

Kubernetes Readiness Guide

Interpreting Your Assessment Score

BY HOWARD M. COHEN



RESEARCH SPONSOR



Table of Contents

Foreward from the Author 5

Kubernetes Primer 6

 Trends in Kubernetes 6

 Benefits of Kubernetes 6

Deploying Kubernetes 8

 Organizational Factors..... 8

 Technology Factors 9

The Redmond Intelligence Kubernetes Readiness Model..... 10

 The Challenge is in the Culture..... 10

 Organizational Readiness..... 12

 IT Readiness 12

 Application Readiness 12

 DevOps Readiness 12

 How the Readiness Assessment Tool Quantifies Metrics and Dimensions 13

Interpreting Kubernetes Readiness Scores 14

 Other Possible Patterns.....14

 Scenario 1.....14

 Scenario 2..... 15

Summary..... 16

REDMOND INTELLIGENCE COPYRIGHT STATEMENT

© 2021, Redmond Intelligence and/or its affiliates. All rights reserved. Unauthorized reproduction is forbidden. Information is based on resources available during the time of preparation of the report and believed to be reliable. Opinions in this report are subject to change without notice. Redmond Intelligence Solution Spotlight, Redmond Intelligence Best Practice Report and Redmond Intelligence Research Report are trademarks of Redmond Intelligence. All other trademarks are the property of their respective companies. For additional information, go to redmondintelligence.com.

About the Author

Senior Resultant **Howard M. Cohen** is a 35+ year executive veteran of the Information Technology industry who today writes about a wide variety of applied technologies and technology strategies. He's a frequent speaker at IT industry events that include Microsoft Inspire, Citrix Synergy/Summit, ConnectWise IT Nation, ChannelPro Forums, Cloud Partners Summit, MicroCorp One-On-One, and CompTIA ChannelCon. He frequently writes reports and hosts/presents webinars for many vendors and publications.



Connect with Howard at hmc@hmcwritenow.com and review his portfolio at www.hmcwritenow.com.

Foreword from the Author

This guide accompanies Redmond Intelligence's Kubernetes Readiness Assessment, an online survey designed to help respondents determine how well prepared their organizations are to leverage the advantages of Kubernetes in managing application delivery in their enterprise. When you complete the online questionnaire, the Assessment Tool immediately provides you with scores that quantify your organization's readiness.

This Readiness Guide provides a primer on Kubernetes and its role in managing and orchestrating cloud-native applications built using microservices in containers, an explanation of the Readiness Model, and tips for interpreting your assessment scores. We recommend that you read this guide before taking the assessment so you are prepared to interpret the scores displayed at the end of the questionnaire; however, the guide and the online tool can both be used independently, so you can work with them in either order.

THE REASON FOR A KUBERNETES READINESS ASSESSMENT

Those using the online assessment are contemplating the deployment of Kubernetes in their application delivery environment. Some of these organizations will have no experience with Kubernetes; others may have executed or are planning to execute a proof-of-concept project but need additional guidance before proceeding. Others are already using Kubernetes but want to expand their use of it, or use it in a different way.

At Redmond Intelligence we regularly communicate with a wide range of people and work with organizations that are considering modernizing their application delivery methods as they convert from traditional monolithic programming to cloud-native container-based architecture. They all have questions about where to focus their efforts in getting ready to fully embrace Kubernetes; the Readiness Assessment provides answers for many of these questions.

THE VALUE OF A KUBERNETES READINESS ASSESSMENT

The success factors built into this Kubernetes Readiness Model teach you the most common and most fundamental areas requiring attention in preparation. Note that the online Assessment Tool, when it displays your scores, will also display the average score of all other people who took the assessment. That way you can look at the state of your readiness in isolation or you can compare your strengths and weaknesses to those of other organizations. Thank you for reading this Readiness Guide and using the Kubernetes Readiness Assessment Tool. I trust you will find both useful.

Howard M. Cohen

Contributing Analyst
Redmond Intelligence

Kubernetes Primer

TRENDS IN KUBERNETES

According to the 2020 Cloud Native Computing Foundation (CNCF) global survey of the container management marketplace, 83 percent of the respondents are running Kubernetes in production (a huge jump from 58 percent in 2018). In the same survey, 92 percent of respondents are using containers in production, a 300 percent jump from the 23 percent in the CNCF's first survey in 2016.

