集群权限注意事项

在尝试权限提升时，想要找到集群中拥有高危权限的SA token 或 角色，尝试利用。那么哪些属于集群中的高危可以利用的权限呢？

[list secrets] 、 [get secrets] 或 [watch secrets]

[list secrets]权限

创建绑定list secrets权限的sa账号 secret-t1

kubectl create serviceaccount secret-t1

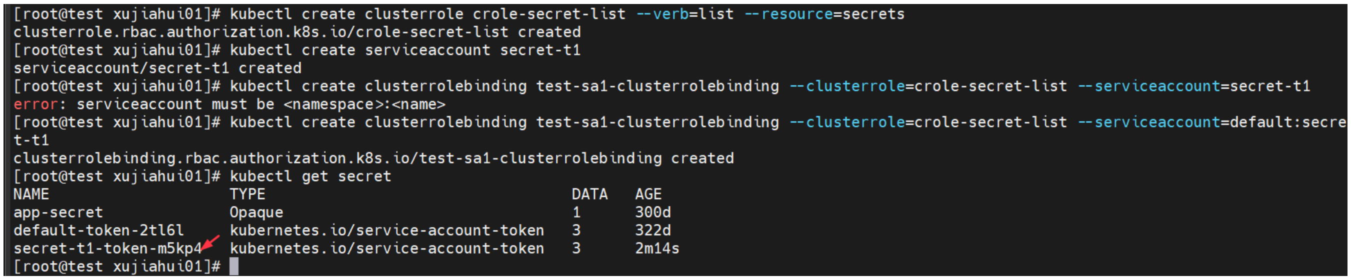
创建集群级别角色clusterrole

kubectl create clusterrole crole-secret-list --verb=list --resource=secrets

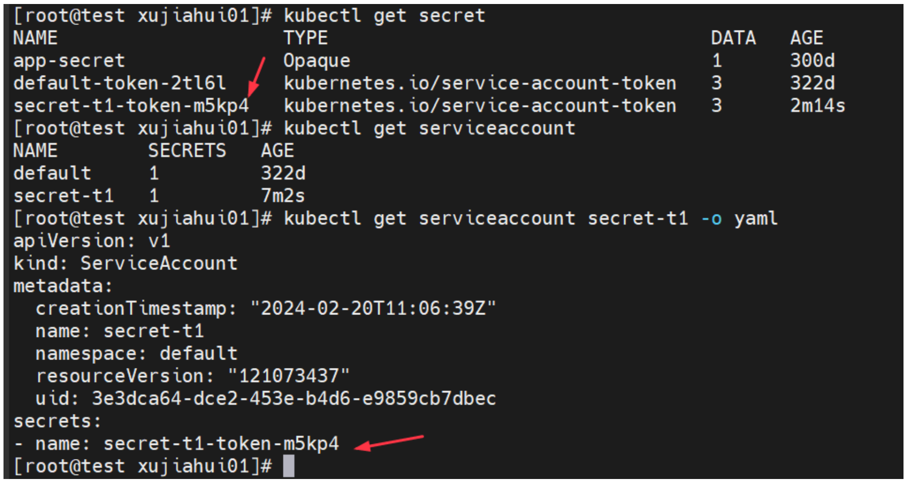
角色绑定

kubectl create clusterrolebinding test-sa1-clusterrolebinding --clusterrole=crole-secret-list --serviceaccount=default:secret-t1

在k8s v1.24 之后，创建serviceaccount不会自动生成 secret，需要手动创建，参考 https://www.soulchild.cn/post/2945/



```
[root@test xujiahui01]# kubectl create clusterrole crole-secret-list --verb=list --resource=secrets
clusterrole.rbac.authorization.k8s.io/crole-secret-list created
[root@test xujiahui01]# kubectl create serviceaccount secret-t1
serviceaccount/secret-t1 created
[root@test xujiahui01]# kubectl create clusterrolebinding test-sa1-clusterrolebinding --clusterrole=crole-secret-list --serviceaccount=secret-t1
error: serviceaccount must be <namespace>:<name>
[root@test xujiahui01]# kubectl create clusterrolebinding test-sa1-clusterrolebinding --clusterrole=crole-secret-list --serviceaccount=default:secre
t-t1
clusterrolebinding.rbac.authorization.k8s.io/test-sa1-clusterrolebinding created
[root@test xujiahui01]# kubectl get secret
NAME                     TYPE                                  DATA   AGE
app-secret               Opaque                                1      300d
default-token-2tl6l      kubernetes.io/service-account-token   3      322d
secret-t1-token-m5kp4    kubernetes.io/service-account-token   3      2m14s
[root@test xujiahui01]#
```



```
[root@test xujiahui01]# kubectl get secret
NAME                     TYPE                                  DATA   AGE
app-secret               Opaque                                1      300d
default-token-2tl6l      kubernetes.io/service-account-token   3      322d
secret-t1-token-m5kp4    kubernetes.io/service-account-token   3      2m14s
[root@test xujiahui01]# kubectl get serviceaccount
NAME        SECRETS   AGE
default     1         322d
secret-t1   1         7m2s
[root@test xujiahui01]# kubectl get serviceaccount secret-t1 -o yaml
apiVersion: v1
kind: ServiceAccount
metadata:
  creationTimestamp: "2024-02-20T11:06:39Z"
  name: secret-t1
  namespace: default
  resourceVersion: "121073437"
  uid: 3e3dca64-dce2-453e-b4d6-e9859cb7dbec
secrets:
- name: secret-t1-token-m5kp4
[root@test xujiahui01]#
```

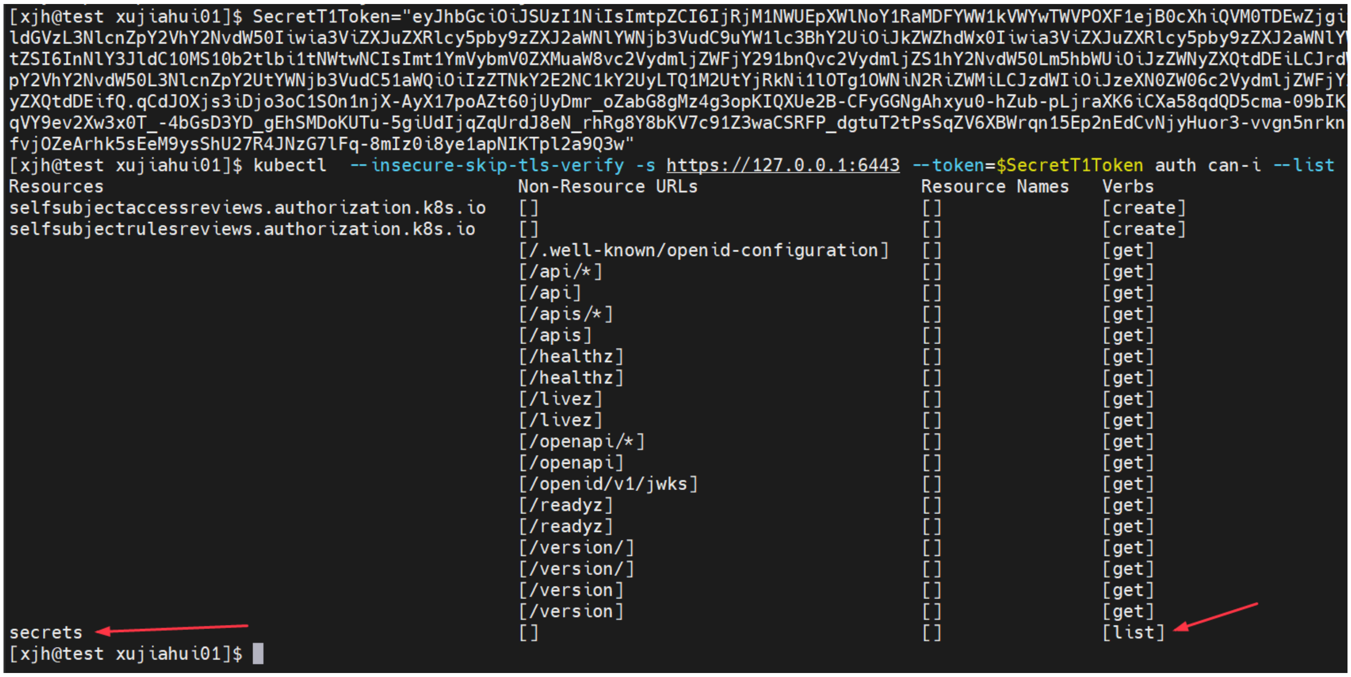
获取复制使用secret-token

kubectl get secret secret-t1-token-m5kp4 -o yaml |grep token: |awk '{print $2}'|base64 -d

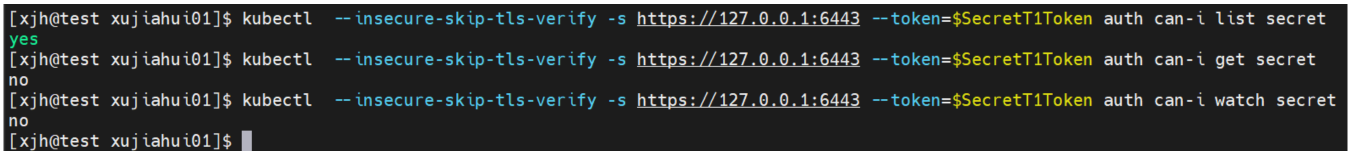
查看是否拥有某个权限

kubectl --insecure-skip-tls-verify -s https://127.0.0.1:6443 --token=$SecretT1Token auth can-i --list

kubectl --insecure-skip-tls-verify -s https://127.0.0.1:6443 --token=$SecretT1Token auth can-i list secret

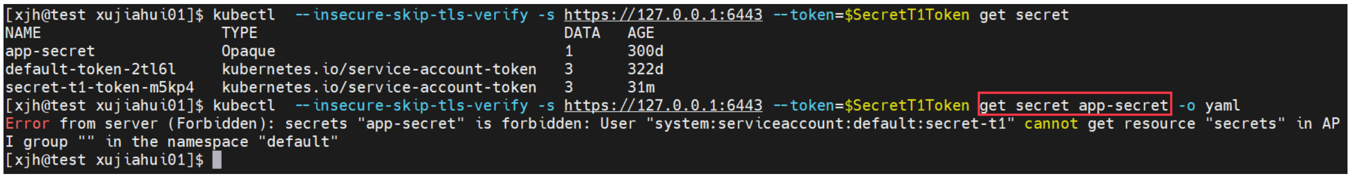


```
[xjh@test xujiahui01]$ SecretT1Token="eyJhbGciOiJSUzI1NiIsImtpZCI6IjRjM1NWUEpXWlNoY1RaMDFYWW1kVWYwTWVPOXF1ejB0cXhiQVM0TDEwZjgi
ldGVzL3NlcnZpY2VhY2NvdW50Iiwia3ViZXJuZXRlcy5pby9zZXJ2aWNlYWNjb3VudC9uYW1lc3BhY2UiOiJkZWZhdWx0Iiwia3ViZXJuZXRlcy5pby9zZXJ2aWNlY
tZSI6InNlY3JldC10MS10b2tlbi1tNWtwNCIsImt1YmVybmV0ZXMuaW8vc2VydmljZWFjY291bnQvc2VydmljZS1hY2NvdW50Lm5hbWUiOiJzZWNyZXQtdDEiLCJrd
pY2VhY2NvdW50L3NlcnZpY2UtYWNjb3VudC51aWQiOiIzZTNkY2E2NC1kY2UyLTQ1M2UtYjRkNi1lOTg1OWNiN2RiZWMiLCJzdWIiOiJzeXN0ZW06c2VydmljZWFjY
yZXQtdDEifQ.qCdJOXjs3iDjo3oC1SOn1njX-AyX17poAZt60jUyDmr_oZabG8gMz4g3opKIQXUe2B-CFyGGNgAhxyu0-hZub-pLjraXK6iCXa58qdQD5cma-09bIK
qVY9ev2Xw3x0T_-4bGsD3YD_gEhSMDoKUTu-5giUdIjqZqUrdJ8eN_rhRg8Y8bKV7c91Z3waCSRFP_dgtuT2tPsSqZV6XBWrqn15Ep2nEdCvNjyHuor3-vvgn5nrkn
fvjOZeArhk5sEeM9ysShU27R4JNzG7lFq-8mIz0i8ye1apNIKTpl2a9Q3w"
[xjh@test xujiahui01]$ kubectl  --insecure-skip-tls-verify -s https://127.0.0.1:6443 --token=$SecretT1Token auth can-i --list
Resources                                     Non-Resource URLs                     Resource Names   Verbs
selfsubjectaccessreviews.authorization.k8s.io   []                                    []               [create]
selfsubjectrulesreviews.authorization.k8s.io    []                                    []               [create]
                                              [/.well-known/openid-configuration]   []               [get]
                                              [/api/*]                              []               [get]
                                              [/api]                                []               [get]
                                              [/apis/*]                             []               [get]
                                              [/apis]                               []               [get]
                                              [/healthz]                            []               [get]
                                              [/healthz]                            []               [get]
                                              [/livez]                              []               [get]
                                              [/livez]                              []               [get]
                                              [/openapi/*]                          []               [get]
                                              [/openapi]                            []               [get]
                                              [/openid/v1/jwks]                     []               [get]
                                              [/readyz]                             []               [get]
                                              [/readyz]                             []               [get]
                                              [/version/]                           []               [get]
                                              [/version/]                           []               [get]
                                              [/version]                            []               [get]
                                              [/version]                            []               [get]
secrets                                       []                                    []               [list]
[xjh@test xujiahui01]$
```



```
[xjh@test xujiahui01]$ kubectl  --insecure-skip-tls-verify -s https://127.0.0.1:6443 --token=$SecretT1Token auth can-i list secret
yes
[xjh@test xujiahui01]$ kubectl  --insecure-skip-tls-verify -s https://127.0.0.1:6443 --token=$SecretT1Token auth can-i get secret
no
[xjh@test xujiahui01]$ kubectl  --insecure-skip-tls-verify -s https://127.0.0.1:6443 --token=$SecretT1Token auth can-i watch secret
no
[xjh@test xujiahui01]$
```

