DataStax

Working and Winning with Kubernetes and Cassandra

3 Key Insights for
Enterprise IT Leaders

INTRODUCTION

Kubernetes, Cassandra, and the future of cloud-native development

The cloud has transformed enterprise technology in ways that nobody could have imagined a decade ago. Enterprise development teams have been challenged to adapt at lightning speed to building, deploying, and running cloud-native applications. They're learning new ways to work with distributed application architectures and containerized environments and are mastering the challenges of designing applications that run flawlessly at a massive scale.

Above all, success with cloud-native applications requires enterprises to make platform choices that will shape everything they do—for better or worse–for many years to come. Organizations face two critical questions when considering how to successfully build apps in the cloud:

1. How can development teams ensure that the applications they run in the cloud will perform at scale, minimize complexity and cost, and adapt easily to multi-cloud and hybrid cloud environments?

2. How can they deploy cloud-native technology stacks that keep applications and data management systems fully aligned and seamlessly integrated?

We believe the technology required to answer these questions is ready and waiting.

As we'll explain, Kubernetes and Apache Cassandra™ have already emerged as ideal platforms for cloud-native application development and cloud-native data management. And now, forward-thinking enterprise development teams are taking the next step: integrating Kubernetes and Cassandra into a single, comprehensive platform for building and running cloud-native, data-driven applications.

Let's look at why Kubernetes and Cassandra are so well-suited to the roles they play today and why they're even more powerful when they work together. We'll also explain why DataStax offers the best way to work with Cassandra and Kubernetes, by giving teams a faster and more confident path to success with their cloud-native data applications.

