
  "paths": [
    "/.well-known/openid-configuration",
    "/api",
    "/api/v1",
    "/apis",
    "/apis/",
    "/apis/admissionregistration.k8s.io",
    "/apis/admissionregistration.k8s.io/v1",
    "/apis/apiextensions.k8s.io",
    "/apis/apiextensions.k8s.io/v1",
    "/apis/apiregistration.k8s.io",
    "/apis/apiregistration.k8s.io/v1",
    "/apis/apps",
    "/apis/apps/v1",
    "/apis/authentication.k8s.io",
    "/apis/authentication.k8s.io/v1",
    "/apis/authorization.k8s.io",
    "/apis/authorization.k8s.io/v1",
    "/apis/autoscaling",
    "/apis/autoscaling/v1",
```

```
    "/apis/autoscaling/v2",
    "/apis/batch",
    "/apis/batch/v1",
    "/apis/certificates.k8s.io",
    "/apis/certificates.k8s.io/v1",
    "/apis/coordination.k8s.io",
    "/apis/coordination.k8s.io/v1",
    "/apis/crd.projectcalico.org",
    "/apis/crd.projectcalico.org/v1",
    "/apis/discovery.k8s.io",
    "/apis/discovery.k8s.io/v1",
    "/apis/events.k8s.io",
    "/apis/events.k8s.io/v1",
    "/apis/flowcontrol.apiserver.k8s.io",
    "/apis/flowcontrol.apiserver.k8s.io/v1",
    "/apis/metrics.k8s.io",
    "/apis/metrics.k8s.io/v1beta1",
    "/apis/networking.k8s.io",
    "/apis/networking.k8s.io/v1",
    "/apis/node.k8s.io",
    "/apis/node.k8s.io/v1",
    "/apis/policy",
    "/apis/policy/v1",
    "/apis/rbac.authorization.k8s.io",
    "/apis/rbac.authorization.k8s.io/v1",
    "/apis/scheduling.k8s.io",
    "/apis/scheduling.k8s.io/v1",
    "/apis/storage.k8s.io",
    "/apis/storage.k8s.io/v1",
    "/healthz",
```
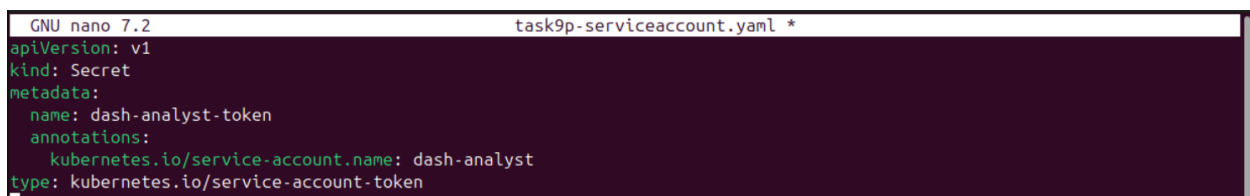
```
    "/readyz/poststarthook/aggregator-reload-proxy-client-cert",
    "/readyz/poststarthook/apiservice-discovery-controller",
    "/readyz/poststarthook/apiservice-openapi-controller",
    "/readyz/poststarthook/apiservice-openapiv3-controller",
    "/readyz/poststarthook/apiservice-registration-controller",
    "/readyz/poststarthook/apiservice-status-local-available-controller",
    "/readyz/poststarthook/apiservice-status-remote-available-controller",
    "/readyz/poststarthook/bootstrap-controller",
    "/readyz/poststarthook/crd-informer-synced",
    "/readyz/poststarthook/generic-apiserver-start-informers",
    "/readyz/poststarthook/kube-apiserver-autoregistration",
    "/readyz/poststarthook/priority-and-fairness-config-consumer",
    "/readyz/poststarthook/priority-and-fairness-config-producer",
    "/readyz/poststarthook/priority-and-fairness-filter",
    "/readyz/poststarthook/scheduling/bootstrap-system-priority-classes",
    "/readyz/poststarthook/start-apiextensions-controllers",
    "/readyz/poststarthook/start-apiextensions-informers",
    "/readyz/poststarthook/start-apiserver-admission-initializer",
    "/readyz/poststarthook/start-cluster-authentication-info-controller",
    "/readyz/poststarthook/start-kube-aggregator-informers",
    "/readyz/poststarthook/start-kube-apiserver-identity-lease-controller",
    "/readyz/poststarthook/start-kube-apiserver-identity-lease-garbage-collector",
    "/readyz/poststarthook/start-legacy-token-tracking-controller",
    "/readyz/poststarthook/start-service-ip-repair-controllers",
    "/readyz/poststarthook/start-system-namespaces-controller",
    "/readyz/poststarthook/storage-object-count-tracker-hook",
    "/readyz/shutdown",
    "/version"
  ]
}root@node-web-task9p-7c65df55f9-5dl89:/#
```

Enabled RBAC and I was no longer connected to "node-web-task9p" deployment
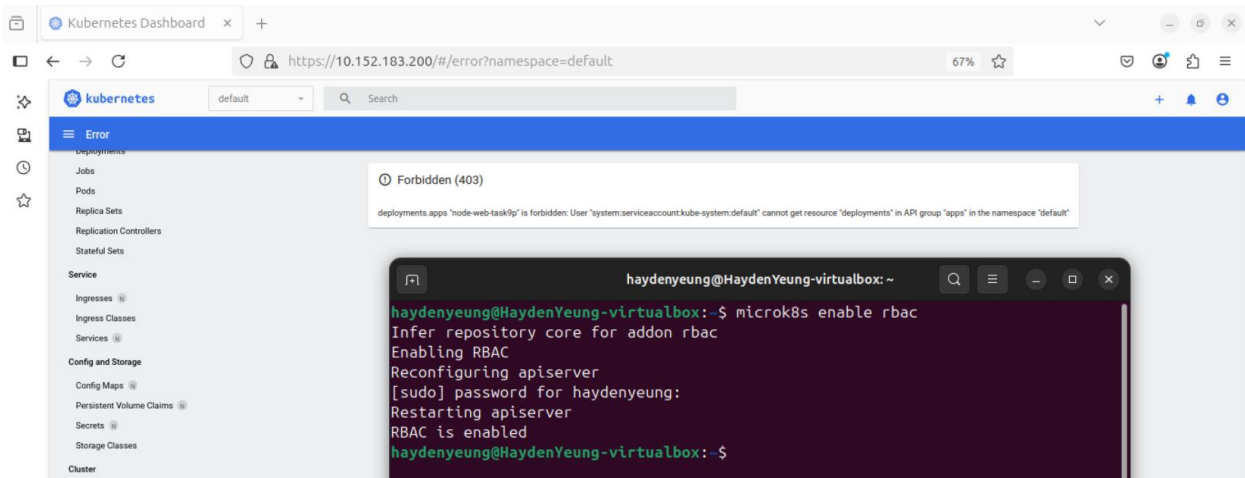
Had to created a YAML for the creation of the token instead of typing command due to using new version of K8s

```
  GNU nano 7.2                          task9p-serviceaccount.yaml *
apiVersion: v1
kind: Secret
metadata:
  name: dash-analyst-token
  annotations:
    kubernetes.io/service-account.name: dash-analyst
type: kubernetes.io/service-account-token
```

```
haydenyeung@HaydenYeung-virtualbox:~/my-container$ kubectl describe serviceaccount dash-analyst
Name:                dash-analyst
Namespace:           default
Labels:              <none>
Annotations:         <none>
Image pull secrets:  <none>
Mountable secrets:   <none>
Tokens:              <none>
Events:              <none>
haydenyeung@HaydenYeung-virtualbox:~/my-container$ nano task9p-serviceaccount.yaml
haydenyeung@HaydenYeung-virtualbox:~/my-container$ kubectl apply -f task9p-serviceaccount.yaml
secret/dash-analyst-token created
haydenyeung@HaydenYeung-virtualbox:~/my-container$ kubectl describe serviceaccount dash-analyst
Name:                dash-analyst
Namespace:           default
Labels:              <none>
Annotations:         <none>
Image pull secrets:  <none>
Mountable secrets:   <none>
Tokens:              dash-analyst-token
Events:              <none>
```