Given that Kubernetes was first open sourced seven years ago in 2014, the sheer growth in use of the platform is impressive. The VMware State of Kubernetes 2020 Survey reports that two-thirds of users are still only using it on-premises despite Kubernetes being clearly designed for cloud use.

What are users looking for? According to VMware, 56 percent are seeking improved resource utilization, while 53 percent hope to shorten their software development cycles.

A third are pursuing public cloud cost reduction, a considerably lower proportion than usual.

Everybody in most enterprises wants a seat at the Kubernetes table, including development teams, infrastructure teams, cloud architects, IT operations, and business application owners.

Kubernetes is hot.

Benefits of Kubernetes

The need for increased application resiliency and reduced downtime led to a dramatic shift from traditional monolithic software development in which all required functions were packaged into a single monolithic block of code. In that architecture, failure of any function could lead to the entire application crashing, rendering all users helpless.

Replacing the monolith, developers are now leveraging microservices packaged into containers along with all libraries and other resources required to run them. Each microservice is called only when needed. Should a microservice fail it is immediately discarded and re-instantiated. As a result, application operation simply continues with nobody aware that anything went wrong. This brings tremendous resilience.

It didn't take long for developers to reason that having all these containers traversing the network to arrive wherever in the cloud they were needed created a tremendous need for management and orchestration of all this movement. This gave rise to the development and rapid adoption of the Kubernetes container management platform.

Kubernetes delivers a framework for running distributed systems with incredible resilience. Application scaling and failover are handled, as well as much more. As explained by the platform's primary website, [Kubernetes.io](https://kubernetes.io), Kubernetes provides:

▶▶ **Service discovery and load balancing.** Kubernetes can expose a container using the DNS name or using their own IP address. If traffic to a container is high, Kubernetes is able to load balance and distribute the network traffic so that the deployment is stable.

▶▶ **Storage orchestration.** Kubernetes allows you to automatically mount a storage system of your choice, such as local storage, public cloud providers, and more.

▶▶ **Automated rollouts and rollbacks.** You can describe the desired state for your deployed containers using Kubernetes, and it can change the actual state to the desired state at a controlled rate. For example, you can automate Kubernetes to create new containers for your deployment, remove existing containers, and adopt all their resources to the new container.

▶▶ **Automatic bin packing.** You provide Kubernetes with a cluster of nodes that it can use to run containerized tasks. You tell Kubernetes how much CPU and memory (RAM) each container needs. Kubernetes can fit containers onto your nodes to make the best use of your resources.

▶▶ **Self-healing.** Kubernetes restarts containers that fail, replaces containers, kills containers that don't respond to your user-defined health check, and doesn't advertise them to clients until they are ready to serve.

▶▶ **Secret and configuration management.** Kubernetes lets you store and manage sensitive information, such as passwords, OAuth tokens, and SSH keys. You can deploy and update secrets and application configuration without rebuilding your container images, and without exposing secrets in your stack configuration.

Deploying Kubernetes

The decision to deploy Kubernetes is simple and straightforward. When you have so many containers in production that you cannot manage them effectively your best strategy is to deploy Kubernetes to manage, schedule, and orchestrate those containers.

The larger decision is whether or not to transition to cloud-native software development and deployment in the first place. As the 300 percent growth rate since 2016 reported by CNCF shows us, more and more companies are making this decision for their enterprise. The sudden increase in need to do more with less created by the pandemic drove more and deeper Kubernetes adoption, according to survey results released by Dataworx this year. In the survey of 500 IT professionals and managers, 68 percent of respondents said they had increased Kubernetes utilization. Fifty-nine percent did so to accelerate deployment frequency, while 54 percent cited the need to increase automation and 46 percent sought to reduce IT costs.

Kubernetes adoption is complex. Onboarding requires significant training effort throughout an organization, including the need to develop a new mindset about IT in general. The various tools and components that comprise a Kubernetes deployment create a wide variety of security challenges and vulnerabilities. Those who seek to create cost reductions will need to demonstrate patience as that will not necessarily come until the entire implementation has had a chance to mature.

ORGANIZATIONAL FACTORS

The opportunities created by Kubernetes must be tempered with the complexity and attendant challenges posed. The shift from traditional monolithic development to cloud-native may appear to some to be a case of fixing something that isn't broken. But the performance and scalability improvements available more than justify the investment, as do the eventual cost savings.

