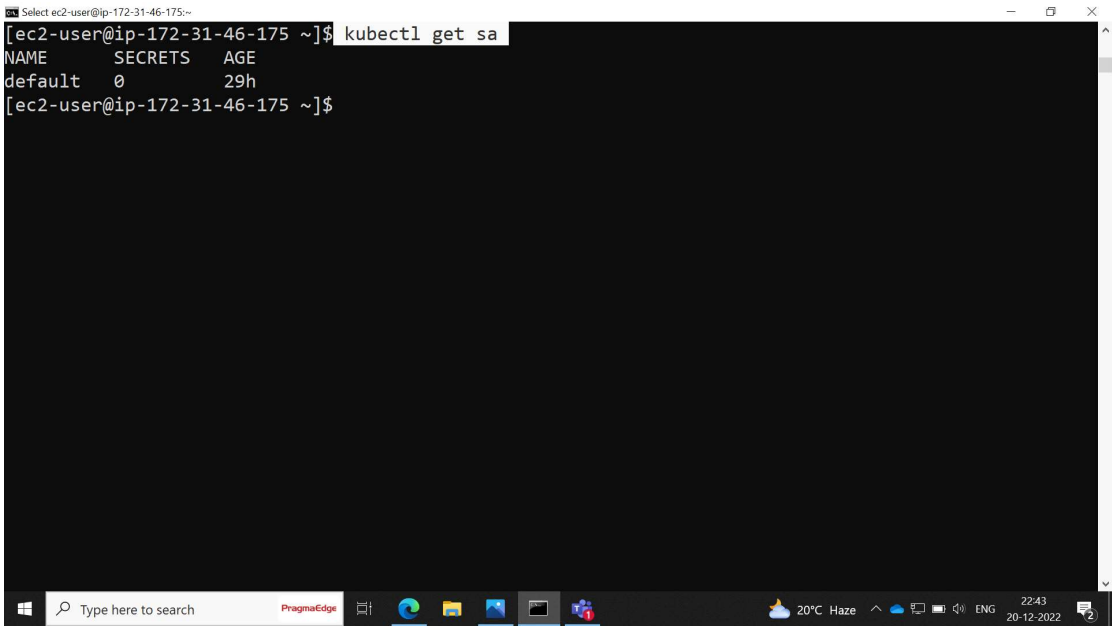
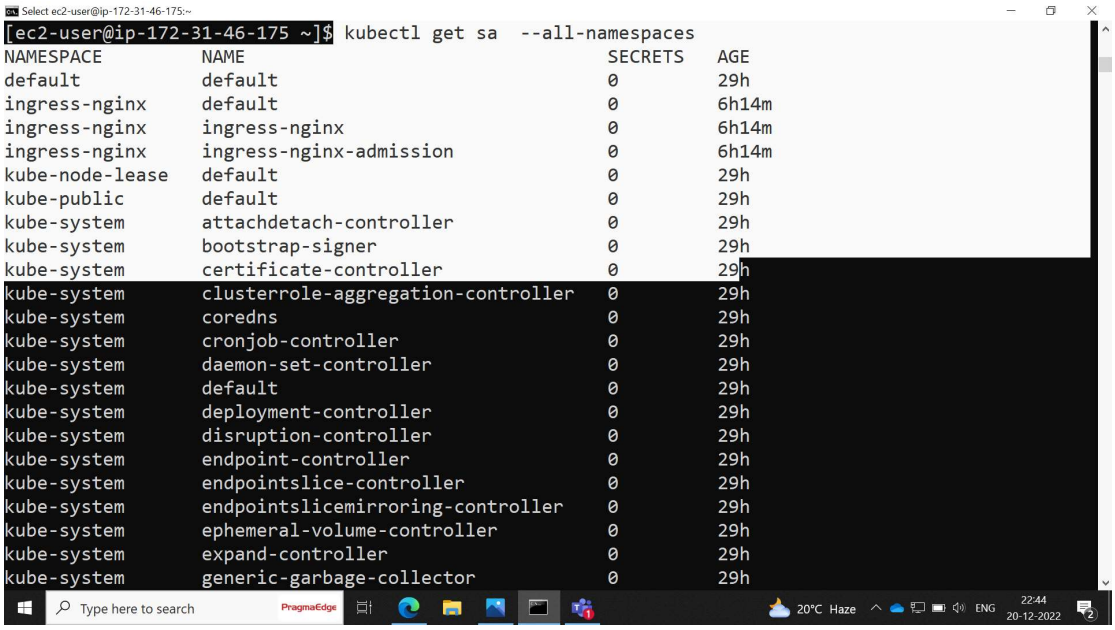
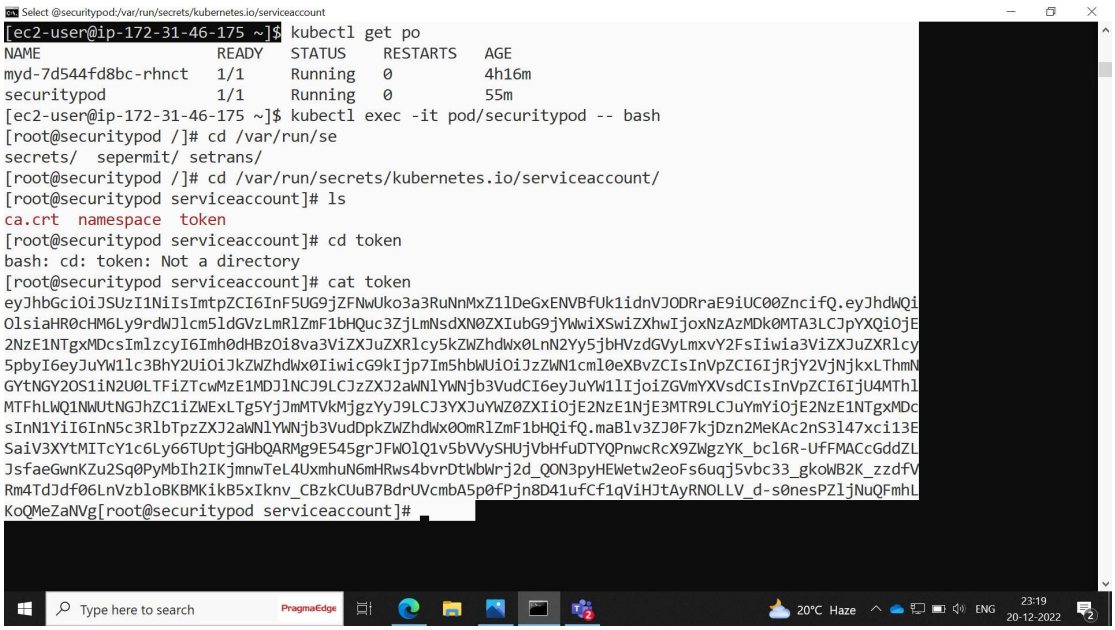
