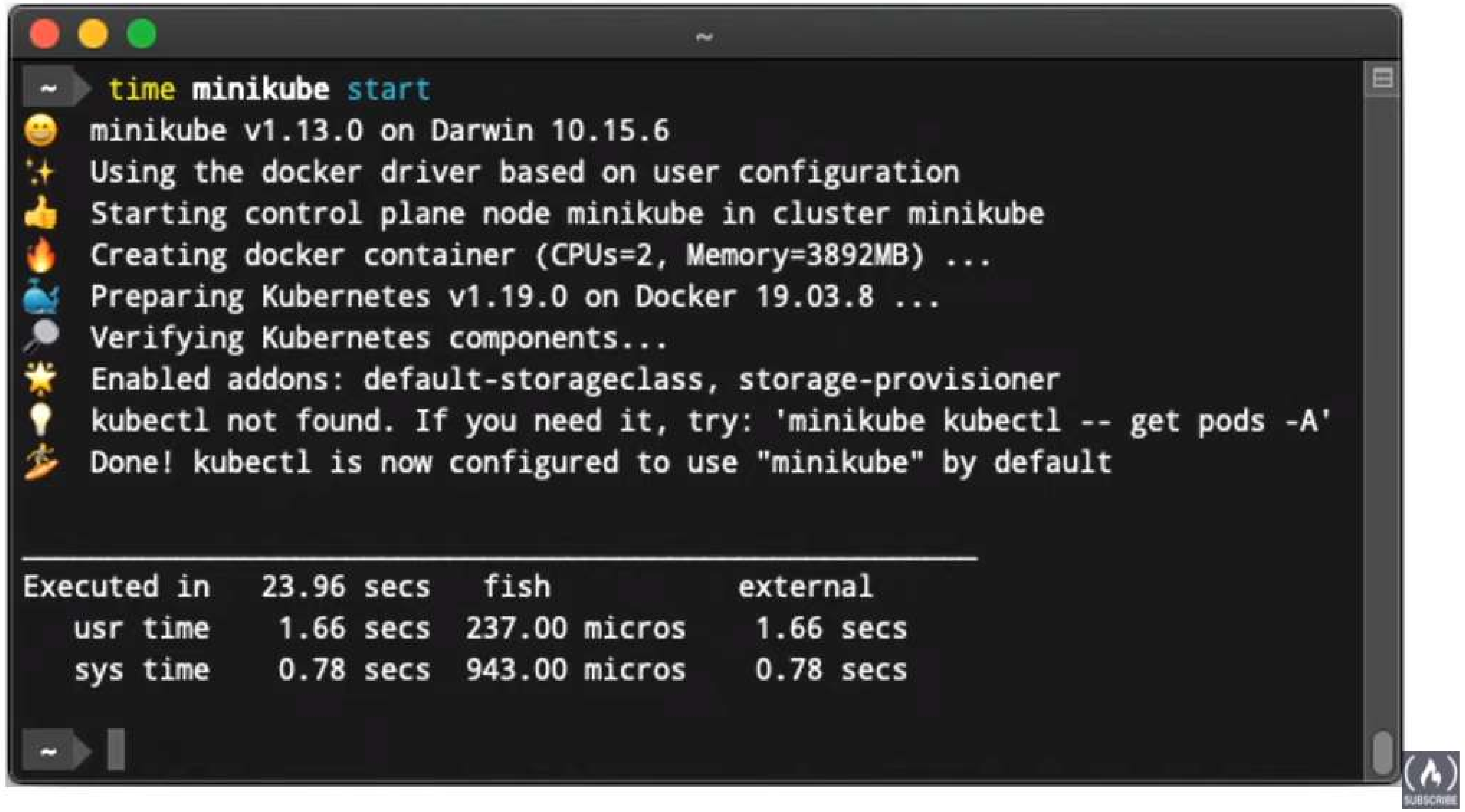
MiniKube

Cheat sheets, Practice Exams and Flash cards 👉 www.exampro.co/kcna

Minikube sets up a local single-node Kubernetes cluster on macOS, Linux, and Windows for learning purposes.