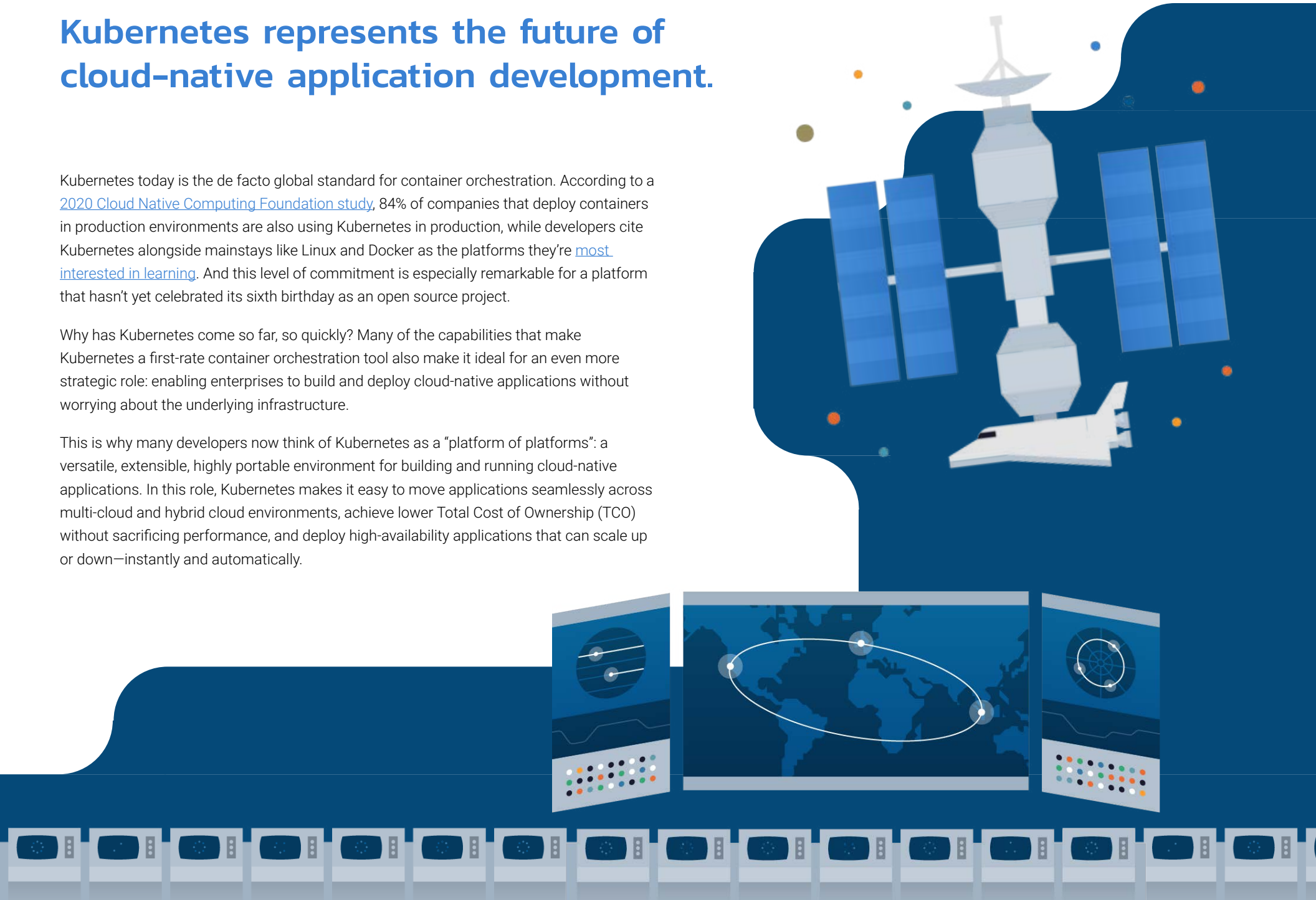
1

Kubernetes represents the future of cloud-native application development.

Kubernetes today is the de facto global standard for container orchestration. According to a 2020 Cloud Native Computing Foundation study, 84% of companies that deploy containers in production environments are also using Kubernetes in production, while developers cite Kubernetes alongside mainstays like Linux and Docker as the platforms they're most interested in learning. And this level of commitment is especially remarkable for a platform that hasn't yet celebrated its sixth birthday as an open source project.

Why has Kubernetes come so far, so quickly? Many of the capabilities that make Kubernetes a first-rate container orchestration tool also make it ideal for an even more strategic role: enabling enterprises to build and deploy cloud-native applications without worrying about the underlying infrastructure.

This is why many developers now think of Kubernetes as a "platform of platforms": a versatile, extensible, highly portable environment for building and running cloud-native applications. In this role, Kubernetes makes it easy to move applications seamlessly across multi-cloud and hybrid cloud environments, achieve lower Total Cost of Ownership (TCO) without sacrificing performance, and deploy high-availability applications that can scale up or down—instantly and automatically.



This role also positions Kubernetes as a game-changing technology for enterprise DevOps teams.

Making them far more productive and efficient, equipping them to do more with less, and above all, freeing them to focus on high-value tasks that create value for their businesses. Kubernetes accomplishes these goals by giving teams powerful new capabilities that are just as valuable whether they're migrating existing applications to the cloud, modernizing and re-platforming legacy apps, or building new applications for deployment in the cloud:

DEPLOYMENTS

Kubernetes handles the process of initiating, pausing, or rolling back container deployments. Once they're deployed, it ensures they're running in a desired state.

DISCOVERY AND PROVISIONING

Kubernetes automatically exposes containers to one another and to networked services and resources—and, when necessary, mounts persistent local or cloud storage.

LOAD BALANCING AND SCALING

Kubernetes ensures that containerized applications scale quickly, automatically, and reliably—and, when necessary, can scale back down again.

HIGH AVAILABILITY

Kubernetes will automatically monitor container health and performance and restart or replace failing containers.

But there's more to this story.

As teams continue to deploy bigger workloads and a wider variety of workloads using Kubernetes, they're identifying more strategic—and far more lucrative—advantages to cloud-native application development:

Working smarter.
Kubernetes liberates teams to focus on what matters—more frequent releases, higher quality, and closer alignment with the business—rather than container-counting drudgery.

Going bigger.
Kubernetes supports features, such as horizontal auto-scaling and rolling updates that open the door to sky's-the-limit scalability for containerized applications.

