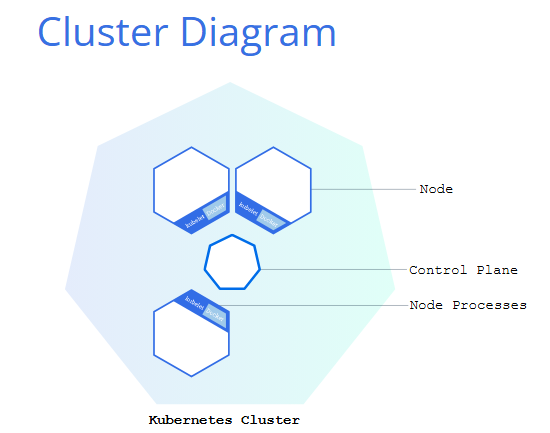
Kubernetes Notes

Kubernetes coordinates a highly available cluster of computers that are connected to work as a single unit. Kubernetes automates the distribution and scheduling of application containers across a cluster in a more efficient way.

A Kubernetes cluster consists of two types of resources:

* The **Control Plane** coordinates the cluster
* **Nodes** are the workers that run applications



The Control Plane coordinates all activities in your cluster, such as scheduling applications, maintaining applications' desired state, scaling applications, and rolling out new updates.

**A node is a VM or a physical computer that serves as a worker machine in a Kubernetes cluster.** Each node has a Kubelet, which is an agent for managing the node and communicating with the Kubernetes control plane. The node should also have tools for handling container operations, such as [containerd](https://containerd.io/docs/" \o "" \t "_blank) or [CRI-O](https://cri-o.io/#what-is-cri-o). A Kubernetes cluster that handles production traffic should have a minimum of three nodes because if one node goes down, both an [etcd](https://kubernetes.io/docs/concepts/architecture/" \l "etcd) member and a control plane instance are lost, and redundancy is compromised. You can mitigate this risk by adding more control plane nodes.

Node-level components, such as the kubelet, communicate with the control plane using the [Kubernetes API](https://kubernetes.io/docs/concepts/overview/kubernetes-api/).

**Hacks: End users can also use the Kubernetes API directly to interact with the cluster.**

By default, the dashboard is only accessible from within the internal Kubernetes virtual network. The dashboard command creates a temporary proxy to make the dashboard accessible from outside the Kubernetes virtual network.

To stop the proxy, run Ctrl+C to exit the process. After the command exits, the dashboard remains running in the Kubernetes cluster. You can run the dashboard command again to create another proxy to access the dashboard.

**A Kubernetes**[***Pod***](https://kubernetes.io/docs/concepts/workloads/pods/)**is a group of one or more Containers, tied together for the purposes of administration and networking. The Pod in this tutorial has only one Container. A Kubernetes**[***Deployment***](https://kubernetes.io/docs/concepts/workloads/controllers/deployment/)**checks on the health of your Pod and restarts the Pod's Container if it terminates. Deployments are the recommended way to manage the creation and scaling of Pods.**

**Imp: The cluster itself has a manifest too!** View it using the config