

kubeadm

ow

Agenda

- **Introduction to kubeadm**
- **Using kubeadm to create a cluster**
- **Configuring a Kubernetes cluster with kubeadm**
- **Kubeadm init**
- **Kubeadm join**

What is kubeadm

- Kubeadm is a tool that can be used to build a minimal viable Kubernetes Cluster of production grade
- Using kubeadm, you can create a minimum viable Kubernetes cluster that conforms to best practices.
- The kubeadm tool is good if you need:
 - A simple way for you to try out Kubernetes, possibly for the first time.
 - A way for existing users to automate setting up a cluster and test their application.
- Kubeadm creates full pki setup and provides full prod equivalent setup
- Kubeadm is a tool built to provide kubeadm init and kubeadm join as best-practice "fast paths" for creating Kubernetes cluster

Kubeadm init workflow

- kubeadm init bootstraps a Kubernetes control-plane node by executing the following steps:
 - Runs a series of pre-flight checks to validate the system state before making changes. Some checks only trigger warnings, others are considered errors and will exit
 - Generates a self-signed CA to set up identities for each component in the cluster. The user can provide their own CA cert and/or key by dropping it in the cert directory configured via --cert-dir (/etc/kubernetes/pki by default).
 - Writes kubeconfig files in /etc/kubernetes/ for the kubelet, the controller-manager and the scheduler to use to connect to the API server, each with its own identity, as well as an additional kubeconfig file for administration named admin.conf.

• •

- Generates static Pod manifests for the API server, controller-manager and scheduler. In case an external etcd is not provided, an additional static Pod manifest is generated for etcd.
- Static Pod manifests are written to `/etc/kubernetes/manifests`; the kubelet watches this directory for Pods to create on startup.
- Apply labels and taints to the control-plane node so that no additional workloads will run there.
- Generates the token that additional nodes can use to register themselves with a control-plane in the future

• •

- Makes all the necessary configurations for allowing node joining with the [Bootstrap Tokens](#) and [TLS Bootstrap](#) mechanism:
 - Write a ConfigMap for making available all the information required for joining, and set up related RBAC access rules.
- Installs a DNS server (CoreDNS) and the kube-proxy addon components via the API server.

Kubeadm join

- The "kubeadm join" command is used to add a new worker node to an existing Kubernetes cluster. When you initially set up a Kubernetes cluster using the "kubeadm init" command on the control plane node, it generates a unique token. This token is required for worker nodes to join the cluster.

```
sudo kubeadm join --token <token> <control_plane_ip>:<port> --  
discovery-token-ca-cert-hash <ca_cert_hash>
```

- After executing the "kubeadm join" command, the worker node will establish a connection with the control plane. It will download the necessary components and dependencies to become part of the Kubernetes cluster.