Microk8s:

Microk8s is a fast and lightweight upstream Kubernetes install isolated from your host but not via a virtual machine. This isolation is achieved by packaging all the upstream binaries for Kubernetes, Docker.io, iptables, and CNI in a single snap package.

The snap package is an application container — you can imagine this as a lighter weight version of a Docker container. It uses a lot of the same underlying technologies for isolation just without all the overhead of network isolation.

Install microk8s:

```
> sudo snap install microk8s --classic
```

At any point you can check MicroK8s' availability with:

```
microk8s.status
```

During installation you can use the --wait-ready flag to wait for the kubernetes services to initialise:

```
microk8s.status --wait-ready
```

Accessing Kubernates:

```
microk8s.kubectl get nodes
microk8s.kubectl get services
```

If you do not already have a version of kubectl installed you can alias microk8s.kubectl to kubectl using the following command:

```
snap alias microk8s.kubectl kubectl
```

This measure can be safely reverted at any time by running:

```
snap unalias kubectl
```

Istio:

Istio is an open source service mesh platform. Istio offers many features like:

- Traffic Management
- Security Policies
- Telemetry
- Performance Tuning

Istio deployment available with:

```
> microk8s.enable istio
```

There is a single question that we need to respond to at this point. Do we want to enforce mutual TLS authentication among sidecars? Istio places a proxy to your services so as to take control over routing, security etc. If we know we have a mixed deployment with non-Istio and Istio enabled services we would rather not enforce mutual TLS:

```
> microk8s.enable istio
Enabling Istio
Enabling DNS
Applying manifest
service/kube-dns created
serviceaccount/kube-dns created
configmap/kube-dns created
deployment.extensions/kube-dns created
Restarting kubelet
DNS is enabled
Enforce mutual TLS authentication (https://bit.ly/2KB4j04)
between sidecars? If unsure, choose N. (y/N): y
```

Believe it or not we are done, Istio v1.0 services are being set up, you can check the deployment progress with:

```
> watch microk8s.kubectl get all --all-namespaces
```

Istioctl is packaged in microk8s:

```
> microk8s.istioctl get all --all-namespaces
NAME           KIND                NAMESPACE      AGE
grafana-ports-mtls-disabled
Policy.authentication.istio.io.v1alpha1   istio-system   2m

DESTINATION-RULE   NAME     HOST  SUBSETS   NAMESPACE    AGE
istio-policy            istio-policy.istio-
system.svc.cluster.local                istio-system   3m
istio-telemetry         istio-telemetry.istio-
system.svc.cluster.local             istio-system   3m

GATEWAY NAME                        HOSTS     NAMESPACE      AGE
istio-autogenerated-k8s-ingress   *         istio-system   3m
```

Stopping and restarting MicroK8s:

At anytime you can pause all the Kubernetes services running by issuing:

```
snap disable microk8s
```

This will not only disable all the running services, but remove the microk8s command. It's effectively the same as uninstalling without the file removal. When you're ready to start again just enable microk8s

```
snap enable microk8s
```

It'll take a few moments for pods to re-settle again. It's worth noting that disabling microk8s will also stop all running containers.

OR

MicroK8s can also be shutdown with:

```
microk8s.stop
```

MicroK8s can be restarted later with:

```
microk8s.start
```

References:

https://microk8s.io/docs/

https://istio.io/docs/concepts/security/#mutual-tls-authentication

https://itnext.io/a-local-kubernetes-with-microk8s-33ee31d1eed9