Seeing more clearly.
Visibility and control are significant challenges in an environment where thousands of containers are bumping elbows—or butting heads. Kubernetes captures a wealth of data that teams can use to understand what's happening within a containerized environment, suggest where and how to target optimizations, and how to troubleshoot problems.

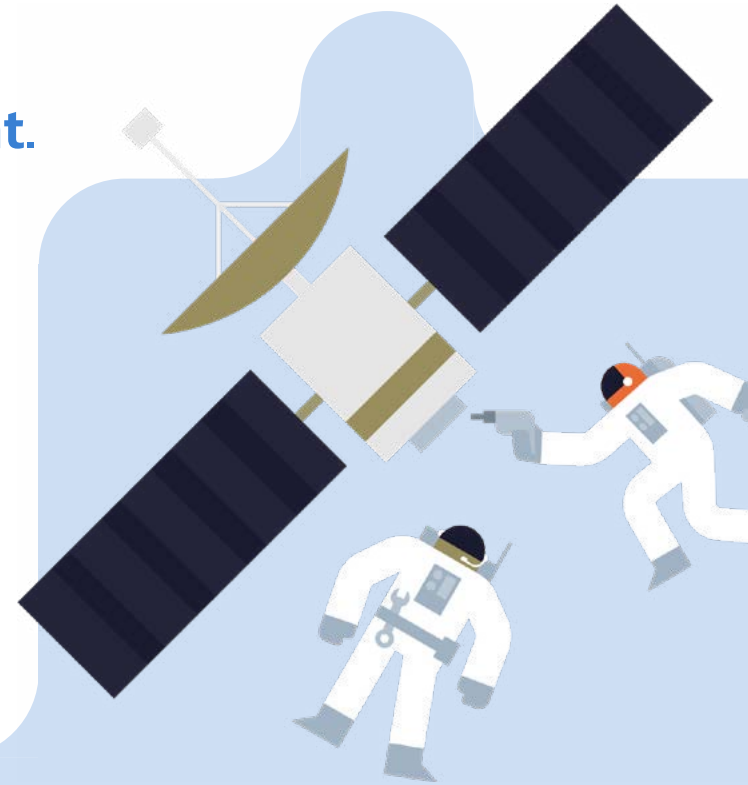
Doing more with less.
Kubernetes brings a great deal of intelligence to the task of running containerized applications across a given set of physical resources, which minimizes waste without introducing unacceptable risk. As we mentioned before, Kubernetes all but eliminates overprovisioning by scaling up and down—ensuring that an enterprise doesn't pay for cloud resources unless it really needs them.

Running longer.
Kubernetes load balancing and distribution services, and its redundancy capabilities, ensure that most failures don't even result in noticeable performance issues, let alone unplanned downtime. These capabilities alone are worth it when you consider the cost and consequences when a customer-facing application goes down.

It's clear that running applications and workloads on Kubernetes provides many advantages when running your stack in a multi-cloud environment. But, should you also run your database, with its complex operations and persistent, consistent data requirements, on Kubernetes? **The answer is a resounding yes.**

2

Cassandra represents the future for cloud-native, data-intensive application development.



You've heard it all before: Data is a modern enterprise's most valuable asset. So, data-driven applications are where it's at for developers. And while most businesses could do great things with the data they're sitting on right now, the tools they've used to work with data in the past hold them back and limit this vast potential.

Let's break down this problem into two parts: Why are traditional data management systems such a poor fit with cloud-native applications, and what can enterprises do about it?

THE PROBLEM

Business data is changing—your business database isn't. Relational database (RDBMS) technology was built for a world where business data looked and behaved in very specific ways. Cloud-native applications play by different rules, and that's where the trouble starts.

STATIC DATA STRUCTURES

Business data today doesn't stick to familiar, pre-defined structures. Instead, it flows from many different sources and is structured in a variety of ways—and very often, it's no longer structured at all. Customer data today, for example, may include call center transcripts, geo-spatial data, software usage telemetry, and dozens of other sources. There's gold in these mountains of data, but your RDBMS has no idea where to start digging.

