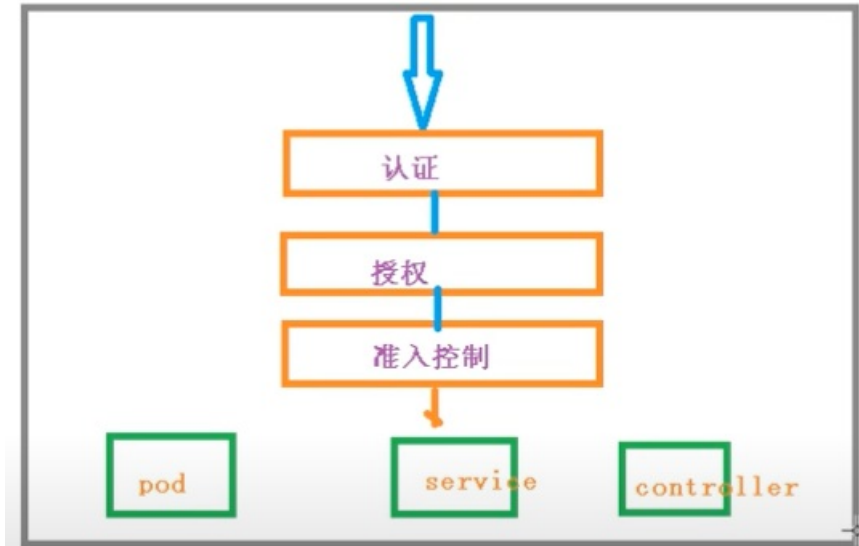


K8s 安全機制

以下是圖片的文字辨識及繁體中文翻譯：

Kubernetes 集群安全機制



1. 概述

1. 訪問 k8s 集群時，需要經過三個步驟完成具體操作：
 - 第一步：認證
 - 第二步：授權
 - 第三步：準入控制
2. 訪問過程：
 - 訪問時需要經過 `apiserver`，`apiserver` 做統一認證和驗證，例如門衛。
 - 訪問過程中需要 證書、token，或者用 用戶名+密碼。
 - 如果訪問 Pod，還需要 `serviceAccount`。

第一步 認證：傳輸安全

- 傳輸安全：
 - 對外不暴露 8080 端口，只能內部訪問，對外使用端口 6443。
- 認證方式：
 - `https` 證書認證：基於 CA 證書。
 - `http` token 認證：通過 token 識別用戶。
 - `http` 基本認證：使用用戶名+密碼認證。

第二步 授權：

- 基於 RBAC 進行授權操作。
- 基於角色訪問控制。

第三步 準入控制：

- 準入控制器的列表，根據列表決定是否允許執行操作。

2. RBAC

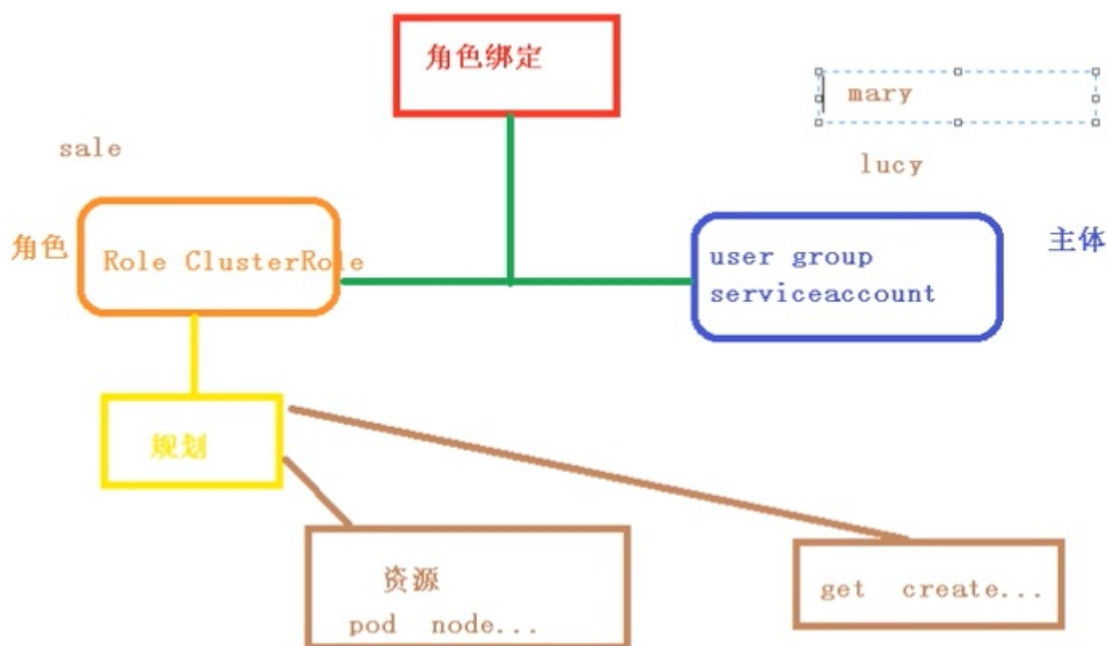
基於角色的訪問控制

- **角色：**
 - **role**：特定命名空間訪問權限
 - **ClusterRole**：所有命名空間訪問權限
- **角色綁定：**
 - **roleBinding**：角色綁定到主體
 - **ClusterRoleBinding**：集群角色綁定到主體

主體：

- **user**：用戶
- **group**：用戶組
- **serviceAccount**：服務帳號

這是基於 RBAC 的角色和權限管理，適用於 Kubernetes 的訪問控制。如需進一步說明或具體範例，請隨時告訴我！☺



以下是圖片中的文字辨識及翻譯成繁體中文：

1. 創建命名空間

```
[root@m1 lucy]# kubectl create ns roledemo
namespace/roledemo created
```

2. 在新創建的命名空間中創建 Pod

```
[root@m1]# kubectl run nginx --image=nginx -n roledemo
```

3. 創建角色

```
[root@m1]# vi rbac-role.yaml
[root@m1]# kubectl apply -f rbac-role.yaml
role.rbac.authorization.k8s.io/pod-reader created
```

```
[root@m1]# kubectl get role -n roledemo
NAME      AGE
pod-reader 19s
```

4. 創建角色綁定

```
[root@m1]# vi rbac-rolebinding.yaml
[root@m1]# kubectl apply -f rbac-rolebinding.yaml
rolebinding.rbac.authorization.k8s.io/read-pods created

[root@m1]# kubectl get rolebinding -n roledemo
NAME      AGE
read-pods 15s
```

5. 使用證書識別身份

```
[root@m1 mary]# vi rbac-user.sh
[root@m1 mary]# cp /root/TLS/k8s/ca* ./
[root@m1 mary]# bash rbac-user.sh
2020/09/03 17:03:10 [INFO] generate received request
2020/09/03 17:03:10 [INFO] received CSR

[root@m1 mary]# kubectl get pods -n roledemo
```

說明：

1. **命名空間** 用於區分不同的資源和工作負載。
2. **角色和角色綁定** 控制對特定命名空間的訪問權限。
3. **使用證書識別身份** 是基於 RBAC 的身份驗證機制，確保用戶能正確訪問資源。

🔄修訂版本 #2

★由 treeman 建立於 20 2025 10:50:14

✍由 treeman 更新於 20 2025 11:12:02