虽然无法直接get secret <secret-name> -o yaml来获取secret token，如下图所示：

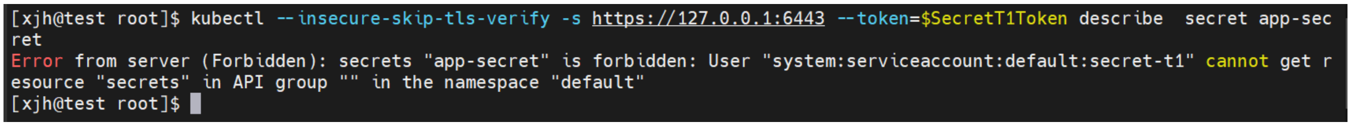


```
[xjh@test xujiahui01]$ kubectl  --insecure-skip-tls-verify -s https://127.0.0.1:6443 --token=$SecretT1Token get secret
NAME                    TYPE                                  DATA   AGE
app-secret              Opaque                                1      300d
default-token-2tl6l     kubernetes.io/service-account-token   3      322d
secret-t1-token-m5kp4   kubernetes.io/service-account-token   3      31m
[xjh@test xujiahui01]$ kubectl  --insecure-skip-tls-verify -s https://127.0.0.1:6443 --token=$SecretT1Token get secret app-secret -o yaml
Error from server (Forbidden): secrets "app-secret" is forbidden: User "system:serviceaccount:default:secret-t1" cannot get resource "secrets" in AP
I group "" in the namespace "default"
[xjh@test xujiahui01]$
```

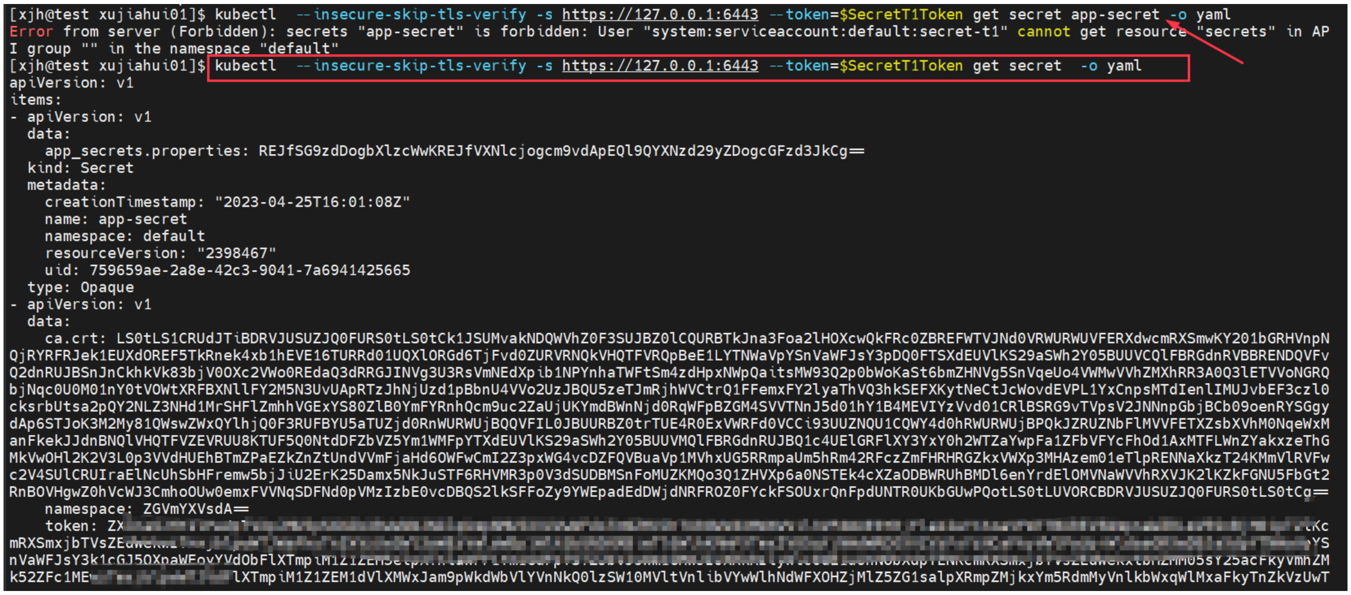
但参考 https://kubernetes.io/docs/concepts/security/rbac-good-practices/#listing-secrets 和 https://cloud.tencent.com/developer/article/2161334

可以直接get secret -A -o yaml 或 get secret -n <namespace> -o yaml 同样可以获取到secret全部内容，只是不针对某个secret对象，和get效果一样：

