CONTAINERS IN THE MULTI-CLOUD

As more and more organizations utilize containers in their production environments, there is an increased demand for cloud infrastructures that support their use. Many companies have chosen to adopt multi-cloud strategies for a variety of reasons, and while this technology is highly beneficial, it is necessary to understand all of the benefits and limitations associated with containers in the multi-cloud.

What is a Container?

If a developer is creating an application on their laptop, this environment has a different configuration than their company's development and production environments, all reliant on specific libraries, dependencies, and files. They will want these environments to be similar as possible in order to ensure that the app not only works across all of them, but also passes quality assurance and gets deployed without major rewrites. Enter: containers.

One of the most basic ways to describe a container is that it is a method of operating system virtualization that allows you to ship a unit of software together. A container encompasses a set of one or more processes that are isolated from the rest of the system, making them extremely portable and consistent from development, to testing, to production. Containers help reduce issues between development and operations teams by separating responsibility; operations can focus on infrastructure and developers can focus on apps.

From a performance perspective, containers are actually a much better foundation for microservice

architectures than virtual machines. Largely in part to their small size, containers are also much more efficient at initialization than virtual machines as they don't require the operating system spin-up time. With all of this, containers are the perfect place to run microservice-based applications.

Multi-cloud

[Multi-cloud has a few definitions](#), but BMC defines it as follows: "Multi-cloud is a deployment model that involves using multiple cloud services from multiple public cloud hosting providers, often in combination with on-premises physical, virtual, and private cloud infrastructure."

This could mean something like two public cloud environments or two private cloud environments, each from various companies.

An advantage of having a multi-cloud environment is that your organization is able to utilize select services it prefers from whatever vendor or delivery platform makes the most sense, including things like the right blend of service-level agreements, uptime, or security protocols. Should certain features be requested that are only available from another vendor, these can easily be added and integrated into your custom app.

Benefits of Containers in the Multicloud

As mentioned in relation to microservices, containers are known for their agility as well as their portability. By building out data stores or services within containers, developers can more efficiently move some, or all, of them to the public cloud.

This portability is also beneficial for developers who want to test their applications across multiple operating systems to see if that makes a difference to any of the results. Should an application crash during testing, it only affects the one specific container versus the entire OS.

Through container clustering (combining multiple containers as a team of microservices), admins can more easily scale container systems while at the same time increasing their resilience. With this, individual services can be updated as necessary without having to take down the whole application.

The use of containers in the multi-cloud is also beneficial in terms of future innovation solutions for existing applications. Containers provide the ability to speed up innovation, [DevOps](#) implementations, and agility to application development.

Container Challenges

As with any solution, there are some limitations associated with the use of containers, especially when considering their relation to a big data cloud-container approach. Some of these include:

- Large data volumes that outgrow a container's capacity
- Exposure of critical data due to poor container security choices or implementations
- Slower performance levels as a result of higher dependency on network communications
- The effects of persistent storage versus in-memory storage and data caching
- Cost of large amounts of physical storage required for each instance that is created

Adopting a Container Management Platform

In order to ensure a smooth implementation process, it is highly recommended that organizations adopt a container management platform. Container management software is a large range of products that simplify how software containers are added or replaced on IT infrastructure. This software can automate the creation, destruction, deployment, and scaling of containers. A few of the most popular container management platforms include Kubernetes and Docker.

Kubernetes is a popular open-source container orchestration system for automating deployment, scaling, and management of containerized applications. Originally designed by Google, this software specializes in grouping containers that make up an application into logical units for easy management and discovery.

Docker, on the other hand, offers Swarm mode, a cluster management utility for containers. With Swarm, IT administrators and developers can establish and manage a cluster of Docker nodes as a single virtual system. It also offers other enterprise-focused container management software features, such as the Docker Secrets and Notary container security utilities. Learn more about docker monitoring.

Conclusion

Containers offer an easy way to build, package, and promote an application or service throughout its entire lifecycle and across multiple environments. While they do have some challenges, the benefits of containers in the multi-cloud guarantee that they are here to stay.

BMC TrueSight AppVisibility

BMC TrueSight AppVisibility fully supports monitoring Docker in many ways and could be the modern solution your organization needs. TrueSight AppVisibility monitors and manages packaged, mobile web, web-based, and modern application ecosystems, from the infrastructure and code to the customer experience.

Utilizing TrueSight AppVisibility, BMC is able to monitor performance and analytics within:

- Docker Hosts, Swarms, & Containers
- Applications running in Docker
- Log files running in Docker
- Docker in the Cloud
- End user experience within Docker

Monitoring Docker with BMC is simple, and will give you the peace of mind you need. BMC monitors all of your containers without having to change your Images or modify any existing Docker configurations. Truesight AppVisibility recognizes when new containers are added and automatically monitors the applications and services encapsulated in them.

Learn more about TrueSight Monitoring for Docker in these brief videos: