

[Open in app](#)[Sign up](#)[Sign In](#)[Search](#)

# The differences between Docker, containerd, CRI-O and runc

Vineet Kumar · [Follow](#)

6 min read · Jun 25

[Listen](#)[Share](#)

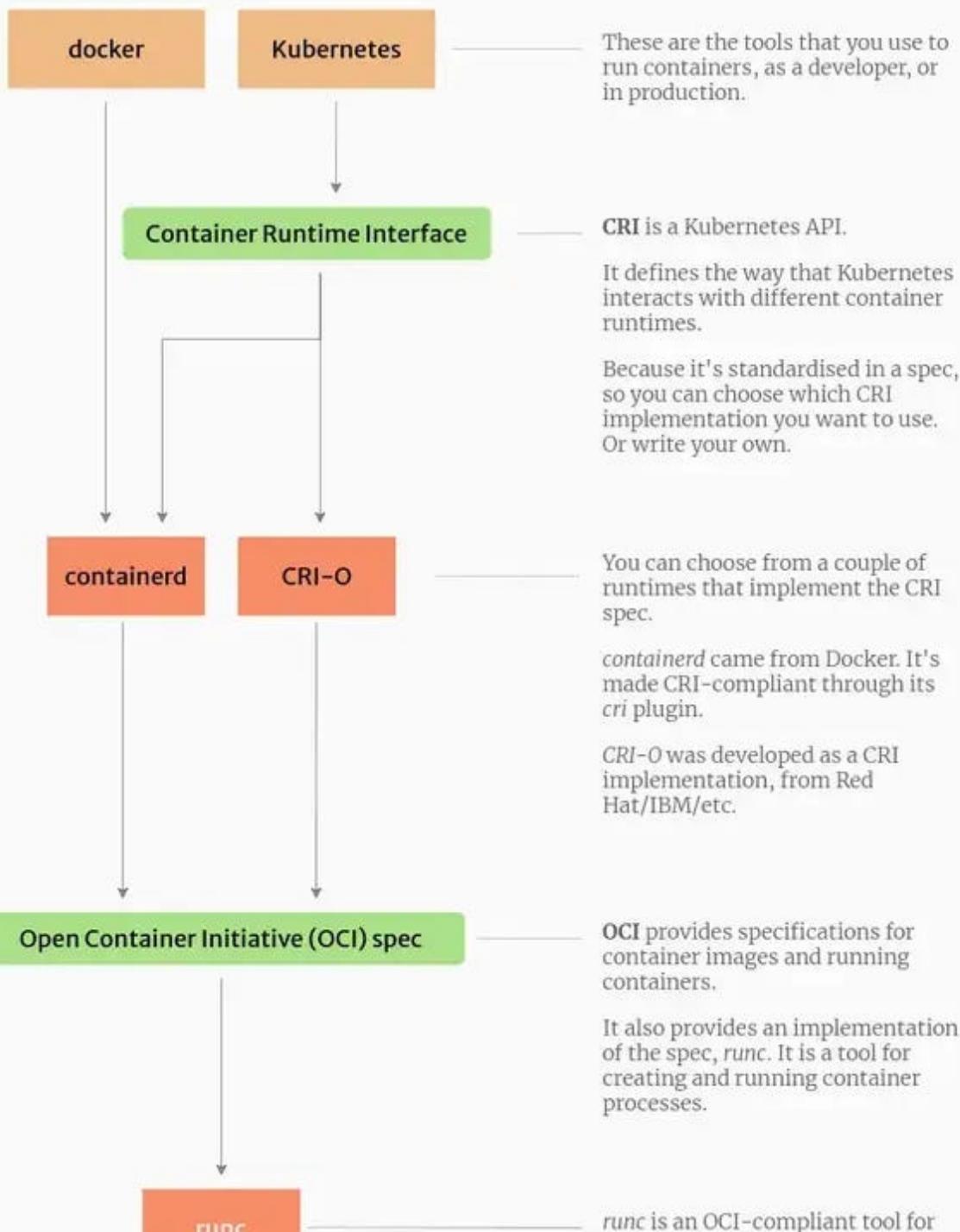
Containers aren't tightly coupled to the name Docker. You can use other tools to run containers

