AWS

SUMMIT

Deep Dive on Elastic Block Storage (Amazon EBS)

SRV413

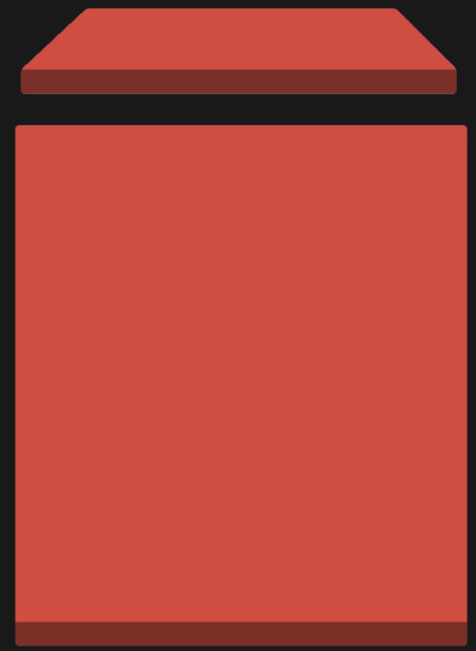
August 14, 2017

amazon web services

© 2017, Amazon Web Services, Inc. or its Affiliates. All rights reserved.

What to expect from this session

- Storage options on AWS
- EBS Overview
- Volume Types
- Deep Dive



The AWS Storage Portfolio

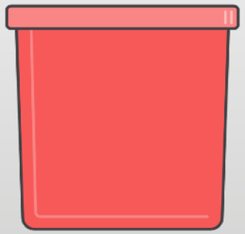
Amazon EBS
(persistent)

Amazon EC2
Instance Store
(ephemeral)

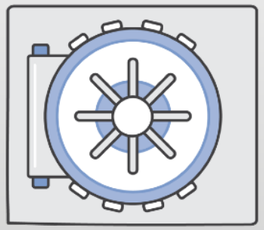
Block

Amazon EFS

File



Amazon S3



Amazon Glacier

Object

Cloud Data Migration

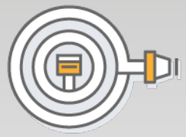
Snow* data
transport family

Storage
Gateway

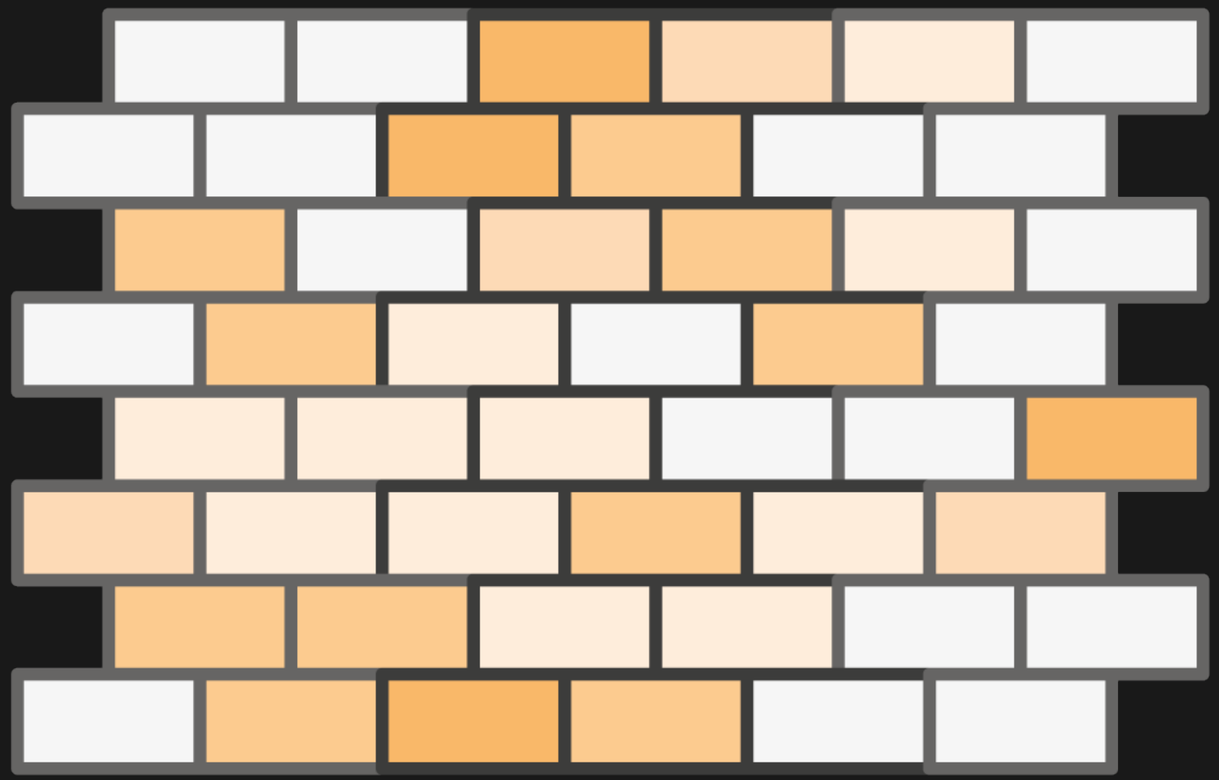
Direct Connect

3rd Party
Connectors

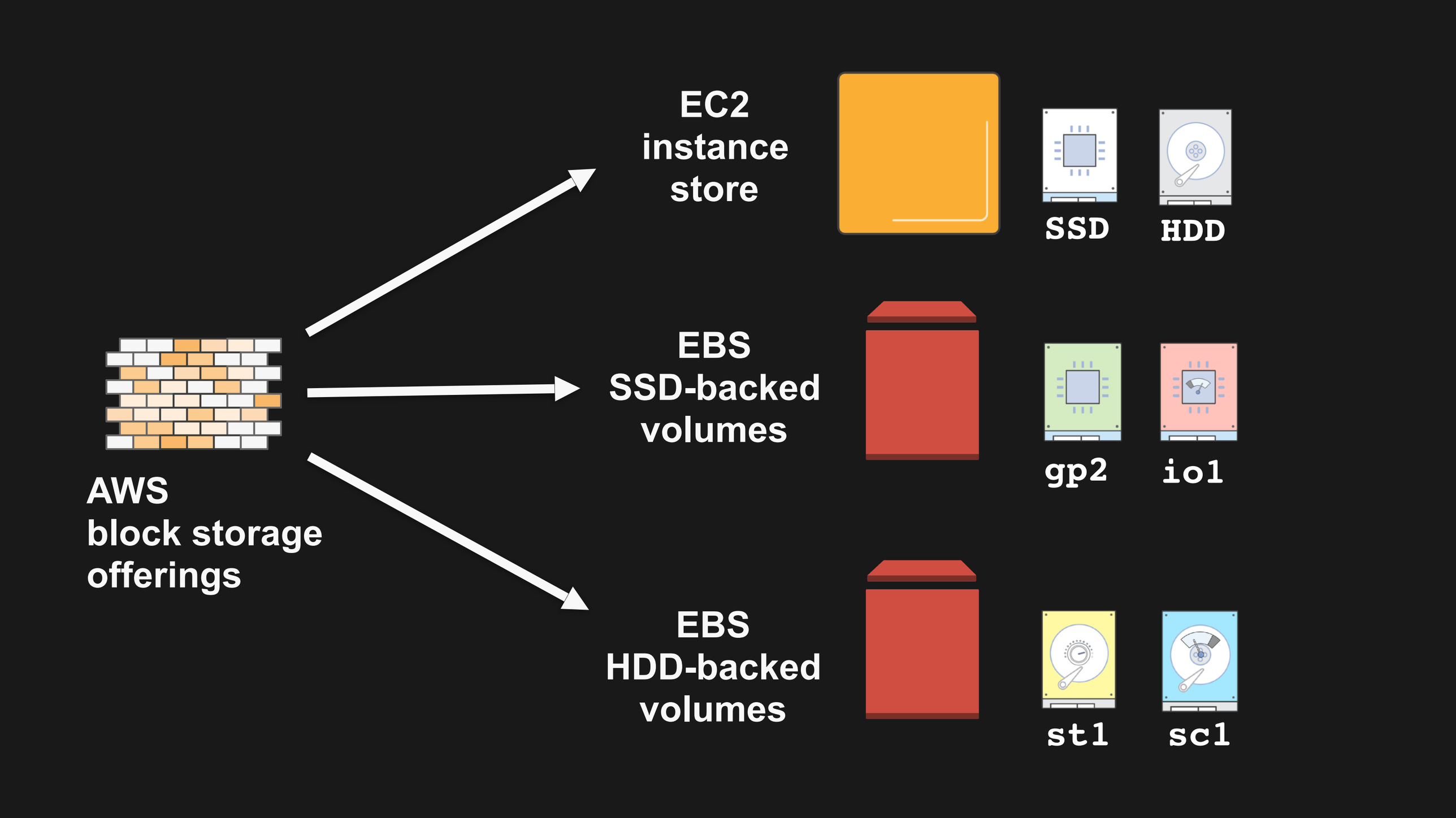
Transfer
Acceleration



Kinesis Firehose



AWS block storage offerings


EC2
instance
store
SSD
HDD
EBS
SSD-backed
volumes
gp2
io1
AWS
block storage
offerings
EBS
HDD-backed
volumes
st1
sc1

What is Amazon EC2 instance store?

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

What is Amazon Elastic Block Store (EBS)?

Elastic Block Store

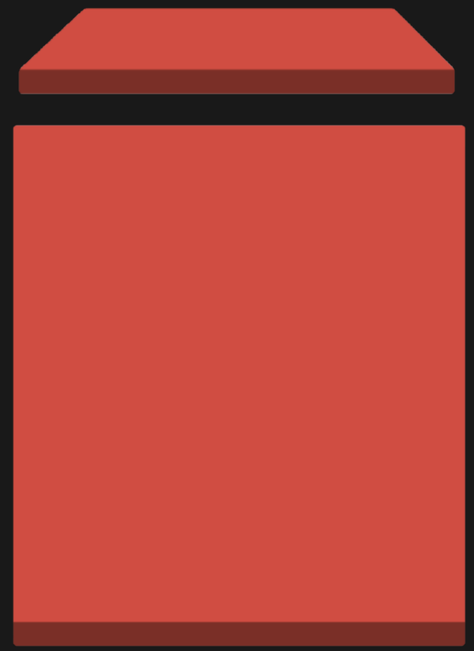
- Persistent block storage volumes
- 99.999% availability
- Automatically replicated within its Availability Zone (AZ)
- Point-in-time snapshot support
- Modify volume type as needs change
- SSD or HDD

