K3s – A Practical Introduction

Thursday, September 30th, 2021

VSHN K3S

Adrian Kosmaczewski, Developer Relations

Speaker notes

Welcome and thank you so much for joining this webinar! My name is Adrian Kosmaczewski, I'm Developer Relations at VSHN, the DevOps company, and I greet you from the beautiful city of Zürich, Switzerland. Today it's Thursday, September 30th, 2021, and today we're going to be talking about the K3s Kubernetes distribution.

There's a Q&A box for your questions, so please feel free to ask questions as the talk progresses, and I will answer them at the end of the presentation.

VSHN

- Pronounced ˈvɪʒn – like "vision"
- *The DevOps Company*
- Founded 2014, 46 VSHNeers located in Zürich (we’re hiring!)
- Switzerland’s leading DevOps, Docker & Kubernetes partner
- 24/7 support
- ISO 27001 certified
- ISAE 3402 Report Type 1 verified
- First Swiss Kubernetes Certified Service Provider

Speaker notes

Just a few words about VSHN; that’s how you pronounce the name, and we’re "The DevOps Company". We’ve been in Zurich since 2014, we’re 46 VSHNeers and we’re Switzerland’s leading DevOps, Docker & Kubernetes partner, offering 24/7 support to our customers. We’ve got a few certifications, and most importantly, we were the First Swiss Kubernetes Certified Service Provider back in 2016.

APPUiO

SWISS CONTAINER PLATFORM

Project Syn

k8up

Speaker notes

We also run our own "Platform as a Service" offering called "APPUiO". We’ve created our own suite of tools to manage lots of Kubernetes services from a central location, called "Project Syn". Last but not least, we have developed our own Kubernetes operator for backups, called K8up, which just like Project Syn is 100% open source on GitHub.

SUSE

EXOSCALE cloudscale.ch

GitLab Google

Red Hat OPENSHIFT

Microsoft Azure

Crossplane docker NGINX Part of F5

Speaker notes

We're partners with many companies very active in the Cloud Native space, you might recognize some of the logos on this slide. Of course tonight we're going to talk about SUSE and one of the their most fantastic products.

Speaker notes

But of course the star of the evening is K3s, so let's get started and learn what is K3s and how we can use it.

Agenda

1. What is K3s?
2. What is it used for?
3. How does it work?
4. Demo time!
5. Where to find more information?

Speaker notes

The agenda for today is quite simple, and will consist of five major sections: first we're going to learn what is K3s, and what it is used for; then we are going to learn how it works, and of course, we're going to have a small demo of K3s in action, live. Finally I will give you some pointers to additional information about K3s, for you to continue your discovery path.

1. What is K3s?

K3S

Speaker notes

So, on to the first point: what is K3s?

K3S

- Fully Conformant Kubernetes distribution
- Lightweight
- Simple to install, use, and maintain
- Optimized for ARM
- Fully open source
- Introduced by Rancher in February 2019
- Version 1.0 in November 2019
- CNCF Sandbox Project since June 2020
- Website: k3s.io

Speaker notes

K3s is a fully conformant Kubernetes distribution. The link on the slide points directly to the location of the automated conformance tests that prove that each release of K3s passes all required criteria.

But it is not just any Kubernetes distribution; it is a very lightweight one, distributed as a single binary of less than 50 MB. Very simple to install, use, and maintain, it is optimized for ARM CPUs, which makes it a great distribution to use with Raspberry Pis and other small boards. It is fully open source, introduced in 2019, and donated to the CNCF in June 2020.

About the Name

$$\texttt{len("Kubernetes") = 10} \Rightarrow \texttt{"K8s"}$$

$$\frac{10}{2} = 5 = \texttt{len("K\_\_\_s")} \Rightarrow \texttt{"K3s"}$$

Nobody knows what the _ stands for!

No official pronunciation

Speaker notes

The same way I explained to you how to pronounce the word "VSHN" at the beginning of this session, let me explain to you the origin of the name K3s. There is an algorithm behind, as you'll see.

You all know that Kubernetes is commonly referred to as K8s; well, it turns out the Kubernetes word is 10 letters long; and since K3s requires half the memory of Kubernetes, they looked for a word with 5 letters, which would be cut as "K3s".

Of course, nobody knows what would go in between the "K" and the "S", and there's no official pronounciation. So, "Kay three ess" is the way to say it.

2. What is K3s Used for?

K3S

Speaker notes

We all associate Kubernetes with big clusters. So, what can we use K3s for?

Use Cases

- Edge
- IoT
- Development
- CI/CD
- Embedded / ARM
- Teaching / Learning

Speaker notes

K3s has a lot of interesting use cases. These days lots of companies are moving into the "Edge computing" area, distributing workloads into laptops and small devices. K3s is an ideal distribution for that, just as for the "Internet of Things". Developers can install it in their laptops to simulate deployments in

It can be used as a CI/CD platform, to embed in your pipelines, applying deployments and performing integration tests. It can work in embedded devices with ARM CPUs, and of course, it’s a great way to learn (and teach) Kubernetes to engineers.