RUNAWAY DATA GROWTH AND SCALABILITY

Your RDBMS is designed to scale, but you must be ready with server space and a storage array when it's time to grow.

Unfortunately, growth in the cloud is more like a Viking raid: You don't know it's here until it's kicking in your door and running amok in your office. And often, all it takes is one viral social post or media mention to light a fire under your business—and put your legacy data systems through a test they weren't designed to pass.

SCALABILITY GAPS

Scalability is one of the most important benefits an enterprise will get from any cloud migration or modernization effort. But when you pair scalable, high-performance cloud applications with legacy database systems that simply can't achieve similar scalability gains, the results are predictable—and invariably disappointing.

THE SOLUTION

Going cloud-native with Cassandra and NoSQL.

NoSQL databases have emerged as a go-to option for dealing with the realities of modern business data management and aligning an enterprise's data management systems with the demands of a cloud-native application environment.

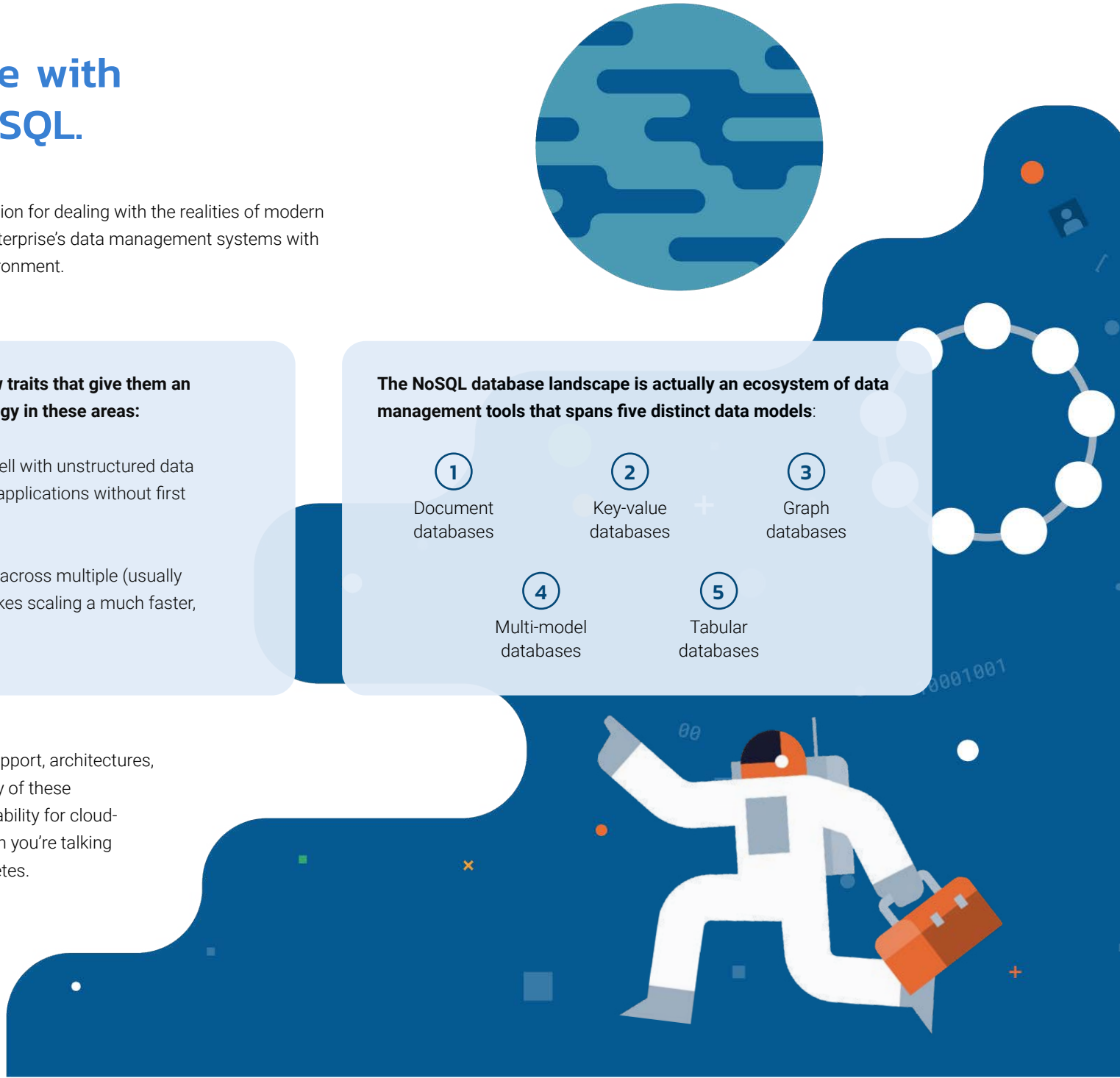
All NoSQL databases share a few traits that give them an advantage over RDBMS technology in these areas:

- A flexible schema that works well with unstructured data and allows developers to build applications without first defining a schema
- The ability to scale horizontally across multiple (usually commodity) servers, which makes scaling a much faster, easier, and cheaper process

The NoSQL database landscape is actually an ecosystem of data management tools that spans five distinct data models:

1. Document databases
2. Key-value databases
3. Graph databases
4. Multi-model databases
5. Tabular databases

NoSQL databases can also differ in their API support, architectures, data distribution models, and other areas. Many of these differences are essential when it comes to suitability for cloud-native application development, especially when you're talking about containerized environments and Kubernetes.



Cassandra stands out precisely because it's so well suited, on so many levels, for a modern, cloud-native application development environment:

Performance and latency.

In Cassandra, every node can perform all read and write operations rather than rely on a single node with read/write privileges to replicate data across a cluster. This approach gives Cassandra a significant performance edge, especially in terms of latency, and also contributes to reliability.

Openness.

Enterprises today are voting with their wallets to embrace open systems, open software, and open standards. Cassandra (which the Apache Foundation maintains) is a key part of the modern open source ecosystem. It supports a flourishing community and also sustains a profitable and fast-growing group of commercial open source vendors.

Availability.

Giving every Cassandra node read/write capability also supports faster data replication across the entire environment. Also, if a node fails, the system routes users automatically to the nearest functioning node, ensuring that applications keep running normally, preventing data loss, and ensuring the best possible user experience. And in many cases, Cassandra's self-healing capabilities can fix the underlying problem without manual intervention.

Scalability.

Cassandra is an excellent example of how NoSQL databases support fast, cost-effective growth by simply spinning up additional nodes in a cluster. This model is easy to automate in a cloud-native environment; it works effectively even at massive scale, and these scalability benefits extend across multi-cloud and hybrid cloud environments.

Multi-cloud performance may be the single most important requirement today for many enterprise teams—and it's an area where Cassandra truly stands out.

3

Kubernetes and Cassandra are an ideal pairing for building and running modern, cloud-native data apps.

At one time, Kubernetes wasn't a platform of choice for running databases or data management applications. It was difficult to run stateful data on Kubernetes, which often resulted in a single point of failure. On the apps side, that's not a big deal, but it causes challenges on the database side. Today, the truth about Kubernetes is very different. It now enables developers to run applications and databases side by side—creating a single, fully containerized technology stack, accessed via a shared control plane and leveraging a common management toolset. In addition, using the K8ssandra operator (described on page), persistent data is now possible on Kubernetes. And Cassandra plays a starring role in this success story.

First, keep in mind that Kubernetes and Cassandra share some valuable common traits:

Both benefit from thriving open source developer communities that support a high level of innovation and growth.

Both are designed for reliability and performance in distributed environments and to perform well at massive scale.

Kubernetes and Cassandra both offer linear scalability, allowing operators to add or remove hardware resources to stay on top of changing loads or data requirements and work well with the same development platforms and tools.

