SDC

STORAGE DEVELOPER CONFERENCE

SNIA SANTA CLARA, 2017

# Breaking the Cloud Storage Chains

**Gregory Touretsky**
**INFINIDAT**

# Company Overview

**Moshe Yanai**
Founder and CEO

400+ highly skilled storage professionals

Storage industry veterans

1,700 PB+ deployed World-wide

125+ filed patents

# InfiniBox™

## Performance Edge
1M+ IOPS @ 130 / 180 µs latency
12GB+/sec throughput

## Multi Petabyte Scale
Scales from 115TB to > 5.5PB of effective capacity in a single 42U rack

## Solutions
Rich integrations with leading business platforms and applications

## Green Storage
Superior power efficiency
8KW Max - less than 2W/TB

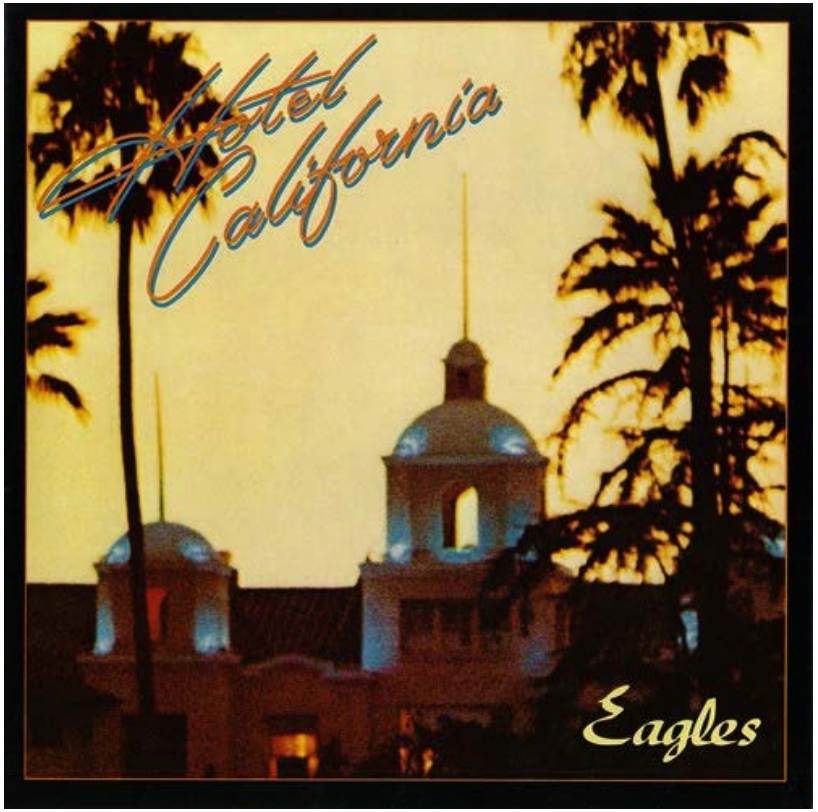

## Unmatched Reliability
Unique architecture enables rock-solid uptime of 99.99999%

## Unified Storage
Truly unified storage built from ground-up for multi protocols


## Easy
Simple & powerful management interfaces powered by HTML 5& REST

## Disruptive Pricing By Design
Order of magnitude more usable capacity and performance per dollar

"You can check out any time you like, but you can never leave"



# Cloud storage challenges

- Migration effort for legacy applications
  - Cloud storage > S3
- Getting data to the cloud
- Cost
  - Cloud storage
  - Egress
- Data sovereignty


**AWS EBS Storage**

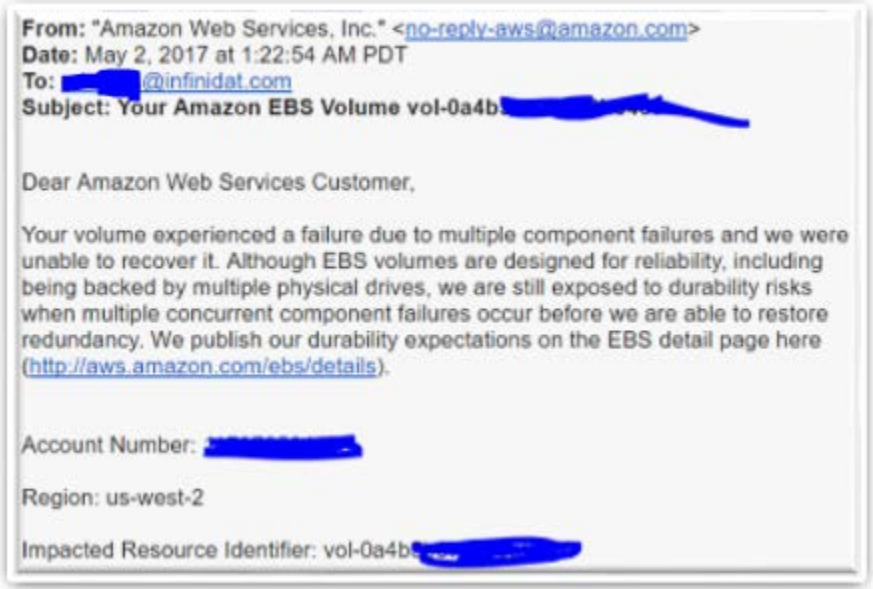
| | |
|---|---|
| • 16TB Storage Cost = | $2,208 |
| • 20k IOPS Cost = | $1,440 |
| • 1st Snap Cost = | $880 |
| • 16TB Cloud-to-DC = | $1411 |
| Total Monthly Cost | **$5,939** |