Minikube is running a Virtual Machine running a control-place and work processes with Docker as the container layer

- Supports the latest Kubernetes release
- Cross-platform
- Deploy as a VM, a container, or on bare-metal
- Multiple container runtimes
- Docker API endpoint for blazing fast image pushes
- Advanced features such as LoadBalancer, filesystem mounts, and FeatureGates
- Addons for easily installed Kubernetes applications
- Supports common CI environments



```
~ > time minikube start
😄  minikube v1.13.0 on Darwin 10.15.6
✨  Using the docker driver based on user configuration
👍  Starting control plane node minikube in cluster minikube
🔥  Creating docker container (CPUs=2, Memory=3892MB) ...
🐳  Preparing Kubernetes v1.19.0 on Docker 19.03.8 ...
🔎  Verifying Kubernetes components...
🌟  Enabled addons: default-storageclass, storage-provisioner
💡  kubectl not found. If you need it, try: 'minikube kubectl -- get pods -A'
🏄  Done! kubectl is now configured to use "minikube" by default

________________________________________________________
Executed in   23.96 secs   fish           external
   usr time    1.66 secs  237.00 micros    1.66 secs
   sys time    0.78 secs  943.00 micros    0.78 secs

~ > 
```

K3s and K3d

K3s is a lightweight tool designed to run production-level Kubernetes workloads for **low-resourced and remotely located IoT and Edge devices and Bare metal**.
Originally Created by **Rancher**, a Sandbox CNCF Project

K3s does not use kubelet, but it runs kubelet on the host machine and uses the host's scheduling mechanism to run containers

K3s uses kube-proxy to proxy the network connections of the nodes.
K8s uses kube-proxy to proxy the network connections of an individual container.

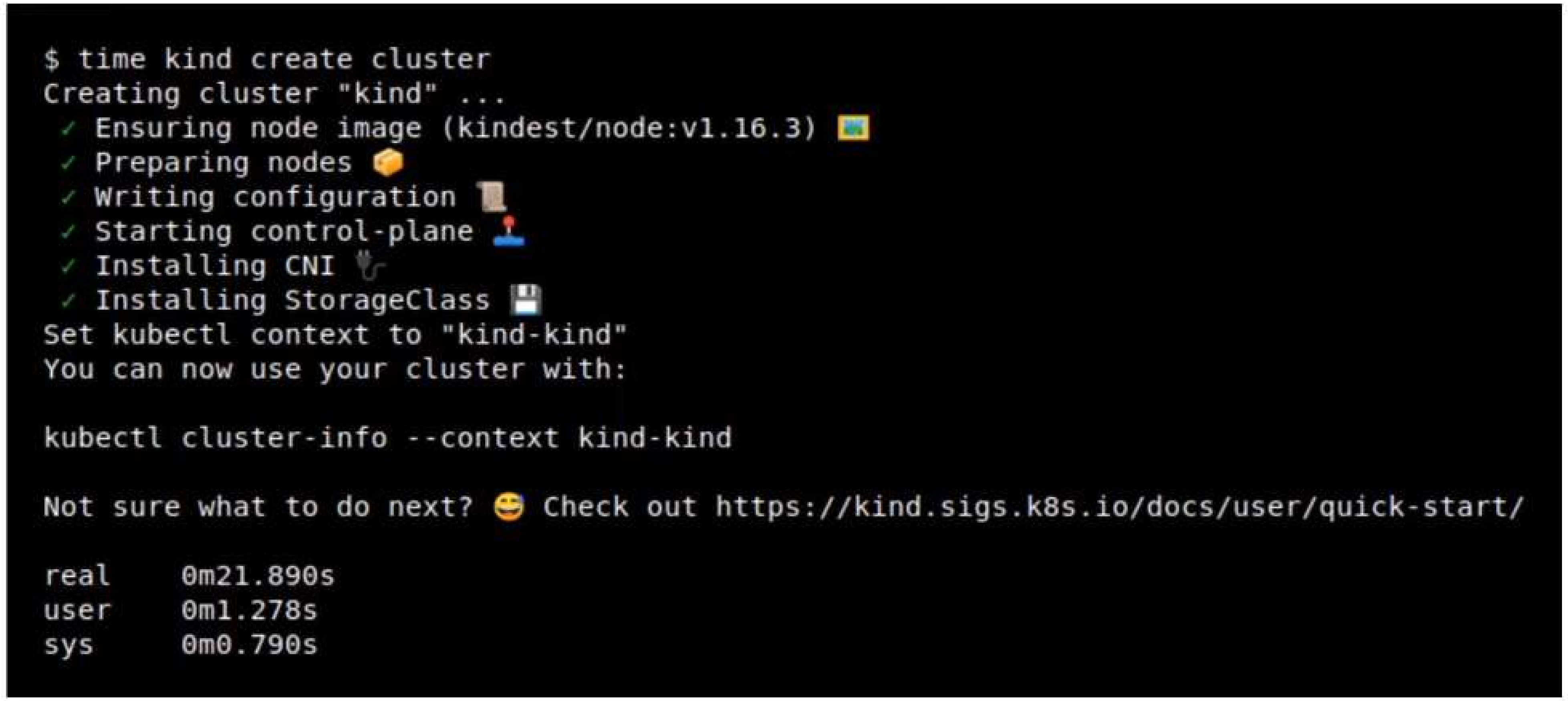
k3s can have tighter security deployment than k8s because of their small attack surface area.