只有list secrets权限不能使用describe



```
[xjh@test root]$ kubectl --insecure-skip-tls-verify -s https://127.0.0.1:6443 --token=$SecretT1Token describe  secret app-sec
ret
Error from server (Forbidden): secrets "app-secret" is forbidden: User "system:serviceaccount:default:secret-t1" cannot get r
esource "secrets" in API group "" in the namespace "default"
[xjh@test root]$
```



```
[xjh@test xujiahui01]$ kubectl  --insecure-skip-tls-verify -s https://127.0.0.1:6443 --token=$SecretT1Token get secret app-secret -o yaml
Error from server (Forbidden): secrets "app-secret" is forbidden: User "system:serviceaccount:default:secret-t1" cannot get resource "secrets" in AP
I group "" in the namespace "default"
[xjh@test xujiahui01]$ kubectl  --insecure-skip-tls-verify -s https://127.0.0.1:6443 --token=$SecretT1Token get secret  -o yaml
apiVersion: v1
items:
- apiVersion: v1
  data:
    app_secrets.properties: REJfSG9zdDogbXlzcWwKREJfVXNlcjogcm9vdApEQl9QYXNzd29yZDogcGFzd3JkCg==
  kind: Secret
  metadata:
    creationTimestamp: "2023-04-25T16:01:08Z"
    name: app-secret
    namespace: default
    resourceVersion: "2398467"
    uid: 759659ae-2a8e-42c3-9041-7a6941425665
  type: Opaque
- apiVersion: v1
  data:
    ca.crt: LS0tLS1CRUdJTiBDRVJUSUZJQ0FURS0tLS0tCk1JSUMvakNDQWVhZ0F3SUJBZ0lCQURBTkJna3Foa2lHOXcwQkFRc0ZBREFWTVJNd0VRWURWUVFERXdwcmRXSmwKY201bGRHVnpN
QjRYRFRJek1EUXdOREF5TkRnek4xb1hEVE16TURRd01UQXlORGd6TjFvd0ZURVRNQkVHQTFVRQpBeE1LYTNWaVpYSnVaWFJsY3pDQ0FTSXdEUVlKS29aSWh2Y05BUUVCQlFBRGdnRVBBRENDQVFv
Q2dnRUJBSnJnCkhkVk83bjV0OXc2VWo0REdaQ3dRRGJINVg3U3RsVmNEdXpib1NPYnhaTWFtSm4zdHpxNWpQaitsMW93Q2p0bWoKaSt6bmZHNVg5SnVqeUo4VWMwVVhZMXhRR3A0Q3lETVVoNGRQ
bjNqc0U0M01nY0tVOWtXRFBXNllFY2M5N3UvUApRTzJhNjUzd1pBbnU4VVo2UzJBQU5zeTJmRjhWVCtrQ1FFemxFY2lyaThVQ3hkSEFXKytNeCtJcWovdEVPL1YxCnpsMTdIenlIMUJvbEF3czl0
cksrbUtsa2pQY2NLZ3NHd1MrSHFlZmhhVGExYS80ZlB0YmFYRnhQcm9uc2ZaUjUKYmdBWnNjd0RqWFpBZGM4SVVTNnJ5d01hY1B4MEVIYzVvd01CRlBSRG9vTVpsV2JNNnpGbjBCb09oenRYSGgy
dAp6STJoK3M2My81QWswZWxQYlhjQ0F3RUFBYU5aTUZjd0RnWURWUjBQQVFIL0JBUURBZ0trTUE4R0ExVWRFd0VCCi93UUZNQU1CQWY4d0hRWURWUjBPQkJZRUZNbFlMVVFETXZsbXVhM0NqeWxM
anFkekJJdnBNQlVHQTFVZEVRUU8KTUF5Q0NtdDFZbVZ5Ym1WMFpYTXdEUVlKS29aSWh2Y05BUUVMQlFBRGdnRUJBQ1c4UElGRFlXY3YxY0h2WTZaYwpFa1ZFbVFYcFhOd1AxMTFLWnZYakxzeThG
MkVwOHl2K2V3L0p3VVdHUEhBTmZPaEZkZnZtUndVVmFjaHd6OWFwCmI2Z3pxWG4vcDZFQVBuaVp1MVhxUG5RRmpaUm5hRm42RFczZmFHRHRGZkxVWXp3MHAzem01eTlpRENNaXkzT24KMmVlRVFw
c2V4SUlCRUIraElNcUhSbHFremw5bjJiU2ErK25Damx5NkJuSTF6RHVMR3p0V3dSUDBMSnFoMUZKMQo3Q1ZHVXp6a0NSTEk4cXZaODBWRUhBMDl6enYrdElOMVNaWVVhRXVJK2lKZkFGNU5FbGt2
RnBOVHgwZ0hVcWJ3CmhoOUw0emxFVVNqSDFNd0pVMzIzbE0vcDBQS2lkSFFoZy9YWEpadEdDWjdNRFROZ0FYckFSOUxrQnFpdUNTR0UKbGUwPQotLS0tLUVORCBDRVJUSUZJQ0FURS0tLS0tCg==
    namespace: ZGVmYXVsdA==
    token: ZX[redacted]Kc
mRXSmxjbTVsZE[redacted]YS
nVaWFJsY3k1cGJ5OXpaWFoyYVdObFlXTmp[redacted]Y25acFkyVmhZM
k52ZFc1ME[redacted]lXTmpiM1Z1ZEM1dVlXMWxJam9pWkdWbVlYVnNkQ0lzSW10MVltVnlibVYwWlhNdWFXOHZjMlZ5ZG1salpXRmpZMjkxYm5RdmMyVnlkbWxqWlMxaFkyTnZkVzUwT
```

