Self-Service Infrastructure Buyer's Guide

A Comprehensive Guide to Evaluating On-Premises Self-Service Infrastructure

nebulon

introduction

Enterprises have been searching for ways to bring the ease and simplicity of the public cloud to their on-premises environment and most have found that deploying a self-service infrastructure is an ideal way to do this. Self-service infrastructure is a key goal for IT organizations who are looking to scale DevOps initiatives quickly and cost effectively. Self-service infrastructure provides organizations with a competitive edge as product teams gain new levels of productivity which allows for accelerated delivery of immersive products and digital experiences. While infrastructure provisioning in the context of the public cloud sets the standard for agility and ease of use, organizations are finding difficulty achieving the same experience on-premises for workloads that simply cannot or should not run in a public cloud.

Why? For starters, legacy infrastructure is difficult to automate and requires intimate, domain-specific expertise to provision and maintain effectively, which ultimately requires IT organizations to remain intimately involved in each and every provisioning action needed by DevOps teams. One thing is certain, companies must re-imagine on-premises infrastructure without the expense and inflexibility of legacy infrastructure, particularly that of arrays and 3-tier architectures.

There is a rise in the number of enterprises who are actively pursuing a path to overcome the challenges they are facing and implement self-service deployment options for their on-premises infrastructure. On this three-part journey, the first step is to modernize infrastructure for specific applications and reduce the complexity of 3-tier architectures. In the second step, organizations should automate routine and trivial infrastructure operations and free up the teams' backlogs from repetitive requests. Finally, the third step of the journey is about establishing self-service catalogs with templates and a platform for product teams to roll out infrastructure rapidly and consistently.

> By 2023, **90 percent of enterprises will fail to scale** DevOps initiatives if shared self-service platform approaches are not adopted.
>
> **-Gartner, *Top 10 Trends Impacting Infrastructure and Operations***

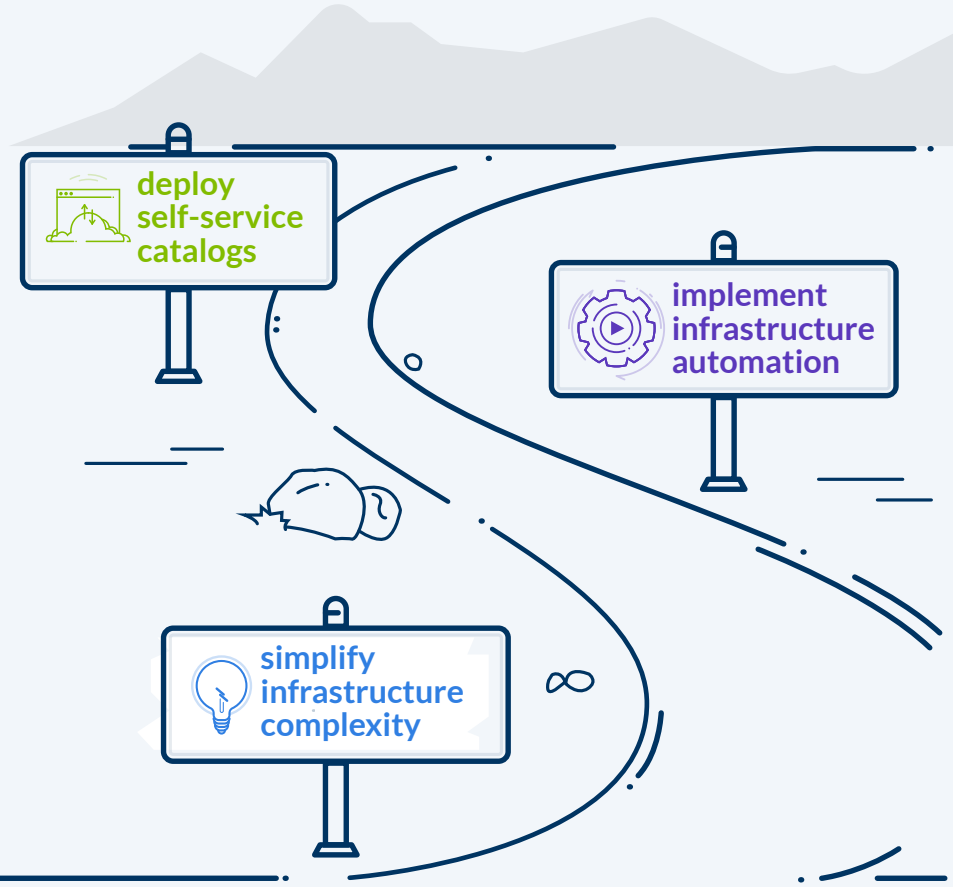
Despite most organizations having not yet implemented full self-service capabilities for an on-premises infrastructure, they are already investing in steps to enable this panacea in the near-term. For example, an organization may be limiting all new infrastructure solutions or services to those with an API-first architecture. They may also be heavily prioritizing automation capabilities in evaluation criteria. So, how do they select the right on-premises infrastructure that will enable a self-service approach? The reality is that it must begin with storage and a shift from 3-tier approaches to one that is server-based and that deliver on claims of simple, cost-effective operational benefits. But is that all too good to be true? This guide delves further into the criteria which will help ensure the right choice is made which will allow for IT to successfully deliver a self-service infrastructure for its product teams.

