CMP301

Deep dive on Amazon EBS

Ashish Palekar
Director of Product Management
Amazon Web Services

aws SUMMIT

Complete set of data building blocks

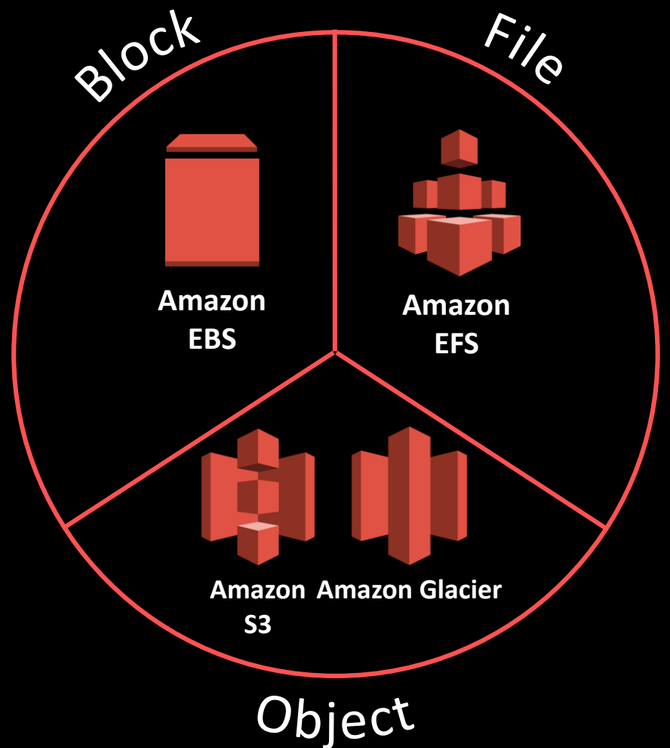
Data movement

Online

- AWS Storage Gateway family
- Amazon Kinesis Video Streams
- Amazon Kinesis Data Streams
- Amazon EFS file sync
- Amazon S3 transfer acceleration
- AWS Direct Connect
- Storage partners

Offline

- AWS Snow family

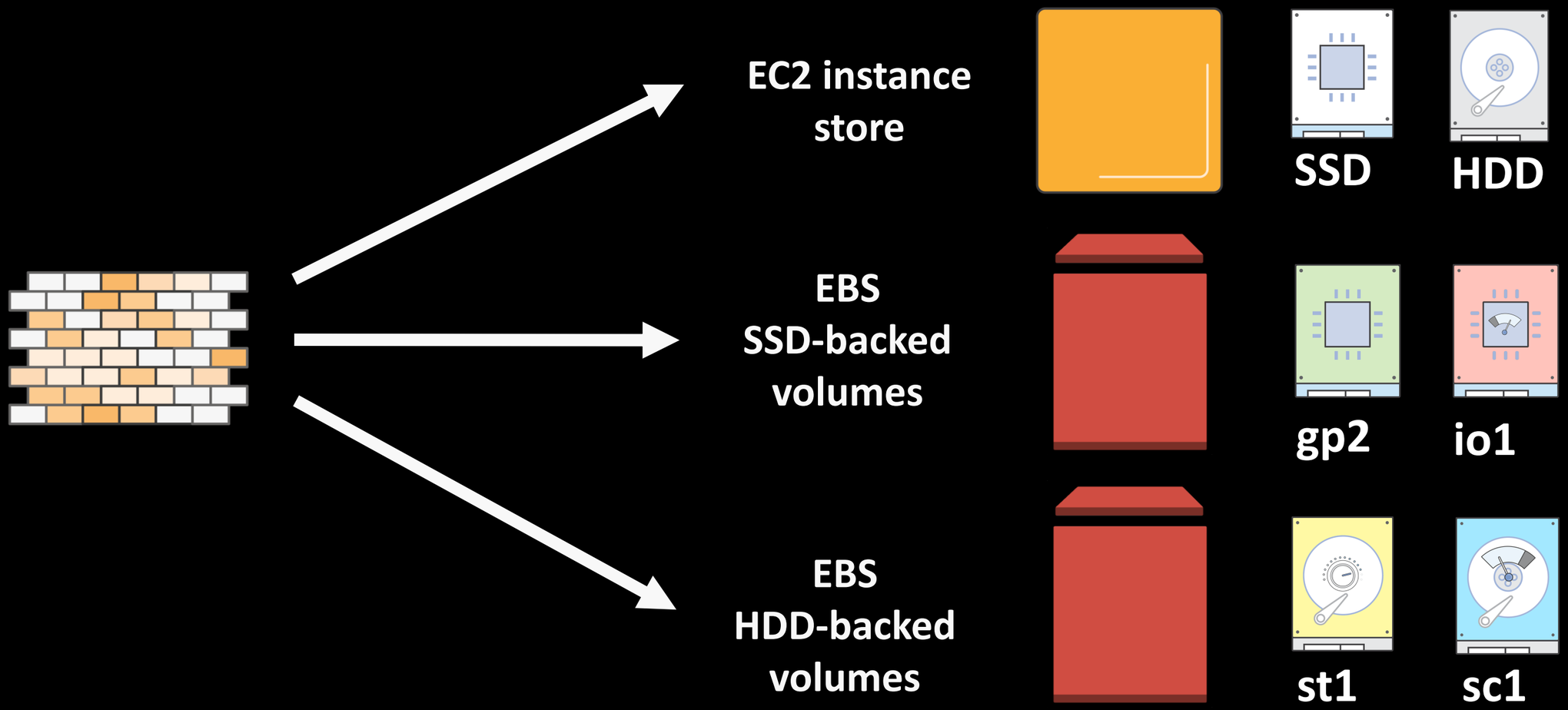


Data security and management

- Amazon Macie
- Amazon QuickSight
- AWS Lambda
- AWS CloudFormation
- AWS CloudTrail
- Amazon CloudWatch
- AWS IAM
- AWS KMS

aws SUMMIT

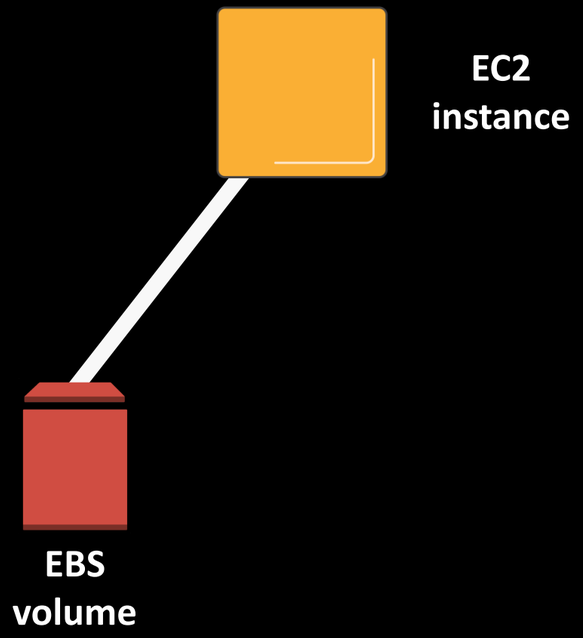
AWS block storage offerings



What is Amazon EC2 instance store?

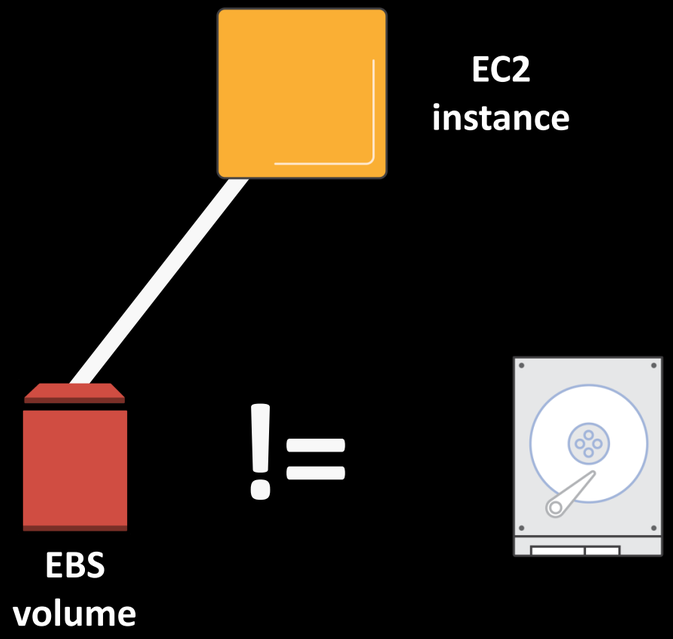
- Local to instance
- Non-persistent data store
- Data not replicated (by default)
- No snapshot support
- Solid state drive (SSD) or hard disk drive (HDD)

What is Amazon EBS?



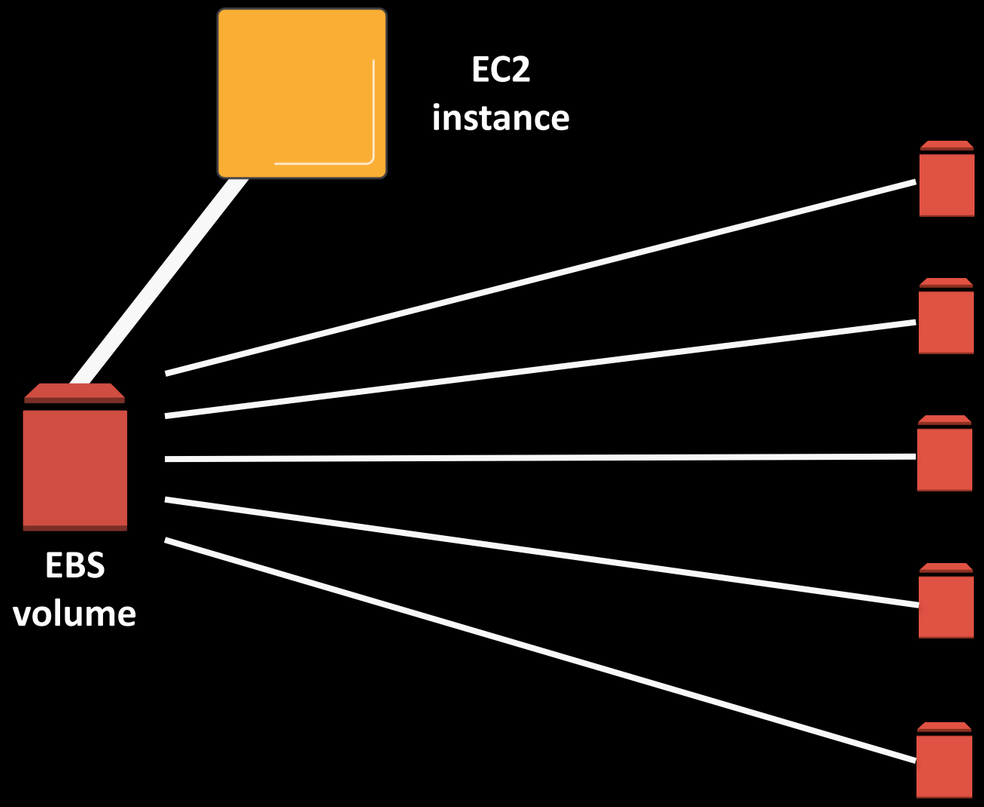
- Block storage as a service
- Create and attach volumes through an API
- Service accessed over the network

What is Amazon EBS?

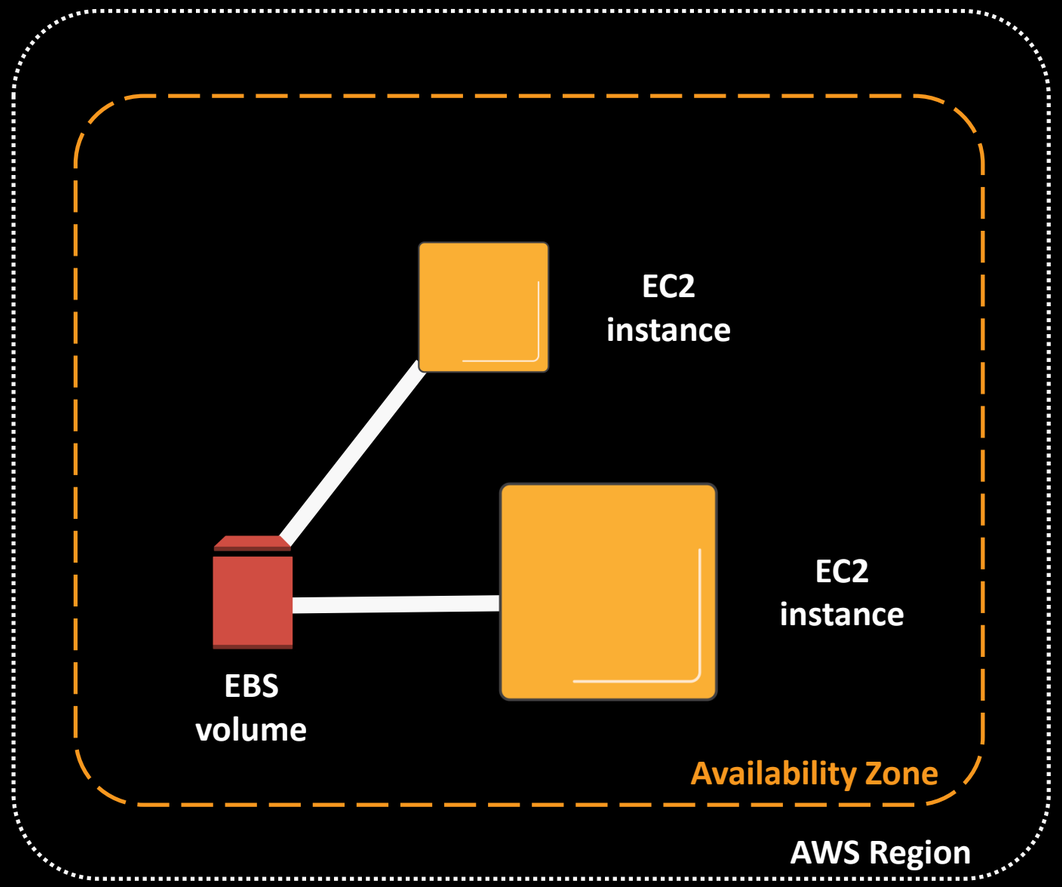


- Block storage as a service
- Create and attach volumes through an API
- Service accessed over the network

What is Amazon EBS?

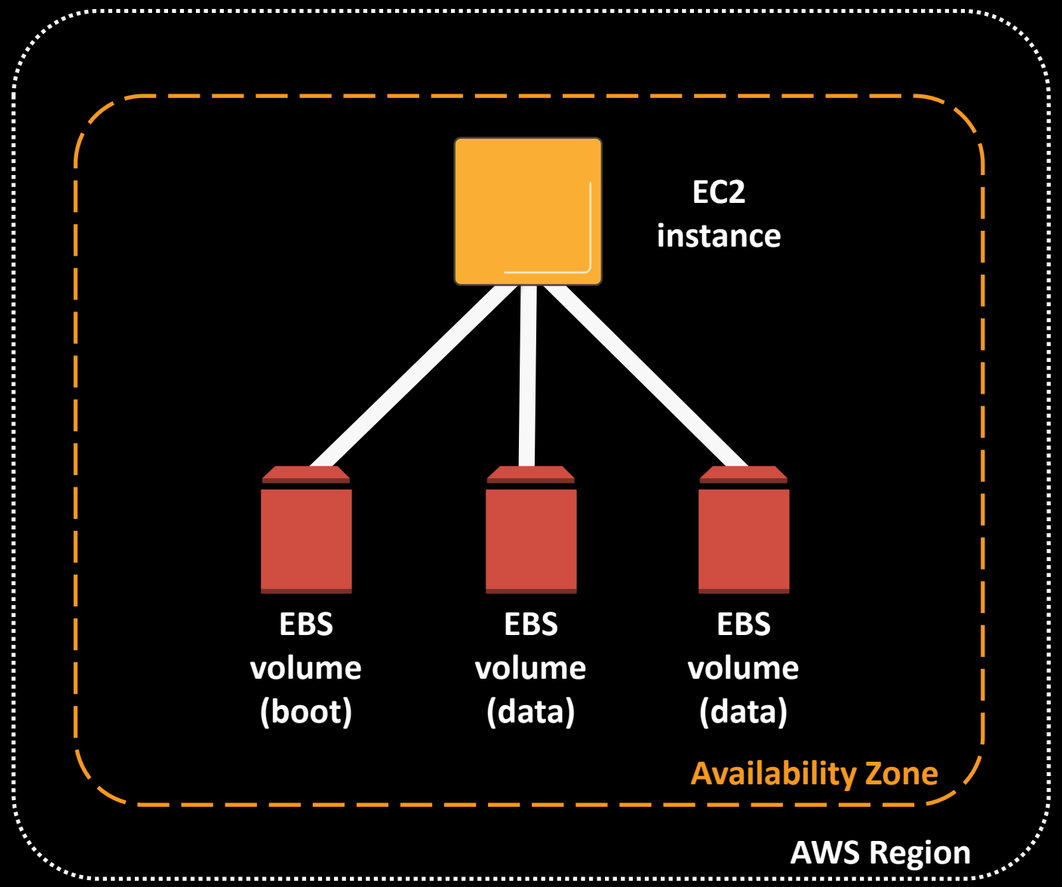


What is Amazon EBS?



- Volumes persist independent of EC2
- Select storage and compute based on your workload
- Detach and attach between instances within the same Availability Zone

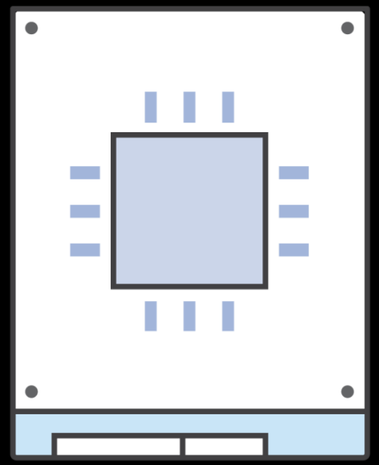
What is Amazon EBS?



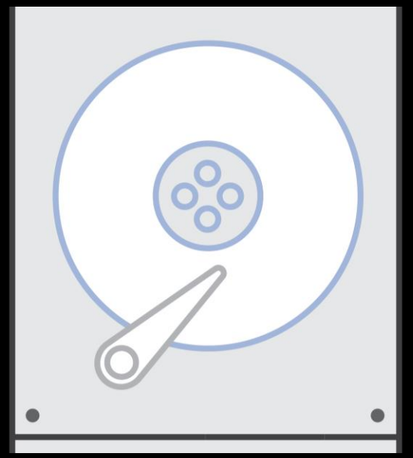
- Volumes attach to one instance
- Many volumes can attach to an instance
- Separate boot and data volumes

Amazon EBS volume types

Amazon EBS volume types

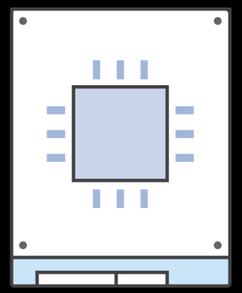


SSD

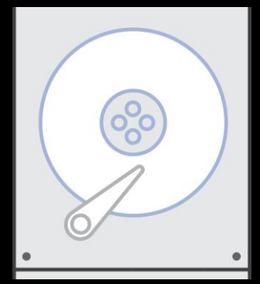


HDD

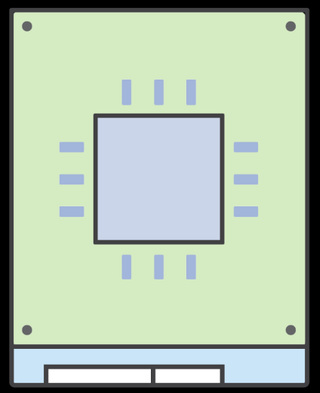
Amazon EBS volume types



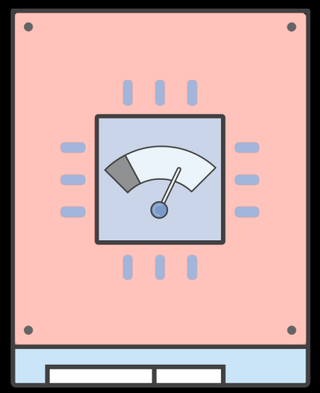
SSD



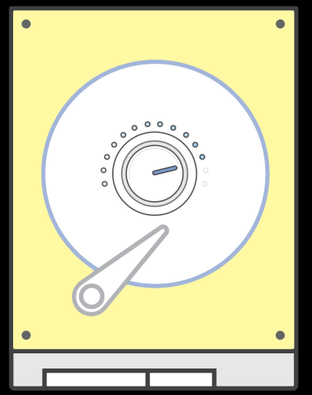
HDD



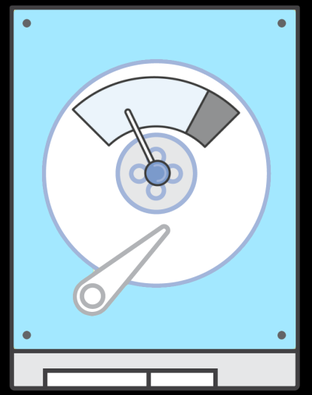
gp2
General Purpose
SSD



io1
Provisioned IOPS
SSD

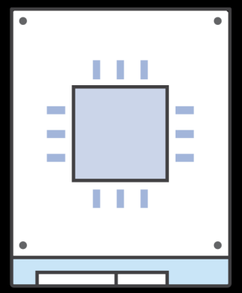


st1
Throughput
Optimized HDD

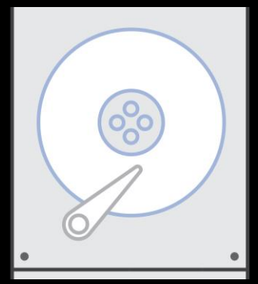


sc1
Cold HDD

Amazon EBS use cases



SSD



HDD

Relational databases

MySQL, SQL Server, PostgreSQL, SAP, Oracle

NoSQL databases

Cassandra, MongoDB, CouchDB

Big data, analytics

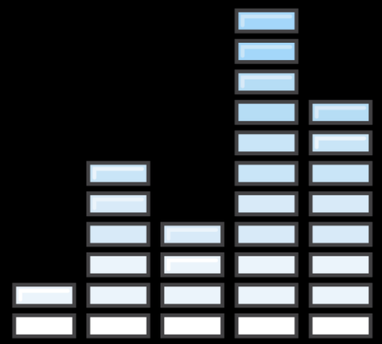
Kafka, Splunk, Hadoop, data warehousing

File/media

CIFS/NFS, transcoding, encoding, rendering

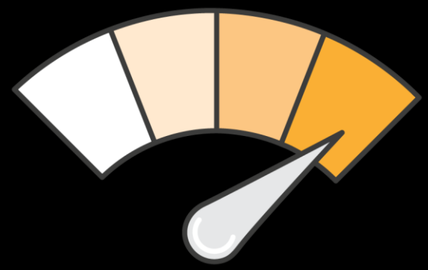
Choosing an Amazon EBS volume type

What is more important to your workload?



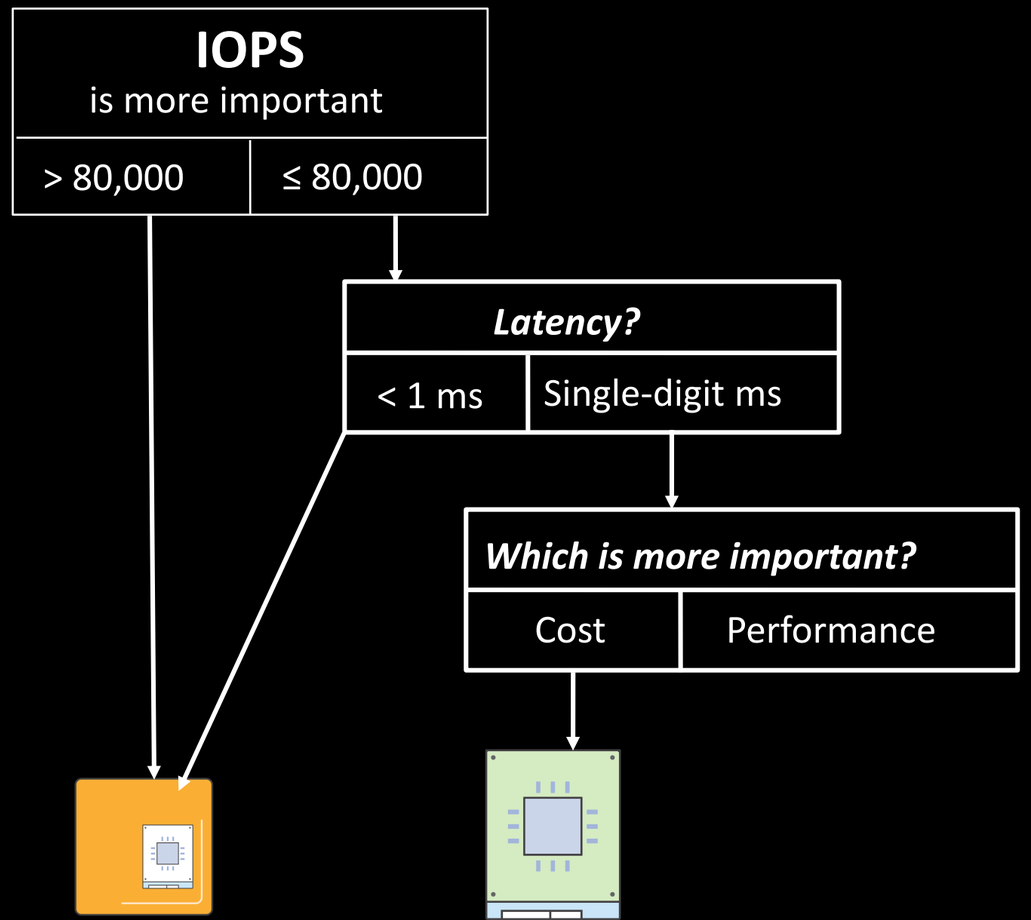
IOPS

or

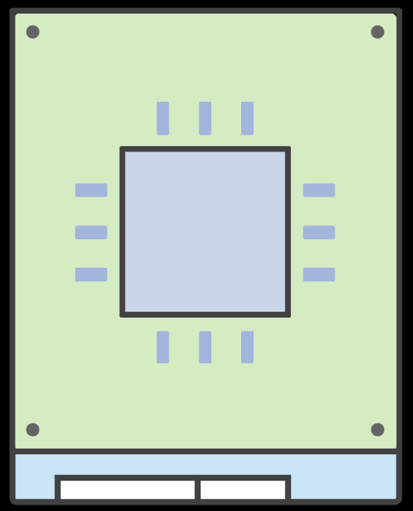


Throughput?