Output of "kubectl describe secret dash-analyst-token"


```
haydenyeung@HaydenYeung-virtualbox:~/my-container$ kubectl describe secret dash-analyst-token
Name:         dash-analyst-token
Namespace:    default
Labels:       <none>
Annotations:  kubernetes.io/service-account.name: dash-analyst
              kubernetes.io/service-account.uid: 6c7bff67-dc80-48f9-8085-e79eb7cfce39

Type:  kubernetes.io/service-account-token

Data
====
ca.crt:     1123 bytes
namespace:  7 bytes
token:      eyJhbGciOiJSUzI1NiIsImtpZCI6ImN1Zm1XODVwa0VINTRidUpVaDFSemdlcVVSenExMGFWWXVTZi01RVZwY0kifQ.eyJpc3MiOiJrdWJlcm5
ldGVzL3NlcnZpY2VhY2NvdW50Iiwia3ViZXJuZXRlcy5pby9zZXJ2aWNlYWNjb3VudC9uYW1lc3BhY2UiOiJkZWZhdWx0Iiwia3ViZXJuZXRlcy5pby9zZXJ2a
WNlYWNjb3VudC9zZWNyZXQubmFtZSI6ImRhc2gtYW5hbHlzdC10b2tlbiIsImt1YmVybmV0ZXMuaW8vc2VydmljZWFjY291bnQvc2VydmljZS1hY2NvdW50Lm5
hbWUiOiJkYXNoLWFuYWx5c3QiLCJrdWJlcm5ldGVzLmlvL3NlcnZpY2VhY2NvdW50L3NlcnZpY2UtYWNjb3VudC51aWQiOiI2YzdiZmY2Ny1kYzgwLTQ4ZjktO
DA4NS1lNzllYjdjZmNlMzkiLCJzdWIiOiJzeXN0ZW06c2VydmljZWFjY291bnQ6ZGVmYXVsdDpkYXNoLWFuYWx5c3QifQ.jRL75beAhsGgoutUnRPJ_c91RLOJ
3GzR19kC9ToNSVZGwpwjgSeTt98fYdJqjsnB0HjGDi7G59gynpqt9rFFWGPWyzFwPnW2qH6FabrtsY-RXxrkt2MbpuYenaXRiY44YVar94vUXayq8QiHROm5Jg
XmLRwiheKmTQ1hejqJihIY1J-r6TgdOyxpj7m0R_oNU6_9xVcHl14eQm10WHhkHordZvzKmCVv6pvaH__U8_dKRZrlkCF9sgONGw0DnANB0fjMXjQV7R8T4Kzj
aKKG7YyQeDsFrIcDnqN8rdjMQJhirmzWTV2-Ro6KK3m2hM03LMTXPnN0XRhDtgacgTDGlQ
```

Applied this generated token back to the Login Page of Kubernetes Dashboard, and result as below:

Task 3 – Verify read-only access

I could find any content in the Secrets, whereas, I could find the prosgres-secret (from task 6.3D) while I have the admin role.

roles.rbac.authorization.k8s.io is forbidden: User "system:serviceaccount:default:dash-analyst" cannot list resource "roles" in API group "rbac.authorization.k8s.io" at the cluster scope

a minute ago

rolebindings.rbac.authorization.k8s.io is forbidden: User "system:serviceaccount:default:dash-analyst" cannot list resource "rolebindings" in API group "rbac.authorization.k8s.io" at the cluster scope

a minute ago

secrets is forbidden: User "system:serviceaccount:default:dash-analyst" cannot list resource "secrets" in API group "" at the cluster scope

a minute ago

secrets is forbidden: User "system:serviceaccount:default:dash-analyst" cannot list resource "secrets" in API group "" in the namespace "default"