K3s has some advantage but comes with limitations and you'll need to investigate for yourself whether it makes sense to use K3s for your use case

K3d is a platform-agnostic, **lightweight wrapper** that **runs K3s in a docker container.**
It helps run and scale single or multi-node K3S clusters quickly without further setup while maintaining a high availability mode.

Kind

Primarily **designed to test Kubernetes**, Kind (Kubernetes in Docker) helps you run Kubernetes clusters locally and in CI pipelines using Docker containers as “nodes”.



```
$ time kind create cluster
Creating cluster "kind" ...
 ✓ Ensuring node image (kindest/node:v1.16.3) 🖼
 ✓ Preparing nodes 📦
 ✓ Writing configuration 📜
 ✓ Starting control-plane 🕹️
 ✓ Installing CNI 🔌
 ✓ Installing StorageClass 💾
Set kubectl context to "kind-kind"
You can now use your cluster with:

kubectl cluster-info --context kind-kind

Not sure what to do next? 😅 Check out https://kind.sigs.k8s.io/docs/user/quick-start/

real    0m21.890s
user    0m1.278s
sys     0m0.790s
```

It is an open source CNCF certified Kubernetes installer that supports highly available multi-node clusters and builds Kubernetes release builds from its source.

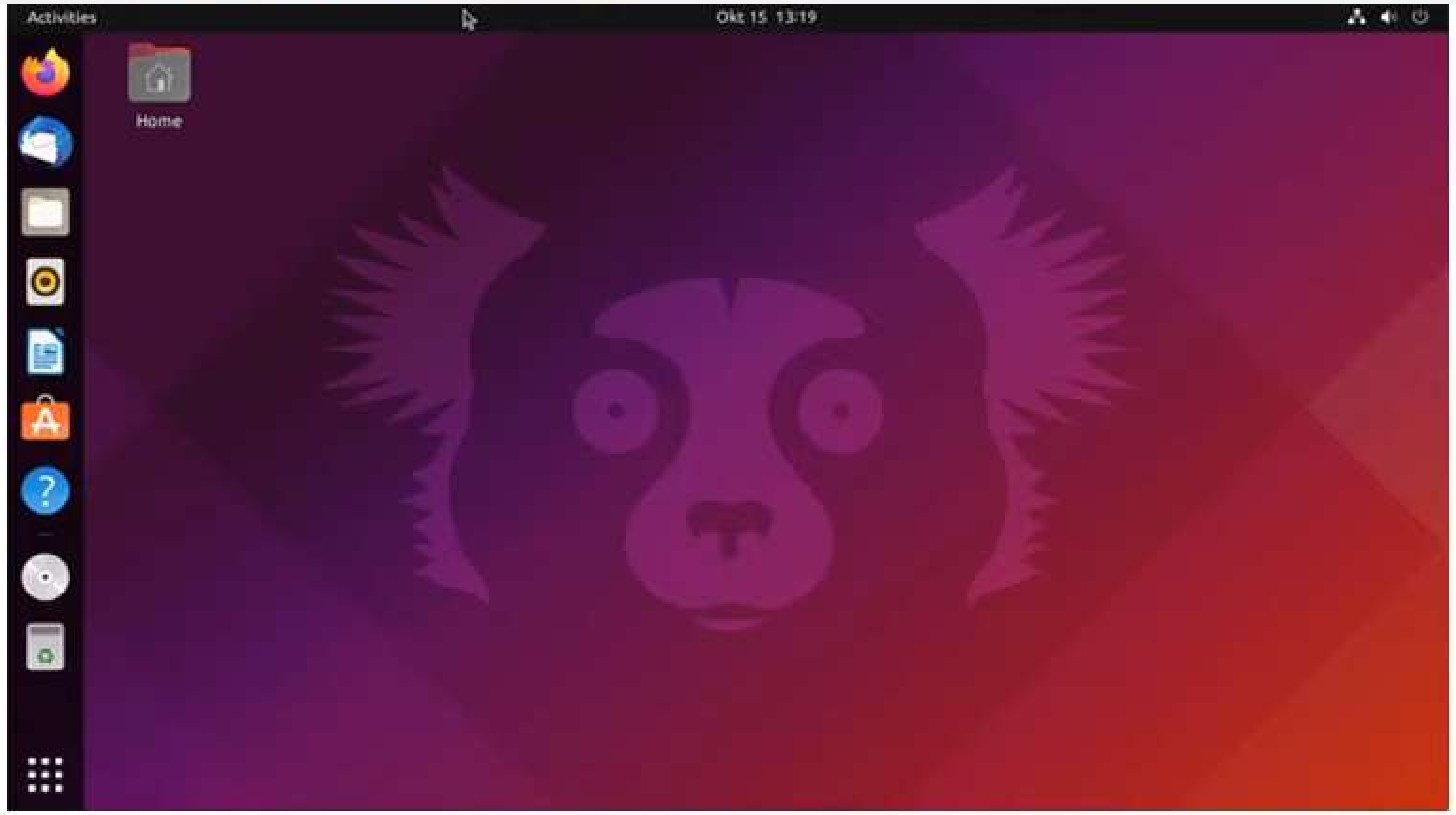
MicroK8S

Cheat sheets, Practice Exams and Flash cards 👉 www.exampro.co/kcna

Ubuntu is a Linux distribution based on Debian.
Ubuntu is known for:

- lots of Linux programs pre-installed
- the of the easiest Linux distributions to use
- More frequent updates
- More progressive on new linux programs and systems

It comes in many editions eg. **Desktop**, Server, Core →



Canonical is the company that publishers of Ubuntu

Snap is a package manager by Canonical that can be installed on many different distributions of Linux.

```
sudo snap install ruby --classic
```

Classic Flag, allows access to your system's resources in much the same way traditional package. Without the flag snaps run in complete isolation.

MicroK8S

Cheat sheets, Practice Exams and Flash cards 👉 www.exampro.co/kcna

MicroK8s is created by Canonical and is installed using **Snap**.