WHAT IS A SELF-SERVICE INFRASTRUCTURE?

With a self-service infrastructure, product teams and application owners are able to rapidly and reliably provision and maintain the IT infrastructure for their applications themselves without dependencies on IT operations teams.

the journey to a self-service infrastructure

Before delving too far into what features must be available to qualify as a self-service infrastructure, organizations must first understand the critical role the foundation, particularly storage, plays. To successfully deliver a self-service model, IT organizations must first overcome the challenges of their traditional infrastructure and simplify management complexities across their datacenter.



STEP 1
simplify infrastructure complexity

While there are elements of an on-prem environment which cannot be eliminated, the architecture of compute, network and storage is critical to both reducing the number and complexity of components required and improving agility or 'composability' of the assets.

Most mission critical on-premises infrastructure environments are built upon a 3-tier architecture which is composed of discrete compute, storage, and networking building blocks. Each storage fabric requires its experts and operational workflows. It is undeniable that a 3-tier approach adds to the budget from an acquisition perspective. There is complexity involved in terms of interoperability of these components. And of course there is the head-count required to monitor and maintain these heterogeneous components for them to work well together.

Virtually every customer who is modernizing their fleet of server-storage infrastructure is seeking to eliminate 3-tier architecture complexities and envisions a more homogenous, server-based datacenter capable of supporting the range of applications required by a modern enterprise. Most have tried moving to a server-based storage alternative like Hyper-Converged Infrastructure (HCI) or Software-Defined Storage (SDS). Both HCI and SDS aim to place the application owner in the driver's seat. However, most IT organizations have found HCI & SDS to be unviable solutions for key use cases as both options impose numerous restrictions including limited operating system, hypervisor, or application support. It is well known and documented that both HCI and SDS also consume valuable host-server CPU, memory and networking resources. For every 4 servers deployed in an HCI environment, the equivalent of one server is dedicated to running the HCI storage software. Adding to the list of issues are HCI and SDS' compounding costs of core-based application licensing models. Whenever host-server resources are being utilized to perform storage services instead of application services, the promised ROI for the app deployment is difficult to achieve. However, the limitations of HCI and SDS do not end there. HCI and SDS were designed such that the architecture moved storage responsibility into the hands of the virtualization, container or application administrator, who is not armed with the necessary experience or training. With this shift, app owners are being forced to carefully plan and operate their storage stack along with their applications. At first sight, this may not seem to be an issue, but over time will create both overhead and duplication of duties between multiple product teams, and eventually, issues with scaling DevOps initiatives.

A core desire for self-service infrastructure provisioning is the ability to change the infrastructure based on the need and preference of the application owner anywhere in the nterprise. For example, a database administrator will not have the foresight to predict the expected capacity and performance needs for a 3-year life span of their infrastructure in most cases as the influencing factors are too broad and unforeseeable. Users now demand infrastructure agility which allows them to scale as their applications require. This means adding and releasing resources on-demand as provided by scale-out architectures, and to quickly modify software stacks — unassisted, anywhere in the datacenter.

