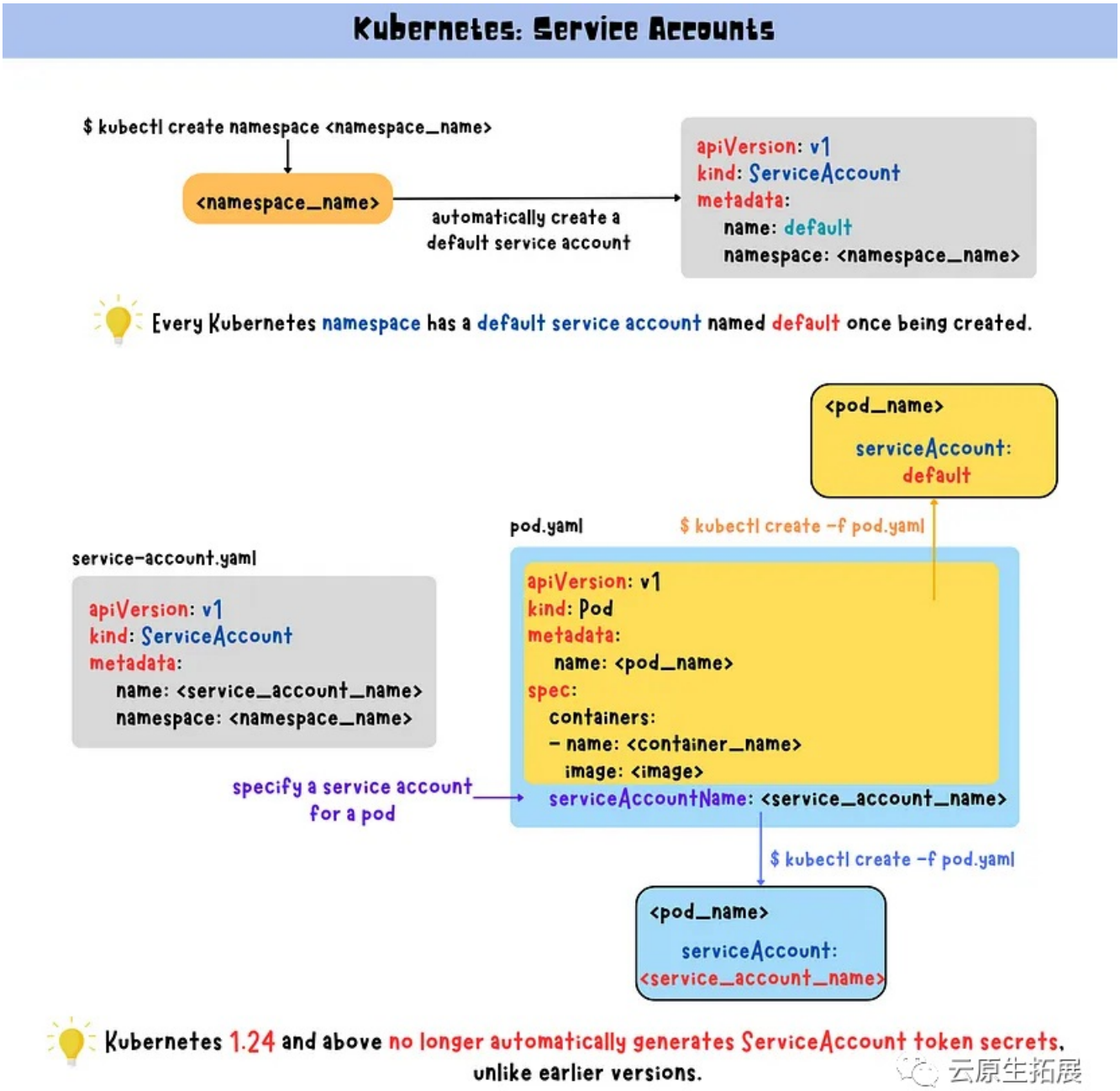
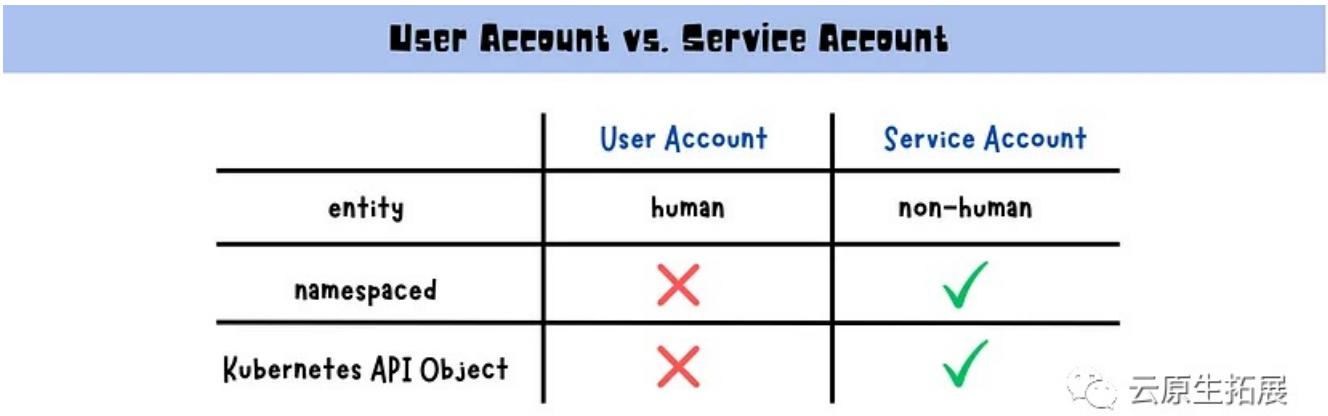
108Kubernetes 系列（一零一）关于 ServiceAccount 的故事

