the GORILLA GUIDE® to...

Kubernetes

Native Application Mobility

Express Edition

DAN SULLIVAN

INSIDE THE GUIDE:

- Application Mobility Drivers
- Application Mobility Use Cases
- Veeam Kasten Application Mobility Solution

veeam

POWERED BY ActualTech MEDIA

THE GORILLA GUIDE TO...®

Kubernetes Native Application Mobility

By Dan Sullivan

www.actualtechmedia.com

PUBLISHER'S ACKNOWLEDGEMENTS

DIRECTOR OF CONTENT DELIVERY
Wendy Hernandez

CREATIVE DIRECTOR
Olivia Thomson

SENIOR DIRECTOR OF CONTENT
Katie Mohr

WITH SPECIAL CONTRIBUTIONS FROM VEEAM
Matt Bator, Principal, Kubernetes-Native Solutions Enablement
David Cackowski, Product Marketing Manager
Thomas Keenan, Sr. Product Marketing Manager

ABOUT THE AUTHOR

Dan Sullivan is a principal engineer and architect specializing in cloud architecture, data engineering, and analytics. He has designed and implemented solutions for a wide range of industries and is the author.

ENTERING THE JUNGLE

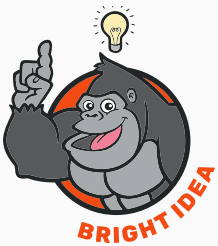
CALLOUTS USED IN THIS BOOK

SCHOOL HOUSE

The Gorilla is the professorial sort that enjoys helping people learn. In this callout, you'll gain insight into topics that may be outside the main subject but are still important.

FOOD FOR THOUGHT

This is a special place where you can learn a bit more about ancillary topics presented in the book.



BRIGHT IDEA

When we have a great thought, we express them through a series of grunts in the Bright Idea section.

DEEP DIVE

Takes you into the deep, dark depths of a particular topic.

EXECUTIVE CORNER

Discusses items of strategic interest to business leaders.

ICONS USED IN THIS BOOK

DEFINITION

Defines a word, phrase, or concept.

KNOWLEDGE CHECK

Tests your knowledge of what you've read.

PAY ATTENTION

We want to make sure you see this!

GPS

We'll help you navigate your knowledge to the right place.

WATCH OUT!

Make sure you read this so you don't make a critical error!

TIP

A helpful piece of advice based on what you've read.

Preserving the Benefits of Modern Computing

Modern computing infrastructure is a key driver to much of the innovation we benefit from today, but this infrastructure is increasingly complex, and it functions in a world that's constantly changing. Technologies and supporting practices, like containerization and orchestration, enable us to build increasingly complex systems with high availability and reliability. The ability to move applications around in the complex environments we've created is an increasingly important aspect of IT practices.

Applications today are not tied to a single server or a single logical storage volume. Applications run in a more abstract space of pods, namespaces, and clusters. Pods and other abstract resources may be managed based on attributes, such as labels that are used with affinity rules to ensure they run on appropriate nodes in a cluster. The Kubernetes way of managing resources is much more flexible and isolates developers from much of the responsibility for managing the state of compute and storage resources. Along with these benefits, there are some significant changes to how infrastructure and applications are managed. For example, we can no longer manage with an expectation of one application running on one server with a configuration that's fixed for extended periods of time.

The implications to managing applications are significant. We find ourselves managing the full lifecycle of applications from development and testing to production, scaling applications and resources

according to demand, and rapidly decommissioning applications and services when they're no longer needed. Add to the mix the fact that businesses run scalable applications across public, private, and hybrid clouds. These factors present significant challenges to maintaining highly performant applications, ensuring infrastructure, data, and applications are secure, and supporting DevOps practices that enable agile development and operations.

CHAPTER 1

Application Mobility in Kubernetes

Applications can be run on a variety of platforms, including on-premises, in a single public cloud, across multiple clouds, or combined in a hybrid environment. Applications are no longer tethered to a single set of servers or a single data center. Virtualization technologies enable more flexibility for running workloads in different environments, in different clusters, and on different infrastructures. Containerization has made application code more portable and has enabled cloud native applications. The ability to run applications in different configurations, clusters, and clouds is known as application mobility and is a crucial feature of modern computing and Kubernetes is a key technology that enables us to deploy and manage these applications.