STEP 2
implement infrastructure automation

Automation promises lower cost, better quality, and improved agility but requires a comprehensive strategy, toolset and supporting platforms. These items will empower IT organizations to systematically identify and implement automated workflows beyond basic task automation in isolated corners of their infrastructure.

A well learned lesson from the adoption of public cloud-based infrastructure is that application developers appreciate infrastructure programmability in an all-inclusive API. While on-premises storage infrastructure automation may be implemented tactically in corners of the datacenter, traditional on-premises storage infrastructure struggles to measure up as deployments use element managers with isolated management stations —each introducing a separate set of APIs at different revision levels. The resulting complication to holistically and consistently automate across infrastructure forces users to default back to manual approaches using command-line or graphical user interfaces.

Most graphical user interfaces have become simple enough that various provisioning tasks are quickly done manually in a matter of minutes. Yet, doing repetitive and trivial operations manually wastes valuable time of the operations teams and does not scale. Most provisioning operations involve multiple steps as graphical or command-line interfaces lack a “fix it” function. Workflows need to be remembered, as well as documented and require users to make conscious decisions at every step. Performing repetitive workflows manually also leaves room for human error, potential service level inconsistencies, and misconfigurations.

It is no surprise that IT operations teams are proud to have implemented automation for large scale storage infrastructure, reducing the burden on their small operations teams. But these companies invested years into abstracting command line interfaces of systems that do not provide comprehensive APIs, cataloging all of their systems that each have a separate management endpoint, and implementing automation workflows. The required personnel and development skills for this endeavor is not negligible and not every organization can shoulder this financial and time intensive burden.

STEP 3
deploy self-service catalogs

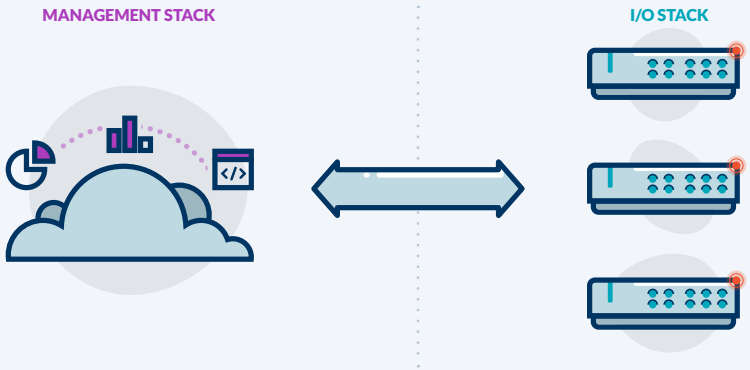
With established automation workflows, applications and product teams gain the agility that they need to successfully compete in a rapidly paced market. This is achievable when application owners and product teams are able to provision their own infrastructure which conforms to organizational policies and best practices and is maintainable without ever involving the infrastructure and operations teams.

Automation can be facilitated on multiple levels of the infrastructure each with its own set of nuanced configuration options and best practices, and therefore require a particular set of skills on each level. While application and product teams are the consumer of infrastructure, they are the subject matter experts of their respective application domain – not infrastructure. Different application teams may demand specific infrastructure for their applications based upon vendor requirements or what they think is the right configuration. Application owners will trust the infrastructure and operations teams to deliver it with reliability and policy conformance. If the variety of infrastructure configurations are not kept in check with defined templates, the infrastructure and operations teams can be easily overwhelmed.

