CloudIQ

Kubernetes Maturity Assessment Checklist

Understand where you are in your Kubernetes adoption journey, identify the gaps, where to focus and why.

Kubernetes CERTIFIED Service Provider

Kubernetes TRAINING PARTNER

Table of Contents

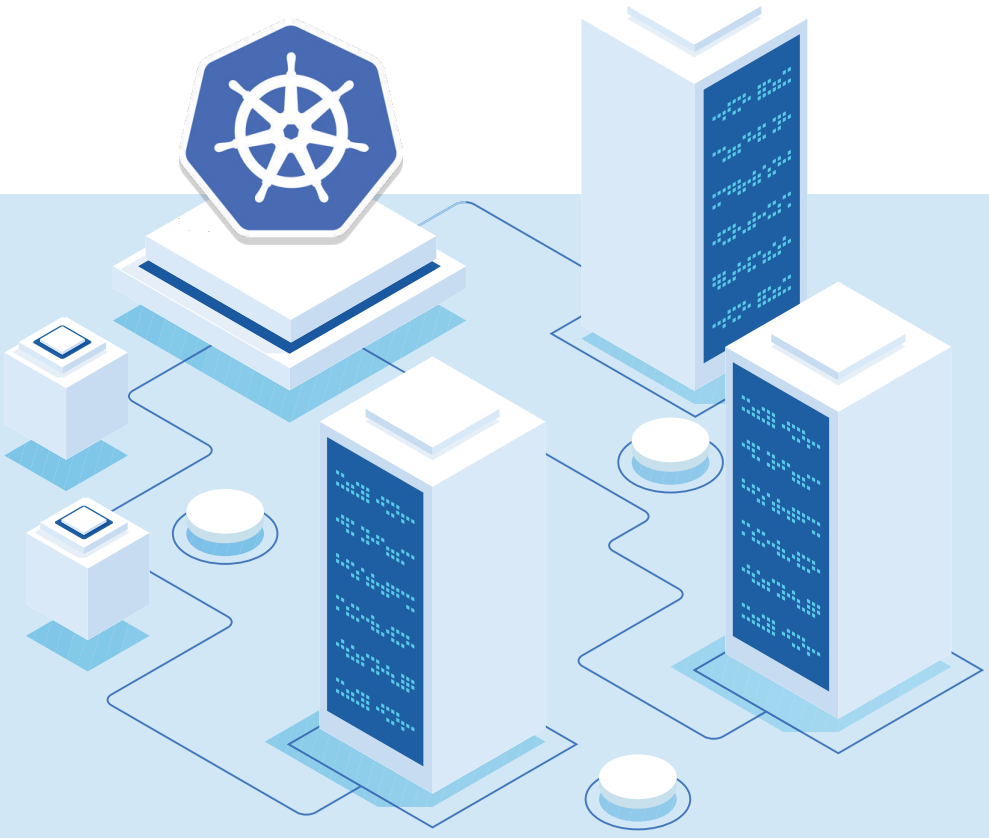
1. Executive Summary

Kubernetes has become the preferred container orchestration engine for automating deployment, scaling, and management of containerized applications. However, it does come with its own set of challenges commonly cited by enterprises – lack of experience and expertise; stakeholder alignment; meeting security and compliance requirements; limited visibility into resources and more.

The smart approach to overcome these challenges would be to have a framework to assess your current Kubernetes capabilities, set the right goals, and have a clear roadmap to achieve those goals. With the experience of multiple enterprise Kubernetes deployments, CloudIQ's team of certified Kubernetes administrators and solution architects have developed this simple Kubernetes maturity scoring model.

The scoring methodology focuses on 3 main areas for assessing maturity – Architecture, Capabilities, and Operations. There is a strong emphasis on cloud-native patterns like GitOps and git pull-request based patterns that enable zero-touch deployment. The assessment provides you a score on a scale of 1-5 showing your maturity level for each of the focus areas. (1 being the lowest and 5 being the highest score).

Our Kubernetes Maturity Assessment Checklist gives you a high-level understanding of where you are in your Kubernetes adoption journey, identify the gaps, where to focus and why.