Moving Application Stack Components

When we consider moving applications to different platforms, we must understand the components of an application and their dependencies. In the simplest case of an application that runs as a single binary, such as a utility running on your laptop, we can think in terms of copying the binary file to another system. Just slightly more complicated is an application that runs as a single binary that stores data locally. For example, if we need to move a simple office productivity application, we'd have to consider moving the binary image as well as

any files we've created. These simple scenarios provide examples of moving applications but do not capture the extent of the challenge of moving cloud-native applications.

Components of Applications

Modern applications are comprised of multiple services typically running in containers on Kubernetes platforms. The Kubernetes platform is often called an orchestration system for good reason. It's responsible for coordinating the operation of multiple components to achieve a state of highly available, scalable, and reliable application services. In addition to core application components such as binary images, cloud-native applications running in Kubernetes take advantage of other types of resources, including the following (see **Figure 1**):

- ConfigMaps to store non-secret configuration data about a specific environment.
- Secrets for securely storing credentials, API keys, and other confidential information that's needed by applications and service accounts.
- ReplicaSets, which define a set of pods that should be running at any time.
- Deployments are declarative updates to pods and ReplicaSets.
- Custom Resources are extensions to the Kubernetes APIs for defining a customization to the Kubernetes platform.
- Persistent Volumes are storage devices that will continue to exist independently of pods accessing the volume.

When we think about application mobility in Kubernetes, we need to account for the variety of resources and resource types that make up an application (see **Figure 1**). Each resource will have an associated configuration that defines the desired state for that resource. There will also be data stored on persistent volumes that may need to be accounted for.

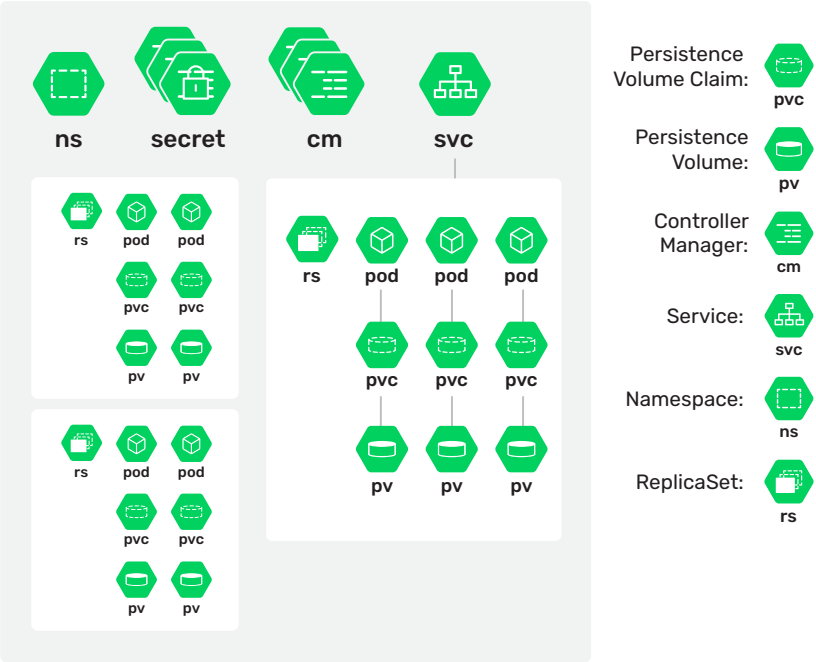


Figure 1: Components of modern applications

For example, stateful workloads, such as databases, require persistent storage and this creates additional challenges for application mobility. It's crucial to understand that application mobility in Kubernetes requires more than copying data and installing a binary image.

VARIETY OF APPLICATION SCOPES

The idea of an application scope is somewhat nebulous, but the idea is that applications can be set up and made available in different ways, for different purposes, and on different infrastructures. Let's consider several application scopes and how they relate to application mobility:

- Namespaces
- Clusters

- Accounts
- Regions
- Clouds

Each of these scopes represents distinct management domains.

NAMESPACES

A namespace in Kubernetes is a logical grouping for related resources within a cluster. For example, Kubernetes uses the kube-system namespace to contain services such as kube-dns and pods used to deploy kube-proxy.