The DevOps team will play a major role in the design, development, deployment, and ongoing ability to extract real value. In fact, organizations considering Kubernetes who do not yet have a DevOps team should probably delay their decision.

When we talk about Kubernetes deployment we need to recognize that we're talking about continuous deployment. Perhaps the greatest advantage created by the move to cloud-native microservices-in-containers development is the ability to rapidly release new iterations of applications to achieve continuous improvement through continuous deployment.

TECHNOLOGY FACTORS

Up until this point we have been referring to “deployment” as the initial process of downloading Kubernetes and getting started.

However, a “deployment” is a very specific activity in the process of using Kubernetes. The Kubernetes official site discusses Kubernetes deployments:

“Once you have a running Kubernetes cluster, you can deploy your containerized applications on top of it. To do so, you create a Kubernetes Deployment configuration. The Deployment instructs Kubernetes how to create and update instances of your application. Once you’ve created a Deployment, the Kubernetes control plane schedules the application instances included in that Deployment to run on individual Nodes in the cluster. Once the application instances are created, a Kubernetes Deployment Controller continuously monitors those instances. If the Node hosting an instance goes down or is deleted, the Deployment controller replaces the instance with an instance on another Node in the cluster. This provides a self-healing mechanism to address machine failure or maintenance.”

In the introduction to the Deployments page, Kubernetes.io adds:

“A Deployment provides declarative updates for Pods and ReplicaSets. You describe a desired state in a Deployment, and the Deployment Controller changes the actual state to the desired state at a controlled rate. You can define Deployments to create new ReplicaSets, or to remove existing Deployments and adopt all their resources with new Deployments.”

To those new to Kubernetes even these simple introductory explanations give us insight into the complexity to come. Fortunately, much of the specification and declaration required in the Kubernetes environment is managed using simple command-line programs such as `kubectl`, and the preferred scripting language is Yet Another Markup Language (YAML).

Kubernetes runs in a “cluster” of server-class machines. These may be physical servers, or virtual machines, or a combination of both. The IT organization will need to work closely with the DevOps team to properly configure and build scalable clusters.

The key takeaway is to recognize that we are creating more and more complex cloud and multi-cloud environments, so it only follows that the scheduling and orchestration of software running in these environments must be similarly complex.

The Redmond Intelligence Kubernetes Readiness Model

The 300 percent increase in the use of containers and the large number of organizations using Kubernetes in production clearly illustrate the accelerating trend toward cloud-native software development.

Classic monolithic software development put all code into one block. All required functions could be found somewhere within that mass of code. It becomes immediately apparent how difficult it was to trace errors, but what isn't immediately obvious becomes so when any process in the program fails or throws an error. The likelihood is that the entire program will grind to a halt and need to be repaired and restarted.

This lack of resilience drove the development of microservices, each designed to perform a specific function and each packaged in a container along with all resources required to run it, including libraries, binaries, and more. This is considered "cloud-native" because these containers readily travel wherever in the network they are required. As multi-cloud continues to grow, cloud-native software becomes even more valuable.

Should any microservice malfunction, it is immediately discarded and reinstantiated without any pause in overall processing. The software continues to run, operations continue to work, as do people.

The challenge comes as these containers proliferate within an environment. Very simply it becomes more and more difficult to manage all of them and orchestrate their proper arrival and processing where they are needed. The solution developed to perform this management and orchestration of containers is Kubernetes.