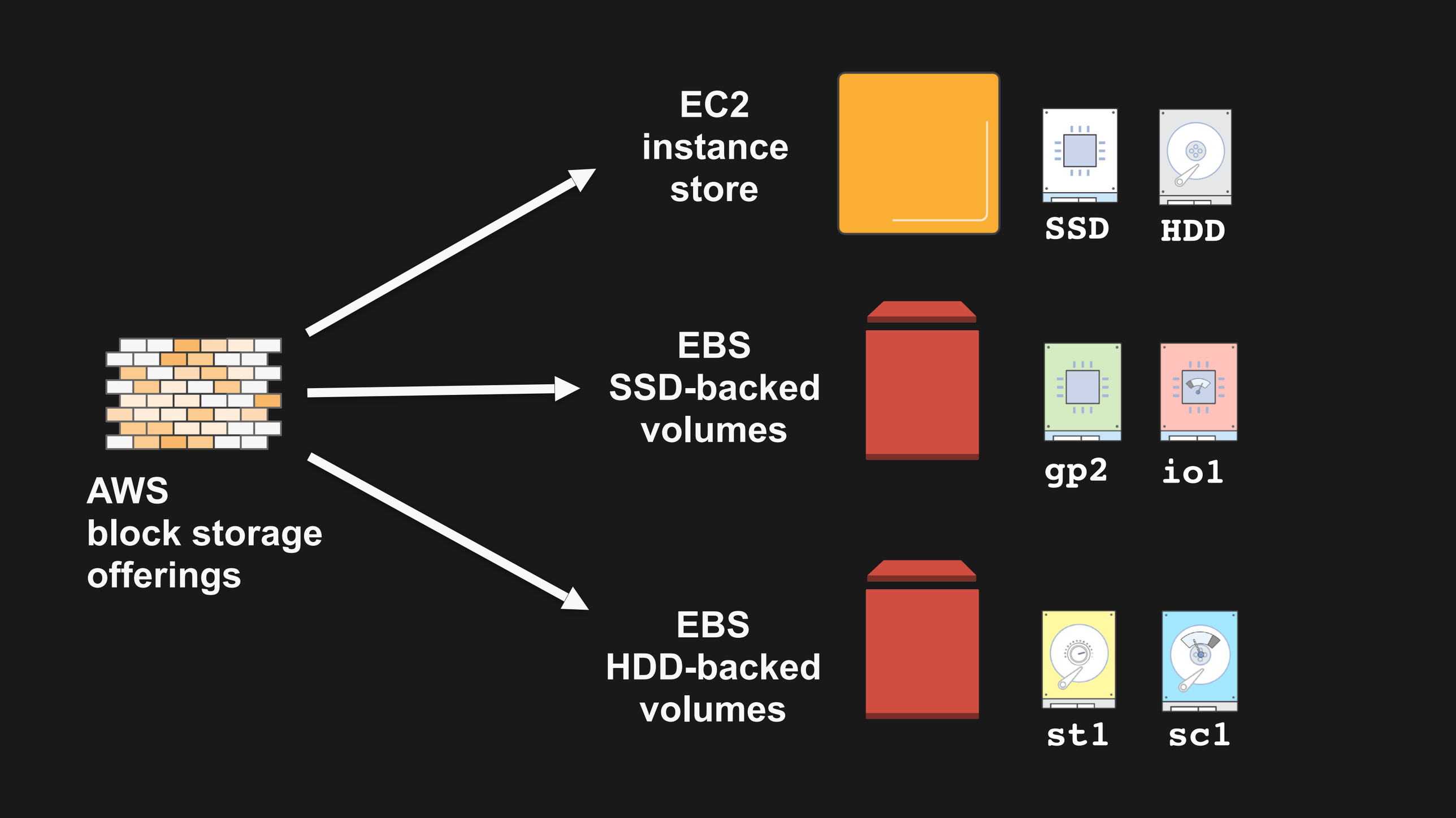
EC2 Instance Store

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

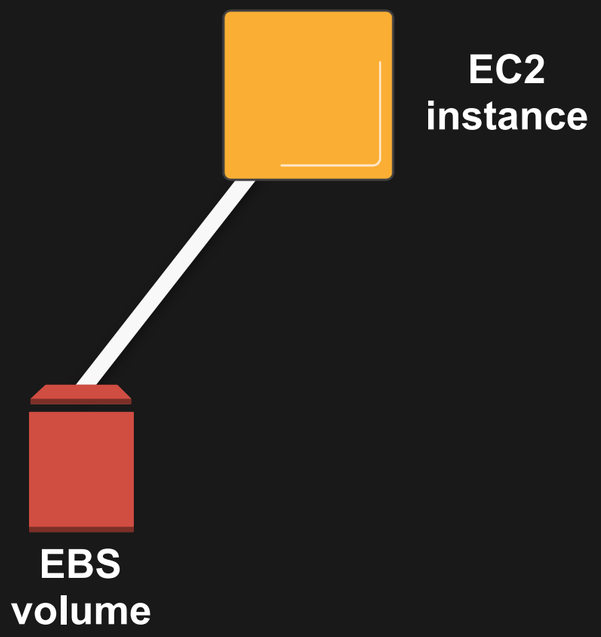
The history of EBS

- 2006 – EC2 launched with instance storage
- 2008 – EBS (Elastic Block Storage) launched on magnetic storage
- 2012 – EBS io1 (SSD) & EBS-Optimized instances
- 2014 – EBS gp2 (SSD)
- 2014 – EBS data volume encryption
- 2015 – Larger/faster EBS volumes
- 2015 – EBS boot volume encryption
- 2016 – EBS st1 (HDD) and sc1 (HDD)
- 2017 – EBS Elastic Volumes!



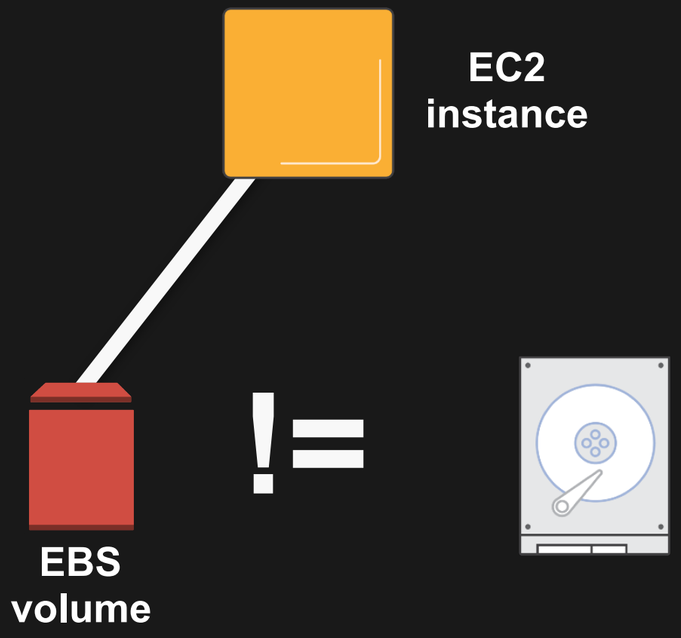

EC2
instance
store
SSD
HDD
EBS
SSD-backed
volumes
gp2
io1
AWS
block storage
offerings
EBS
HDD-backed
volumes
st1
sc1

What is EBS?

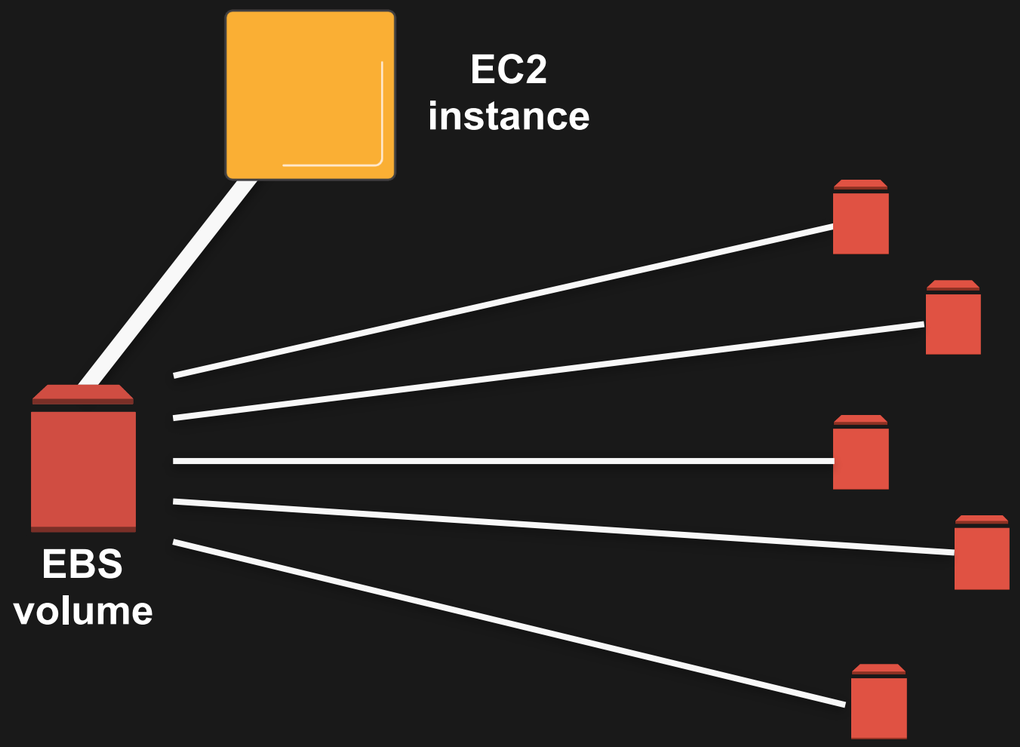


- Block storage as a service
- Create, attach volumes through an API
- Service accessed over the network

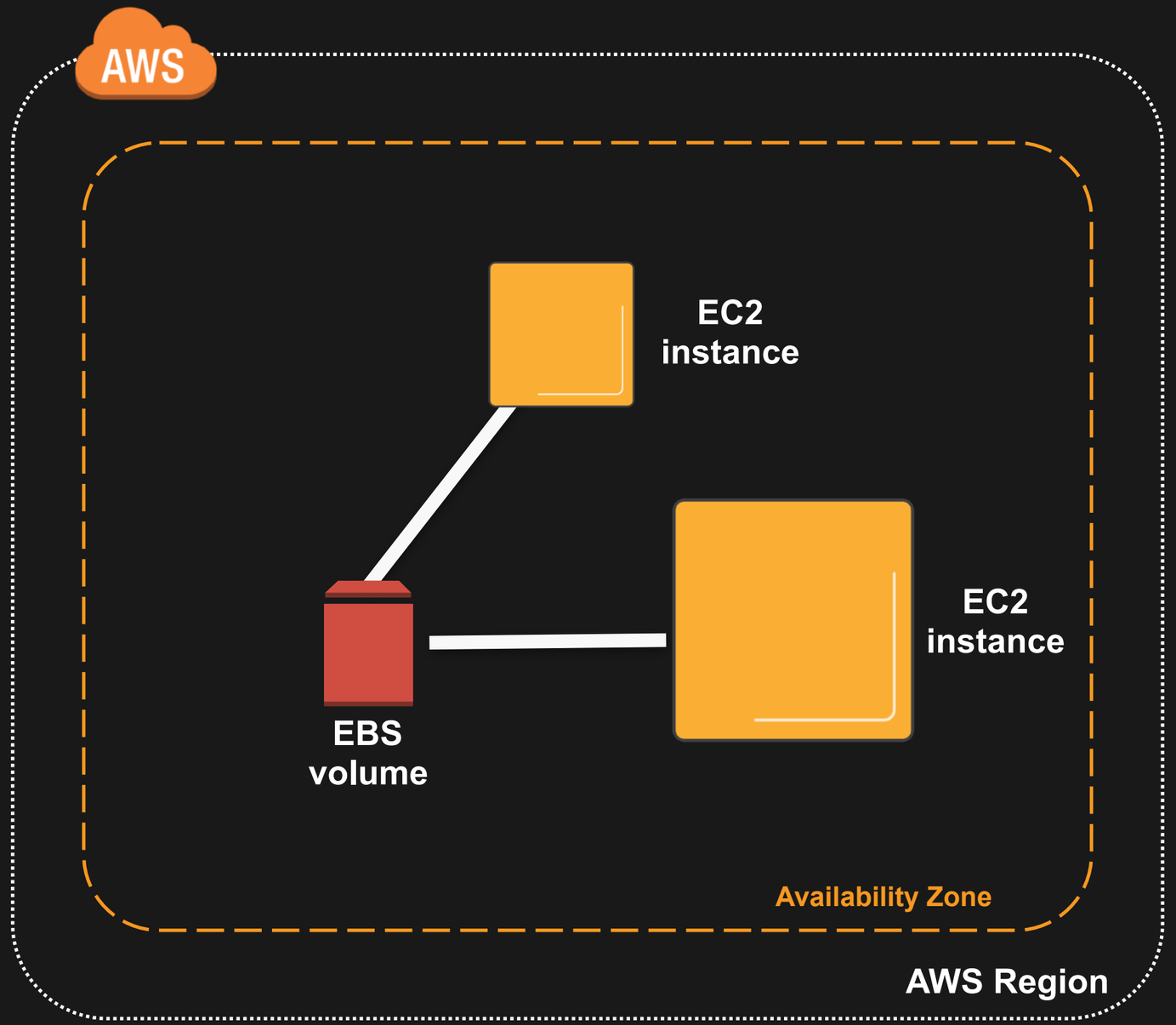
What is EBS?



What is EBS?