Other namespaces are typically created to contain resources related to an application. An application that analyzes sensor data from a fleet of vehicles may be in one namespace of a cluster while an application that manages vehicle inventory of the fleet uses a different namespace within the same cluster. It would certainly be possible to have multiple instances of an application within a single cluster by using a namespace. We could, for example, deploy an application for developers to a namespace called fleet-sensor-dev while deploying the same application for production use to a namespace called fleet-sensor-prod.

CLUSTERS

Another application scope is the cluster. Clusters are a set of compute, storage, and networking infrastructure that is managed as a single unit. An application that runs in one cluster may need to be moved to a different cluster at some point in time. Moving an application from an on-premises Kubernetes cluster to a cluster running in the cloud is an example of application mobility at the cluster scope. The motivations for moving clusters are varied. The lease on your on-premises data center equipment may be expiring and you prefer to move your application to a public cloud provider rather than renewing the lease on your on-premises equipment. Also, you may want to take advantage

of specialized services, such as AI models, available in another cloud provider and the overall cost of operations may be less if you move your cluster to the same cloud as the specialized services.

ACCOUNTS

Scopes can be defined in terms of the infrastructure that the application runs on, but they can also be defined in terms of organization-based structures, such as accounts associated with different departments within a company. In this case, an enterprise may have multiple budget and billing accounts and an application may need to run on resources controlled by several different accounts.

REGIONS

Regions are geographic areas in which cloud providers operate multiple data centers. The data centers are typically grouped into geographically close zones. With such proximity within zones and regions, network latency within regions is minimized. It's common for workloads to use compute resources in the same region data is stored.

CLOUDS

As enterprises work to optimize their IT resources, they often find that a mix of on-premises and public cloud resources is the best option. Some workloads are best left on-premises while others can take advantage of specialized services, such as managed data warehousing or machine learning services available from a cloud provider. Working with multiple cloud providers also has advantages regarding mitigating the risk of vendor lock-in and having more robust disaster recovery options. Of course, these latter advantages can be realized only if applications can easily move across clouds as needed.

Just as these scopes have different characteristics, there are different reasons for wanting readily implemented application mobility.

Reasons for Implementing Application Mobility

Software engineers have developed an array of methodologies and practices to accommodate change. Requirements that seem essential at the start of a project may no longer be required after a change in business strategy or feedback from early adopters might highlight some missing features. The same kinds of uncertainty and change that drive the need to be adaptive to software development can also influence where and how we run our applications. There are many drivers behind the need for application mobility, which include changes to data sovereignty and avoiding vendor lock-in, but the three most common are disaster recovery, cost, and load balancing, which we'll cover in more detail in the following sections.

DISASTER RECOVERY

One of the advantages of cloud computing and modern application architectures is that they're designed for high availability. By using clusters of machines instead of a single machine, an application can continue to function even if a member of the cluster fails. If machines in a cluster are all located in the same data center and that data center loses all network connectivity or power due to a natural disaster, then any application running on that cluster will be unavailable. Disaster recovery planning can help us prepare for this kind of disruption, but the execution of such plans will depend on the ability to move applications quickly and reliably.

COST

The cost of running applications can vary depending on several factors, including the infrastructure used to run workloads, the efficiency of workload allocation, and egress charges for data copied from one cloud to another. The more experience we have running an application, the more likely we are to find ways to optimize the cost of running that application. For example, we may find that running a

data transformation service in the same region as we store its source data can eliminate egress charges for cross-region or cross-cloud data transfers.

LOAD BALANCING

Load balancing is the practice of allocating workloads to infrastructure with the capacity to run those workloads. Load balancing is common at multiple levels of application stacks, from implementing application programming interface (API) functions to full application-level load balancing. When our applications can be readily moved between different clouds or regions, we can ensure that we'll have the compute, storage, and networking resources we need to meet the demands of varying workloads.

Mobility, Modernization, and Service Transfer

Cloud computing and Kubernetes have prompted the development of terminology to describe ways in which we use these technologies. Some of these terms may sound synonymous but describe distinct concepts. Let's clarify a couple of terms that might be confused with application mobility.