Choosing an Amazon EBS volume type



Amazon EBS volume types: General Purpose SSD



gp2

General Purpose SSD

Baseline: 100 to 16,000 IOPS; 3 IOPS per GiB

Burst: 3,000 IOPS (for volumes up to 1,000 GiB)

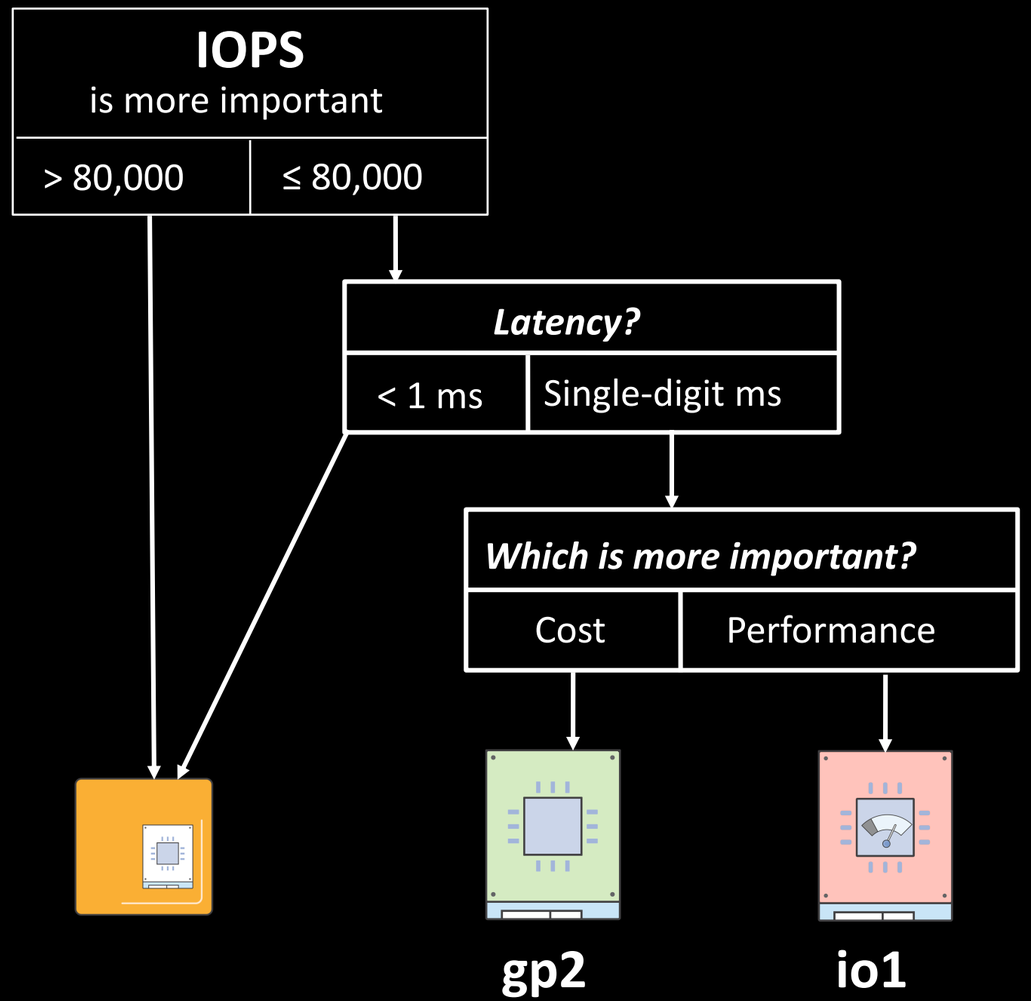
Throughput: Up to 250 MiB/s

Latency: Single-digit ms

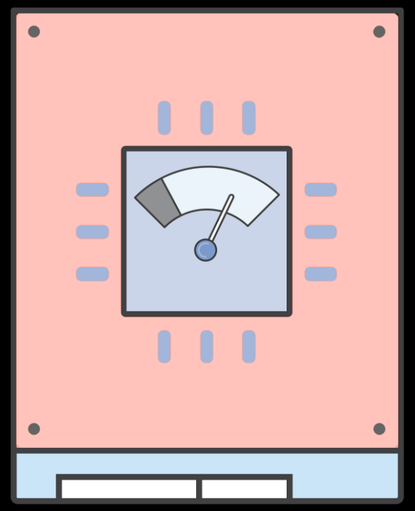
Capacity: 1 GiB to 16 TiB

Great for boot volumes, low-latency applications, and bursty databases

Choosing an Amazon EBS volume type



Amazon EBS volume types: Provisioned IOPS



io1

Provisioned IOPS

Baseline: 100 to 64,000 IOPS

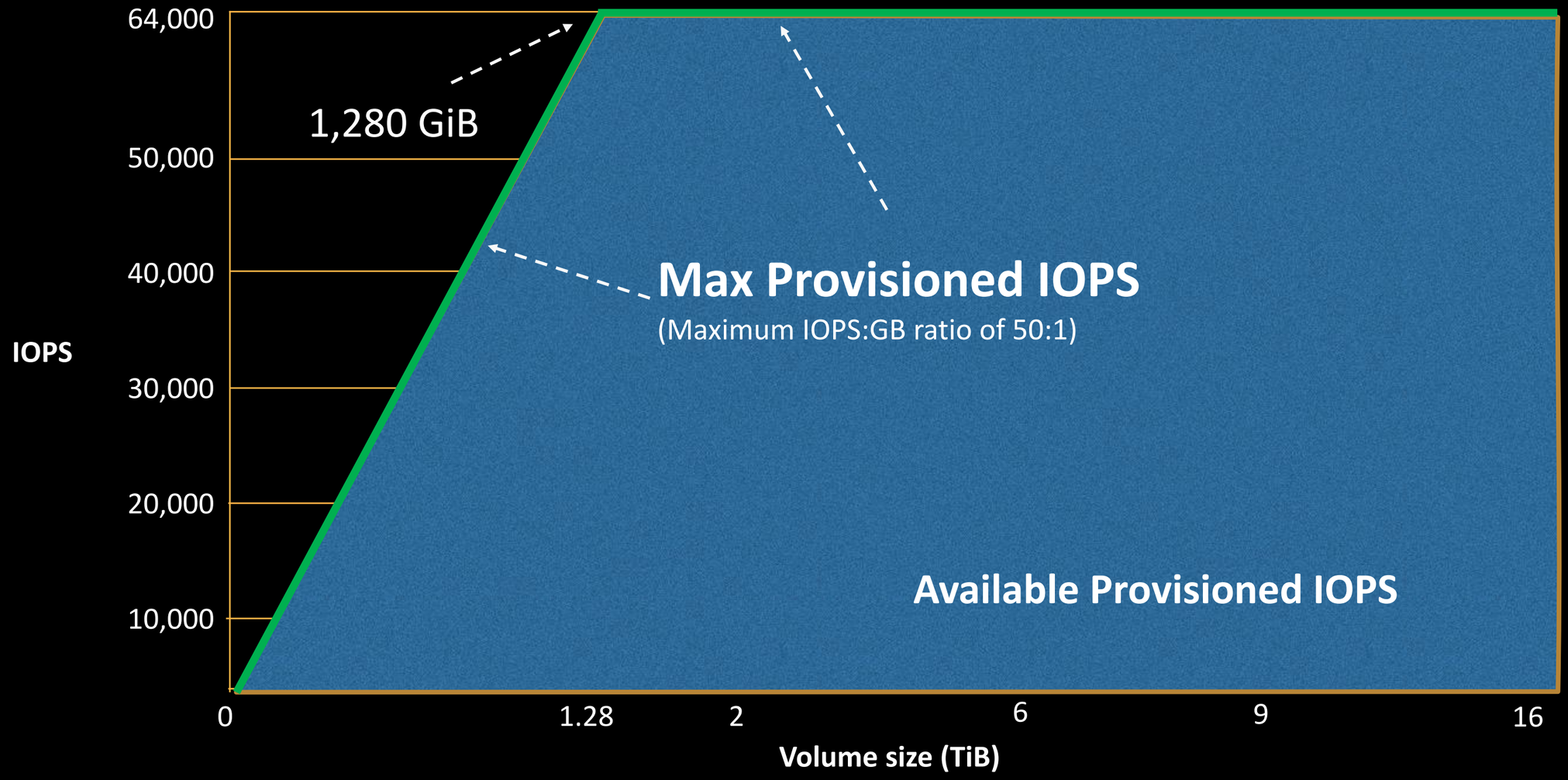
Throughput: Up to 1,000 MiB/s

Latency: Single-digit ms

Capacity: 4 GiB to 16 TiB

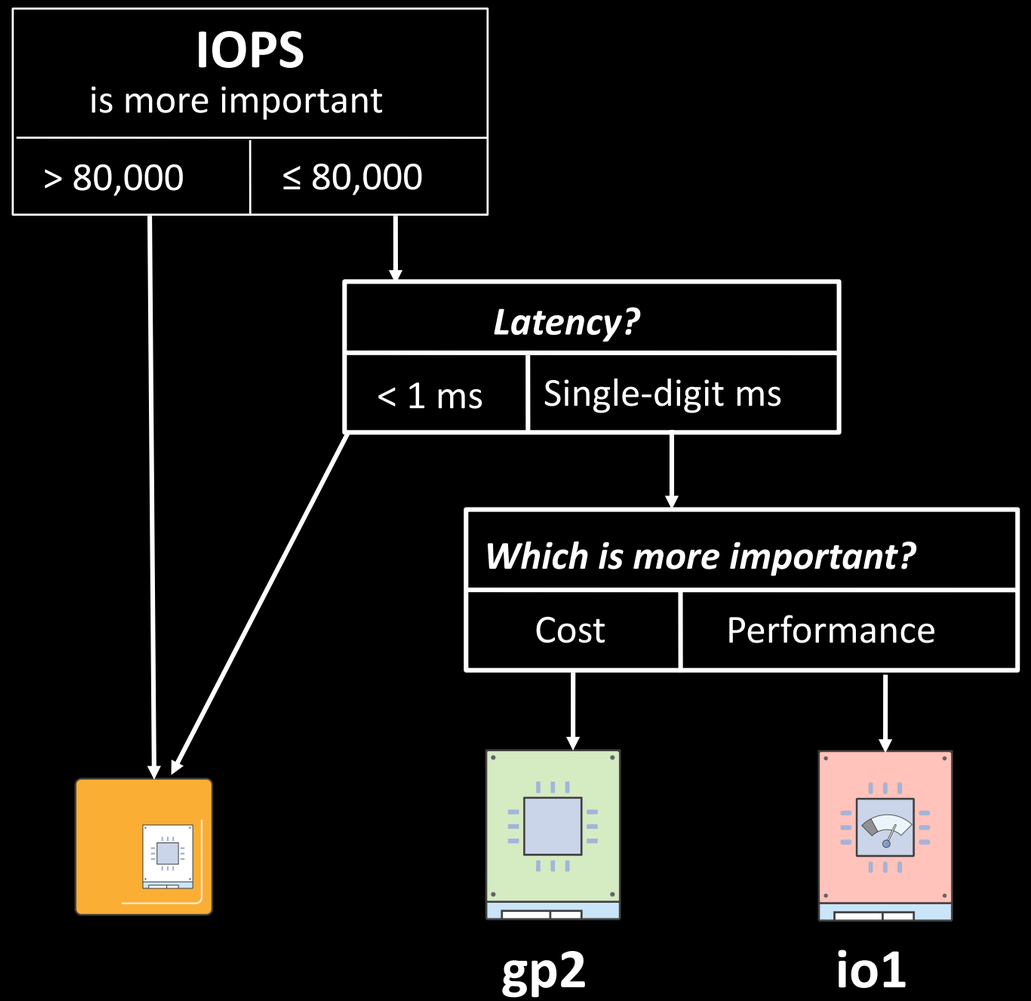
Ideal for critical applications and databases with sustained IOPS

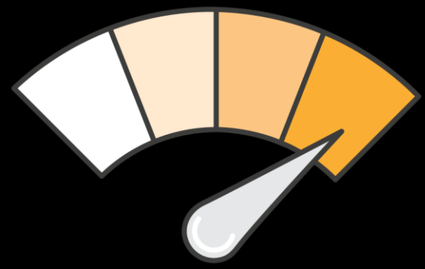
Scaling Provisioned IOPS SSD (io1)



aws SUMMIT

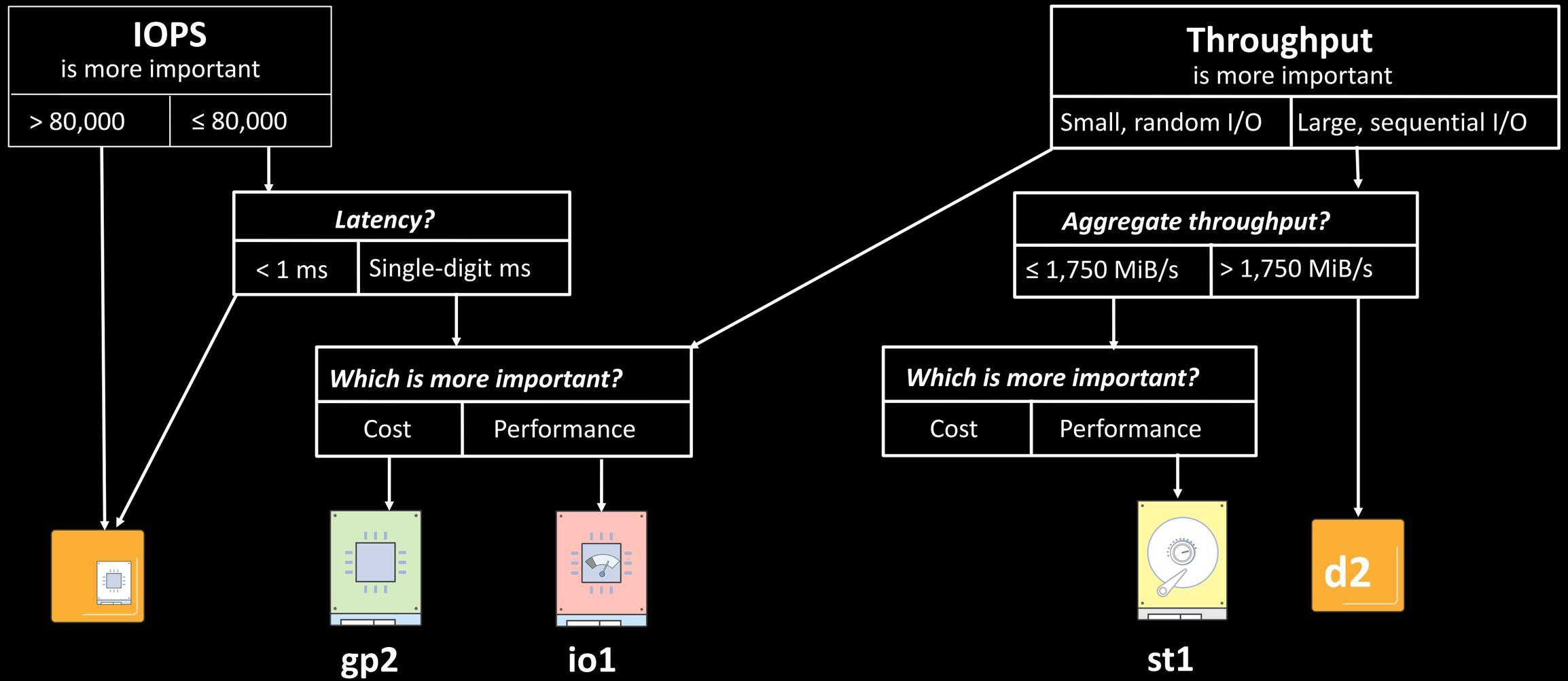
Choosing an Amazon EBS volume type



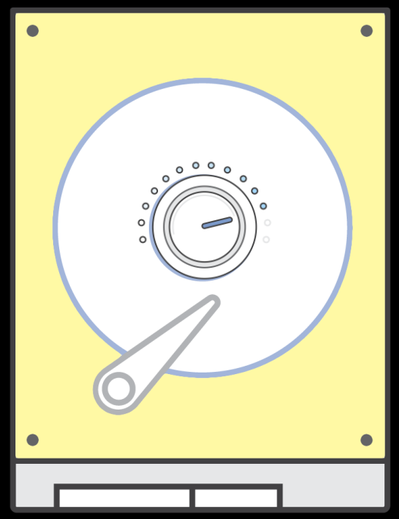


Throughput?

Choosing an Amazon EBS volume type



Amazon EBS volume types: Throughput provisioned



st1

Throughput Optimized HDD

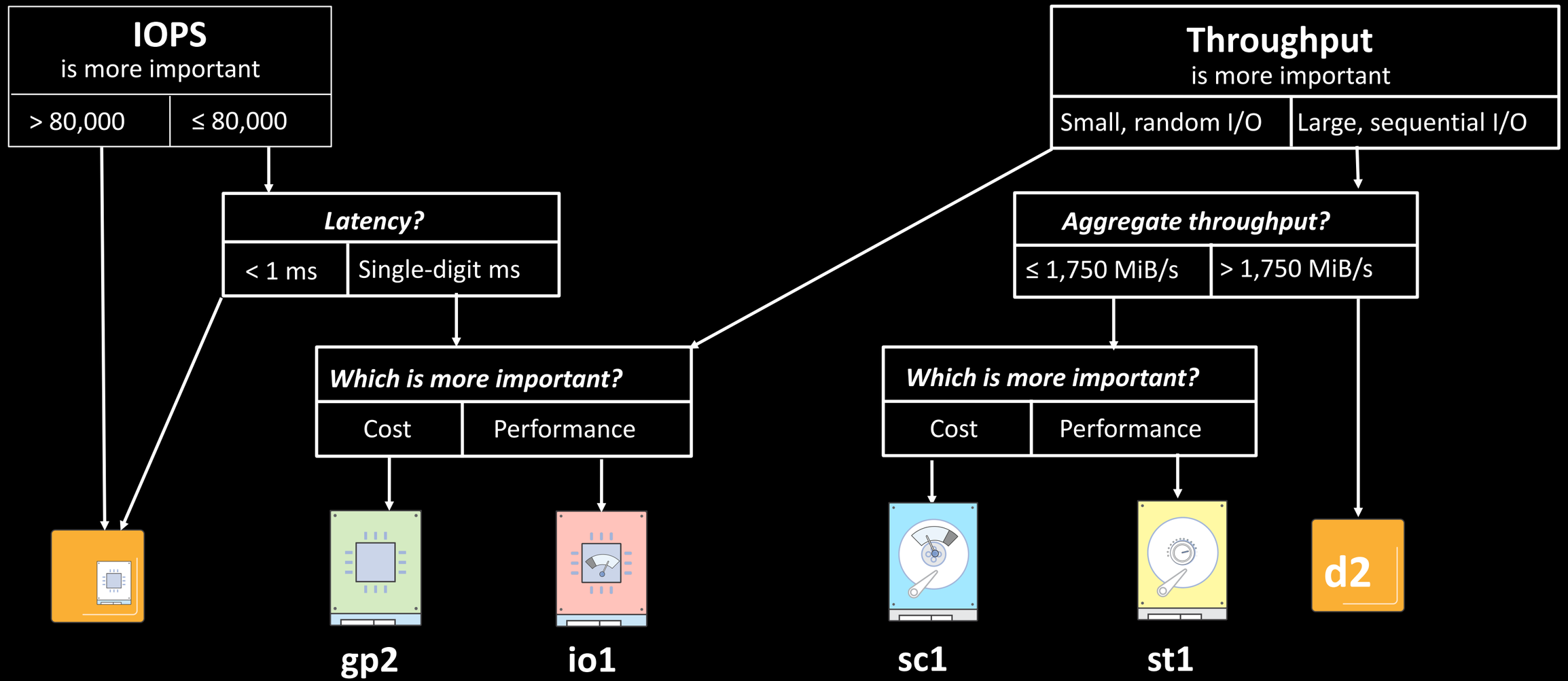
Baseline: 40 MiB/s per TiB up to 500 MiB/s

Burst: 250 MiB/s per TiB up to 500 MiB/s

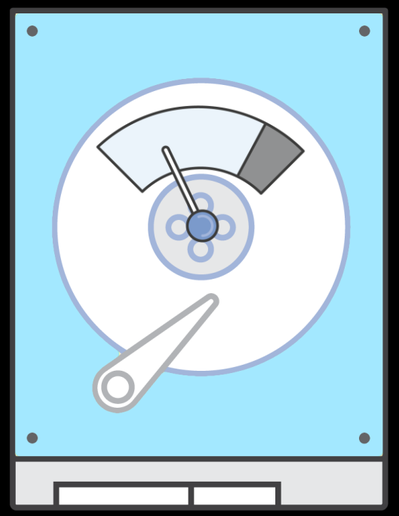
Capacity: 500 GiB to 16 TiB

Ideal for large-block, high-throughput sequential workloads

Choosing an Amazon EBS volume type



Amazon EBS volume types: Throughput provisioned



sc1

Cold HDD

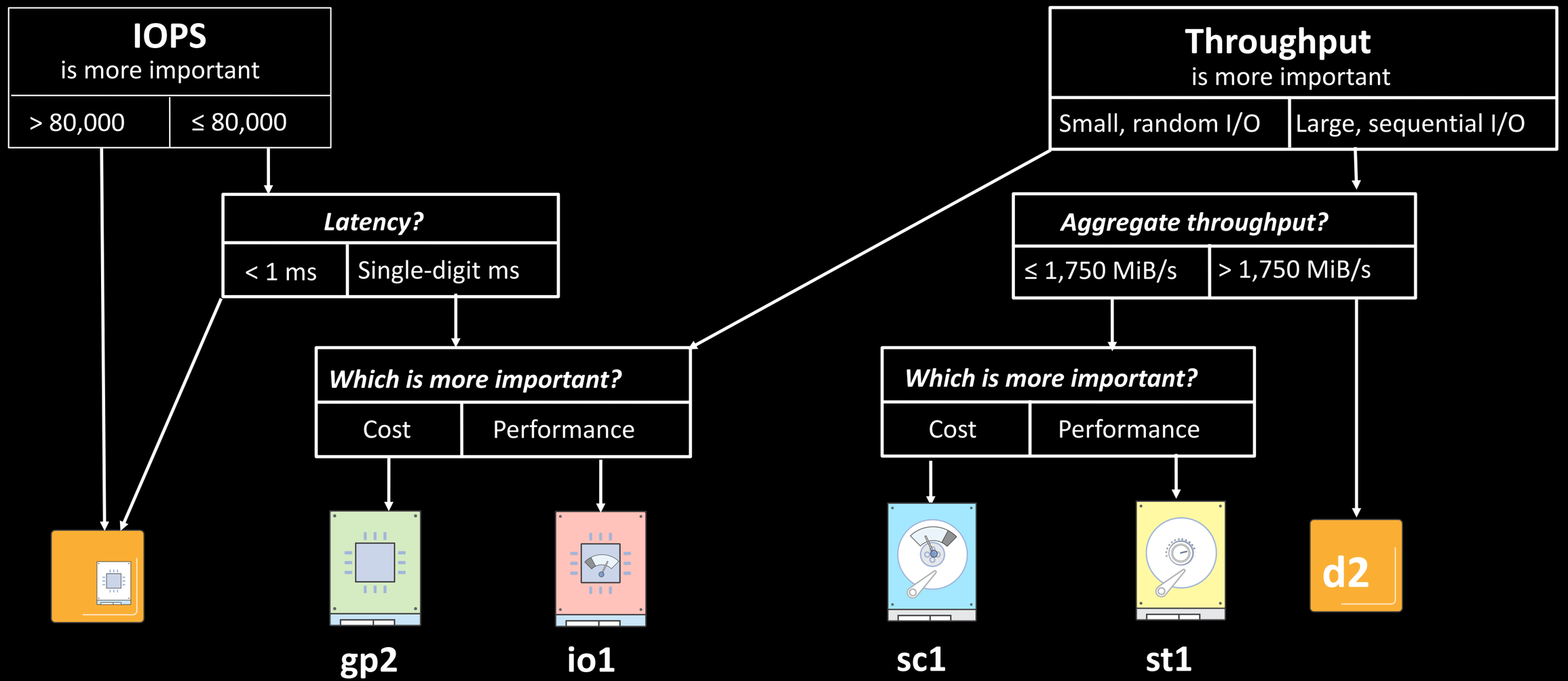
Baseline: 12 MiB/s per TB up to 192 MiB/s

Burst: 80 MiB/s per TB up to 250 MiB/s

Capacity: 500 GiB to 16 TiB

Ideal for sequential throughput workloads, such as logging and backup

Choosing an Amazon EBS volume type



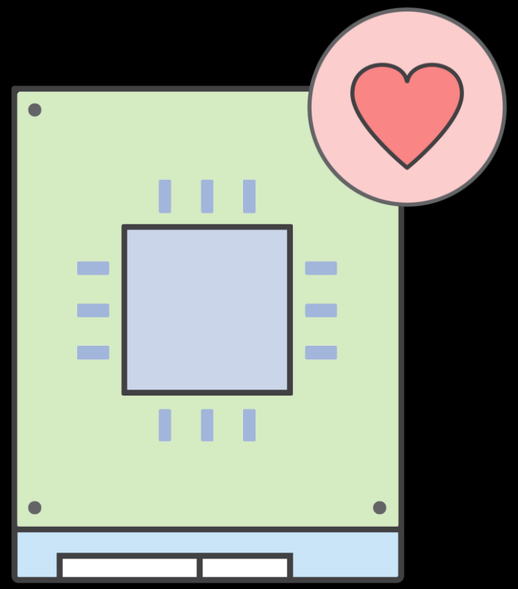
Choosing an Amazon EBS volume type

Don’t know

your workload

yet?