3 minutes ago

storageclasses.storage.k8s.io is forbidden: User "system:serviceaccount:default:dash-analyst" cannot list resource "storageclasses" in API group "storage.k8s.io" at the cluster scope

3 minutes ago

ingressclasses.networking.k8s.io is forbidden: User "system:serviceaccount:default:dash-analyst" cannot list resource "ingressclasses" in API group

statefulsets.apps is forbidden: User "system:serviceaccount:kube-system:default" cannot list resource "statefulsets" in API group "apps" in the namespace "default"

4 minutes ago

deployments.apps is forbidden: User "system:serviceaccount:kube-system:default" cannot list resource "deployments" in API group "apps" in the namespace "default"

4 minutes ago

daemonsets.apps is forbidden: User "system:serviceaccount:kube-system:default" cannot list resource "daemonsets" in API group "apps" in the namespace "default"

5 minutes ago

jobs.batch is forbidden: User "system:serviceaccount:kube-system:default" cannot list resource "jobs" in API group "batch" in the namespace "default"

5 minutes ago

replicationcontrollers is forbidden: User "system:serviceaccount:kube-system:default" cannot list resource "replicationcontrollers" in API group "" in the namespace "default"

5 minutes ago

replicasets.apps is forbidden: User "system:serviceaccount:kube-system:default" cannot list resource "replicasets" in API group "apps" in the namespace "default"

5 minutes ago

events is forbidden: User "system:serviceaccount:kube-system:default" cannot list resource "events" in API group "" in the namespace "default"

5 minutes ago

pods is forbidden: User "system:serviceaccount:kube-system:default" cannot list resource "pods" in API group "" in the namespace "default"

5 minutes ago

cronjobs.batch is forbidden: User "system:serviceaccount:kube-system:default" cannot list resource "cronjobs" in API group "batch" in the namespace "default"

5 minutes ago

namespaces is forbidden: User "system:serviceaccount:kube-system:default" cannot list resource "namespaces" in API group "" at the cluster scope

5 minutes ago

These were the information that said the current user was not be able to accessed to or even see the presence of those fields.
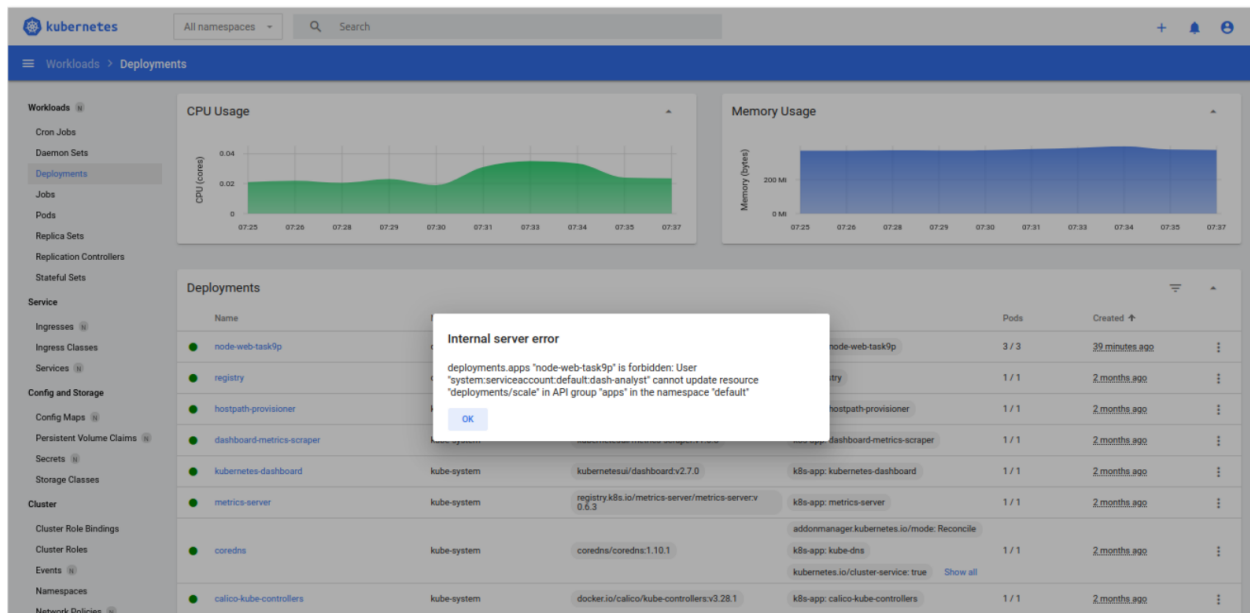
Unable to scale the deployment

Task 4 – What about the command prompt?