[get secrets]权限

创建绑定get secrets权限的sa账号 secret-t2

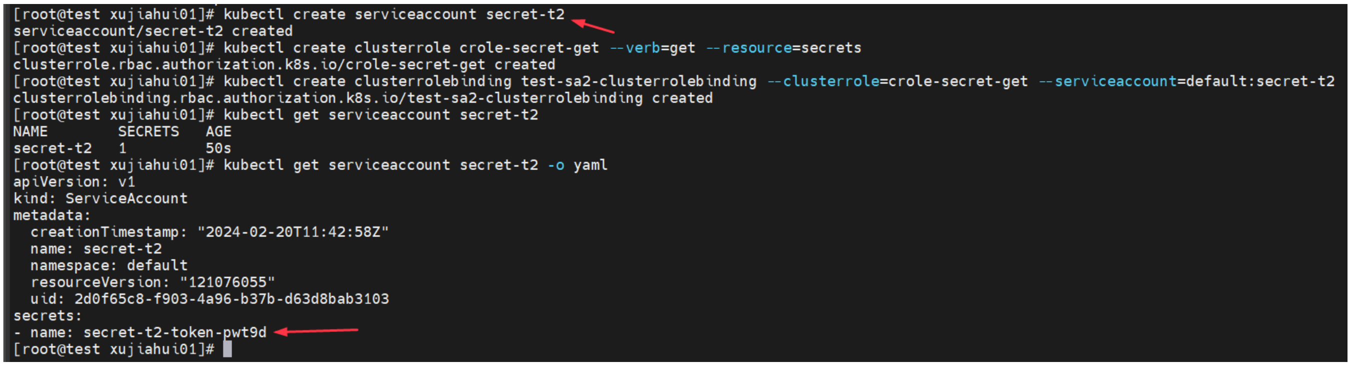
kubectl create serviceaccount secret-t2

创建集群级别角色clusterrole

kubectl create clusterrole crole-secret-get --verb=get --resource=secrets

角色绑定

kubectl create clusterrolebinding test-sa2-clusterrolebinding --clusterrole=crole-secret-get --serviceaccount=default:secret-t2



```
[root@test xujiahui01]# kubectl create serviceaccount secret-t2
serviceaccount/secret-t2 created
[root@test xujiahui01]# kubectl create clusterrole crole-secret-get --verb=get --resource=secrets
clusterrole.rbac.authorization.k8s.io/crole-secret-get created
[root@test xujiahui01]# kubectl create clusterrolebinding test-sa2-clusterrolebinding --clusterrole=crole-secret-get --serviceaccount=default:secret-t2
clusterrolebinding.rbac.authorization.k8s.io/test-sa2-clusterrolebinding created
[root@test xujiahui01]# kubectl get serviceaccount secret-t2
NAME        SECRETS   AGE
secret-t2   1         50s
[root@test xujiahui01]# kubectl get serviceaccount secret-t2 -o yaml
apiVersion: v1
kind: ServiceAccount
metadata:
  creationTimestamp: "2024-02-20T11:42:58Z"
  name: secret-t2
  namespace: default
  resourceVersion: "121076055"
  uid: 2d0f65c8-f903-4a96-b37b-d63d8bab3103
secrets:
- name: secret-t2-token-pwt9d
[root@test xujiahui01]#
```

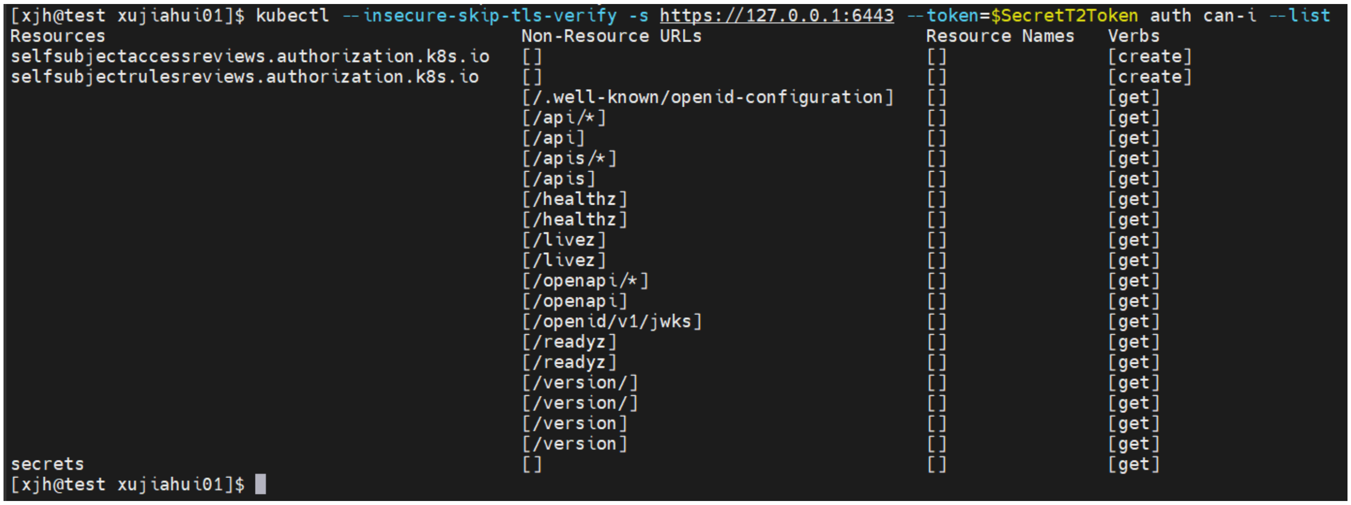
获取复制使用secret-token

kubectl get secret secret-t2-token-pwt9d -o yaml |grep token: |awk '{print $2}'|base64 -d

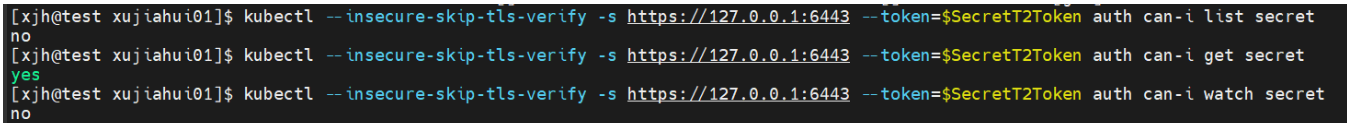
查看是否拥有某个权限

kubectl --insecure-skip-tls-verify -s https://127.0.0.1:6443 --token=$SecretT2Token auth can-i --list

kubectl --insecure-skip-tls-verify -s https://127.0.0.1:6443 --token=$SecretT2Token auth can-i list secret