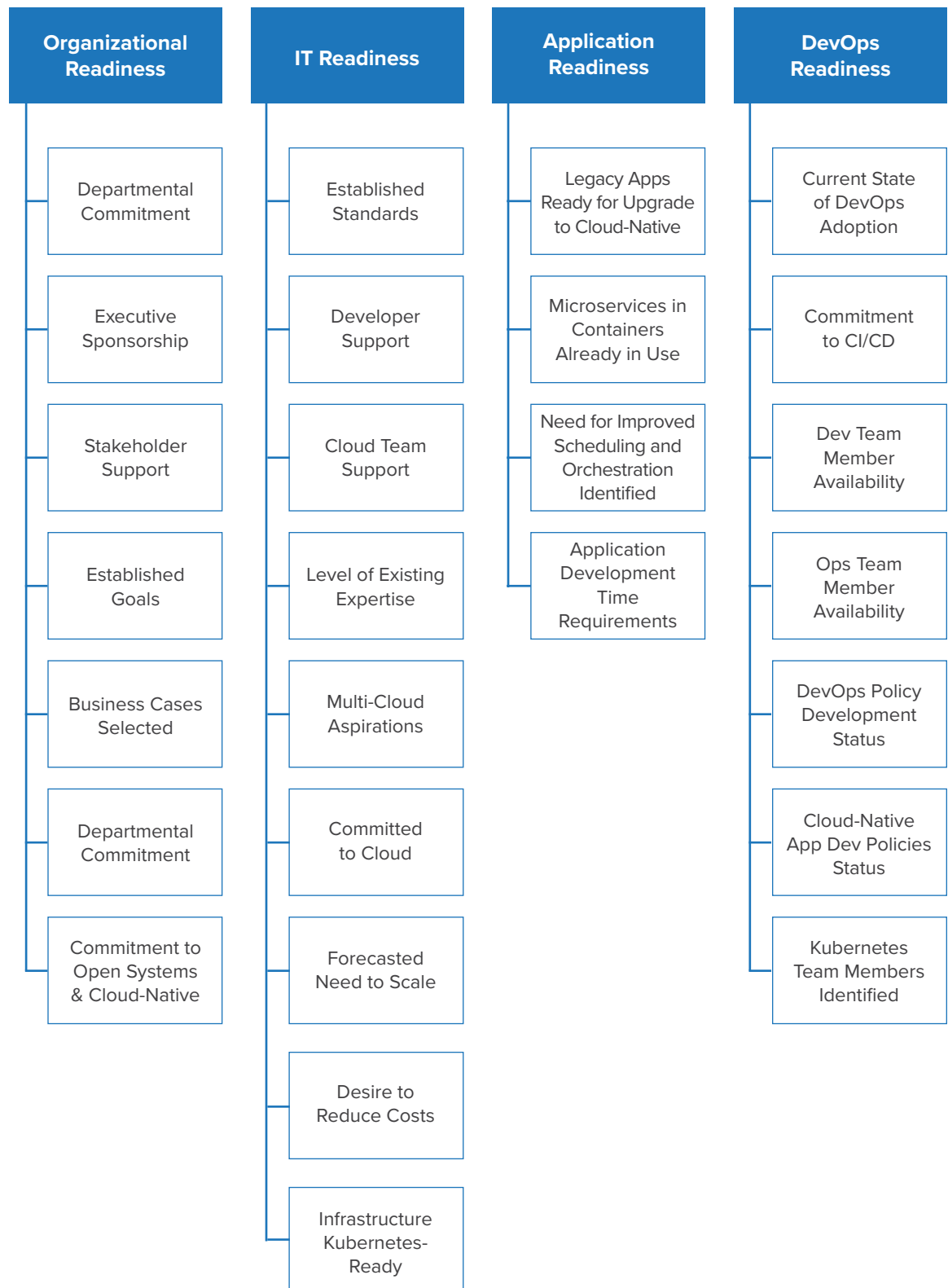
THE CHALLENGE IS IN THE CULTURE

Much like DevOps, the use of microservices in containers and Kubernetes requires a change of mind-set, a paradigm shift, changes which impact the culture of the organization. All are designed to achieve the goal of continuous improvement through continuous deployment (CI/CD), which requires broad cooperation to achieve the desired velocity that enables software updates to be delivered rapidly, often multiple times per day.

This Kubernetes Readiness Model was designed to gauge how well prepared each relevant department in the organization is. Changes can be anticipated in everything that touches IT, from software development to computer and communication infrastructure operations along with most or all lines-of-business (LoB) and even the executive suite. In this last case, lack of enthusiastic buy-in and support at the executive level will almost always prevent any Kubernetes project from ultimately succeeding.

With this need for broad adoption in mind, the survey is structured into four segments (See Figure 1).

Figure 1: The dimensions and metrics of the Kubernetes Readiness Model.



Here's a general description of each dimension, with a few examples of the questions that the Assessment Tool asks for each dimension and its metrics in order to quantify readiness for implementing Kubernetes. You'll see the connection between the questions asked and the technology and organizational factors described above.