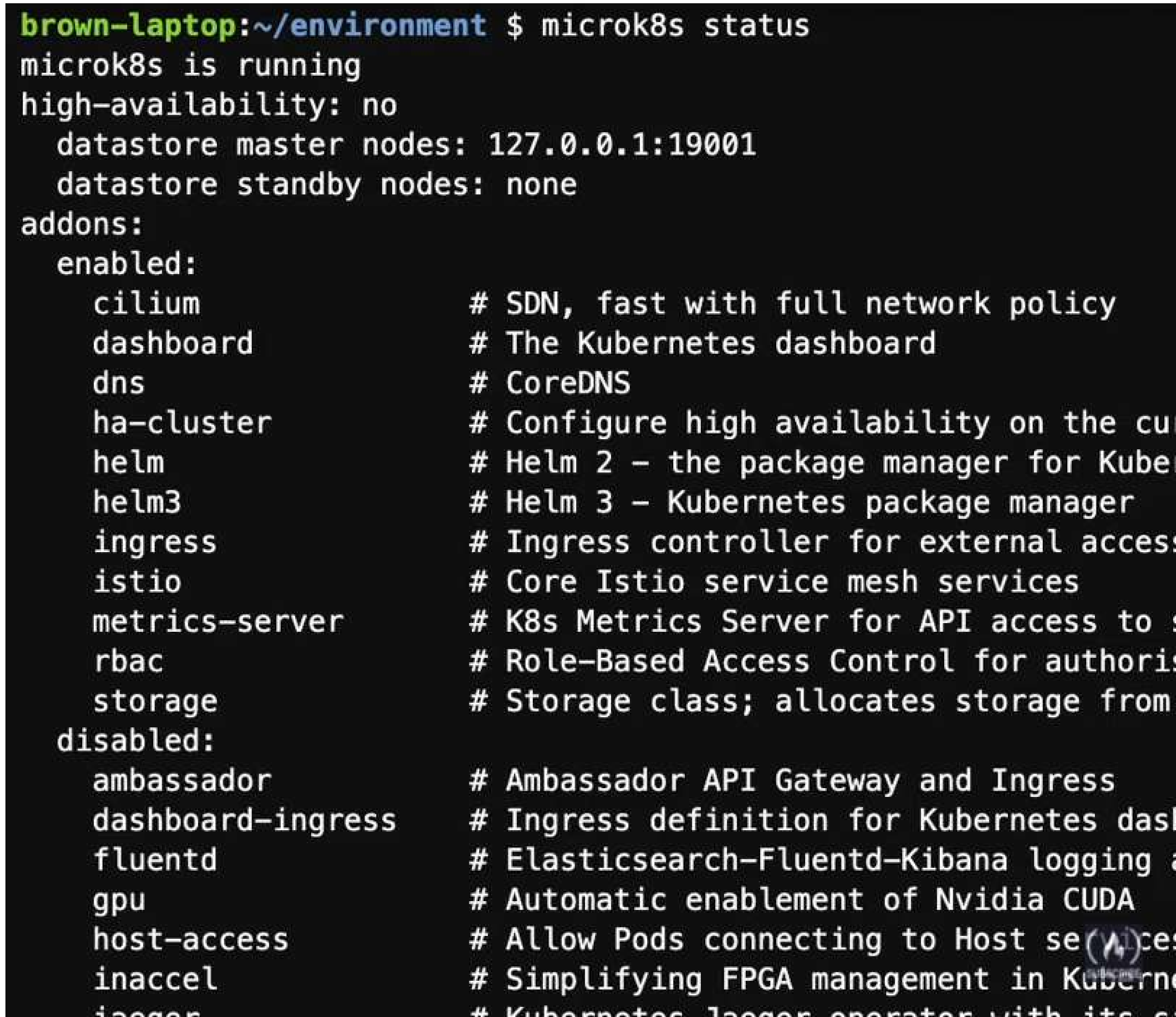
```
sudo snap install microk8s --classic
```

It is a a Kubernetes distribution designed to run fast, self-healing, and highly available Kubernetes clusters.

It is optimized for quick and easy installation of single and multi-node clusters on multiple operating systems, including macOS, Linux, and Windows *(as long as you have snap).*

It is ideal for running Kubernetes in the cloud, local development environments, and Edge and IoT devices

Microk8s is modular in design, you start with nothing and can **enable addons** to quickly use exactly what you need and nothing more:



```
brown-laptop:~/environment $ microk8s status
microk8s is running
high-availability: no
  datastore master nodes: 127.0.0.1:19001
  datastore standby nodes: none
addons:
  enabled:
    cilium               # SDN, fast with full network policy
    dashboard            # The Kubernetes dashboard
    dns                  # CoreDNS
    ha-cluster           # Configure high availability on the cu
    helm                 # Helm 2 - the package manager for Kube
    helm3                # Helm 3 - Kubernetes package manager
    ingress              # Ingress controller for external acces
    istio                # Core Istio service mesh services
    metrics-server       # K8s Metrics Server for API access to s
    rbac                 # Role-Based Access Control for authori
    storage              # Storage class; allocates storage from
  disabled:
    ambassador           # Ambassador API Gateway and Ingress
    dashboard-ingress    # Ingress definition for Kubernetes dash
    fluentd              # Elasticsearch-Fluentd-Kibana logging a
    gpu                  # Automatic enablement of Nvidia CUDA
    host-access          # Allow Pods connecting to Host services
    inaccel              # Simplifying FPGA management in Kuberne
    jaeger               # Kubernetes Jaeger operator with its
```

