SUSE

White Paper

Deliver Applications Faster

Containers, Kubernetes and the Road to Cloud-native

A pragmatic guide for business and IT leaders

Index

Why Read this White Paper

The Olympic motto is "Citius, Altius, Fortius," which is Latin for "Faster, Higher, Stronger." This might also be a highly appropriate mantra for almost every modern enterprise organization. Driven by new customer demands and rapidly changing market conditions, businesses are continually striving for faster response, higher levels of efficiency and stronger performance.

As we move into the 2020s, we find ourselves living in a technology-driven and predominantly digital economy underpinned by software-enabled services, products and solutions. Profound cultural changes are underway that many now refer to as the fourth industrial revolution. Technologies such as Artificial Intelligence (AI), Machine Learning (ML), robotics, mobile devices, the Internet of Things (IoT), smart cities or spaces and Virtual Reality (VR) are creating an age of business disruption.

With technology now at the forefront of business value creation, forward-thinking business and IT leaders recognize that a new, smarter and more agile way to deliver modern applications is needed in order to:

- Bring new and better products to market more rapidly, responding faster to customer demands.
- Accelerate innovation, reduce costs and improve operational efficiency.
- Fully exploit the new business opportunities on offer, maximizing profitability and market penetration.
- Gain a competitive advantage.

The spotlight is increasingly moving toward a cloud-native approach to delivering applications that enables valuable technology improvements to be introduced more rapidly. This approach embraces agile development practices, containerized applications, microservices-based architectures, integrated DevOps teams and processes, and end-to-end application lifecycle automation. These are all key components of the digital transformation that is widely perceived as a modern business imperative.
The objectives of this white paper are to provide business and IT leaders with:

1. An overview of the cloud-native approach, along with guidance on how it can be adopted in a pragmatic, realistic manner.
2. The rationale to assist in making progress along the road toward cloud-native by more fully embracing containers and modern container and application platforms.
3. Recommendations for how to reap the benefits of this new cloud-native approach.

Moving Toward Cloud-native Applications

Developers, IT leaders and anyone who tracks or analyzes technology trends will be fully aware that containers and cloud-native applications are hot topics. They are constantly in the news and any industry conference focused on them is likely to draw large crowds of attendees.

Forward thinking business and IT leaders recognize that a new approach to application delivery is needed to gain a competitive advantage.

What exactly does 'cloud-native' mean? While the phrase suggests a class of applications intended to run in public, private or hybrid clouds, cloud-native means much more than that. Cloud-native applications scale efficiently and autonomously; expect and tolerate partial failures; move fluidly from one environment to another for cost or performance optimization, and much more. In short, they operate well in today's highly dynamic technology environments, which are typically modern cloud environments. However, equally essential to the definition of cloud-native is that cloud-native applications are delivered rapidly, efficiently, and at scale. New capabilities are introduced quickly and frequently, in a highly automated way that can support ongoing delivery of thousands of applications every day. Because of this, taking a cloud-native approach to delivering applications can speed innovation and accelerate a business' ability to provide valuable new capabilities for competitive advantage.

Simply put, a cloud-native approach is built from the ground up to enable a new class of modern applications to be delivered rapidly and efficiently at scale. While that may sound straightforward, it actually involves a paradigm shift. It requires a rethink of how applications are designed, developed, deployed, and operated. It combines new technologies, changes to application architectures, alterations to business processes and organizational dynamics.

These are all hurdles that need to be overcome. However, for any organization focused on achieving greater agility – especially for those wanting to develop, deploy and manage applications at factory speed and scale – the business benefits on offer are well worth the effort and investment.

For many organizations, transformation to a cloud-native approach is already underway and is happening concurrently on a number of levels.