Organizational Readiness

Deploying Kubernetes represents a significant change in the way applications run in your enterprise, and there is significant complexity in making this change. Setting everyone's expectations accordingly is critical. This includes the stakeholders who are invested in the overall success and fiduciary health of the enterprise, the executives charged with management of that achievement, and all the operational and development people who will now be "fixing something that isn't broke." It is absolutely necessary that organizational goals are stated clearly and everyone becomes personally invested in the success of the project.

IT Readiness

Kubernetes is not plug'n'play. Many changes will be required before any IT environment originally designed to support monolithic software will be ready for cloud-native code and processes. Before those changes can be made, qualified personnel must be in place. This can be accomplished either through re-skilling, training, or recruiting and hiring. Also, someone will need to focus only on this one project for the duration of the deployment, and probably well beyond, to assure full and accurate adoption. While many current Kubernetes implementations are deployed on-premises, it is anticipated that the organization is now or will soon be capable of moving applications and workloads to the cloud or multi-cloud.

Application Readiness

The whole purpose of cloud-native, microservices in containers, and Kubernetes is to support the faster, more efficient, more effective execution of application software. If there are no applications there is no reason to consider Kubernetes. Organizations will start new application development projects in the cloud-native framework, but there are also many monolithic applications already operating that are candidates for improvement via conversion to cloud-native code. In fact, the installation of Kubernetes doesn't become cost-effective until there are so many applications generating so many containers that proactive management, scheduling, and orchestration becomes necessary.

DevOps Readiness

Kubernetes is not an application development platform, but it is the tool that enables operations to effectively run applications developed in cloud-native form. This makes the unification of developers and operators that is the core of DevOps essential for the success of these applications. Both sides of the team must be fully familiarized with what they will be managing as they themselves make the transition from a monolithic mindset to a cloud-native one.