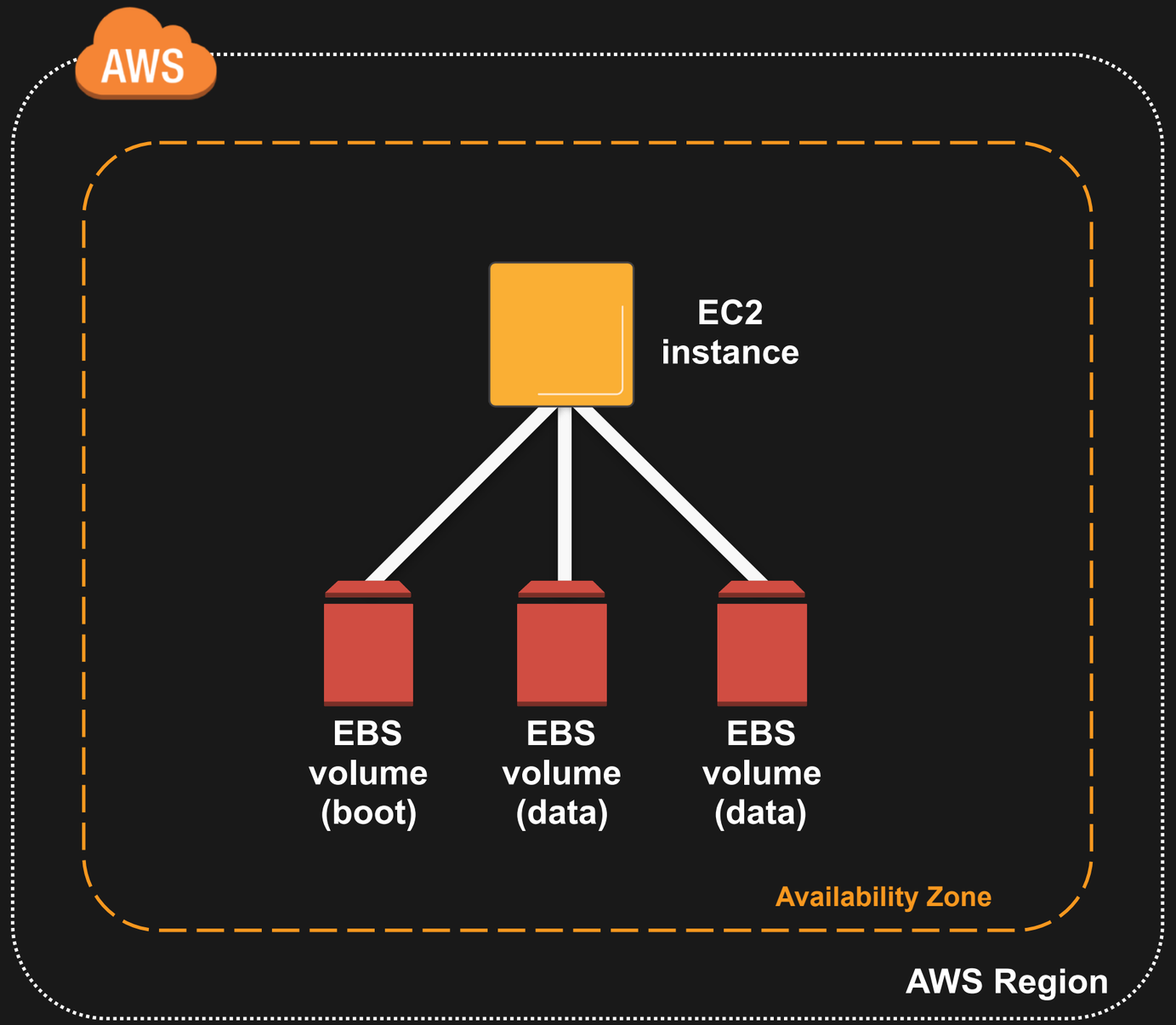


What is EBS?



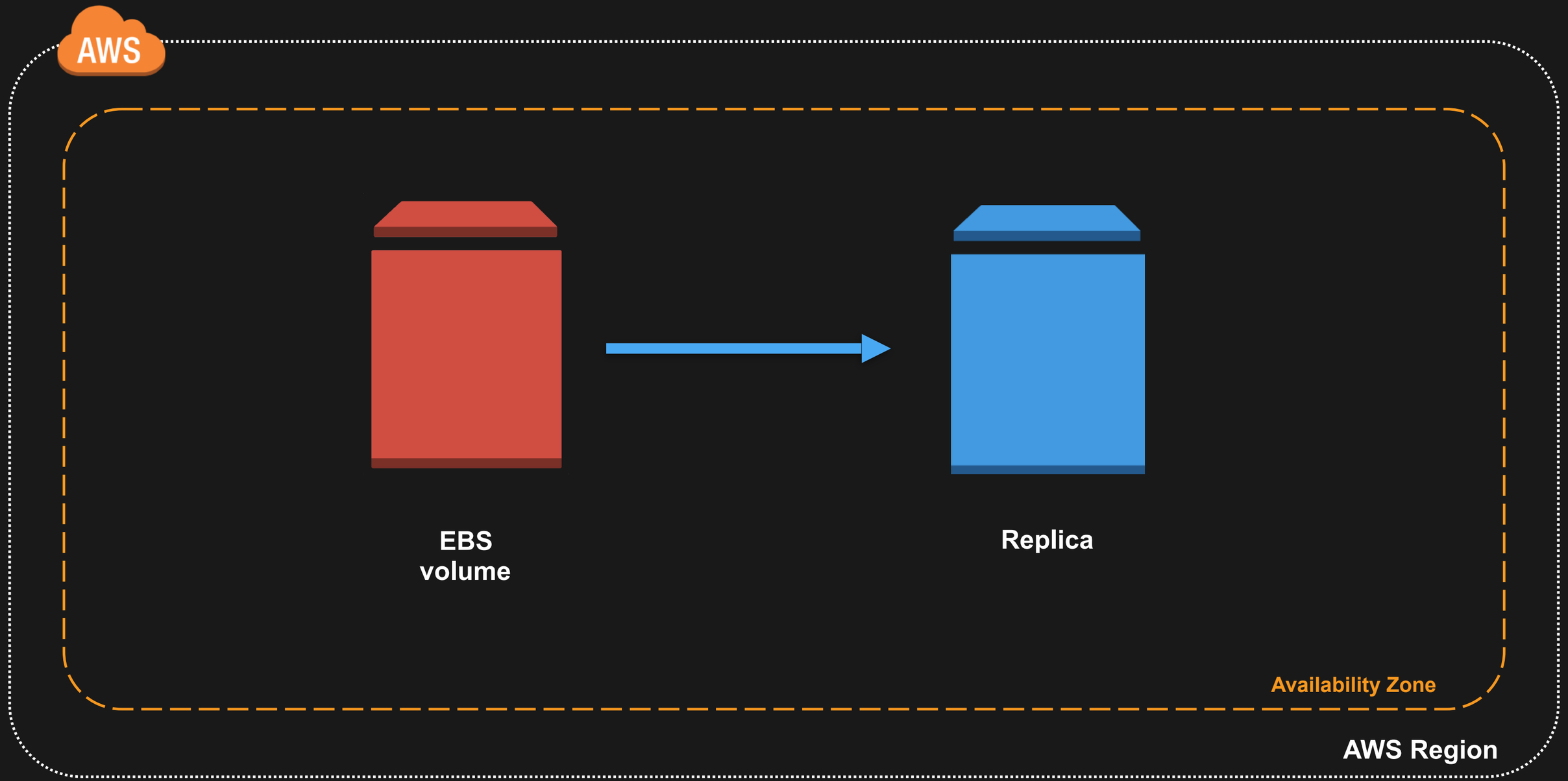
- Volumes persist independent of EC2
- Select storage and compute based on your workload
- Detach and attach between instances within the same AZ

What is EBS?



- Volumes attach to one instance at a time
- Many volumes can attach to an instance
- Separate boot volume from data volumes

What is EBS?



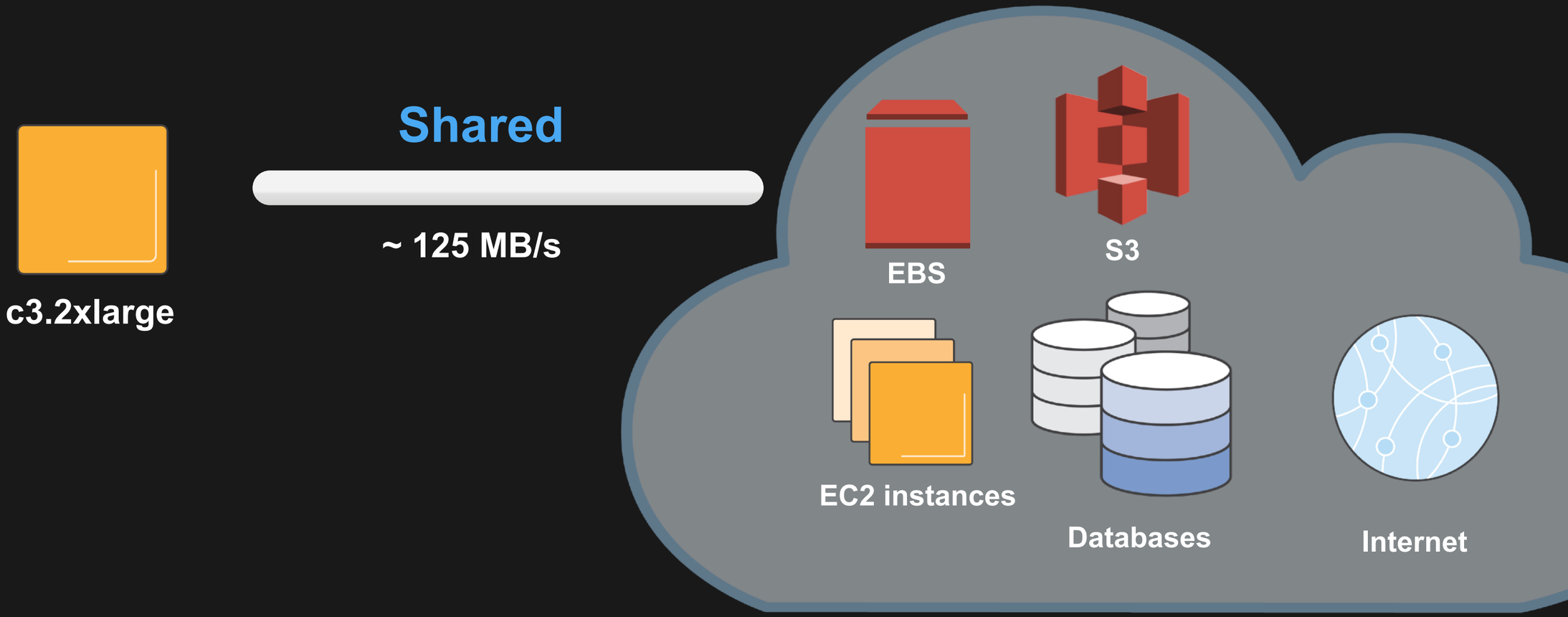
What is EBS?