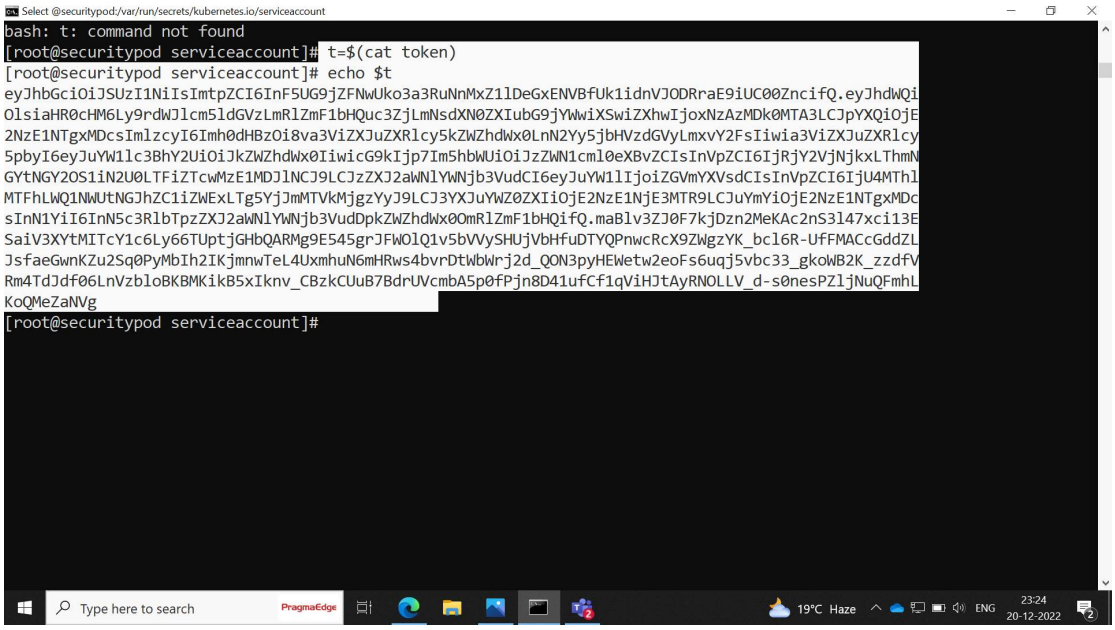
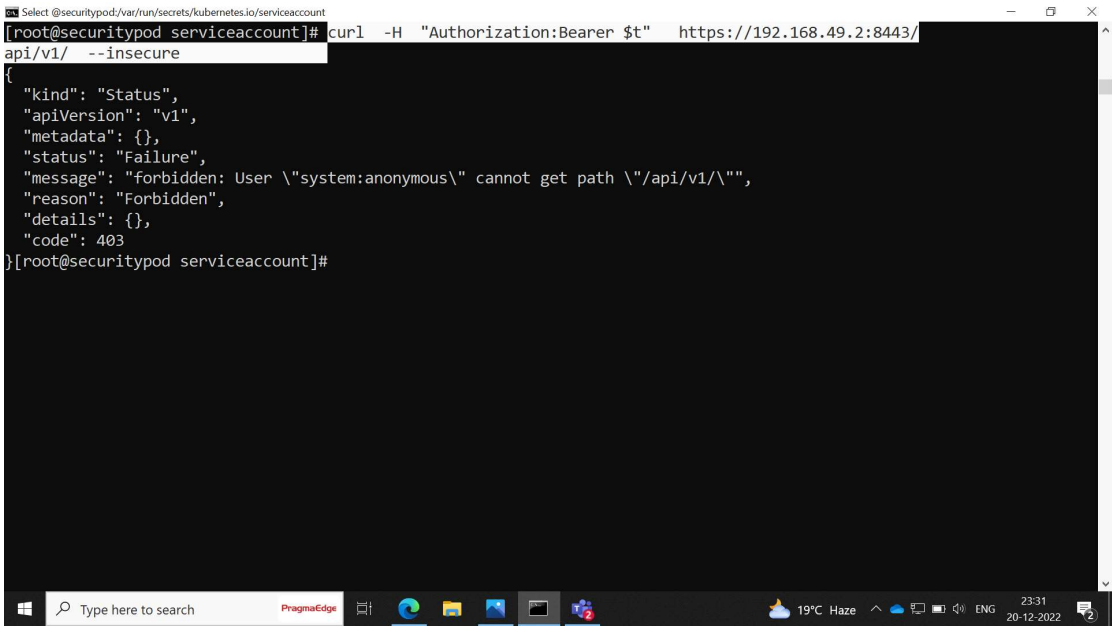
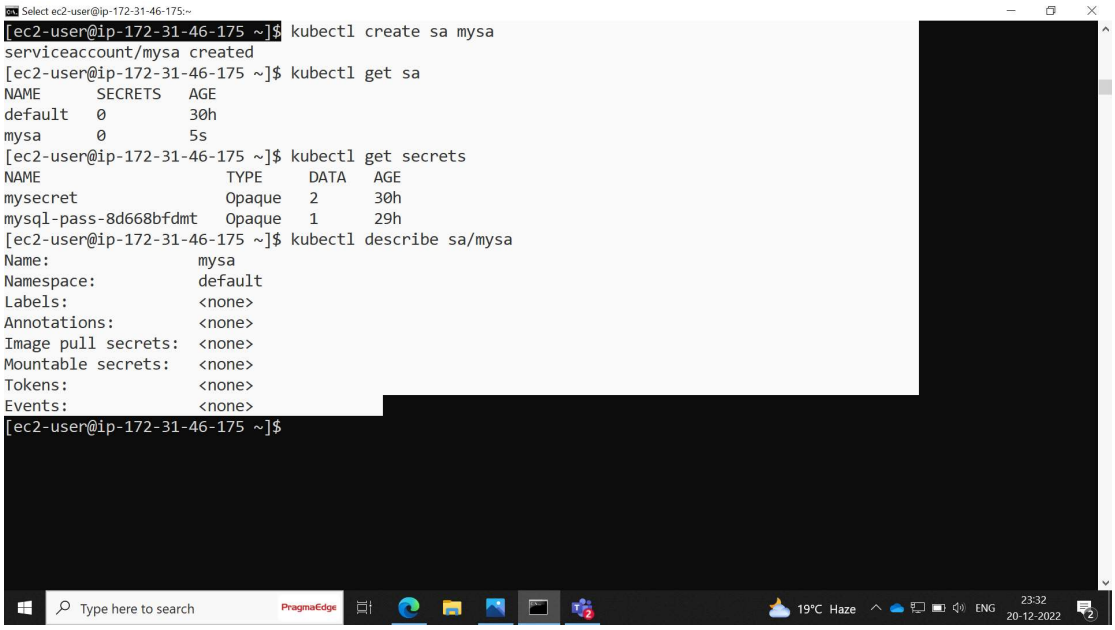