Amazon EBS volume types: General Purpose SSD



gp2

General Purpose SSD

Baseline: 100 to 16,000 IOPS; 3 IOPS per GiB

Burst: 3,000 IOPS (for volumes up to 1,000 GiB)

Throughput: Up to 250 MiB/s

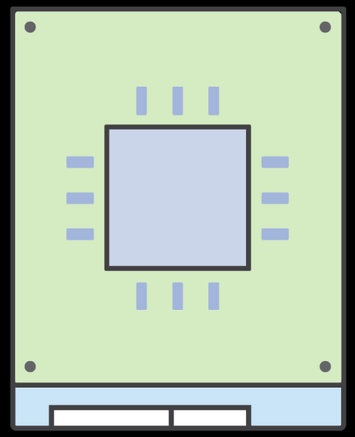
Latency: Single-digit ms

Capacity: 1 GiB to 16 TiB

Great for boot volumes, low-latency applications, and bursty databases

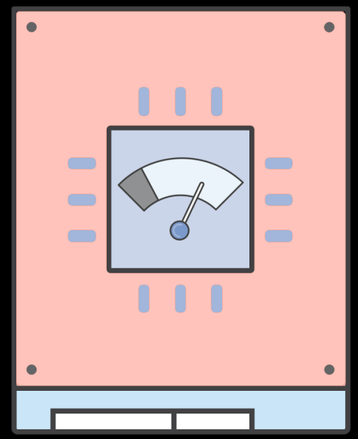
I/O provisioned volumes

Throughput provisioned volumes



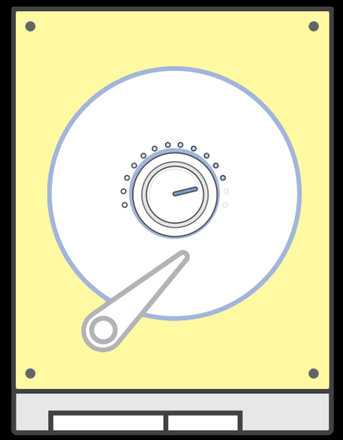
gp2

$0.10 per GiB



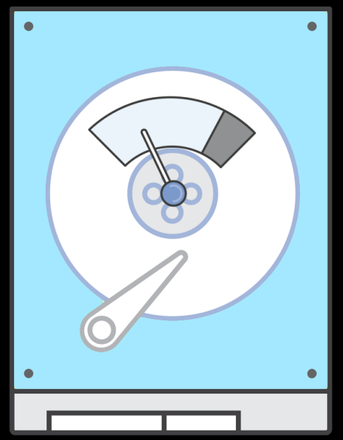
io1

$0.125 per GiB
$0.065 per PIOPS



st1

$0.045 per GiB



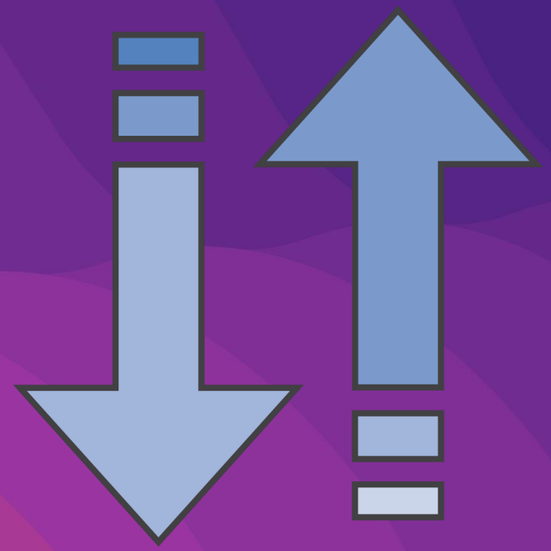
sc1

$0.025 per GiB

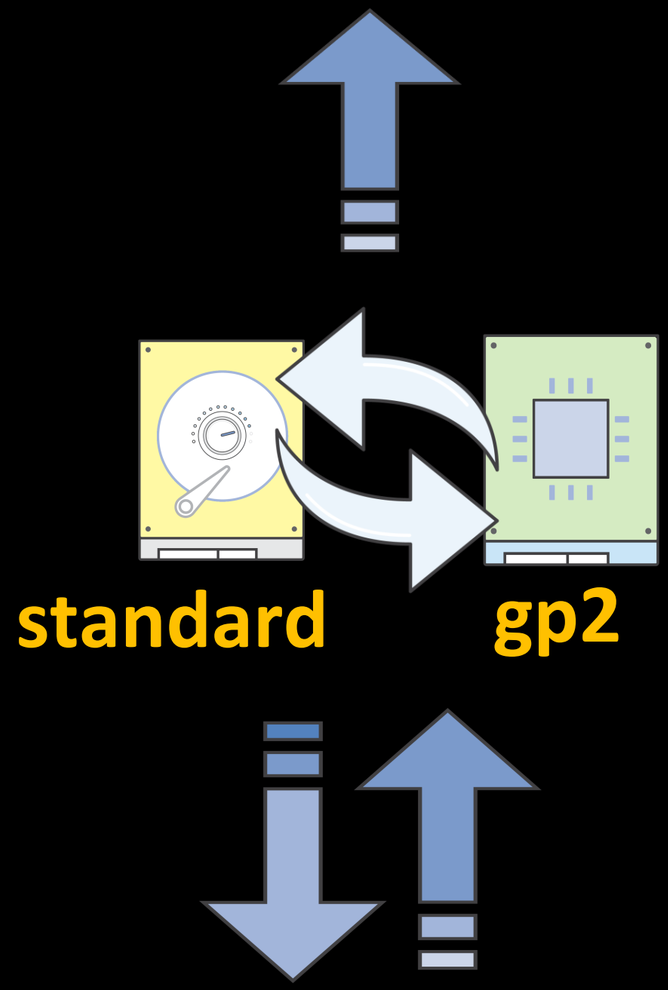
Snapshot storage for all volume types is $0.05 per GiB per month

All prices are per month, prorated to the second, and are from the us-west-2 Region as of April 2018

Amazon EBS Elastic Volumes



Elastic Volumes: Features

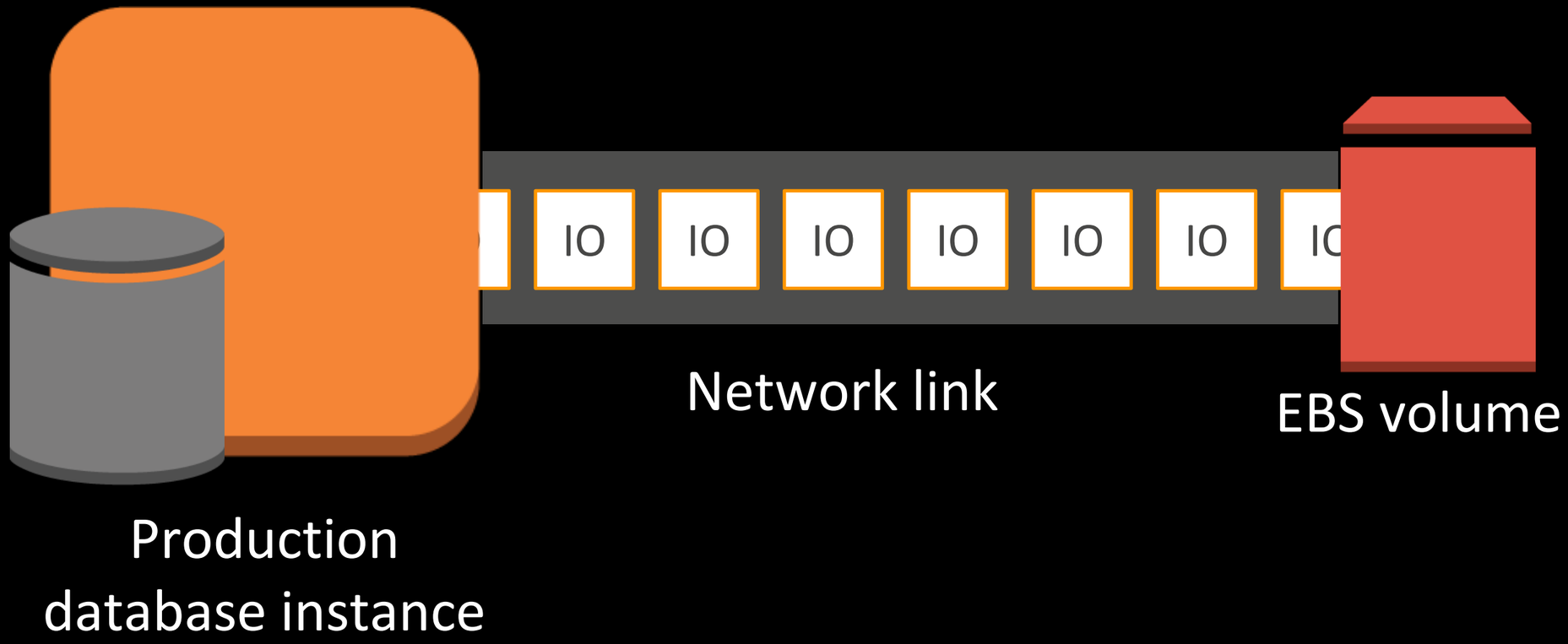


Increase volume size

Change volume type

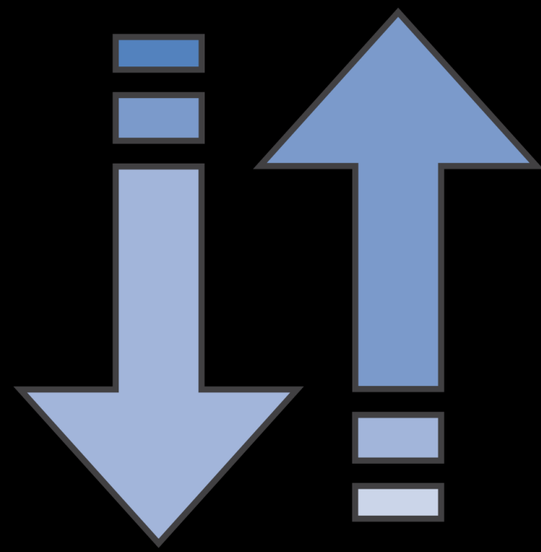
Increase/decrease provisioned IOPS

Elastic Volumes: Overview



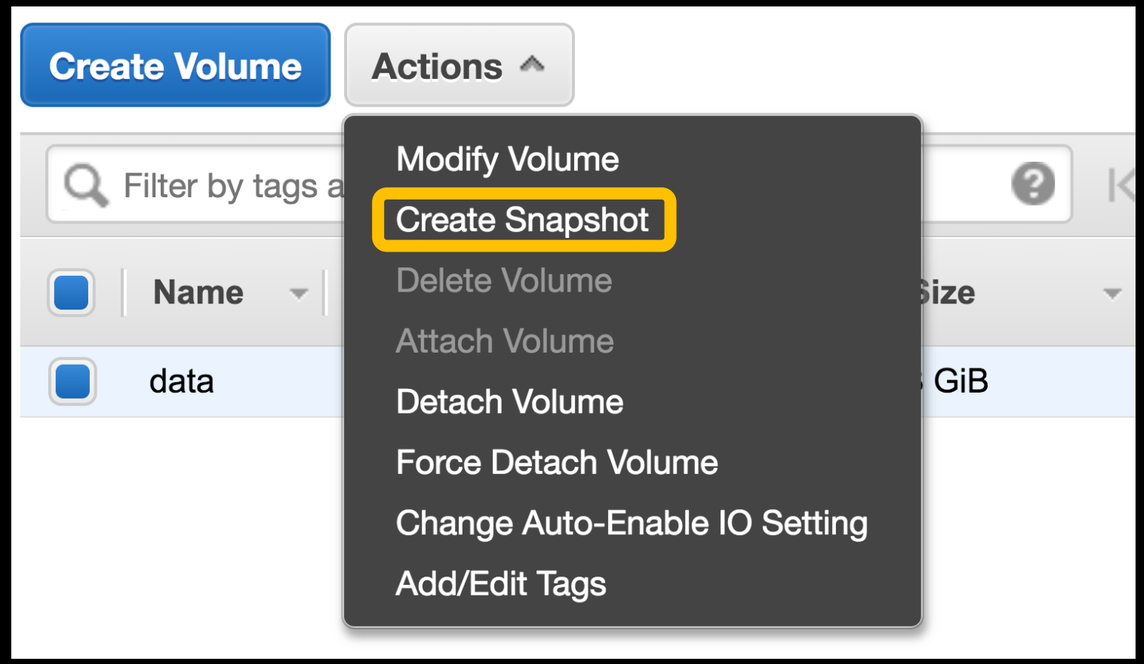
How to modify

1. Snapshot volume
2. Modify volume
3. Monitor modification
4. Extend file system (if necessary)



Step 1: Snapshot volume

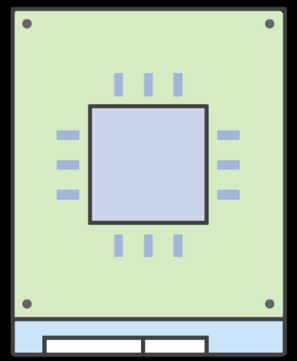
```
aws ec2 create-snapshot
  --volume-id vol-00077cd243d4af642
  --description ”data before resize”
```



Step 2: Modify volume

```
aws ec2 modify-volume
 --volume-id vol-00077cd243d4af642
 --size 5000
 --volume-type io1
 --iops 32000
```

Original volume

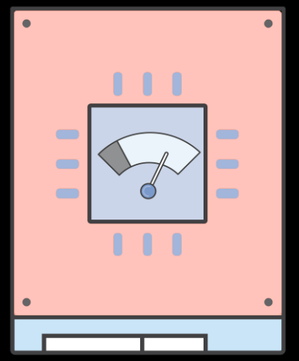


Type: gp2

IOPS: 3,000

Size: 1 TiB

Modified volume



Type: io1

IOPS: 32,000

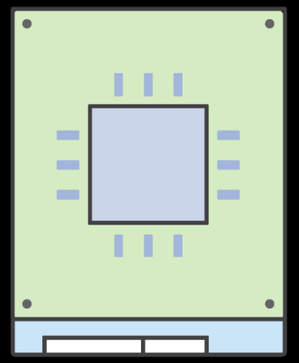
Size: 5 TiB

Step 3: Monitor volume

```
aws ec2 describe-volumes-modifications
 --volume-id vol-00077cd243d4af642
```


```
{
 "VolumesModifications": [
  {
   "TargetSize": 5000,
   "TargetVolumeType": "io1",
   "ModificationState": "optimizing",
   "VolumeId": "vol-00077cd243d4af642",
   "TargetIops": 32000,
   "StartTime": "2018-03-30T17:09:18.486Z",
   "Progress": 99,
   "OriginalVolumeType": "gp2",
   "OriginalIops": 3000,
   "OriginalSize": 1000
  }
 ]
}
```


Original volume

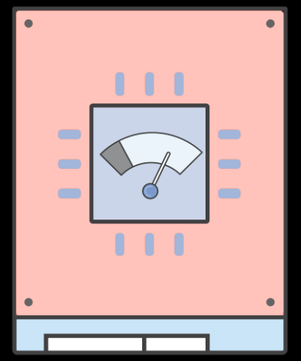


Type: gp2

IOPS: 3,000

Size: 1 TiB

Modified volume

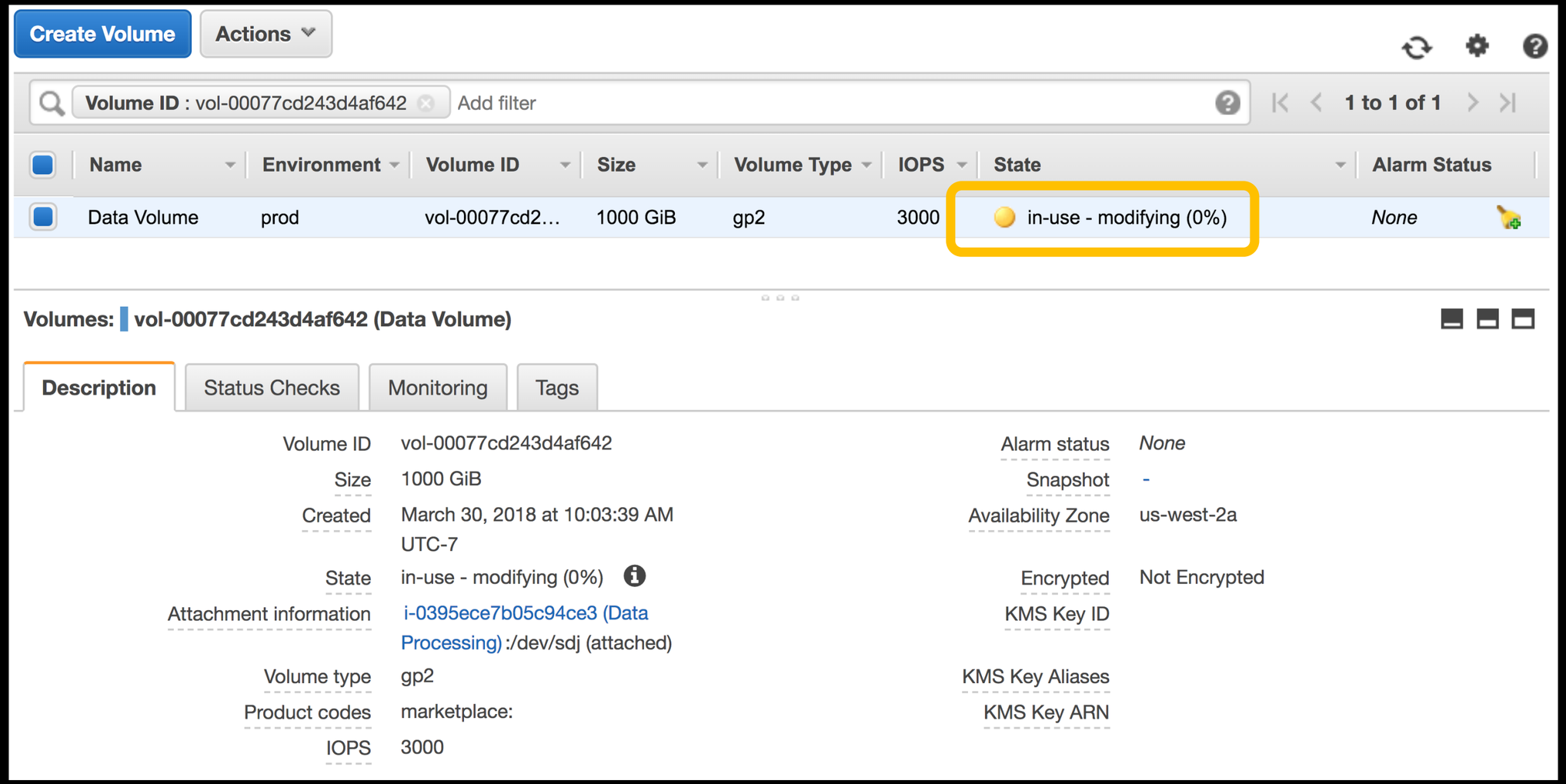


Type: io1

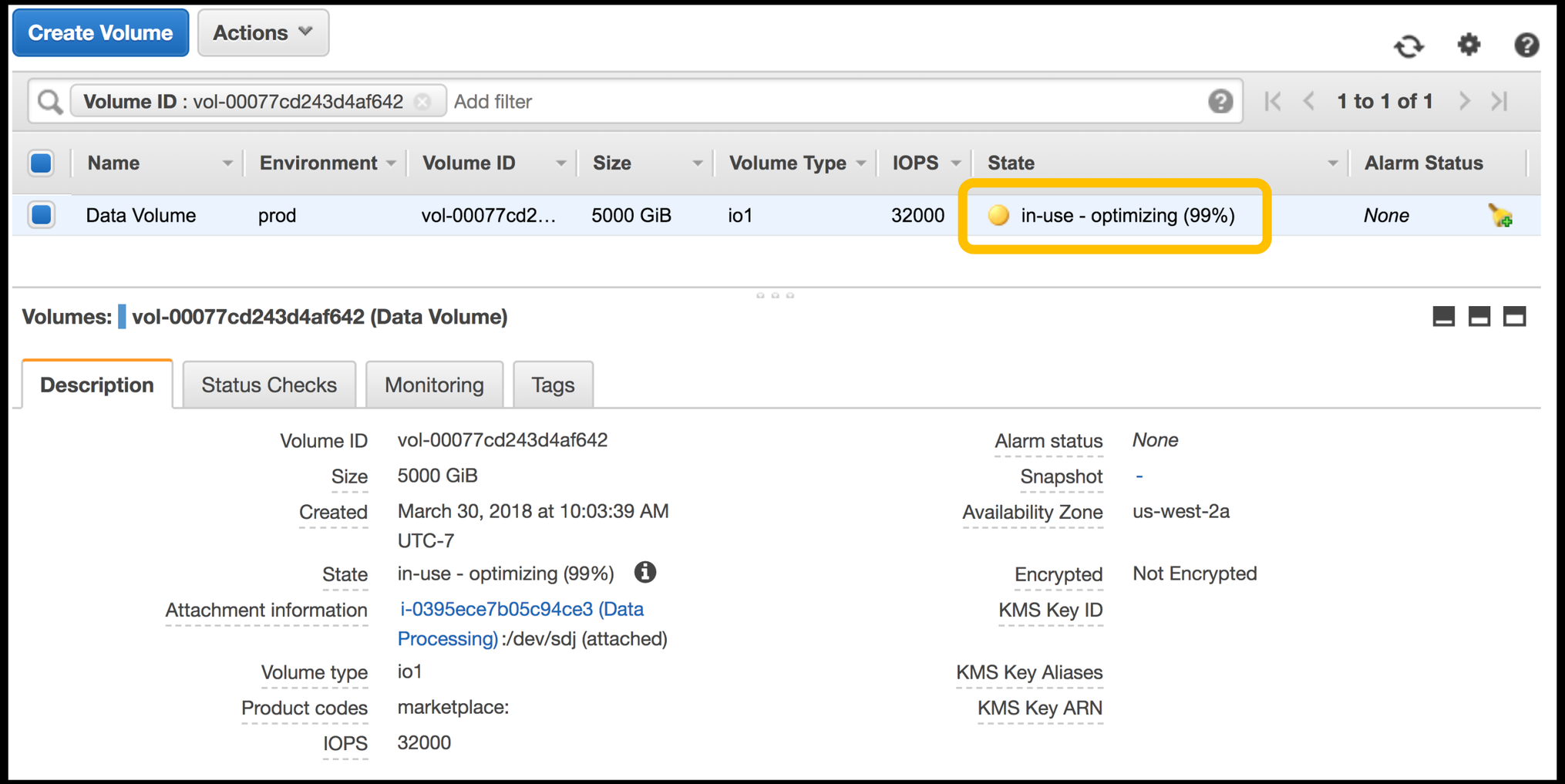
IOPS: 32,000

Size: 5 TiB

Step 3: Monitor volume (EC2 console)



Step 3: Monitor volume (EC2 console)



Step 4: Extending the file system (Linux)

- Determine your file system

```
[ec2-user ~]$ sudo file –s /dev/xvd*
/dev/xvda1: Linux rev 1.0 ext4 filesystem data ...
/dev/xvdf: SGI XFS filesystem data ...
```

- Compare block device size to file system disk usage

```
[ec2-user ~]$ lsblk
[ec2-user ~]$ df –h
```

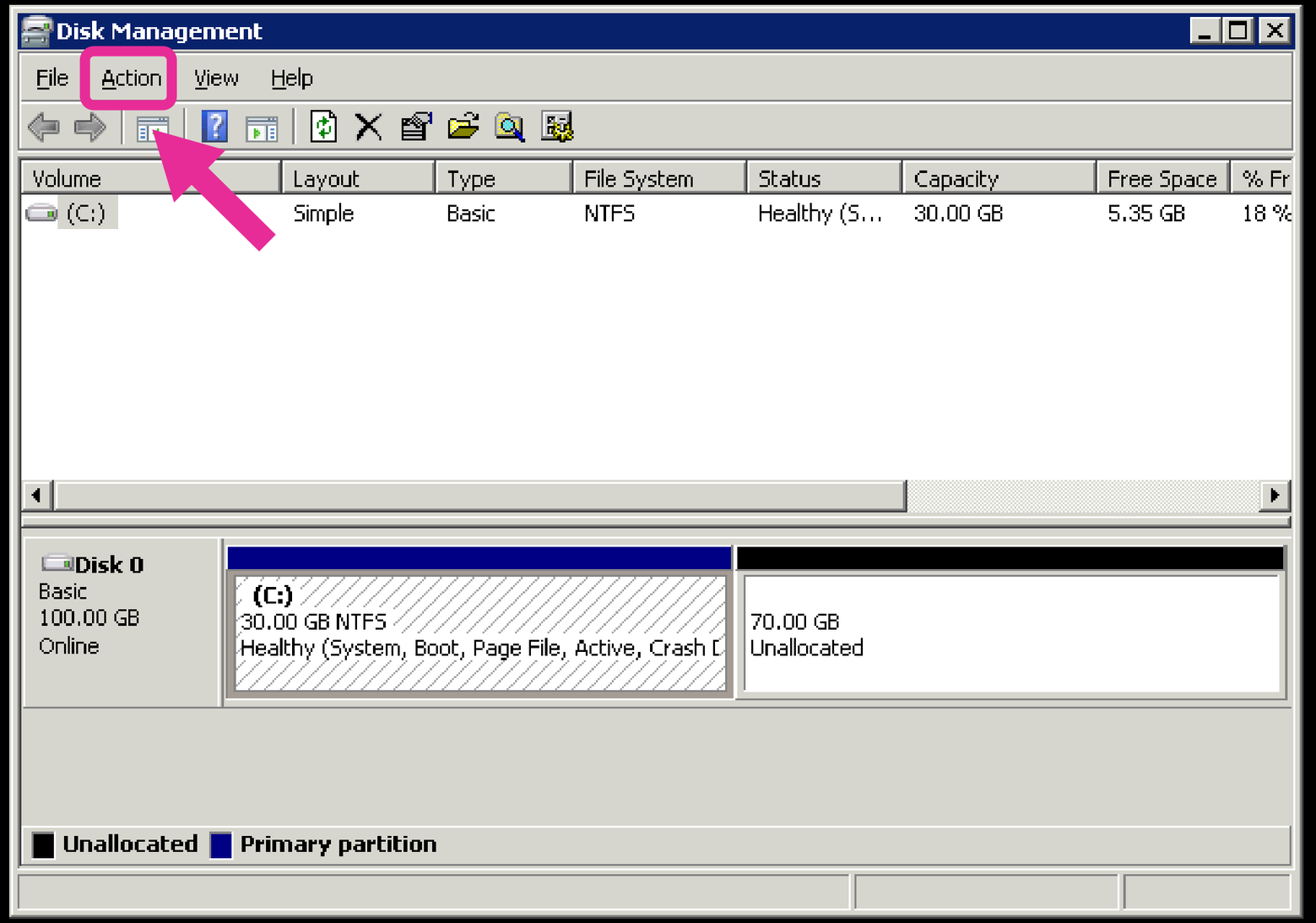
- Extend the file system (if needed)

  Note: If the drive is partitioned, first grow the partitions

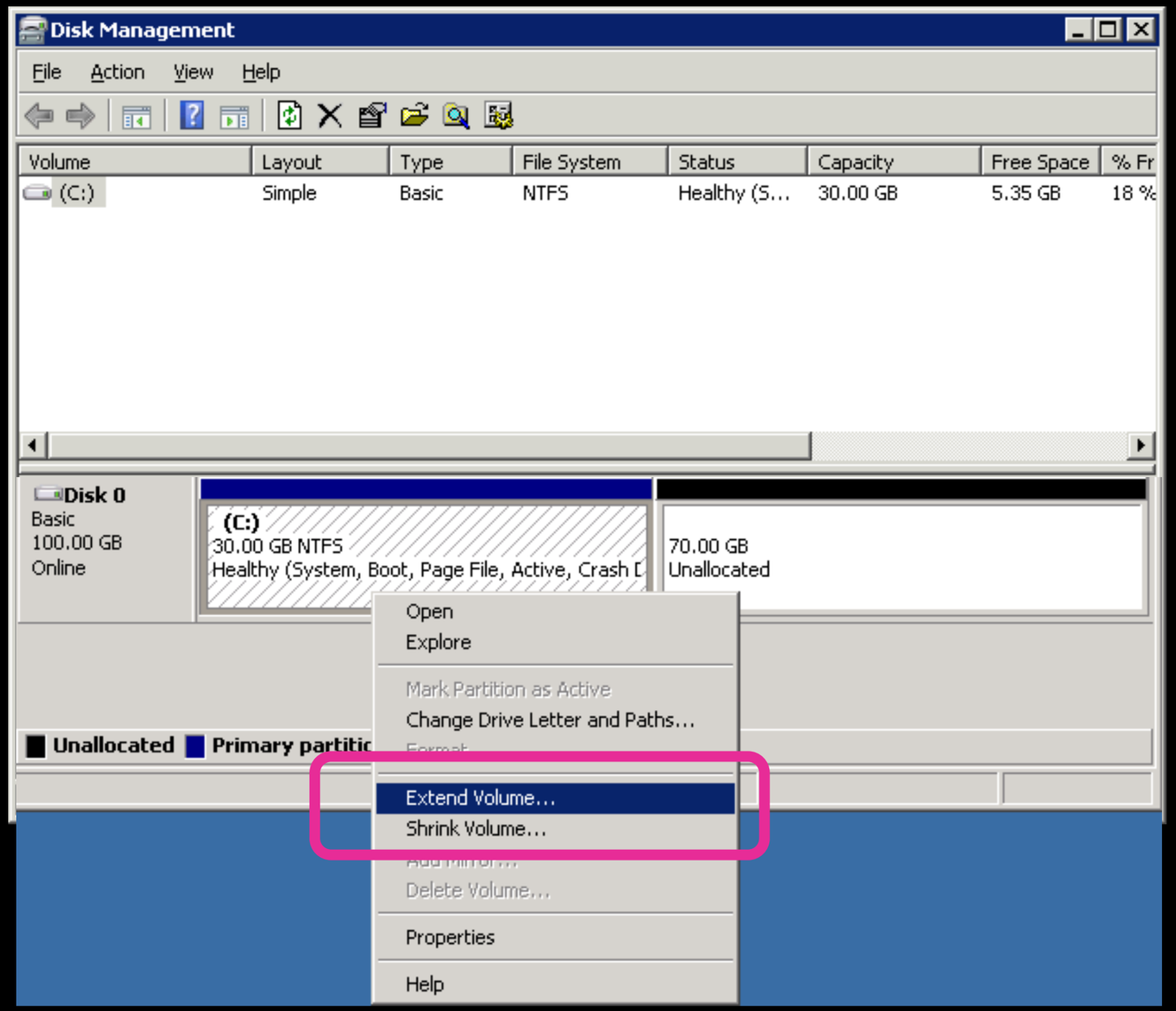
```
EXT:
[ec2-user ~]$ sudo resize2fs device_name

XFS:
[ec2-user ~]$ sudo xfs_growfs –d mount_point
```

https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/recognize-expanded-volume-linux.html

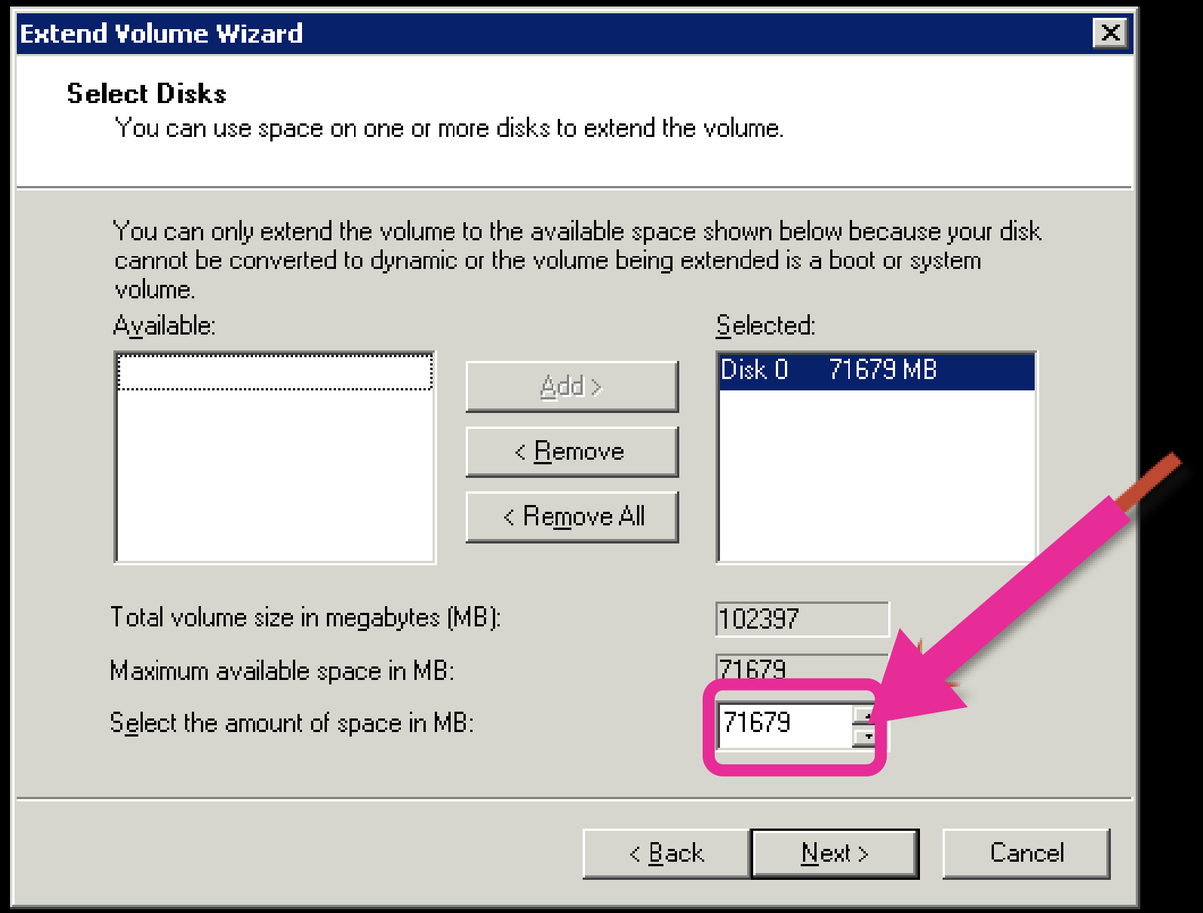
Step 4: Extending the file system (Windows)



Step 4: Extending the file system (Windows)



Step 4: Extending the file system (Windows)



Resize partition with PowerShell:

https://docs.microsoft.com/en-us/powershell/module/storage/resize-partition?view=win10-ps

Volume modification tips

- Modification must fit within volume specs (1 GiB gp2 != 1 GiB st1)
- Can modify volumes once every 6 hours
- Current generation instances do not need a stop/start or attach/detach
- Volumes created before 11/1/2016 require a stop/start or attach/detach

Amazon EBS deep dive

aws SUMMIT

How do volumes attach?