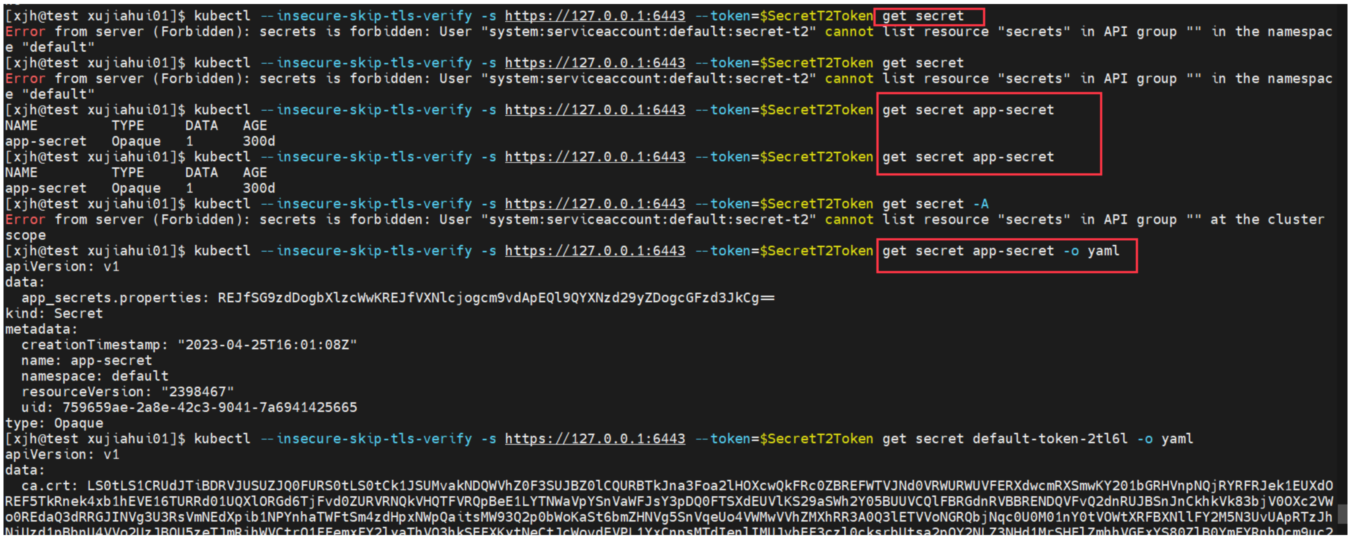


```
[xjh@test xujiahui01]$ kubectl --insecure-skip-tls-verify -s https://127.0.0.1:6443 --token=$SecretT2Token auth can-i --list
Resources                                       Non-Resource URLs                     Resource Names   Verbs
selfsubjectaccessreviews.authorization.k8s.io   []                                    []               [create]
selfsubjectrulesreviews.authorization.k8s.io    []                                    []               [create]
                                                [/.well-known/openid-configuration]   []               [get]
                                                [/api/*]                              []               [get]
                                                [/api]                                []               [get]
                                                [/apis/*]                             []               [get]
                                                [/apis]                               []               [get]
                                                [/healthz]                            []               [get]
                                                [/healthz]                            []               [get]
                                                [/livez]                              []               [get]
                                                [/livez]                              []               [get]
                                                [/openapi/*]                          []               [get]
                                                [/openapi]                            []               [get]
                                                [/openid/v1/jwks]                     []               [get]
                                                [/readyz]                             []               [get]
                                                [/readyz]                             []               [get]
                                                [/version/]                           []               [get]
                                                [/version/]                           []               [get]
                                                [/version]                            []               [get]
                                                [/version]                            []               [get]
secrets                                         []                                    []               [get]
[xjh@test xujiahui01]$ 
```



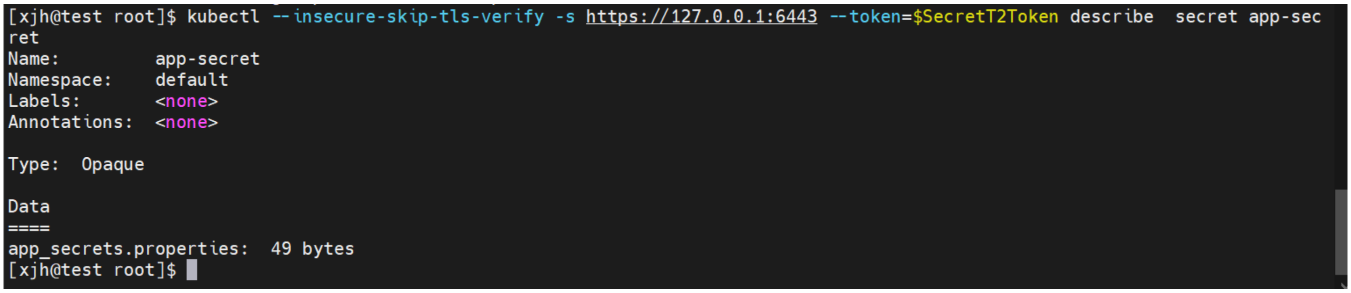
```
[xjh@test xujiahui01]$ kubectl --insecure-skip-tls-verify -s https://127.0.0.1:6443 --token=$SecretT2Token auth can-i list secret
no
[xjh@test xujiahui01]$ kubectl --insecure-skip-tls-verify -s https://127.0.0.1:6443 --token=$SecretT2Token auth can-i get secret
yes
[xjh@test xujiahui01]$ kubectl --insecure-skip-tls-verify -s https://127.0.0.1:6443 --token=$SecretT2Token auth can-i watch secret
no
```

只有get权限时，不能获取secret列表，kubectl get secret 在不写明具体的secretname时都会报错，只有指定具体的secret-name才能显示内容，不指定无法查看获取secret名



```
[xjh@test xujiahui01]$ kubectl --insecure-skip-tls-verify -s https://127.0.0.1:6443 --token=$SecretT2Token get secret
Error from server (Forbidden): secrets is forbidden: User "system:serviceaccount:default:secret-t2" cannot list resource "secrets" in API group "" in the namespac
e "default"
[xjh@test xujiahui01]$ kubectl --insecure-skip-tls-verify -s https://127.0.0.1:6443 --token=$SecretT2Token get secret
Error from server (Forbidden): secrets is forbidden: User "system:serviceaccount:default:secret-t2" cannot list resource "secrets" in API group "" in the namespac
e "default"
[xjh@test xujiahui01]$ kubectl --insecure-skip-tls-verify -s https://127.0.0.1:6443 --token=$SecretT2Token get secret app-secret
NAME         TYPE     DATA   AGE
app-secret   Opaque   1      300d
[xjh@test xujiahui01]$ kubectl --insecure-skip-tls-verify -s https://127.0.0.1:6443 --token=$SecretT2Token get secret app-secret
NAME         TYPE     DATA   AGE
app-secret   Opaque   1      300d
[xjh@test xujiahui01]$ kubectl --insecure-skip-tls-verify -s https://127.0.0.1:6443 --token=$SecretT2Token get secret -A
Error from server (Forbidden): secrets is forbidden: User "system:serviceaccount:default:secret-t2" cannot list resource "secrets" in API group "" at the cluster
scope
[xjh@test xujiahui01]$ kubectl --insecure-skip-tls-verify -s https://127.0.0.1:6443 --token=$SecretT2Token get secret app-secret -o yaml
apiVersion: v1
data:
  app_secrets.properties: REJfSG9zdDogbXlzcWwKREJfVXNlcjogcm9vdApEQl9QYXNzd29yZDogcGFzd3JkCg==
kind: Secret
metadata:
  creationTimestamp: "2023-04-25T16:01:08Z"
  name: app-secret
  namespace: default
  resourceVersion: "2398467"
  uid: 759659ae-2a8e-42c3-9041-7a6941425665
type: Opaque
[xjh@test xujiahui01]$ kubectl --insecure-skip-tls-verify -s https://127.0.0.1:6443 --token=$SecretT2Token get secret default-token-2tl6l -o yaml
apiVersion: v1
data:
  ca.crt: LS0tLS1CRUdJTiBDRVJUSUZJQ0FURS0tLS0tCk1JSUMvakNDQWVhZ0F3SUJBZ0lCQURBTkJna3Foa2lHOXcwQkFRc0ZBREFWTVJNd0VRWURWUVFERXdwcmRXSmwKY201bGRHVnpNQjRYRFRJek1EUXdO
REF5TkRnek4xb1hEVE16TURRd01UQXlORGd6TjFvd0ZURVRNQkVHQTFVRQpBeE1LYTNWaVpYSnVaWFJsY3pDQ0FTSXdEUVlKS29aSWh2Y05BUUVCQlFBRGdnRVBBRENDQVFvQ2dnRUJBSnJnCkhkVk83bjV0OXc2VW
o0REdaQ3dRRGJINVg3U3RsVmNEdXpib1NPYnhaTWFtSm4zdHpxNWpQaitsMW93Q2p0bWoKaSt6bmZHNVg5SnVqeUo4VWMwVVhZMXhRR3A0Q3lETVVoNGRQbjNqc0U0M01nY0tVOWtXRFBXNllFY2M5N3UvUApRTzJh
NjUzd1pBbnU4VVo2UzJBOU5zeTJmRjhWVCtrO1FEemxFY2lvaThVO3hkSEFXKytNeCtJcWoydEVPL1YxCnpsMTdIenlIMUJvbFF3czl0cksrbUtsa2pOY2NLZ3NHd1MrSHFlZmhhVGExYS80ZlB0YmFYRnhOcm9uc2
```