EBS is designed for:

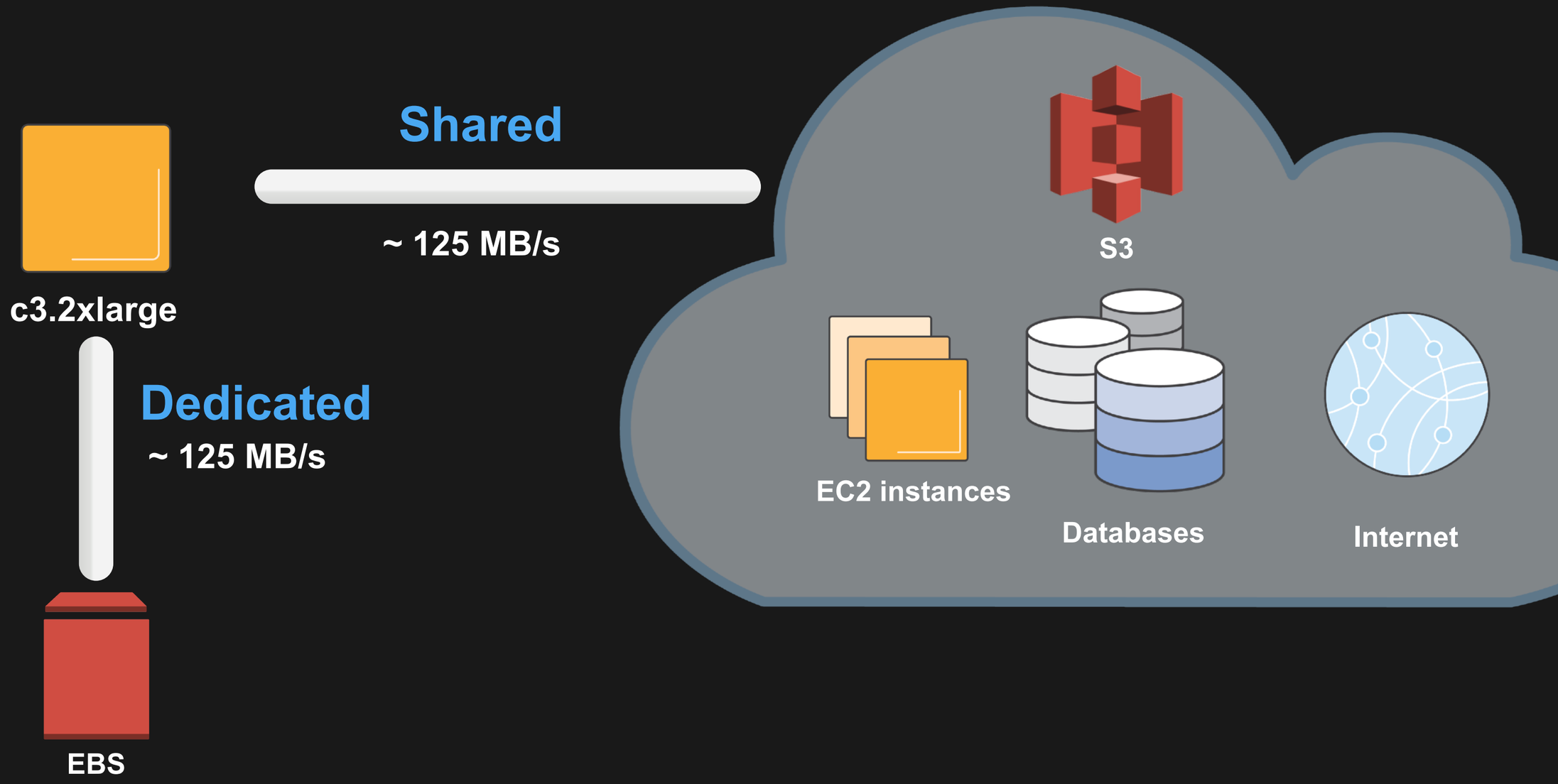
99.999% service availability

0.1% to 0.2% annual failure rate (AFR)

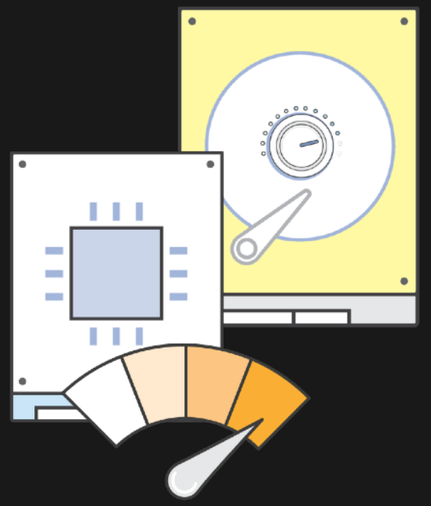
What is an EBS-optimized instance?



What is an EBS-optimized instance?



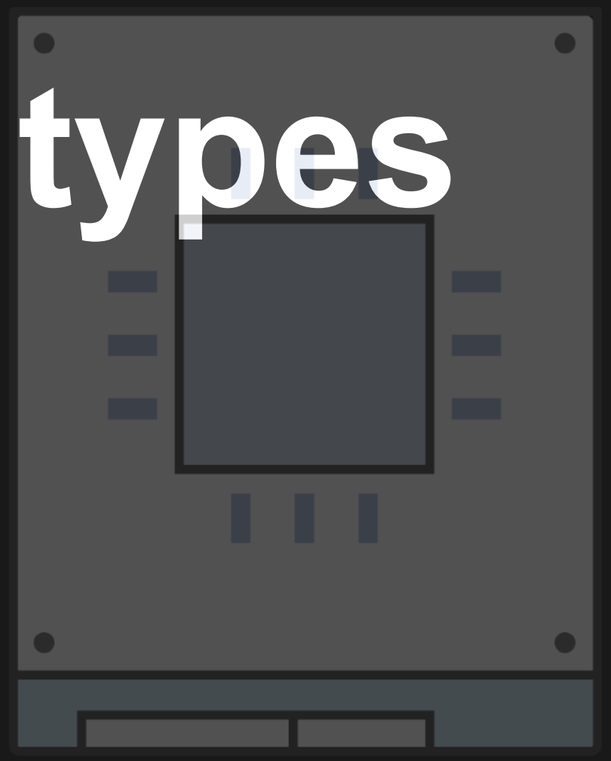
What is an EBS-optimized instance?



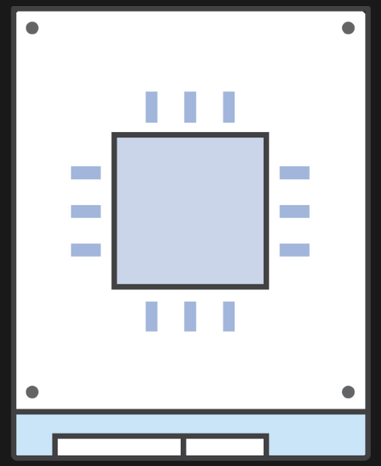
- Dedicated network bandwidth for EBS I/O
- Enabled by default on c4, d2, m4, r4, p2, and x1 instances
- Can be enabled at instance launch or on a running instance
- Not an option on some 10 Gbps instance types
  - (c3.8xlarge, r3.8xlarge, i2.8xlarge)

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

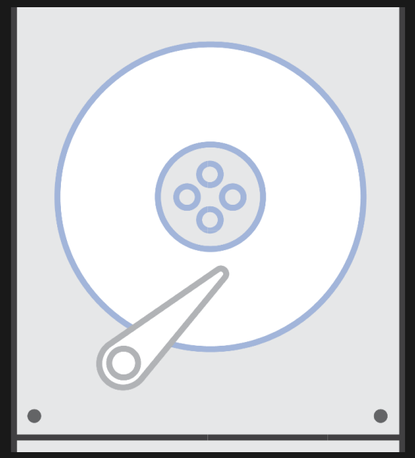
EBS volume types



EBS volume types

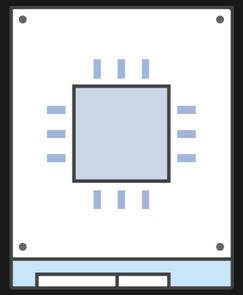


Solid-State Drives (SSD)

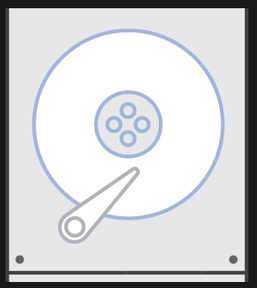


Hard Disk Drives (HDD)

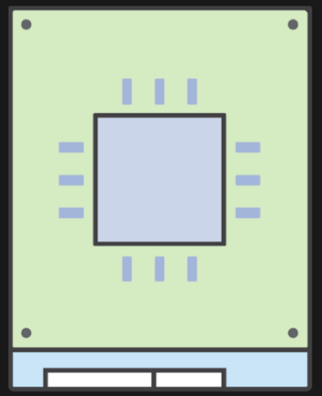
EBS volume types



SSD

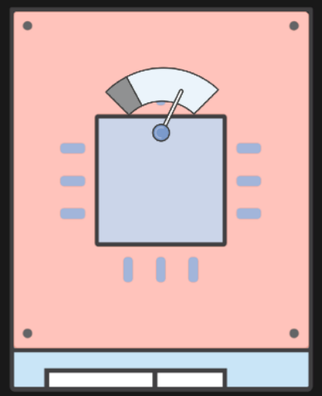


HDD



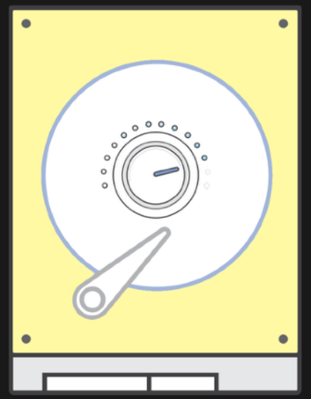
gp2

General Purpose SSD



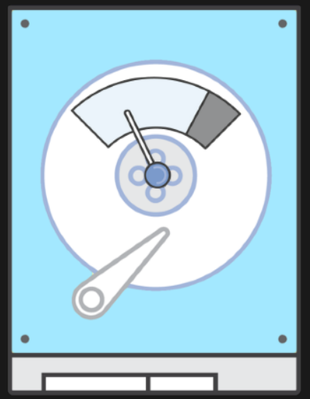
io1

Provisioned IOPS SSD



st1

Throughput Optimized HDD

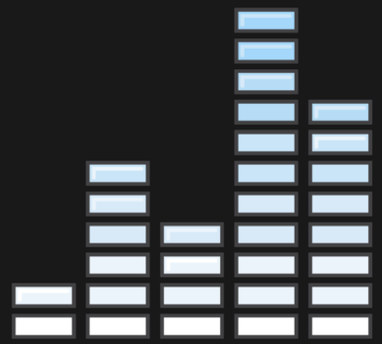


sc1

Cold HDD

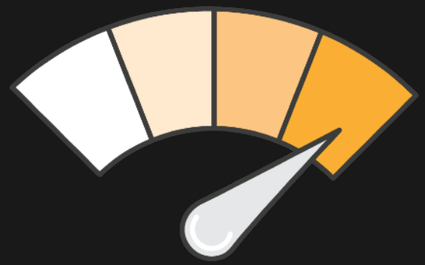
Choosing an EBS volume type

What is more important to your workload?



IOPS

or

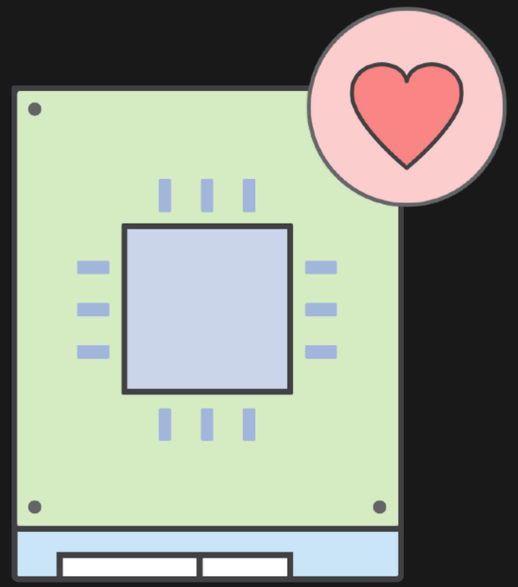


Throughput?

Choosing an EBS volume type

Don’t know your workload yet?

EBS volume types: General Purpose SSD (gp2)



gp2

General Purpose SSD

Baseline: 100 IOPS + 3 IOPS per GiB up to 10,000 IOPS

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

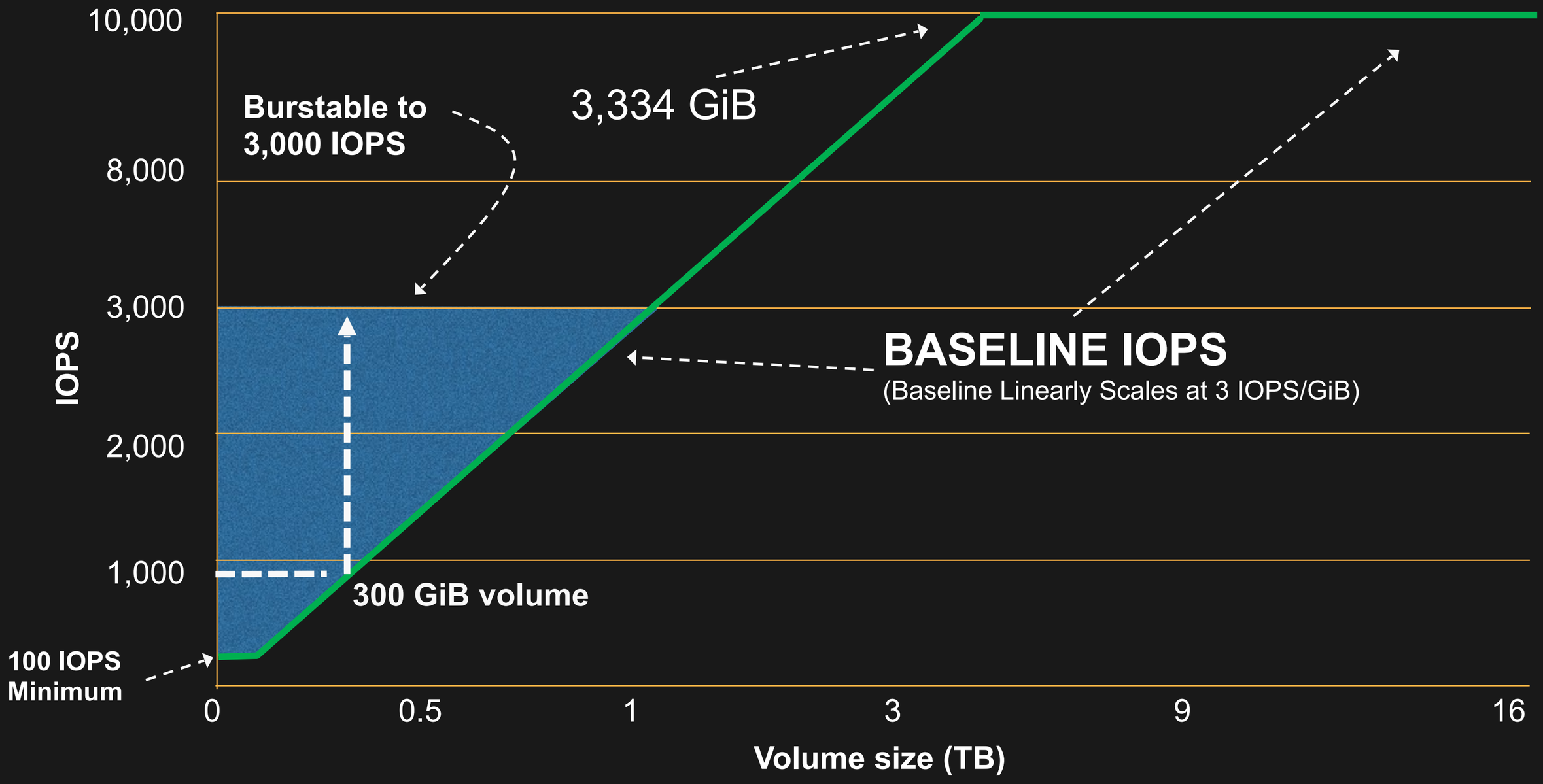
Throughput: 160MiB/s

Latency: Single-digit ms

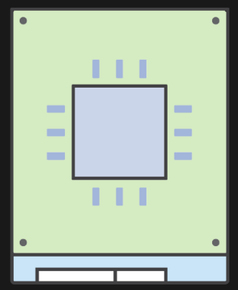
Capacity: 1 GiB to 16 TiB

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

Burst and baseline: General Purpose SSD (gp2)

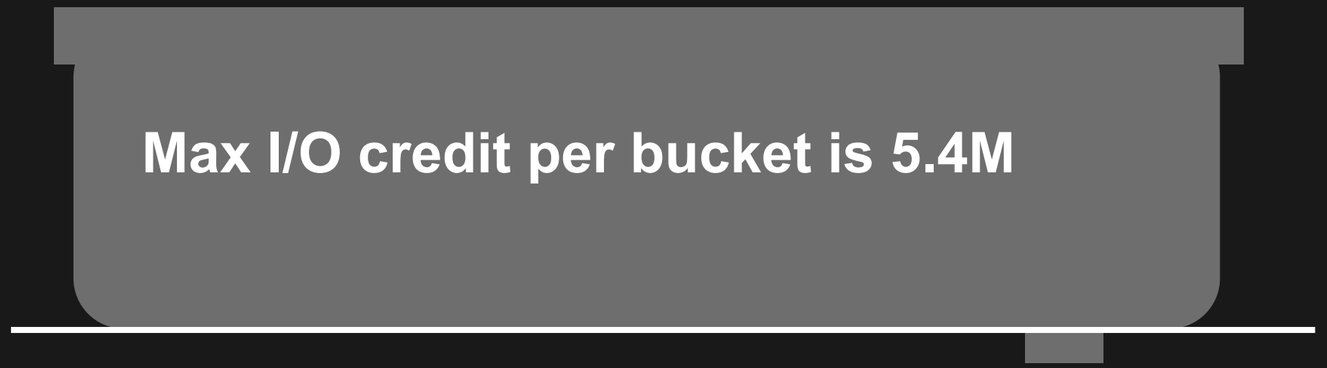


Burst bucket: General Purpose SSD (gp2)



gp2

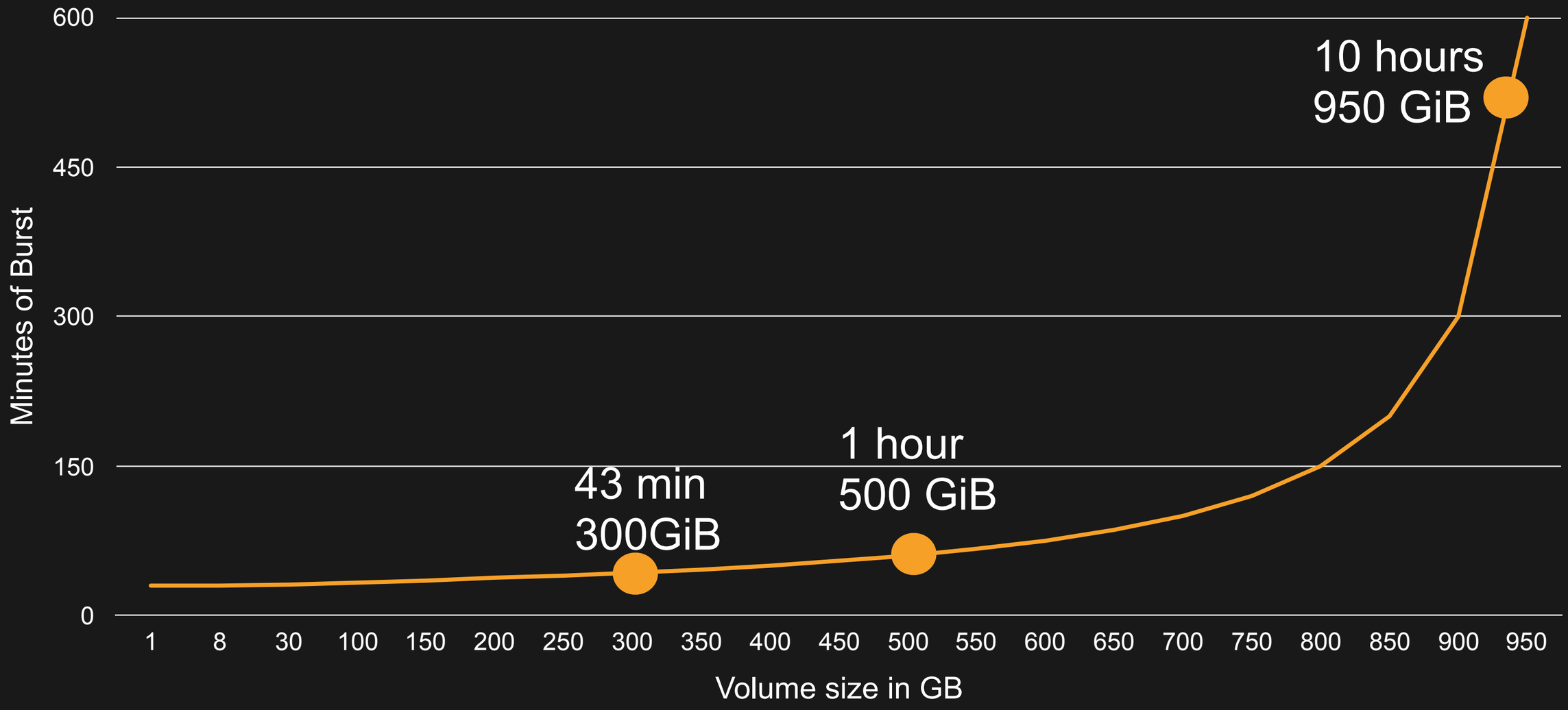
Always accumulating
3 IOPS per GiB per second



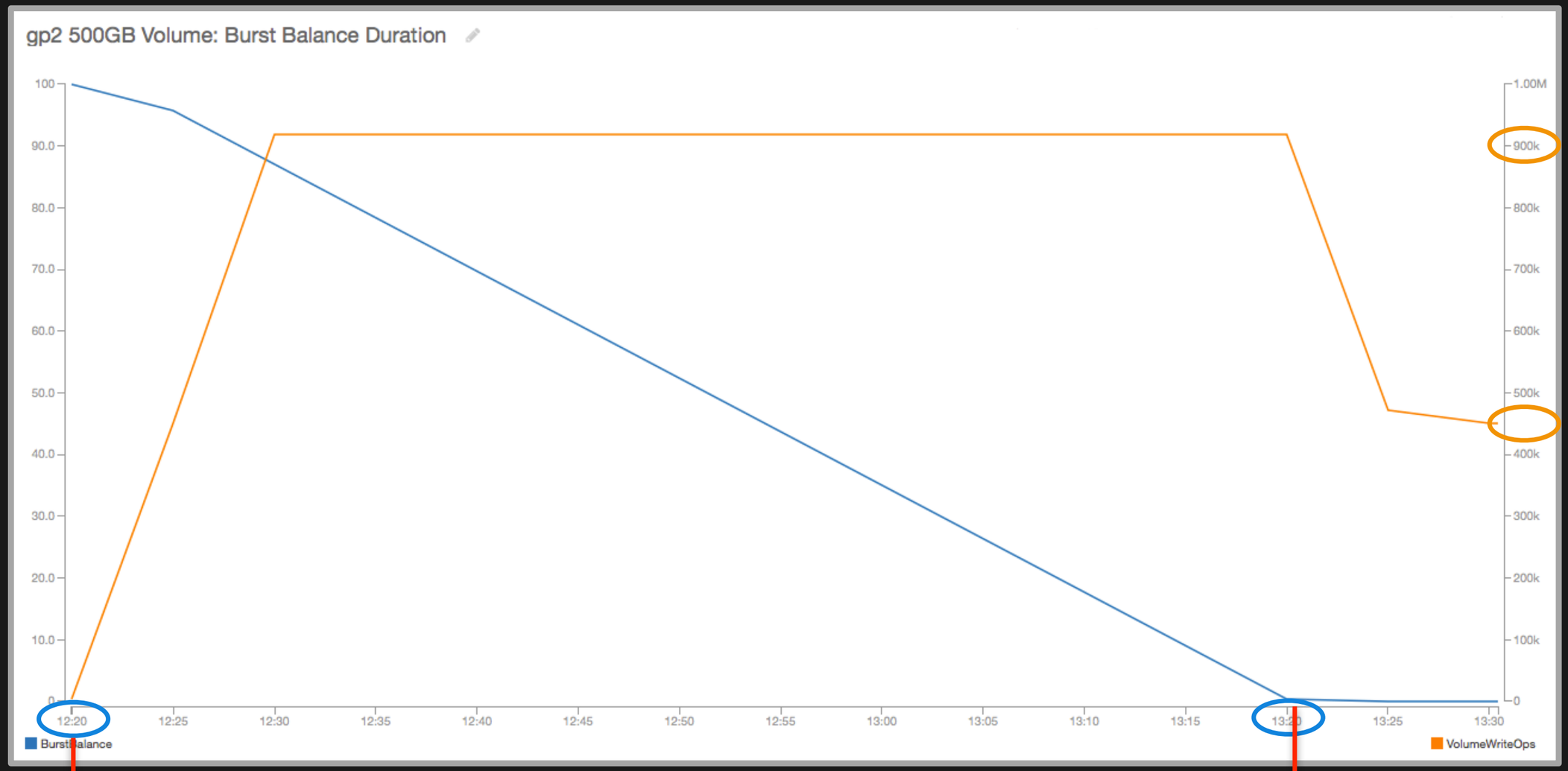
You can spend up to
3000 IOPS per second

Baseline performance = 3 IOPS per GiB or 100 IOPS

How long can I burst on gp2?