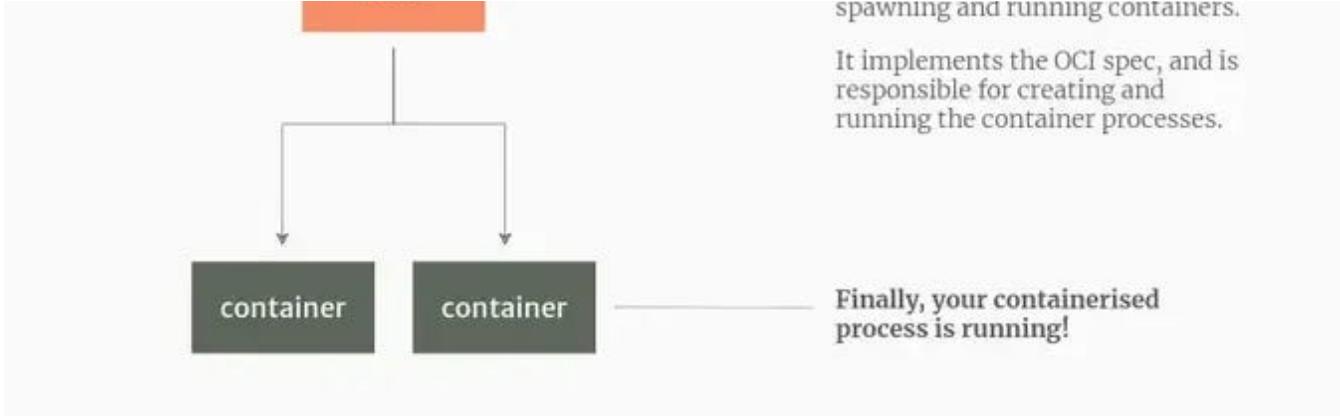
You can be running containers with Docker, or a bunch of other tools which aren't Docker. `docker` is just one of the many options, and Docker (the company) creates some of the awesome [tools](#) in the ecosystem, but not all.

There are two big standards around containers:

- **Open Container Initiative (OCI):** a set of standards for containers, describing the image format, runtime, and distribution.
- **Container Runtime Interface (CRI) in Kubernetes:** An API that allows you to use different container runtimes in Kubernetes.

## Docker, Kubernetes, OCI, CRI-O, containerd & runc: How do they work together?





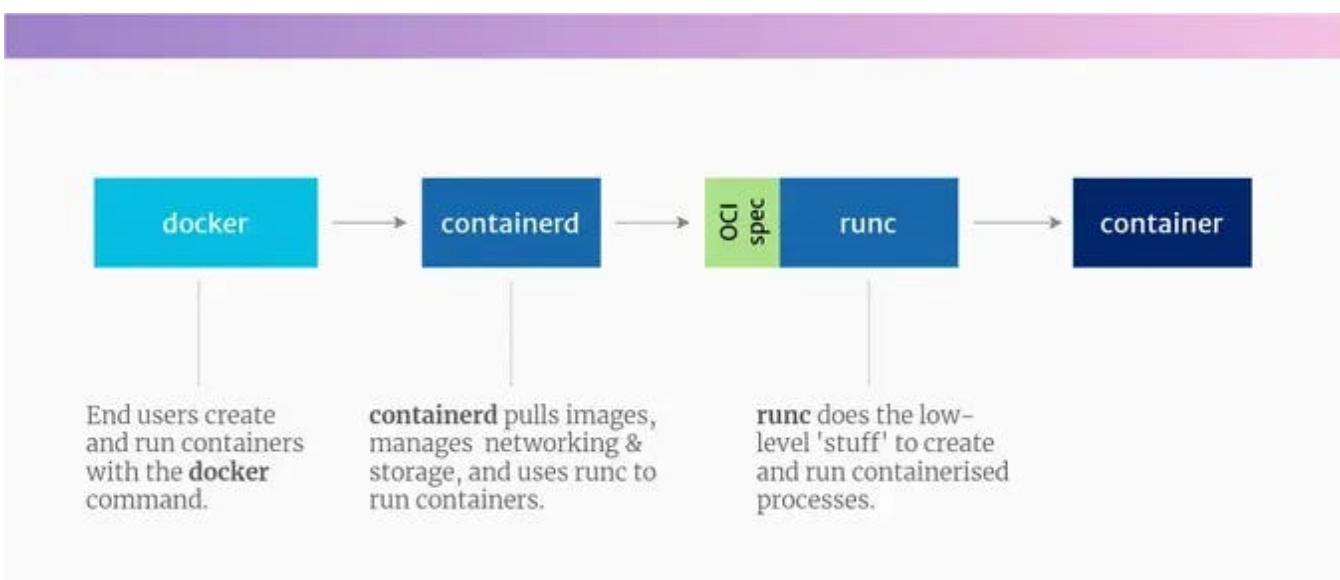
overview of how **Docker**, **Kubernetes**, **CRI**, **OCI**, **containerd** and **runc** fit together in this ecosystem