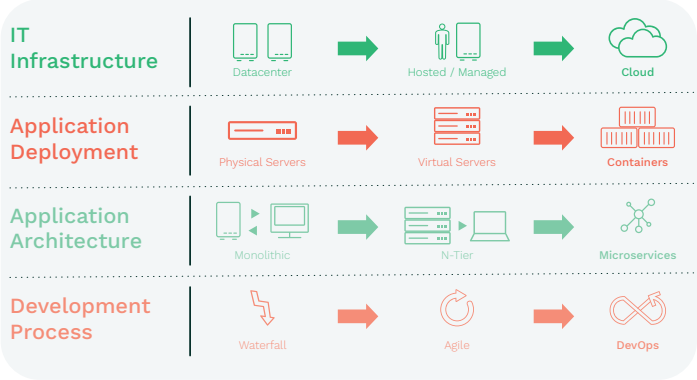

Figure 1. Cloud-native Transformation Happens at Multiple Levels

- From an infrastructure perspective, the move away from a purely hardware-oriented data center to a vastly more flexible software-defined infrastructure (SDI) or cloud-based approach is well underway.
- For application deployment, physical server-centric environments have largely been overtaken by virtualized environments and virtual machines (VMs). The current shift to containers provides a new packaging mechanism that is even leaner on resources, as well as being more portable and agile.
- Early monolithic applications have broadly been superseded by multi-tiered architectures that enable developers to build more flexible and reusable code. The latest transition to a microservices architecture makes it possible to compose cloud-native applications from smaller units of loosely-coupled code. Each microservice can be independently deployed and scaled.
- Software development practices have been evolving simultaneously. Replacing the traditional waterfall methodology, which follows a highly structured linear and sequential workflow with typically long software release cycles, the agile development method is now a mature and widely used alternative approach. It involves iterative, collaborative and team-based techniques that focus on enhancing code incrementally and frequently to shorten release cycles. By incorporating application operations into the loop, new DevOps practices are the natural cloud-native extension of agile development practices. They add automated application deployment and lifecycle management, facilitating more rapid uptake of new applications into production. They also help break down barriers that have traditionally existed between development and operations teams, dramatically accelerating software delivery.

Organizations of all sizes and in almost all market segments are already focused on getting new applications into production or out to market faster. They recognize their need for greater agility and for improving how efficiently they build, deliver and manage applications across their entire enterprise. However, the goal of cloud-native solutions remains aspirational for many – at least for now.

Most business decision-makers are aware of highly successful digitally native, e-commerce or internet-based smart businesses that have successfully leveraged cloud-native to their advantage. Even so, it would be a mistake to assume that cloud-native is uniquely for startups, that it is only for organizations that can adopt a “greenfield” approach, or is solely the realm of organizations with a “born in the cloud era” mentality.

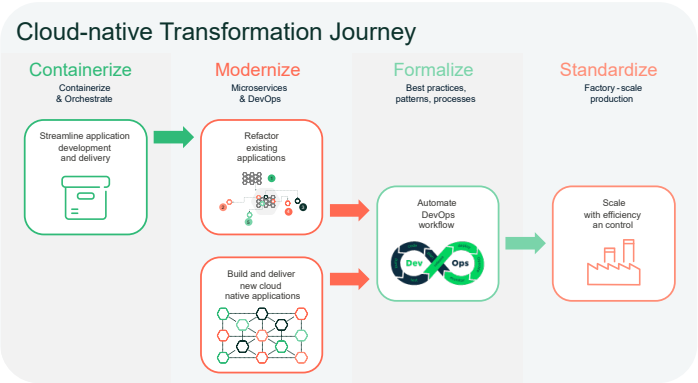

Figure 2. The Path to Cloud-native

As illustrated in Fig 2, the path to cloud-native is a journey that can be executed in stages, each of which is an independently valuable stepping stone along the way.

This means that it's also appropriate for more established businesses seeking a pragmatic, balanced and realistic approach to building new applications in new ways. It's perfectly feasible to begin the journey with new containerized applications, while continuing to run existing applications and systems in parallel. Older applications can then be modernized if and when it's appropriate or optimal to do so. As things mature over time, the whole approach can be extended, standardized and scaled when appropriate.