How do I monitor gp2 burst balance?



900,000 write IOs over 5 min = 3000 IOPS

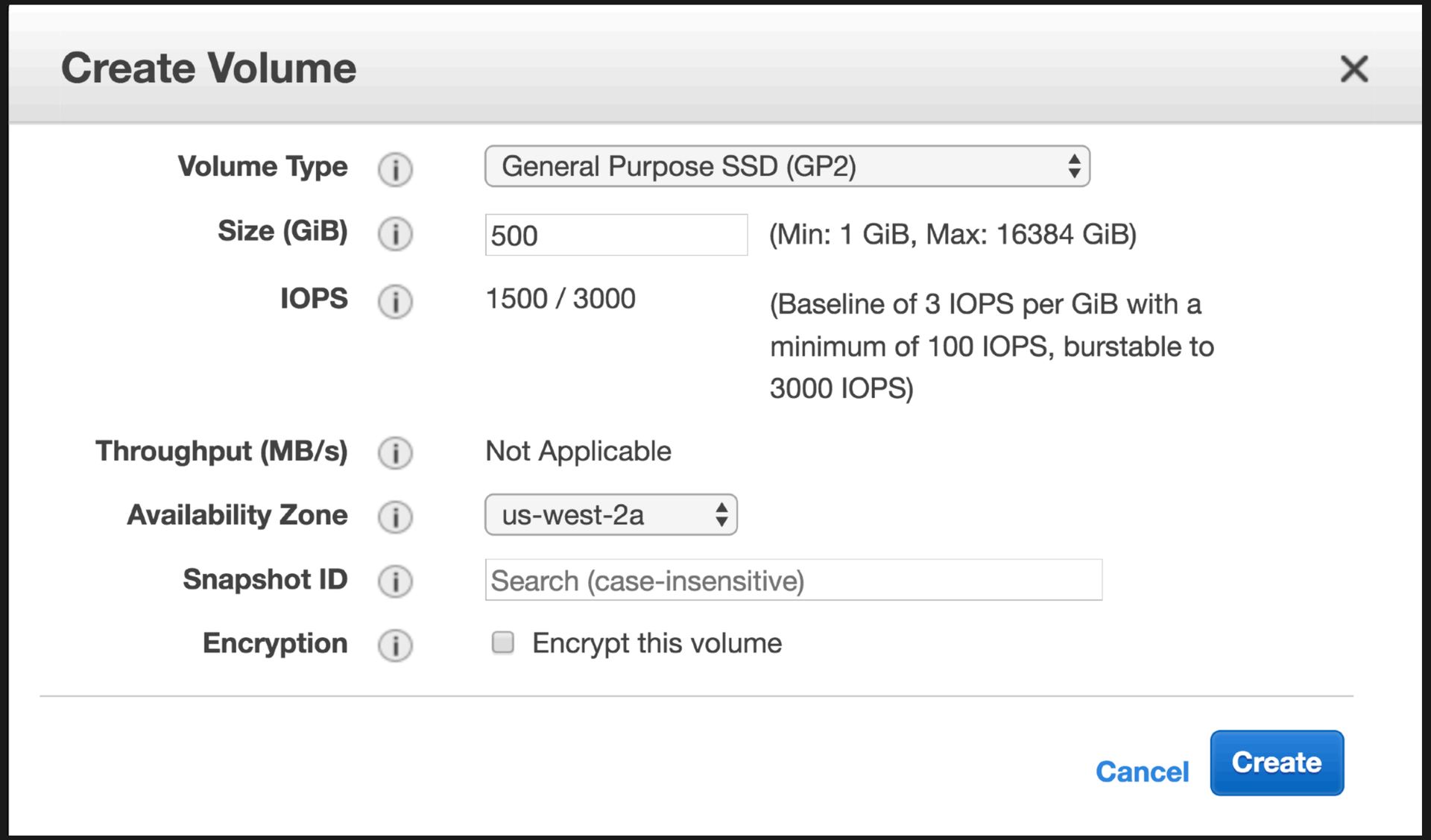
450,000 write IOs over 5 min = 1500 IOPS

BurstBalance

VolumeWriteOps

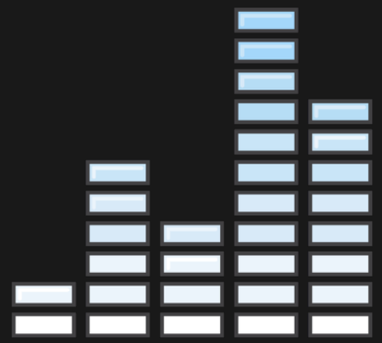
One Hour

How do I monitor gp2 burst balance?



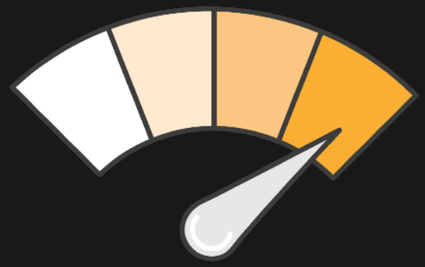
Choosing an EBS volume type

What is more important to your workload?



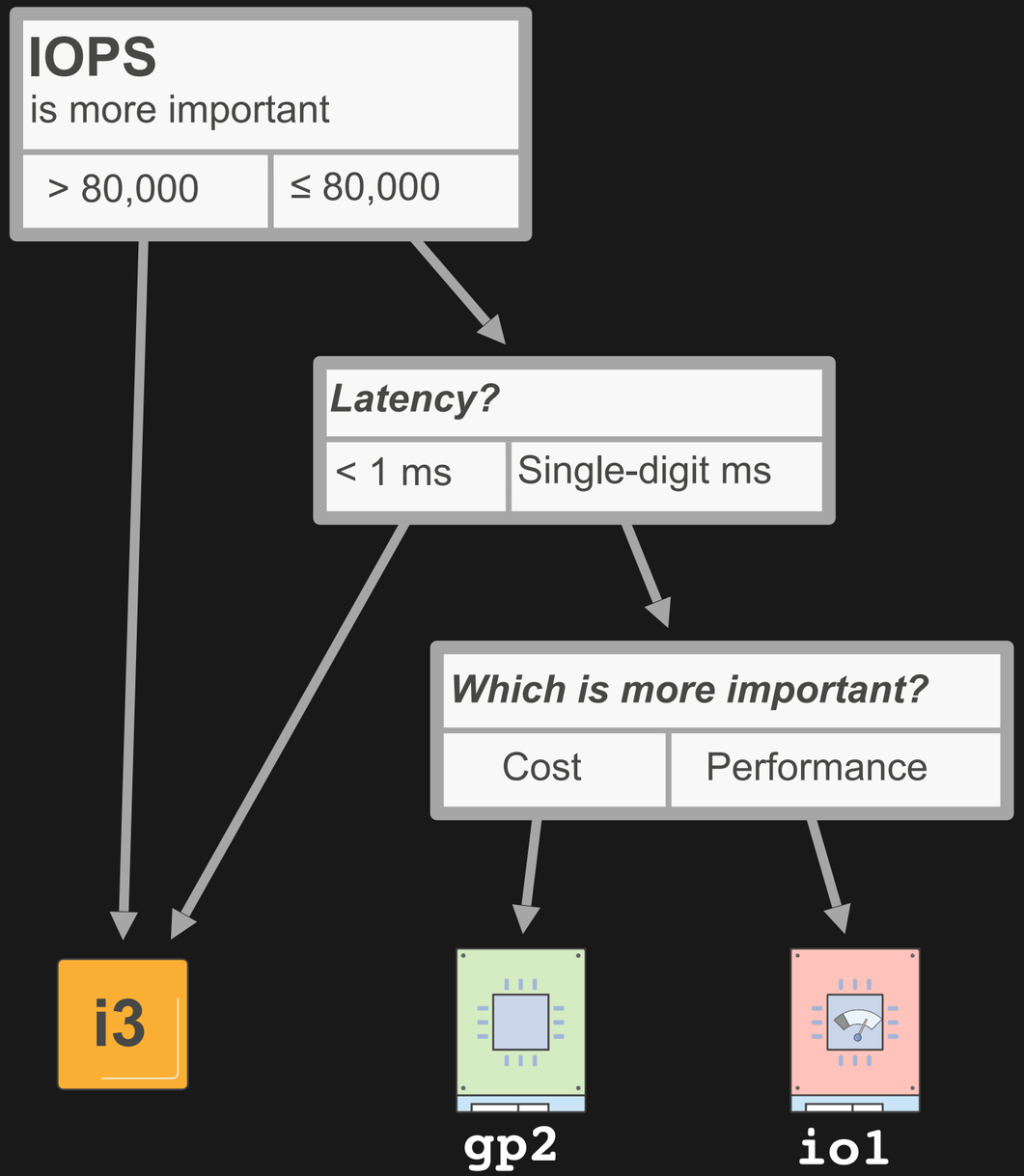
IOPS

or

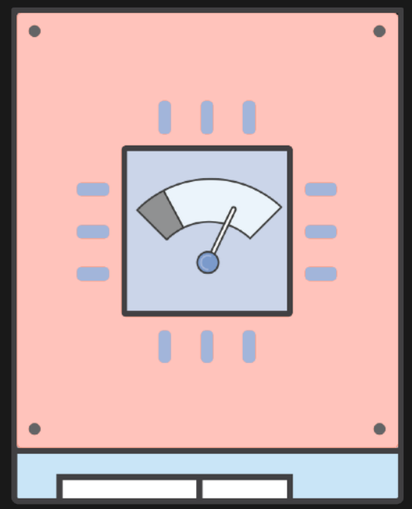


Throughput?

Choosing an EBS volume type



EBS volume types: I/O Provisioned



`io1`

Provisioned IOPS SSD

Baseline: 100 to 20,000 IOPS

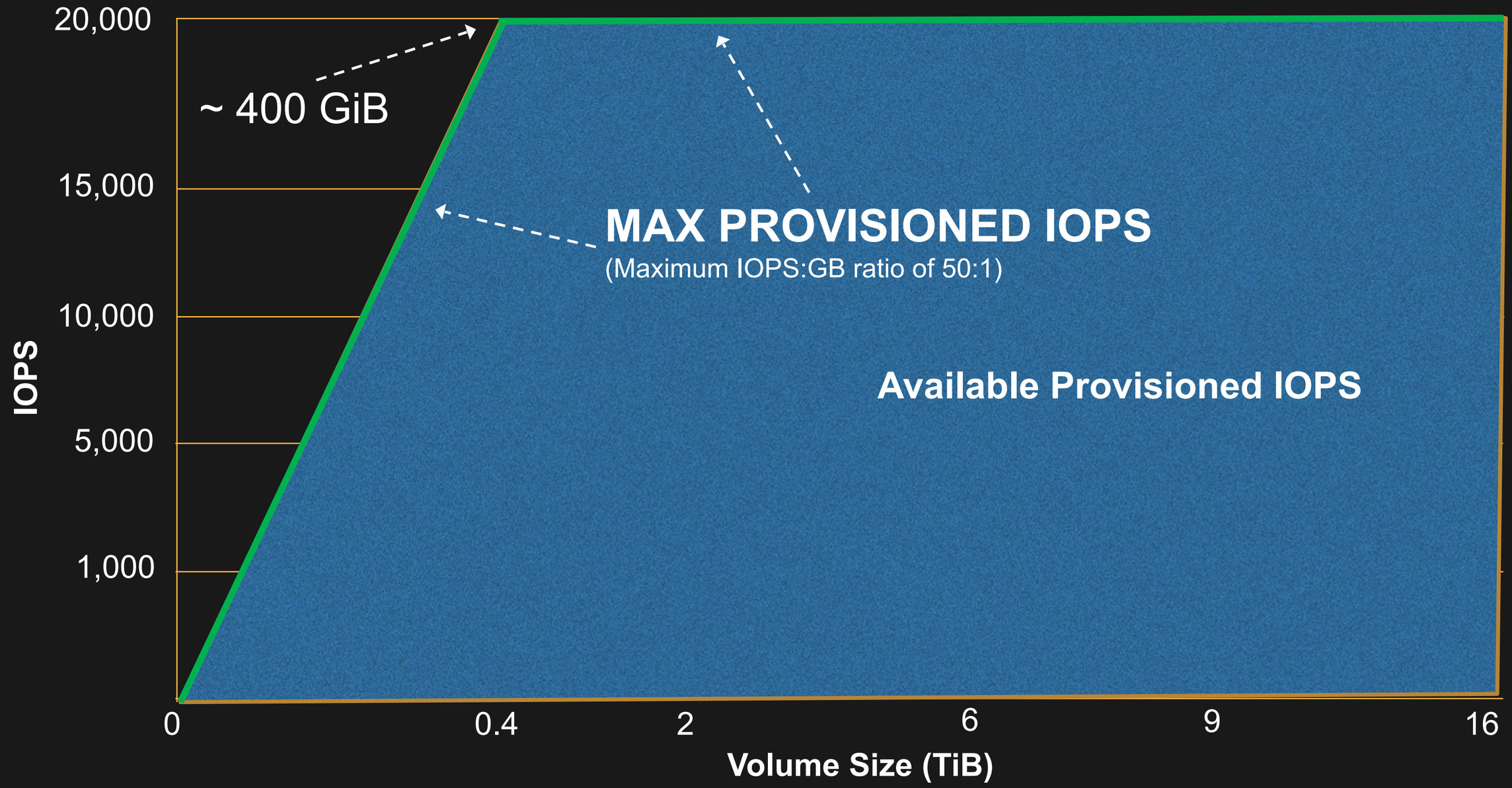
Throughput: 320 MiB/s

Latency: Single-digit ms

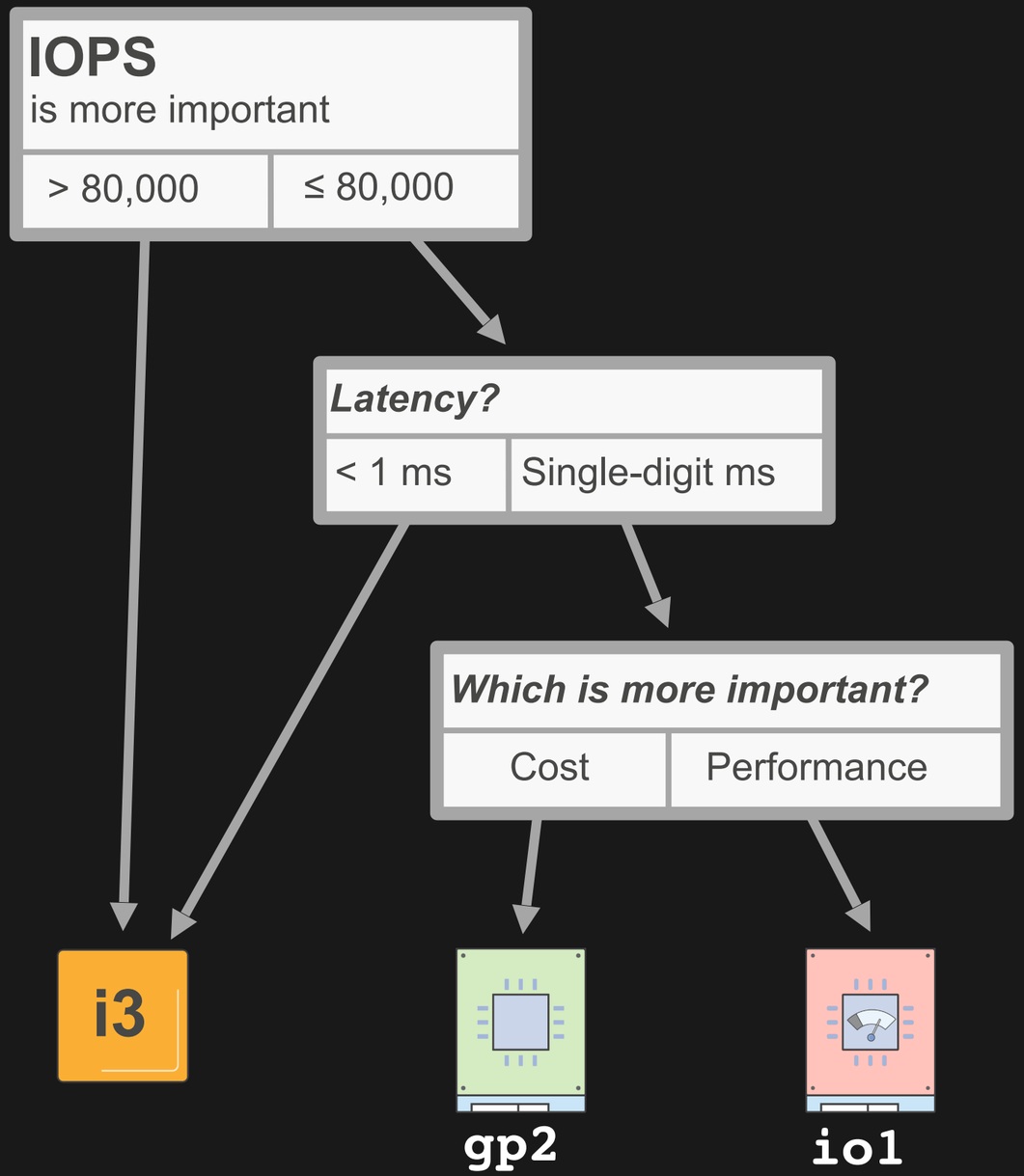
Capacity: 4 GiB to 16 TiB

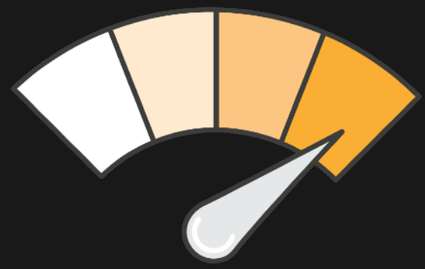
Ideal for critical applications and databases with sustained IOPS

Scaling Provisioned IOPS SSD (io1)



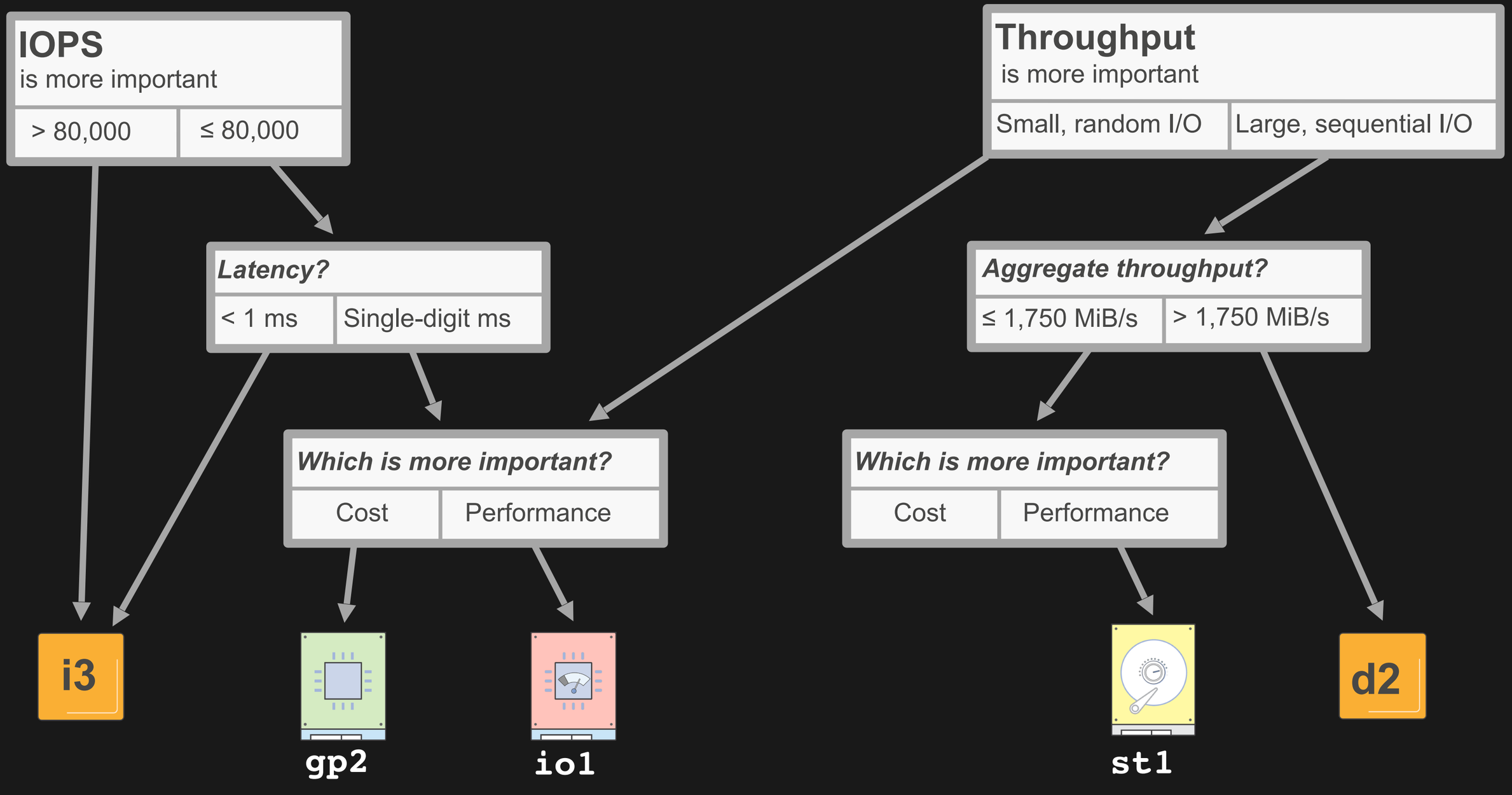
Choosing an EBS volume type



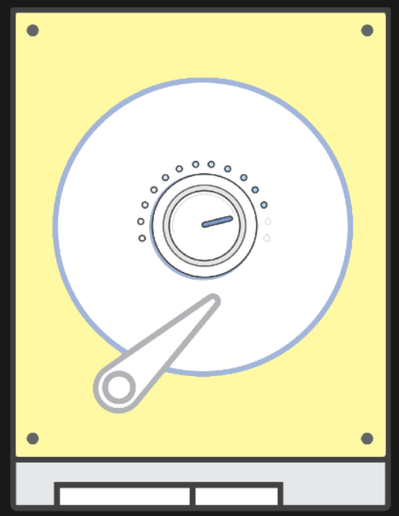


Throughput?