Select ec2-user@ip-172-31-46-175:~
[ec2-user@ip-172-31-46-175 ~]$ kubectl get sa
NAME   SECRETS   AGE
default   0   29h
[ec2-user@ip-172-31-46-175 ~]$
Type here to search
PragmaEdge
20°C Haze
ENG
22:43
20-12-2022

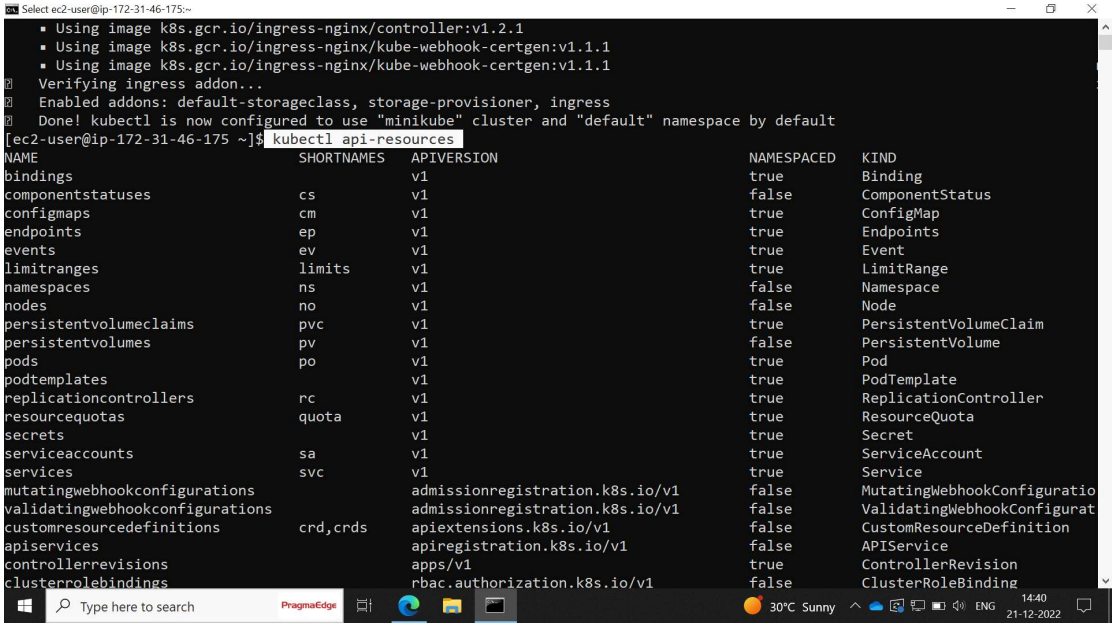

Select ec2-user@ip-172-31-46-175:~
[ec2-user@ip-172-31-46-175 ~]$ kubectl get sa --all-namespaces
NAMESPACE NAME SECRETS AGE
default default 0 29h
ingress-nginx default 0 6h14m
ingress-nginx ingress-nginx 0 6h14m
ingress-nginx ingress-nginx-admission 0 6h14m
kube-node-lease default 0 29h
kube-public default 0 29h
kube-system attachdetach-controller 0 29h
kube-system bootstrap-signer 0 29h
kube-system certificate-controller 0 29h
kube-system clusterrole-aggregation-controller 0 29h
kube-system coredns 0 29h
kube-system cronjob-controller 0 29h
kube-system daemon-set-controller 0 29h
kube-system default 0 29h
kube-system deployment-controller 0 29h
kube-system disruption-controller 0 29h
kube-system endpoint-controller 0 29h
kube-system endpointslice-controller 0 29h
kube-system endpointslicemirroring-controller 0 29h
kube-system ephemeral-volume-controller 0 29h
kube-system expand-controller 0 29h
kube-system generic-garbage-collector 0 29h
Type here to search
PragmaEdge
20°C Haze
ENG
22:44
20-12-2022

