Monitoring Kubernetes with ktop: A Simple CLI Tool

Yousaf K Hamza · Follow
3 min read · 15 hours ago

Listen Share ••• More

Monitoring Kubernetes clusters is crucial to maintaining a healthy infrastructure. While most administrators are familiar with popular tools like Prometheus or Grafana, there's a lightweight option that may be ideal for smaller environments or quick diagnostic tasks: **ktop**. It's a command-line tool that allows you to monitor your Kubernetes clusters directly from the terminal without the need for a fully interactive or web-based session.

In this post, I'll guide you through setting up **ktop**, understanding its functionality, and adding autocompletion for an even smoother experience.

What is ktop?

ktop is a CLI-based monitoring tool for Kubernetes, inspired by Unix-style system monitors like `top`. It allows users to inspect the state of resources within a Kubernetes cluster quickly. Although not interactive `htop` or `kubectl top`, it offers a snapshot of the current health of Kubernetes resources like pods, services, and nodes.

ktop is accessible directly via your terminal and integrates well into most CI/CD pipelines or development environments, where you can quickly gain insights without launching any additional UI components.

Key Features:

- Lightweight CLI tool
- Simple installation and usage

- Minimal system footprint
- Kubernetes resource inspection (e.g., Pods, Nodes, Cluster summery)

Installation Guide

To install **ktop**, follow these simple steps. You can do it using `go install` or use the binary, but for simplicity, we'll use a bash script to handle the installation.

1. Install Go (If not already installed)

```
# On Ubuntu:
sudo apt-get update
sudo apt-get install -y golang-go
```

2. Install ktop using Go

```
go install github.com/vladimirvivien/ktop@latest
```



```
export PATH=$PATH:$(go env GOPATH)/bin
```

Autocompletion for ktop

To make your experience with **ktop** even smoother, you can enable Bash autocompletion. This feature will suggest the appropriate flags and options while you type in commands.

Here's how to enable autocompletion:

Step 1: Create the Completion Script

Create a directory for custom completion scripts if it doesn't exist:

```
mkdir -p ~/.bash_completion.d/
```

Step 2: Add the following content to a file named:

`ktop_completion.sh` inside `~/.bash_completion.d/`:

```
#!/usr/bin/env bash

_ktop() {
    local cur opts
    COMPREPLY=()
    cur="${COMP_WORDS[COMP_CWORD]}"

# Retrieve both Flags and Global Options
    opts="$(ktop --help | awk '/^Flags:/{flag=1; next} /^Global
Options:/{flag=2} flag==1 && NF {print $1} {print $2} flag==2 && NF
{print $1}' | grep -v ^$ | grep ^"-" | sed 's/[,]//g')"

# Generate completion suggestions based on current word
    COMPREPLY=( $(compgen -W "${opts}" -- ${cur}) )
    return 0
}

complete -F _ktop ktop
```

Step 3: Enable Autocompletion

To activate the autocompletion script, you'll need to source it within your `.bashrc` or `.bash_profile` file. Open your `.bashrc` and append the following line:

```
source ~/.bash_completion.d/ktop_completion.sh
```

Then, reload your bash configuration:

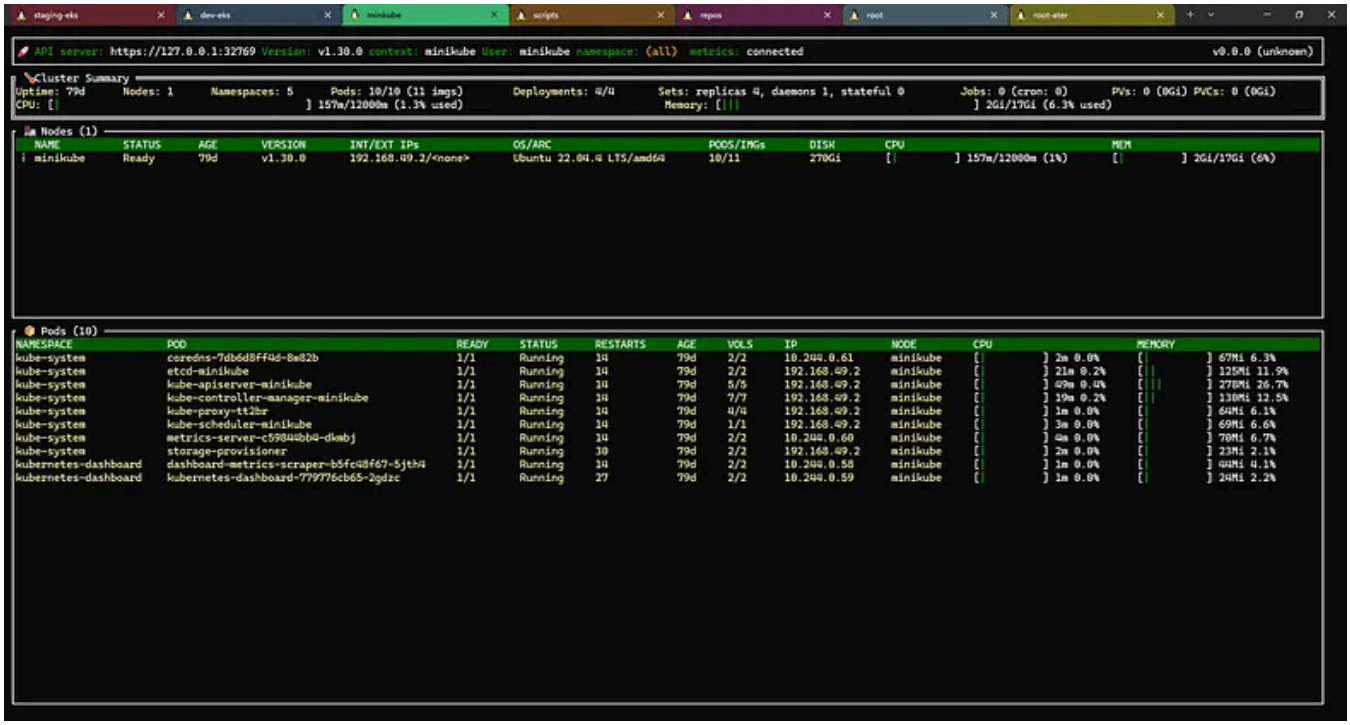
```
source ~/.bashrc
```

Now, when you type `ktop` in your terminal, it will suggest available flags and global options, enhancing your experience with the tool.

Using ktop

Once installed, **ktop** is very easy to use. Run the following command to get a quick view of the Kubernetes resources:

```
ktop
```

