

Kubernetes Day-5 With Amazon Q and MCP servers

```
> can you create kubernetes manifest files for statefulsets, configmaps and secrets
I'll help you create Kubernetes manifest files for StatefulSets, ConfigMaps, and Secrets. Let me create these files for you with proper examples and explanations.

Let's start by creating these manifest files in a new directory to keep them organized:

✂ Using tool: execute_bash
...
• I will run the following shell command:
mkdir -p k8s-manifests
...
↳ Purpose: Creating a directory to store Kubernetes manifest files

Allow this action? Use 't' to trust (always allow) this tool for the session. [y/n/t]:

> t
...
• Completed in 0.9s

Now I'll create the manifest files:

## 1. StatefulSet Manifest

✂ Using tool: fs_write
...
• Path: k8s-manifests/statefulset.yaml
+ 1: apiVersion: apps/v1
+ 2: kind: StatefulSet
+ 3: metadata:
+ 4:   name: web
+ 5:   namespace: default
+ 6: spec:
+ 7:   selector:
+ 8:     matchLabels:
+ 9:       app: nginx # has to match .spec.template.metadata.labels
+ 10:   serviceName: "nginx"
+ 11:   replicas: 3 # by default is 1
+ 12:   minReadySeconds: 10 # by default is 0
+ 13:   template:
+ 14:     metadata:
+ 15:       labels:
+ 16:         app: nginx # has to match .spec.selector.matchLabels
+ 17:     spec:
+ 18:       terminationGracePeriodSeconds: 10
+ 19:       containers:
+ 20:         - name: nginx
+ 21:           image: nginx:1.25.1
+ 22:           ports:
+ 23:             - containerPort: 80
+ 24:               name: web
+ 25:           volumeMounts:
+ 26:             - name: www
+ 27:               mountPath: /usr/share/nginx/html
+ 28:       volumeClaimTemplates:
+ 29:         - metadata:
+ 30:           name: www
```

```
+ 5: namespace: default
+ 6: spec:
+ 7:   selector:
+ 8:     matchLabels:
+ 9:       app: nginx # has to match .spec.template.metadata.labels
+ 10:   serviceName: "nginx"
+ 11:   replicas: 3 # by default is 1
+ 12:   minReadySeconds: 10 # by default is 0
+ 13:   template:
+ 14:     metadata:
+ 15:       labels:
+ 16:         app: nginx # has to match .spec.selector.matchLabels
+ 17:     spec:
+ 18:       terminationGracePeriodSeconds: 10
+ 19:       containers:
+ 20:       - name: nginx
+ 21:         image: nginx:1.25.1
+ 22:         ports:
+ 23:         - containerPort: 80
+ 24:           name: web
+ 25:         volumeMounts:
+ 26:         - name: www
+ 27:           mountPath: /usr/share/nginx/html
+ 28:       volumeClaimTemplates:
+ 29:       - metadata:
+ 30:         name: www
+ 31:         spec:
+ 32:           accessModes: [ "ReadWriteOnce" ]
+ 33:           storageClassName: "standard"
+ 34:           resources:
+ 35:             requests:
+ 36:               storage: 1Gi
+ 37: ---
+ 38: # Headless service for StatefulSet
+ 39: apiVersion: v1
+ 40: kind: Service
+ 41: metadata:
+ 42:   name: nginx
+ 43:   namespace: default
+ 44:   labels:
+ 45:     app: nginx
+ 46: spec:
+ 47:   ports:
+ 48:   - port: 80
+ 49:     name: web
+ 50:   clusterIP: None
+ 51:   selector:
+ 52:     app: nginx
```

Allow this action? Use 't' to trust (always allow) this tool for the session. [y/n/t]:

> t

```
...
● Completed in 0.5s

I've created the Kubernetes manifest files for you in a directory called k8s-manifests/. Here's what I've included:

1. StatefulSet (statefulset.yaml):
  • A StatefulSet with 3 replicas of an nginx container
  • Persistent storage using volumeClaimTemplates (1Gi per pod)
  • A headless service for stable network identities

2. ConfigMap (configmap.yaml):
  • Configuration data for applications
  • Examples of both key-value pairs and configuration files
  • Includes sample nginx configuration

3. Secret (secret.yaml):
  • Contains base64-encoded sensitive information
  • Includes an example for Docker registry authentication
  • Demonstrates different types of secrets (Opaque and dockerconfigjson)

4. README.md:
  • Documentation explaining each file
  • Usage instructions
  • Important notes about security and implementation

These files provide a good starting point that you can customize for your specific application needs. The StatefulSet is configured with persistent storage, databases. The ConfigMap contains application configuration, and the Secrets store sensitive information securely.

To use these files, you can apply them to your Kubernetes cluster with:

bash
kubectl apply -f k8s-manifests/
```