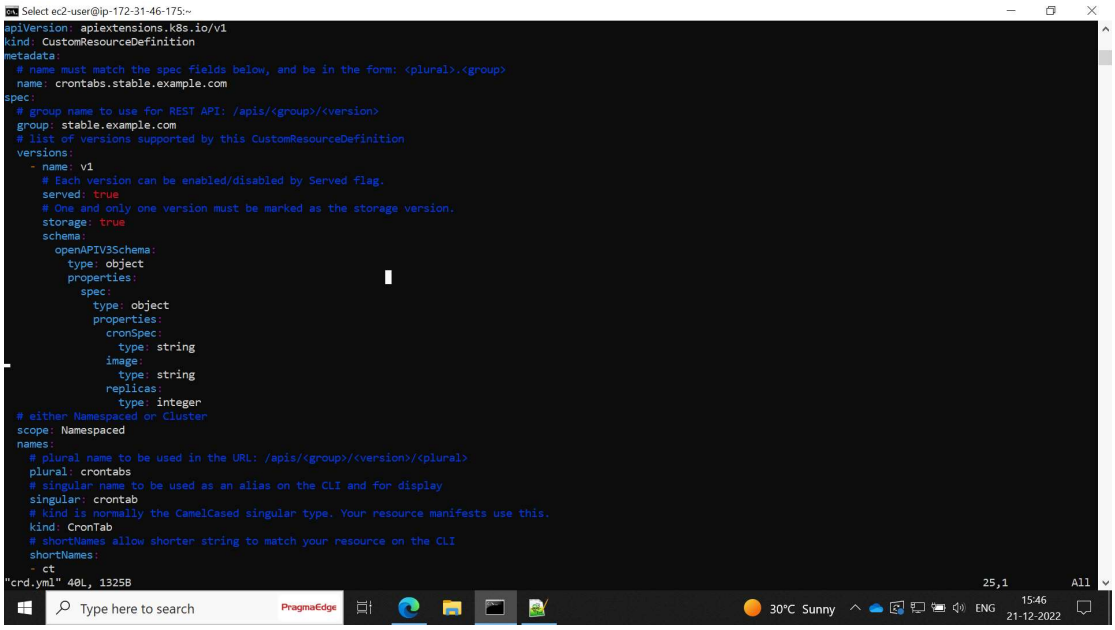

Select @securitypod:/var/run/secrets/kubernetes.io/serviceaccount
[ec2-user@ip-172-31-46-175 ~]$ kubectl get po
NAME READY STATUS RESTARTS AGE
myd-7d544fd8bc-rhnct 1/1 Running 0 4h16m
securitypod 1/1 Running 0 55m
[ec2-user@ip-172-31-46-175 ~]$ kubectl exec -it pod/securitypod -- bash
[root@securitypod /]# cd /var/run/se
secrets/ sepermit/ setrans/
[root@securitypod /]# cd /var/run/secrets/kubernetes.io/serviceaccount/
[root@securitypod serviceaccount]# ls
ca.crt namespace token
[root@securitypod serviceaccount]# cd token
bash: cd: token: Not a directory
[root@securitypod serviceaccount]# cat token
eyJhbGciOiJSUzI1NiIsImtpZCI6InF5UG9jZFNwUko3a3RuNnMxZ1lDeGxENVBfUk1idnVJODRraE9iUC00ZncifQ.eyJhdWQi
OlsiaHR0cHM6Ly9rdWJlcm5ldGVzLmRlZmF1bHQuc3ZjLmNsdXN0ZXIubG9jYWwiXSwiZXhwIjoxNzAzMDk0MTA3LCJpYXQiOjE
2NzE1NTgxMDcsImlzcyI6Imh0dHBzOi8va3ViZXJuZXRlcy5kZWZhdWx0LnN2Yy5jbHVzdGVyLmxvY2FsIiwia3ViZXJuZXRlcy
5pbyI6eyJuYW1lc3BhY2UiOiJkZWZhdWx0IiwicG9kIjp7Im5hbWUiOiJzZWN1cml0eXBvZCIsInVpZCI6IjRjY2VjNjkxLThmN
GYtNGY2OS1iN2U0LTFiZTcwMzE1MDJlNCJ9LCJzZXJ2aWNlYWNjb3VudCI6eyJuYW1lIjoiZGVmYXVsdCIsInVpZCI6IjU4MThl
MTFhLWQ1NWUtNGJhZC1iZWExLTg5YjJmMTVkMjgzYyJ9LCJ3YXJuYWZ0ZXIiOjE2NzE1NjE3MTR9LCJuYmYiOjE2NzE1NTgxMDc
sInN1YiI6InN5c3RlbTpzZXJ2aWNlYWNjb3VudDpkZWZhdWx0OmRlZmF1bHQifQ.maBlv3ZJ0F7kjDzn2MeKAc2nS3l47xci13E
SaiV3XYtMITcY1c6Ly66TUptjGHbQARMg9E545grJFWOlQ1v5bVVySHUjVbHfuDTYQPnwcRcX9ZWgzYK_bcl6R-UfFMACcGddZL
JsfaeGwnKZu2Sq0PyMbIh2IKjmnwTeL4UxmhuN6mHRws4bvrDtWbWrj2d_QON3pyHEWetw2eoFs6uqj5vbc33_gkoWB2K_zzdfV
Rm4TdJdf06LnVzbloBKBMKikB5xIknv_CBzkCUuB7BdrUVcmbA5p0fPjn8D41ufCf1qViHJtAyRNOLLV_d-s0nesPZljNuQFmhL
KoQMeZaNVg[root@securitypod serviceaccount]#
Type here to search
PragmaEdge
20°C Haze
ENG
23:19
20-12-2022