Kubernetes 中的 ServiceAccount 是非人类帐户，为系统组件和应用程序 Pod 提供唯一的身份。它们是 Kubernetes API 服务器中的命名空间级对象。每个 Kubernetes 命名空间都有一个名为 default 的默认 ServiceAccount，该帐户没有分配特殊的角色或权限。在 1.24 之前的 Kubernetes 版本中，当创建 ServiceAccount 并将其挂载到 Pod 的文件系统中时，会自动生成令牌。但是，从 Kubernetes 1.24 开始，令牌不再自动生成，必须使用 TokenRequest API 或通过为令牌控制器创建 Secret API 对象来获取，以使用 ServiceAccount 令牌填充。



ServiceAccount

User Account VS Service Account

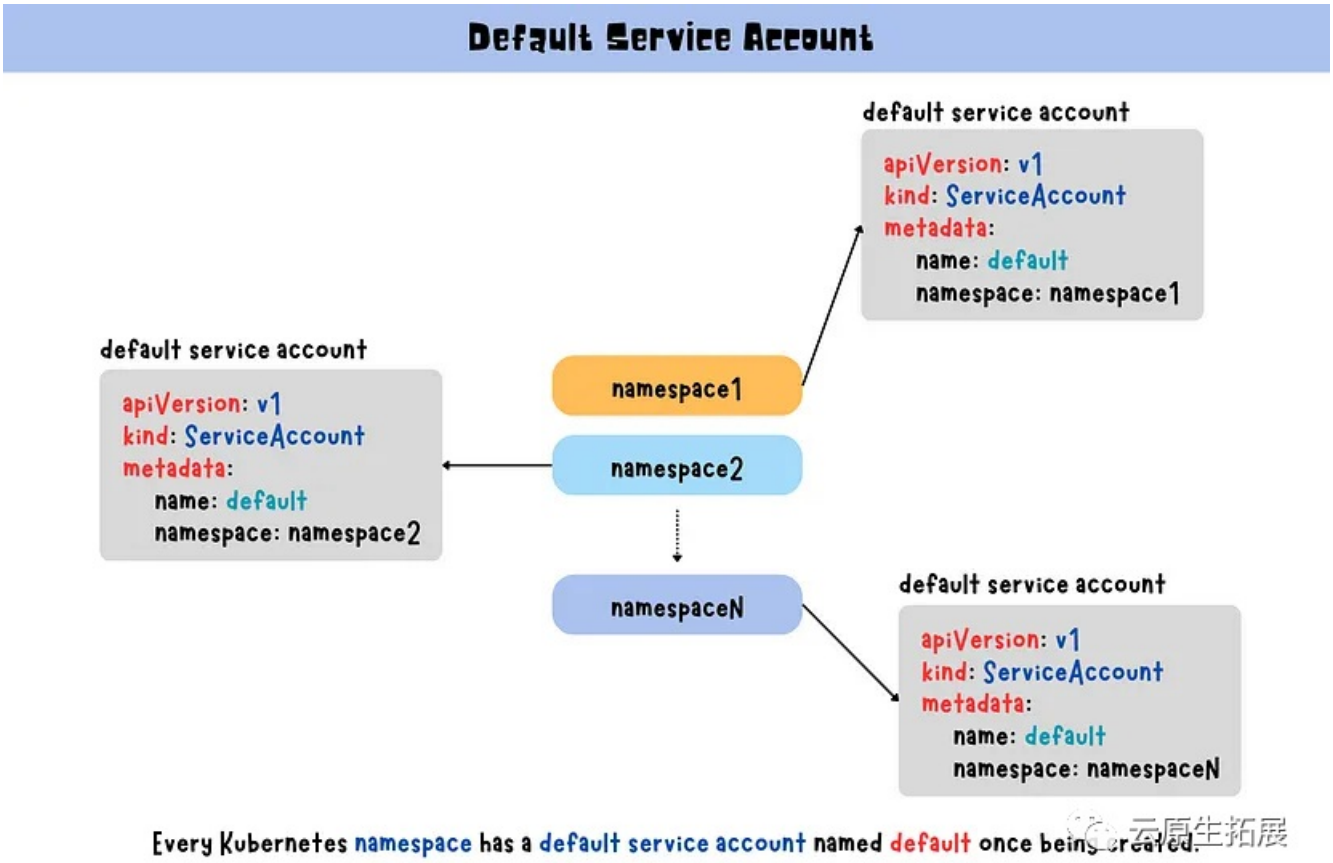


Service Account 是一种非人类帐户，它为 Kubernetes 集群中的应用程序 Pod 和系统组件等实体提供独特的身份。它们是命名空间级的，这意味着它们绑定到特定的 Kubernetes 命名空间，并且可以使用 Kubernetes RBAC 或其他授权机制轻松管理。Service Account 作为 Kubernetes API 服务器中的对象存在。

用户通常是通过外部系统（例如 LDAP 或 Active Directory）进行身份验证和管理的人。虽然用户可以使用 TLS 证书在 Kubernetes 中进行身份验证和授权，但需要设置更复杂的基础设施来管理用户身份和访问控制。与Service Account 不同，用户是全局的，并不表示为 Kubernetes API 服务器中的对象。

