

The background is a dark blue gradient. On the left, there is a large, semi-transparent circular inset showing a detailed view of a microchip or circuit board. Overlaid on the top left of this circle are two overlapping triangles: a blue one in front and a light green one behind it. In the top right corner, there is a faint, stylized pattern of white lines and squares, resembling a circuit board or a data grid.

# Kubernetes Installation Micro K8s



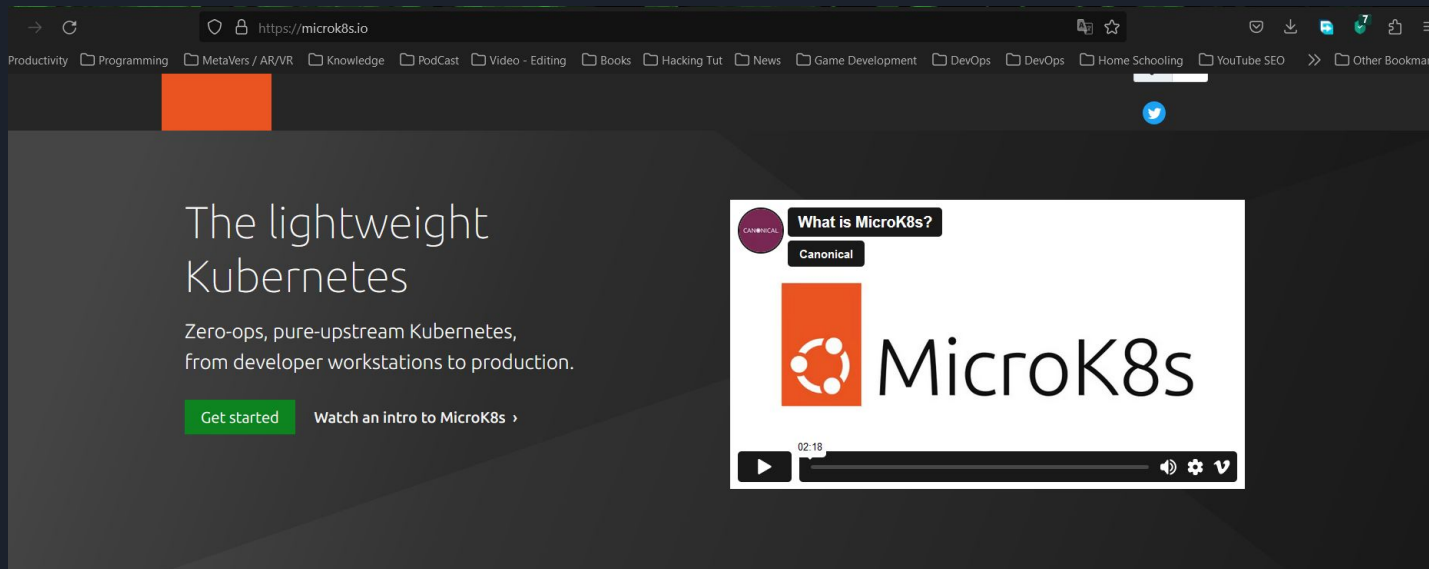
# Installation

Kubernetes can be install in three ways

- **MicroK8s** is a lightweight and easy-to-use Kubernetes distribution developed by Canonical, the company behind Ubuntu. It is designed to run on desktops, servers, and IoT devices, and provides a complete Kubernetes environment on a single node.
- **Minikube** is a tool that allows users to run a single-node Kubernetes cluster on their local machine. It is designed for developers and individuals who want to test and develop Kubernetes applications locally before deploying them to a production Kubernetes cluster
- **Kubeadm** is a tool for bootstrapping a Kubernetes cluster. It is designed for advanced users and administrators who want to set up a custom Kubernetes cluster from scratch. Kubeadm provides a set of commands for initializing and joining nodes to a Kubernetes cluster, and can be used to set up highly available and secure Kubernetes clusters.

