Why Edge Computing Needs Container-Native Data Storage

Over the last decade, enterprise adoption of public-cloud services has increased significantly. Mainstream public-cloud platforms like AWS, Microsoft Azure, and Google Cloud Platform have become extensions of the enterprise data center.

Although the public cloud offers a cost-effective model for storage and compute, it's not the answer for all workloads. Some workloads need a faster turnaround time while others have to comply with data sovereignty policies. Healthcare and financial organizations, for example, deal with sensitive data for which they may need local processing when using public-cloud services.

The answer: container-native data storage.

Benefits of Edge Computing

In edge computing, compute moves closer to the source of data. Enterprises are taking notice of this new paradigm for several reasons. Edge computing:

- Addresses many challenges you face when running data-centric workloads in the public cloud
- Reduces the amount of data that flows back and forth between the data center and the public cloud

- Enables you to retain sensitive data on-premises while still taking advantage of the elasticity of public cloud
- Reduces the latency involved in dealing with public-cloud platforms

Hyperscale-cloud providers such as Amazon, Google, and Microsoft have invested in edge computing platforms, making them an extension of the public cloud. The core building blocks of the cloud are compute, storage, and network. They now seamlessly extend to the edge to deliver a consistent experience to developers and ops teams. The edge computing layer mimics the cloud by exposing the same services, APIs, and programmability aspects of compute, storage, network, and databases.

Containerization and Kubernetes

Containerization is emerging as the standard for running modern workloads. And Kubernetes is the preferred platform. Managed Kubernetes offerings have become the fastest-growing service for public-cloud providers. Kubernetes is an ideal platform for the edge due to the automation, extensibility, and standard workflows it offers for deploying applications. Edge platforms such as AWS Snowball, Azure Stack, and Google Anthos are based on Kubernetes. These environments run data ingestion, data storage, data processing, data analytics, and machine-learning workloads at the edge.

If you're deploying Kubernetes at the edge, you face the same challenges whether you're it in the public cloud or a data center. To ensure high availability of workloads, many IT teams deploy multi-node Kubernetes clusters. These clusters run on inexpensive hardware such as Intel NUC or Zotac Mini PC. Each node of the cluster runs both the control plane and the worker node components to maximize efficiency. Red Hat has a reference architecture of OpenShift running at the edge based on a three-node cluster.

A Container-Native Storage Engine

Storage and data management are key to powering data-centric workloads running at the edge. These workloads typically have:

- A time-series database such as InfluxDB
- Object storage based on MinIO
- A NoSQL database for device metadata
- A shared storage layer for hosting the machine-learning inference engine

Each stateful workload has diverse characteristics and storage needs, which makes it challenging to run in the context of Kubernetes.

A multi-node Kubernetes cluster running at the edge needs an efficient, container-native storage engine that caters to the specific needs of data-centric workloads. This storage engine acts as the foundation for streaming data, object storage, unstructured data, and machine-learning inference. IT teams will be able to create different volumes that are closely aligned with the characteristics of individual workloads.

As well as providing the core storage services and persistent storage, the container-native storage engine should also ensure high availability, durability, and security of data. Each block of data written to any node in the cluster should be automatically replicated to other nodes to ensure high availability. Through workload-aware backups, the storage engine should be able to periodically take application-consistent snapshots and store them in the cloud. And all data should be encrypted with keys to ensure bad actors can't make off with proprietary information or customer data.

Run Advanced Workloads with Container-native Storage from Portworx

As you run advanced workloads at the edge, container-native storage can help you migrate seamlessly to the public cloud or even to another enterprise data center. You'll be able to move applications between the cloud and edge with minimal effort. Companies also need a storage solution that provides support for disaster recovery. The storage should deliver application business continuity through failover and failback to other edge deployments or cloud environments.

Portworx® by Pure Storage is an enterprise cloud-native storage platform ideal for running stateful workloads in the cloud or at the edge. Through tight integration with Kubernetes, it's available on Anthos, Azure Stack, AWS Outposts, K3s, and Red Hat OpenShift.