2. Kubernetes Maturity Assessment Checklist

2.1 Architecture

Current Deployment Approach	Tools	Ratings
Right sizing clusters		
Multi-cluster for redundancy (HA DR)		
Availability zones vs multi-region clusters		
Naming conventions & standards for deployment (k8s and apps)		1=None; 5=NC for NS, Tags, Requests/limits
Namespace standards, Blue green in single vs unique cluster for each env.		
Create NS, then deploy services against prescribed standards		1 =Manual; 5 =using Harness with zero touch
Ingress-egress, image build, other k8s resources		
Automation & Pipeline Strategy		
Pipeline to match dependency of K8s deployment model		
Config Map settings from Pipeline	Salt, Ansible or any advanced CI/CD tool set (Harness, GH actions)	1 =Manual; 5 =using Harness with zero touch
Resources Limits, Quotas (either per deployment or at NS level)	Salt, Ansible or any advanced CI/CD tool set (Harness, GH actions)	1 =Manual; 5 =using Harness with zero touch
CD for container images	Salt, Ansible or any advanced CI/CD tool set (Harness, GH actions)	1 =Manual; 5 =using Harness with zero touch
Ingress policies, NS level RBAC	Flux, Argo CD or any GitOps based tool	1 =Manual; 5 =using flux with git pull
Networking models (private, CNI with Network policy engines) Service Mesh	Cilium, Calico Ent. For Network policies. Linkerd	1 =no mTLS or Network policies 5 =Np controlled in Flux, with modular service mesh
Compute capacity planning	Sysdig +KEDA for HPA scaling	
App onboarding	CI/CD tools (Harness.io, GitLab etc.)	
Build confidence	Once standards are setup, pipeline pillar files feed to CI/CD	1=Manual;5=Terraform enterprise with git Pull-request based approval
IaC strategy	Terraform, CF, Pulumi	1 if using Basic version 5 = if using these with Ent or Harness type CI/CD tools

2.1 .2 Best Practices Adoption (EA)

Best Practices Adoption (EA)	Tools	Ratings
Self-Healing	Probes, KEDA, Application retry patterns on failures. Soft locks etc.	
Health Probes	Liveness and readiness probe (Strategy from EA team level)	
Environment as Code	Modularize IaC to create env based self-service modules	
Service Discovery (Internal and external)	Consul, Linkerd, Traefik Enterprise	
Manage Drift	Flux, Argo CD Ansible/salt stack	
Cloud Provider's Best practices		

2.1.3 Security Standards and Deployment Patterns

Security Standards & Deployment Patterns	Tools	Ratings (1=none; 5=Security rating by product. Different thresholds for different clusters and workloads)
IAM & RBAC Strategy		
IAM & RBAC policies as Infra as code (users & apps)		
GitOps enabled flow for IAM and RBAC policies		1=manuial;5=Yaml policies for K8s RBAC
Bootstrapping of above for new cluster deployments	Aquasec, Sysdig, Twistlock (Harness for CI/CD)	
Artifacts & Cluster Security	Aquasec, Sysdig, Twistlock (Harness for CI/CD)	
Node & Image Patching (repo - scanning and then hardening)	Aquasec, Sysdig, Twistlock (Harness for CI/CD)	
Container Image Build Process		

Security Standards & Deployment Patterns	Tools	Ratings (1=none; 5=Security rating by product. Different thresholds for different clusters and workloads)
Build and patch process (image hardening standards)	Aquasec, Sysdig, Twistlock (Harness for CI/CD)	
Visibility - Container (3 levels - at rest, deployment, at runtime), Node, Cluster Security	Aquasec, Sysdig, Twistlock (Harness for CI/CD)	
Static Analysis, IaC scan, k8s manifest scans, Vulnerability scan (at pre-commit level) and registry (for image scanning)	IaC - Synk or similar, Aquasec (Pipeline) (Trivy)	1=manuial;5=Yaml policies for K8s RBAC
MTLS, Micro Segmentation		
Using CIS benchmarks, NIST etc.	Aquasec, Sysdig, HashiCorp sentinel	
Zero Trust Architecture (practical approach)	Cilium/Calico for network layer. Service Mesh for MTLS, Private endpoints to restrict API server	1=No modern CI/CD as zero-touch becomes a blocker for app onboarding;5=zero-touch deploy based on prescribed, agreed upon standards, approvals.
For all vs based on data categorization and data flow		
Pre-approved architecture for repeatable deployments		
Auditing	Kube Audit, etc., going to SIEM and log Analytics tools; or at least an S3 bucket	1= if only info is collected; 5= if info can be easily accessed for investigation, Alerts etc.
In-house vs third party	Twistlock, Wiz.io etc.	

2.2 Capability Building

2.2.1 For Architecture, Deployment, Security and Operations

For Architecture, Deployment, Security & Operations	Tools	Ratings
Cluster Operation & Troubleshooting Playbooks	Markdown based auto-doc generators for technical doc on tools (e.g., terraform)	1=none; 5=automated tech docs and playbooks for each scenario
Observability using logs, metrics (Cloud, K8s, App layer)	Sysdig (Metrics), Datadog, Sumo logic etc. App dynamics, new relic for Java workloads	
KEDA	Requires custom tooling	
Cluster and Application scaling based on custom metrics KEDA etc.	Requires Tooling (sysdig can make OS metrics appear like Prom metrics)	
Other upcoming technologies	Harness.io	

2.2.2 Adopting Advanced GitOps and DevOps Practices

Adopting Advanced GitOps and DevOps Practices	Tools	Ratings
Monitoring		
Faster response to cluster upgrades, kube events, cloud throttling, security concerns etc.,		
DevOps and GitFlow based operations	Flux, Argo CD, Terraform Enterprise	1=if using none; 5=if using multi cluster setup in Flux