> *"Situations where a PhD in K8s clusterology is infeasible"*

Source: rancher.com/docs/k3s/latest/en

Speaker notes

But by far the best use case is the one quoted in this slide, which is actually mentioned in the K3s documentation: "K3s is a great Kubernetes option when a PhD in clusterology is infeasible." Just perfect.

K3s at VSHN

- K8up Getting Started guide
- Writing Project Syn components tutorial
- APPUiOli at KubeCon 2019 Barcelona

Speaker notes

At VSHN we’ve been great fans of K3s since it first came out. We started using it in demos and presentations as soon as it appeared. For example, the getting started guide for K8up, our Kubernetes backup operator, uses K3s; just like the tutorial to learn how to write Commodore components to use with Project Syn.

Last but not least, we’ve demoed APPUiOli during our visit to KubeCon 2019 in Barcelona.

Speaker notes

And if you wonder what is APPUiOli, well, here it is: a small Kubernetes cluster made of ARM boards, running K3s in its core.

Source: tobru.ch/kubernetes-on-orangepi-arm64

Features

- Lightweight
  - 50% memory requirements
- Single binary
  - < 50 MB binary with all components
- SQLite3 backend
- Packaged addons
- Secure by default

Speaker notes

K3s has many features, but by far the most important is that it is very lightweight; not only it consumes "half the memory" required by a standard Kubernetes cluster, it is also deployed as a lightweight single binary (written in Go) which weighs less than 50 MB.

Instead of using etcd, it uses SQLite by default; it has many packaged addons (which can be enabled and disabled as needed at startup) and it has been built with security in mind.

Packaged Addons

- Flannel
- Helm
- Traefik (default ingress)
- CoreDNS
- …

Speaker notes

Speaking about the addons, here are some of the most common: Flannel, Helm, Traefik, CoreDNS, and many more.

By the way, Traefik is the default ingress, but as I said, it can be disabled, so that you can install your own ingress component.

3. How does K3s Work?

K3S

Speaker notes

Let us dive a bit deeper in the mechanisms that make K3s so good.

Architecture

1. Server
   - Packaged control plane
   - API + Scheduler + Controller Manager + sqlite3 + tunnel proxy
2. Agent
   - Worker node
   - Tunnel proxy + kube proxy + kubelet + Flannel
3. Outside of K3s processes:
   - containerd + pod + pod + pod + …

Speaker notes

Just like any other Kubernetes cluster, we have master nodes, and worker nodes. In K3s terminology, we have the Server, which is the master node; it contains a full packaged control plane, with all of the required components in it.

On the other side, we have one or many Agents, which are the worker nodes; they also contain all they need to run pods and workloads.

Finally, all pods run outside of the main K3s processes, using containerd.

Requirements

- At least 2 GB of RAM
  - Documentation states 512 MB but pull request
- 1 CPU (2 recommended for K3s server)
- SSD (recommended)

Speaker notes

To run the latest versions of K3s we need at least 2 GB of RAM available for the Server, and around 1 GB available per Agent. Pay attention, since the documentation still states 512 MB of RAM minimum, but there’s an open pull request on GitHub to correct that mistake.

Also, each Server and Agent require a CPU, but in the case of the Server, it’s better to assign 2 CPUs.

For storage, of course, using an SSD drive is way faster.

Operating System

- Most Linux distributions
- Special instructions for
  - Raspian Buster
  - Alpine Linux
  - RHEL / CentOS
- Windows not supported

Speaker notes

K3s has been designed with Linux in mind; K3s runs in the most popular distributions, starting with SUSE Linux and openSUSE of course.

In the documentation there are some special sections for Raspian (for use in Raspberry Pi devices), Alpine Linux, and RHEL & CentOS.

Be aware, however, that at this time K3s does not run natively in Windows.

Network Ports

- API server (TCP port 6443)
- Flannel VXLAN (UDP port 8472)
- Metrics server (TCP port 10250)

Speaker notes

A K3s cluster with default settings will expose, as expected, the Kubernetes API on port 6443, plus some other ports that expose VXLAN and metrics information.

Database

- Default SQLite
- For large clusters:
  - MySQL
  - PostgreSQL
  - etcd
- Datastore shim: Kine github.com/k3s-io/kine

Speaker notes

I mentioned earlier that K3s uses SQLite by default; but for larger clusters, you can switch it to MySQL, PostgreSQL, or even etcd, thanks to the Kine project, which provides a layer of abstraction between K3s and the underlying data store.

Installation

1. Server

```
$ curl -sfL https://get.k3s.io | sh -
$ cat /var/lib/rancher/k3s/server/node-token
$ cat /etc/rancher/k3s/k3s.yaml
```

2. Agent(s)

```
$ export K3S_URL=https://server:6443
$ export K3S_TOKEN=...
$ curl -sfL https://get.k3s.io | sh -
```

Each node must have a unique hostname: `K3S_NODE_NAME` to override.