These shared features contribute to some powerful synergies when you run Cassandra on Kubernetes. For example, Cassandra handles scaling and replication with a peer-to-peer database architecture that scales linearly across cloud infrastructure. A peer-to-peer model requires only one Docker image and a single Kubernetes toolset to run the database, making development and deployment with containers and orchestration dramatically easier than databases with multiple node types.

Kubernetes works with Cassandra to provide enterprises a simpler, largely automated approach to distributed systems lifecycle management.

Resizing a Cassandra cluster becomes as simple as telling Kubernetes what you want (programmatically,) and letting Kubernetes manage the details.

HERE'S THE BOTTOM LINE:

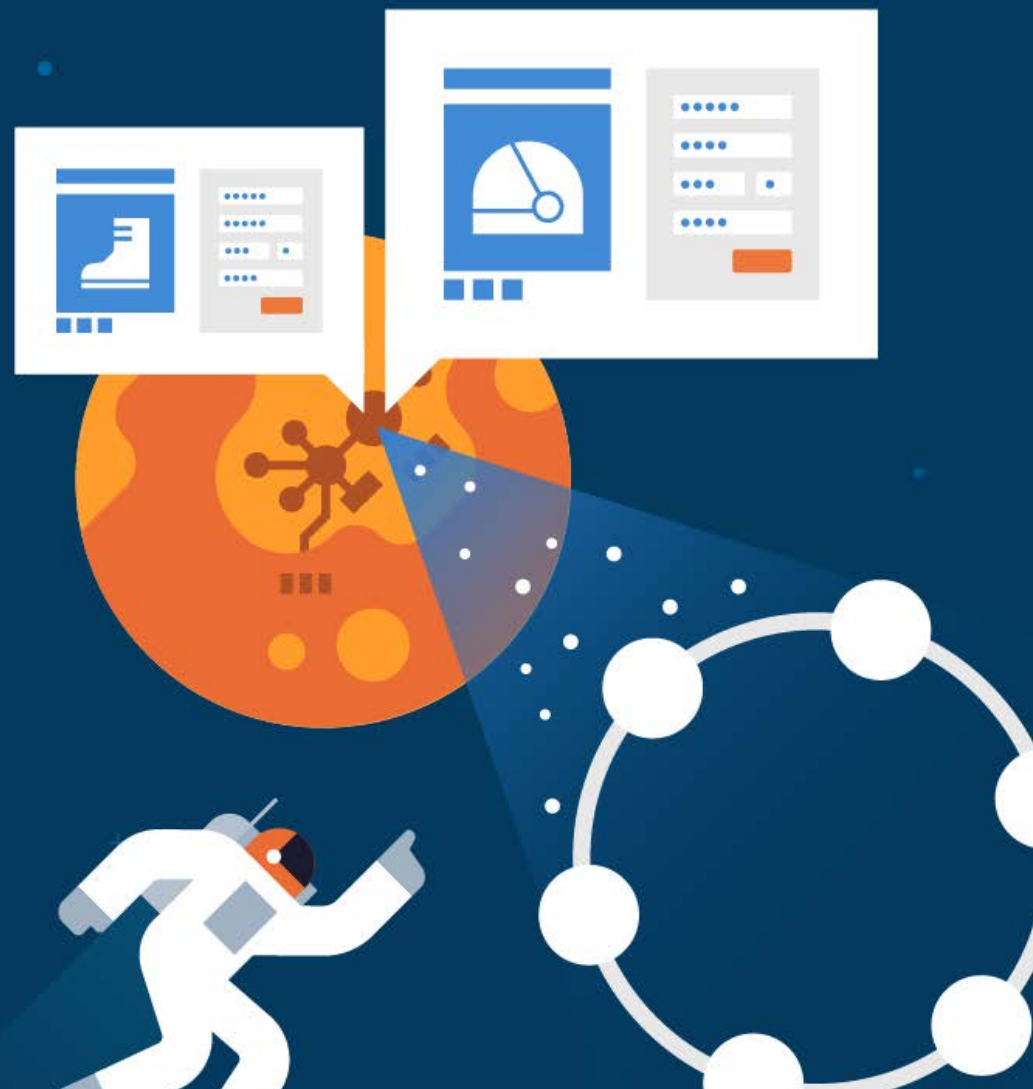
Kubernetes offers a simpler, faster, easier path to building data-driven enterprise applications with Cassandra. It ensures that teams can scale their Cassandra environments to accommodate growth without adding significant risk or complexity. All of these benefits, in turn, contribute to a lower total cost of ownership (TCO) for a production Cassandra environment—making Cassandra a more cost-effective option for a wider range of companies than ever before.