Step 1

The Cloud-native Journey Begins with Containers

Wherever your organization finds itself in its cloud-native transformation journey, one thing is undeniable: application containers are an essential component. Industry analysts predict that the container marketplace will reach $4.3 billion by 2022.1 The Linux Foundation reports that the popularity and importance of containers are growing so rapidly that 57 percent of hiring managers are actively recruiting new employees with container expertise for their companies.[2]

60 percent of enterprises are already leveraging Kubernetes and another 27 percent are planning to in the near future.

Containers are already a mature, mainstream technology. They are being used in virtually all industries and business segments today. These include finance, science and research, utilities, retail, urban planning, smart spaces, entertainment, manufacturing, healthcare, agriculture, and more.

Streamline Delivery of Existing Applications

Many organizations are utilizing containers, along with automated container management, as the first and easiest step toward streamlining existing application development and deployment practices.

Containers provide developers with a convenient way to package code, with a number of useful advantages when compared to virtual machines. They are smaller than VMs, faster to initiate and easier for developers to build and deploy themselves. But most importantly, containers are highly portable; they include everything needed to run the code they contain, ensuring they operate consistently in any environment.

The result is that development and operations teams can move containerized code across any number of different environments on the path from development to test to production without needing to manually configure each environment in exactly the same way. This removes the stress that can arise between owners of different environments, all of whom need to run the same code, but who, for important reasons, may simply not be able to configure their environments identically.

Without containers, the cost of those differences can be very high, counted in hours spent debugging environment-related problems. With containers, different teams can maintain different environments and more easily run the same application code across them all.

The time savings containers provide make them a perfect fit for organizations using an agile and iterative development model, where applications are built rapidly and with much faster cycle times. Their consistent operation also reduces risks associated with moving containerized code from dev/test environments into production, and makes it easier to integrate application development and operations processes.

Kubernetes Enhances the Container Experience for Operations Teams

Even greater gains can be achieved by combining the use of containers with Kubernetes to automate the deployment and management of containerized applications. This includes container scheduling and service discovery; monitoring of container performance and availability; container scaling, load balancing and self-healing capabilities; as well as application maintenance, including automated rollout and rollback of updates.

Kubernetes has quickly emerged as the industry's de-facto standard for container management, and is expected to become the dominant platform upon which modern applications will run. Surveys suggest that 60 percent of enterprises are already leveraging Kubernetes and that a further 27 percent are planning to in the near future.[3]

1 Source: 451 Research
2 Source: Linux Foundation. 2018 open source jobs report
3 Source: RightScale 2019 State of the Cloud Report

All major cloud services providers currently offer Kubernetes platform services, and any enterprise that wishes to deploy containerized applications that wishes to deploy containerized applications on-premises will likely also require a Kubernetes platform in its own datacenter.

The portability of containers combined with the universal availability of public cloud Kubernetes services makes it easier to take advantage of multi-cloud infrastructures. Application teams can easily adopt a "lift and shift" technique to move existing applications to new or different platforms as needed. Managing applications across multi-cloud environments has been promised for many years, but only now, with containers and Kubernetes, is that promise finally being realized in an economically rational way.

Many Independent Software Vendors (ISVs) are also now offering containerized versions of their solutions. Utilizing containers instead of VMs can result in significant resource savings. Using Kubernetes to automate the deployment and ongoing operational management of this ISV software can deliver even greater advantages – especially if the application is designed in a cloud-native way. Frequent application updates can be smoothly incorporated into the already running solution, while application availability or scaling can be automatically and efficiently managed.

Step 2

Bolder Modernization Strategies

The next steps on the cloud-native journey require enterprises to start architecting, developing, deploying, and managing applications in a more cloud-native way. For many organizations, this stage is most easily initiated with a new application project, since building from scratch generally imposes fewer pre-existing constraints that could impede a successful outcome. Once a simpler application has been delivered, teams may be ready to tackle more complex applications. At some point, they may apply what they've learned to refactor existing applications so these can also benefit from a more cloud-native approach.