A key success criterion for self-service infrastructure is that it can be rolled out across the enterprise and not just in pockets within the environment. A common challenge for large organizations is that application and product teams are unaware of existing infrastructure automation workflows and services, and can therefore duplicate existing work within their teams. While the public cloud provides a single management console for the entire infrastructure where services are centrally accessible through a catalog, traditional infrastructure on-premises and their management stations are compartmentalized and tough to make available as a centralized service. This is not only because of isolated and separate tooling domains and lack of API access, but also because of the inflexibility of legacy storage technologies which are not built or optimized for the cloud. They are built for either shared or non-shared infrastructure, but not both – a core tenant of an agile and modern cloud environment.

self-service infrastructure — what to look for

1. THE I/O STACK IS SEPARATED FROM THE MANAGEMENT STACK



In traditional infrastructure, the I/O stack and the management stack sit together. Why is this an issue? The I/O stack and management are very different pieces of software and their updates and changes should be handled independently. The I/O stack is extremely performant, highly optimized, and works with a strictly defined set of interfaces. The purpose of the I/O stack does not change much over time. Management, on the other hand, is human facing and tends to change frequently because of conventions, UX practices, or customer feedback. This alone is enough to understand why these two pieces of software do not live well together.

Looking at it closer, the I/O stack is the most important since without it, the storage system has no value. As a result of this, a vendor is going to be very cautious about releasing new firmware versions and customers, rightfully so, are going to be reluctant to install and make the updates to these new firmware versions. From a customer's perspective, there is no need to change software that is working. As a result, a customer's management stack is going to fall far behind because the latest available firmware version is updated infrequently. Even worse, different systems in different environments will run different versions of firmware. This makes it difficult to expect the same performance and reliability levels across the infrastructure. By splitting the I/O and management stacks and moving management into the cloud, customers can take advantage of significant benefits for both which include the fact that the separate software stacks are able to be updated independently of each other.

For the I/O stack, firmware is much smaller, much simpler, and requires much fewer updates. When updates are required, it will be easier to administer them because of the smaller size and the control and management planes in the cloud remain unchanged. In addition, when firmware updates are facilitated through the cloud, they can be made at a much larger scale and consistently across the infrastructure fleet. This is an important factor to consider when updating hundreds to thousands of devices — physical drives for example.

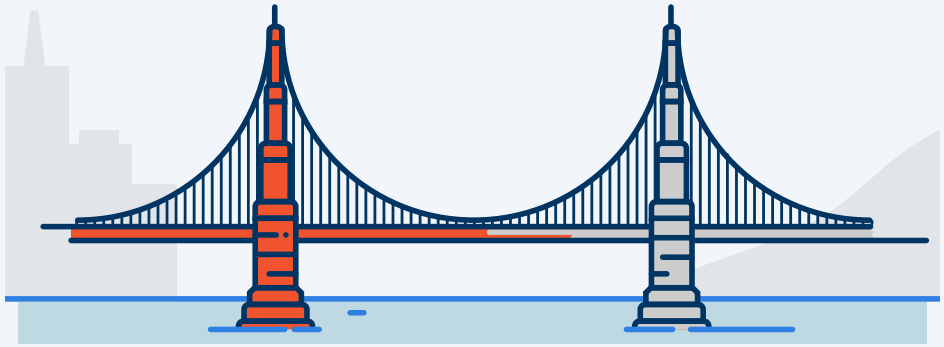
Additionally, by separating the I/O from the management stack and moving the control plane to the cloud, vendors can be much more agile. The management stack can be updated at a much faster rate. Features are released in hours vs months or even up to a year with traditional models. When new features are released, they are immediately available across the entire organization. This means that any user interaction and the entire on-premises deployment will benefit from the new management interface and a consistent API for automation. This is critical when vendors continuously add new backend capabilities derived from customer feedback, system telemetry and use of AI techniques—none of which requires customers to change anything on the systems they own.

2. MINIMIZING COMPLEXITY OF MAINTAINING HW & SW

A key reason why users like to make use of software as-a-service (SaaS) or infrastructure-as-a-service (IaaS) offerings is that the cumbersome maintenance of device firmware or software revisions is handled by the cloud provider. Yet, in the datacenter, patching software is an ongoing task for infrastructure and operations teams. We like to compare it to painting the Golden Gate Bridge – as soon as the job is completed it is time to restart. Customers have many infrastructure devices that once the software upgrades and firmware maintenance is completed, they must start all over again because a new software version has already been released. Plus, coordinating a time when the systems can be upgraded between users, teams, and organizations in a shared storage environment is a nightmare and often results in upgrades during non-working hours (for example, 2:00 AM on a Saturday).