# Micro K8s

<https://microk8s.io/>



The screenshot shows the MicroK8s website in a web browser. The browser's address bar displays <https://microk8s.io/>. The website's navigation bar includes links to various topics: Productivity, Programming, MetaVers / AR/VR, Knowledge, PodCast, Video - Editing, Books, Hacking Tut, News, Game Development, DevOps, DevOps, Home Schooling, YouTube SEO, and Other Bookmar. The main content area features the heading "The lightweight Kubernetes" and the subtext "Zero-ops, pure-upstream Kubernetes, from developer workstations to production." Below this, there are two buttons: "Get started" and "Watch an intro to MicroK8s >". On the right side, there is a video player showing a video titled "What is MicroK8s?" by Canonical. The video player includes a play button, a progress bar, and a timestamp of 02:18.

→ ↻ <https://microk8s.io/> 📄 ☆


Productivity Programming MetaVers / AR/VR Knowledge PodCast Video - Editing Books Hacking Tut News Game Development DevOps DevOps Home Schooling YouTube SEO >> Other Bookmar

**The lightweight  
Kubernetes**

Zero-ops, pure-upstream Kubernetes,  
from developer workstations to production.

[Get started](#) [Watch an intro to MicroK8s >](#)

**What is MicroK8s?**  
Canonical

 **MicroK8s**

02:18

→ Install Micro K8s on Linux

```
bilal@bilal-virtual-machine: ~  
bilal@bilal-virtual-machine:~$ sudo snap install microk8s --classic  
[sudo] password for bilal:  
Download snap "microk8s" (4595) from channel "1.26/stable"      0%      0B/s ages!
```

→ Check the status while Kubernetes starts

```
bilal@bilal-virtual-machine:~$ microk8s status --wait-ready  
microk8s is running  
high-availability: no  
  datastore master nodes: 127.0.0.1:19001  
  datastore standby nodes: none  
addons:  
  enabled:  
    ha-cluster      # (core) Configure high availability on the current node  
    helm            # (core) Helm - the package manager for Kubernetes  
    helm3           # (core) Helm 3 - the package manager for Kubernetes  
  disabled:  
    cert-manager    # (core) Cloud native certificate management  
    community        # (core) The community addons repository  
    dashboard        # (core) The Kubernetes dashboard  
    dns              # (core) CoreDNS
```



→ Turn on the services you want

```
registry # (core) Private image registry exposed on localhost:32000
storage # (core) Alias to hostpath-storage add-on, deprecated
root@bilal-virtual-machine:~# microk8s enable dashboard dns registry istio
Infer repository core for addon dashboard
Enabling Kubernetes Dashboard
Infer repository core for addon metrics-server
Enabling Metrics-Server
serviceaccount/metrics-server created
clusterrole.rbac.authorization.k8s.io/system:aggregated-metrics-reader created
clusterrole.rbac.authorization.k8s.io/system:metrics-server created
rolebinding.rbac.authorization.k8s.io/metrics-server-auth-reader created
clusterrolebinding.rbac.authorization.k8s.io/metrics-server:system:auth-delegator created
```

## → Start using Kubernetes

```
microk8s dashboard-proxy
Checking if Dashboard is running.
Infer repository core for addon dashboard
Waiting for Dashboard to come up.
Trying to get token from microk8s-dashboard-token
Waiting for secret token (attempt 0)
Dashboard will be available at https://127.0.0.1:10443
Use the following token to login:
eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXLTZCIiwiaWF0IjoiMTUzMTIzWjU5FTQtaWZlBXVhWPXU0wyt
```

## The screenshot shows a web browser window at the URL https://192.168.60.146:10443/#/login. The page title is "Kubernetes Dashboard". There are two main options for authentication: "Token" (selected) and "Kubeconfig". The "Token" section explains that every Service Account has a Secret with a valid Bearer Token and provides a link to the "Authentication" section. The "Kubeconfig" section asks the user to select a kubeconfig file and provides a link to the "Access to Multiple Clusters" section. Below these sections is a form labeled "Enter token \*" and a "Sign in" button. In the foreground, there is a terminal window titled "root@bilal" showing the execution of the "microk8s dashboard-proxy" command. The output indicates that the proxy is running, checks if the dashboard is up, waits for tokens, and finally displays a long Bearer token for login.