## How the Docker stack works

Docker Engine comes with a bunch of tools to make it easy to build and run containers as a developer, or a systems administrator. It is basically a command-line interface (CLI) for working with containers.

So, in reality, when you run a container with `docker`, you're actually running it through the Docker daemon, which calls containerd, which then uses runc.

But the `docker` command is just one piece of the puzzle. It actually calls down to some lower-level tools to do the heavy lifting:



The projects involved in running a container with Docker

## What are the lower-level tools in the Docker stack?

From the bottom up, these are the tools that `docker` uses to run containers:

- Lowest-level  **The low-level container runtime.** runc is a low-level **container runtime**. It uses the native features of Linux to create and run containers. It follows the OCI standard, and it includes libcontainer, a Go library for creating containers.
-  **The high-level container runtime.** containerd sits above the low-level runtime, and adds a bunch of features, like transferring images, storage, and networking. It also fully supports the OCI spec.
-  **The Docker daemon.** dockerd is a daemon process (a long-running process that stays running in the background) which provides a standard API, and talks to the container runtime
- Highest level  **The Docker CLI tool.** Finally, docker-cli gives you the power to interact with the Docker daemon using `docker ...` commands. This lets you control containers without needing to understand the lower levels.

## Does Kubernetes use Docker?

A really common question is “how do containers run in Kubernetes?”. Does Kubernetes use Docker? Well, it doesn’t anymore — but it used to.

Originally, Kubernetes used Docker (Docker Engine) to run containers.

But, over time, Kubernetes evolved into a container-agnostic platform. The Container Runtime Interface (CRI) API was created in Kubernetes, which allows different container runtimes to be plugged into it.

Docker Engine, being a project older than Kubernetes, doesn’t implement CRI. So to help with the transition, the Kubernetes project included a component called **dockershim**, which allowed Kubernetes to run containers with the Docker runtime.

It bridged the gap between the old world and the new.

### Death of the shim also

But, as of Kubernetes 1.24, the dockershim component was removed completely, and Kubernetes no longer supports Docker as a container runtime. Instead, you

need to choose a container runtime that implements CRI.

The logical successor to Docker Engine in Kubernetes clusters is... containerd. (*10 points if you got that correct!*) Or you can use an alternative runtime, like CRI-O.

**This doesn't mean that Kubernetes can't run so-called Docker-formatted containers.** Both containerd and CRI-O can run Docker-formatted and OCI-formatted images in Kubernetes; they can do it without having to use the `docker` command or the Docker daemon.

## **Open Container Initiative (OCI) specifications**

The OCI was one of the first efforts at creating some standards for the container world. It was established in 2015 by Docker and others.