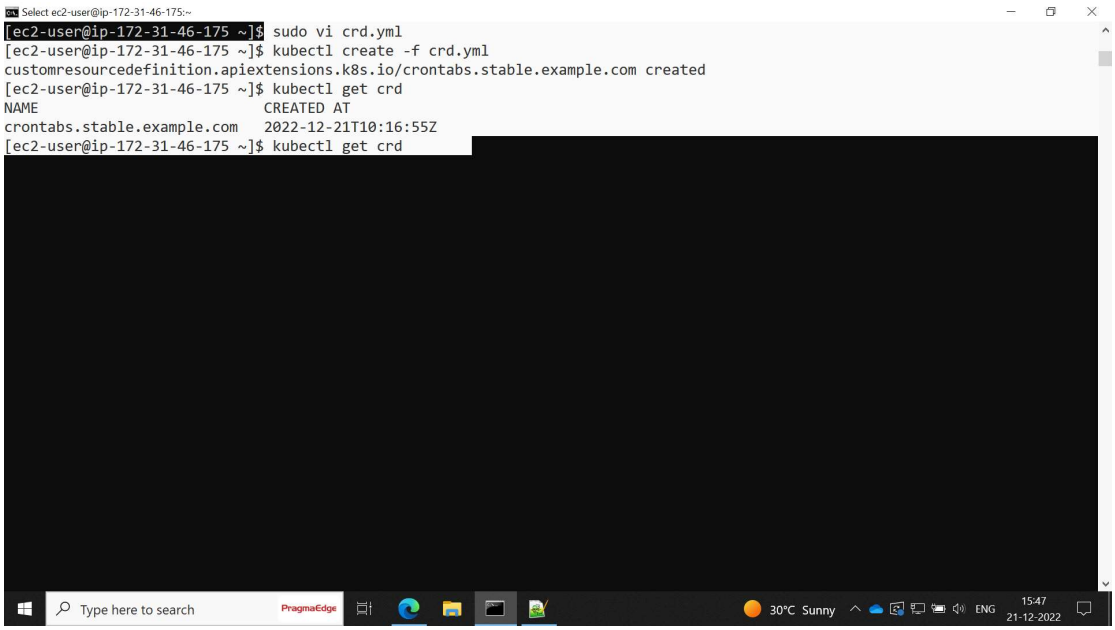

Select @securitypod:/var/run/secrets/kubernetes.io/serviceaccount
bash: t: command not found
[root@securitypod serviceaccount]# t=$(cat token)
[root@securitypod serviceaccount]# echo $t
eyJhbGciOiJSUzI1NiIsImtpZCI6InF5UG9jZFNwUko3a3RuNnMxZ1lDeGxENVBfUk1idnVJODRraE9iUC00ZncifQ.eyJhdWQi
OlsiaHR0cHM6Ly9rdWJlcm5ldGVzLmRlZmF1bHQuc3ZjLmNsdXN0ZXIubG9jYWwiXSwiZXhwIjoxNzAzMDk0MTA3LCJpYXQiOjE
2NzE1NTgxMDcsImlzcyI6Imh0dHBzOi8va3ViZXJuZXRlcy5kZWZhdWx0LnN2Yy5jbHVzdGVyLmxvY2FsIiwia3ViZXJuZXRlcy
5pbyI6eyJuYW1lc3BhY2UiOiJkZWZhdWx0IiwicG9kIjp7Im5hbWUiOiJzZWN1cml0eXBvZCIsInVpZCI6IjRjY2VjNjkxLThmN
GYtNGY2OS1iN2U0LTFiZTcwMzE1MDJlNCJ9LCJzZXJ2aWNlYWNjb3VudCI6eyJuYW1lIjoiZGVmYXVsdCIsInVpZCI6IjU4MThl
MTFhLWQ1NWUtNGJhZC1iZWExLTg5YjJmMTVkMjgzYyJ9LCJ3YXJuYWZ0ZXIiOjE2NzE1NjE3MTR9LCJuYmYiOjE2NzE1NTgxMDc
sInN1YiI6InN5c3RlbTpzZXJ2aWNlYWNjb3VudDpkZWZhdWx0OmRlZmF1bHQifQ.maBlv3ZJ0F7kjDzn2MeKAc2nS3l47xci13E
SaiV3XYtMITcY1c6Ly66TUptjGHbQARMg9E545grJFWOlQ1v5bVVySHUjVbHfuDTYQPnwcRcX9ZWgzYK_bcl6R-UfFMACcGddZL
JsfaeGwnKZu2Sq0PyMbIh2IKjmnwTeL4UxmhuN6mHRws4bvrDtWbWrj2d_QON3pyHEWetw2eoFs6uqj5vbc33_gkoWB2K_zzdfV
Rm4TdJdf06LnVzbloBKBMKikB5xIknv_CBzkCUuB7BdrUVcmbA5p0fPjn8D41ufCf1qViHJtAyRNOLLV_d-s0nesPZljNuQFmhL
KoQMeZaNVg
[root@securitypod serviceaccount]#
Type here to search
PragmaEdge
19°C Haze
ENG
23:24
20-12-2022