The purpose of the cloud-native approach is to enable the delivery of a new class of modern applications rapidly and efficiently at scale.

Building new Cloud-native Applications

As mentioned earlier, cloud-native involves building a new class of modern applications from the ground up in a way that enables those applications to be delivered rapidly and efficiently at scale. Application release cycles times shrink from months to weeks, days to hours, and in some cases even hours to minutes, facilitating more rapid innovation and new value creation. Cloud-native applications themselves are highly scalable, resilient and rapidly evolvable. Further, they are capable of managing many of their own operational requirements dynamically and automatically, enabling extraordinary new levels of operational scalability.

Gartner describes cloud-native application development this way:

> Cloud-native application development involves designing and developing an application to support deployment to the cloud — free from constraints on specific hardware, tight coupling with other components and shared state. Developing applications to run in the cloud allows development and operations teams to focus on the functionality of the application rather than hardware and infrastructure to support the application. Additionally, capabilities and tooling in cloud-native platforms provide significant benefits in agile architectures, testing practices, DevOps deployment models, elastic scaling and resiliency.[4]

The cloud-native approach touches nearly every aspect of the application lifecycle. It embraces the following practices:

- Adopting microservices-based architectures to ensure application components can be deployed, scaled, and operated independently.
- Applying an iterative and agile development approach supported by continuous integration/continuous deployment (CI/CD) pipeline automation.
- Using containers for efficient code packaging and portability.
- Leveraging a powerful container management platform such as Kubernetes to automate application deployments and lifecycle management operations.
- Integrating application development, operational platforms and processes (DevOps), so that services and applications can be deployed more frequently and automatically.

4 Gartner, "Essential Skills for Modern Application Development," Brad Dayley, 11 March 2020

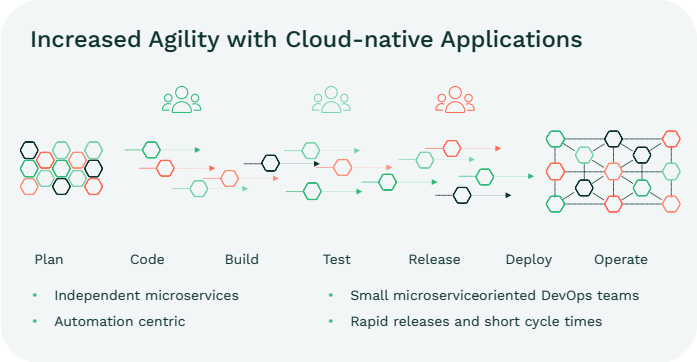

Figure 3. Cloud-native Workflow

While most applications are not currently delivered in a cloud-native way, many enterprises are already employing one or more cloud-native practices. Many, for example, are already using an agile development process, or have implemented CI/CD pipeline automation, or are using containers to some extent. Each of these elements is independently valuable, and worthwhile implementing alone. As more cloud-native practices are used in combination however, the results become increasingly compelling. Fortunately, adding new elements into existing environments can be natural next steps: extending an agile dev/test team to include an ops person, or integrating automated Kubernetes deployment into an existing CI/CD pipeline are examples.

It's self-evident that the move to cloud-native described here is more complex and involved than simply using containers, even though containers are a key component. However, the rewards of embracing a richer cloud-native approach for appropriate enterprise applications and at the opportune time for your specific organization can yield compelling results.

Modernizing Existing Applications

Using containers to streamline application development and deployment is undeniably gaining momentum, while adopting a cloud-native approach for new applications is now clearly on the horizon for many organizations. Modernization of existing applications is a third use case that is arguably the most complex and difficult to implement. It involves restructuring an application without changing its essential external behavior, but with the objective of improving non-functional characteristics, including scalability, stability, maintainability, and the like.