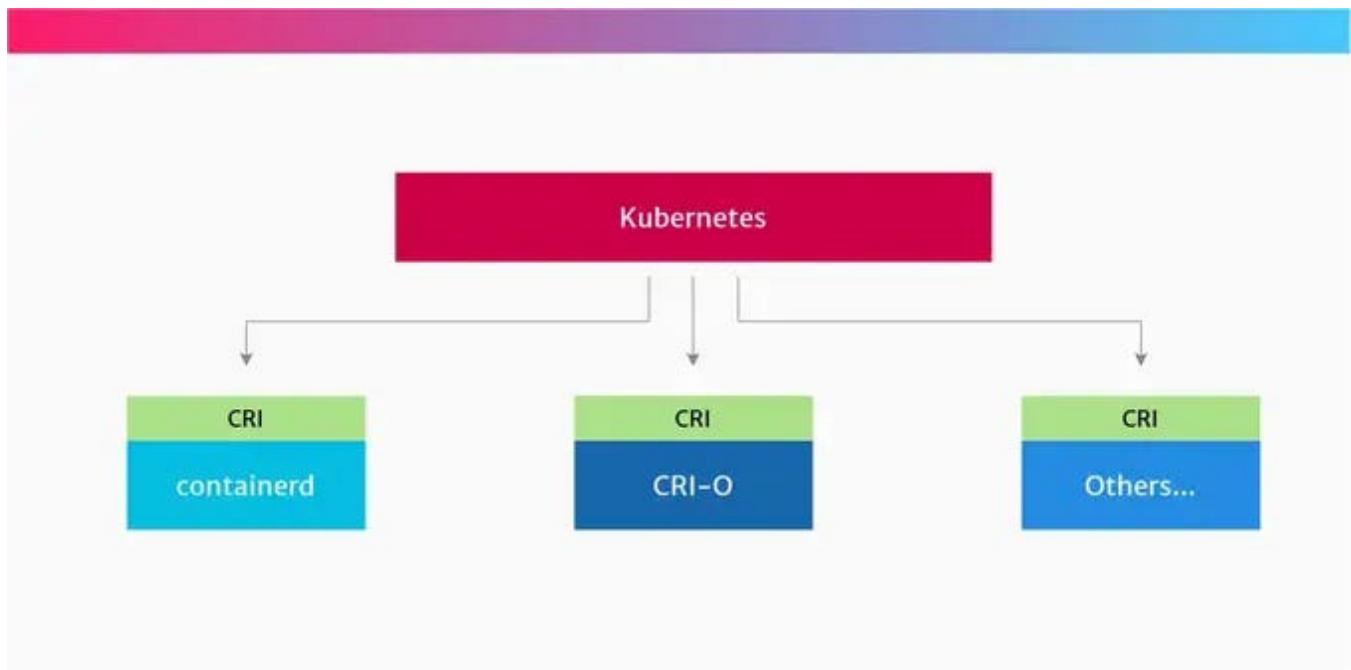
**The OCI is backed by a bunch of tech companies and maintains a specification for the container image format, and how containers should be run.**

**For example:** you might use one OCI-compliant runtime for your Linux hosts, but a different runtime for your Windows hosts.

## **Kubernetes Container Runtime Interface**

The other standard we need to talk about is the Container Runtime Interface (CRI). This is an API that was created by the Kubernetes project.

**CRI is an interface used by Kubernetes to control the different runtimes that create and manage containers.**



You can choose your own container runtime for Kubernetes

So if you prefer to use **containerd** to run your containers in Kubernetes, you can! Or, if you prefer to use **CRI-O**, then you can. This is because both of these runtimes implement the CRI specification.

But, if you pay to get support (security, bug fixes etc) from a vendor, your choice of container runtime might be made for you. For example, Red Hat's [OpenShift](#) uses **CRI-O**, and offers support for it. Docker provides support for their own **containerd**.

## **containerd and CRI-O**

We've seen that Docker Engine calls down to a bunch of lower-level tools. But what are these tools? And how do they fit together?

The first layer is the high-level runtimes: **containerd**, created by Docker, and **CRI-O**, created by Red Hat.

### **containerd**

**containerd** is a high-level container runtime that came from Docker. It implements the CRI spec. It pulls images from registries, manages them and then hands over to a lower-level runtime, which uses the features of the Linux kernel to create processes we call 'containers'.

