Application Portability and Multi-Cloud environments

Application Portability is the process of moving working applications from one runtime environment to another. Portability differs from migration by being an ongoing process triggered multiple times and does not require fundamental change in the application architecture or configuration. Migration on the other hand is a one time process and usually entails significant reconfiguration or event re architecture of the software being migrated. Portability also means that moving the application does not require highly customized changes to the host environment.

Current business and technical landscapes mandate that DevOps teams operate in an agile manner and being able to port applications between environments in one such requirement. Below we describe several factors that might be drivers for portability.

Driving factors for application portability.

There are multiple factors that might force an IT team to move applications from one environment to another.

Avoiding Lock-In

Integration between applications and the host environment is important, however too tight integration and design without portability in mind can lead to rigid architectures which are hard to change. While public cloud providers compete for price and offered services, users should be flexible to take advantage of most cost effective offering and best of breed services without being locked in to a specific provider.

Cost optimization

One such factor is cost saving. Public cloud providers often compete on price for commodity services like storage and compute, which can have a big financial impact on their users. Being able to quickly move applications between providers to utilize the most cost effective resources can translate to significant cost savings.

Reduce latency to improve customer experience

Companies which run time-sensitive applications with geographically dispersed customers can benefit from migration of their applications to a data center, which is closer to their clients, thus reducing latency and improving the user experience in the region. Often applications require partial migration of only time-sensitive modules, which is exceedingly difficult to achieve in a disconnected host environment.

Technical factors - flexibility, rapid prototyping, testing

For technical reasons we want to be able to move applications from one environment to the other. Current state of the art development practices require this level of flexibility for development, testing, integration and production environments. Different teams and units within an organization may also require their own host environment.

Challenges

Existing architectures.

Often, existing application architectures are not designed with portability in mind. For example monolithic and legacy applications are very difficult to migrate, because of their complexity and strict runtime requirements. Such systems are designed for a specific target environment and are highly integrated with it. Being monolithic is a hurdle, because it makes it impossible to migrate just part of the application to another environment.

Non-standardized environments.

Applications often utilize non-standardized resources in their host environment, which might not be available elsewhere.

Synchronization of data and state.

Migration doesn’t necessarily mean that the new instance of the application is completely isolated. It often requires synchronization of data and state to other systems like ActiveDirectory or other modules of the same system running elsewhere. This is hard to achieve when different environments are loosely connected, because application communication must traverse multiple layers between environments like firewalls, ingresses and proxies.

Zero-downtime requirements

Some applications cannot tolerate downtime and must be ported in real time. This is a highly complex scenario in which user traffic must be constantly served by running application which it is being moved between host environments.

Daitera‘s solution for seamless application portability

Daitera’s unique approach to application portability provides a solution to the above mentioned challenges and makes the process of moving legacy and containerized workloads between local data centers, private and public clouds seamless and easy. We achieve this in two major ways. First, the environment creation process is fully automated and standardized, so that clients are able to quickly spin multiple identical environments on any target infrastructure. Second, Daitera creates highly integrated multi-cloud environments, making the access of resources, connectivity and migration within such environments transparent in regard to the underlying IT provider.

Environment automation

Daitera comes with built in automation blueprints, which make the creation of new environments very easy, quick and repeatable. Users select target infrastructure like on-premise VMs, public cloud providers or edge devices and Daitera automatically creates a fully functioning host environment in hours. A manual implementation of similar processes would usually require several months of work by highly skilled DevOps and Cloud professionals. Each environment is also standardized, meaning that it contains basic services required for the deployment of any use case like multi-cloud connectivity, routing, traffic encryption, authorization and authentication, monitoring and alerting.

Provider integration

Daitera’s multi-cloud environments feature a very high level of integration between cloud providers. On a network level, traffic between providers is routed directly via VPNs, so all resources and Virtual Private Clouds are reachable on their private address space. Daitera’s peer to peer VPN solution only leverages existing VPN services in each cloud provider, making it very resilient, flexible and secure.

Migration of legacy applications

When dealing with legacy applications, we often mean monolithic, highly coupled designs running on physical hardware or virtual machines. Daitera Multi-Cloud VM (DMCV) is a type of multi-cloud environment which connects VMs from different providers on their private addresses. Migration with DMCV is seamless, because you can move an application from a VM in one provider to an identical VM on another directly, without crossing cloud environment boundaries. Also, it enables migration of different modules of the application, while leaving the rest. Being within the same environment means that these modules will connect directly, without needing to change their underlying architecture.

Migration of containerized applications

Daitera's Kubernetes Engine (DKE) is a Kubernetes type of environment, where one cluster spans nodes running on multiple cloud providers and/or on-premise VMs. By having Kubernetes as the underlying platform, containers can be moved between clouds by only using the built in Kubernetes APIs without leaving the cluster. This removes the need to build separate CI/CD pipelines to target each provider separately, and also makes communication between different instances of the application very straightforward.

Dynamic workload management

DKE enables dynamic workload management. Since Kubernetes Nodes that belong to the same cluster are distributed between multiple cloud providers, users can target providers and provider specific services only by allocating Pods to run on specific nodes. This is achieved using K8S functionality like NodeSelectors and Labels. It is possible to target apps to run specifically on one provider (App 1) or to run on all cloud providers in parallel (App 2), or any combination of the two (App 3).

Think IaaS is right for you? Want to learn more about Daitera ?

Contact us for more info and to book a demo.