Application Migration

Application migration can sometimes be used to refer to app modernization, in which we move applications from running on bare metal servers or virtual machines to running in containers. This kind of shift in platform allows us to take advantage of containers, which have less overhead than virtual machines and allow for more applications to run on a single physical server. Even more importantly, containers combined with Kubernetes orchestration provide the basis for implementing scalable, resilient, and highly available applications.

Service Transfer

Service transfer is another term you may come across when working with applications and related infrastructure. The term refers to shifting workloads from one platform to another. For example, from on-premises to a public cloud or from one public cloud to another. Application mobility technologies enable service transfer so in that way we can think of service transfer as a use case for application mobility.

Challenges of Application Mobility

Application mobility allows enterprises to be more flexible with how they run workloads, but that flexibility comes with some challenges that need to be addressed.

Security

When applications move, data moves with them. That means there will be times when data is not protected by application security controls, such as access controls on who can view or change data. It's imperative that data in transit be protected in ways that maintain confidentiality and integrity of the data. Using secure transfer protocols, such as Transport Layer Security (TLS), is a minimal requirement but we also need to ensure that access controls are in place when data is written to the target data store. Additional checks should be used to verify all data was correctly transferred and none was dropped in the process.

Moving an Application Stack Across Non-Federated Scopes

The complexity of moving an application increases with the number and kinds of differences between the source and target scopes. For example, an application that moves from one account to another account owned by the same team will require less coordination than moving to an account managed by a different division within the same company. Similarly, moving from one cloud to another cloud will

involve working with two sets of authentication mechanisms, billing accounts, and audit controls.

Automation: Scheduling and Failure Recovery

Application mobility can be a complicated operation, so automation is especially important. Automation allows for scheduling and reliably repeating a process. This is especially important when application mobility is used repeatedly, for example, to create development or test environments that are used for short periods of time and then shut down. Automation also helps recover from failures. Many things can go wrong when moving an application, including network disruptions, unavailable services, misconfigured servers, and other unexpected events. By designing automation to accommodate failures, we can more easily recover from those failures.

Getting application mobility right is important because there are multiple business interests that depend on it.

CHAPTER 2

Application Mobility Use Cases

There are several business needs that are well served by application mobility, including:

- Cross-cloud portability
- Multi-cloud load balancing
- Cluster upgrade testing
- Data management

This isn't an exhaustive list of use cases, but it is a representative sample and together these examples highlight a variety of business problems that are addressed using application mobility.

Cross-Cloud Portability

It wasn't long after enterprises started moving workloads into public clouds that concerns about vendor lock-in began to emerge. This is understandable, as customers of any service can benefit from the flexibility of choosing from multiple providers. When a cloud customer uses commodity services, like object storge or a set of virtual machines running a Linux distribution, it's fairly easy to move workloads from on-premises to the cloud or from one cloud provider to another.

Major cloud providers also provide managed Kubernetes services. While the open source version of Kubernetes will be essentially the same

across cloud providers, there may be features related to management and observability that distinguish one provider from another. Also, some vendors may offer related services, such as support for machine learning pipelines, which are beneficial to some users of Kubernetes.

As discussed earlier, modern applications are a complex set of services, configurations, and resources. Ideally, we would be able to move our modern applications running in Kubernetes to the platform that offers the best combination of services for a particular set of requirements.

Multi-Cloud Load Balancing

Even if we don't need specialized services offered by a cloud vendor, there may be other drivers that lead us to run our workloads in different clouds. Factors like the cost of running a workload, geographic location requirements, and other business issues may favor running some workload, or some parts of a larger workload, in a variety of clouds.

When this is the case, it's important to have tools and automation that can help us ensure we're running consistent environments across clouds. Continuous integration and continuous delivery (CI/CD) pipelines are standard tools to support the deployment of new features and updates to software. It's important to similarly support the consistent update and delivery of other components, such as reference data, access control policies, and other elements of modern applications.

Cluster Upgrade Testing

As new features are released in Kubernetes and other software, we typically test new versions before deploying them to production. Initially, developers and DevOps engineers may deploy and test new versions in sandbox environments to perform an initial evaluation.

Assuming those initial tests go well, and the new features are to be deployed to production use, it's a good practice to test in an environment that's similar to production. This means running the new

version on nodes similar to what's used in production as well as with data and configurations used in production. Software testing is difficult and often bugs are exposed only after the applications encounter some inputs and conditions that had not been anticipated. Working with production data is one of the best ways to understand how a new version of software will perform under expected production conditions.