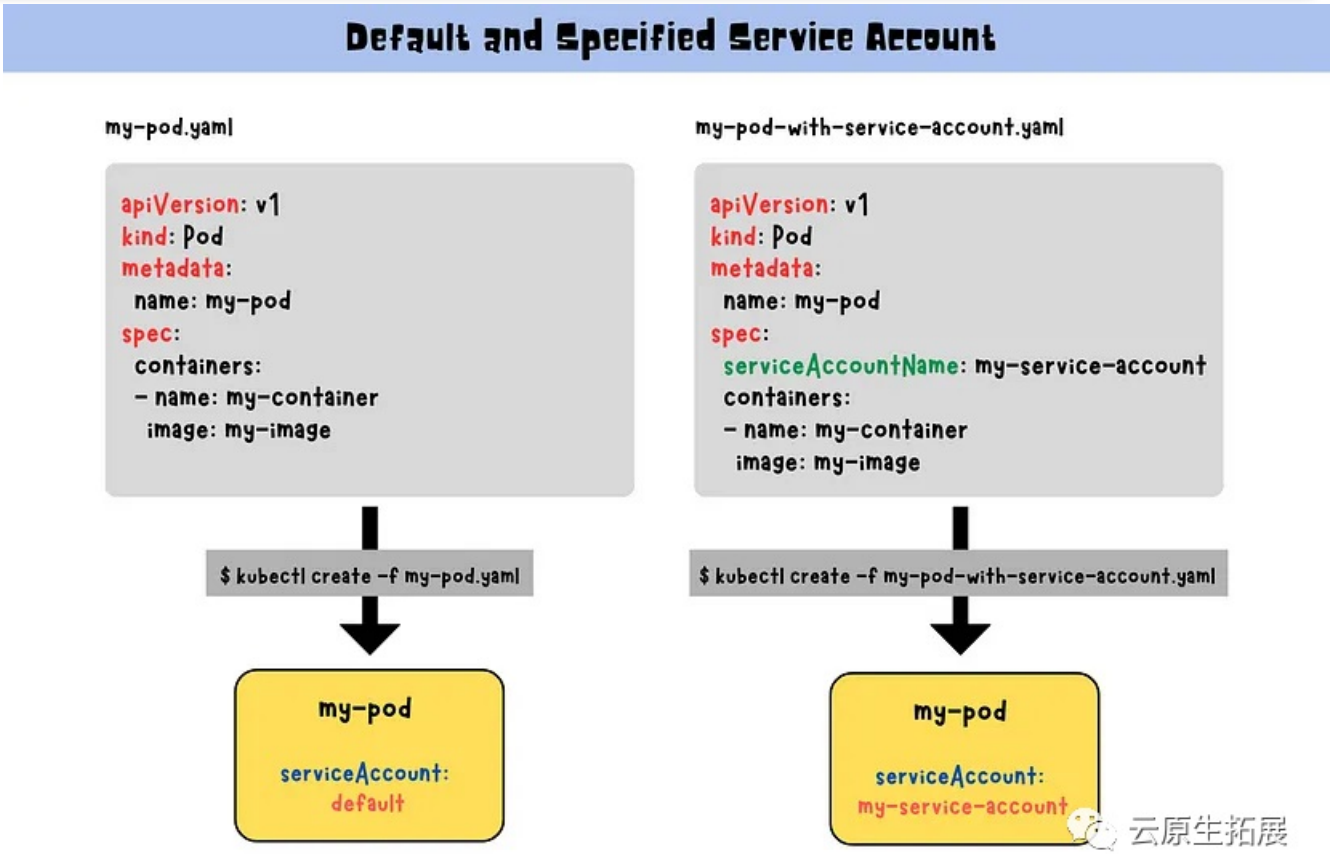
默认 Service Account

每个 Kubernetes 命名空间都有一个名为 **default** 的默认 ServiceAccount，该帐户在创建命名空间时自动创建。默认情况下，此 default ServiceAccount 没有分配给它的特殊权限或角色。



如果创建 pod 时未指定 Service Account，它将使用 default ServiceAccount。但是，您还可以通过在 pod 的 YAML 配置文件中包含 spec.serviceAccountName 字段来显式指定 pod 要使用的 ServiceAccount。例如：

```yaml
apiVersion: v1
kind: Pod
metadata:
  name: my-pod
spec:
  serviceAccountName: my-service-account
  containers:
  - name: my-container
    image: my-image
```



Service Account Token

在 Kubernetes 版本 1.24 之前，ServiceAccount 令牌 secret 是在创建 ServiceAccount 时自动生成的。创建 Pod 时，与 ServiceAccount 关联的令牌会自动挂载到 Pod 的文件系统中。令牌的挂载路径通常为 /var/run/secrets/kubernetes.io/serviceaccount 。然而，从 Kubernetes 1.24 开始，这个流程发生了变化。 ServiceAccount 令牌 Secret 不再自动生成。相反，您可以使用 TokenRequest API 获取 ServiceAccount 令牌或为令牌控制器创建 Secret API 对象以使用 ServiceAccount 令牌填充。

简称：sa

```
$ kubectl api-resources
NAME            SHORTNAMES   APIVERSION    NAMESPACED   KIND
serviceaccounts  sa           v1            true         ServiceAccount
```

ServiceAccount 的 YAML

```yaml
apiVersion: v1
kind: ServiceAccount
metadata:
  name: <service_account_name>
  namespace: <namespace_name>
```

命令

Commands(ServiceAccount)

Commands	Functionality in Kubernetes
$ kubectl create serviceaccount <service_account_name>	Create a new service account in the current namespace.
$ kubectl get serviceaccounts	List all the service accounts in the current namespace.
$ kubectl describe serviceaccount <service_account_name>	Retrieve detailed information about a specific service account.
$ kubectl create token <service_account_name>	Create a new token associated with the specified service account.

1. 在当前命名空间中创建一个新的 ServiceAccount:

```
$ kubectl create serviceaccount <service_account_name>
```

2. 列出当前命名空间中的所有 ServiceAccount:

```
$ kubectl get serviceaccounts
```

3. 检索有关特定 ServiceAccount 的详细信息:

```
$ kubectl describe serviceaccount <service_account_name>
```

4. 创建与指定 ServiceAccount 关联的新令牌:

```
$ kubectl create token <service_account_name>
```