## CRI-O

CRI-O is another high-level container runtime which implements the Kubernetes Container Runtime Interface (CRI). It's an alternative to containerd. It pulls container images from registries, manages them on disk, and launches a lower-level runtime to run container processes.

Yes, CRI-O is another container runtime. It was born out of Red Hat, IBM, Intel, SUSE .

## runc and other low-level runtimes

runc is an OCI-compatible container runtime. It implements the OCI specification and runs the container processes.

runc is sometimes called the “reference implementation” of OCI.

## Other low-level runtimes

But, runc isn't the only low-level runtime. The OCI specification is allowing other tools to implement the same functionality in a different way:

- crun a container runtime written in C (by contrast, runc is written in Go.)
- firecracker-containerd from AWS, which implements the OCI specification as individual lightweight VMs (and it is also the same technology which powers AWS Lambda)
- gVisor from Google, which creates containers that have their own kernel. It



## Summary

There is a set of open standards which, theoretically, make it easier to swap out different implementations. Projects like containerd, runc and CRI-O implement parts of those standards.

In Kubernetes, you can choose which container runtime you want to use, as long as it supports the CRI API. You can use **containerd** or **CRI-O**.



Moral of the story is :

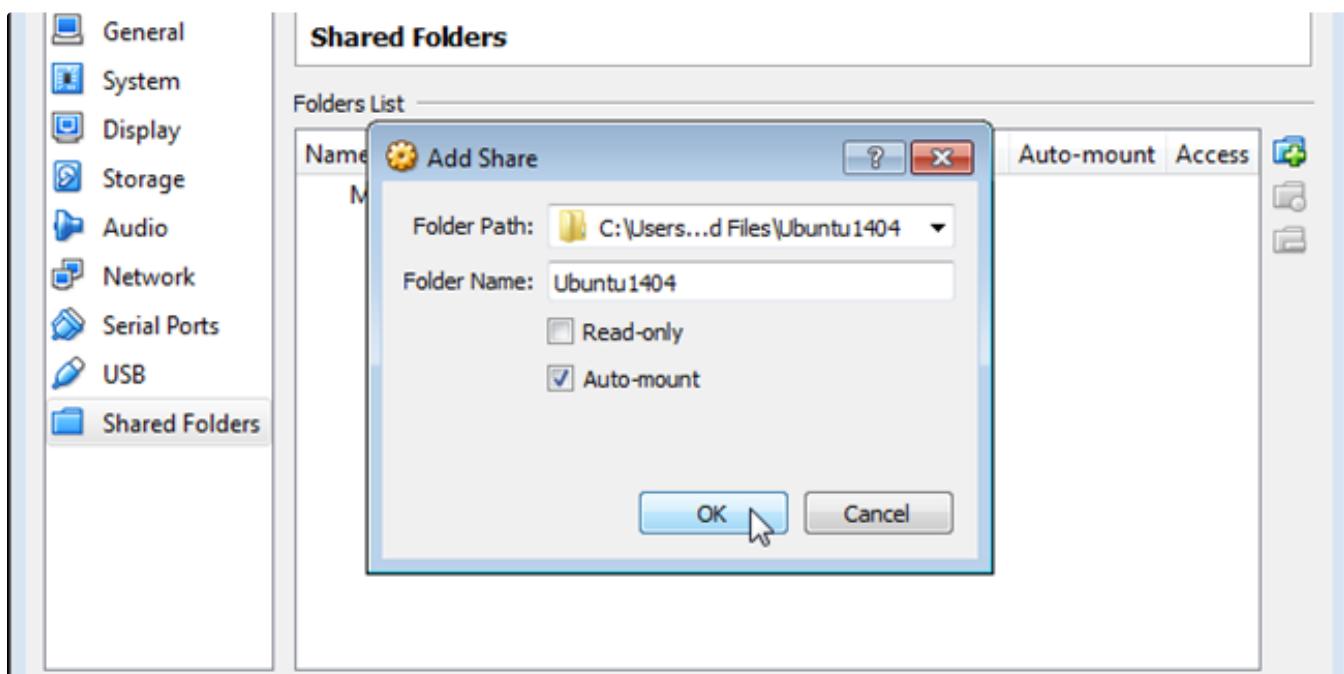
Follow



Written by **Vineet Kumar** · Mar 5, 2021 · 134 reads · Dockershim has been deprecated i.e. container can't be managed by docker command  
23 Followers

but u can use other container run time which supports CRI (K8s native API) Devops Engineer <https://vineetcumar03.github.io> by-using crictl commands (or other supported commands) to manage containers/pods instead of docker commands.