Application mobility is a key enabler of realistic testing because it allows us to deploy production-like environments using production data without putting production services or data at risk.

Data Management

Data management requirements may dictate more about where an application is deployed than other factors. For example, applications that process personal identifiable information (PII) may need to be in a specific geographic region or encrypted with customer managed keys that may be available in limited environments. In addition to compliance and security concerns, there are performance benefits to having data close to compute resources. This reduces the distance data must travel and therefore reduces network latency.

With application mobility we can readily move applications as needed to comply with regulations and service-level expectations.

We've described the nature of application mobility in Kubernetes along with its challenges and benefits. Now it's time to consider features of application mobility tools that would allow us to realize the full extent of application mobility benefits.

CHAPTER 3

Veeam Kasten Application Mobility Solution

Application mobility solutions need to address several overall requirements, including:

- Ability to automatically import and restore applications
- Support for environment isolation
- Performant, reliable workflows
- Data security

Together, these features provide the foundation for application mobility solutions such as Veeam Kasten.

Automatic Import and Restore

We protect applications with backups and snapshots. These are created based on a policy that discovers the components to be protected and schedules the creation of the necessary backups and snapshots (see **Figure 2**). Snapshots are efficient ways to create copies of data without putting an excessive load on a system while providing for fast restore times and incremental data capture. A drawback of snapshots, however, is that they're not always durable. Since snapshots are typically stored on storage systems along with primary data, a failure results in the loss of both the primary data and snapshots. Backups complement snapshots by creating copies of data that can be stored in alternative locations. Backups include snapshots of data from application volumes along with metadata about the application.

