# MicroK8s with Ubuntu on LinuxONE and IBM Z

## Intro

MicroK8s [1] is the smallest, fastest, fully-conformant Kubernetes that tracks upstream releases and makes clustering trivial. MicroK8s works across developer workstations, edge clusters and IoT devices.

Mission critical applications require a resilient Kubernetes, that can lose any node and still provide reliable services while zero downtime is a necessity in production environments. Fault tolerance is especially critical for remote, unattended clusters, appliances and industrial IoT, where access is limited and nodes are distributed across geographies. MicroK8s makes high availability resilient and self-healing with no administrative intervention required.

The IBM Z platform is a perfect fit for such types of Kubernetes infrastructural and aaS workloads - thus Container As A Service / CaaS.
While Charmed Kubernetes [7] is extremely configurable and modular when it comes to setup options as it can be setup using manual provider, local provider (LXD or KVM) or MAAS (KVM or DPM/LPAR) - even across multiple Z systems - the initial setup, ramp-up phase and complexity shouldn't be underestimated.

To reduce complexity, MicroK8s supports IBM Z systems and can bring up a fully-conformant Kubernetes in just a minute, hosted on the most reliable hardware platform available. You can run it on LPAR to get native speed without losing scalability options, since IBM Z allows to scale and adjust LPAR resources on the fly in terms of processors, RAM and disk space.

You can also fully control the resource consumption of your MicroK8s cluster on IBM Z LPAR, by adding more cluster nodes - or just quickly scale up, with a single command. Depending on the size of your IBM Z or LinuxONE machine and the dimensions of your MicroK8s setup, you may run dozens or even hundreds of MicroK8s instances on a single hardware box.

Let's now see how to set MicroK8s on IBM Z.

# What you need

- An Ubuntu 20.04 LTS environment (LPAR, z/VM guest or LXD system container) to run the commands. Please refer to the “Installer” section of the Ubuntu Server Guide [2] to get Ubuntu installed on an LPAR or z/VM.
- At least 20G of disk space and 8G of memory are recommended
- At least one qeth network interface, where two are recommended for MicroK8s clustering purposes
- Internet connectivity (for the Ubuntu archives and the Snap store)

# Installation

MicroK8s can run either standalone (inside a single Ubuntu host only) or as a clustered (HA) configuration across several Ubuntu hosts.

## MicroK8s Standalone

Follow these installation steps to get a standalone Kubernetes environment:

### 1) Run the commands below:

```
$ sudo snap install --channel latest/edge microk8s --classic
```

```
$ sudo snap install kubectl --classic
```

2) Once installed, update the `/var/snap/microk8s/current/args/containerd-env` file as described in the MicroK8s website section [3] if you are working behind a proxy

3) Enable the DNS as explained in [4]

```
$ microk8s enable dns
```

Now you have a working Kubernetes, 100% conformant, standalone installation.

You can check for it and whether it properly works. Follow the steps as in the "Verification of MicroK8s" section at the bottom of this document.

## MicroK8s Clustered/HA

A highly available Kubernetes cluster is a cluster that can withstand a failure on any one of its components and continue serving workloads without interruption. There are three components necessary for a highly available Kubernetes cluster:

1. There must be more than one node available at any time.
2. The control plane must be running on more than one node so that losing a single node would not render the cluster in-operable.
3. The cluster state must be in a datastore that is itself highly available.

You need at least 3 Ubuntu LPARs or Ubuntu z/VM guests running on the same Z system to form a cluster. Note that, as with almost all networked services, it is also important that these Ubuntu instances have the correct time (e.g. updated from an ntp server) for inter-node communication to work.

Here [6] there are more detailed technical informations about MicroK8s HA

- Install a standalone MicroK8s on each Ubuntu LPAR/VM you plan to use for the Kubernetes cluster, by running steps 1-3 in the "MicroK8s Standalone" section above
- Pick up one of the just installed hosts and on this initial host, run:

  ```
  $ microk8s add-node
  ```

- This will output a command with a generated token such as:

  ```
  Join node with:
  microk8s join
  10.128.63.86:25000/567a21bdfc9a64738ef4b3286b2b8a69
  ```

  Copy this command and run it from the next node. It may take a few minutes to successfully join.

- Repeat this process (generate a token, run it from the joining node) for the third and any additional nodes.
- Check the status of the cluster by running the command below on the initial node:

```
$ microk8s status
```

- This will now inform you of the HA status and the addresses and roles of additional nodes. For example:

```
microk8s is running
high-availability: yes
     datastore master nodes: 10.128.63.86:19001
     10.128.63.166:19001 10.128.63.43:19001
     datastore standby nodes: none
```

You can check for it and whether it properly works. Follow the steps as in the "Verification of MicroK8s" section below.

## Verification of the MicroK8s installation

Check for it:

```
$ microk8s kubectl get all -A
```

Or using the standard kubectl tool installed previously:

```
$ microk8s config > .kube/config

$ kubectl get all -A
```

Let's onboard a first service, the microbot application, following [5] but customizing it a little bit:

```
$ kubectl create deployment microbot
--image=cdkbot/microbot-s390x

$ kubectl scale deployment microbot --replicas=10
```

```
$ kubectl expose deployment microbot --type=NodePort
--port=80 --name=microbot-service
```

Run the following command and look at the "microbot-service" NodePort and at the port number:

```
$ kubectl get svc
```

To check whether the application is properly working run:

```
$ curl http://localhost:port-number
```

(usually it's a port like 3XYZK i.e. 31211, 32275, etc. )

These are the essential steps you need to know to get your first MicroK8s environment up and running on IBM Z - just give it a try and let us know your feedback [8]!

[1] - https://microk8s.io/
[2] - https://ubuntu.com/server/docs/install/general
[3] - https://microk8s.io/docs/install-proxy
[4] - https://microk8s.io/docs/addon-dns
[5] - https://ubuntu.com/tutorials/install-a-local-kubernetes-with-microk8s#5-host-your-first-service-in-kubernetes
[6] - https://microk8s.io/docs/high-availability
[7] - https://ubuntu-on-big-iron.blogspot.com/2019/08/deploy-cdk-on-ubuntu-s390x.html
[8] - https://github.com/ubuntu/microk8s/issues