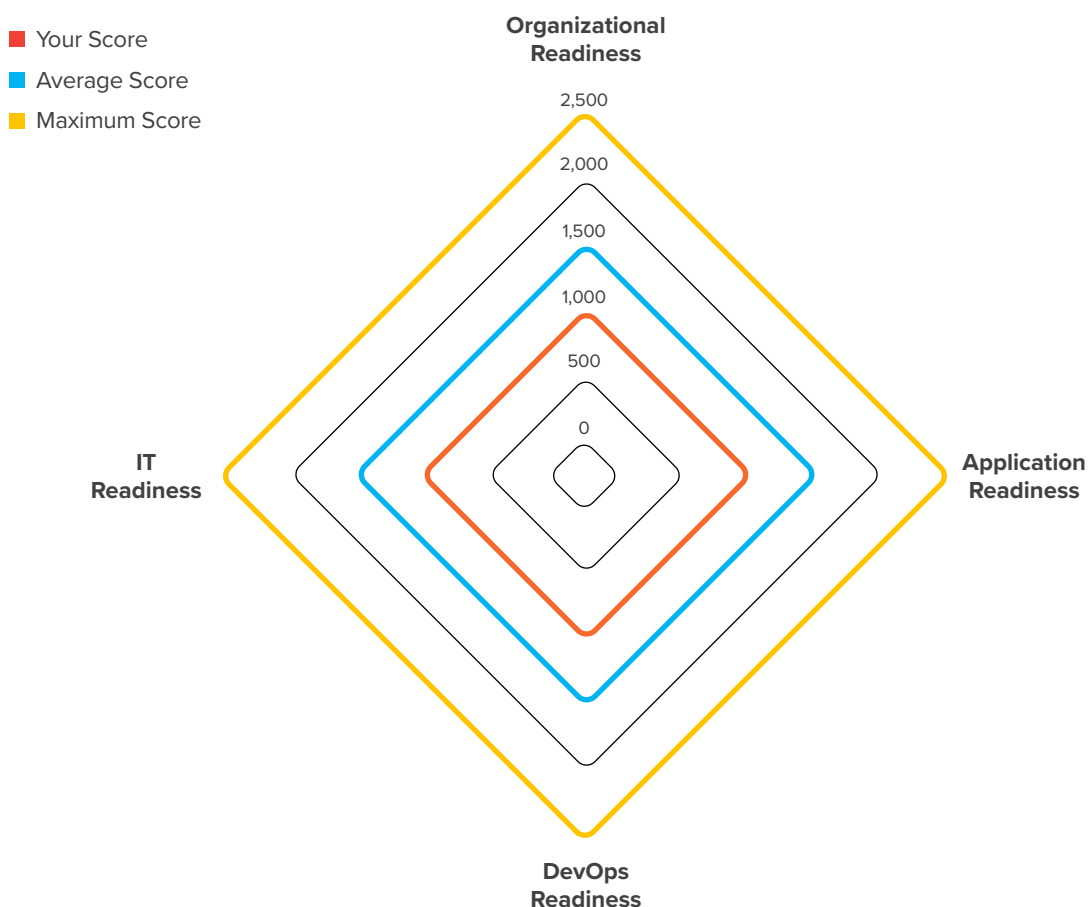
HOW THE READINESS ASSESSMENT TOOL QUANTIFIES METRICS AND DIMENSIONS

When you select answers to questions in the Kubernetes Readiness Assessment Tool, the score for particular dimensions is calculated. For many questions, the multiple-choice answer you select determines the score for that question.

When you take the online assessment, resist the urge to inflate your score by answering the questions based on aspirations for the future. For your assessment to be accurate and useful for Kubernetes planning, you should answer all questions as accurately and honestly as you can.

The highest score for each single dimension is 25. Multiplying 25 by the four dimensions yields 100 as the highest possible total score. At the end of the assessment, the Readiness Assessment Tool displays your scores per dimension (out of 25) and overall score (out of a possible 100), plus the average dimensional and overall scores of all respondents. That way, you have a context for determining whether your organization is ahead of or behind the curve. You will also see the average for your industry and company size.

Figure 2: An example of the radar chart produced by the Assessment Tool.



If your organization is completely prepared to leverage Kubernetes today (and may be doing so), your score across all dimensions might tally to 100, but that's rare for someone taking the assessment (who is likely looking to implement Kubernetes or thinking about it); most overall assessment scores will fall between 40 and 70. An overall score of 50 is a reliable watershed benchmark. Above that, users should most likely proceed with Kubernetes deployment and further preparation can be successfully executed concurrently during the implementation. Below that, there are likely improvements that should be made to use-case commitments, data management, and technical infrastructure before you undertake a Kubernetes implementation. However, we recommend looking at the score in each dimension separately to evaluate readiness, as this will provide more reliable measures.

Interpreting Kubernetes Readiness Scores

As mentioned earlier, the online tool will show you your scores after you complete your session, as well as an average of scores across all assessment respondents. That way, you can “benchmark” your scores against those of your peers at organizations that are also contemplating an investment in Kubernetes. It will also provide information in a graphical form—the radar chart (see Figure 2 for an example).

Based on previous Redmond Intelligence research, we see certain patterns when it comes to Kubernetes data and analytics readiness. First, some organizations are not ready, across the board. This group deals mostly with time-tested monolithic applications or cloud-native applications involving fewer containers. They may not have a volume of containers building up and they haven't really thought through a move to Kubernetes, nor do they really need the scale and scheduling that Kubernetes provides. They will score low on the assessment.