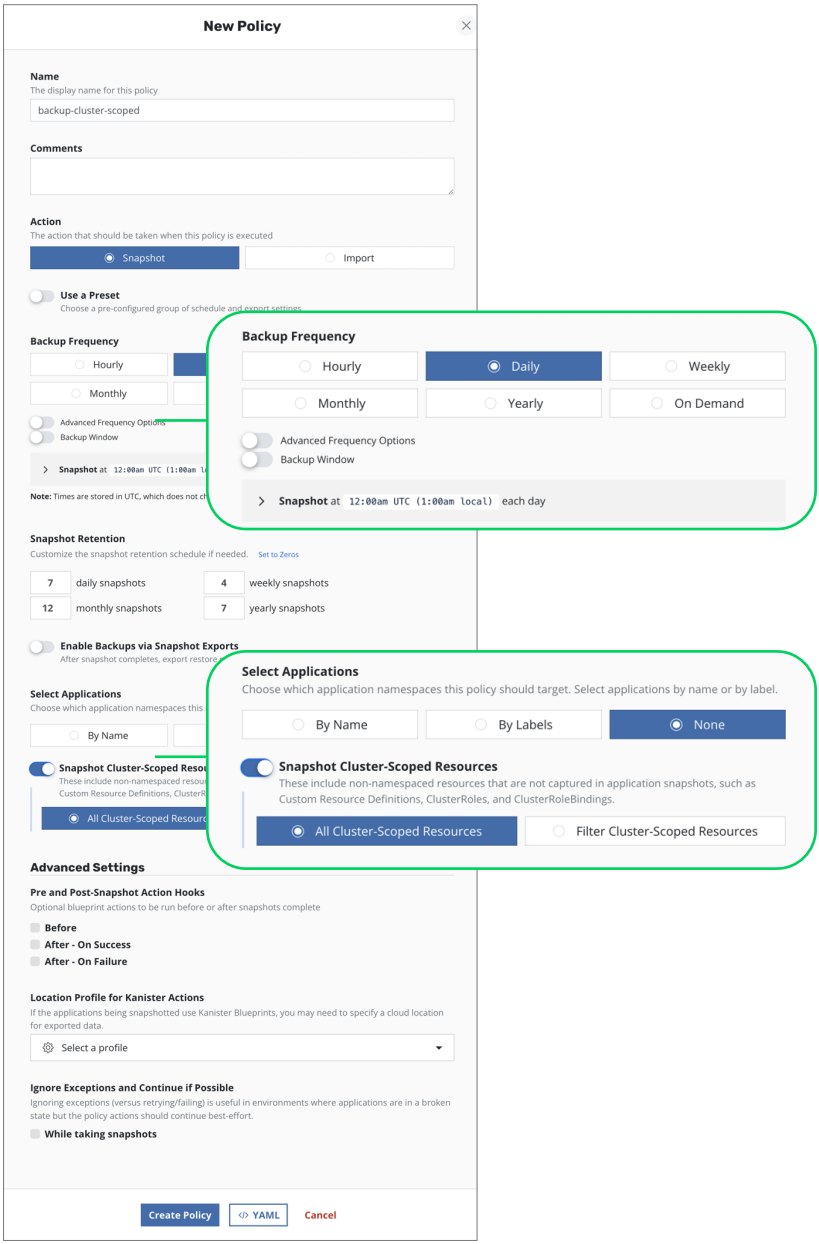


Figure 2: Policies define the contents and scheduling of snapshots and backups

The components that are included in backups and snapshots should be defined based on the needs of an application. They will typically include namespaced Kubernetes resources such as Deployments, StatefulSets, ConfigMaps and Secrets, as well as non-namespaced resources such as StorageClasses, CustomResourceDefinitions, Roles/ClusterRoles, and their associated Bindings.

Application mobility solutions must be able to manage multiple backups and snapshots. This would include support for a unified backup catalog, as well as role-based actions and self-service capabilities. They must also ensure data consistency since backup data is essential for application mobility. Application recovery capabilities must support coordinated recovery that allows for application-specific protection. For example, an application may include custom resources or have specific dependencies that constrain restore order.

Environment Isolation

The ability to move data and applications across different accounts, resource groups, and clusters is another essential feature of an application mobility solution. These capabilities build on the automatic import and restore features just described and allow for deploying an application and associated data in target environments outside the source environment. Since applications are being restored from backups, deploying additional instances of an application doesn't impact the performance of the primary workload.

Environment isolation is particularly important for disaster recovery. The nature of disaster recovery means that services that might be used for managing and moving workloads across federated clusters aren't available. Also, because of the impact of the disaster, data, applications, and other components will be deployed in other data centers, zones, or regions.

Typically, backup solutions restore resources from backups or snapshots as they existed in the source system. There are times, such as disaster recovery, when the target environment is different from the source environment. In those cases, it's important to be able to transform resources to accommodate the target environment (see **Figure 3**). For example, when moving an application from one public cloud provider to another, you may need to change storage class settings or specify new container image URLs. This requirement is addressed by Transforms, such as add, remove, copy, move, and replace operations, which in turn are organized into TransformSets, a custom resource for saving and reusing Transforms in restore operations.

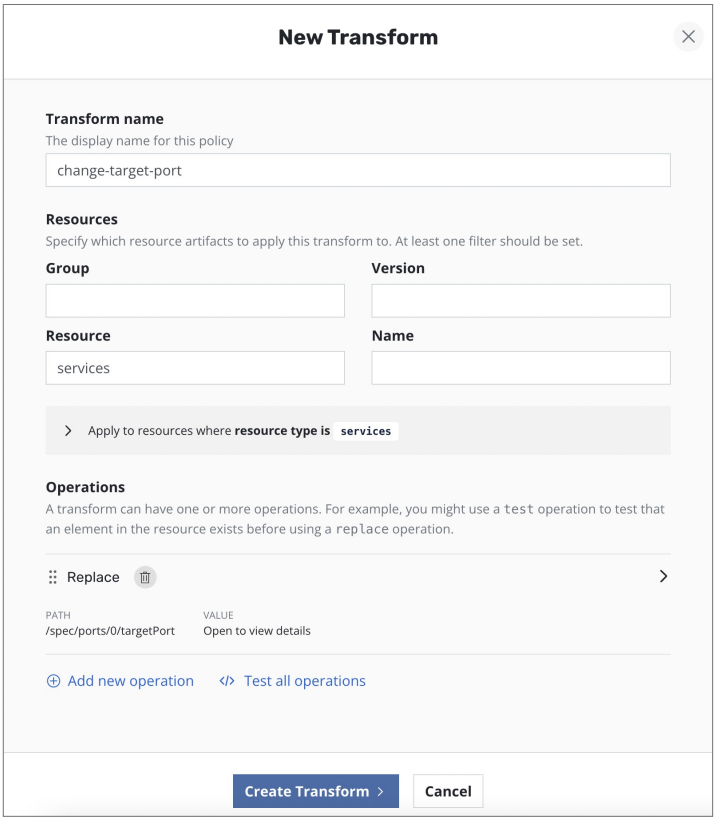


Figure 3: Resources may need to be transformed when an application is moved to a different environment