This inconvenience leads to customers falling behind on their software updates. As a result, customers can run into unnecessary stability issues or potential downtime that would have been avoided had they upgraded to a newer software version. The security implications of running out of date code also cannot be ignored. Most customers understand this as it is typical that vendor support will advise customers to update to the latest software version before they are willing to even assist with a customer issue.

This issue directly relates to the separation of management and I/O stack in a storage platform. By moving management – the larger and more frequently changing part of a storage operating system – into the cloud and providing it as-a-service, customers can enjoy the benefits of always-up-to-date management software across their infrastructure and they do not need to worry about maintaining it – locally and across the enterprise.



Software and firmware updates are just a few of the repetitive and ongoing operations for maintaining infrastructure. Continuous performance, security, and capacity optimization are others. Today, many storage technologies built on cloud monitoring and analytics tools can alert customers regarding hardware issues, upcoming capacity shortages, or performance anomalies. Yet, for performing corrective actions, customers depend on isolated on-premises management tools. With a cloud control plane that combines monitoring, analytics, and administrative functions, customers gain a one-stop-shop for detecting issues and performing corrective measures with “single-click” simplicity.

A point that is important to remember is when deploying an infrastructure solution with fewer components or infrastructure layers, the monitoring and administrative complexity is drastically reduced. With fewer components which adhere to industry standard protocols, there are fewer things that can go wrong, fewer things that need to be administrated, and fewer compatibility problems. For storage, that primarily means abandoning the traditional 3-tier architecture setup with a server-based storage platform that does not require any software to be run on the server itself.

3. USE OF MODERN TECHNOLOGIES TO SIMPLIFY OPERATIONS

Using infrastructure telemetry for support and maintenance operations is an established practice across many storage vendors in the market. The collection of tens of thousands of metrics from storage, server, and applications enables use of cross stack analytics and provides support organizations with the necessary data to proactively support customers. However, not all of these cloud-based analytics platforms are created equally.

Some vendors upload diagnostic bundles a few times per day. More modern approaches stream in real-time to their cloud control plane where the customer has a central view of their infrastructure. **With all their information in one place, enterprises can capture deeper insights into their environment and start to answer the otherwise difficult questions they have about their infrastructure:**

- Which applications are performing the best or the worst?
- Where in the infrastructure are there resource constraints?
- When and where will applications run out of capacity?
- Where in my infrastructure are capacities for additional workloads?
- How are failures affecting applications?

The more up-to-date the information is within the cloud, the more effective and well-informed the decision making. Therefore, when comparing cloud-analytics offerings, customers should take into consideration how often information is updated in the cloud and how this information is used.

Furthermore, there are numerous tiers and components that can cause infrastructure issues, and it is undeniable that applications become more and more complicated, especially in a world of microservices and distributed applications at the edge. Each application has hundreds of knobs to tune with infrastructure devices that have a plethora of settings. With all of these added layers, customers can quickly get to a point where it is nearly impossible for a human to keep track of all the configuration parameters and statistics of each of these individual settings, or how to configure them for optimal results. A robust way to make sense of the data is through an analytics platform that helps customers identify issues from telemetry data.