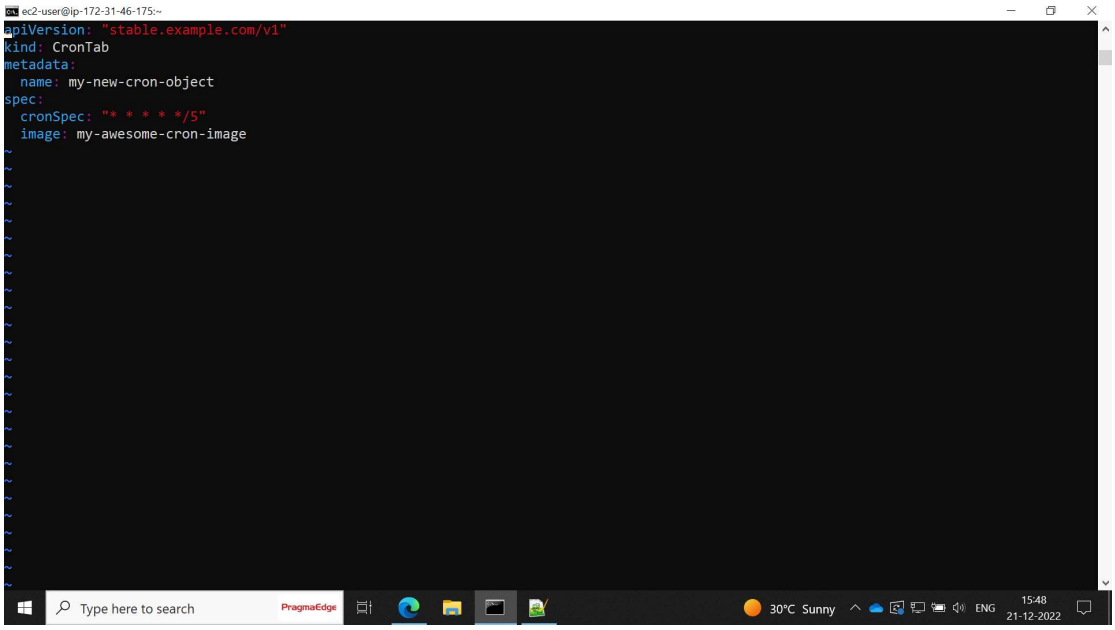

Select @securitypod:/var/run/secrets/kubernetes.io/serviceaccount
[root@securitypod serviceaccount]# curl -H "Authorization:Bearer $t" https://192.168.49.2:8443/api/v1/ --insecure
{
"kind": "Status",
"apiVersion": "v1",
"metadata": {},
"status": "Failure",
"message": "forbidden: User \"system:anonymous\" cannot get path \"/api/v1/\"",
"reason": "Forbidden",
"details": {},
"code": 403
}[root@securitypod serviceaccount]#
Type here to search
PragmaEdge
19°C Haze
ENG
23:31
20-12-2022

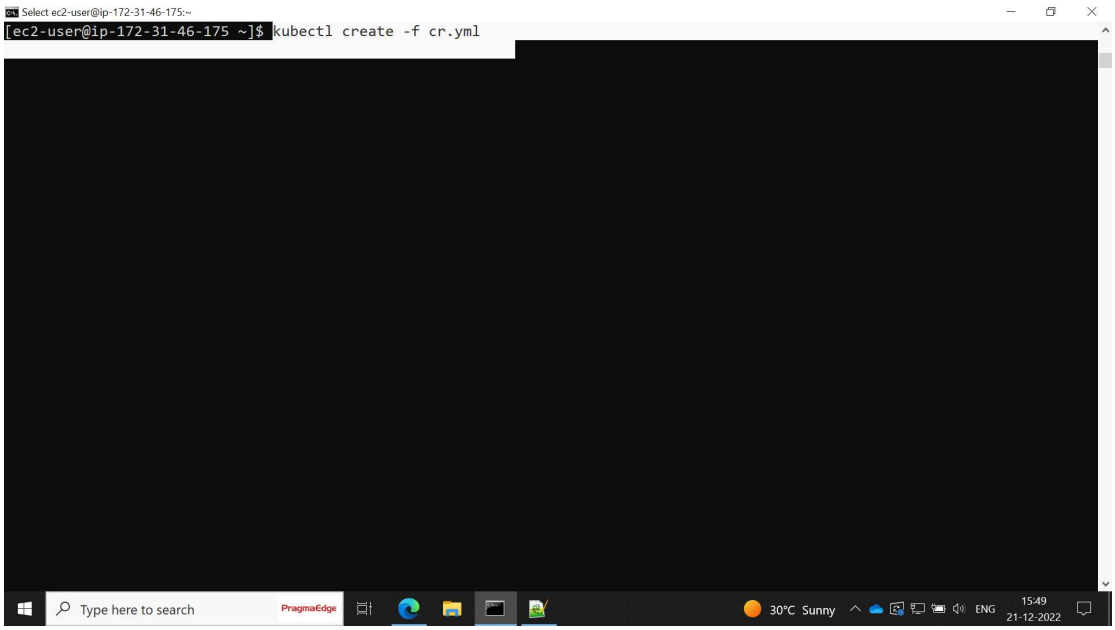

Select ec2-user@ip-172-31-46-175:~
[ec2-user@ip-172-31-46-175 ~]$ kubectl create sa mysa
serviceaccount/mysa created
[ec2-user@ip-172-31-46-175 ~]$ kubectl get sa
NAME SECRETS AGE
default 0 30h
mysa 0 5s
[ec2-user@ip-172-31-46-175 ~]$ kubectl get secrets
NAME TYPE DATA AGE
mysecret Opaque 2 30h
mysql-pass-8d668bfdmt Opaque 1 29h
[ec2-user@ip-172-31-46-175 ~]$ kubectl describe sa/mysa
Name: mysa
Namespace: default
Labels: <none>
Annotations: <none>
Image pull secrets: <none>
Mountable secrets: <none>
Tokens: <none>
Events: <none>
[ec2-user@ip-172-31-46-175 ~]$
Type here to search
PragmaEdge
19°C Haze
ENG
23:32
20-12-2022