For many existing business-critical applications, it might be simpler to take the "if it isn't broken, don't fix it" approach and leave well enough alone. And indeed, some applications may not be well suited to redesigning, restructuring or rebuilding, or there may be no business or operational advantage in doing so. For other aging applications, modernization may go no further than simply using containers for the "lift and shift" advantages described above.

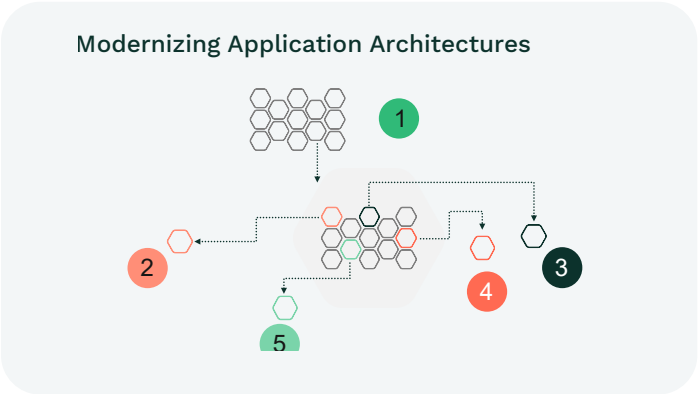

Figure 4. Modernize Existing Applications

However, there are some situations where there are obvious advantages to be gained by modernizing existing applications. Typically, the process involves refactoring parts of the application into independent microservices. While it may not be worthwhile to refactor the application completely, candidate areas can be readily identified. If the application implements some capability that could also be used by other applications, then it may make good sense to create a separate microservice to provide that capability. For example, a single login microservice could ensure logins are handled consistently across multiple applications and save development time as well. Modernizing may also allow part of an existing application to scale as a standalone microservice without requiring the entire application to follow. Perhaps one area of an application requires more frequent updating than the rest – that component might be better implemented as a microservice that can evolve independently and more rapidly. If an application has been well designed and maintained over time, with cleanly encapsulated functional logic and code that can be extracted without a significant ripple or any knock-on effects, it may benefit from some level of refactoring.

Step 3

Maturing Cloud-Native Practices

Early experiences with cloud-native application development are useful learning experiences. These include using containers and Kubernetes, embracing microservices-based application design and development, managing distributed systems, and adopting new workflows that tie people and processes together. Mistakes are made, insights are gained, and improvements are implemented. Over time, these learnings are captured and a set of best practices are drawn up. These serve as guidelines to smooth the way for repeatable success. The process does not end however, as new learnings are continually integrated and the entire system matures and evolves.

A standardized process is used to deliver modern applications at factory-scale production volumes and speeds.

Formalizing Best Practices, Processes and Methods

This formalization process happens within application teams, and also across business units, communities, or entire industries. Over the past decade, cloud-native concepts have been introduced and refined by technology innovators and early adopters. Today, mainstream organizations can benefit from the learnings of the past to shortcut their own cloud-native learning curve and implementation timeline. Leveraging a large ecosystem of freely available resources, they can now take advantage of proven models such as The Twelve-Factor App[5] methodology, and even complete cloud-native application platforms such as Cloud Foundry,[6] that provide ready-to-use automation of the entire cloud-native application lifecycle.

Scaling Across the Enterprise

Recall that the purpose of the cloud-native approach is to enable delivery of a new class of modern applications very rapidly and efficiently at scale. Each cloud-native artifact, design principle, and process workflow is intended to fit into a well-defined and highly automated application delivery framework. As a result, when put together a complete system operates like an application factory – one that can support high application throughput at low cost. And it is in a full-scale, high production mode that the ultimate promise of cloud-native is truly realized.

Think of the way Henry Ford revolutionized the automotive industry with the factory production line and assembly methods he introduced to manufacture his famous Model T. His automated and streamlined continuous flow manufacturing process was a paradigm shift away from the custom-build approach that it replaced. It enabled rapid and efficient production at scale, making automobiles far more affordable and accessible.

