k3s with FogMan and FogLAMP

Problem Statement

Setup FogLAMP Manage and FogLAMP containers in Kubernetes environment (container orchestration tool) and achieve following objectives

1. Persist data even after restart
2. How can we upgrade if a new docker image published/DB containers need an up/down script.
3. Multi foglamp, if we use custom Dockerfile and not pull from registry

Containerization tool selection

Container orchestration solves the problem by automating the scheduling, deployment, scalability, load balancing, availability, and networking of containers. Container orchestration is the automation and management of the lifecycle of containers and services. There are many options available in the market but below are the reasons why we chose Rancher k3s and not others

- Minikube - The main downside of Minikube is that it's only designed for testing. It's not a practical solution for running production-grade clusters.

- K3s - K3s is designed to be a full-fledged, production-ready Kubernetes distribution that is also lightweight. Rancher developed it especially for use cases involving infrastructure like internet of things (IoT) and edge devices.

- MicroK8s - MicroK8s is a little more complicated to use than K3s or Minikube, particularly because it has a modular architecture and only runs a minimal set of services by default. To turn on things like DNS support or a web-based dashboard, you have to launch them explicitly.

So based on the above study we chose to proceed with K3s for our problem statement.

Environment

Three Ubuntu 18.04 instances with 4 GB RAM (t2.medium AWS instances)
YT-kube-fogman-server - 100.25.190.74
YT-kube-foglamp-worker - 34.232.69.227
YT-kube-foglamp-worker-2 - 52.201.228.224

Prerequisites

Add the IP address along with tag in the /etc/hosts file on all the machines (both master and worker nodes)

```
100.25.190.74 k3s-master
34.232.69.227 k3s-worker1
52.201.228.224  k3s-worker2
```

Master Configuration

Install k3s using following command

Command:

```
curl -sfL https://get.k3s.io | sh -
```

Expected Output:

```
ubuntu@ip-10-0-0-48:~$ curl -sfL https://get.k3s.io | sh -
[INFO]  Finding release for channel stable
[INFO]  Using v1.24.6+k3s1 as release
[INFO]  Downloading hash https://github.com/k3s-io/k3s/releases/download/v1.24.6+k3s1/sha256sum-amd64.txt
[INFO]  Downloading binary https://github.com/k3s-io/k3s/releases/download/v1.24.6+k3s1/k3s
[INFO]  Verifying binary download
[INFO]  Installing k3s to /usr/local/bin/k3s
[INFO]  Skipping installation of SELinux RPM
[INFO]  Creating /usr/local/bin/kubectl symlink to k3s
[INFO]  Creating /usr/local/bin/crictl symlink to k3s
[INFO]  Creating /usr/local/bin/ctr symlink to k3s
[INFO]  Creating killall script /usr/local/bin/k3s-killall.sh
[INFO]  Creating uninstall script /usr/local/bin/k3s-uninstall.sh
[INFO]  env: Creating environment file /etc/systemd/system/k3s.service.env
[INFO]  systemd: Creating service file /etc/systemd/system/k3s.service
[INFO]  systemd: Enabling k3s unit
Created symlink /etc/systemd/system/multi-user.target.wants/k3s.service → /etc/systemd/system/k3s.service.
[INFO]  systemd: Starting k3s
```

Verify Installation:

```
ubuntu@ip-10-0-0-48:~$ systemctl status k3s
● k3s.service - Lightweight Kubernetes
     Loaded: loaded (/etc/systemd/system/k3s.service; enabled; vendor preset: enabled)
     Active: active (running) since Tue 2022-10-04 14:31:41 UTC; 10min ago
       Docs: https://k3s.io
    Process: 58240 ExecStartPre=/bin/sh -xc ! /usr/bin/systemctl is-enabled --quiet nm-cloud-setup.service (code=exited, status=0/SUCCESS)
    Process: 58242 ExecStartPre=/sbin/modprobe br_netfilter (code=exited, status=0/SUCCESS)
    Process: 58259 ExecStartPre=/sbin/modprobe overlay (code=exited, status=0/SUCCESS)
   Main PID: 58261 (k3s-server)
      Tasks: 102
```

Worker Configuration

Get k3s token from Master node using following command

```
ubuntu@ip-10-0-0-48:~$ sudo cat /var/lib/rancher/k3s/server/node-token
K10a6a44eddb2e32b8616c160084a1ad110b247f99153f17875c74677dbd33ff6be::server:2b4ccfdce9efaff2d509cd0f27fcb
4d0
```

Set following variables on worker node

```
ubuntu@ip-10-0-0-253:~$ k3s_url="https://k3s-master:6443"
ubuntu@ip-10-0-0-253:~$
k3s_token="K10a6a44eddb2e32b8616c160084a1ad110b247f99153f17875c74677dbd33ff6be::server:2b4ccfdce9efaff2d
509cd0f27fcb4d0"
```

Install k3s-agent on worker node

```
ubuntu@ip-10-0-0-253:~$ curl -sfL https://get.k3s.io | K3S_URL=${k3s_url} K3S_TOKEN=${k3s_token} sh -
[INFO]  Finding release for channel stable
[INFO]  Using v1.24.6+k3s1 as release
[INFO]  Downloading hash https://github.com/k3s-io/k3s/releases/download/v1.24.6+k3s1/sha256sum-amd64.txt
[INFO]  Downloading binary https://github.com/k3s-io/k3s/releases/download/v1.24.6+k3s1/k3s
[INFO]  Verifying binary download
[INFO]  Installing k3s to /usr/local/bin/k3s
[INFO]  Skipping installation of SELinux RPM
[INFO]  Creating /usr/local/bin/kubectl symlink to k3s
[INFO]  Creating /usr/local/bin/crictl symlink to k3s
[INFO]  Creating /usr/local/bin/ctr symlink to k3s
[INFO]  Creating killall script /usr/local/bin/k3s-killall.sh
[INFO]  Creating uninstall script /usr/local/bin/k3s-agent-uninstall.sh
[INFO]  env: Creating environment file /etc/systemd/system/k3s-agent.service.env
[INFO]  systemd: Creating service file /etc/systemd/system/k3s-agent.service
[INFO]  systemd: Enabling k3s-agent unit
Created symlink /etc/systemd/system/multi-user.target.wants/k3s-agent.service →
/etc/systemd/system/k3s-agent.service.
[INFO]  systemd: Starting k3s-agent
```

Verify the cluster

```
ubuntu@ip-10-0-0-48:~$ sudo kubectl cluster-info
Kubernetes control plane is running at https://127.0.0.1:6443
CoreDNS is running at https://127.0.0.1:6443/api/v1/namespaces/kube-system/services/kube-dns:dns/proxy
Metrics-server is running at
https://127.0.0.1:6443/api/v1/namespaces/kube-system/services/https:metrics-server:https/proxy

To further debug and diagnose cluster problems, use 'kubectl cluster-info dump'.
```

```
ubuntu@ip-10-0-0-48:~$ sudo kubectl get nodes
```

```
NAME           STATUS   ROLES                 AGE     VERSION
ip-10-0-0-48    Ready    control-plane,master   105m    v1.24.6+k3s1
ip-10-0-0-253   Ready    <none>                2m29s   v1.24.6+k3s1
```

Pull images from insecure registry

Add following lines in all the nodes (master + worker) in `/etc/rancher/k3s/registries.yaml`

```
mirrors:
    "54.204.128.201:5000":
        endpoint:
            - "http://54.204.128.201:5000"
```

After adding restart k3s service
Master: *sudo systemctl restart k3s*
Worker: *sudo systemctl restart k3s-agent*

Reference: https://docs.k3s.io/installation/private-registry#without-tls

Deployment

Sample yaml file for foglamp which pulls image from registry

```
ubuntu@ip-10-0-0-48:~$ cat foglamp.yaml
apiVersion: v1
kind: Pod
metadata:
    name: foglamp-pod
    labels:
        env: prod
spec:
    containers:
        - name: foglamp
          image: 54.204.128.201:5000/foglamp:nightly-ubuntu1804
          ports:
          - containerPort: 80
ubuntu@ip-10-0-0-48:~$
```

Below command will create resources mentioned in yaml file

```
sudo kubectl apply -f foglamp.yaml
```

Expose ports

If the manifest file does not expose the ports we can do it manually by following command

```
sudo kubectl expose pod foglamp-pod --port=80 --type=LoadBalancer
```

```
ubuntu@ip-10-0-0-48:~$ sudo kubectl get services
NAME         TYPE           CLUSTER-IP     EXTERNAL-IP   PORT(S)        AGE
kubernetes   ClusterIP      10.43.0.1      <none>        443/TCP        12d
foglamp-pod  LoadBalancer   10.43.52.111   <pending>     80:30186/TCP   2d22h
```

With the above config foglamp is available on port 30186.

Debugging commands

```
sudo kubectl describe pods fogman-pod
sudo kubectl logs fogman-pod -c fogman
sudo crictl image ls
sudo kubectl delete pod fogman-pod
sudo kubectl get service
sudo kubectl get nodes -o wide
sudo kubectl get pods -o wide
sudo kubectl apply -f fogman.yaml --dry-run=client
```