Amazon EBS volume attachment

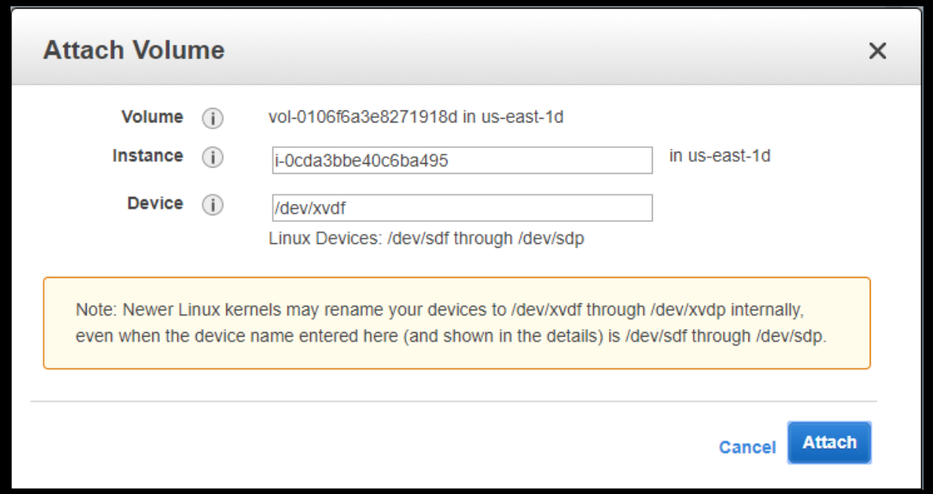
API

AttachVolume

CLI

```
aws ec2 attach-volume --instance-id i-0cda3bbe40c6ba495
      --volume-id vol-01324f611e2463981 --device /dev/xvdf
```

Console



Amazon EBS volume attachment: Xen

```
[ec2-user@ip-10-0-22-96 ~]$ lsblk
NAME      MAJ:MIN  RM   SIZE  RO  TYPE MOUNTPOINT
xvda      202:0     0     8G   0  disk
└─xvda1   202:1     0     8G   0  part /
xvdf      202:80    0     1G   0  disk
```

aws SUMMIT

Amazon EBS volume attachment: Nitro Hypervisor

```
[ec2-user@ip-10-0-12-183 ~]$ lsblk
NAME           MAJ:MIN RM SIZE RO TYPE MOUNTPOINT
nvme1n1        259:1    0   1G  0 disk
nvme0n1        259:0    0   8G  0 disk
├─nvme0n1p1    259:2    0   8G  0 part /
└─nvme0n1p128  259:3    0   1M  0 part
```

```
[ec2-user@ip-10-0-12-183 ~]$ sudo nvme id-ctrl -v /dev/nvme1n1 | egrep "mn|sn|^00[01]0"
sn      : vol01324f611e2463981
mn      : Amazon Elastic Block Store
0000: 2f 64 65 76 2f 78 76 64 66 20 20 20 20 20 20 20 "/dev/xvdf......."
0010: 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 "................"
```

Amazon EBS volume attachment: Nitro Hypervisor

```
[ec2-user@ip-10-0-12-183 ~]$ lsblk
NAME          MAJ:MIN RM SIZE RO TYPE MOUNTPOINT
nvme1n1       259:1    0   1G  0 disk
nvme0n1       259:0    0   8G  0 disk
├─nvme0n1p1   259:2    0   8G  0 part /
└─nvme0n1p128 259:3    0   1M  0 part
```

With ec2-utils

```
[ec2-user@ip-10-0-12-183 ~]$ sudo /sbin/ebsnvme-id /dev/nvme1n1
EBS Volume ID: vol-01324f611e2463981
Block device : /dev/xvdf

[ec2-user@ip-10-0-12-183 ~]$ ls -la /dev/xvdf
lrwxrwxrwx 1 root root 7 Oct 12 02:55 /dev/xvdf -> nvme1n1

[ec2-user@ip-10-0-12-183 ~]$ ls -la /dev/disk/by-id/nvme-Amazon_Elastic_Block_Store_vol01324f611e2463981-ns-1
lrwxrwxrwx 1 root root 13 Oct 12 02:55 /dev/disk/by-id/nvme-Amazon_Elastic_Block_Store_vol01324f611e2463981-
ns-1 -> ../../nvme1n1
```

Amazon EBS best practices

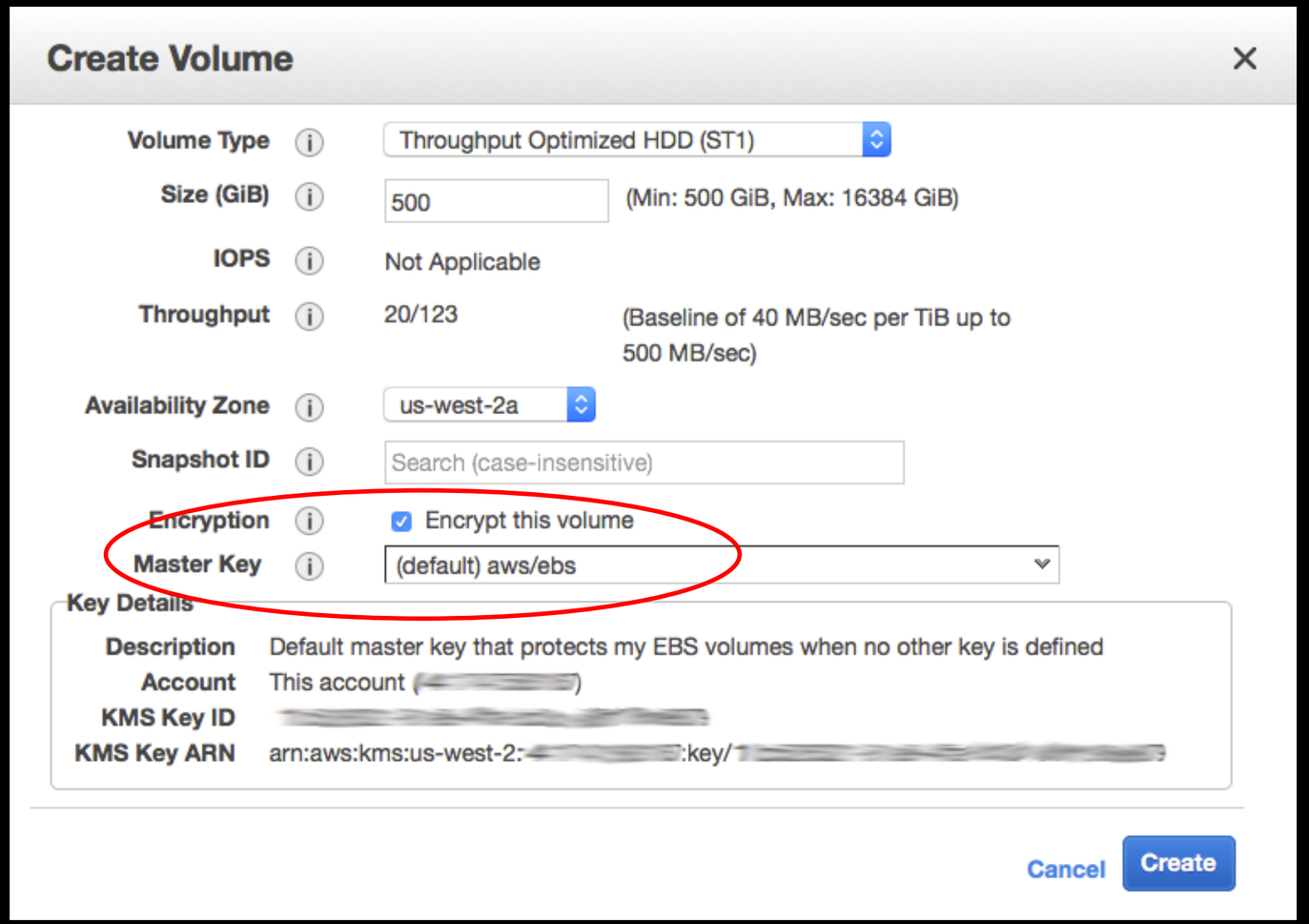
aws SUMMIT

Amazon EBS best practices

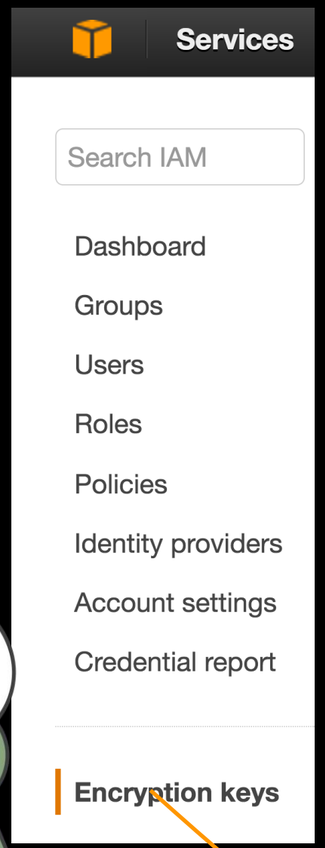
Security

Best practice: Encryption

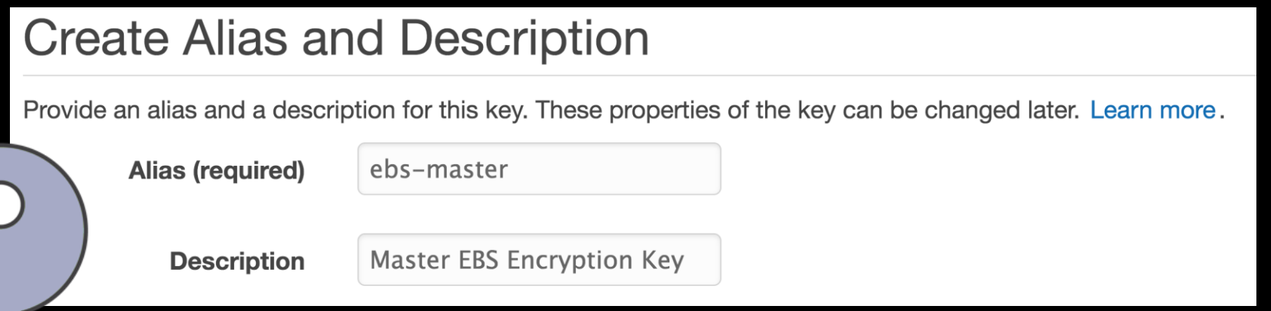
Amazon EBS encryption: Data volumes



Best practice: Encryption



Create a new AWS KMS master key for Amazon EBS

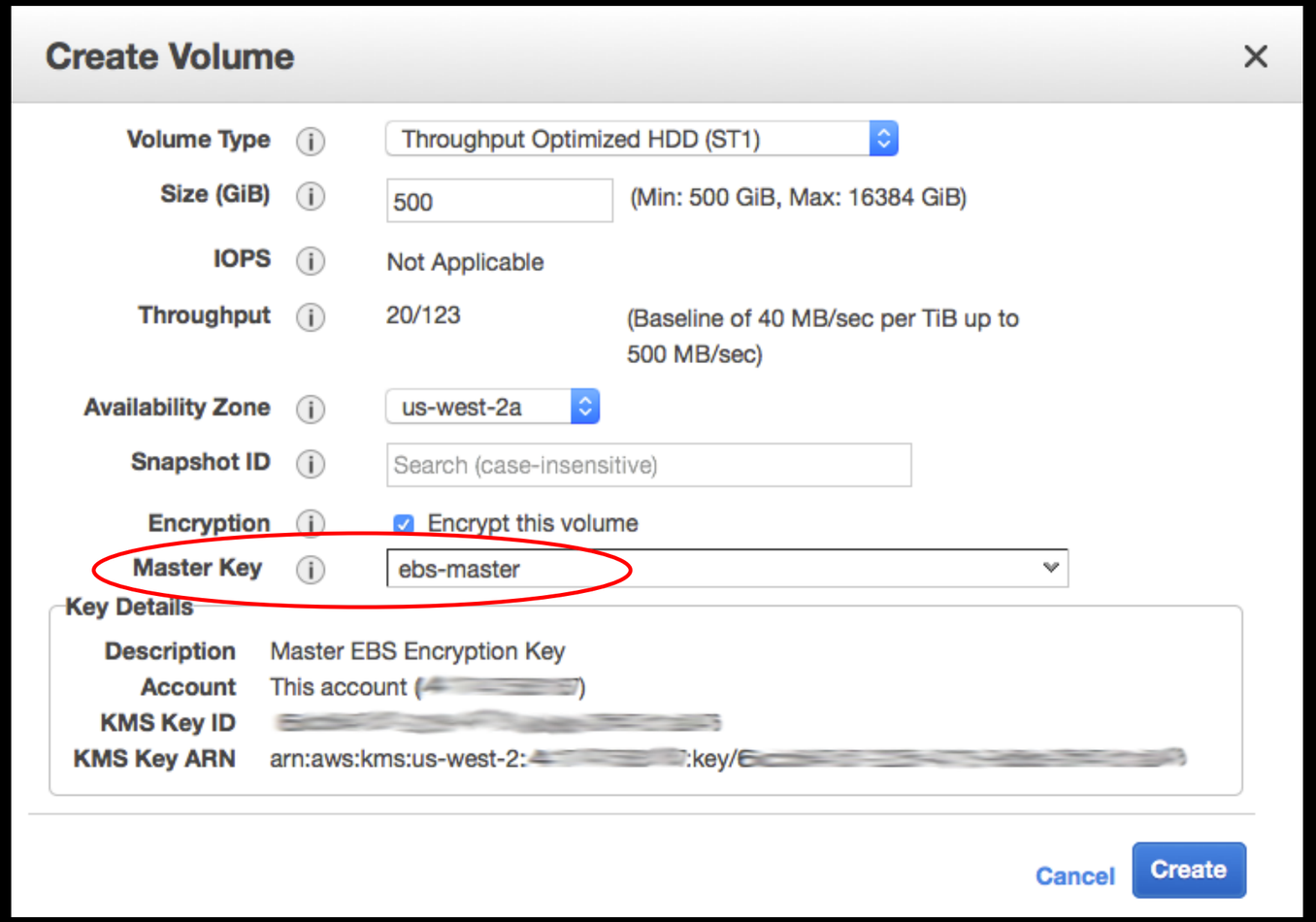


- Define key rotation policy
- Enable AWS CloudTrail auditing
- Control who can use key
- Control who can administer key

aws SUMMIT

Best practice: Encryption

Amazon EBS encryption: Data volumes



Three new features to make encryption easier

- Launch encrypted volumes from unencrypted snapshots/AMIs

  Launch volumes encrypted with different CMKs from encrypted snapshots/AMIs

- Share snapshots encrypted with custom CMKs across accounts

- Encryption by default for Amazon EBS for an account in a Region with a single setting

Feature: Encrypted volumes from unencrypted snapshots or AMIs

aws SUMMIT

Previously…

Unencrypted snapshot/AMI

Encrypted snapshot/AMI

Encrypted Amazon EBS volumes

Similar steps if you need to change encryption on a snapshot/AMI

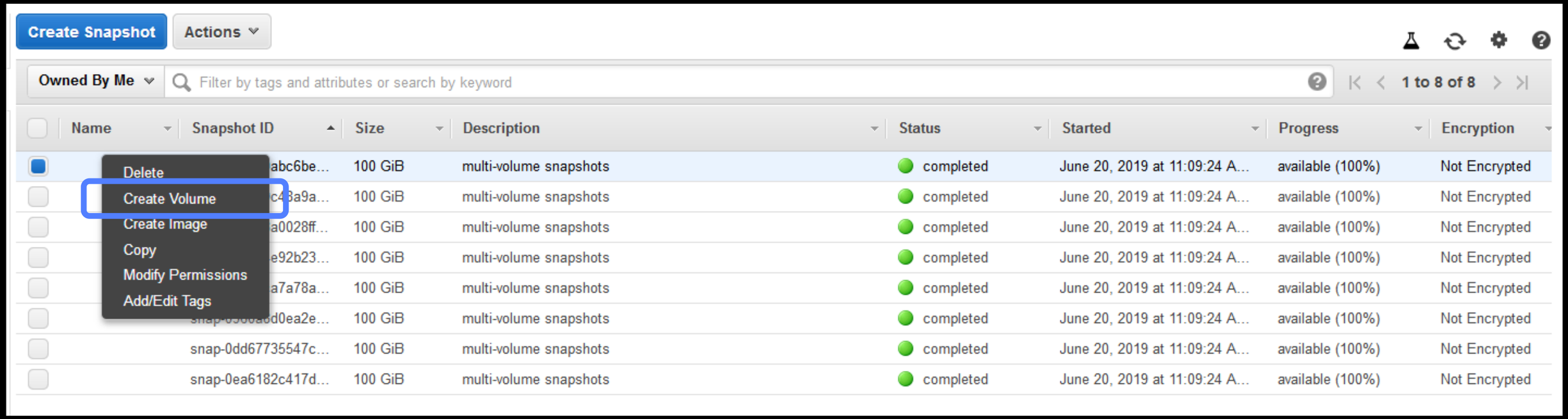
Now...

Unencrypted snapshot/AMI

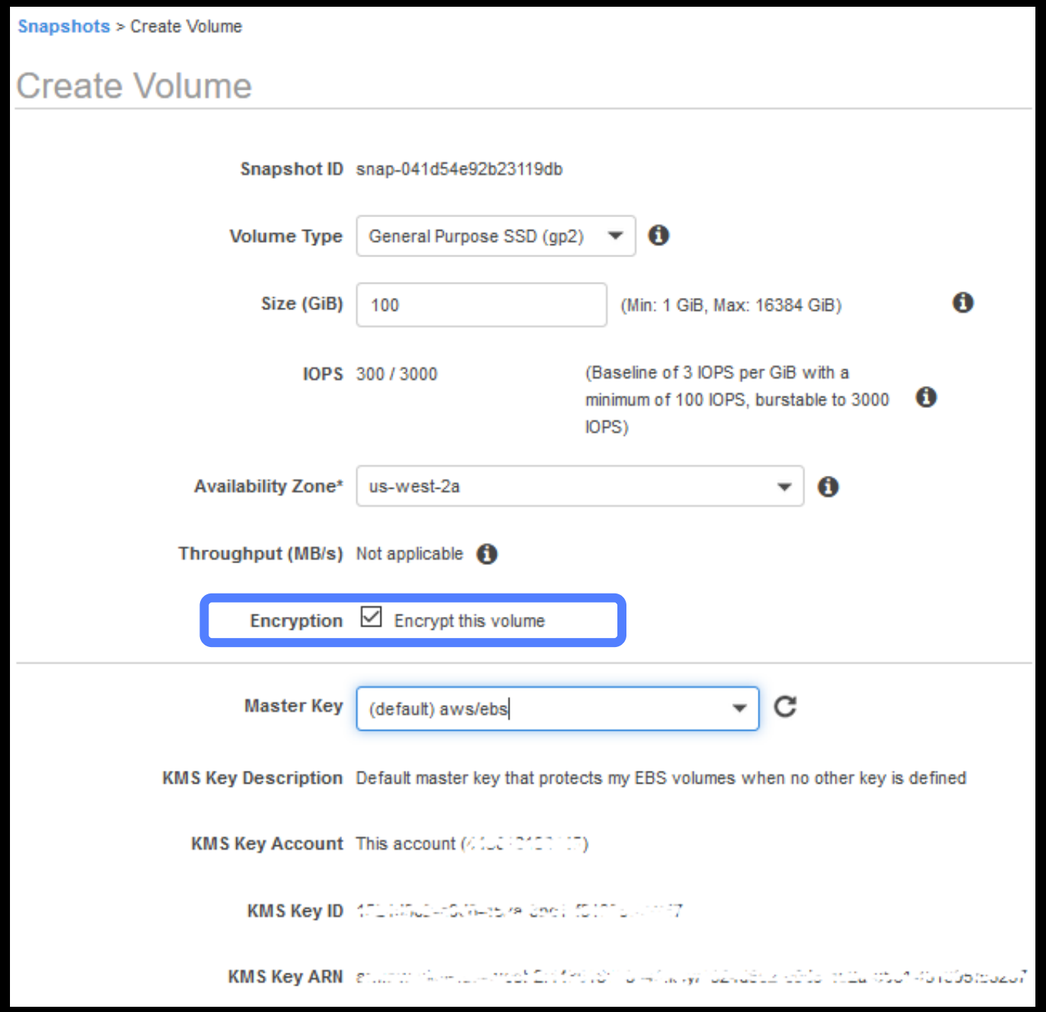
Encrypted Amazon EBS volumes

Similar steps if you need to change encryption on a snapshot/AMI

How to do this



How to do this

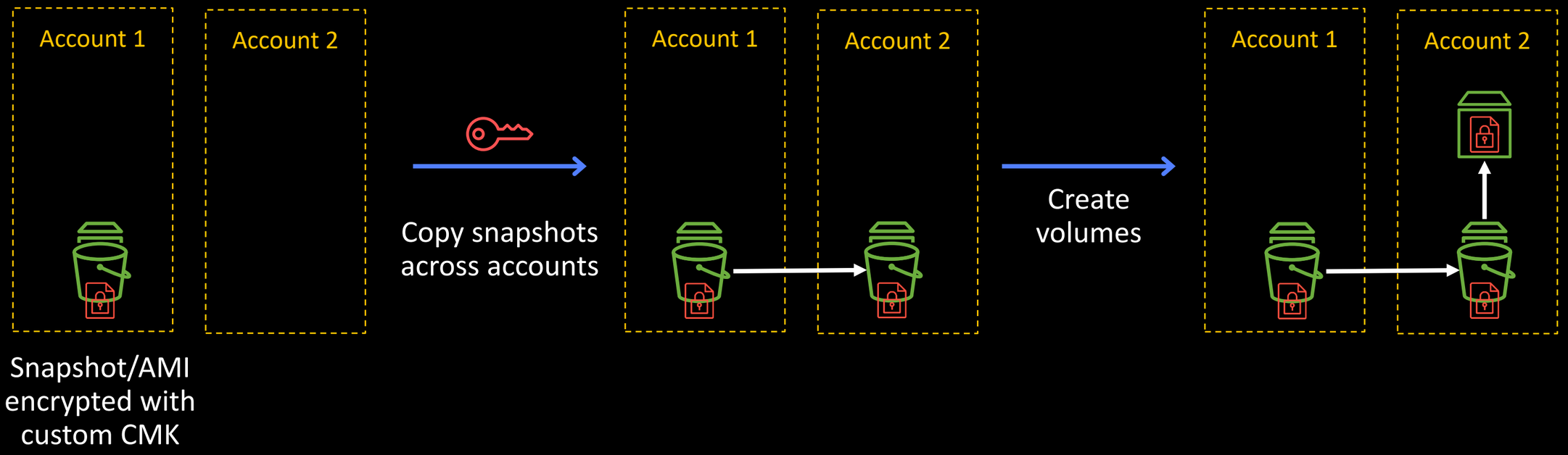



```
aws ec2 create-volume --snapshot-id snap-010bb9c48a9a4c237 --availability-zone us-west-2a --encrypted --volume-type gp2
```

Feature: Share encrypted snapshots/AMIs across accounts

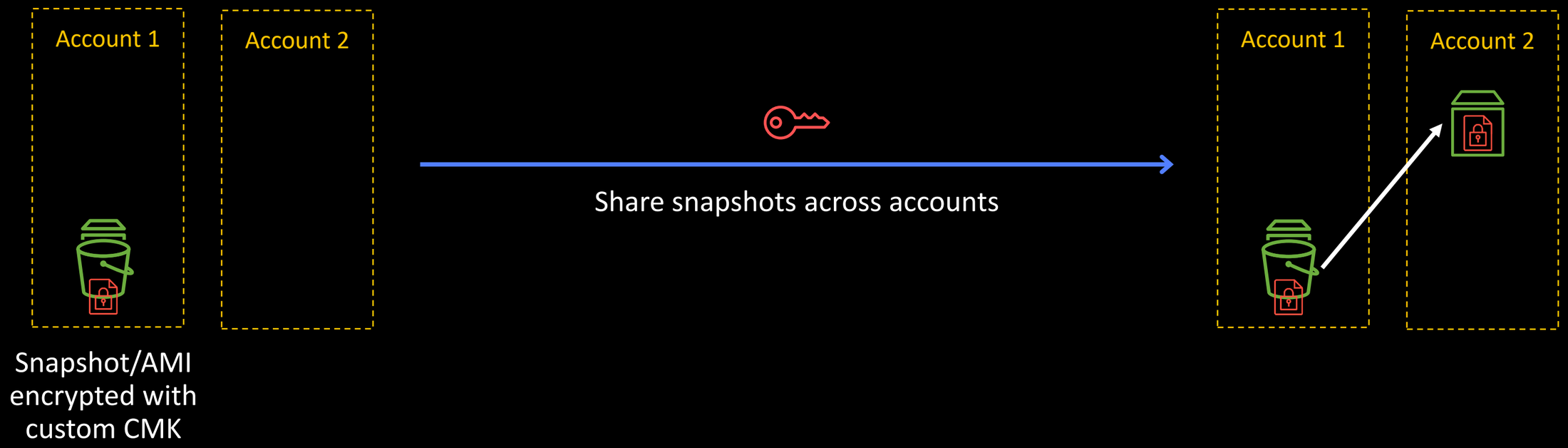
aws SUMMIT

Previously…



Encrypted snapshots could only be copied across accounts

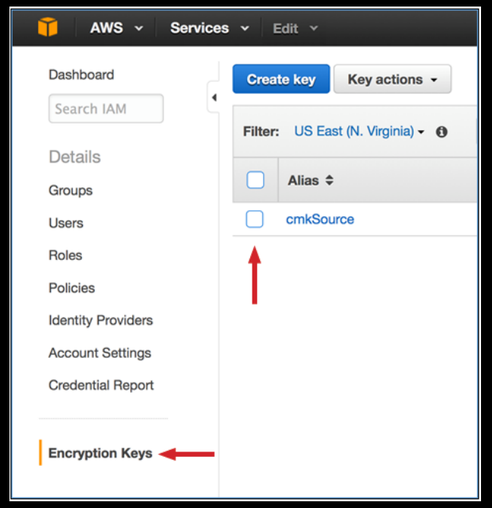
Now...

