

Kubernetes On-premises – Challenges

Etc – manage highly available etcd cluster. You need to take frequent backups to ensure business continuity in case the cluster goes down and the etcd data is lost.

Load balancing – Load balancing may be needed both for your cluster master nodes and your application services running on Kubernetes. Depending on your existing networking setup, you may want to use a load balancer such as F5 or use a software load balancer such as metallb.

Availability – Its critical to ensure that your Kubernetes infrastructure is highly available and can withstand data center and infrastructure downtimes. This would mean having multiple master nodes per cluster, and when relevant, having multiple Kubernetes clusters, across different availability zones.

Auto-scaling – Auto-scaling for the nodes of your cluster can help save resources because the clusters can automatically expand and contract depending on workload needs. This is difficult to achieve for bare metal Kubernetes clusters, unless you are using a bare metal automation platform such as open source Ironi or Platform9's Managed Bare Metal.

Networking – Networking is very specific to your data center configuration.

Persistent storage – Majority of your production workloads running on Kubernetes will require persistent storage – block or file storage. The good news is that most of the popular enterprise storage vendors have CSI plugins and supported integrations with Kubernetes. You will need to work with your storage vendor to identify the right plugin and install any additional needed components before you can integrate your existing storage solution with Kubernetes on-premises.

Upgrades – You will need to upgrade your clusters roughly every 3 months when a new upstream version of Kubernetes is released. The version upgrade may create issues if there are API incompatibilities introduced with newer version of Kubernetes. A staged upgrading strategy where your development / test clusters are upgraded first before upgrading your production clusters is recommended.

Monitoring – You will need to invest in tooling to monitor the health of your Kubernetes clusters in your on-premises Kubernetes environment. If you have existing monitoring and log management tools such as Datadog or Splunk, most of them have specific capabilities around Kubernetes monitoring. Or you may

consider investing in an open-source monitoring stack designed to help you monitor Kubernetes clusters such as Prometheus and Grafana.