describe也需要指定具体的secret-name



```
[xjh@test root]$ kubectl --insecure-skip-tls-verify -s https://127.0.0.1:6443 --token=$SecretT2Token describe  secret app-sec
ret
Name:         app-secret
Namespace:    default
Labels:       <none>
Annotations:  <none>

Type:  Opaque

Data
====
app_secrets.properties:  49 bytes
[xjh@test root]$ 
```

[watch secrets]权限

创建绑定watch secrets权限的sa账号 secret-t3

kubectl create serviceaccount secret-t3

创建集群级别角色clusterrole

kubectl create clusterrole crole-secret-watch --verb=watch --resource=secrets

角色绑定

kubectl create clusterrolebinding test-sa3-clusterrolebinding --clusterrole=crole-secret-watch --serviceaccount=default:secret-t3

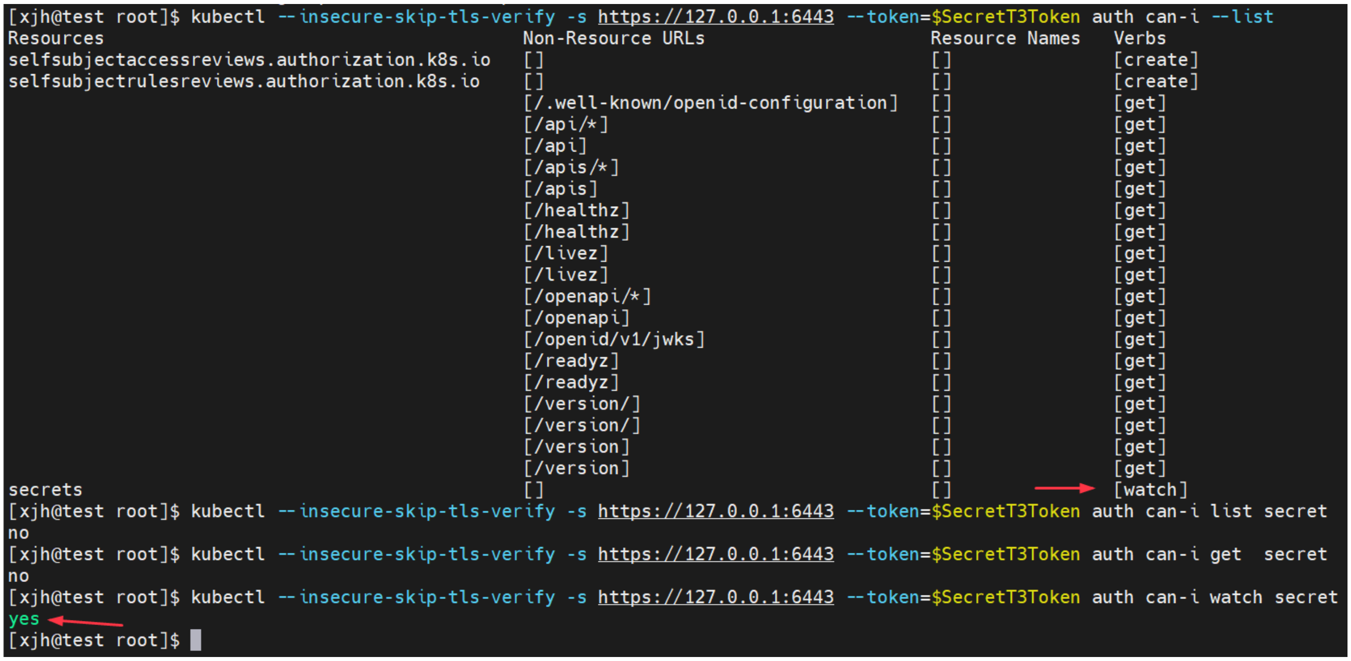
获取复制使用secret-token

kubectl get secret secret-t3-token-dx5rk -o yaml |grep token: |awk '{print $2}'|base64 -d

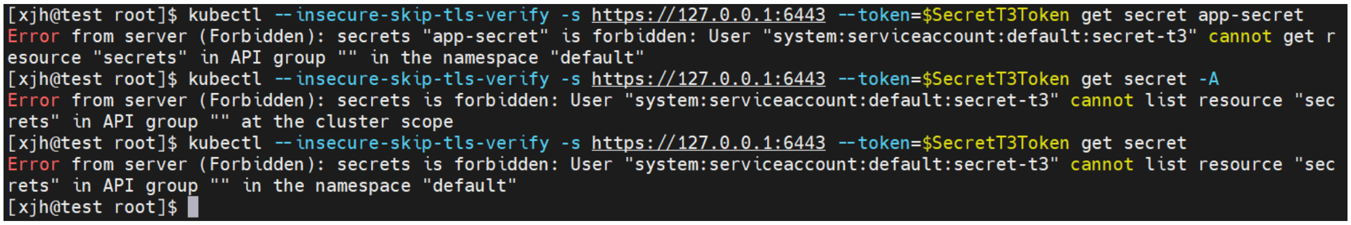
查看是否拥有某个权限

kubectl --insecure-skip-tls-verify -s https://127.0.0.1:6443 --token=$SecretT3Token auth can-i --list