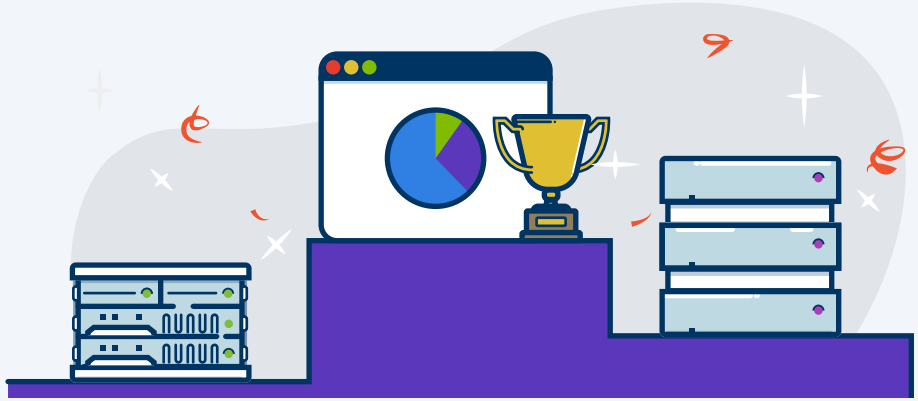
True cloud-managed approaches give enterprises the right tools to administer their infrastructure, benefit from cloud-based monitoring and analytics–including reporting capabilities–and helps customers troubleshoot performance or capacity issues. Leveraging a single tool for administration and analytics means a customer will not only be notified about these issues but will have the ability to immediately fix them in one place without having to utilize a different tool or console. Ideally, the cloud-based platform targets application teams and gives them the capability to self-service their infrastructure with a click of a button rather than relying on a team of storage expert.

4. API ACCESSIBILITY FOR AUTOMATION

Automation is a key step to achieving a self-service infrastructure, specifically when it comes to provisioning new infrastructure. Not every enterprise storage solution makes it easy on customers to implement automated workflows across their infrastructure. The key reasons are the lack of comprehensive APIs and simple API SDKs. In many cases, customers are building automation on top of command line interfaces, which are prone to human error and weakens the reliability of automation workflows.

When deciding on a self-service infrastructure deployment model, it is imperative to look for a solution which has been designed with automation in mind, meaning it is using an API-first design concept where there are no undocumented or 'secret' APIs and that all product capabilities are accessible to customers for automation through deterministic, versioned, and well-tested APIs; even better with convenience features provided by published SDKs in the preferred programming language.

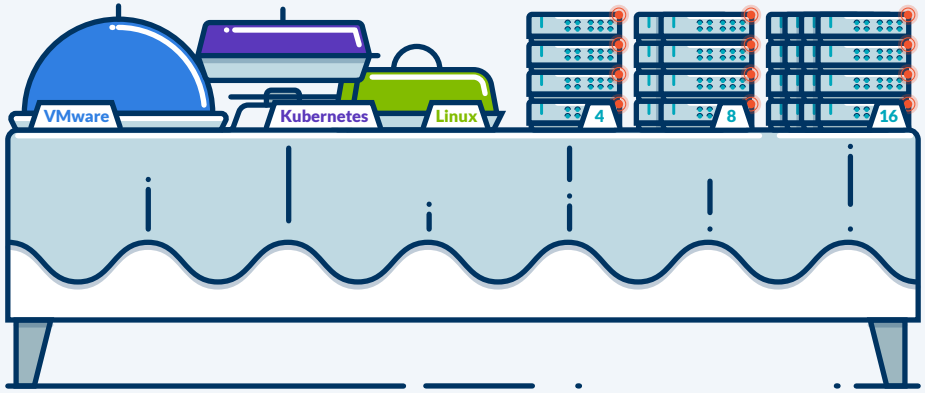
Automation is further simplified when provided through a single cloud-based API-endpoint. This means customers do not need to have an inventory of IP addresses or credentials of their systems, nor do they have the burden of querying every system in their datacenter running a different version of an



API. Instead, IT organizations can control every single system worldwide from the single point in the cloud which offers the latest and greatest automation capabilities.

But remember, while the primary consumers of these APIs will be developers and application owners, having an API does not remove the need for a graphical user interface (GUI). Users will want to ensure that the GUI is as simple to use as technically possible so that they can do it themselves (self-service) instead of relying on IT.

5. CLOUD-LIKE SIMPLICITY IN ON-PREMISES ENVIRONMENTS



When user's self-service provision their infrastructure, the expected experience is much like picking a service from an application catalog in the public cloud. Application and product teams want to make use of a customizable catalog of application templates that would automatically provision the right IT infrastructure for their own applications in their datacenters.