This template of manifest file is generated by amazon Q

```

apiVersion: apps/v1
kind: StatefulSet
metadata:
  name: web
  namespace: default
spec:
  selector:
    matchLabels:
      app: nginx # has to match .spec.template.metadata.labels
  serviceName: "nginx"
  replicas: 3 # by default is 1
  minReadySeconds: 10 # by default is 0
  template:
    metadata:
      labels:
        app: nginx # has to match .spec.selector.matchLabels
    spec:
      terminationGracePeriodSeconds: 10
      containers:
        - name: nginx
          image: nginx:1.25.1
          ports:
            - containerPort: 80
              name: web
          volumeMounts:
            - name: www
              mountPath: /usr/share/nginx/html
  volumeClaimTemplates:
    - metadata:
        name: www
      spec:
        accessModes: [ "ReadWriteOnce" ]
        storageClassName: "standard"
        resources:
          requests:
            storage: 1Gi
---
# Headless service for StatefulSet
apiVersion: v1
kind: Service
metadata:
  name: nginx
  namespace: default
  labels:
    app: nginx
spec:
  ports:
    - port: 80
      name: web
  clusterIP: None
  selector:
    app: nginx
"statefulset.yaml" 52L, 1060B

```

And this is customized by me according to my requirements.

StatefulSets

are used to manage stateful applications they ensure that each pod gets unique identity (so that the pods are maintained in sequence), persistent storage and the pods are created and terminated in sequence.

- Stateless: they don't maintain a state
- For states less we use deployments, daemonsets, replicaset
- For stateful we use statefulsets
- If a pod is terminated it will automatically transfer the state of that pod to new pod
- The pods are numbered in sequence

```

apiVersion: apps/v1
kind: StatefulSet
metadata:
  name: mysql-statefulset
  namespace: mysql
spec:
  replicas: 3
  selector:
    matchLabels:
      app: mysql
  template:
    metadata:
      labels:
        app: mysql
    spec:
      containers:
        - name: mysql
          image: mysql:8.0
          ports:
            - containerPort: 3306
          env:
            - name: MYSQL_ROOT_PASSWORD
              value: root

            - name: MYSQL_DATABASE
              value: devops
          volumeMounts:
            - name: mysql-data
              mountPath: /var/lib/mysql
      volumeClaimTemplates:
        - metadata:
            name: mysql-data
          spec:
            accessModes: [ "ReadWriteOnce" ]
            resources:
              requests:
                storage: 1Gi
---
# Headless service for StatefulSet
apiVersion: v1
kind: Service
metadata:
  name: mysql-service
  namespace: mysql
  labels:
    app: mysql
spec:
  clusterIP: None
  selector:
    app: mysql
  ports:
    name: mysql
    - port: 3306
      targetPort: 3306

```

And pushed the file to GitHub Link:

🌐 [GitHub - pooja-bhavani/Kubernetes-Manifest-Files](#): This repository is all about k8s manif...

```

ubuntu@ip-172-31-2-131:~/.aws/amazonq/k8s-manifests$ cp statefulset.yaml Kubernetes-Manifest-Files
ubuntu@ip-172-31-2-131:~/.aws/amazonq/k8s-manifests$ cd Kubernetes-Manifest-Files
ubuntu@ip-172-31-2-131:~/.aws/amazonq/k8s-manifests/Kubernetes-Manifest-Files$ ls
README.md      cron-job.yaml  deployment.yaml  job.yaml        persistentVolume.yaml  pod.yaml        services.yaml
annotations.yaml  daemonsets.yaml  ingress.yaml    namespace.yaml  persistentVolumeClaim.yaml  replicaset.yaml  statefulset.yaml
ubuntu@ip-172-31-2-131:~/.aws/amazonq/k8s-manifests/Kubernetes-Manifest-Files$ git add .
ubuntu@ip-172-31-2-131:~/.aws/amazonq/k8s-manifests/Kubernetes-Manifest-Files$ git commit -m "added statefulsets file"
[master 5bde9d7] added statefulsets file
Committer: Ubuntu <ubuntu@ip-172-31-2-131.ap-south-1.compute.internal>
Your name and email address were configured automatically based
on your username and hostname. Please check that they are accurate.
You can suppress this message by setting them explicitly. Run the
following command and follow the instructions in your editor to edit
your configuration file:

    git config --global --edit

After doing this, you may fix the identity used for this commit with:

    git commit --amend --reset-author

1 file changed, 55 insertions(+)
create mode 100644 statefulset.yaml
ubuntu@ip-172-31-2-131:~/.aws/amazonq/k8s-manifests/Kubernetes-Manifest-Files$ git push origin master
Username for 'https://github.com': pooja-bhavani
Password for 'https://pooja-bhavani@github.com':
Enumerating objects: 4, done.
Counting objects: 100% (4/4), done.
Compressing objects: 100% (3/3), done.
Writing objects: 100% (3/3), 716 bytes | 716.00 KiB/s, done.
Total 3 (delta 1), reused 0 (delta 0), pack-reused 0
remote: Resolving deltas: 100% (1/1), completed with 1 local object.
To https://github.com/pooja-bhavani/Kubernetes-Manifest-Files.git
7eef921..5bde9d7 master -> master
ubuntu@ip-172-31-2-131:~/.aws/amazonq/k8s-manifests/Kubernetes-Manifest-Files$ q

```

GitHub repository: pooja-bhavani / Kubernetes-Manifest-Files

Navigation: <> Code Issues Pull requests Actions Projects Wiki Security Insights Settings

Files: master + Q

Go to file t

- README.md
- annotations.yaml
- cron-job.yaml
- daemonsets.yaml
- deployment.yaml
- ingress.yaml
- job.yaml
- namespace.yaml
- persistentVolume.yaml
- persistentVolumeClaim.yaml
- pod.yaml
- replicaset.yaml
- services.yaml
- statefulset.yaml**

Kubernetes-Manifest-Files / statefulset.yaml

Ubuntu added statefulsets file 5bde9d7 · 2 minutes ago History

Code Blame 55 Lines (53 loc) · 975 Bytes

```

1  apiVersion: apps/v1
2  kind: StatefulSet
3  metadata:
4    name: mysql-statefulset
5    namespace: mysql
6  spec:
7    replicas: 3
8    selector:
9      matchLabels:
10       app: mysql
11  template:
12    metadata:
13      labels:
14       app: mysql
15    spec:
16      containers:
17       - name: mysql
18         image: mysql:8.0
19         ports:
20         - containerPort: 3306
21         env:
22         - name: MYSQL_ROOT_PASSWORD
23           value: root
24
25       - name: MYSQL_DATABASE
26         value: devops
27    volumeMounts:
28     - name: mysql-data
29       mountPath: /var/lib/mysql
30  volumeClaimTemplates:
31    -

```

```
> history
```

I'll help you explore the Kubernetes manifest files in your current directory. Let's first check what files are available in your current directory:

```
✂ Using tool: fs_read (trusted)
```

```
⋮  
● Reading directory: /home/ubuntu/.aws/amazonq/k8s-manifests/Kubernetes-Manifest-Files with maximum depth of 0  
⋮  
● Completed in 0.0s
```

I see you're working with a collection of Kubernetes manifest files. Your directory contains various Kubernetes resource definitions including:

- deployment.yml
- namespace.yaml
- job.yml
- statefulset.yaml
- persistentVolumeClaim.yml
- pod.yml
- ingress.yml
- services.yml
- persistentVolume.yml
- annotations.yml
- cron-job.yml
- daemonsets.yml
- replicaset.yml

There's also a README.md file that contains instructions for setting up a KIND (Kubernetes IN Docker) cluster with a 3-node configuration (1 control plane and 2 workers).

Would you like me to:

1. Show you the contents of any specific manifest file?
2. Explain how to set up the KIND cluster based on the README?
3. Provide information about a specific Kubernetes resource type?

```
> █
```