Kubernetes-Pod Lifecycle

Pod Life Cycle Phases:

1. **Pending**: The pod is created but has not yet been scheduled.
2. **Running**: The pod is running and executing containers.
3. **Succeeded**: The pod has completed its execution successfully.
4. **Failed**: The pod has failed or crashed.
5. **Unknown**: The pod's status is unknown.

Pod Lifecycle Management:

Kubernetes provides various controllers, APIs, and commands to manage the life cycle of pods. Here are some key concepts:

- **Pod controller**: A controller responsible for managing a specific type of pod (e.g., Deployment, ReplicaSet).
- **Create**: Use `kubectl create` or API calls to create a new pod.
- **Update**: Update an existing pod's configuration using `kubectl update` or API calls.
- **Scale**: Scale the number of replicas for a pod controller using `kubectl scale`.
- **Delete**: Delete a pod using `kubectl delete` or API calls.

Example:

Suppose we have a simple web application with two containers (web server and database) that run in the same pod. We want to create, manage, and delete this pod using Kubernetes APIs.

Here's an example YAML file (`pod.yaml`) for creating a pod:

```
 apiVersion: v1
kind: Pod
metadata:
  name: web-app-pod
spec:
  containers:
  - name: web-server
    image: nginx
  - name: database
    image: mysql
```

We can create this pod using `kubectl create` or API calls:

```
kubectl create -f pod.yaml
```

To manage the life cycle of the pod, we can use various Kubernetes commands and APIs. For example:

- To get the pod's status, use `kubectl get pod web-app-pod`.
- To update the pod's configuration, run `kubectl update -f pod.yaml`.
- To scale the number of replicas for this pod controller (if it were a Deployment or ReplicaSet), use `kubectl scale deployment <deployment-name> --replicas=3`.

When we're done with the pod and want to delete it, we can use `kubectl delete` or API calls:

```
kubectl delete -f pod.yaml
```

Note that in this example, we're using a YAML file to create and manage the pod. In real-world scenarios, you'd likely use Kubernetes' configuration management tools (e.g., Helm, Kustomize) to manage your applications.

This summary provides an overview of pod lifecycle management with examples of creating, updating, scaling, and deleting pods in Kubernetes.

Curated by Brajesh Kumar