Map docker cli over crictl : <https://medium.com/@vineetcic/mapping-from-dockercli-to-crictl-life-after-docker-is-cri-a39ea5649d6c>



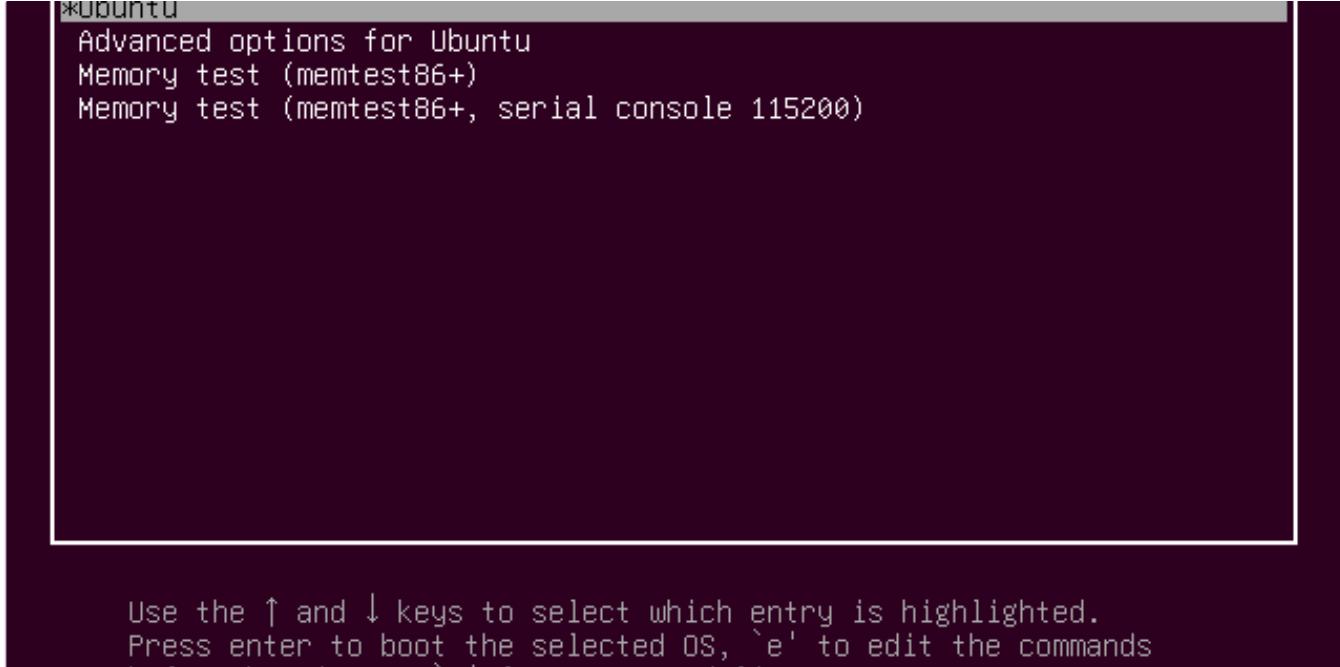
Vineet Kumar

## How to Access Folders on Ubuntu Host Machine from an Ubuntu Virtual Machine in VirtualBox

Open VirtualBox

2 min read · Mar 5, 2021





 Vineet Kumar

## How to reset password in single user mode in Ubuntu 18.04

BOOT INTO GRUB MENU: The first step is to reboot into the GRUB menu. If Ubuntu 18.04 Bionic Beaver is the only operating system installed...