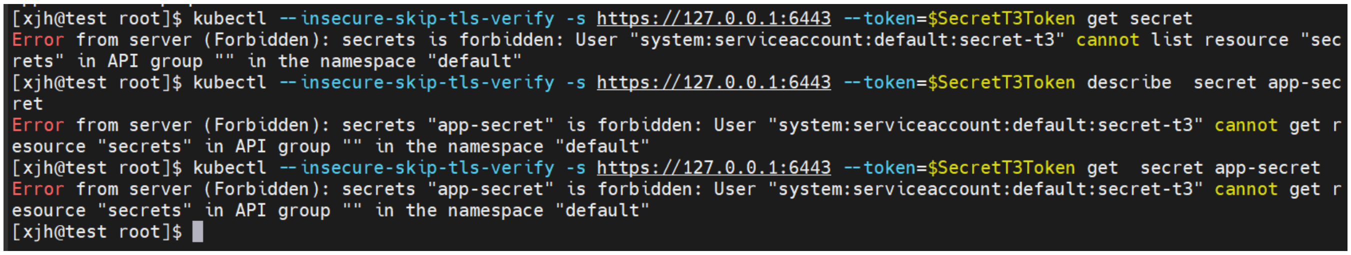
只有watch权限时，无法获取到secret内容；get describe都不能用



```
[xjh@test root]$ kubectl --insecure-skip-tls-verify -s https://127.0.0.1:6443 --token=$SecretT3Token auth can-i --list
Resources                                       Non-Resource URLs                     Resource Names   Verbs
selfsubjectaccessreviews.authorization.k8s.io   []                                    []               [create]
selfsubjectrulesreviews.authorization.k8s.io    []                                    []               [create]
                                                [/.well-known/openid-configuration]   []               [get]
                                                [/api/*]                              []               [get]
                                                [/api]                                []               [get]
                                                [/apis/*]                             []               [get]
                                                [/apis]                               []               [get]
                                                [/healthz]                            []               [get]
                                                [/healthz]                            []               [get]
                                                [/livez]                              []               [get]
                                                [/livez]                              []               [get]
                                                [/openapi/*]                          []               [get]
                                                [/openapi]                            []               [get]
                                                [/openid/v1/jwks]                     []               [get]
                                                [/readyz]                             []               [get]
                                                [/readyz]                             []               [get]
                                                [/version/]                           []               [get]
                                                [/version/]                           []               [get]
                                                [/version]                            []               [get]
                                                [/version]                            []               [get]
secrets                                         []                                    []               [watch]
[xjh@test root]$ kubectl --insecure-skip-tls-verify -s https://127.0.0.1:6443 --token=$SecretT3Token auth can-i list secret
no
[xjh@test root]$ kubectl --insecure-skip-tls-verify -s https://127.0.0.1:6443 --token=$SecretT3Token auth can-i get  secret
no
[xjh@test root]$ kubectl --insecure-skip-tls-verify -s https://127.0.0.1:6443 --token=$SecretT3Token auth can-i watch secret
yes
[xjh@test root]$ 
```



```
[xjh@test root]$ kubectl --insecure-skip-tls-verify -s https://127.0.0.1:6443 --token=$SecretT3Token get secret app-secret
Error from server (Forbidden): secrets "app-secret" is forbidden: User "system:serviceaccount:default:secret-t3" cannot get r
esource "secrets" in API group "" in the namespace "default"
[xjh@test root]$ kubectl --insecure-skip-tls-verify -s https://127.0.0.1:6443 --token=$SecretT3Token get secret -A
Error from server (Forbidden): secrets is forbidden: User "system:serviceaccount:default:secret-t3" cannot list resource "sec
rets" in API group "" at the cluster scope
[xjh@test root]$ kubectl --insecure-skip-tls-verify -s https://127.0.0.1:6443 --token=$SecretT3Token get secret
Error from server (Forbidden): secrets is forbidden: User "system:serviceaccount:default:secret-t3" cannot list resource "sec
rets" in API group "" in the namespace "default"
[xjh@test root]$ 
```



```
[xjh@test root]$ kubectl --insecure-skip-tls-verify -s https://127.0.0.1:6443 --token=$SecretT3Token get secret
Error from server (Forbidden): secrets is forbidden: User "system:serviceaccount:default:secret-t3" cannot list resource "sec
rets" in API group "" in the namespace "default"
[xjh@test root]$ kubectl --insecure-skip-tls-verify -s https://127.0.0.1:6443 --token=$SecretT3Token describe  secret app-sec
ret
Error from server (Forbidden): secrets "app-secret" is forbidden: User "system:serviceaccount:default:secret-t3" cannot get r
esource "secrets" in API group "" in the namespace "default"
[xjh@test root]$ kubectl --insecure-skip-tls-verify -s https://127.0.0.1:6443 --token=$SecretT3Token get  secret app-secret
Error from server (Forbidden): secrets "app-secret" is forbidden: User "system:serviceaccount:default:secret-t3" cannot get r
esource "secrets" in API group "" in the namespace "default"
[xjh@test root]$ 
```

impersonate(用户伪装)

https://kubernetes.io/zh-cn/docs/reference/access-authn-authz/authentication/#user-impersonation

Kubernetes 还持模拟(Impersonation)；也就是说，个可以充当另个。例如，作为集群管理员，您可以使模拟来调试任何授权问题。

用户模拟常用作调试，kubectl 客户端有一个子命令是auth can-i可以快速检查执行者是否有对应的API权限，该命令使用SelfSubjectAccessReview API来确定当前用户是否可以执行给定的操作，并且无论使用何种授权模式都可以工作。

```
kubectl auth can-i create deployments --namespace dev
```

输出是 yes 证明有权限，否则就是 no。

https://blog.51cto.com/wzlinux/4937662 参考这个 管理员可以将此与user impersonation结合使用，以确定其他用户可以执行的操作。 伪装成用户 dave 来探测是否有相应的权限。

```
kubectl auth can-i list secrets --namespace dev --as dave
```