Performant, Reliable Workloads

Although there are many advantages to having the ability to move applications if mobility operations put significant additional load on primary systems, there is a cost that will constrain the use of application mobility. The ability to create and move snapshots efficiently and convert them to backups, such as done by Veeam Kasten, is a key enabler of performant application mobility operations.

In addition to the core backup operations, we also need reliable and automated workflows. API-driven automated workflows are recommended because they allow for rapid customization of workflows. Automation enables reliable application of data and application protection policies. For example, we could integrate backups with CI/CD to create backups immediately prior to deploying new code or with a developer portal to allow easy cloning to create development and test environments.

Data Security

When applications move, their data may not be protected by application security controls during the operation. For example, users of an application may be assigned roles and those roles have associated permissions, such as the ability to view, edit, or delete data. Application code can enforce the access rules defined by roles and permissions, but those controls aren't available when data is copied to snapshots and backups. It's important for mobility solutions to have sufficient mechanisms in place to protect the confidentiality and integrity of data in transit. This can include support for encrypting data and employing access controls of object storage systems that house backups.

10 Key Takeaways

Application mobility is an essential element of modern computing platforms. Here are 10 key takeaways to keep in mind as you consider the role of application mobility in your organization.

1. Applications are no longer tied to a single server, or a single logical storage volume. They are not monolithic but are comprised of multiple components. The dynamic demands facing enterprises are driving application mobility to help with cost control, load balancing, and disaster recovery.
2. Challenges of application mobility include security, moving across organizational scopes and boundaries, and the need for automation.
3. Application mobility solutions must include data protection mechanisms.
4. Partial solutions, for example those that move data but no other components, are insufficient to meet the needs of full application stack mobility.
5. Sufficient security controls are required to ensure the confidentiality, integrity, and availability of data and application resources.
6. Performance and efficiency are also key considerations, particularly the need to not place undue loads on source systems.
7. Automation of workflows is required to enable reliable, policy-driven application mobility.
8. Application mobility solutions should support the ability to transform components to accommodate differences in the way we configure and deploy applications.
9. Integrate backups into CI/CD pipelines and developer workflows to improve the reliability of these operations.
10. As with other enterprise solutions, ease of use is essential.

Learn more about how Veeam Kasten can protect your applications and data by viewing the January 2024 webinar, "Harnessing Application Mobility for DevOps Success." The Veeam Kasten data sheet has additional details. For more specifics on application mobility, read the Veeam blog.

ABOUT CLIENT

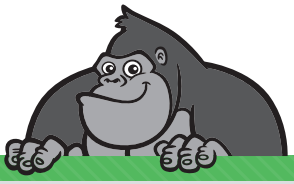
Veeam is the leader in Kubernetes Data Protection and Mobility. Trusted by the world's largest organizations, Veeam Kasten delivers secure, Kubernetes native data protection and application mobility, at scale, and across a wide range of distributions and platforms. Proven to recover entire applications quickly and reliably, coupled with its core tenet, simplicity, Veeam gives operations and app teams confidence to withstand the unexpected. For more information visit veeam.com or follow @Veeam on X.

ABOUT ACTUALTECH MEDIA

ActualTech Media, a Future company, is a B2B tech marketing company that connects enterprise IT vendors with IT buyers through innovative lead generation programs and compelling custom content services.

ActualTech Media's team speaks to the enterprise IT audience because we've been the enterprise IT audience.

Our leadership team is stacked with former CIOs, IT managers, architects, subject matter experts and marketing professionals that help our clients spend less time explaining what their technology does and more time creating strategies that drive results.



If you're an IT marketer and you'd like your own custom Gorilla Guide® title for your company, please visit https://www.gorilla.guide/custom-solutions/