Kubernetes and Cassandra is a cloud-native match made in heaven—almost.

Here's how DataStax brings it all together.

Cloud-native technology is a story that goes in cycles: As cloud environments scale and evolve, they tend to get more complex. Developers build tools to address the complexity, and they're wildly successful—but, in turn, they introduce new sources of complexity that require new tools to manage.

Cassandra and Kubernetes are no strangers to this "two steps forward, one step back" dance: Setting up a Kubernetes cluster can be an intimidating task, especially for a team that's new to the technology. Managing a Cassandra cluster can be just as challenging, and the stress level really starts to ratchet up when you're building a cloud-native environment that will run business-critical or revenue-generating applications.



Hat trick: 3 ways to elevate your Kubernetes and Cassandra experience

DataStax is on a mission to make working with Cassandra as simple, easy, and profitable as possible. We understand Cassandra because we played a major role in launching the project and setting the groundwork for a thriving and diverse development community—one that continues to make Cassandra one of the world's most versatile, robust, and reliable NoSQL database systems.

Today, the Cassandra community also maintains a number of additional projects that complement Cassandra and amplify its value as a cloud-native development platform:

The open source **cass-operator** developed by DataStax abstracts away the complexities of deploying Cassandra on Kubernetes with a simpler, logical interface for describing a containerized application. A YAML file delivers information about a deployment to the Kubernetes cluster, including any configuration changes that deviate from the defaults.

Using an Operator can be a truly game-changing moment for teams that deploy Cassandra or DataStax Enterprise on Kubernetes, transforming tasks that once took hours into a relatively trivial, repeatable, one-step process.