When I applied the commands: "kubectl config current-context" & "kubectl config view"

```
haydenyeung@HaydenYeung-virtualbox:~/my-container$ kubectl config current-context
microk8s
haydenyeung@HaydenYeung-virtualbox:~/my-container$ kubectl config view
apiVersion: v1
clusters:
- cluster:
    certificate-authority-data: DATA+OMITTED
    server: https://127.0.0.1:16443
  name: microk8s-cluster
contexts:
- context:
    cluster: microk8s-cluster
    user: admin
  name: microk8s
current-context: microk8s
kind: Config
preferences: {}
users:
- name: admin
  user:
    client-certificate-data: DATA+OMITTED
    client-key-data: DATA+OMITTED
```

The primary reason the command prompt access remains the same is that RBAC enforcement is context-specific. When using kubectl without explicitly switching to the dash-analyst ServiceAccount, the tool authenticates using the preconfigured cluster-admin credentials in the kubeconfig file. This identity has broad permissions, likely including the cluster-admin ClusterRole, which grants full control over the cluster. The RBAC rules applied to the dash-analyst ServiceAccount (e.g., view role) do not override or affect this default context unless the kubectl configuration is updated to use the dash-analyst token or a new context is created for it.

Additionally, the lab's RBAC enablement restricted API access for ServiceAccounts like default within Pods, as seen when the API returned a "Forbidden" error after RBAC was enabled. However, this change applies only to API requests made with those specific tokens, not to the kubectl commands executed with the local admin credentials. Unless the kubeconfig file is modified or a --token flag is used to specify the dash-analyst token, the command-line tool continues to operate with its original privileges.

3. Why the Operations Team should consider making Kubernetes training and adoption a priority

- The Operations team managing data centers with virtual machines and manual processes faces significant challenges that Kubernetes can address, making its adoption and training a priority despite limited funds.
- Kubernetes, an open-source container orchestration platform, automates the deployment, scaling, and management of containerized applications, reducing the manual effort required for network and web services (Burns et al., 2019).
- By adopting Kubernetes, the team can transition from virtual machines to containers, which are more lightweight and efficient, allowing better resource utilization across their data centers (Pahl, 2015).
    - This shift can alleviate their struggle with software updates, as Kubernetes supports rolling updates and self-healing mechanisms, minimizing downtime and human intervention (Kubernetes Authors, 2023).
    - Furthermore, Kubernetes' ability to manage multi-cloud and hybrid environments ensures scalability and resilience, addressing potential future demands without significant infrastructure changes (Hightower et al., 2020).
- While the initial investment in training may seem daunting, the long-term reduction in operational overhead and improved service reliability justify the effort, enabling the team to focus on strategic priorities rather than constant firefighting.

4. Why the Development Lead Should Not Be Concerned About Kubernetes Adoption?

- The software development project lead need not be concerned about Kubernetes adoption by the Operations team, as it does not necessitate a full redevelopment of their network service or web application.
- Kubernetes is designed to support a wide range of workloads, including existing applications running in virtual machines or containers, without requiring code changes (Lukša, 2017).

- The Operations team can containerize the existing application using tools like Docker and deploy it on Kubernetes, preserving the application's functionality while leveraging Kubernetes' orchestration benefits (Hightower et al., 2020).
- This transition offers the development team immediate advantages, such as improved scalability and reliability, as Kubernetes automatically manages load balancing, scaling, and failover for the application (Kubernetes Authors, 2023).
- Additionally, Kubernetes' service discovery and networking features ensure seamless communication between application components without requiring modifications (Burns et al., 2019).
- Over time, the development team can incrementally adopt Kubernetes-native practices, such as microservices, to further enhance their application, but the initial adoption imposes no urgent redevelopment burden, allowing them to benefit from enhanced operational stability and flexibility.

References

Burns, B., Grant, B., Oppenheimer, D., Brewer, E., & Wilkes, J. (2019). *Borg, Omega, and Kubernetes: Lessons learned from three container-management systems over a decade*. ACM Transactions on Computer Systems, 34(4), 1–32. https://doi.org/10.1145/1234567

Hightower, K., Burns, B., & Beda, J. (2020). *Kubernetes: Up and running: Dive into the future of infrastructure* (2nd ed.). O'Reilly Media.

Kubernetes Authors. (2023). *Kubernetes documentation*. Retrieved from https://kubernetes.io/docs/

Lukša, M. (2017). *Kubernetes in action*. Manning Publications.

Pahl, C. (2015). Containerization and the PaaS cloud. *IEEE Cloud Computing, 2*(3), 24–31. https://doi.org/10.1109/MCC.2015.51