2 min read · Jan 15, 2021

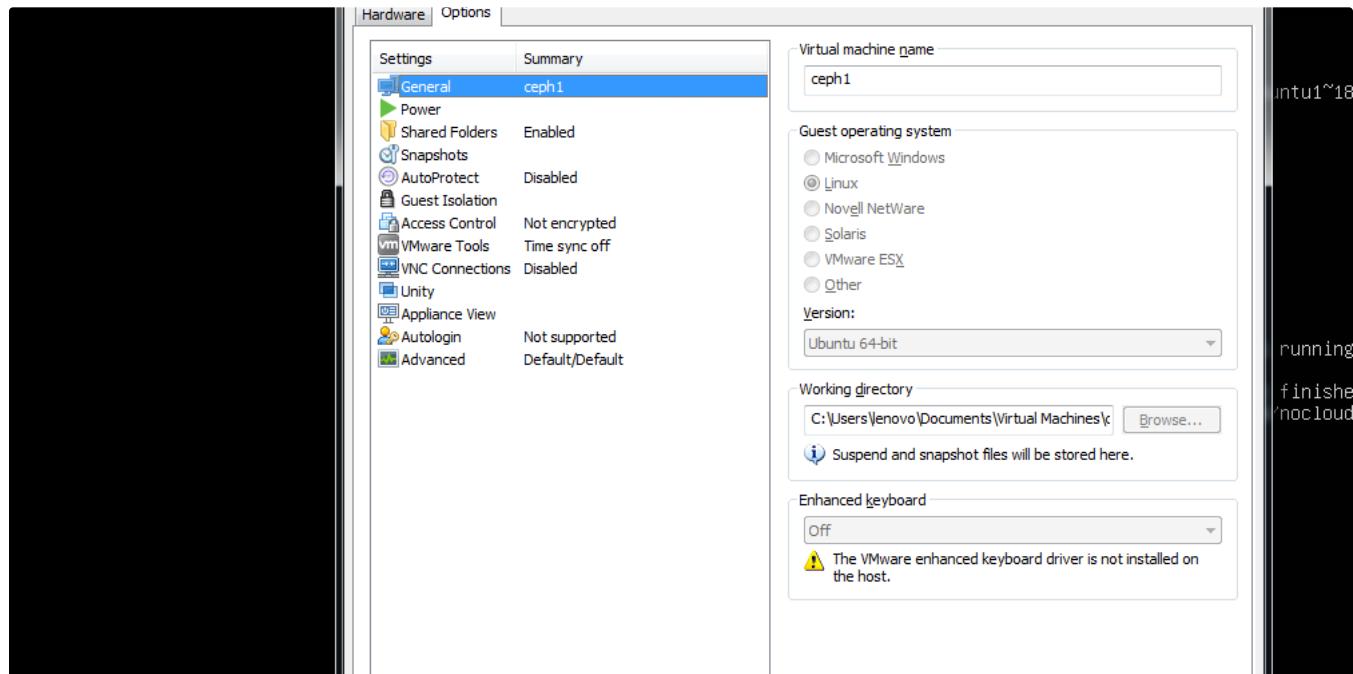


 Vineet Kumar

## How to Set the Proxy for APT on Ubuntu 18.04

### Overview

1 min read · Oct 26, 2020

 Vineet Kumar

## Configuring Shared Folders in VMWare Workstation with a UBUNTU Guest OS and Windows Host

Start VMWare Workstation, then power on Ubuntu guest OS

1 min read · Apr 3, 2020



7



See all from Vineet Kumar

## Recommended from Medium



Seifeddine Rajhi

### AWS Controllers for Kubernetes (ACK): Managing AWS Services from Kubernetes

A Path to AWS Service Control in Kubernetes

8 min read · 3 days ago

112



Muhammed Said Kaya

## Creating Air-gapped K3s System Images with Private Registry

When setting up a lightweight Kubernetes cluster with K3s, there are certain images used by the system. In air-gapped environments, you...

5 min read · Oct 2



18



### Lists



#### Coding & Development

11 stories · 255 saves



#### New\_Reading\_List

174 stories · 174 saves



#### Natural Language Processing

797 stories · 363 saves

 Addo Zhang

## Introduction to Container Network Interface (CNI)

In this article, we will learn about the CNI specification, the execution flow of network plugins, and gain a general understanding of CNI.

6 min read · May 22

 70 3

# kubernetes



## POD NETWORKING IN KUBERNETES





Extio Technology

## Mastering Kubernetes Pod-to-Pod Communication: A Comprehensive Guide

Introduction

6 min read · Jun 17



106



{ JSON } is slow?

```
{ "name": "JSON is slow!", "blog": true, "writtenAt": 1695884403, "topics": ["JSON", "Javascript"] }
```

Alternatives?



Vaishnav Manoj in DataX Journal

## JSON is incredibly slow: Here's What's Faster!

Unlocking the Need for Speed: Optimizing JSON Performance for Lightning-Fast Apps and Finding Alternatives to it!

16 min read · Sep 28

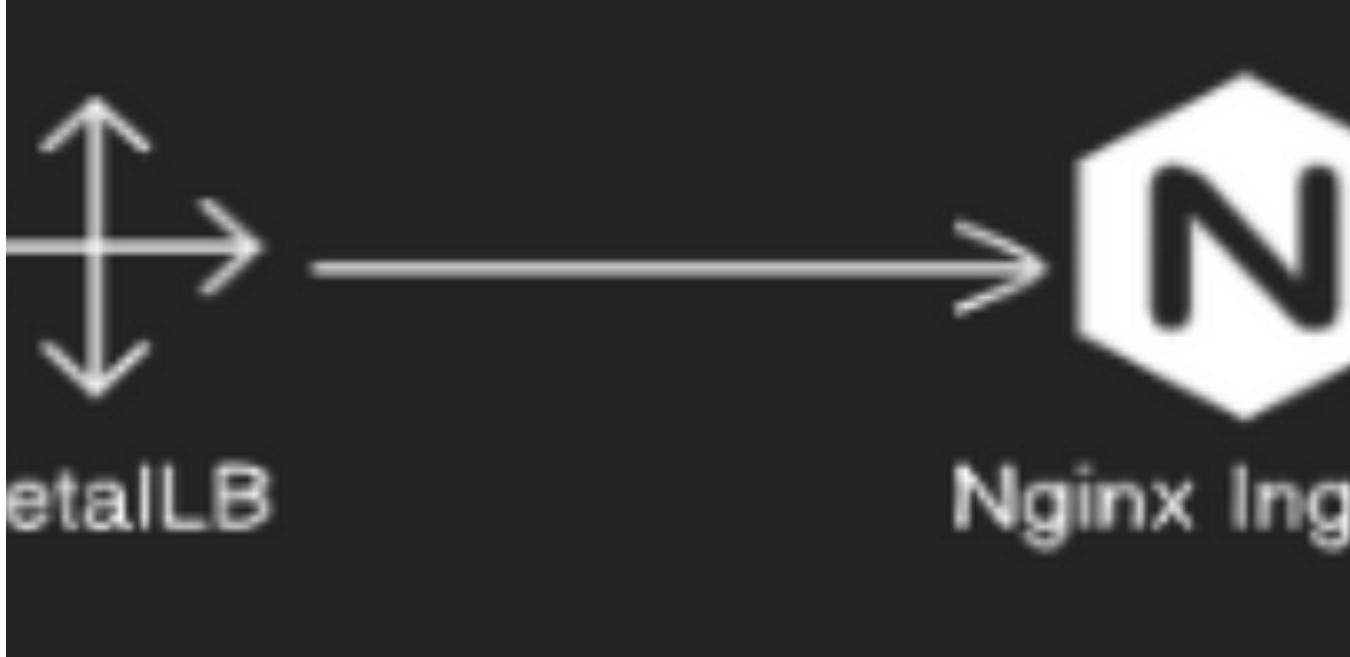


6.6K



81





 vishal acharya

## Creating an Efficient and Lightweight Kubernetes Cluster on Linux with K3D, MetalLB, and Nginx...

Prerequisites

5 min read · Aug 27



See more recommendations