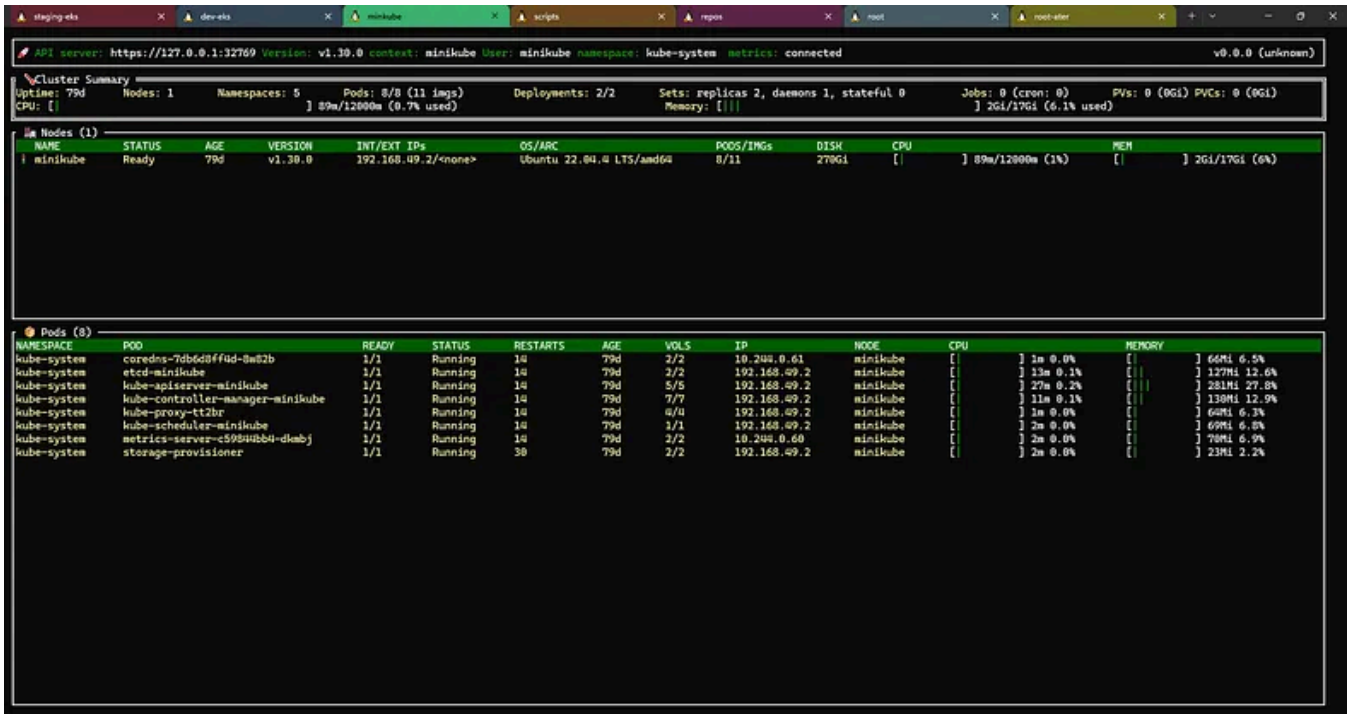


ktop

You'll immediately see a snapshot of your Kubernetes cluster's health, including pods, nodes, services, and their status. By passing different flags, you can further customize the output to suit your monitoring needs.

For example:

```
ktop --namespace kube-system
```



ktop — namespace kube-system

Reference

https://github.com/vladimirvivien/ktop

Conclusion

For anyone who needs a lightweight, no-frills way to monitor Kubernetes clusters from the terminal, **ktop** is an excellent tool. It doesn't require a lot of resources, yet it provides enough information to make quick decisions about the health of your Kubernetes environment.

With the added bash autocompletion, you can now monitor your Kubernetes clusters more efficiently, with all the available options at your fingertips. Happy monitoring!

