# Azure Kubernetes Service – Best practices from Microsoft

**On Multi-tenancy in Kubernetes**
Multi-tenancy in Kubernetes is not easy but it is doable. We have published a few guidelines around that:

- Best practices for cluster isolation
  - Includes multi-tenancy core components and logical isolation with namespaces.
- Best practices for basic scheduler features
  - Includes using resource quotas and pod disruption budgets.
- Best practices for advanced scheduler features
  - Includes using taints and tolerations, node selectors and affinity, and inter-pod affinity and anti-affinity.
- Best practices for authentication and authorization
  - Includes integration with Azure Active Directory, using Kubernetes role-based access control (Kubernetes RBAC), using Azure RBAC, and pod identities.

**On RBAC**
One approach is to use Azure RBAC for Kubernetes authorization which in essence allows for the integration of AAD users, groups with K8s RBAC. We have a detailed description of the various implementations of these solutions here: https://docs.microsoft.com/en-us/azure/aks/concepts-identity#azure-rbac-to-authorize-access-to-the-aks-resource.

**On Day-2 Operations**
We have published a guide that details our vision for Day-2 operations with Kubernetes. The document can be found here: https://docs.microsoft.com/en-us/azure/architecture/operator-guides/aks/day-2-operations-guide

We break Day-2 operations into the following:

- Triage practices for AKS operations
- Patch and upgrade AKS worker nodes
- Monitoring Azure Kubernetes Service (AKS) with Azure Monitor
- AKS troubleshooting

We do base our recommendations out of the baseline architecture for an AKS cluster.

**On a reference architecture for highly regulated workloads**
There is also a version of the baseline that was designed to run sensitive workloads. You can see the reference architecture here and the reference implementation here.

**On a security baseline for Azure Kubernetes Services**
You should also look into the security baseline for Azure Kubernetes Services as we have detailed guidance on various security domains such as *network security, Incident response and Red Team exercises.*

**On cluster lifecycle management**

Currently, we recommend that customers use our upgrade solutions, either through az aks upgrade or through the auto-upgrade channels:

For a node image upgrade, you'd still use az aks upgrade: https://docs.microsoft.com/en-us/azure/aks/node-image-upgrade

There are known issues when using kured on a cluster where the Kubernetes cluster autoscaler is enabled. This has been documented on these issues: https://github.com/Azure/AKS/issues/1773 and https://github.com/weaveworks/kured/issues/93