

PureAccelerate TFDx

\*Cloud Block Store

Kunal Kapoor - Director, Product Management

Naveen Neelakantam, Software Architect

Van Phan, Technical Marketing Engineer

\*Kunal

Cloud Vision

Effortless from Day 1 - FA Getting Started

Cloud divide

On-premises and hosted versus public cloud

- different application architectures
- different management and consumption experience
- different storage

What if?

Pure could build application portability?

Common shared data services

Three Pillars

- Build Your Cloud
- Run anywhere
- Protect everywhere

Cloud Block Store for AWS

- high reliability, efficiency
- hybrid mobility and protection
- consistent APIs, automation

100% software - deploys instantly as a virtual appliance in the cloud, runs only as long as you need it

Efficient - deduplication, compression, and thin provisioning deliver capacity and performance economically

Hybrid - easily migrate data bidirectionally, delivering data portability and protection across your hybrid cloud

Consistent APIs - developers connect to storage the same way on-premises and in the cloud. Automated deployment with Cloud Formation templates

Reliable, secure - delivers industrial-strength performance, reliability & protection with Multi-AZ HA, NDU, instant snaps and data at rest encryption

Flexible - pay as you go consumption model to best match your needs for production and development

Use case: DR

- production instance on-premises
- replicate data to public cloud
- fail over in DR event
- fail back and recover

Use case: Lift and shift

Use case: Dev / test

- replicate data to public cloud
- instantiate test / dev instances in public cloud
- refresh test / dev periodically
- bring changes back on-premises
- snapshots are more costly and slower to restore in native AWS

Use case: ActiveCluster

- HA within an availability zone and / or across availability zones in an AWS region (ActiveCluster needs <11ms latency)
- no downtime when a Cloud Block Store Instance goes away or there is a zone outage
- Pure1 Cloud Mediator Witness (simple to manage and deploy)

Use case: Migrating VMware environments

VMware Challenges

- AWS does not recognise VMFS
- Replicating volumes with VMFS will not do any good

Workaround

- convert VMFS datastore into vVOLs
- now each volume has the Guest VM's file system (NTFS, EXT3, etc)
- replicate VMDK vVOLs to CBS
- Now the volumes can be mounted to EC2 with matching OS

Note: This is for the VM's data volumes. The VM boot volume will not be usable in AWS. VM's application will need to be redeployed in native AWS EC2.

Use case: VMware Cloud

VMware Challenges

- VMware Cloud does not support external storage, it only supports vSAN

Workaround

- connect Guest VMs directly to CBS via iSCSI

CloudSnap

CloudSnap to Azure

CloudSnap to AWS S3

Snap to NFS - offload snapshots to an NFS server for inexpensive long term retention

CloudSnap - natively move snapshots to the cloud for backup and compliance

Self-describing snapshot - data and metadata embedded in it

Use case: Purity CloudSnap restoration as DR alternative

- replicate data to public cloud
- instantiate test / dev instances in public cloud
- refresh test / dev periodically
- bring changes back on-premises

\*Architecture

Naveen

### Design Goals

- highly-available and predictable block storage
- located within a single AZ
- APIs and functionality consistent with a FlashArray (including snapshots and replication)
- cost comparable with EBS

### FlashArray Architecture

- HA architecture
- NVRAM write buffer - fast writes
- Flash module - bulk storage, fast reads

### Native Solution: Simple Port

- run Purity in two EC2 instances
- use EBS volumes instead of flash volumes
- Not HA - EBS attaches to a single EC2
- Cannot control placement - RAID may not help

### \*Cloud Block Store Architecture

EC2: CBS Controllers

EC2: Virtual Drives

Virtual Shelf: 7 Virtual drives in Spread Placement Group

EBS IO1: NVRAM, Write Buffer (7 total)

S3: Durable persistent storage

Instance Store: Non-Persistent Read Mirror

[architecture photo]

### Fault Tolerance

- designed to tolerate component failure
- example: self-healing drives
- EC2 auto scaling
- High bandwidth restore from S3

Use DynamoDB to keep track of state

### NDU - Purity Upgrade

- Fault tolerance is basis of NDU
- rolling upgrade without disruption
- evergreen - evolves along with AWS

Use Case: Same app deployment across hybrid cloud

- seamless application portability
- easy migration to / from cloud
- same deployment yaml

### Pure Service Orchestrator

- automated container storage, on demand
- scale across multiple arrays seamlessly
- file and block on shared infrastructure
- automatic failover for unhindered service

### Transparent Recovery

- automatic failover
- enterprise reliability
- data access integrity

### Elastic Scaling

- scale across multiple failure domains
- scalable cluster size
- scale across multiple protocols
- 1 command to add new storage array

### \*Procurement & Deployment

Van Phan

### Procurement Options

#### A - Pure as-a-Service

- offered via SLED / CLED process
- minimums 100TiB effective used capacity
- unified hybrid contracts (on-premises and CBS, CBS)
- 1 year to 3 year contracts

#### B - AWS Marketplace

- direct to customer
- minimum, 10 TiB effective used capacity
- CBS only
- month to month contract or 1 year contract

Procurement process - you end up with a license key

Deployment process

- deploy using AWS Marketplace
- AWS Marketplace will trigger AWS CloudFormation (CF)
- CF will automatically provision the AWS resources to spin up new CBS instance

yaml file to deploy everything using CloudFormation

Takes about 10 minutes to deploy

### Management

\*Demos