# Cloud storage challenges, 2

- Availability
  - EBS: 99.999%
- Reliability
  - EBS: AFR 0.1-0.5%
    - 1-5 failed volumes out of 1,000 per year – for 20GB
    - Size dependent



**From:** "Amazon Web Services, Inc." <no-reply-aws@amazon.com>
**Date:** May 2, 2017 at 1:22:54 AM PDT
**To:** [redacted]@infinidat.com
**Subject:** Your Amazon EBS Volume vol-0a4b[redacted]

Dear Amazon Web Services Customer,

Your volume experienced a failure due to multiple component failures and we were unable to recover it. Although EBS volumes are designed for reliability, including being backed by multiple physical drives, we are still exposed to durability risks when multiple concurrent component failures occur before we are able to restore redundancy. We publish our durability expectations on the EBS detail page here (http://aws.amazon.com/ebs/details).

Account Number: [redacted]

Region: us-west-2

Impacted Resource Identifier: vol-0a4b[redacted]

https://aws.amazon.com/ebs/details/

# Data protection?

- AWS EBS volume: S3-based snapshot

```
$ time aws ec2 create-snapshot --volume-id vol-02eab175ada257e45
real    0m0.458s

$ while [ 1 ]; do date;aws ec2 describe-snapshots --snapshot-ids snap-
0da5ec7ef918ca5c3 | grep Progress; sleep 30; done
"Progress": "0%"
"Progress": "0%"
"Progress": "0%"
...
"Progress": "0%"
"Progress": "100%"
```

50GB volume, 4GB used

Create-snapshot: 01:27:48

4 minutes to create a snapshot

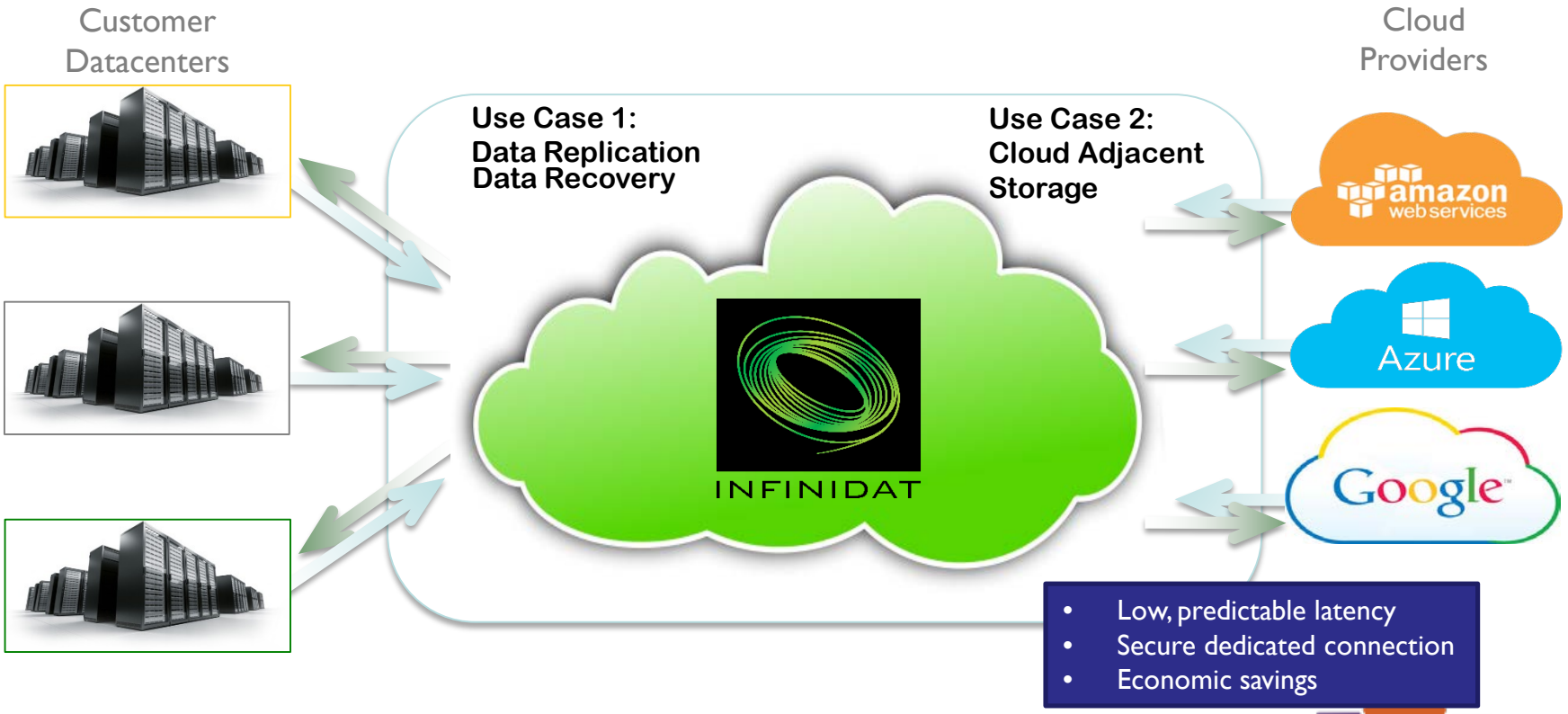
Status completed: 01:31:51

- AWS EFS file system: …

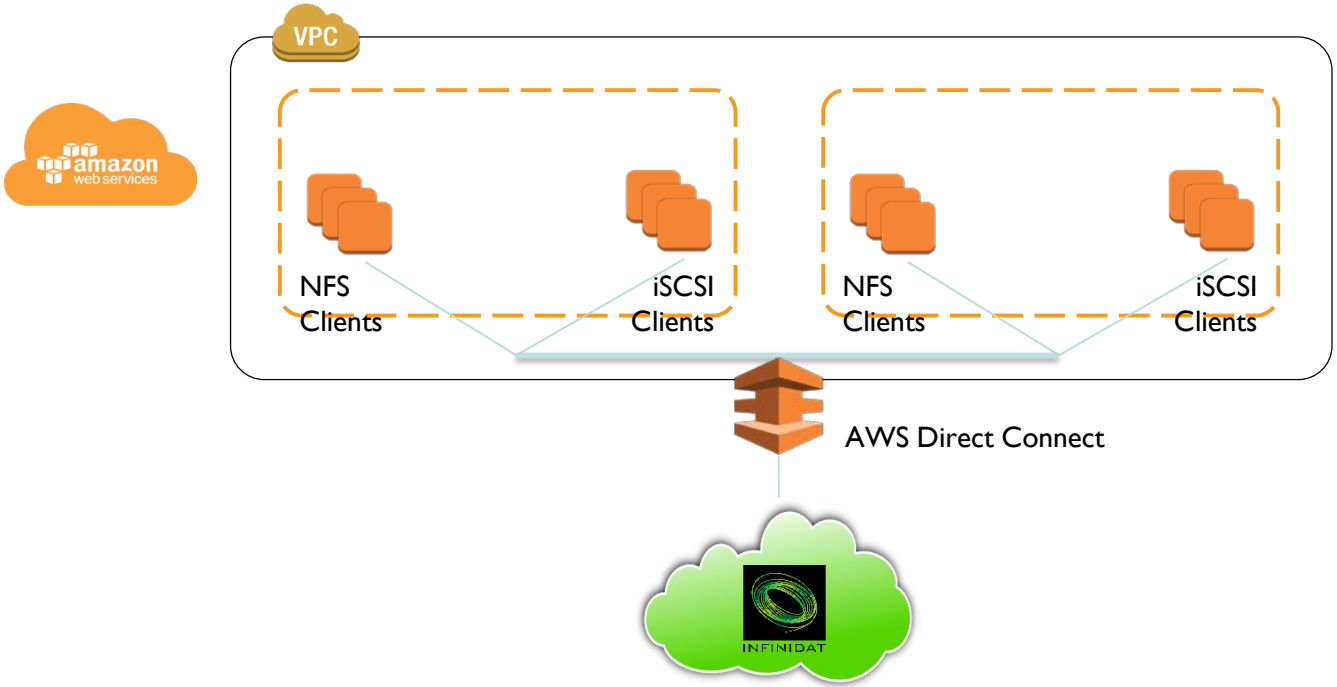




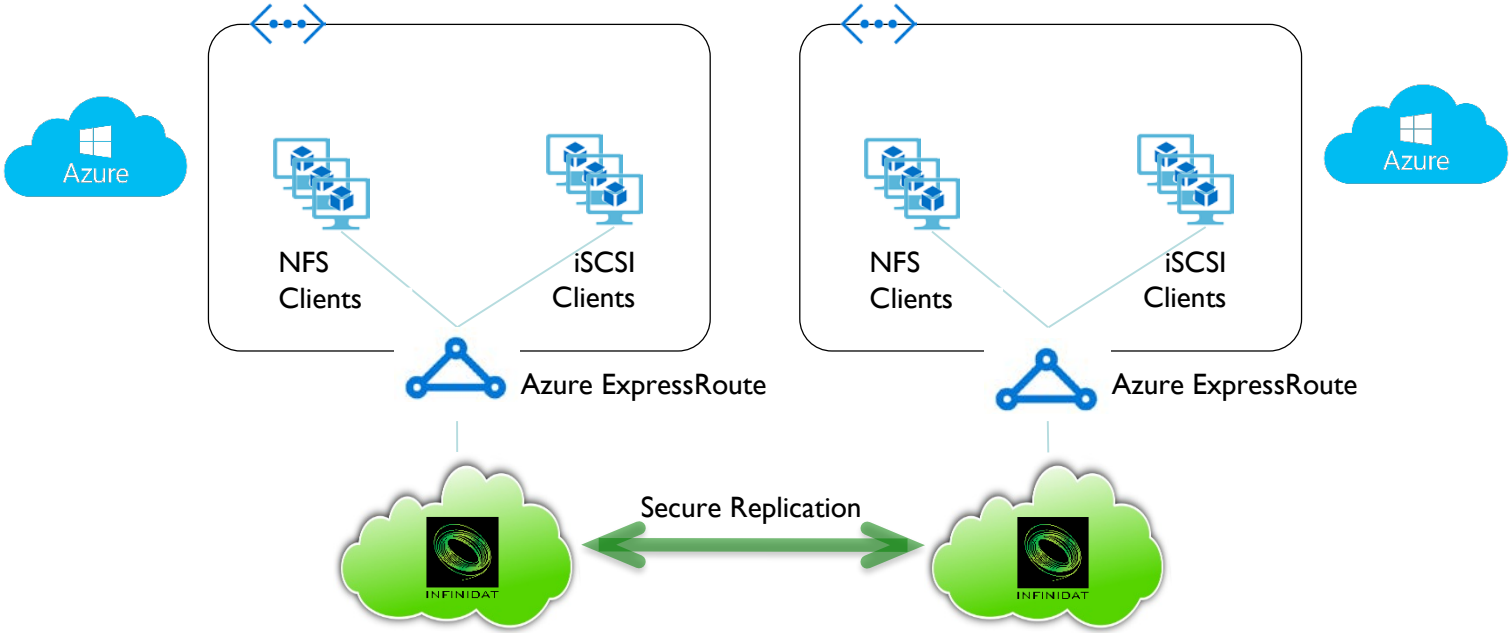
# Meet INFINIDAT Multicloud Storage



# Single region / Multiple AZs



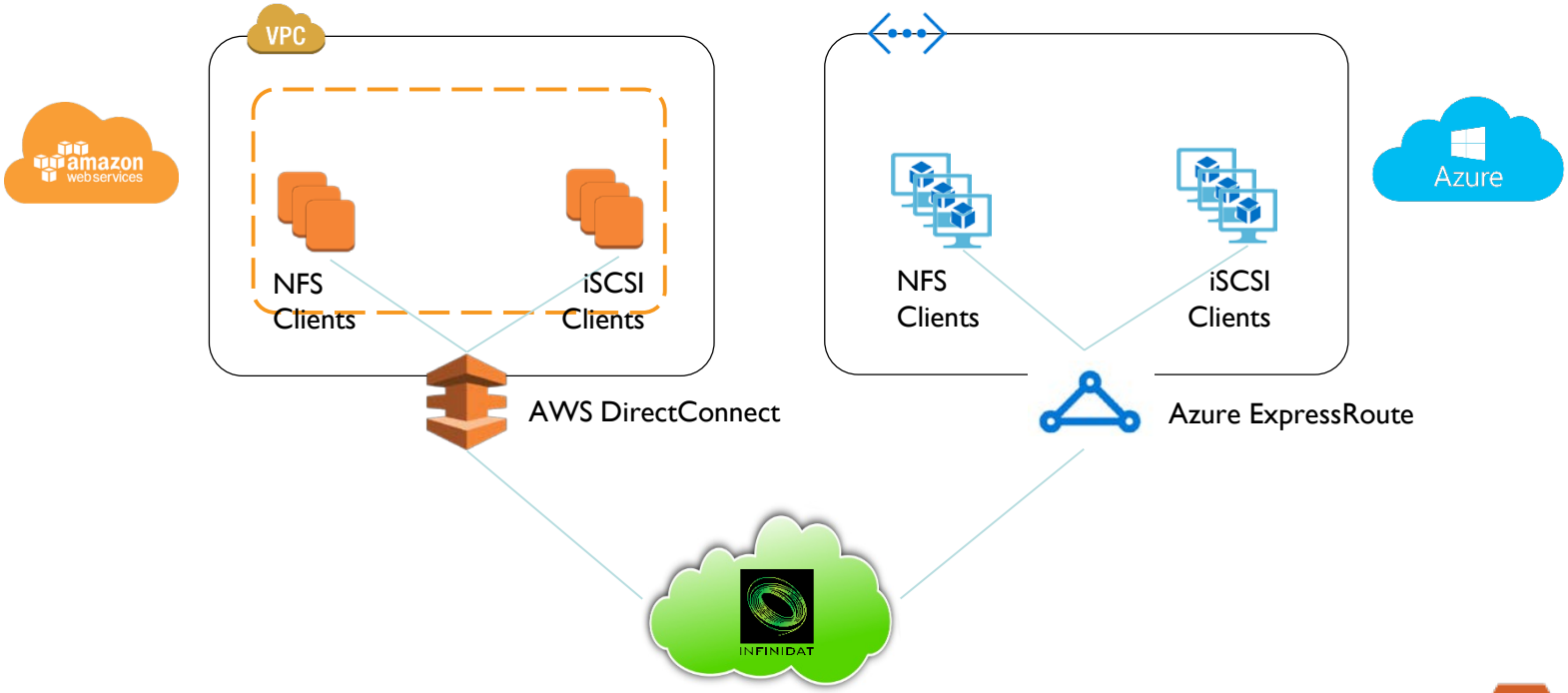
# Multiple regions



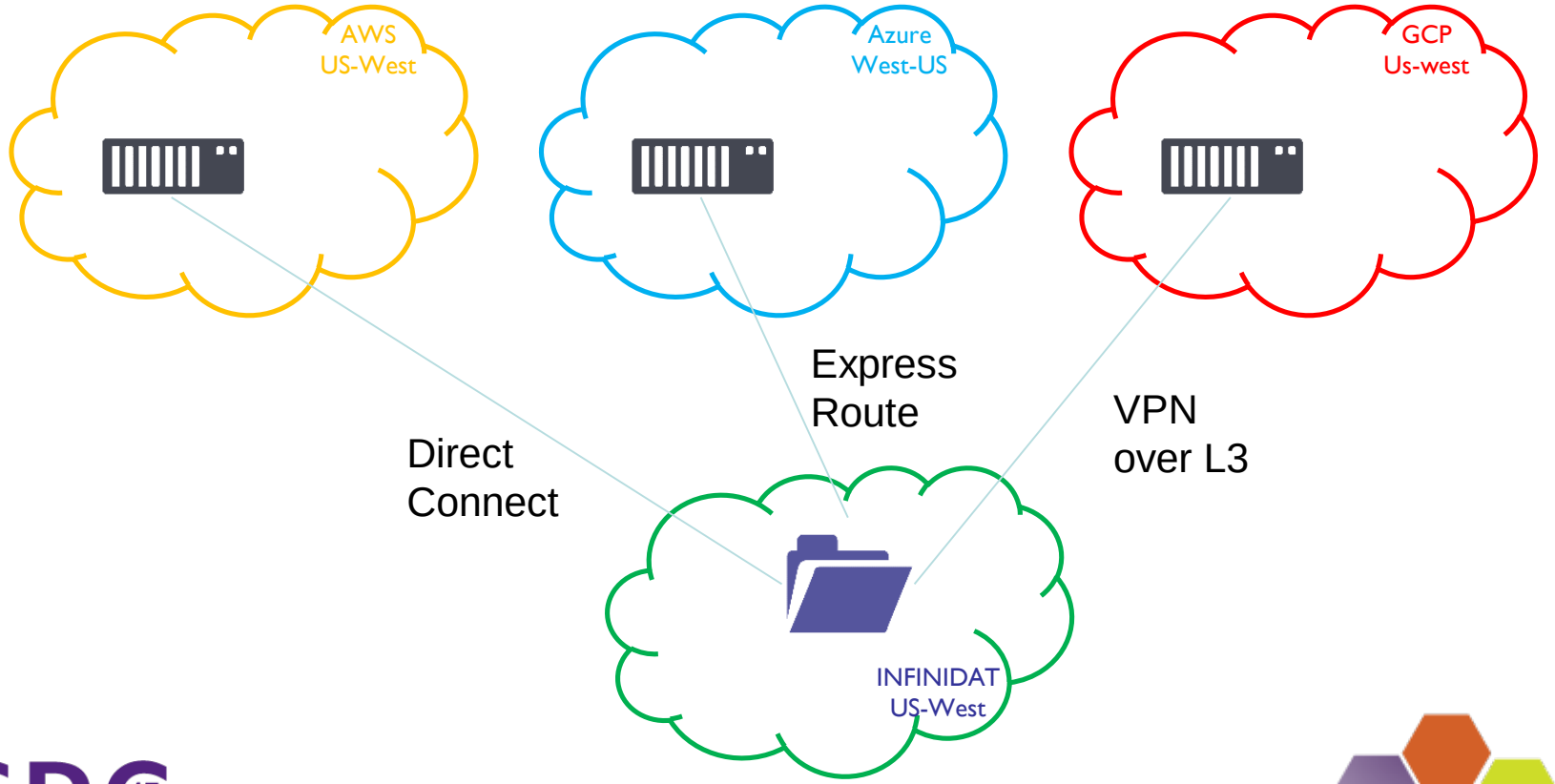





# Multiple clouds



# Demo: shared file system



# Create a file system

```
IMS> fs create --name fs02 --size_gb 100
Create fs fs02 succeeded.
IMS> export create --exportpath /fs02 --fs fs02
Create export /fs02 succeeded.
IMS> export update --exportpath /fs02 --op add --optype permission --client
10.11.0.0-10.11.0.254 --access RW --no_root_squash
{'access': 'RW', 'no_root_squash': True, 'optype': 'permission', 'exportpath':
'/fs02', 'client': '10.11.0.0-10.11.0.254', 'op': 'add'}
Update export /fs02 succeeded.

$ sudo mount 10.50.10.51:/DemoInc_gtouret/fs02 /mnt/tstMnt
$ df /mnt/tstMnt
Filesystem                          1K-blocks  Used Available Use% Mounted on
10.50.10.51:/DemoInc_gtouret/fs02 104857600  1600 104856000   1% /mnt/tstMnt
```





# Shared access

```
ubuntu@ip-10-11-0-23:/mnt/nfsLeitrim$ ./getCloudInstance.py
userName        hostName        privateIP       provider        AZ              instanceType
ubuntu          ip-10-11-0-23   10.11.0.23      AWS             us-west-1c      t2.micro

ubuntu@ip-10-11-0-23:/mnt/nfsLeitrim$ while [ 1 ]; do echo AWS `date` >> file; sleep 1; done
```

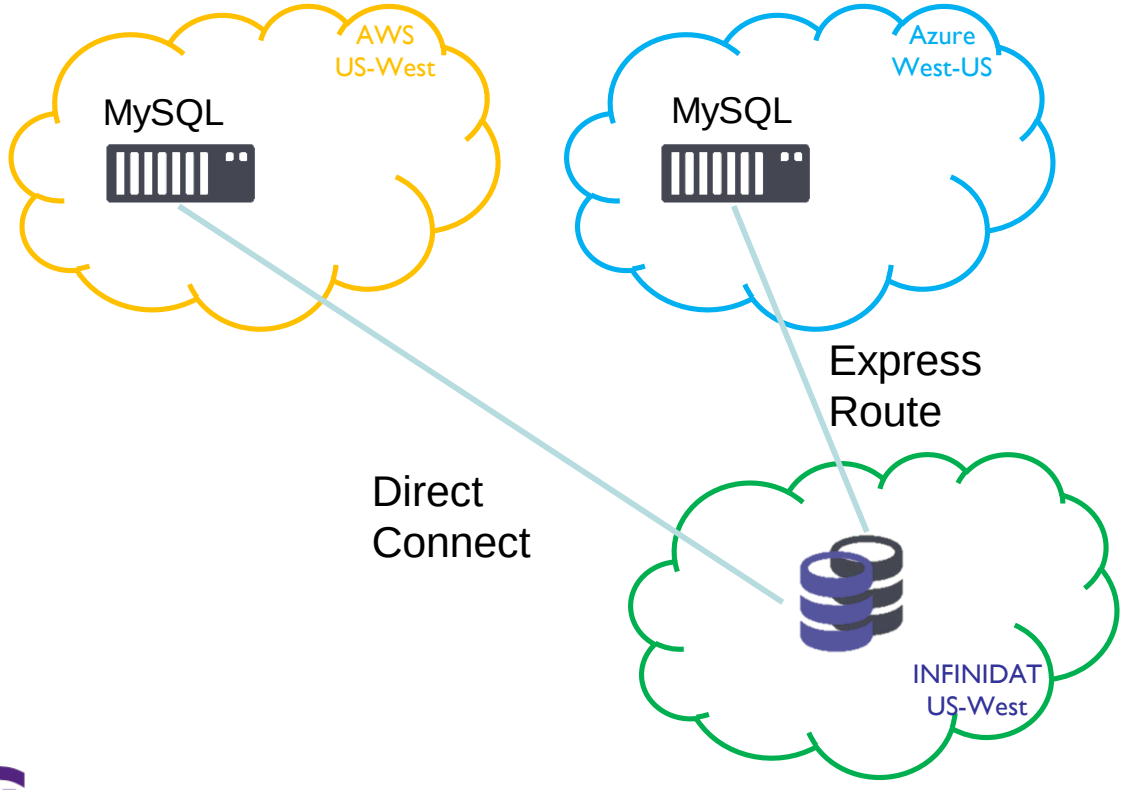
```
[gtouretsky@gtouret01:/mnt/nfsLeitrim]$ ./getCloudInstance.py
userName        hostName        privateIP       provider AZ              instanceType
gtouretsky      gtouret01       10.21.1.8       AZURE    westus          Standard_DS5_v2

[gtouretsky@gtouret01:/mnt/nfsLeitrim]$ while [ 1 ]; do echo AZURE `date` >> file; sleep 1;
done
```

```
gtouretsky@gtouret01:/mnt/nfsLeitrim$ ./getCloudInstance.py
userName        hostName        privateIP       provider        AZ              instanceType
gtouretsky      gtouret01       10.138.0.2      GCP             us-west1-a      n1-standard-1

gtouretsky@gtouret01:/mnt/nfsLeitrim$ tail -f file
AZURE Tue Apr 25 05:35:51 UTC 2017
AWS Tue Apr 25 05:35:37 UTC 2017
AZURE Tue Apr 25 05:35:52 UTC 2017
AWS Tue Apr 25 05:35:38 UTC 2017
AZURE Tue Apr 25 05:35:53 UTC 2017
```

# Demo: cross-cloud DB and snapshots



# Create an iSCSI volume attached to AWS

```
IMS> vol create --name vol001 --size_gb 50
Create vol vol001 succeeded.
IMS> host create --name mysql --ipaddress 10.11.0.146
Create host mysql succeeded.
IMS> host update --name mysql --op add --optype map --vol vol001
Update host mysql succeeded.
IMS> host update --name mysql --op add --optype port --address iqn.1993-08.org.debian:01:d92c1790b776
Update host mysql succeeded.

$ mkfs.ext4 -L mysql /dev/mapper/36742b0f000005880000000006b2944p1
$ mount -t ext4 /dev/mapper/36742b0f000005880000000006b2944p1 /var/lib/mysql
$ service mysql start
```





# Create and populate the DB in AWS

```
$ mysql -p -u root  --local-infile=1
mysql> CREATE TABLE pet (name VARCHAR(20), owner VARCHAR(20), species
VARCHAR(20), sex CHAR(1), birth DATE, death DATE);
Query OK, 0 rows affected (0.10 sec)
mysql> LOAD DATA LOCAL INFILE '/tmp/pets5.txt' INTO TABLE pet;
Query OK, 9999999 rows affected
...
mysql> select count(*) from pet;
| count(*) |
| 99000005 |

$ df –h
/dev/mapper/36742b0f000000588000000000006b2944p1   50G   6.0G   41G   13%
/var/lib/mysql
```

# Take a snapshot, make it writable and attach to Azure instance

```
IMS> snap create --name vol001_snap --vol vol001
Create snap vol001_snap succeeded.
IMS> snap update --name vol001_snap --op write_protect --oparg no
Update snap vol001_snap succeeded.
IMS> host create --name mysqlazure --ipaddress 10.21.1.14
Create host mysqlazure succeeded.
IMS> host update --name mysqlazure --op add --optype map --vol
vol001_snap
IMS> host update --name mysqlazure --op add --optype port --address
iqn.1993-08.org.debian:01:d21e95109bbf
```





# Launch DB in Azure – see the same data


```
$ mount -t ext4 /dev/mapper/36742b0f000000588000000000006b3844-part1
/var/lib/mysql
$ service mysql start
$ mysql -p -u root  --local-infile=1

mysql> select count(*) from pet;
| count(*) |
| 99000005 |


mysql> LOAD DATA LOCAL INFILE '/tmp/pets8.txt' INTO TABLE pet;
Query OK, 29999999 rows affected
```





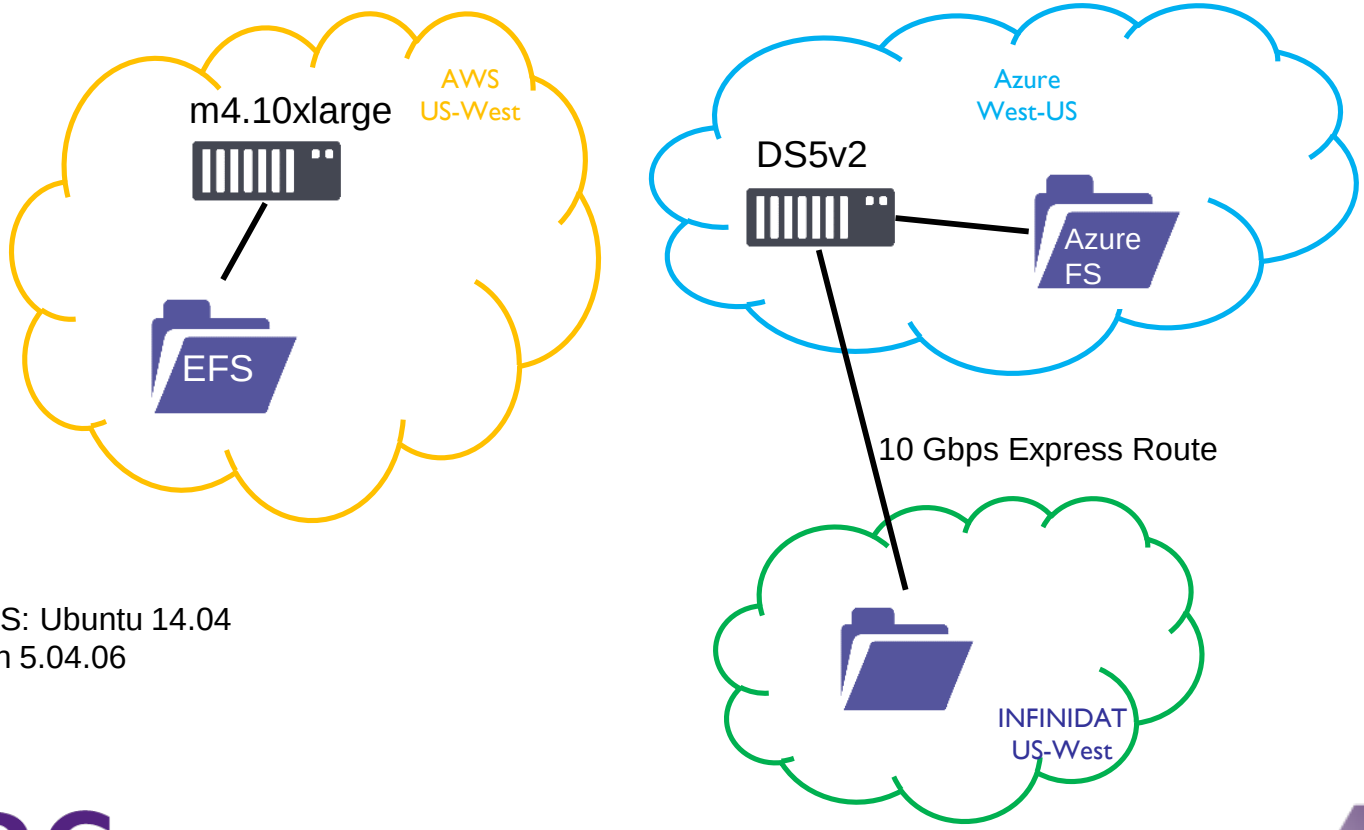
# Snapshots: Amazon EBS vs INFINIDAT cloud iSCSI

| | AWS EBS | INFINIDAT cloud snapshots |
|---|---|---|
| Time to create a snapshot | Several hours for large initial snapshots or subsequent snapshots where many blocks have changed | Instant |
| Performance impact | multiple pending snapshots of a volume may result in reduced volume performance until the snapshots complete. Limit of 5 pending snapshots for a single io1 or gp2 volume | No impact |
| Storage consumption | First snapshot – size equal to original volume. Subsequent snaps – store deltas | All snapshots store only deltas |
| Writable snapshots | No | Yes |



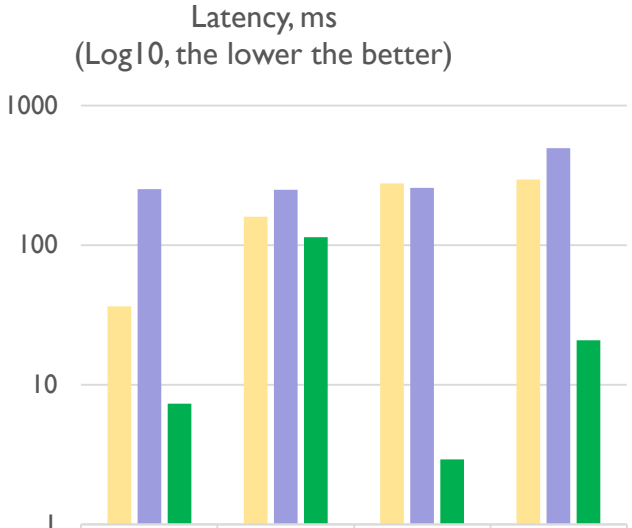


# Cloud file storage performance

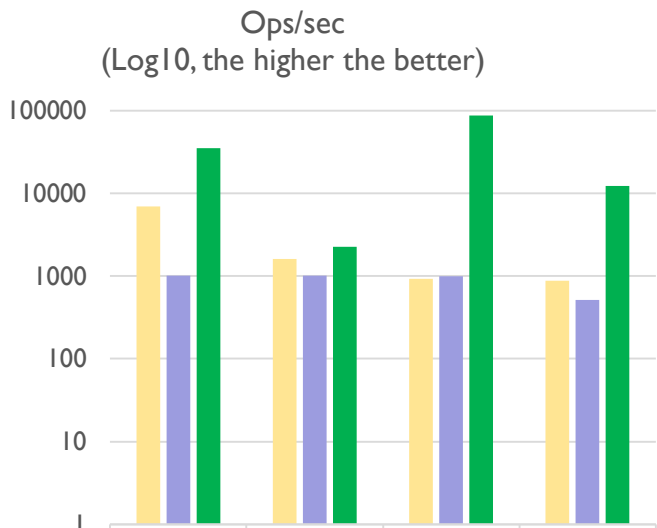



- Client OS: Ubuntu 14.04
- Vdbench 5.04.06

# Cloud file storage performance - results




| | RndRead_4k | RndRead_64k | SeqWrite_4k | SeqWrite_64k |
|---|---|---|---|---|
| AWS | 36.5 | 159 | 275 | 293 |
| Azure | 250.3 | 249 | 257 | 495 |
| INFINIDAT | 7.3 | 113.4 | 2.9 | 20.7 |




| | RndRead_4k | RndRead_64k | SeqWrite_4k | SeqWrite_64k |
|---|---|---|---|---|
| AWS | 7012 | 1602 | 929 | 873 |
| Azure | 1021 | 1024 | 996 | 517 |
| INFINIDAT | 35095 | 2254 | 87151 | 12347 |





# Provisioning a block storage

| | EBS volume | Leitrim iSCSI |
|---|---|---|
| Create a volume | aws ec2 create-volume --size 10 --availability-zone us-west-1c --volume-type gp2 | vol create --name volgt01 --size_gb 10 |
| Attach volume to a host | aws ec2 attach-volume –volume-id vol-0ca238dbfcd5bf355 --instance-id i-01593be600b15921a --device /dev/sdh | host update--name ubuntu001 --op add --optype map --vol volgt01 |
| Create a file system | mkfs.xfs –L ebs –f /dev/sdh | mkfs.xfs –L iscsi –f /dev/mapper/mpatha |







23

# File: Amazon EFS vs Azure File vs INFINIDAT Multicloud NFS

| | Amazon EFS (gp) | Azure File storage | INFINIDAT cloud NFS |
|---|---|---|---|
| File systems, max | 10 per region | | 1000s |
| Max NFS ops/sec per file system | 7,000 | 1,000 | Limited by network bandwidth. > 87,000 with 10Gbps link |
| Max bandwidth per file system | Depends on usage. Initial 50MB/sec with burst to 100MB/sec, Additional 100MB/sec for every TiB used | 60MB/sec | Limited by network bandwidth. >770MB/sec with 10Gbps link |
| Max size of a file | 52TiB | 1TB | 1PB+ |
| Capacity control | No (file system size is unlimited) | Max file system size = 5TB. Max per acct = 500TB | Can define max file system size. Max file system size >1PB |
| Multi-VPC | No | *No* | Yes |
| Multi-cloud | No | *No* | Yes |
| Snapshots | No | No | Yes – same as INFINIDAT block |





# Sample use cases

- Cloud onramp
- Hybrid cloud
  - Dev/Test
  - Burst capacity
- Data protection
- Multicloud storage





# Summary

- Cloud storage > object storage
- INFINIDAT multicloud storage:
  - Flexibility: on-prem, multi-cloud and multi-protocol
  - Economics: commoditize cloud compute, at scale
  - Features: enterprise-grade snapshots, NAS, etc
  - Performance





# Questions?

**Demo:** https://tinyurl.com/y859aked

Interested to participate in our pilot program?

cloudstorage@infinidat.com





# Backup





# InfiniBox Architecture

- **Triple-Redundant Power & Data Path**
- **3 Active-Active-Active Nodes**
- **Infiniband Interconnectivity**
- **24 x 8 Gb FC Ports**
- **6/12 x 10 GbE Ports**
- **480 Disk Drives**
- **Up to 3 TB DRAM**
- **Up to 210 TB SSD**


*"… Deployment and management of our InfiniBox systems has been, for lack of a better term, stupidly simple."*

**-- Financial Institution, IT Director**

# Provisioning a file system

| | EFS file system | Leitrim NAS |
|---|---|---|
| Create file system | aws efs create-file-system --creation-token $(uuidgen) --performance-mode generalPurpose --region us-east-1 | filesystem create --name fs01 --size_gb 100 |
| Export file system | aws efs create-mount-target --file-system-id filesystemid --subnet-id subnetid --security-groups groupid --region us-east-1 | export create --exportpath /fs01 --filesystem_name fs01;<br>update export --exportpath /fs01 --op add --optype permission --client 10.11.0.0-10.11.0.254 --access RW --no_root_squash |
| Mount file system | mount *x.x.x.x*:/ /mnt/efs | mount *x.x.x.x*:/ /mnt/fs01 |







30