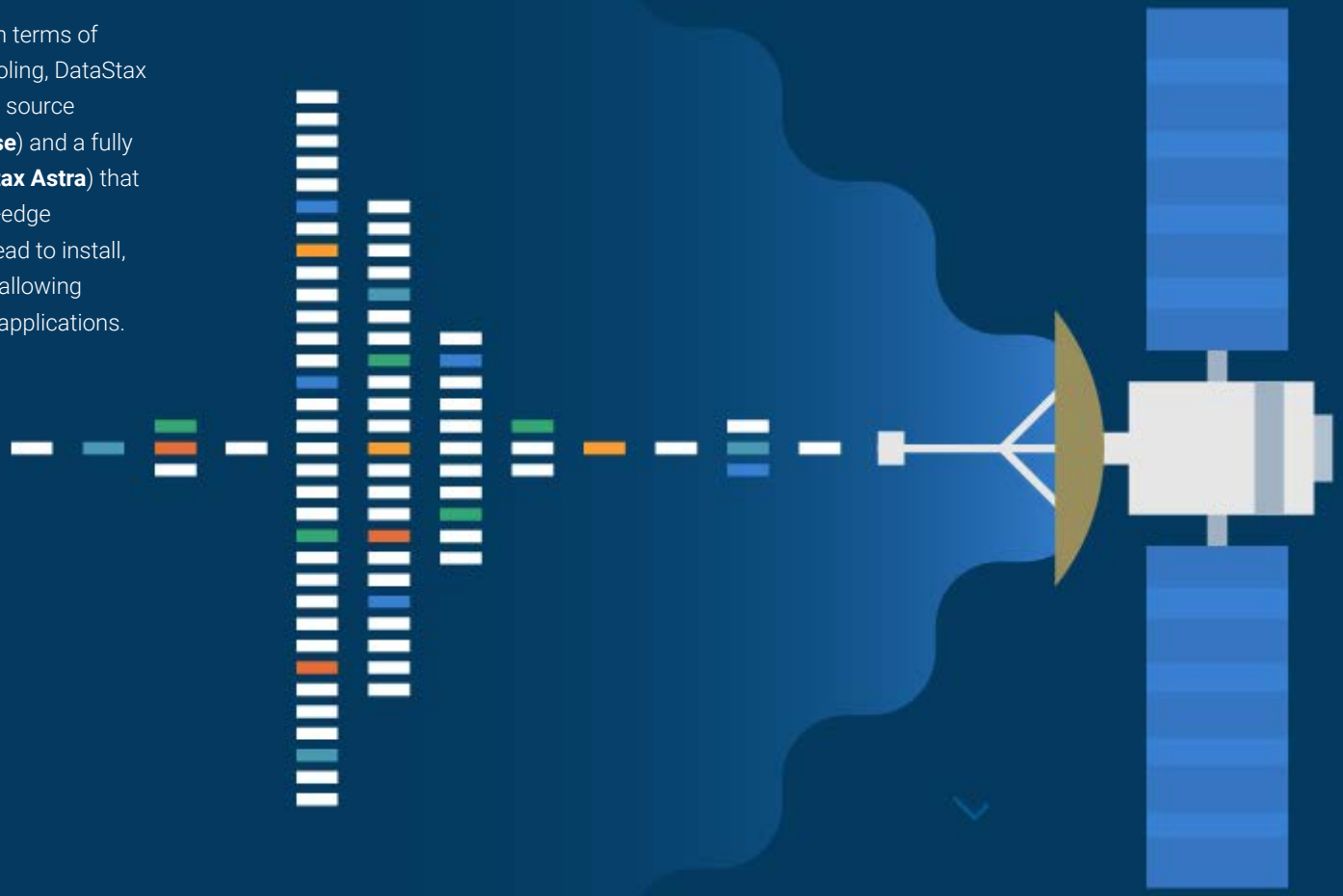
The open source **K8ssandra project** provides a cloud-native distribution of Apache Cassandra optimized to run on Kubernetes. K8ssandra extends the Operator with automation, health and monitoring, backup and replication, and management tools—all preconfigured and integrated to provide the best possible experience running Cassandra on Kubernetes.

Like our Kubernetes Operator, K8ssandra gives teams a great starting point for leveraging the power of both platforms: lowering the learning curve, eliminating complexity, and steering clear of avoidable mistakes.

Project Stargate, another open source contribution by DataStax, is a data gateway that sits between the application and database and supports multiple API optionsDevelopers no longer have to learn Cassandra Query Language (CQL) to write applications on Cassanda. They can quickly bring innovative apps using massive, real-time data to market using native APIs such as JSON, REST, and GraphQL, and operators can spend less time monitoring runbooks and individual nodes. Because Stargate improves performance by separating read and write workloads, overall performance is also improved.

All of these tools give those working with open source Cassandra environments a big advantage—whether they're deploying a database on Kubernetes or another platform.

And for teams that need the very best in terms of support, performance, reliability, and tooling, DataStax offers an on-premise, commercial open source Cassandra offering (**DataStax Enterprise**) and a fully managed, cloud-based offering (**DataStax Astra**) that leverages Kubernetes to deliver cutting-edge capabilities while eliminating the overhead to install, operate, and scale Cassandra clusters, allowing developers to focus on building crucial applications.



Kubernetes and Cassandra: Full of surprises—and potential

Some of what we've discussed here may come as a surprise to enterprise technology leaders who might have functioned under some very different assumptions about working with Cassandra—or with any NoSQL database—using Kubernetes and containerization. That's a testament to just how quickly technology innovation moves in the cloud and also to just how powerful Kubernetes has become as a magnet for cloud innovation.

If you'd like to discover just what Kubernetes and Cassandra can help your enterprise development teams accomplish, take the next step:

Try Astra for free today.

DataStax