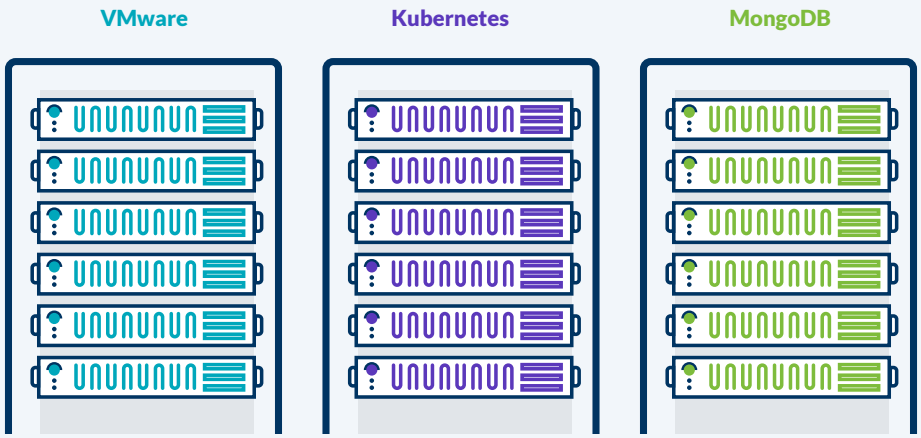
When considering a self-service infrastructure solution on-premises, customers should consider a solution that provides infrastructure templates that are constantly improved and optimized based on customer workloads. This means that when a customer deploys a new application or scales an existing application, the tailored template is used to modify the infrastructure for the individual application workload, removing the need for storage and broader infrastructure expertise.

NOTE

enterprises who are not able to fully control boot drive and boot drive images, will not be able to provision the full application infrastructure—a shortcoming of traditional storage and management solutions.

Infrastructure and operations teams will be the ones to define the catalog of service offerings and their policies. These include operating system, storage artifacts, and data protection levels. When consumed by application and product teams, they will consistently and reliably roll out infrastructure on their on-premises servers with great flexibility—one day an application may run on VMware vSphere, the next day the application may be containerized and run on Kubernetes. The application and product teams gain infrastructure agility which is achieved by just picking a different template and deploying.

6. OPTIMAL FOR MULTIPLE SOLUTIONS



Information technology is rapidly iterating and constantly evolving. The number of new software solutions, frameworks, and libraries that are available in this modern era are immense. While each technology may have different requirements for the infrastructure, developers and application teams will want to evaluate new technologies and assess their viability and benefits for their current and future projects. This is a common best practice and is how organizations are able to gain competitive advantages. These evaluations often times include technology which may not be formally supported by infrastructure and operations teams and is self-supported by the application and product teams.

It is imperative that infrastructure and operations teams make conscious decisions for their infrastructure that supports traditional and modern applications alike and allows stacking of technologies without enforcing proprietary restrictions. The infrastructure must be capable of supporting the needs of containerized applications, virtualized environments, and bare metal to prevent infrastructure silos, underutilization and sprawl; allow running generic software stacks; and to prevent complicating automation and promote self-service. By making a storage infrastructure choice that can support the needs of multiple software stacks, application teams can be assured that their infrastructure is ready for the application needs and that it operates consistently no matter where it is implemented – in the core datacenter or at the edge.

what does it all mean?

Modernizing your datacenter infrastructure and elevating it so that it delivers a cloud experience for application and product teams across the enterprise is not an easy journey. Such strategic investments require careful planning and must be architected to support the business long-term.

Determining the right storage infrastructure for an organization's datacenter, although difficult, has high returns when done correctly. Self-service infrastructure is a key enabler for IT organizations to quickly scale DevOps initiatives with predictable quality and to give companies a competitive edge. Enabling enterprises to make use of self-service for their workloads that cannot or should not run in a public cloud, allows product teams the ability to bring the simplicity and ease of the public cloud to their on-premises environment and ultimately deliver better products in less time.

The guidance in this document will help any organization determine where and how to invest time and money to improve agility for its on-premises IT infrastructure.

If you have questions or need more information, email us at info@nebulon.com and we'll get you going through the process of selecting the best self-service infrastructure for your on-premises environment and business critical workloads.

For more information on **smart**Infrastructure, visit **nebulon.com/product.**

nebulon®

smartInfrastructure