In the same way, following cloud-native principles allows enterprises to transition from using numerous different deployment and operations processes, custom made for individual applications. Instead, a standardized process is used for delivering modern applications that is automated and streamlined to handle factory-scale production volumes and speeds.

That doesn't mean that enterprises will no longer need custom-created workflows and processes to manage applications that are not fully cloud-native. There will likely always be applications that fit into this category. But the trend over time will be to produce more and more cloud-native applications that can be efficiently developed, deployed and managed by standard automated processes. And just as with the Model T Ford, there are tremendous efficiency, cost and competitive advantages to be gained.

Not surprisingly, there is real work involved in scaling application production across an enterprise, and that work generally falls to IT Operations teams. New requirements will include the need for consistency, governance, and control. This is especially true where scaling involves integrating and consolidating across disparate teams with different practices and processes. Establishing standardized, global practices is critical to ensure overall operational efficiency. Managing multiple teams and organizations, including their access to resources, becomes increasingly important as more people become engaged with application delivery systems. And resource management itself becomes more complex too, requiring deft control over a potentially large array of multi-cloud infrastructure and myriad Kubernetes clusters.

To support this factory model, you could create and maintain your own standard workflow and resource management framework. As an alternative, you could leverage one that already exists, with the aim of becoming more productive sooner and of maintaining optimum productivity over time. Our recommendation is to evaluate the Cloud Foundry model, which is a mature and well-proven open source option.

---

5 Source: 12factor.net
6 Source: Cloudfoundry.org

Cloud-native Platform Considerations

Making the right technology, platform and vendor choices is critical to the success of any transformation strategy. Here are some factors to consider.

As the world's largest purely open source software vendor, SUSE should be on your shortlist as you weigh up your cloud-native technology and platform options.

Open Source: the Engine Driving the Cloud-native Journey

When it comes to cloud-native, it's worth noting that nearly all the supporting technologies are open source at their core. A prime example is Kubernetes, which is hosted under the stewardship of the CNCF (Cloud Native Computing Foundation) and the Linux Foundation. Other examples include Cri-o (container runtime), Helm (package manager), Prometheus (monitoring), Grafana (visualization), Cilium (network security), Cloud Foundry Application Runtime (cloud-native application platform), and Stratos (multi-cluster management), to name just a few. The virtues of open source software and the benefits of using it are well understood, as is the value and importance of working with vendors that can ensure enterprise readiness and world-class support for open source software solutions.

As the world's largest purely open source software vendor, SUSE should be on your shortlist as you weigh up your cloud-native technology and platform options.

SUSE CaaS Platform: Kubernetes for the Agile Enterprise

While Kubernetes has quickly gained almost universal popularity and acceptance, it is not all plain sailing. For any organization intent on setting up a Kubernetes environment using upstream code, the process can be complicated and arduous. A minimally useful Kubernetes environment requires not only Kubernetes but many additional complementary components and an operating system. All of these evolve independently and require ongoing interoperability testing to ensure enterprise readiness. The rapid code release schedule for Kubernetes has helped it come a long way in terms of features and functionality in an incredibly short time. But keeping up with this release schedule, as well as maintaining and managing the platform, is hard work. The level of difficulty is a step too far for many companies, especially for those wanting to concentrate on their own core development strengths and wanting to avoid operational complexity.

Employing a packaged, supported and CNCF-certified Kubernetes distribution such as SUSE CaaS Platform can help remove many of these complications and problems. SUSE CaaS Platform is an enterprise-grade Kubernetes distribution that includes everything needed to make Kubernetes easy to consume and manage, thus delivering an exceptional experience for IT operations teams. Platform operators can get production-ready Kubernetes clusters up and running seventy-five percent faster7 than a typical deployment of upstream Kubernetes code, whether that's using on-premises or public cloud infrastructure. Day two operations, such as cluster scaling and software maintenance, are also greatly simplified.

