

CLOUD ASSESSMENT FRAMEWORK

Our approach is based on these 5 pillars and work with Agile principles:

RKON's Cloud Operating Model stems from AWS and Microsoft Azure's well architected frameworks. It's an application-centric approach that continually reevaluates cloud resources. Working with strategic and tactical inputs, we provide our customers with an application roadmap so their business can rely on IT to provide cost effective, scalable, secure, and reliable solutions.

Whether an organization is new to cloud, already has a small footprint, had a false start in their cloud journey, or were "born in cloud", RKON applies the same approach to align IT with business objectives. The key focus is mapping workloads to these business objectives, understanding the impact of workload outages to the business, and developing partnerships with the business through open lines of communication and a commitment to excellence.



Cost Optimization

The ability to run systems to deliver business value at the lowest price point.



Operational Excellence

The ability to run and monitor systems to deliver business value and to continually improve supporting processes and procedures.



Reliability

The ability of a system to recover from infrastructure or service disruptions, dynamically acquire computing resources to meet demand, and mitigate disruptions, such as misconfigurations or transient network issues.



Performance Efficiency

The ability to use computing resources efficiently to meet system requirements and to maintain that efficiency as demand changes and technologies evolve.



Security

The ability to protect information, systems, and assets while delivering business value through risk assessments and mitigation strategies.

Cloud Maturity Model

Base

Add-Ons

Cloud Optimization

Operational Excellence

Reliability

Performance Efficiency

Security

Choose the correct resources	Optimize build and release processes	Design for business requirements	Establish baselines	Plan resources and how to harden them
Setup Budgets and cost restraints	Understand operational health - Infrastructure Monitoring	Design for failure	Run load and stress testing	Automate and use least privilege
Dynamically allocate and deallocate resources	Rehearse recovery and practice failure - Backup/Restore	Drive automation	Identify bottlenecks	Classify and encrypt data
Optimize workloads, aim for scalable costs	Embrace continuous operational improvement	Design self-healing	Identify improvement opportunities with resolution planning	Monitor system security, plan incident response
Continuously monitor and optimize cost management	Use loosely coupled architecture (micro-services when available)	Design for scale-out	Date driven approach, monitor current workload health, troubleshoot issues - Infrastructure Monitoring	Identify and protect cloud assets
		Observe application health	Run load and stress testing	Protect against code-level vulnerabilities
			Performance testing	Model and test against potential threats