MANAGING STATE WITH DEPLOYMENTS

Managing State with Deployments

Deployment Configuration Pod Template

Next, we will take a look at a configuration template for the pods to be deployed. We will see some similar values.

```
template:
  metadata:
    creationTimestamp: null
    labels:
      app: dev-web
  spec:
    containers:
    - image: nginx:1.17.7-alpine
      imagePullPolicy: IfNotPresent
      name: dev-web
      resources: {}
      terminationMessagePath: /dev/termination-log
      terminationMessagePolicy: File
    dnsPolicy: ClusterFirst
    restartPolicy: Always
    schedulerName: default-scheduler
    securityContext: {}
    terminationGracePeriodSeconds: 30
```

If the meaning is basically the same as what was defined before, we will not repeat the definition.

Click on the boxes to learn more about each configuration element.

Explanation of Configuration Elements

template Close

Data being passed to the ReplicaSet to determine how to deploy an object (in this case, containers).

containers

Close

Key word indicating that the following items of this indentation are for a container.

image

Close

This is the image name passed to the container engine, typically Docker. The engine will pull the image and create the Pod.

imagePullPolicy

Close

Policy settings passed along to the container engine, about when and if an image should be downloaded or used from a local cache.

name

Close

The leading stub of the Pod names. A unique string will be appended.

resources

Close

By default, empty. This is where you would set resource restrictions and settings, such as a limit on CPU or memory for the containers.

terminationMessagePath

Close

A customizable location of where to output success or failure information of a container.

terminationMessagePolicy

Close

The default value is **`File`**, which holds the termination method. It could also be set to **`FallbackToLogsOnError`**, which will use the last chunk of container log if the message file is empty and the container shows an error.

dnsPolicy

Close

Determines if DNS queries should go to **coredns** or, if set to **`Default`**, use the node's DNS resolution configuration.

restartPolicy

Close

Should the container be restarted if killed? Automatic restarts are part of the typical strength of Kubernetes.

scheduleName

Close

Allows for the use of a custom scheduler, instead of the Kubernetes default.

securityContext

Close

Flexible setting to pass one or more security settings, such as SELinux context, AppArmor values, users and UIDs for the containers to use.

terminationGracePeriodSeconds

Close

The amount of time to wait for a **`SIGTERM`** to run until a **`SIGKILL`** is used to terminate the container.