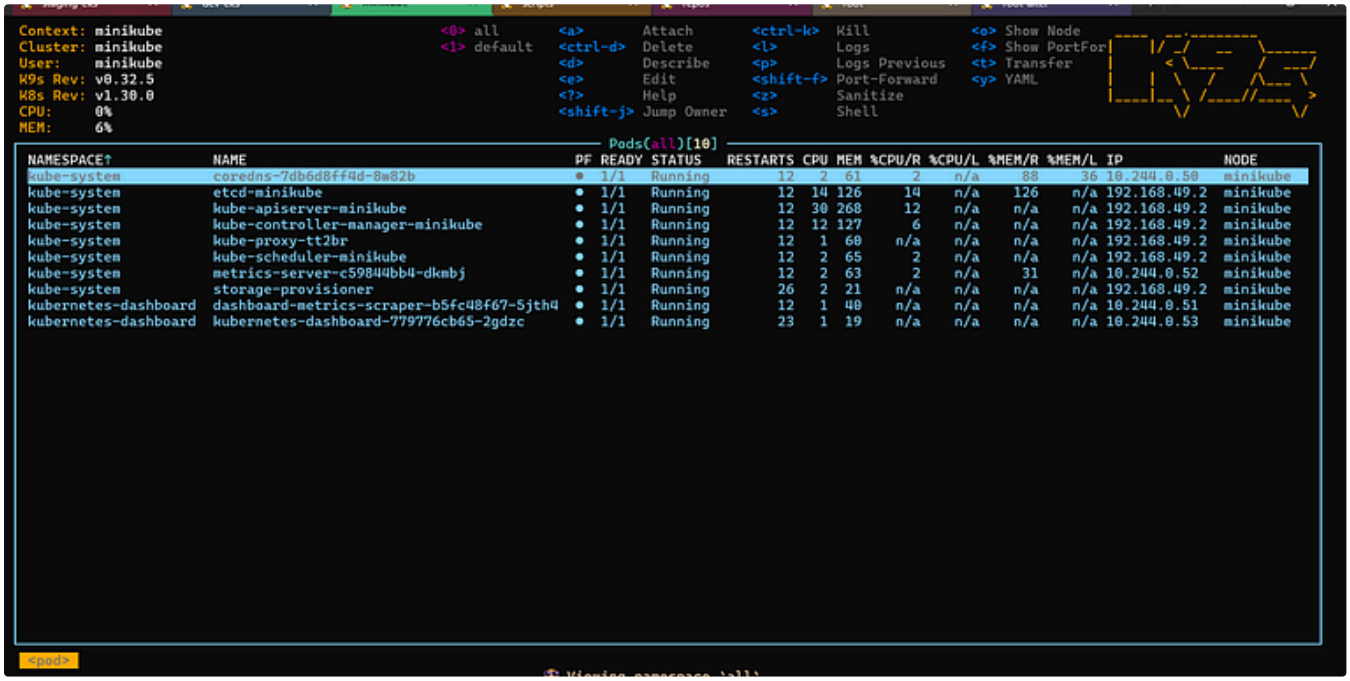
Kubernetes Monitoring Cli Lightweight

Follow

Written by Yousaf K Hamza

4 Followers

More from Yousaf K Hamza



Yousaf K Hamza

K9s: Your Powerful Ally in Kubernetes Management

Managing Kubernetes clusters can be a complex task in the ever-evolving container orchestration world. Enter K9s, a terminal-based UI...

Sep 18 •••

See all from Yousaf K Hamza

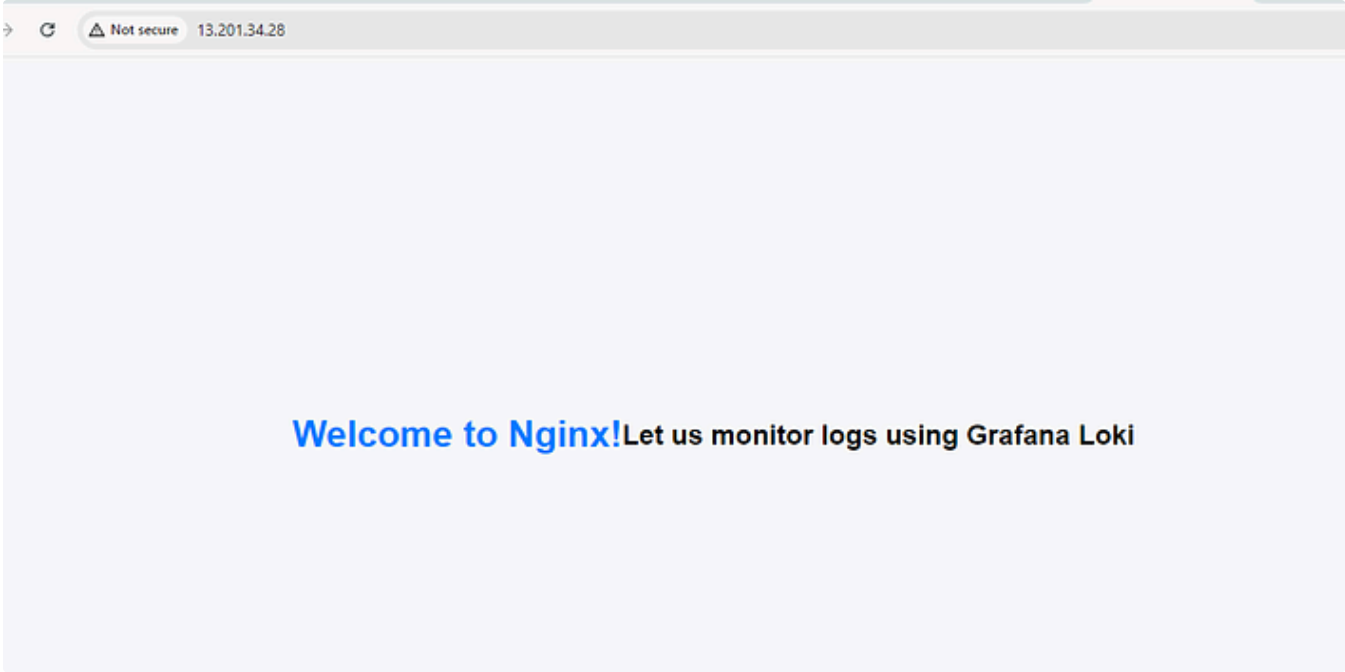
Recommended from Medium

Mr.PlanB

Docker, Proxmox, and VMs: The Ideal Server Setup for Maximum Control

In today's tech landscape, the demand for flexibility, efficiency, and control in server management has never been higher. As businesses...