SUSE Cloud Application Platform: Cloud-native Productivity at Scale

Cloud-native requires shifting up a gear in terms of building and deploying larger numbers of applications, at much faster speeds, and more frequently. Cloud-native involves applying "factory style" techniques to achieve new levels of automation, scale and speed. It involves ramping up production by applying a standardized approach and processes to handle the vast majority of applications.

There is now a general acceptance that Kubernetes is the right foundation for cloud-native. However, Kubernetes alone will not provide the full answer to every cloud-native problem. As an example, while it delivers a flexible and powerful platform for operations teams, it does not provide an inherently intuitive developer experience.

To achieve productivity at scale, a well-integrated, fully automated, and highly scalable platform that can be used equally well by all participants in the application lifecycle is crucial.

7 Source: Tymlez success story

Fortunately, Kubernetes has been designed to serve as a 'platform for other platforms,' meaning that it is intended to be used as a foundational technology layer. Additional abstraction layers and functionality can be added to provide better experiences for different sets of users. This opens up the possibility of creating a richer platform running on top of Kubernetes that more comprehensively address the specific needs of cloud-native application developers. As mentioned previously, you could custom-build such a platform yourself, but you don't need to. SUSE has already done it for you with SUSE Cloud Application Platform.

SUSE Cloud Application Platform is an outstanding modern application platform that fully caters to the needs of both developers and IT operations professionals. It implements the proven Cloud Foundry workflow to simplify the developer experience, streamline cloud-native application deployment and boost productivity, efficiency and agility. It runs on SUSE CaaS Platform, as well as the Kubernetes services offered by all the major public cloud service providers, so that it can be used across multi-cloud environments. With additional support for multi-tenant, multi-cluster, and multi-cloud management, SUSE Cloud Application Platform is a choice that you can start with, and stay with, no matter how large your cloud-native application development practice grows.

Conclusions and Recommendations

In our predominantly digital era, modern cloud-native applications hold the key to success. New customer demands and rapidly changing market conditions are compelling enterprise organizations to seek new, smarter and more agile solutions as they strive for faster response, higher levels of efficiency and stronger performance.

Against this backdrop, the scene is set for the journey toward new application development and deployment methods and eventually onward to fully cloud-native solutions. Along the way, some of the stepping stones include adopting agile development practices, containerized applications, microservices-based architectures, integrated DevOps teams and processes, and end-to-end application lifecycle automation. The ultimate aim is to deliver greater agility, by delivering a new class of modern applications rapidly, efficiently, and at scale.

How can organizations successfully undertake this journey? Here are SUSE's recommendations for business and IT decision-makers:

1. Review how applications, products and services are currently built and delivered to create a baseline for how IT resources and day to day operations are managed today.

2. Prepare a realistic plan for when and how to transform application development and deployment methods and processes. Be prepared to start small with simple projects to deliver short term wins and measurable benefits. Then expand the scope and complexity of these projects in well-defined stages. This should be done as organizational experience and confidence builds. Begin by using containers to streamline delivery of existing applications. Then move on to building new cloud-native applications and eventually to modernizing existing applications.

3. Recognize that this is a transformational journey. Traditional and cloud-native applications will coexist for a long time. However, over time, the center of gravity will shift towards cloud-native. A mature plan will eventually need to include how to standardize and scale best practices, processes, methods, and platforms across the entire enterprise.

4. Fully embrace Kubernetes as the foundational platform to support your end-to-end transformation strategy for containerized applications, modernized workloads, and new cloud-native initiatives.

5. Research and evaluate ready-made application platforms that can enhance your Kubernetes environment. Choose options that are most appropriate to your business needs to help accelerate delivery of cloud-native applications at scale.

6. Choose an expert and professional technology partner such as SUSE, who will help you map out, navigate and be your guide for the journey toward cloud-native. As a thoroughbred open source veteran, SUSE has a comprehensive portfolio of cloud-native solutions, tools, and services, as well as a wealth of experience to share.

Additional contact information and office locations:
www.suse.com

www.suse.com