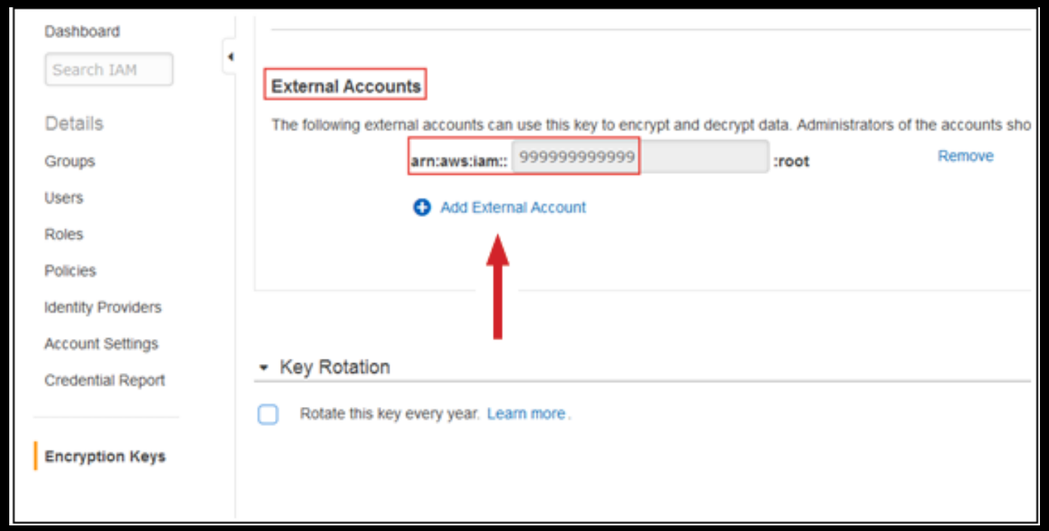


Share snapshots/AMIs encrypted with custom CMKs across accounts

Note: Snapshots/AMIs encrypted with default CMKs cannot be shared across accounts

How to do this






```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "kms:DescribeKey",
                "kms:ReEncrypt*",
                "kms:CreateGrant",
                "kms:Decrypt"
            ],
            "Resource": [
                "arn:aws:kms:us-east-
1:<111111111111>:key/<key-id of cmkSource>"
            ]
        }
    ]
}
```

How to do this

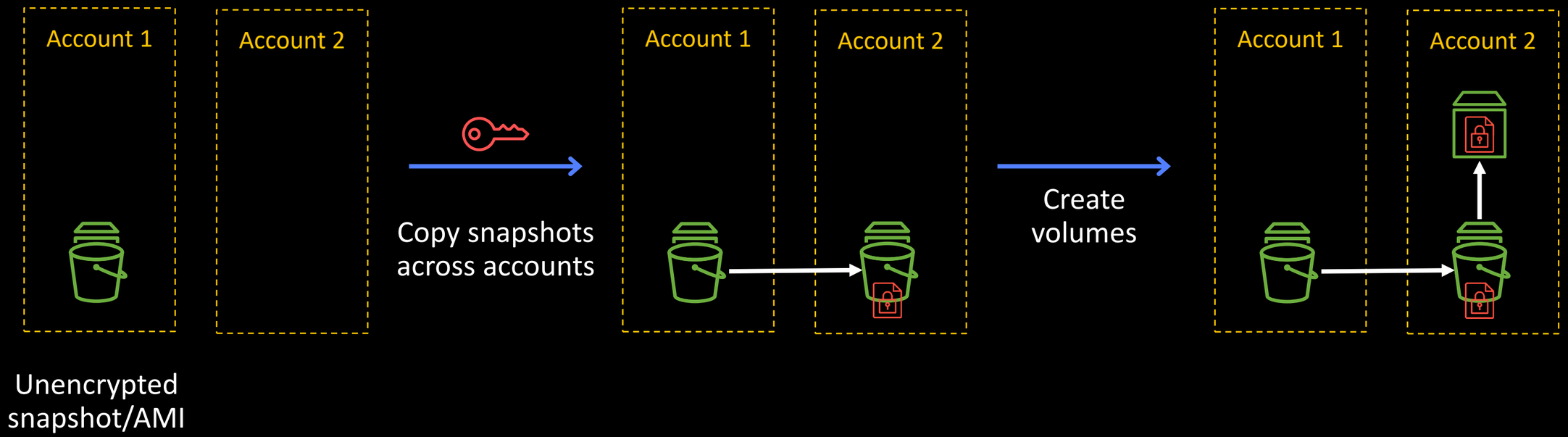
```
$> aws ec2 run-instances
    --image-id ami-XXXXX
    --count 1
    --instance-type m4.large
    --region us-east-1
    --subnet-id subnet-aec2fc86
    --key-name 2016KeyPair
    --security-group-ids sg-f7dbc78e
subnet-id subnet-aec2fc86
    --block-device-mappings
file://mapping.json
```

mapping.json

```
[
    {
        "DeviceName": "/dev/xvda",
        "Ebs": {
                "Encrypted": true,
                "KmsKeyId":
"arn:aws:kms:us-east-
1:<999999999999>:key/<abcd1234-a123-
456a-a12b-a123b4cd56ef>"
        }
    }
]
```

Combining the two features

Previously…



Unencrypted snapshot/AMI

Now...

Launch encrypted volumes across accounts from unencrypted snapshots/AMIs

Note: Snapshots/AMIs encrypted with default CMKs cannot be shared across accounts

Feature: Enable Amazon EBS account-level encryption by default

aws SUMMIT

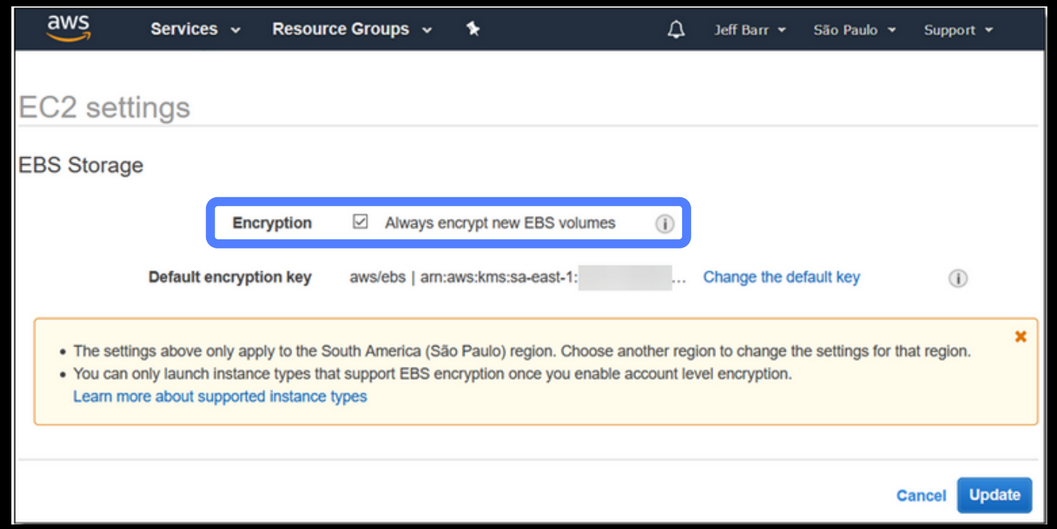
Previously…

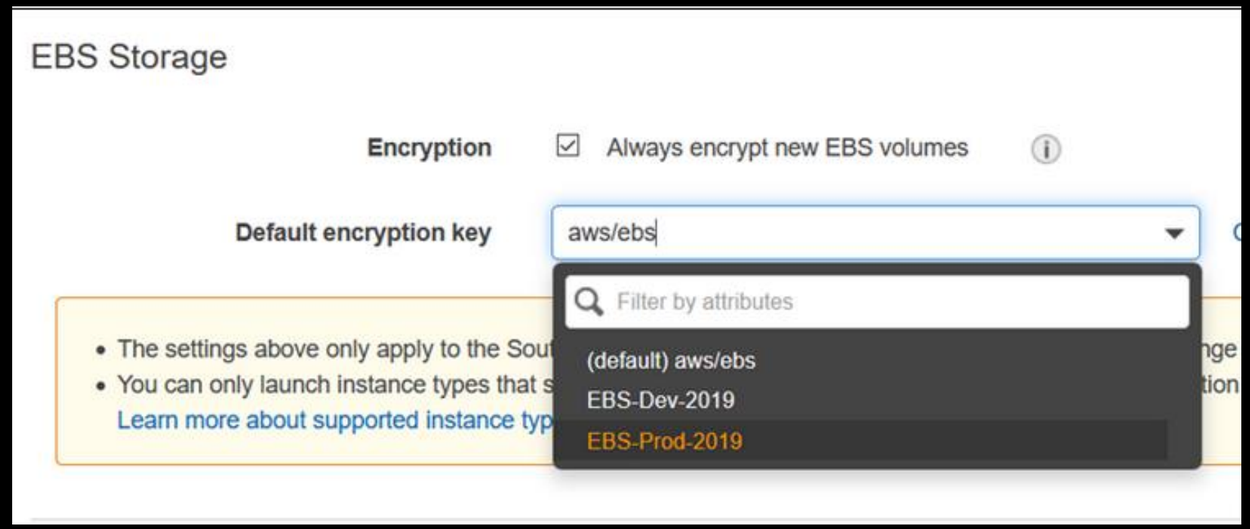
With AWS Identity and Access Management (IAM) policies,
you can prevent unencrypted volumes from launching

Now...

Without change to workflows, newly launched volumes + snapshots are encrypted

How to do this





And that's it!

Amazon EBS best practices

Security

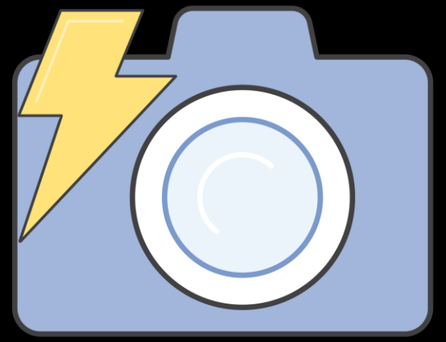
Reliability

Amazon EBS is designed for

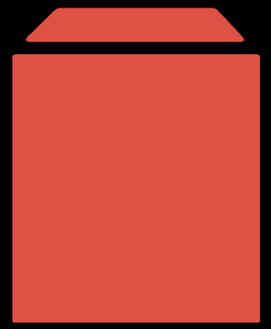
99.999% service availability

0.1–0.2% annual failure rate (AFR)

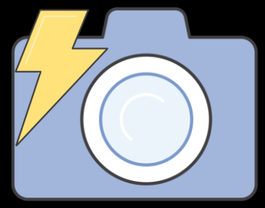
Snapshots



How does an Amazon EBS snapshot work?



EBS volume

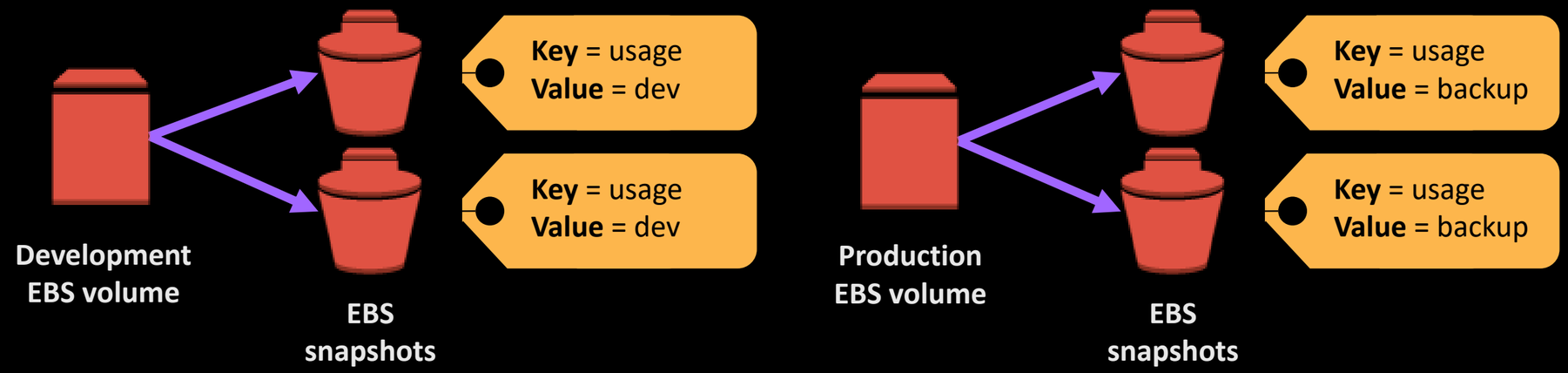


EBS snapshot

- Point-in-time backup of modified volume blocks
- Stored in Amazon S3, accessed via Amazon EBS APIs
- Subsequent snapshots are incremental
- Deleting snapshot only removes data exclusive to that snapshot
- Crash consistent

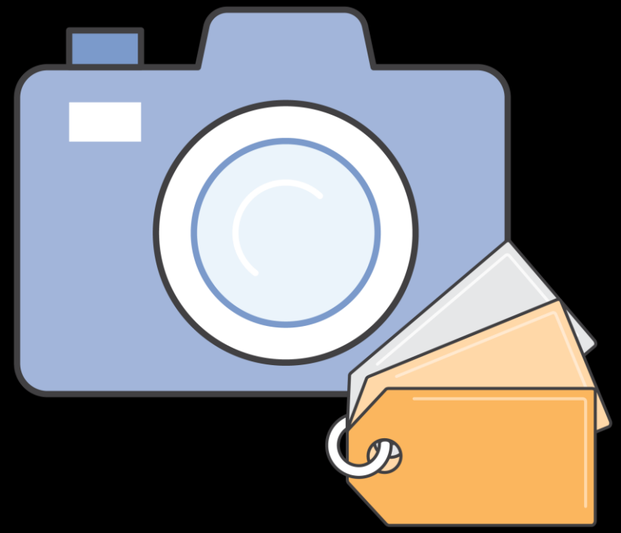
Tracking snapshots and costs (custom tagging)

- Tags provide the ability to assign key/value pairs to AWS resources
- Snapshots support tags for identification and management
- Snapshot tags can be activated as “cost allocation” tags allowing for greater visibility into snapshot storage costs



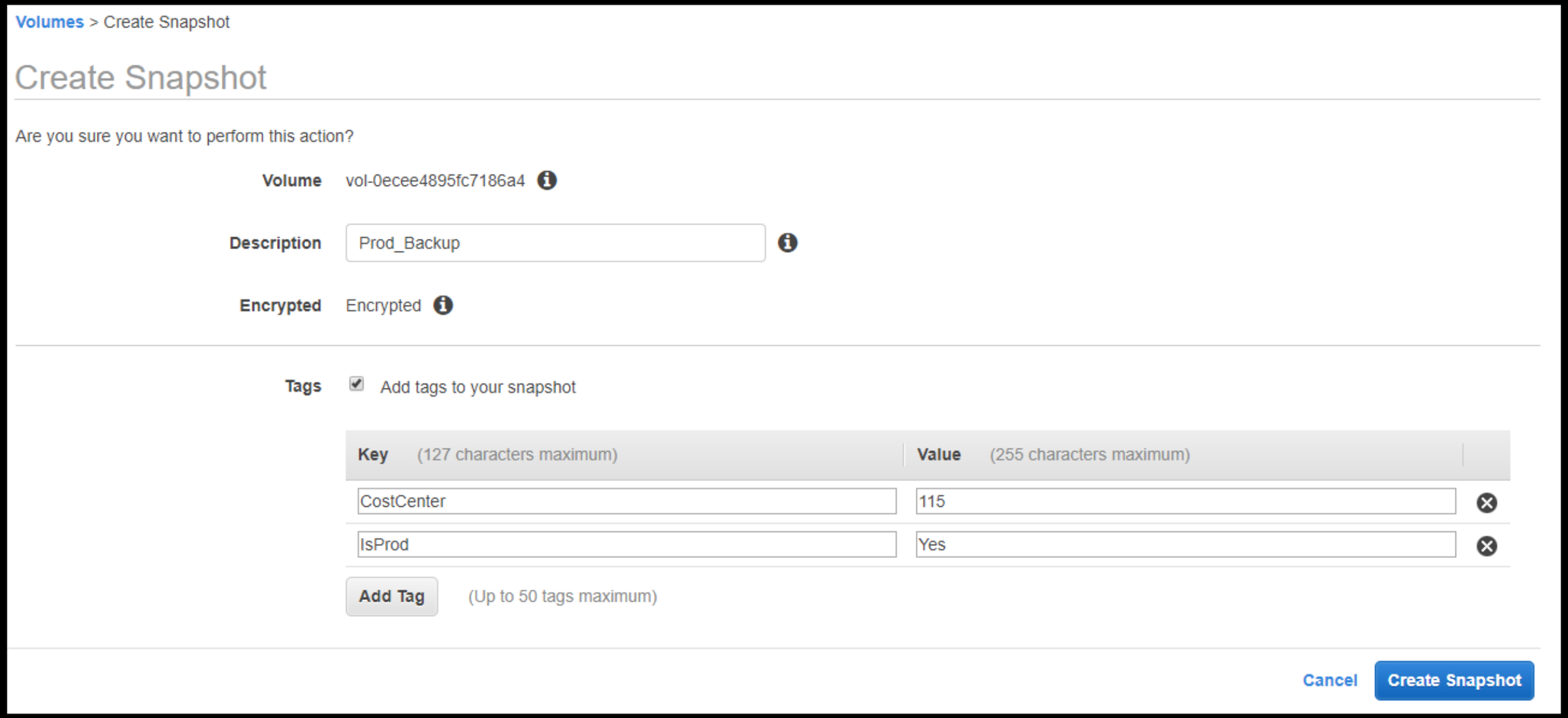
Amazon EBS snapshot tagging upon creation

```
aws ec2 create-snapshot
   --volume-id vol-03f3c34ded2e3398f
   --tag-specifications
    'ResourceType=snapshot,
    Tags=[{Key=CostCenter,Value=115},
    {Key=IsProd,Value=Yes}]'
```



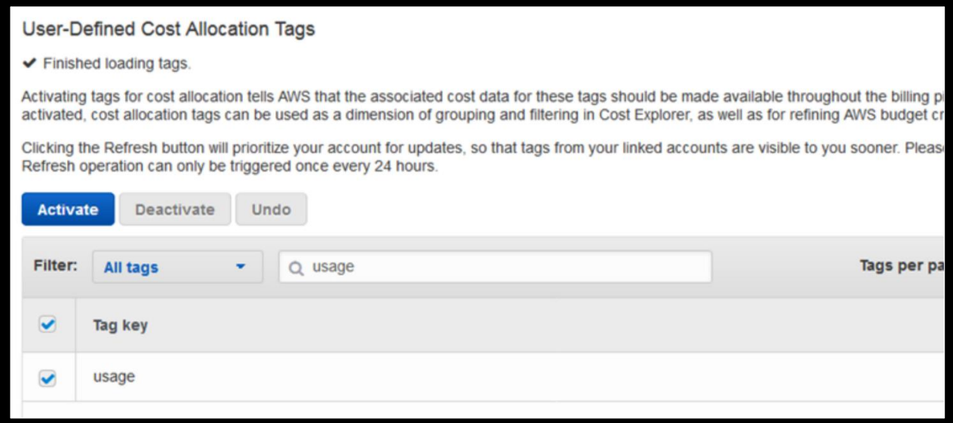
Amazon EBS snapshot tagging upon creation

Specify tags for Amazon EBS snapshots as part of the API call that creates the resource or via the Amazon EC2 console when creating an Amazon EBS snapshot

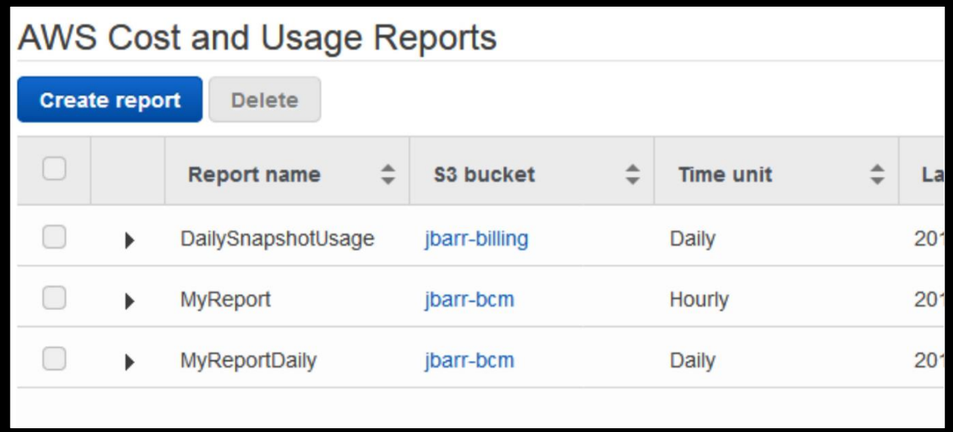


Tracking snapshots and costs (Cost Explorer)

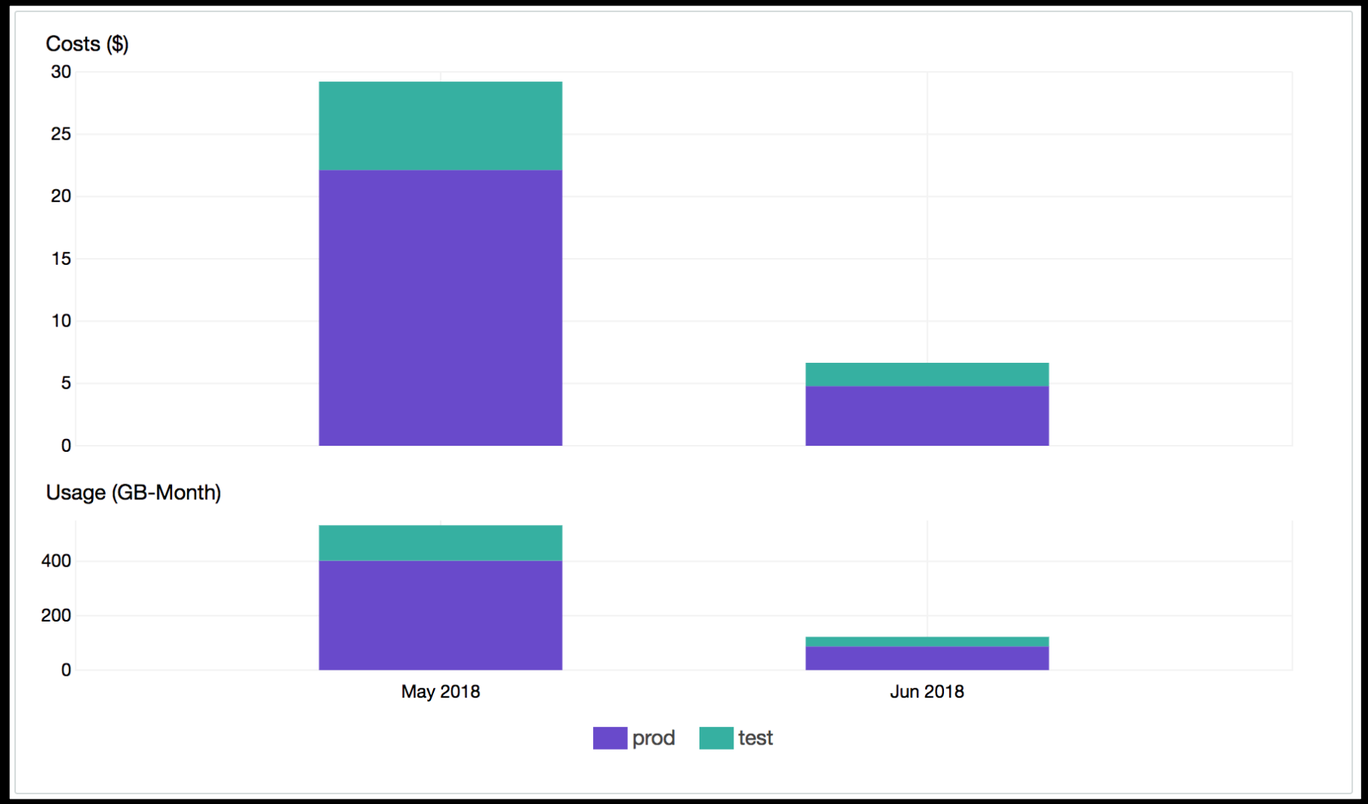
First, activate customer tag for cost allocation



Generate reports with…



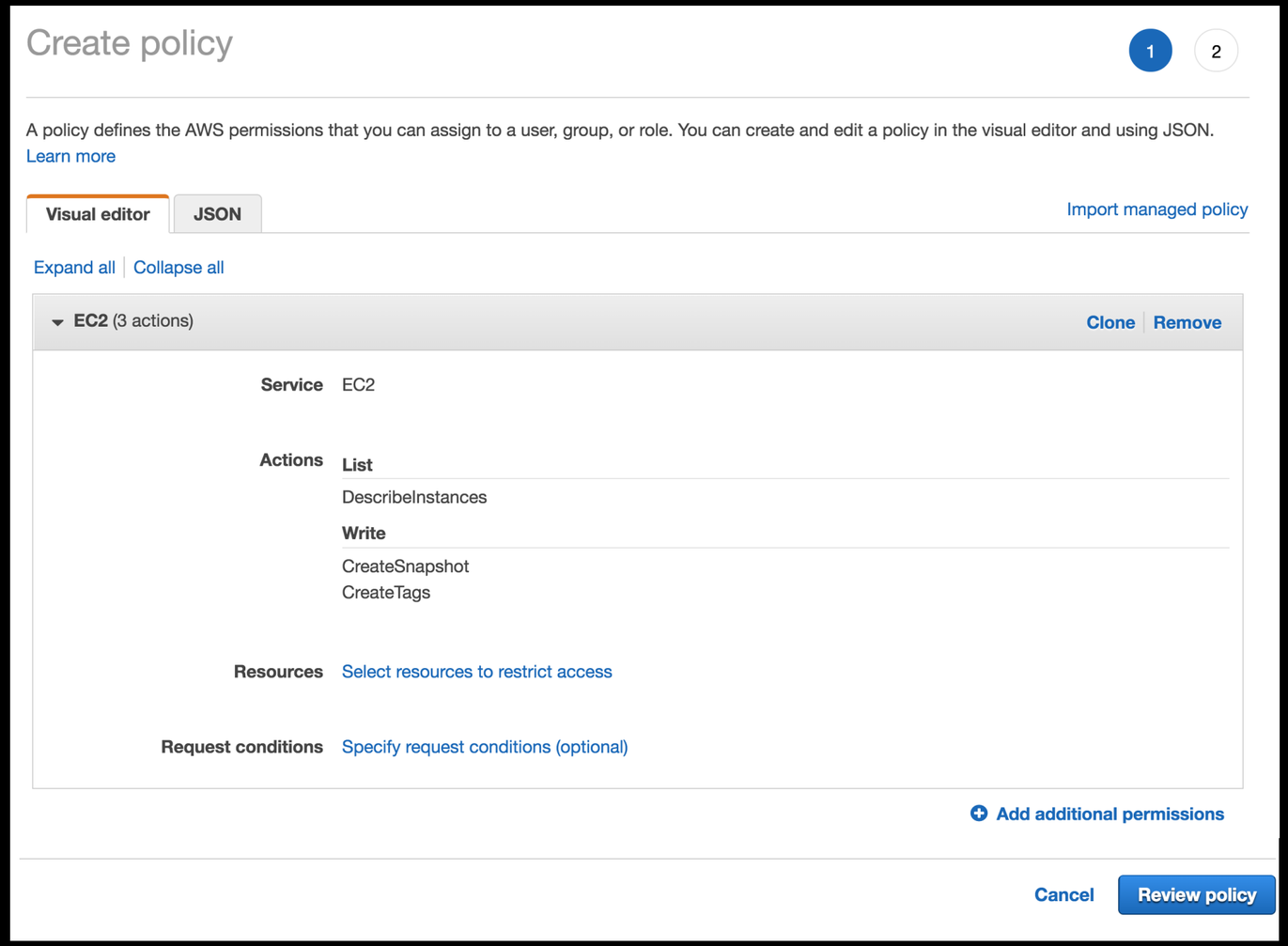
View usage and costs broken down by “usage” tag value
(Example: Metrics, dev, backup)



aws SUMMIT

VSS support via Amazon EC2 SSM

- Use Policy Generator to create IAM policy for AWS service, AWS Systems Manager
- *Actions: DescribeInstances, CreateTags, and CreateSnapshot*
- *Create Amazon EC2 type IAM role and attach to Windows instances*