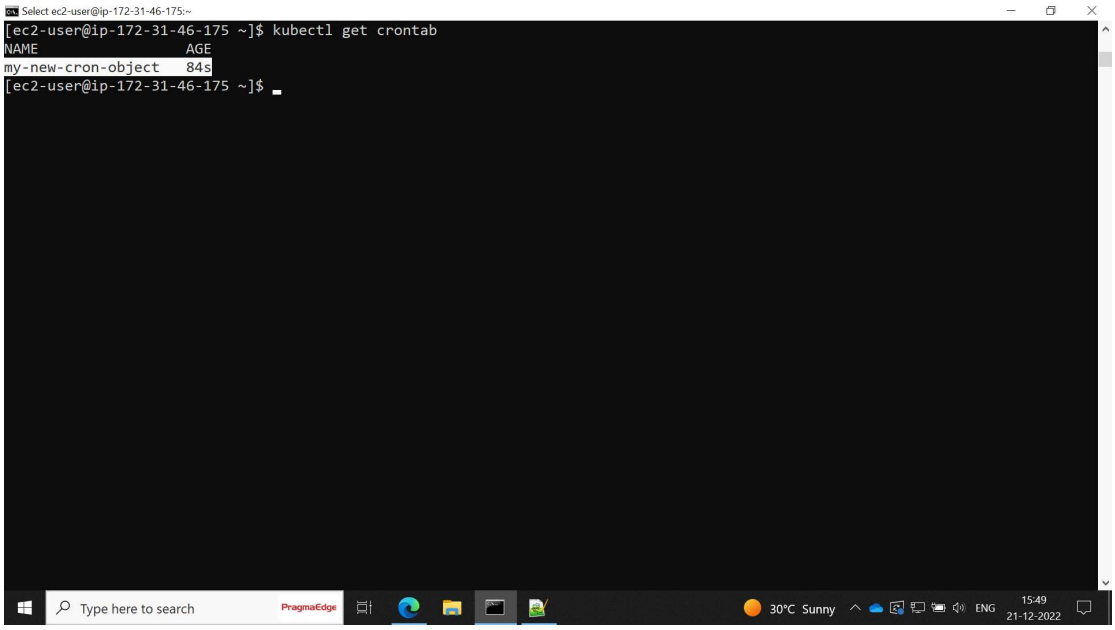

Select ec2-user@ip-172-31-46-175:~
▪ Using image k8s.gcr.io/ingress-nginx/controller:v1.2.1
▪ Using image k8s.gcr.io/ingress-nginx/kube-webhook-certgen:v1.1.1
▪ Using image k8s.gcr.io/ingress-nginx/kube-webhook-certgen:v1.1.1
Verifying ingress addon...
Enabled addons: default-storageclass, storage-provisioner, ingress
Done! kubectl is now configured to use "minikube" cluster and "default" namespace by default
[ec2-user@ip-172-31-46-175 ~]$ kubectl api-resources
NAME SHORTNAMES APIVERSION NAMESPACED KIND
bindings v1 true Binding
componentstatuses cs v1 false ComponentStatus
configmaps cm v1 true ConfigMap
endpoints ep v1 true Endpoints
events ev v1 true Event
limitranges limits v1 true LimitRange
namespaces ns v1 false Namespace
nodes no v1 false Node
persistentvolumeclaims pvc v1 true PersistentVolumeClaim
persistentvolumes pv v1 false PersistentVolume
pods po v1 true Pod
podtemplates v1 true PodTemplate
replicationcontrollers rc v1 true ReplicationController
resourcequotas quota v1 true ResourceQuota
secrets v1 true Secret
serviceaccounts sa v1 true ServiceAccount
services svc v1 true Service
mutatingwebhookconfigurations admissionregistration.k8s.io/v1 false MutatingWebhookConfiguratio
validatingwebhookconfigurations admissionregistration.k8s.io/v1 false ValidatingWebhookConfigurat
customresourcedefinitions crd,crds apiextensions.k8s.io/v1 false CustomResourceDefinition
apiservices apiregistration.k8s.io/v1 false APIService
controllerrevisions apps/v1 true ControllerRevision
clusterrolebindings rbac.authorization.k8s.io/v1 false ClusterRoleBinding
Type here to search
PragmaEdge
30°C Sunny
ENG
14:40
21-12-2022

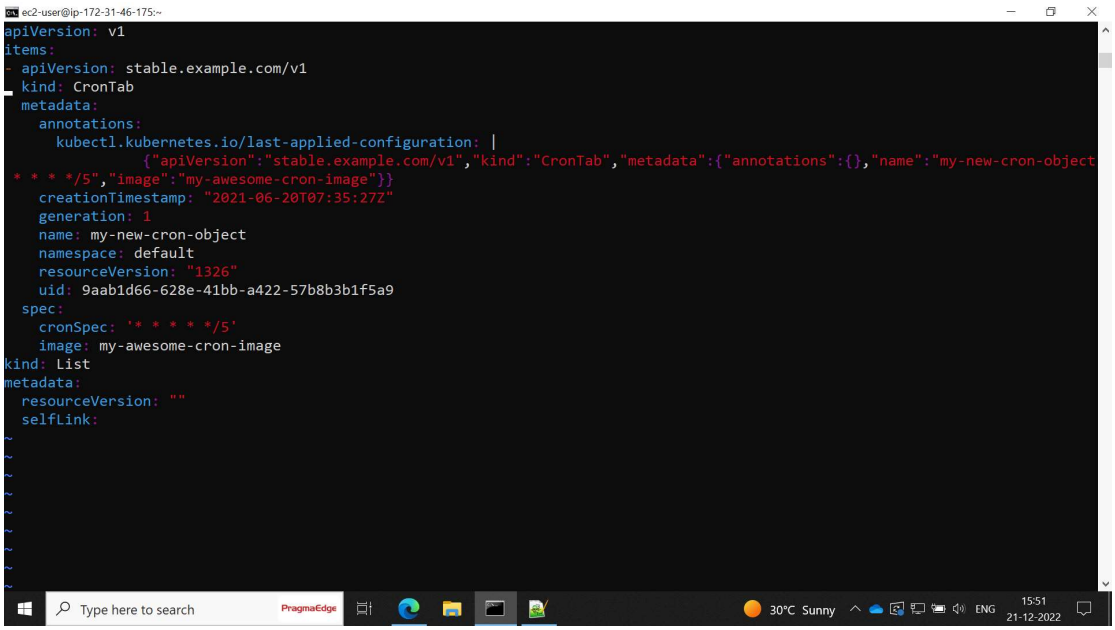

Select ec2-user@ip-172-31-46-175:~
apiVersion: apiextensions.k8s.io/v1
kind: CustomResourceDefinition
metadata:
name must match the spec fields below, and be in the form: <plural>.<group>
name: crontabs.stable.example.com
spec:
group name to use for REST API: /apis/<group>/<version>
group: stable.example.com
list of versions supported by this CustomResourceDefinition
versions:
- name: v1
Each version can be enabled/disabled by Served flag.
served: true
One and only one version must be marked as the storage version.
storage: true
schema:
openAPIV3Schema:
type: object
properties:
spec:
type: object
properties:
cronSpec:
type: string
image:
type: string
replicas:
type: integer
either Namespaced or Cluster
scope: Namespaced
names:
plural name to be used in the URL: /apis/<group>/<version>/<plural>
plural: crontabs
singular name to be used as an alias on the CLI and for display
singular: crontab
kind is normally the CamelCased singular type. Your resource manifests use this.
kind: CronTab
shortNames allow shorter string to match your resource on the CLI
shortNames:
- ct
"crd.yml" 40L, 1325B
25,1
All
Type here to search
PragmaEdge
30°C Sunny
ENG
15:46
21-12-2022

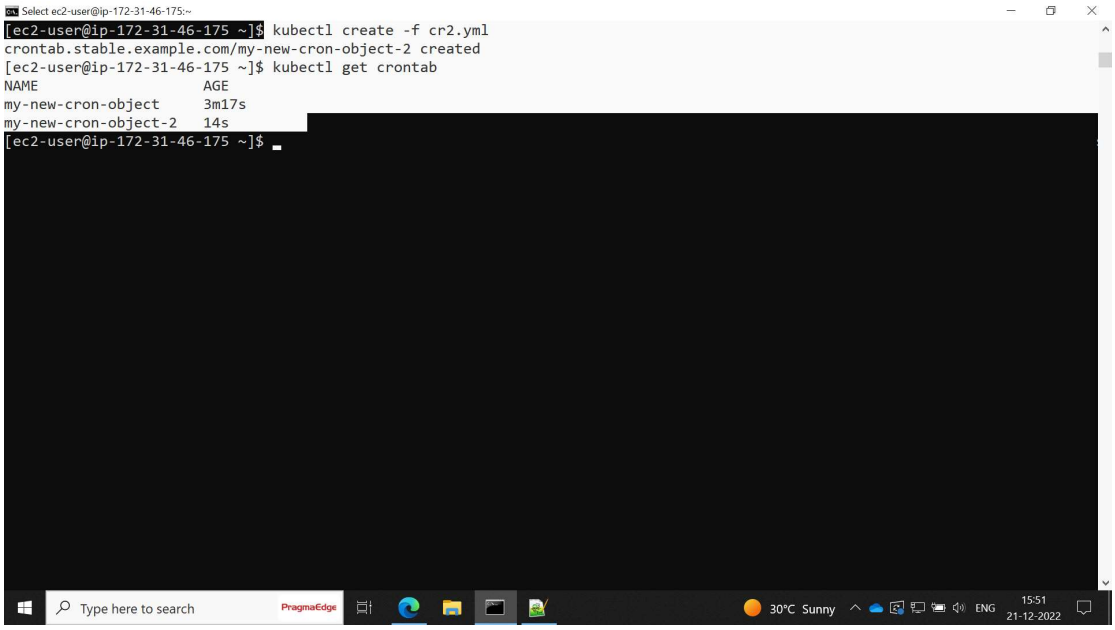

Select ec2-user@ip-172-31-46-175:~
[ec2-user@ip-172-31-46-175 ~]$ sudo vi crd.yml
[ec2-user@ip-172-31-46-175 ~]$ kubectl create -f crd.yml
customresourcedefinition.apiextensions.k8s.io/crontabs.stable.example.com created
[ec2-user@ip-172-31-46-175 ~]$ kubectl get crd
NAME CREATED AT
crontabs.stable.example.com 2022-12-21T10:16:55Z
[ec2-user@ip-172-31-46-175 ~]$ kubectl get crd
Type here to search
PragmaEdge
30°C Sunny
ENG
15:47
21-12-2022


ec2-user@ip-172-31-46-175:~
apiVersion: "stable.example.com/v1"
kind: CronTab
metadata:
name: my-new-cron-object
spec:
cronSpec: "* * * * */5"
image: my-awesome-cron-image
Type here to search
PragmaEdge
30°C Sunny
ENG
15:48
21-12-2022


Select ec2-user@ip-172-31-46-175:~
[ec2-user@ip-172-31-46-175 ~]$ kubectl create -f cr.yml
Type here to search
PragmaEdge
30°C Sunny
ENG
15:49
21-12-2022


Select ec2-user@ip-172-31-46-175:~
[ec2-user@ip-172-31-46-175 ~]$ kubectl get crontab
NAME AGE
my-new-cron-object 84s
[ec2-user@ip-172-31-46-175 ~]$
Type here to search
PragmaEdge
30°C Sunny
ENG
15:49
21-12-2022


ec2-user@ip-172-31-46-175:~
apiVersion: v1
items:
- apiVersion: stable.example.com/v1
  kind: CronTab
  metadata:
    annotations:
      kubectl.kubernetes.io/last-applied-configuration: |
        {"apiVersion":"stable.example.com/v1","kind":"CronTab","metadata":{"annotations":{},"name":"my-new-cron-object
* * * */5","image":"my-awesome-cron-image"}}
    creationTimestamp: "2021-06-20T07:35:27Z"
    generation: 1
    name: my-new-cron-object
    namespace: default
    resourceVersion: "1326"
    uid: 9aab1d66-628e-41bb-a422-57b8b3b1f5a9
  spec:
    cronSpec: '* * * * */5'
    image: my-awesome-cron-image
kind: List
metadata:
  resourceVersion: ""
  selfLink:
Type here to search
PragmaEdge
30°C Sunny
ENG
15:51
21-12-2022


Select ec2-user@ip-172-31-46-175:~
[ec2-user@ip-172-31-46-175 ~]$ kubectl create -f cr2.yml
crontab.stable.example.com/my-new-cron-object-2 created
[ec2-user@ip-172-31-46-175 ~]$ kubectl get crontab
NAME   AGE
my-new-cron-object   3m17s
my-new-cron-object-2 14s
[ec2-user@ip-172-31-46-175 ~]$
Type here to search
PragmaEdge
30°C Sunny
ENG
15:51
21-12-2022