2.3 Operational Maturity

2.3.1 Proactive Operations

Proactive Operations	Tools	Ratings
Increase Velocity of app deployments	New age pipeline tools like harness.io	1=legacy tools with lots of plugins; 5=tools created post cloud/k8s boom
Capture efficiency of usage	Ortelius, Harness.io	tools that give visibility to choose between mono-poly repo
Pod density to node size	Custom metrics	
Cost Tracking and optimization	Kubecost, Sysdig	
Faster image building	Harness (If starting new) or spinnaker	1= if using more than 5 tools; 5 =if using 1 tool to do CI/CD for container image hardening and builds

2.3.2 Reactive Operations

Reactive Operations	Tools	Ratings
Issue Detection	Sysdig, Datadog, Sumo logic etc.	
Alert and incident matrix	CloudIQ's list of what to monitor in K8s	
Metrics Signals matrix (sysdig includes golden metrics etc. OOB)	Sysdig covers most	1=if using none; 5=if using multi cluster setup in Flux

2.3.3 Operations

Operations	Tools	Ratings
Reduce RCA, detection response	Integration of ITSM with Service discovery and AGILE tools (need an Agile practice)	
Alert consolidation and correlation to incident management solutions	AI Ops with well-defined incident management strategy.	
Practical and prescriptive approach to Onboarding	List of approved services in categories of K8s (e.g., Web/API with data sec tier 2 will go to cluster X. Security team knows what to approve.	1=if using none; 5=if using multi cluster setup in Flux

3. Conclusion

Like we mentioned earlier the smart approach to overcome Kubernetes challenges would be to have a framework to assess your current Kubernetes capabilities, set the right goals, and have a clear roadmap to achieve those goals. This Kubernetes Maturity Assessment Checklist was built to help organizations for this purpose.

We hope this checklist helped you understand where you are in your Kubernetes adoption journey, identify the gaps, where to focus and why.

If you would like us to take you through the assessment, please feel free to reach out to us.

sales@cloudiq.io +1 (206) 203-4151 www.cloudiq.io

Gold Microsoft Partner

Microsoft

Kubernetes CERTIFIED Service Provider

Cloud Architects from CloudIQ will be happy to work collaboratively with your team to complete the assessment and provide the summary of findings and recommendations based on our Kubernetes Maturity Assessment.

We don't use automated scripts that generate templated reports unlike a traditional infrastructure assessment. Each assessment is unique to a client environment mostly because of different Kubernetes addons and integrations.

Kubernetes Training to Mature your Kubernetes Practice

CloudIQ's 2-day hands-on workshop is designed to give your team the opportunity to skill-up and learn Kubernetes design, deployment, and management. The sessions are guided by CloudIQ's Kubernetes platform engineers and will focus on imparting applicable knowledge through in-depth exercises.

Training Levels	High Level Topics Covered
Basic 4-8 Hour workshop, that can either be developer or operator focused. The workshop can either be taken online or in person with hands-on labs/demos included.	• Core Concepts in Kubernetes • Exercise/demo for developers ○ Docker deployment ○ Pushing docker images to a registry ○ Deployment on AKS / EKS / GKE ○ Ingress and service mesh deployment (Istio/Traefik/Linkerd) ○ KeyVault for Secrets Management ○ KMS or Hashivault for secret management with workload identity ○ Deployment using Helm • Exercise/demo for operators ○ Cluster creation ○ Terraform templates ○ Monitoring, Security, Cost & Governance
Intermediate 2 Days hands on workshop recommended to be taken in person with focused topics for developers on day 1 and for operators on day 2 with hands-on labs.	• Kubernetes Adoption Journey • Containerize, package, and deploy using Helm • Application & Developer security patterns • Challenges to traditional IT with Kubernetes and Cloud • Security, Identity & Network Design • AKS / EKS / GKE Pre-provisioning, provisioning, and post provisioning
Advanced Offline workshop that covers advanced Kubernetes topics with PCI compliance, network policy, continuous deployment at the operator level. Available as booklet or GitHub repo	• Customer scenario with PCI compliance, Hot upgrades etc • Network & Security Policies using Dataplane V2 • Admission controllers for Kubernetes Policies • Continuous Deployment for K8s at Operator level • Operations, including cluster upgrades • E2E GitOps based management for AKS / EKS / GKEs • Security, Governance & Cost • Operational checklist • GitOps based Kubernetes

Enroll in this hands-on workshop

Equip yourself with comprehensive Kubernetes skills to automate deployment, scale, and manage containerized applications.

sales@cloudiq.io +1 (206) 203-4151 www.cloudiq.io

Gold Microsoft Partner

Microsoft

Kubernetes CERTIFIED Service Provider