Lightweight K8s Distribution Comparison

Cheat sheets, Practice Exams and Flash cards 👉 www.exampro.co/kcna

Minikube runs in a Virtual Machine
Intended just for development purposes.
Very easy to use, very popular

Kind is designed to run anywhere container runs.
Intended just for development purposes
Faster startup time than a Minikube since its not spinning up a VM

K3s and K3D is a K8s distribution by rancher
Can be used for production use-cases
Designed for embedded, edge devices, or limited resources.

MicoK8s is created by Canonical, Need Snap to install
Modular, starts with nothing installed.
Restarts everything if there is a crash.
We'll suited for self-hosted production use-cases

Managed Kubernetes Providers

Cheat sheets, Practice Exams and Flash cards 👉 www.exampro.co/kcna

Managed Kubernetes providers are Cloud Service Providers (CSPs) or platforms that **abstracts away** the effort of setting up, maintaining (updating and patching) a cluster. They can easy autoscaling as well.

Google Kubernetes Engine (GKE)
The easiest to use with the richest amount of features

Amazon Elastic Kubernetes Service (EKS)
Difficult to use via the UI, powerful CLI tool
Can be worth it for integrations with other AWS services

Azure Kubernetes Service (AKS)
Fairly easy to use. Unique service offers for debugging live containers.

IBM Cloud Kubernetes Service
Easy to use, not feature rich. More expensive than any other cloud service provider

Oracle Container Engine for Kubernetes
Cost effective for a cloud service provider, worst UI, limited options

DigitalOcean Kubernetes (DOKS)
Very easy to use, predictable spend.
Beautiful UI

CIVO Kubernetes
Most cost effective. Simple UI
A cloud platform specifically focused on just Kubernetes.

Management Layers

Cheat sheets, Practice Exams and Flash cards 👉 www.exampro.co/kcna

A management layer for running Kubernetes on other platforms or allows you to extend your control plane to multiple platform

Weave Kubernetes Platform (WKP)
All of Weaves open-sources tools packaged as a platform so you can build out a GitOps enabled cluster.

Azure Arc *multi-cluster-management*
Governing compute such as K8s from other CSPs or on-premise or the edge.

Rafay
Similar to OpenShift with a larger focus on governance and GitOps-based management for any K8s clusters running on anything (including OpenShift)

Google Anthos *multi-cluster-management*
Is GKE being extended to mange clusters deployed to VMs on other cloud's or on-premise. Its focused on managing K8.

VMWare Tanzu
Wherever vSphere runs, you can mange you can deploy and monitor Kubernetes clusters.

Platform9
Similar to RayFay but relies more on third-party tooling instead of trying to leverage native functionality from public cloud service providers.

Management Layers

Red Hat OpenShift *Platform as a Service for K8s*

- Openshift is Kubernetes with a commercial platform by RedHat built on-top.
- Kubectl is extended with additional functionality with the oc cli
- quickly deploy local code to a remote OpenShift cluster via odo
- A quality assurance pipeline built into the platform
- Fixing critical bugs earlier instead of waiting for next K8s release
- Using Redhat CoreOS (an operating system optimized for running containers)
- OperatorsHub, an automated installation tool (one click marketplace)
- Graphical UI developer console
- CodeReady workspaces, Cloud Developer Environment for Kubernetes

Rancher Kubernetes Engine (RKE)

- Runs entirely within Docker containers.
- It works on bare-metal and virtualized servers.
- RKE solves the problem of installation complexity, a common issue in the Kubernetes community.
- Installation and operation of Kubernetes is both simplified and easily automated,
- It's entirely independent of the operating system and platform you're running.
- As long as you can run a supported version of Docker, you can deploy and run Kubernetes with RKE.