Speaker notes

To install K3s, you need two or more machines (physical, virtual, or containers); the first one will contain the Server, and it can be bootstrapped very easily using the `curl` command shown on screen.

Once this is done, K3s is configured to automatically restart in case of a reboot, and it exposes two important files: the token used to attach new Agents to this server, and the Kubeconfig file, which can be used by `kubectl` or similar tools to talk to the Kubernetes API.

For each Agent, we need at least to specify two environment variables: `K3S_URL` contains the URL of the Kubernetes API exposed by the server; and `K3S_TOKEN` contains the value of the token stored in the server.

Other Ways to Install

K3D

K3OS

github.com/eezhee/eezhee

k3sup.dev

Speaker notes

Of course, installing K3s is fun once or twice, but it can quickly become boring; so there are more ways to run a K3s cluster these days.

First of all, we need to mention the `k3d` project, which allows, as the name implies, to run "K3s in Docker".

Another option is to use K3OS, a Linux distribution specially created just to run K3s. We're going to talk more about it in a minute.

And if you need more help, here's the `eezhee` and `k3sup` projects, which both allow to easily bootstrap K3s clusters. `eezhee` is meant to be create clusters in DigitalOcean, Linode and Vultr, while `k3sup` works for any node where you have SSH access.

K3D

```
#!/usr/bin/env bash
HOSTNAME=rancher-demo.eu.ngrok.io

k3d cluster create k3s-rancher --servers 1 --agents 3 ❶
k3d kubeconfig get k3s-rancher > k3s.kubeconfig ❷

# Install cert-manager
helm --kubeconfig k3s.kubeconfig install cert-manager jetstack/cert-manager --namespace cert-manager \
       --create-namespace --version v1.3.1 --set installCRDs=true
kubectl --kubeconfig k3s.kubeconfig -n cert-manager rollout status deploy/cert-manager

# Install Rancher
helm --kubeconfig k3s.kubeconfig install rancher rancher-latest/rancher --namespace cattle-system \
       --create-namespace --set hostname=$HOSTNAME
kubectl --kubeconfig k3s.kubeconfig -n cattle-system rollout status deploy/rancher
```

❶ Creates a K3s cluster in Docker with 1 server and 3 agents

❷ Retrieves `kubeconfig`

Website: k3d.io

Speaker notes

Let's talk a bit about k3d, which is at VSHN our preferred way to spin up a new K3s cluster easily. The command `k3d cluster create` will basically do that for us, in a few seconds, and then we can retrieve the kubeconfig using the `k3d kubeconfig get` command.

In the example in this slide, we are then using the Kubeconfig to install cert-manager and the Rancher cluster management tool.

K3OS

- Full Linux distribution just for K3s!
  - Boots with a K3s cluster ready to use
- Built with linuxkit
- Ubuntu 20.04 kernel
- Alpine userland (BusyBox)
- OpenRC init system
- Announced April 2019
- Website: k3os.io

Speaker notes

And finally, a word about K3OS; this is a crazy project, meant to create a small Linux distribution that boots a K3s cluster, ready to use! It is built with linuxkit, and its upstream kernel is Ubuntu 20.04. It uses an Alpine userland based on BusyBox, and uses OpenRC as init system. It was announced in April 2019.

4. See K3s in Action!

Speaker notes

Finally! Lots of words, zero action; let’s see how K3s work in real life.

In this demo we’re going to see K3s running in VirtualBox with Vagrant, and then we’re going to use K3d to spin up a new cluster quickly.

5. More Information?

K3S

Speaker notes

Now that we've seen what K3s is and how it works, let's see what the community is doing with it.

Ideas

GitHub Action for K3s

github.com/debianmaster/actions-k3s

Visual Studio Code extension

github.com/inercia/vscode-k3d

"i built a Raspberry Pi SUPER COMPUTER!! // ft. Kubernetes (k3s cluster w/ Rancher)" (July 2021)

www.youtube.com/watch?v=X9fSMGkjtug

Speaker notes

People are becoming crazy with K3s, and there's a lot of interesting things being built around it. Suffice to mention the GitHub Actions for K3s, which allow developers to create a small cluster in their GitHub Action pipelines; and the Visual Studio Code extension for K3d, which gives you quick access to your local clusters.

Links

K3s with Vagrant

gitlab.com/akosma/k3s-in-vagrant

Introduction by Rancher (May 2021)

rancher.com/blog/2021/introduction-to-k3s

A Cloud Guru course

acloudguru.com/course/introduction-to-k3s

5 Reasons to Use K3s (Sept 2020)

acloudguru.com/blog/engineering/5-reasons-to-use-k3s

Speaker notes

And here are some interesting links for you to discover more about K3s.

Thanks!

VSHN K3S

Adrian Kosmaczewski, Developer Relations – adrian@vshn.ch

VSHN AG – Neugasse 10 – CH-8005 Zürich – +41 44 545 53 00 – vshn.ch – info@vshn.ch

Speaker notes

Thanks a lot for your attention! I hope that this short introduction to K3s has been interesting to you and that you will launch many K3s clusters in the future.