SSM VSS included in Microsoft Windows Server AMI version 2017.11.21 and up

Resource-level permissions

IAM policies can mandate the use of specific tags when taking actions on Amazon EBS snapshots

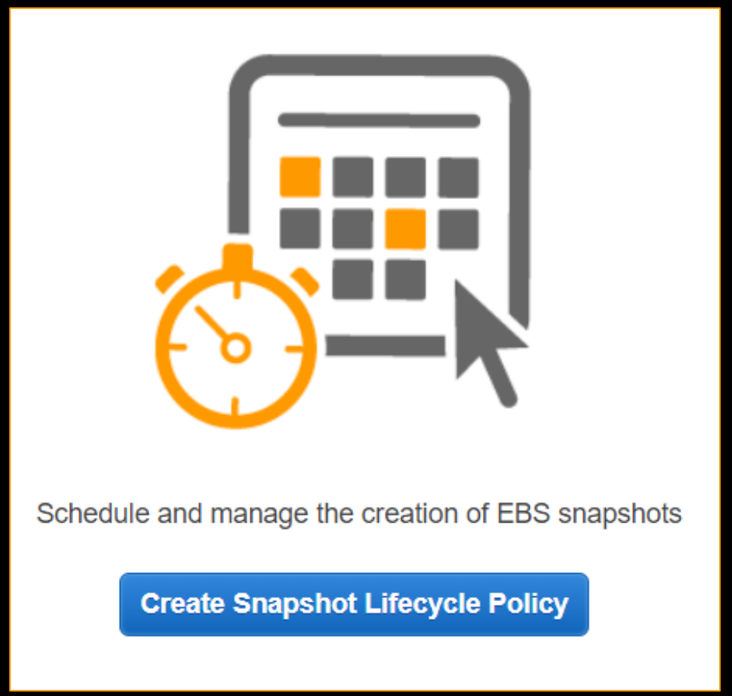
Ideas

- Require use of specific tags
- Specify which users could take snapshots for a given set of volumes
- Restrict access to `DeleteSnapshot`

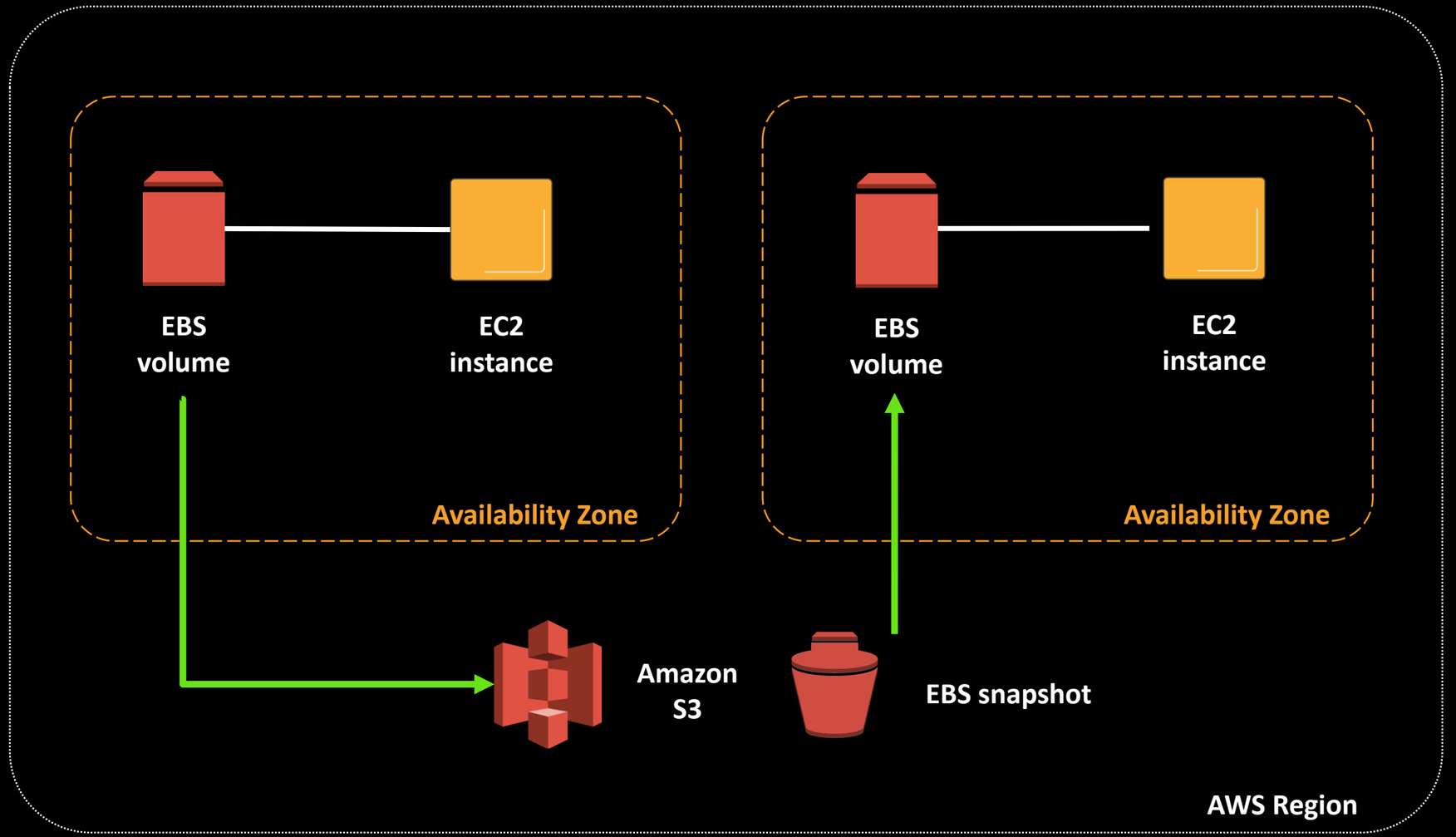
Amazon Data Lifecycle Manager

Simple, automated way to back up data stored on Amazon EBS volumes by ensuring that Amazon EBS snapshots are created and deleted on a custom schedule

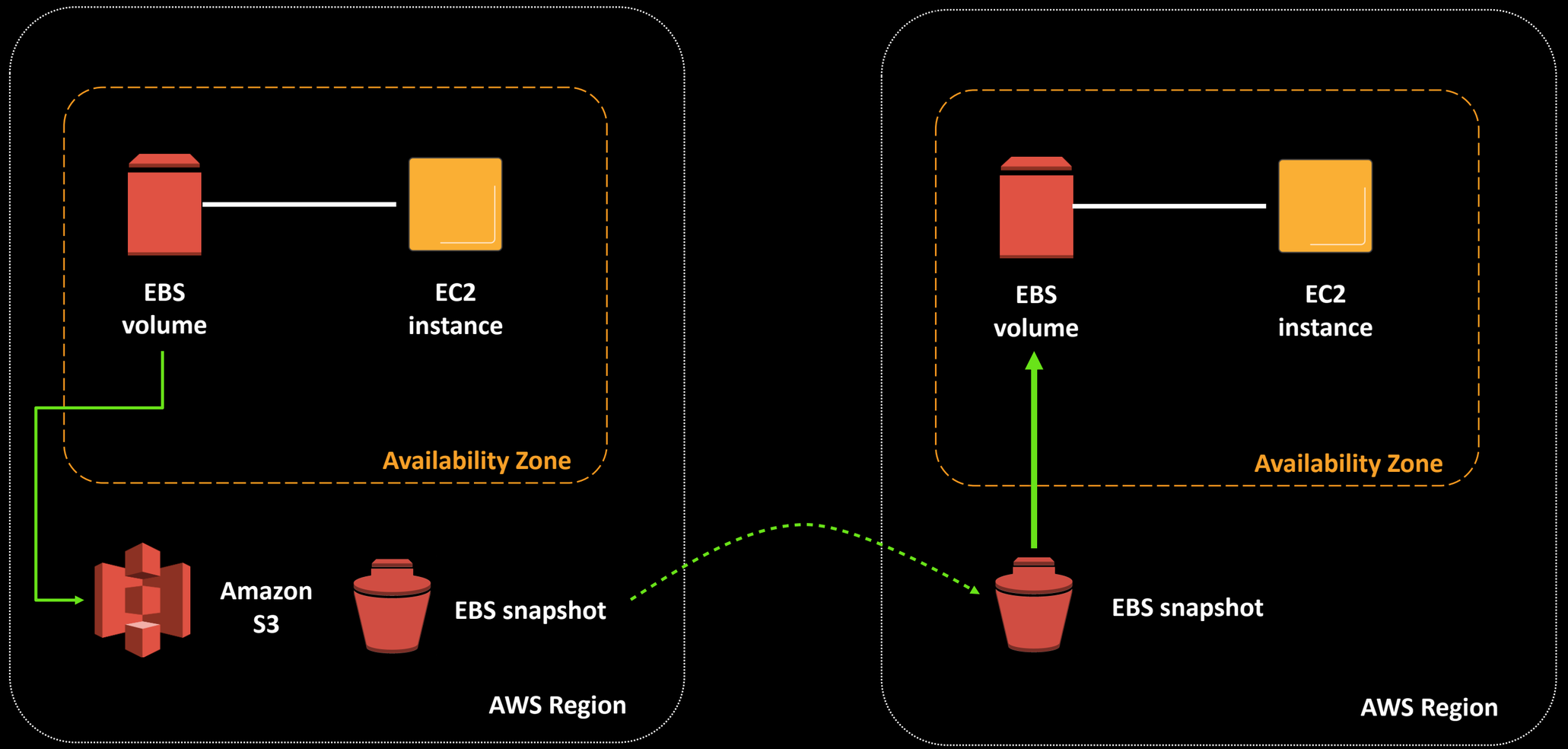
- Use policies to enforce regular backup schedules
- Policies use tags to identify volumes to back up
- Retain backups for compliance/audit purposes
- Control snapshot costs by automatically deleting old backups
- Use IAM to control policy access
- No cost to use



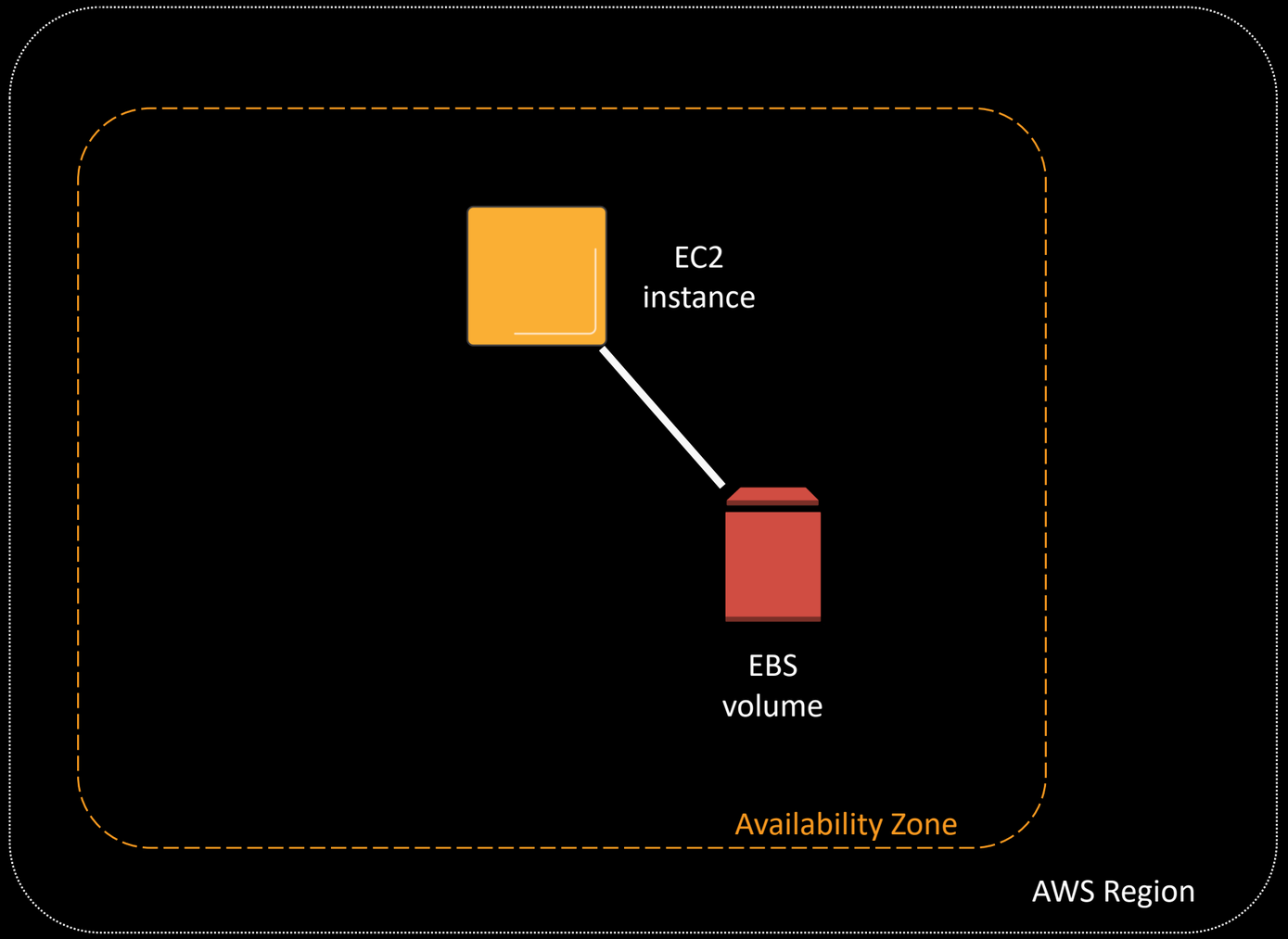
What can you do with a snapshot?



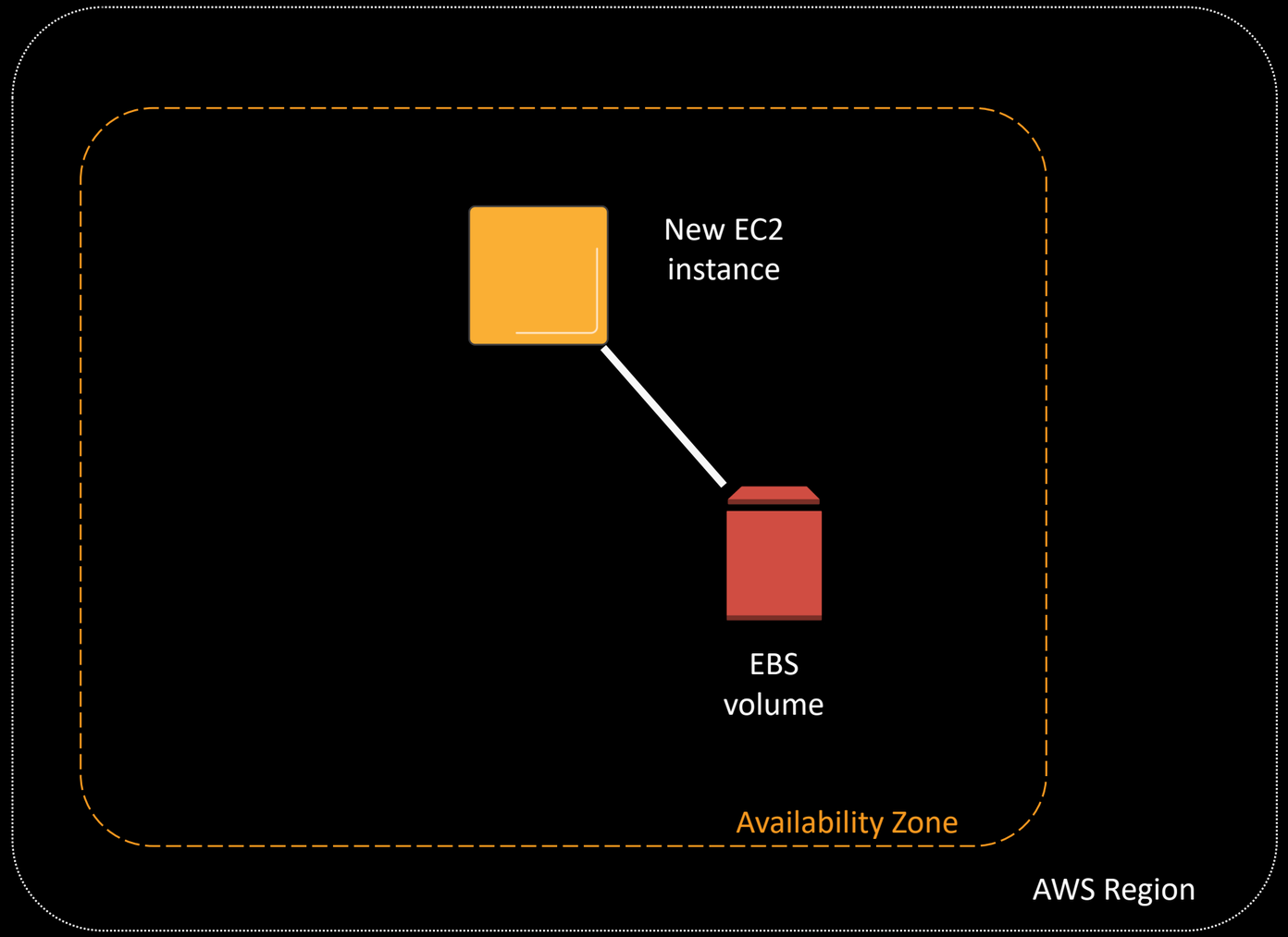
What can you do with a snapshot?



What about EC2 instance failure?



What about EC2 instance failure?



Amazon EBS enables EC2 instance recovery

Amazon CloudWatch
per-instance metric alarm

`StatusCheckFailed_System`

When alarm triggers?

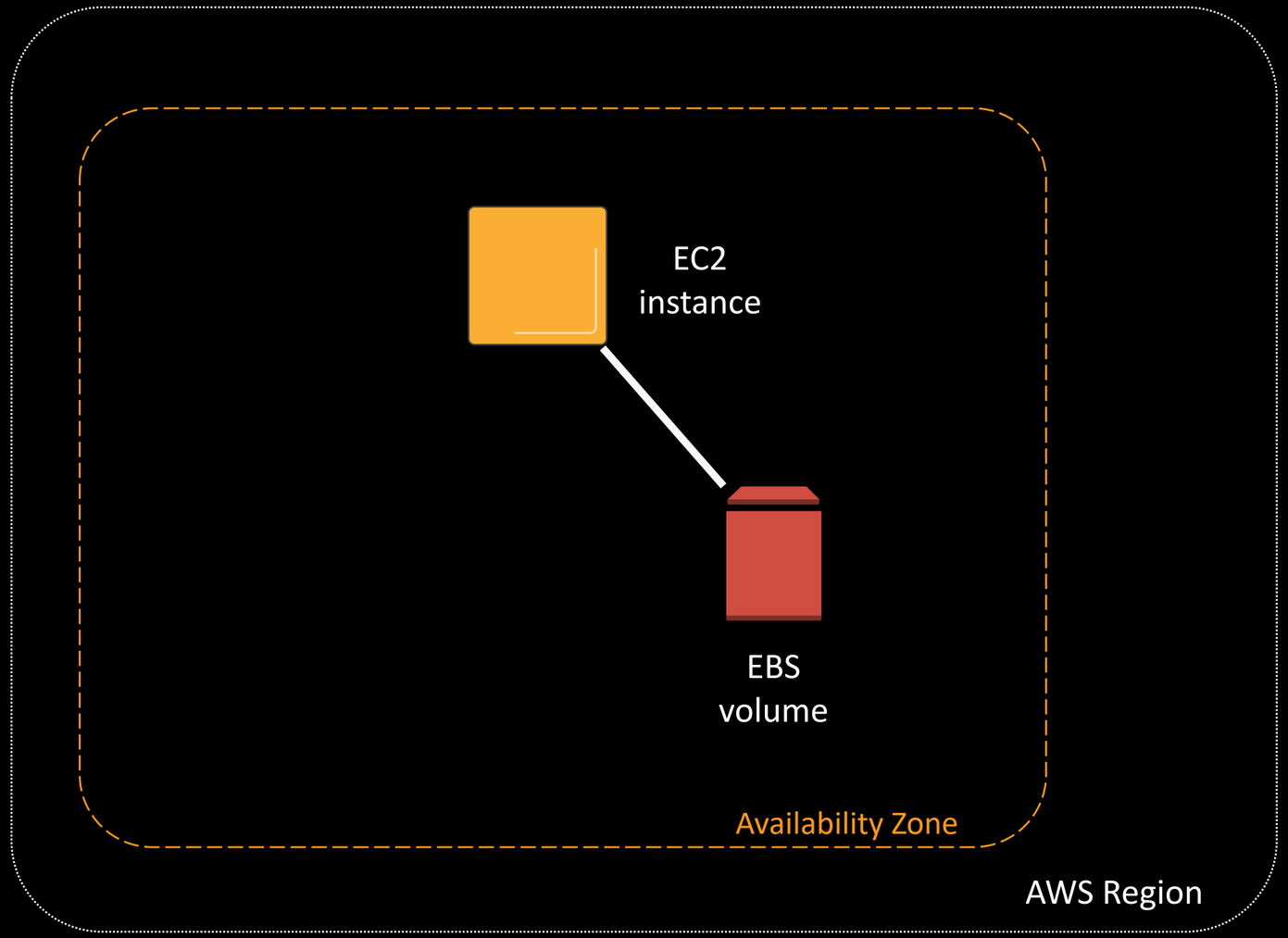
RECOVER instance

Instance retains

- Instance ID
- Instance metadata
- Private IP addresses
- Elastic IP addresses
- EBS volume attachments

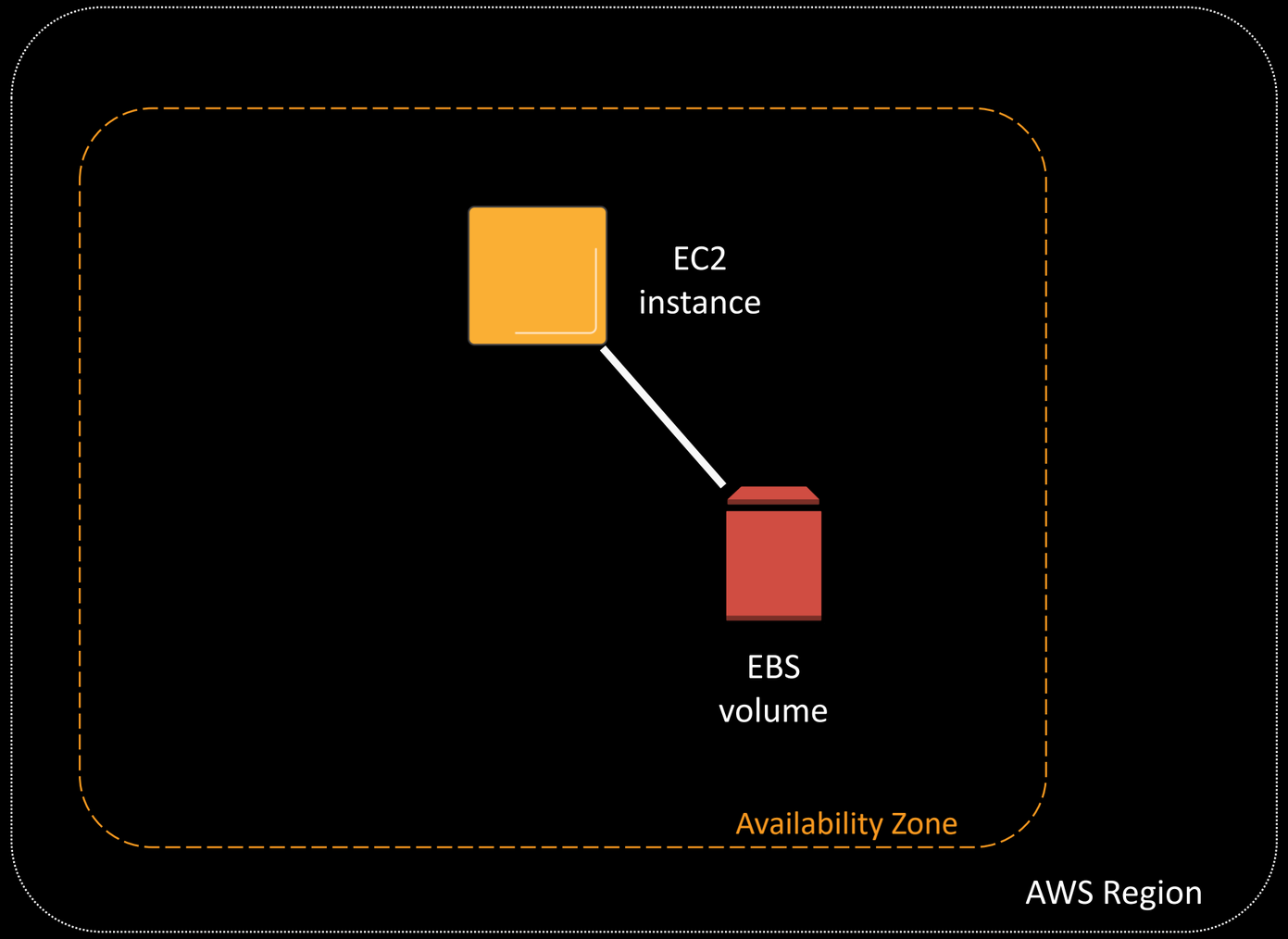
Supported on C3, C4, C5, M3, M4, M5, R3, R4, T2, and X1 instance types within a VPC and with EBS-only storage

What about EC2 instance termination?



`DeleteOnTermination = False`

What about EC2 instance termination?



`DeleteOnTermination = True`

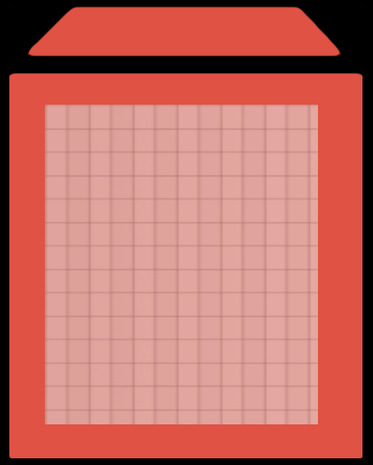
Amazon EBS best practices

Security

Reliability

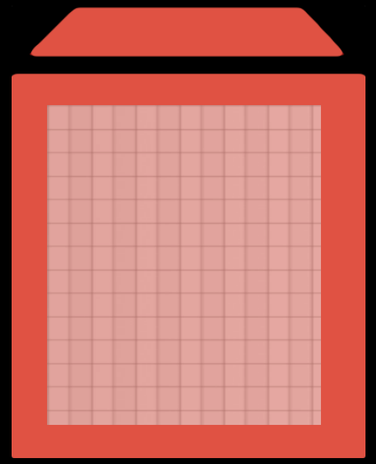
Performance

EBS volume initialization



New EBS volume?

Attach and it’s ready to go



New EBS volume from snapshot?

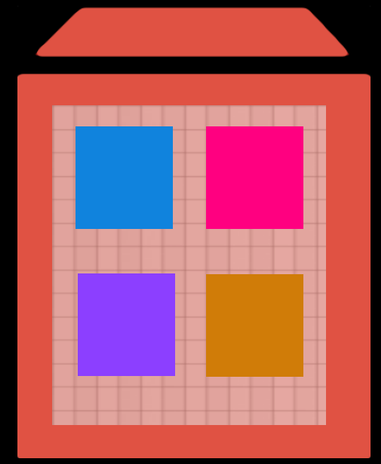
- Initialize for best performance
- Random read across volume

https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/ebs-initialize.html

Best practice: EBS volume initialization

Fio-based example:

```
$ sudo yum install –y fio
$ sudo fio --filename=/dev/xvdf --rw=randread --bs=128k --iodepth=32
--ioengine=libaio --direct=1 --name=volume-initialize
```



https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/ebs-initialize.html

How do we count I/Os?

When possible, we **logically** merge sequential I/Os

...to *minimize* I/O charges on io1,
and *maximize* burst on gp2, sc1, and st1

io1 and gp2: Up to 256 KiB
st1 and sc1: Up to 1 MiB

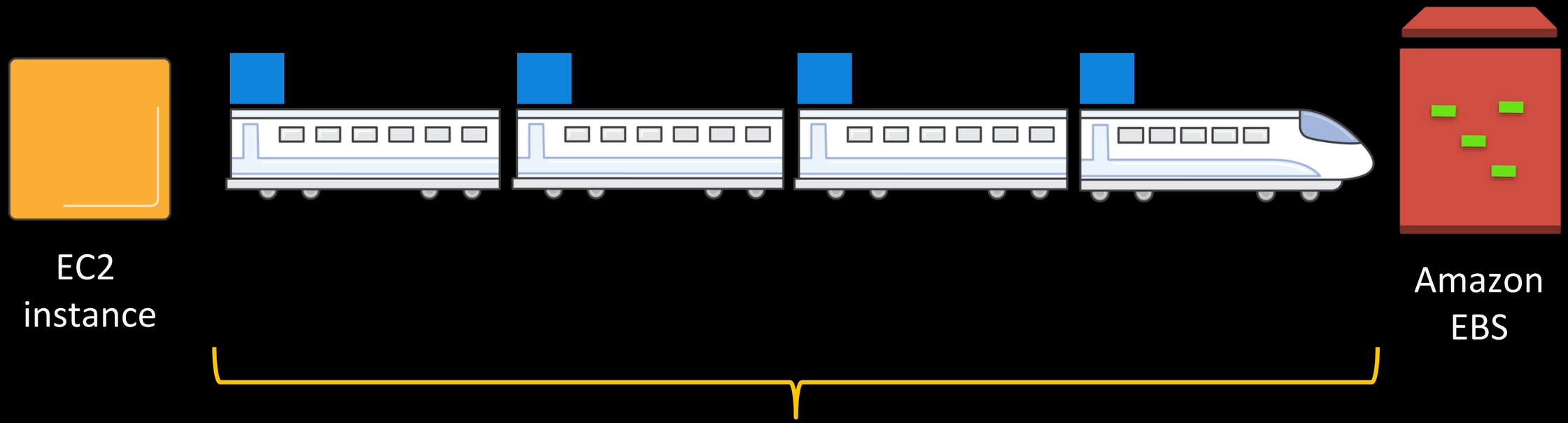
How do we count I/Os?

- Example 1: Random I/Os
- Example 2: Sequential I/O (SSD and HDD)
- Example 3: Large I/O
- Example 4: Mixed I/O

How do we count I/Os?

Example 1: Random I/Os

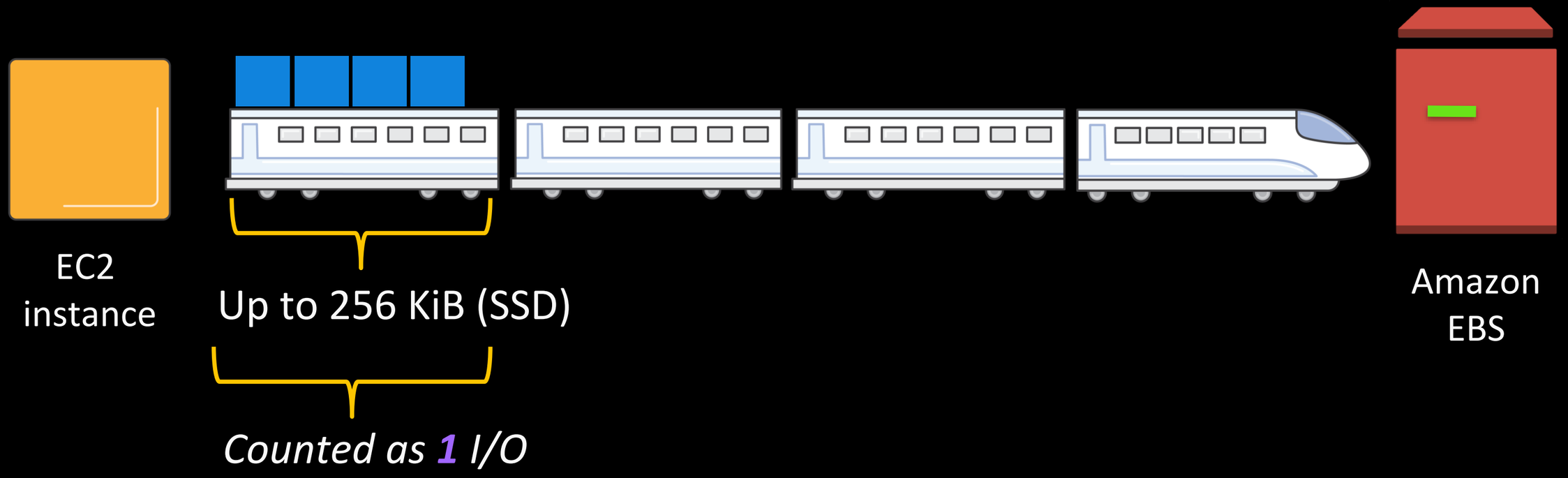
- 4 random I/Os (i.e., non-sequential I/Os)
- Each I/O **64 KiB**



How do we count I/Os?

Example 2: Sequential I/O

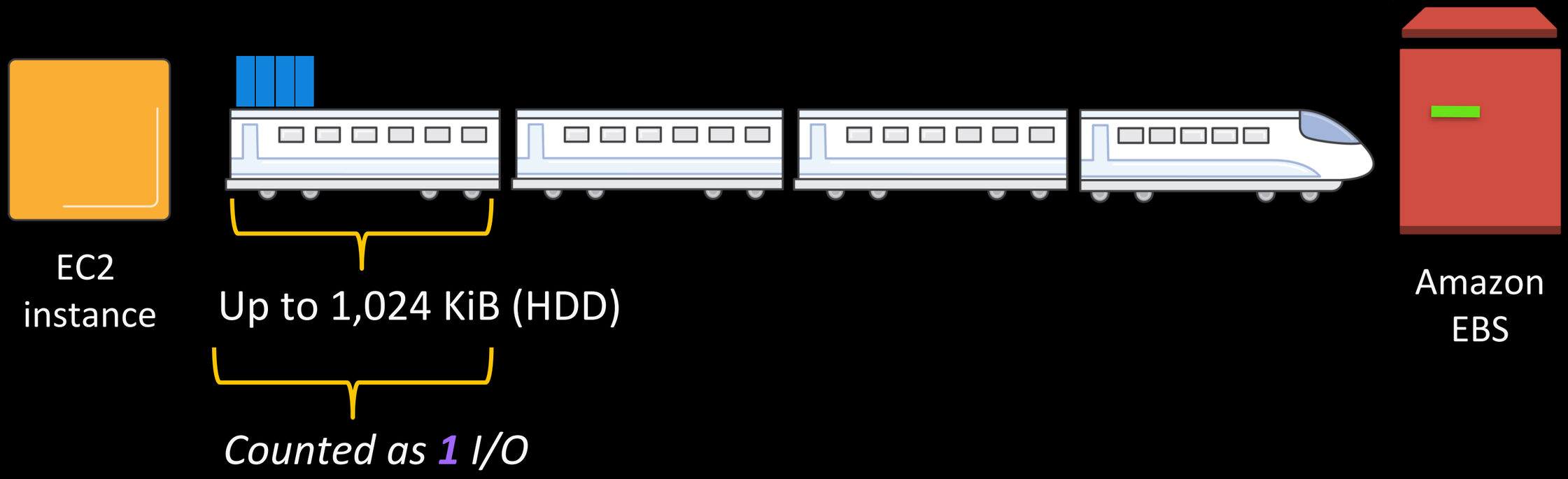
- 4 sequential I/Os
- Each I/O **64 KiB**



How do we count I/Os?

Example 2: Sequential I/O

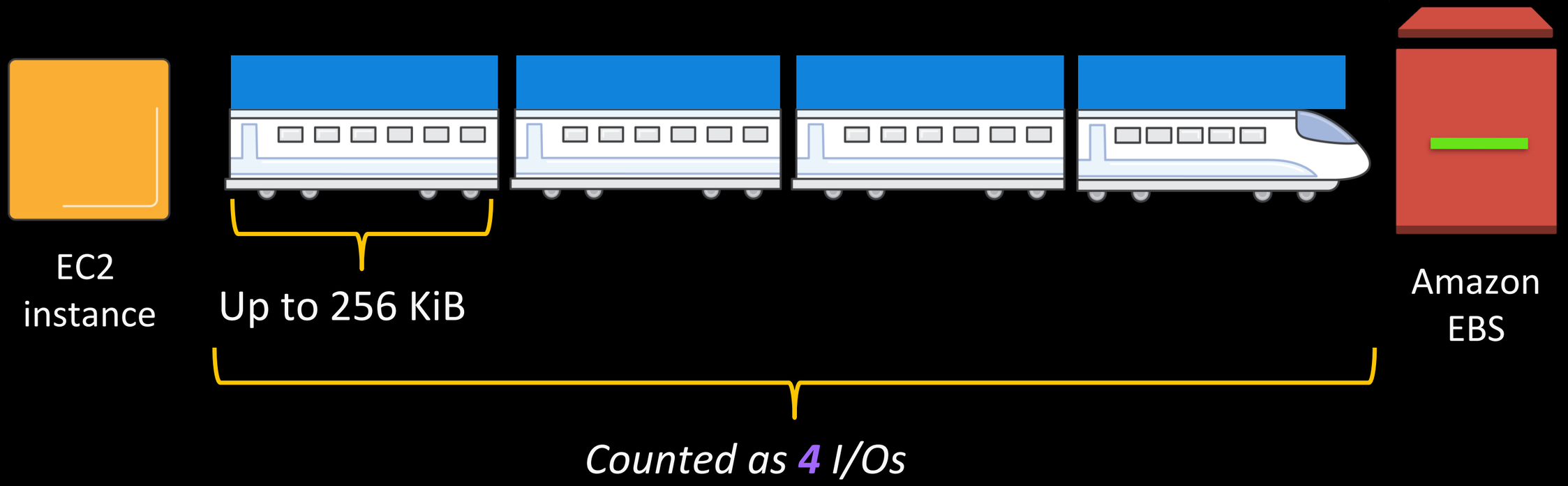
- 4 sequential I/Os
- Each I/O **64 KiB**



How do we count I/Os for gp2 and io1?

Example 3: Large I/O

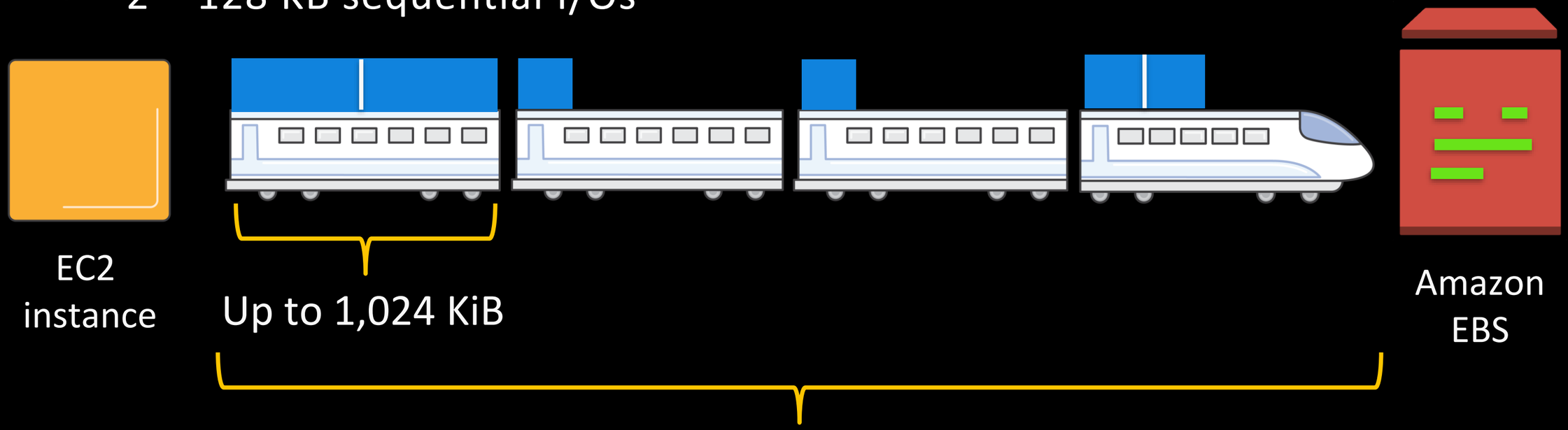
- 1 I/O
- **1,024 KiB**



How do we count I/Os for ST1 and SC1?

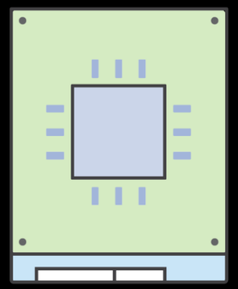
Example 4: Mixed I/O

- 2 * 512 KB sequential I/Os
- 2 * 64 KB random I/Os
- 2 * 128 KB sequential I/Os



*Counted as **4** I/Os or **4** MiB/s of burst (but only ~1.4 MiB of data transferred)*

Burst bucket: gp2

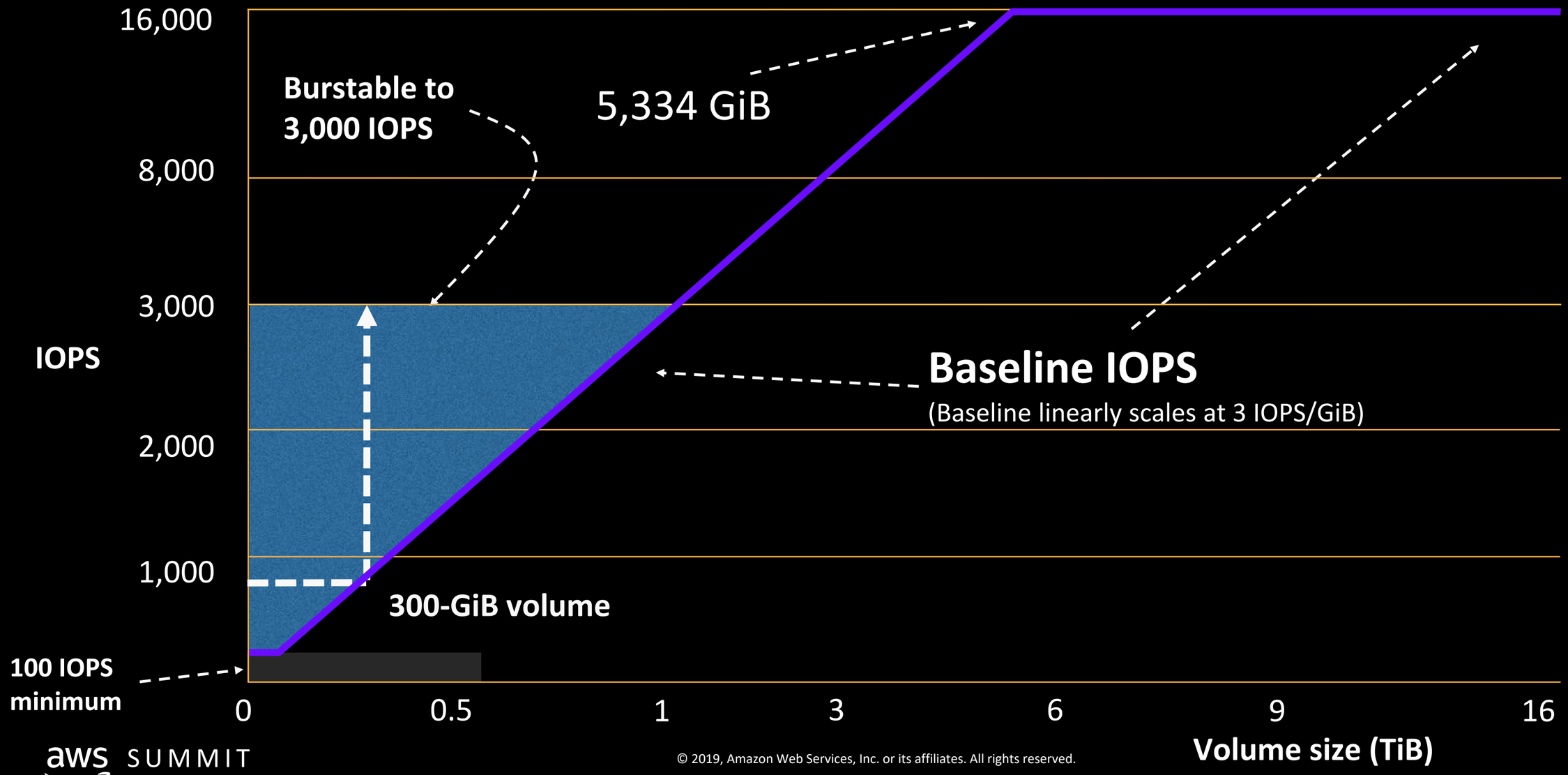


gp2

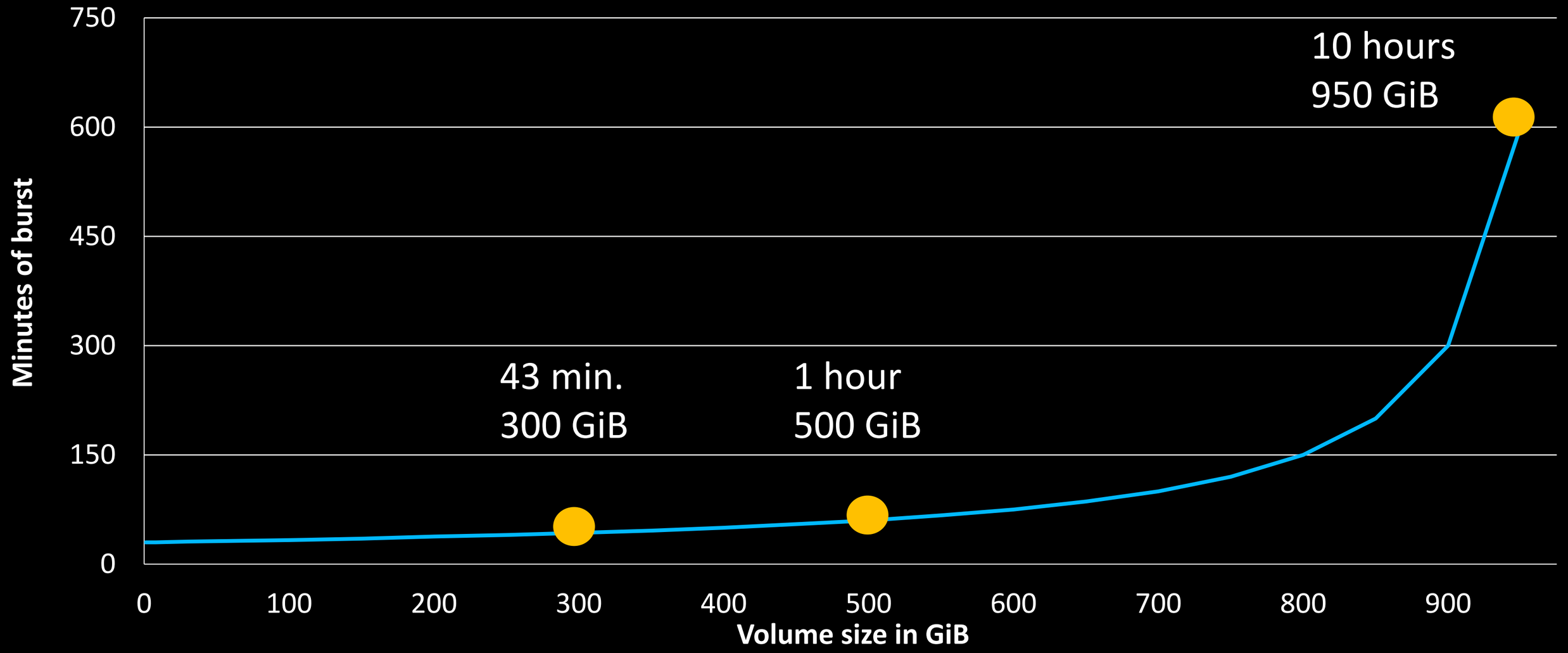
Baseline performance
Always accumulating
3 IOPS per GiB per second

Burst performance
You can spend up to 3,000 credits per second

Burst and baseline: gp2

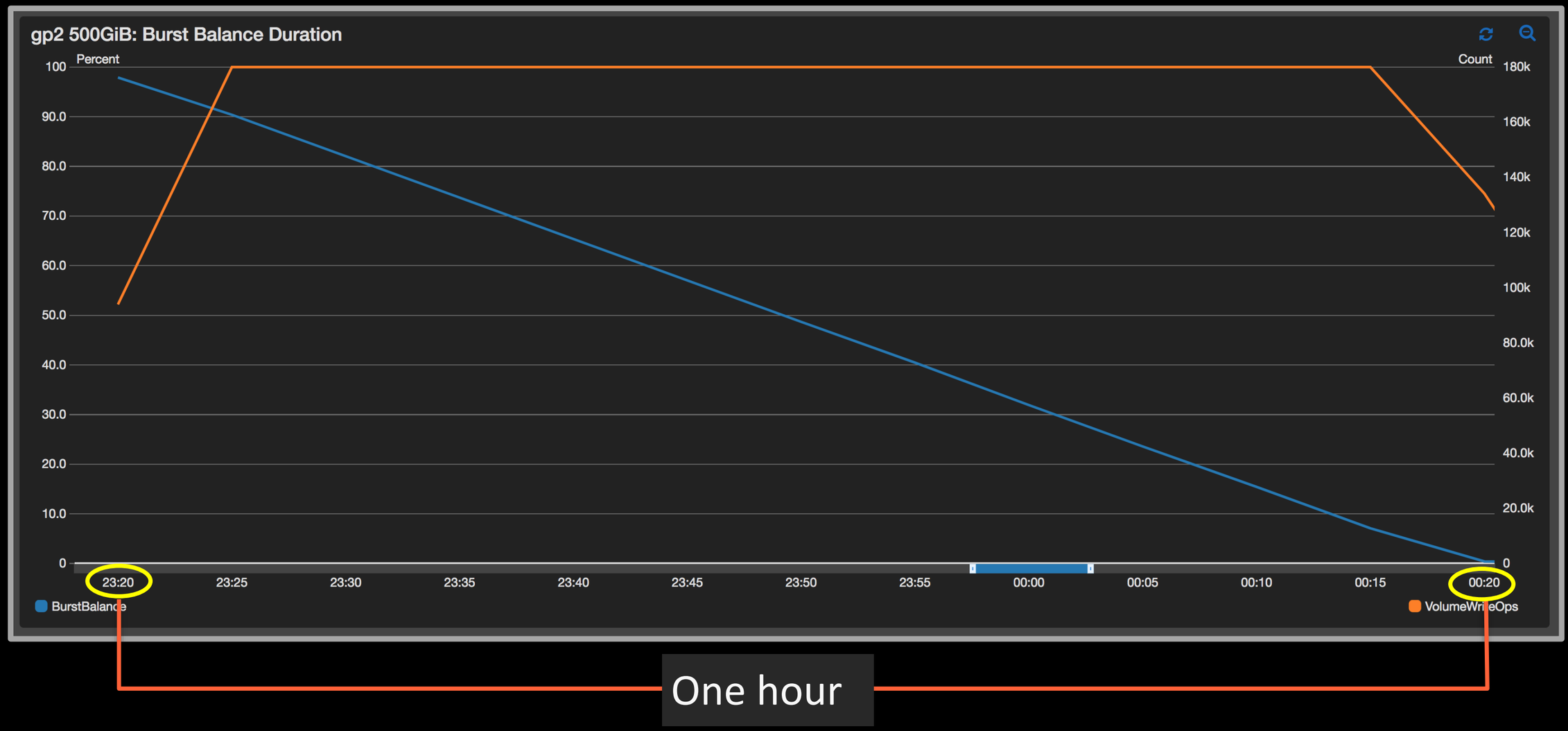


How long can I burst on gp2?



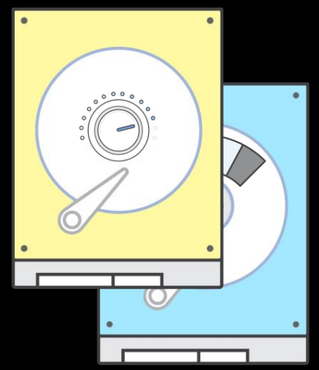
aws SUMMIT

How do I monitor my burst balance?



How do I monitor my burst balance?