Sep 25 52



Naman Sharma in DevOps.dev

Harnessing the Power of Grafana Loki: Pro Tips for Nginx Log Management

What is Grafana , Grafana-Loki and Promtail.

4d ago 1

Lists

Natural Language Processing

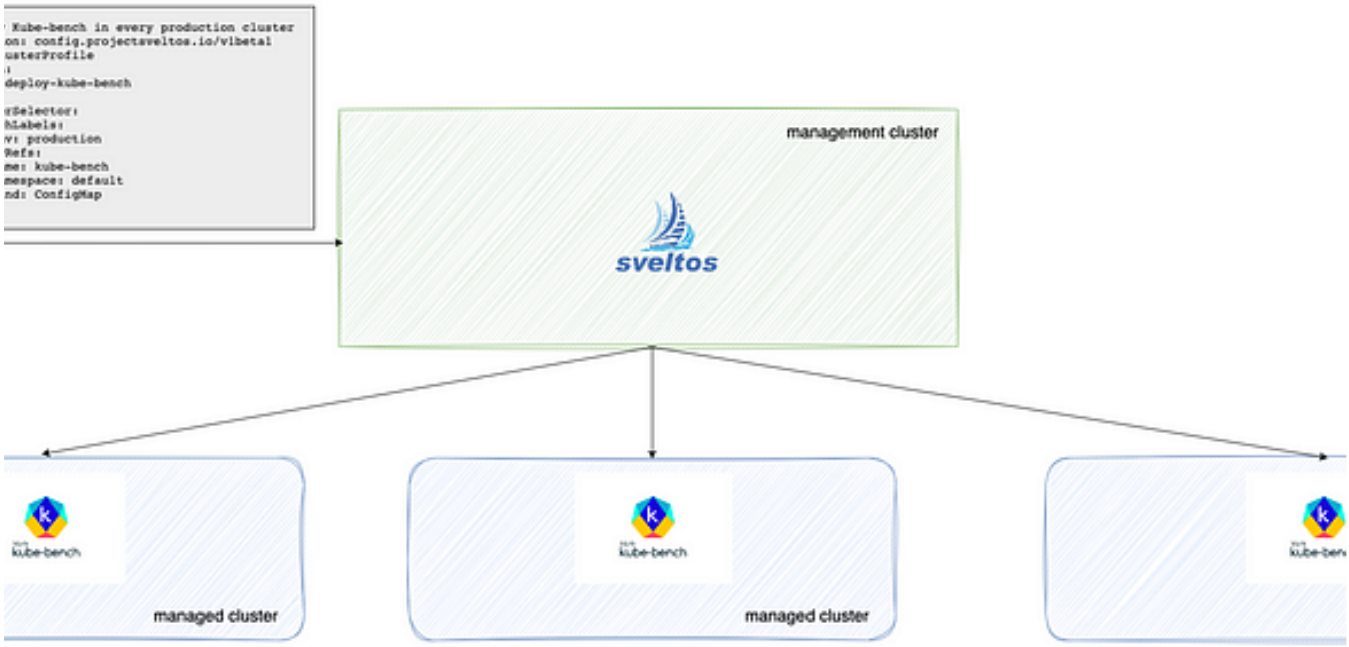
1733 stories · 1314 saves

Harendra

Top 10 Most Used Open Source SaaS Products

Start using top open-source products for your daily tasks and save money

Sep 24 799 5



Gianluca Mardente in ITNEXT

CIS Benchmark Compliance Across Multiple Kubernetes Clusters

CIS Kubernetes Benchmarks are a set of security best practices and recommendations developed by the Center for Internet Security (CIS)...

Sep 25 64 1

Rahul Sharma in AWS in Plain English

I have Asked This SSH Question in Every AWS Interview—And Here's the Catch

When I interview people, I always ask questions about problems that people face in the real world.

Sep 16 1.1K 45

Kevin

Helm Cheat Sheet

Helm cheatsheet features all the commands required to manage an application through Helm.

Sep 22 7

See more recommendations