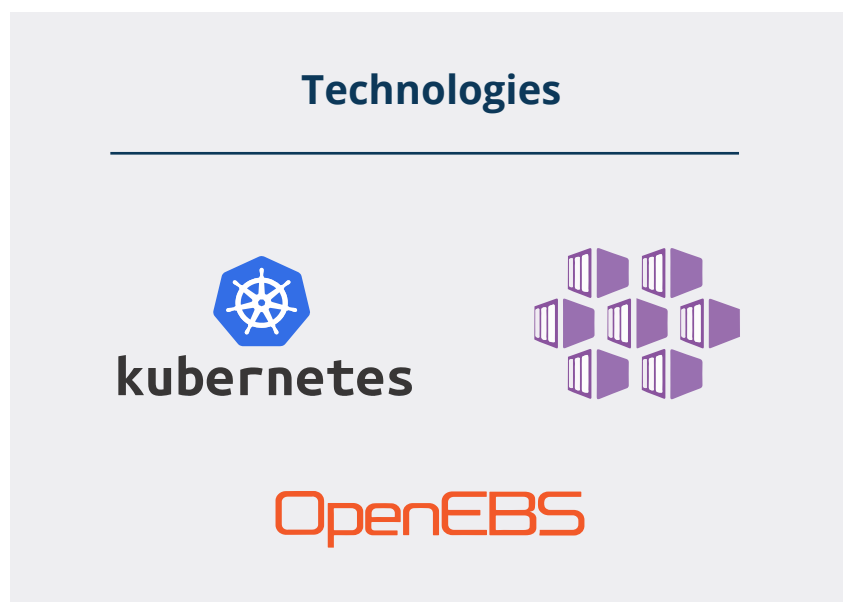


Global Insurance Kubernetes, Azure and OpenEBS

Summary

A leading insurance company has adopted Kubernetes on their premises via Azure connected cloud plus a Kubernetes distribution and on the cloud via Azure Kubernetes Service. OpenEBS is used across each of these Kubernetes environments to better perform CI/CD and other workloads in a manner that is consistent across environments.



"This universalization benefit of OpenEBS fit with a strategic mandate of the organization to reduce risks of vendor lock-in while also decreasing the low value-added customization work that would otherwise distract developers when they changed underlying cloud environments."

Background

This large insurance company competes globally in wholesale and retail markets. Software development is crucial for retaining and competitive advantages. Software is used in all pieces of the business from retail consumer apps to the financial modeling of risk exposures and regulatory reporting.

Data can be thought of as the “lifeblood” of the organization, used along with software to make more intelligent decisions in all pieces of the business from marketing to pricing and design of products.

While containers were used for stateless workloads before Kubernetes, using Docker Swarm primarily, it was not until the emergence of Kubernetes and Kubernetes services such as Azure Kubernetes Services that the organization felt that it was appropriate to begin to run stateful workloads on containers. The emergence of Kubernetes as a de-facto standard reduced some of the risks of being locked into a particular environment and otherwise increased confidence in the long-term viability of containers - now orchestrated by Kubernetes - as a fundamental piece of data pipelines.

Technology adoption

In 2017 the organization, which has a long-standing relationship with Microsoft, undertook a project to improve their development agility by accelerating their adoption of containers as a service for their internal teams. These efforts included:

- Started to move from Docker Swarm to Kubernetes
- Started to test some workloads on the then beta Azure Kubernetes Service
 - AKS was made generally available in June of 2018
- Began to run Kubernetes on premises on Azure as well with the help of an open source Kubernetes distribution
- Learned about OpenEBS and started to research it and other storage options

As of the summer of 2018, OpenEBS was being used across public cloud and on-premises environments for a variety of workloads. They selected OpenEBS to address a number of initial challenges in their increased usage of Kubernetes, including:

- Limitations in the performance and flexibility of underlying Azure storage, and Azure managed disks

- Azure managed disks are only available in a ratio of 2x to the number of CPUs; this fixed scaling ratio fits on a small subset of workloads.
- OpenEBS enables this ratio to be far larger, improving resilience while reducing costs
- Technical and performance limitations of Azure file, which is limited with SMB protocol only and can be tricky to manage. OpenEBS provides block storage available on the network via iSCSI protocol and NFS service, protected with OpenEBS replicas, can be easily provisioned. Also costly Azure file sync service, which is also not available in every region, can be replaced with simple rsync.
- Azure managed disks at times also have some challenges with attaching and detaching disks. Users frequently experience “Failed Attach Volume” and “Failed Mount” errors. Due to 1-to-1 relationship, disks cannot be detached or unmounted from the node, also new nodes may run out of mountpoints, if they already have too many Azure disks. In addition to that, API call may also fail. For example, a disk would appear as free when it was actually not yet detached.
- Limitations in the flexibility of ElasticSearch services from Azure, which led them to want to run their own ElasticSearch, a common workload for OpenEBS.

Additionally, they have found that OpenEBS enables them to abstract away from the underlying namespace limitations and other assumptions of their on-premises and in the cloud deployments so that they can continue to use their existing approach to CI/CD.

Lastly, they have found that OpenEBS help to provide a common abstraction layer so they can more easily move workloads in part thanks to the ability to now deploy essentially identical application stacks irrespective of location. According to the lead engineer on the project:

“This universalization benefit of OpenEBS fit with a strategic mandate of the organization to reduce risks of vendor lock-in while also decreasing the low value-added customization work that would otherwise distract developers when they changed underlying cloud environments.”

Areas of improvement

The upgrade to OpenEBS 0.7 from 0.6 was not entirely smooth, thanks in part to an apparently minor problem in 0.7 having to do with the interpretation of volume sizes in YAML. While no data was lost, some time was lost in troubleshooting this issue.

OpenEBS is working to improve the automated management of underlying disks so that more workflows can be entirely automated. For example, by extending the OpenEBS dynamic provisioner with the help of emerging Node Disk Manager capabilities, it will be possible to have underlying disks from Azure claimed, pooled, presented and managed without any manual steps. These requirements are being worked on for OpenEBS 0.8 and 0.9.

Conclusion


A forwarding looking very large insurance company has successfully proven the advantages of moving towards Kubernetes plus OpenEBS as a common abstraction layer running across on premises and on cloud environments.



 linkedin.com/company/mayadata/

 medium.com/mayadata

 facebook.com/MayaDataOfficial/

 [@mayadata_inc](https://twitter.com/mayadata_inc)

www.mayadata.io



 linkedin.com/company/openeps/

 medium.com/openeps

 facebook.com/openeps/

 [@openeps](https://twitter.com/openeps)

www.openeps.io