OTHER POSSIBLE PATTERNS

Scenario One: The People are Ready but the Tech Is Not

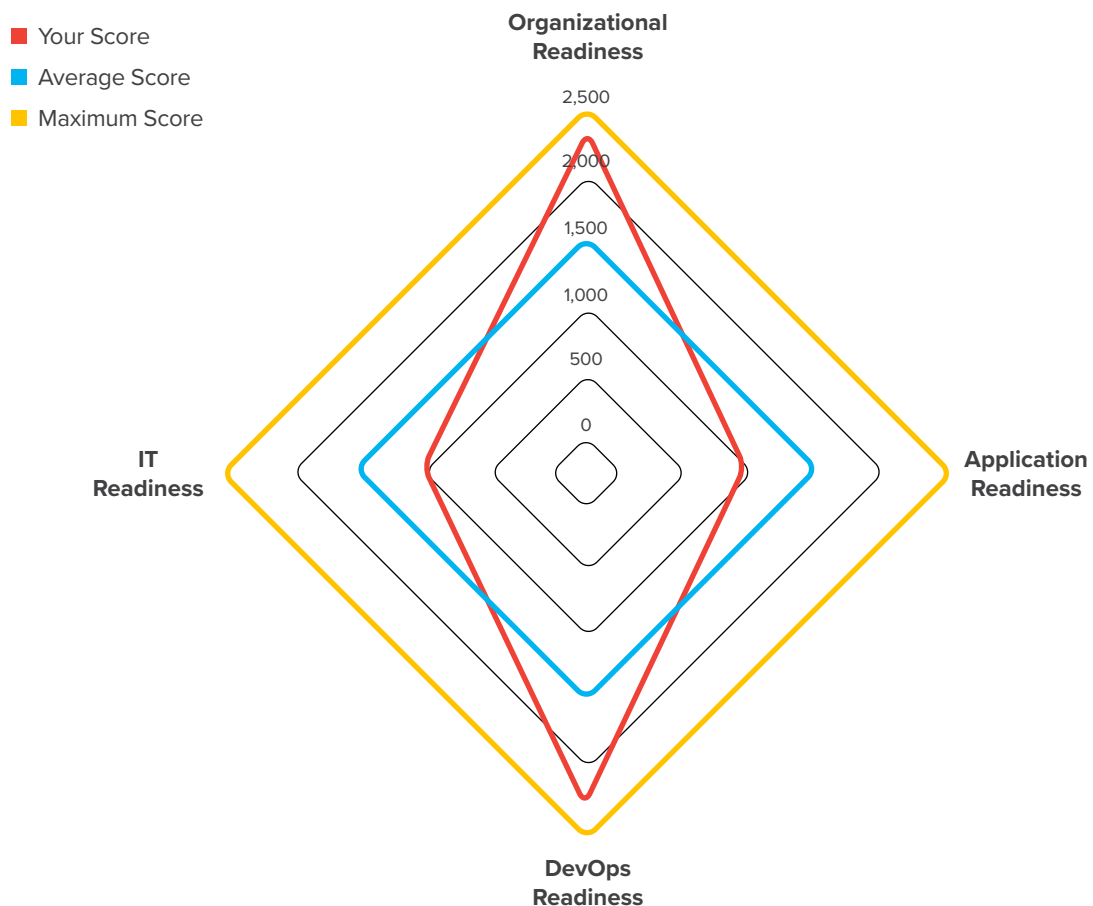
Technology personnel are constantly self-motivated and driven to continuously expand their skills. Given the growing popularity of cloud-native software built using microservices in containers and Kubernetes, it would not be surprising to find many members of any given IT team studying them and becoming prepared to use them. When they begin to identify applications that would benefit from running in a cloud-native framework they may run into a wall because their current IT infrastructure is completely designed to support monolithic applications and will need to be significantly reworked before it can support that framework.

The radar map will likely skew strongly toward Organizational Readiness and perhaps DevOps Readiness as well, but show weakness in IT and Application Readiness (See Figure 3). Much work will need to be done starting with careful assessment leading to the building of cogent business plans for creating the ability to support some cloud-native applications.

Recommendations: Those staff experts interested in bringing cloud-native technologies to their organization should start by thoroughly evaluating each legacy monolithic application running in the organization to determine which would benefit from conversion to cloud-native. The business plan they build must carefully define:

- The advantages to be gained from the conversion
- The cost in time and funds to make the conversion
- The cost to upgrade the hardware to support the new environment
- The personnel required to do it
- The likely return to the enterprise

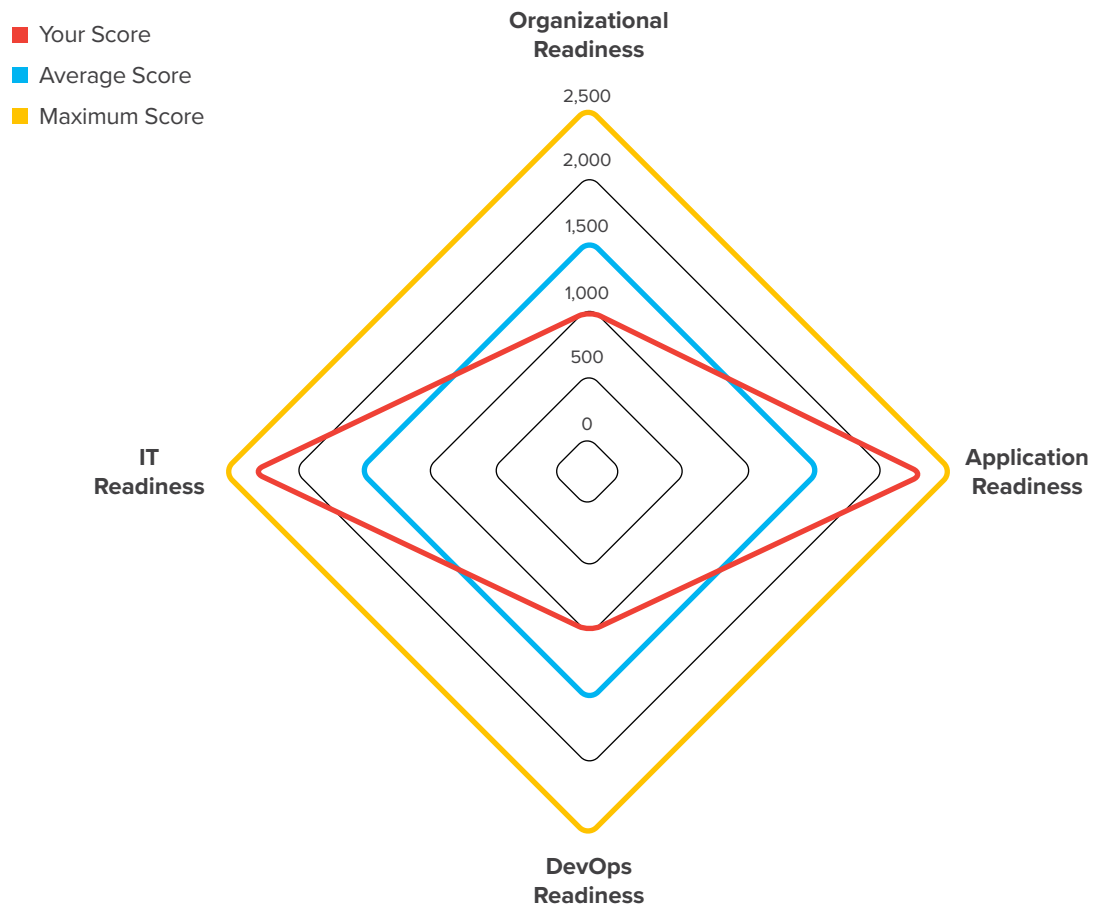
Figure 3: Scenario One where people are ready but tech is lagging.



Scenario Two: The Tech is Ready but the People Are Not

This radar map will fundamentally be the opposite of the first, skewing more to Application and IT Readiness and away from Organizational Readiness.

Care must be taken in approaching this situation. If people at the top of the organization are not enfranchised, no movement will take place and the monolithic applications will continue to run.

Figure 4: Scenario Two where the tech is ready but the culture is not

Even if there is enthusiasm at the top, a lack of skilled operators could result in a terrible, failed conversion that will discourage the user community and may not be recoverable.

Finally, organizations in this scenario face a challenge in being able to accomplish the overall cultural transformation that best enables cloud-native applications to thrive.

Summary

The Redmond Intelligence Kubernetes Readiness Assessment provides a quick way for your organization to assess its readiness for using Kubernetes to manage your cloud-native environment and to compare itself in an objective way to others. The assessment is based on the Redmond Intelligence Kubernetes Readiness Framework, which consists of approximately 40 questions across four dimensions. Although this assessment serves as a relatively coarse measure of readiness, we know you will find it useful.



SUSE

RESEARCH SPONSOR

SUSE is a global leader in innovative, reliable and enterprise-grade open source solutions, relied upon by more than 60% of the Fortune 500 to power their mission-critical workloads. We specialize in Enterprise Linux, Kubernetes Management, and Edge solutions, and collaborate with partners and communities to empower our customers to innovate everywhere — from the data center, to the cloud, to the edge and beyond. SUSE puts the “open” back in open source, giving customers the agility to tackle innovation challenges today and the freedom to evolve their strategy and solutions tomorrow. The company is headquartered in Nuremberg, Germany, and employs nearly 2000 people globally. SUSE is listed in the Prime Standard of the Frankfurt Stock Exchange.

**Learn more about Kubernetes
management at [SUSE.com](https://suse.com)**



Redmond Intelligence provides independent and objective research and advisory services to technology buyers and vendors. Written by technical subject matter experts, Redmond Intelligence reports dive into the details of information technology to provide actionable insights and concrete guidance. Projects range in scope from Solution Spotlights covering individual products to survey-based research reports to custom papers. For more information, visit redmondintelligence.com.