Burst bucket: Throughput



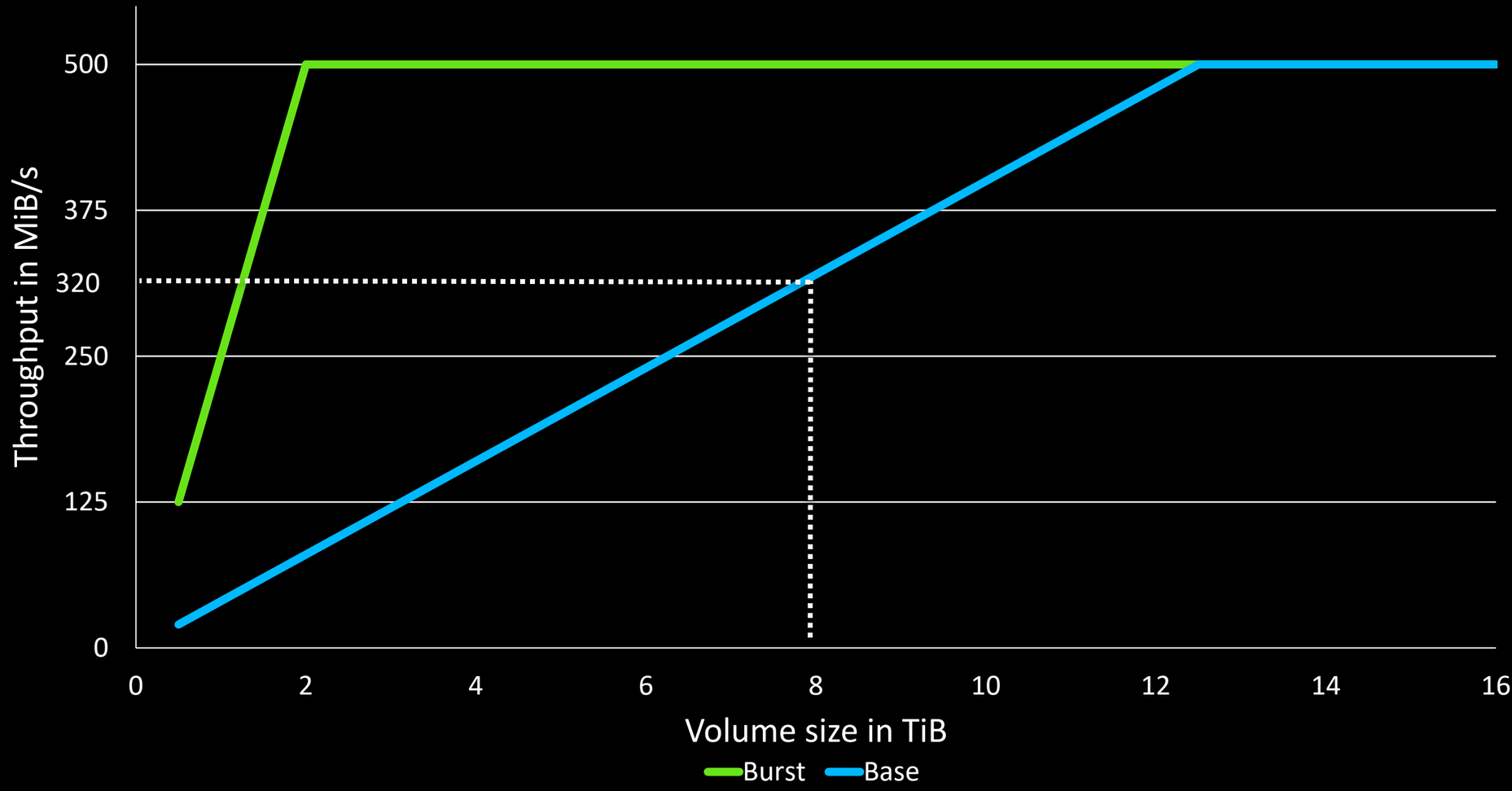
st1/sc1

Baseline performance
Always accumulating
st1: 40 MiB/s per TiB
sc1: 12 MiB/s per TiB

Max I/O bucket credit is 1 TiB of credit per TiB in volume

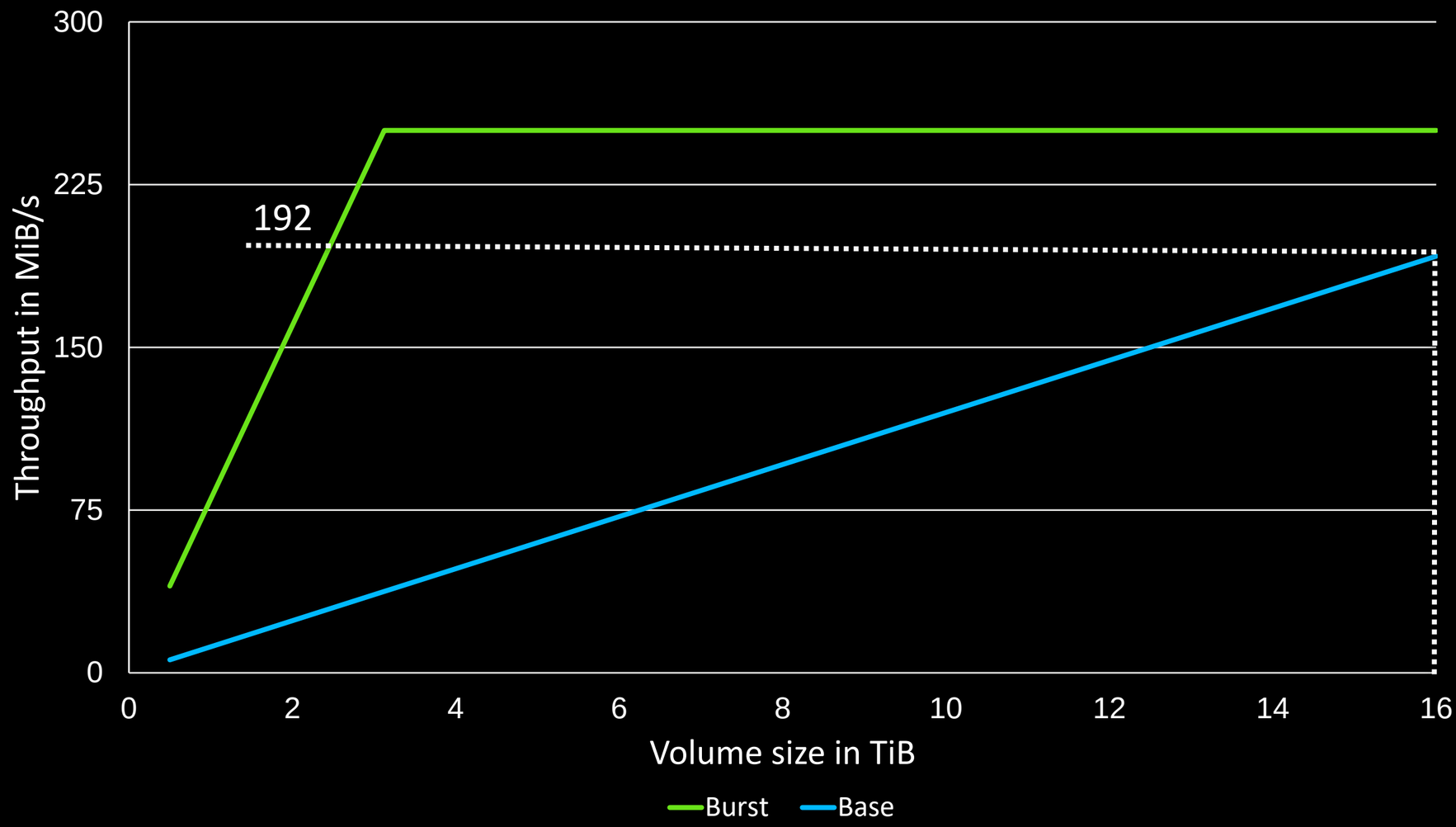
Burst performance
Spend up to
st1: 500 MiB/s
sc1: 250 MiB/s

Burst and baseline: st1



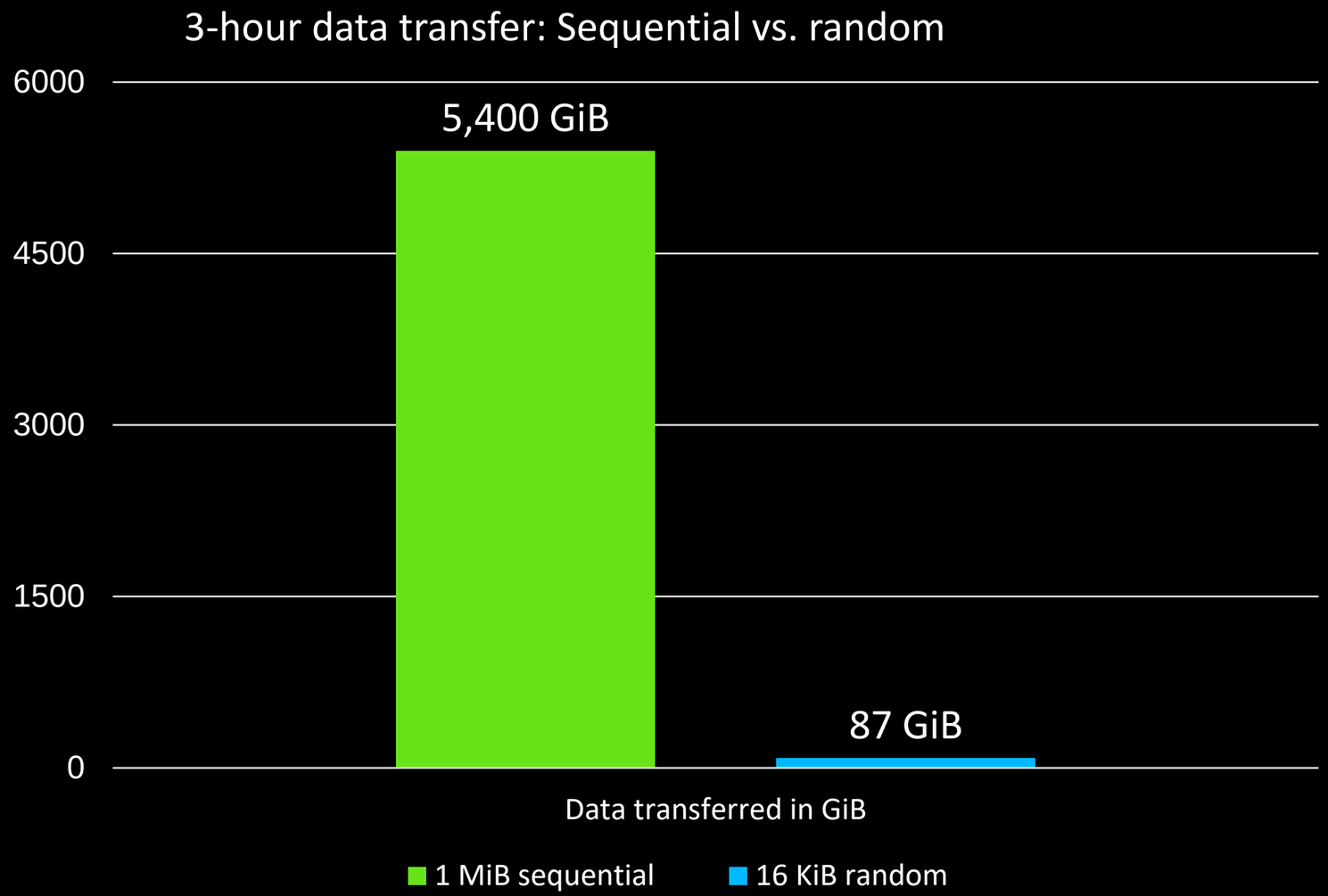
st1

Burst and baseline: sc1



sc1

Burst balance for st1 and sc1



4 TiB st1 column

1 MiB sequential:
5.4 TiB transferred

16 KiB random:
87 GiB transferred

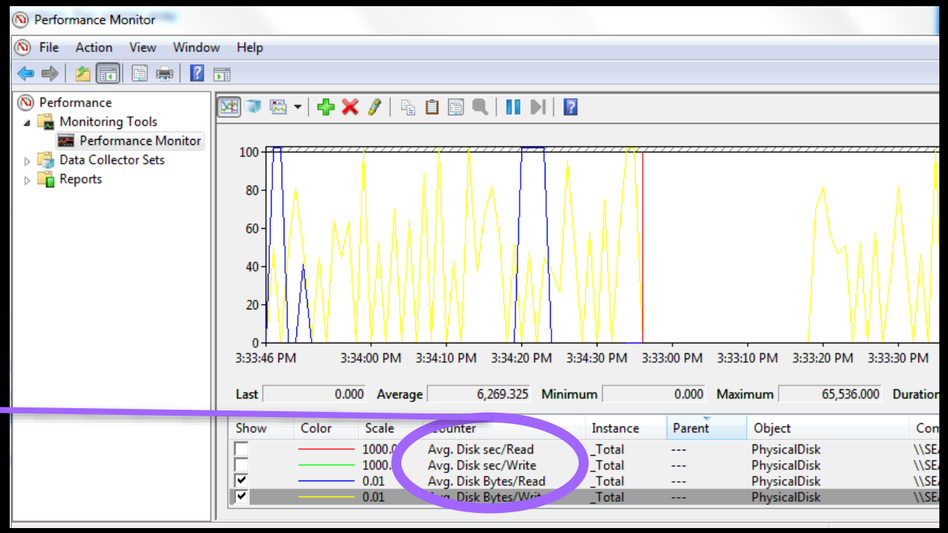
Verify workload I/O patterns

iostat for Linux

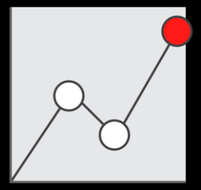
```
$ iostat -xm
Device: rrqm/s   wrqm/s     r/s     w/s    rMB/s    wMB/s  avgrq-sz  avgqu-sz   await  svctm   %util
xvdf      0.00     0.20    0.00  523.40     0.00   523.00   2046.44      3.99    7.62   1.61  100.00
```

2,046 sectors * 512 bytes/sector = ~1,024 KiB

Perfmon for Windows

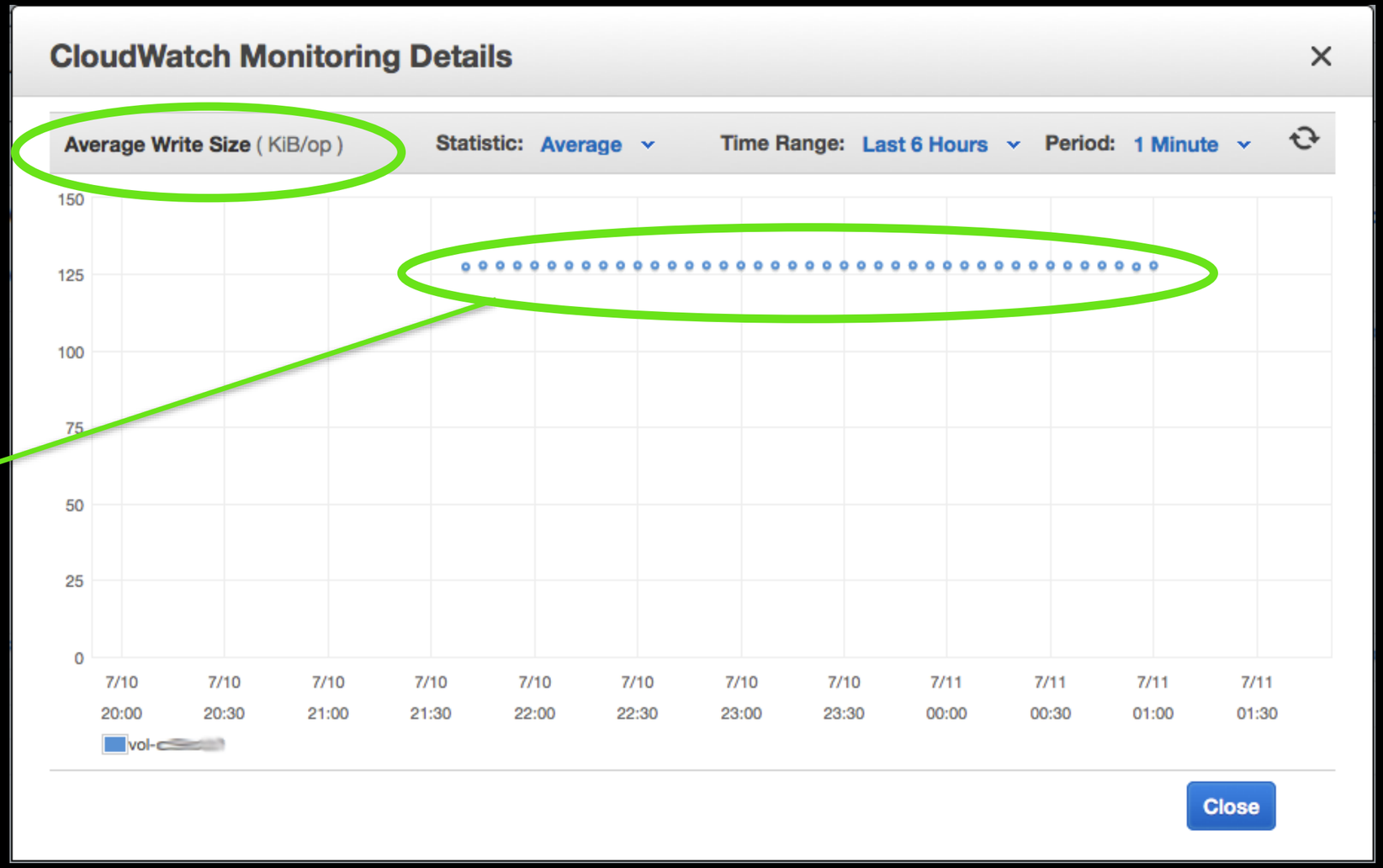


Verify st1 and sc1 workloads

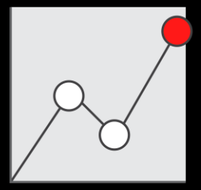


Amazon CloudWatch Console

128 KiB



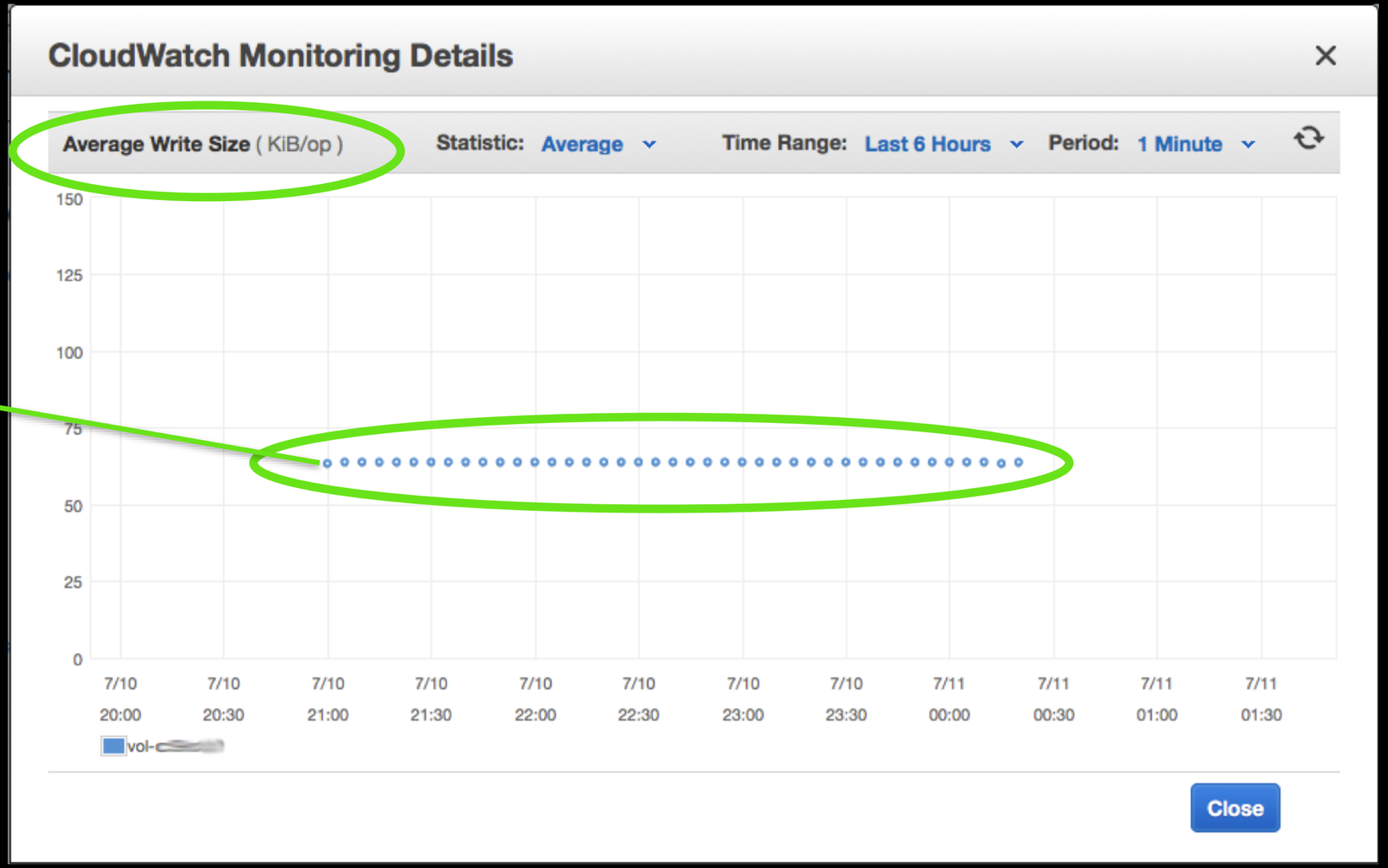
Verify st1 and sc1 workloads



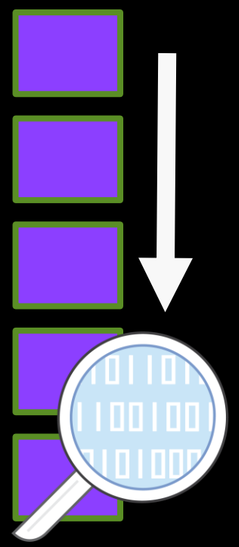
Amazon CloudWatch Console

Under 64 KiB?

Small or random IOPS likely



Linux performance tuning: st1 and sc1



Increase read-ahead buffer

- Recommended for high-throughput read workloads
- Per-volume configuration
- Default is 128 KiB (256 sectors) for Amazon Linux
- Smaller or random I/O will degrade performance

Increase read-ahead:

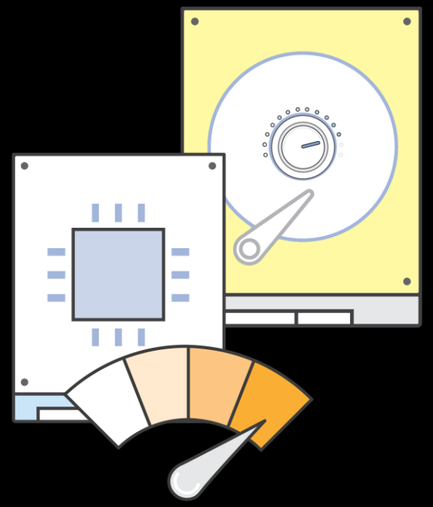
```
$ sudo blockdev --setra 2048 /dev/<device>
```

Examine current value:

```
$ sudo blockdev --getra 2048 /dev/<device>
```

https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/EBSPerformance.html

What is an EBS-optimized instance?



- Dedicated network bandwidth for Amazon EBS I/O
- Enabled by default on all current generation instance types
- Can be enabled at instance launch or on a running instance
- Not an option on some 10-Gbps instance types (c3.8xlarge, r3.8xlarge, i2.8xlarge)

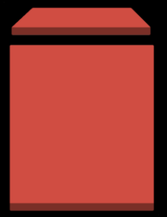
More details:
https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/EBSOptimized.html

Performance: EBS-optimized

c4.large

Dedicated to EBS

500 Mbps ~ 62.5 MiB/s
4,000 16K IOPS



2 TiB GP2 volume:
6,000 IOPS
250 MiB/s max throughput

c4.2xlarge

Dedicated to EBS

1 Gbps ~ 125 MiB/s
8,000 16K IOPS

2 TiB GP2 volume:
6,000 IOPS
250 MiB/s max throughput

Performance: EBS-optimized burst

c5.large

Dedicated to EBS

525 Mbps ~ 64 MiB/s

8,000 16K IOPS

2 TiB GP2 volume:
6,000 IOPS
250 MiB/s max throughput

30 minutes

c5.large

Dedicated to EBS

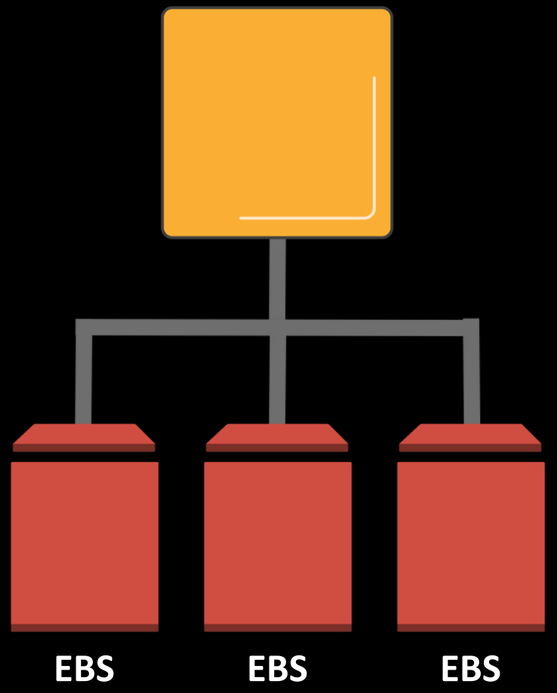
2,250 Mbps ~ 275 MiB/s

16,000 16K IOPS

2 TiB GP2 volume:
6,000 IOPS
250 MiB/s max throughput

aws SUMMIT

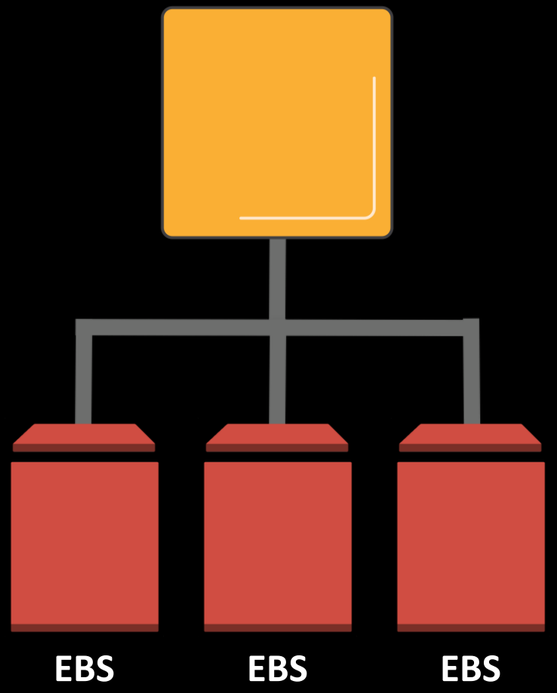
Best practice: RAID



When to use RAID?

- Storage requirement >16 TiB
- Throughput requirement > 1,000 MiB/s
- IOPS requirement > 64,000 @ 16K

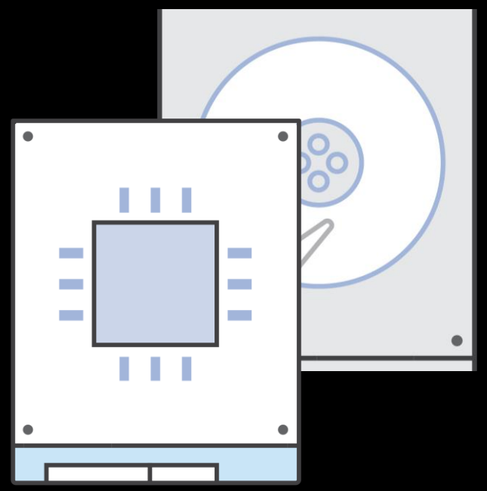
Best practice: RAID



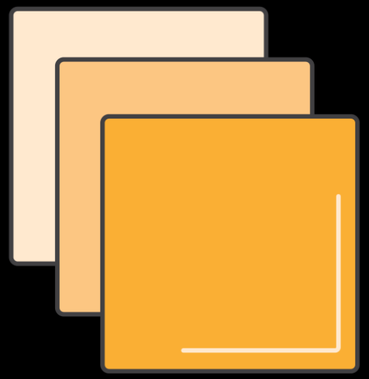
Avoid RAID for redundancy

- EBS data is already replicated
- RAID1 halves available EBS bandwidth
- RAID5/6 loses 20% to 30% of usable I/O to parity

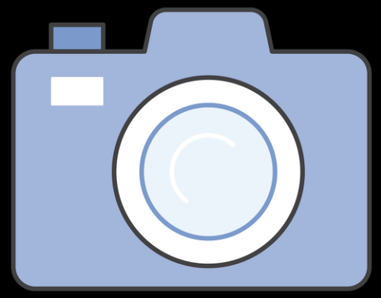
Summary



Select the right volume for your workload



Select the right instance for your workload



Take snapshots, tag snapshots

Use encryption

Thank you!

Ashish Palekar
@logicalblock

aws SUMMIT