Choosing an EBS volume type



EBS volume types: Throughput Provisioned



`st1`

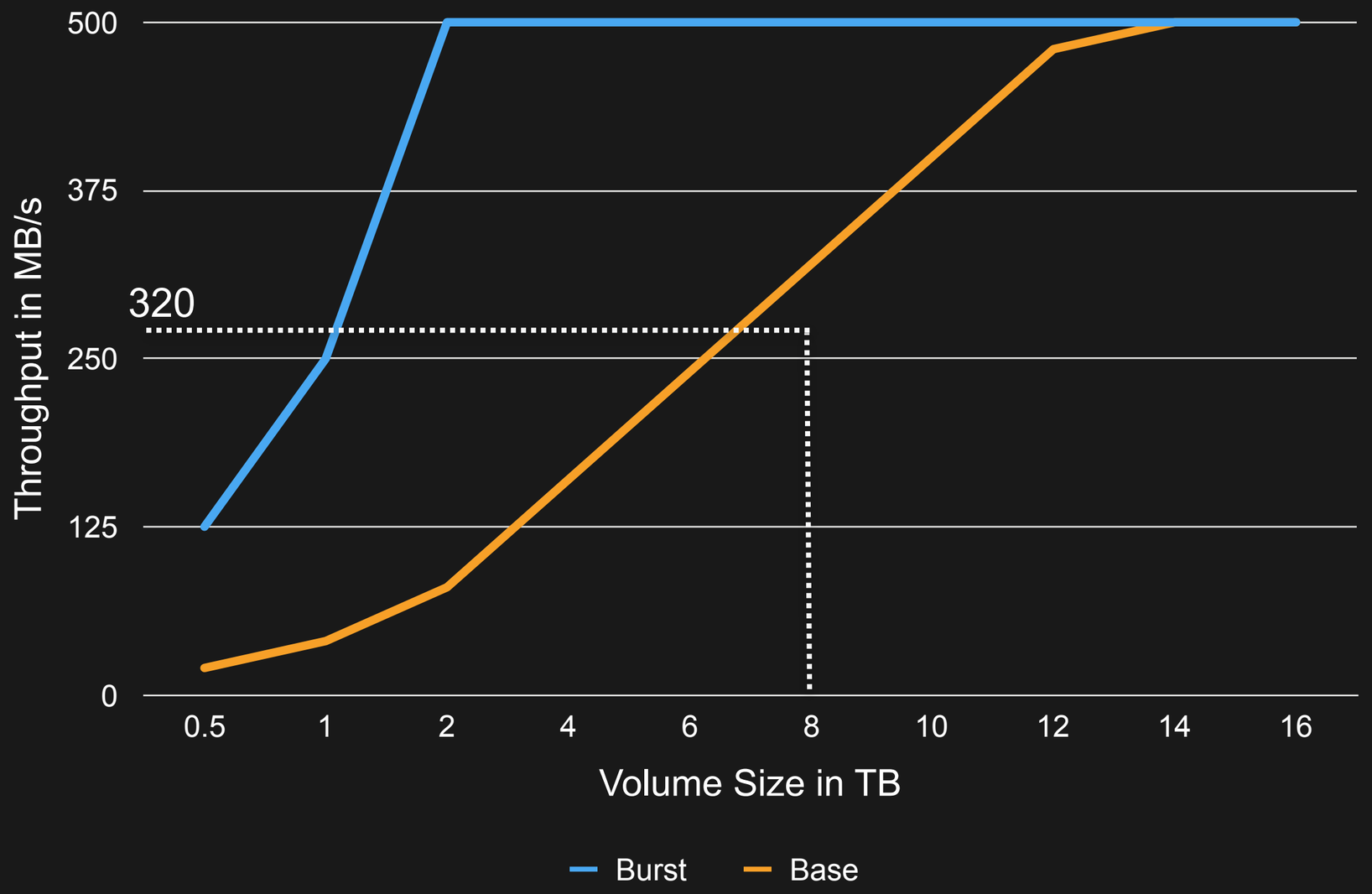
Baseline: 40 MiB/s per TiB up to 500 MiB/s

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

Capacity: 500 GiB to 16 TiB

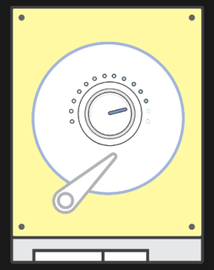
Ideal for large-block, high-throughput sequential workloads

Throughput Optimized HDD – burst and base



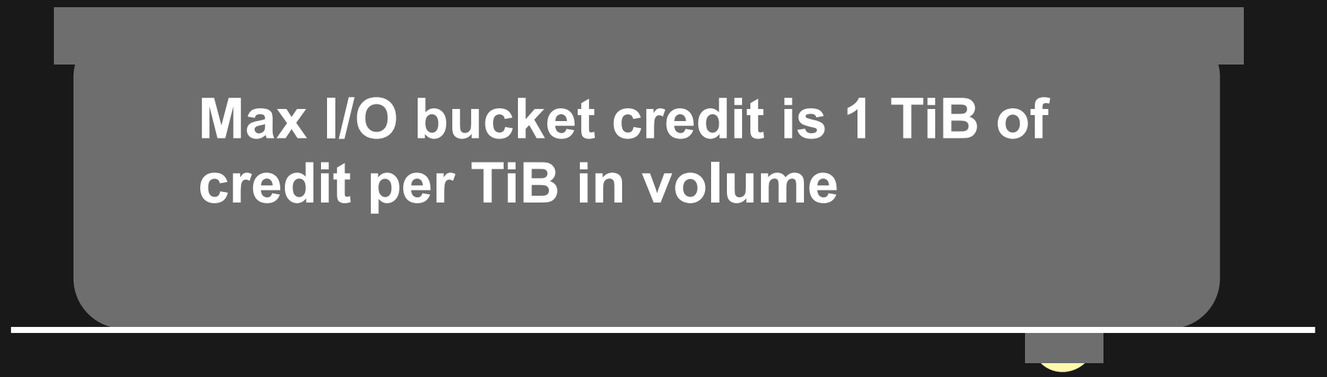
ST1

Burst bucket: Throughput Optimized HDD (st1)



st1

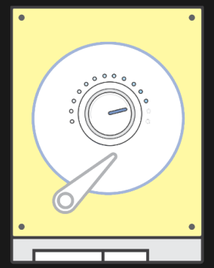
Always accumulating 40 MiB/s per TiB



You can spend up to 250 MiB/s per TiB

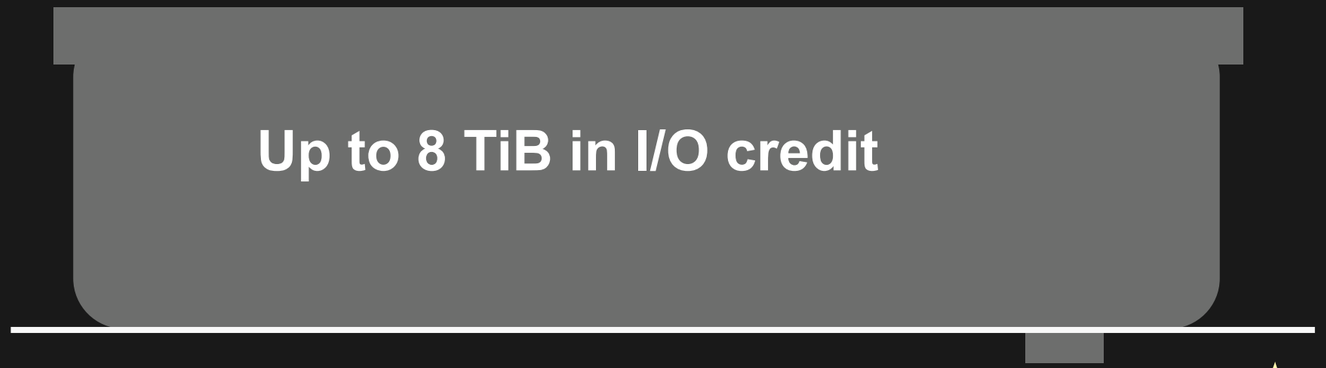
Baseline performance = 40 MiB/s per TiB

Burst bucket: example 8 TiB st1 volume



st1

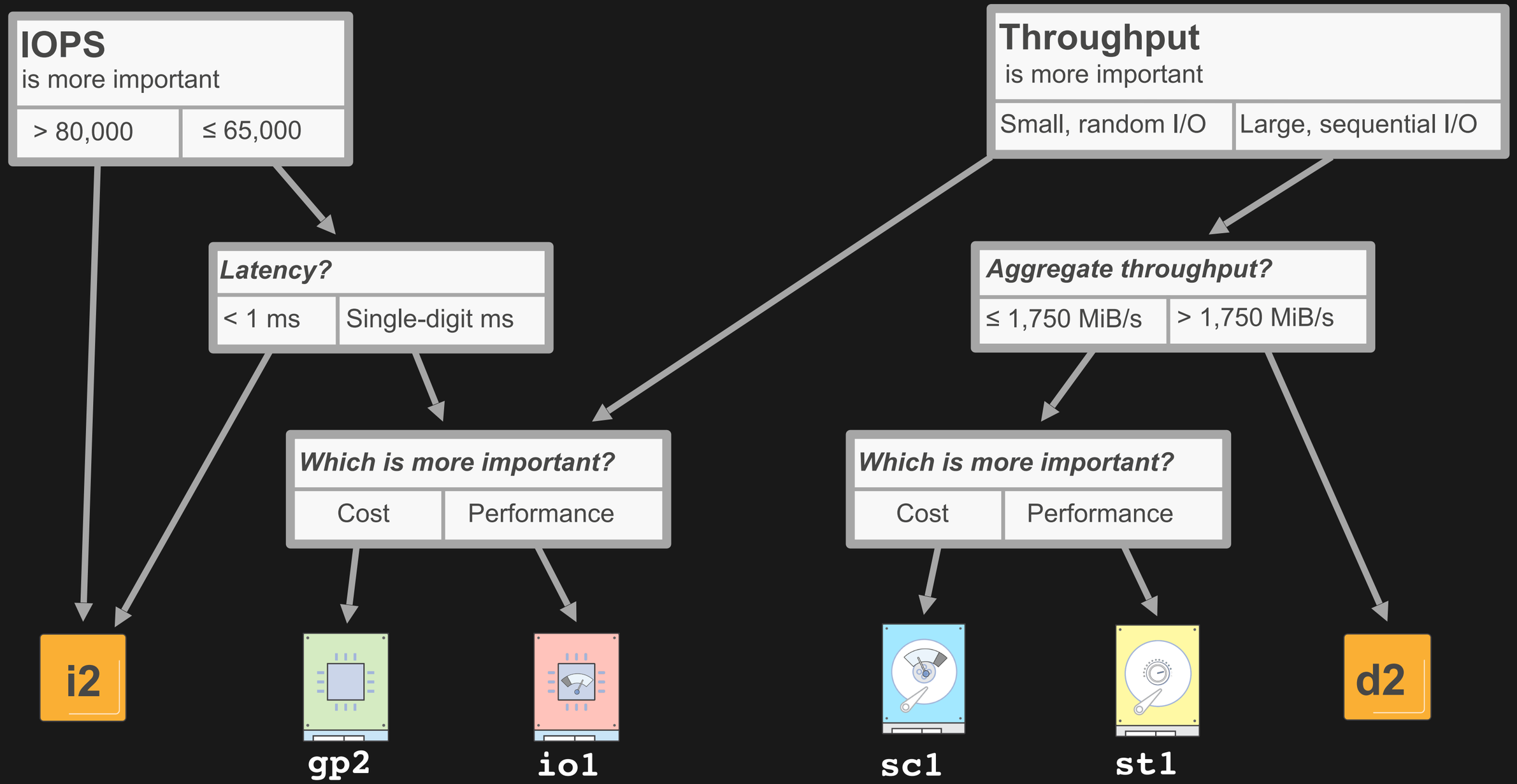
Always accumulating 320 MiB/s



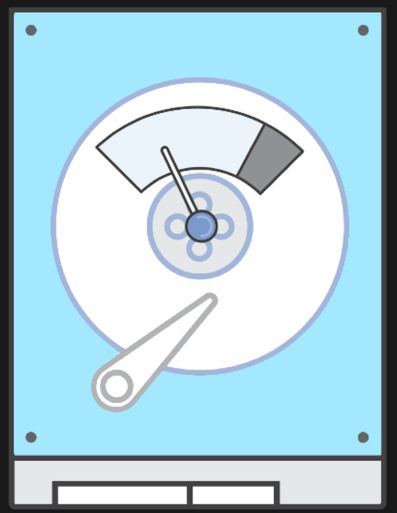
You can spend up to 500 MiB/s

Baseline performance = 320 MiB/s

Choosing an EBS volume type



EBS volume types: Throughput Provisioned



sc1

Cold HDD

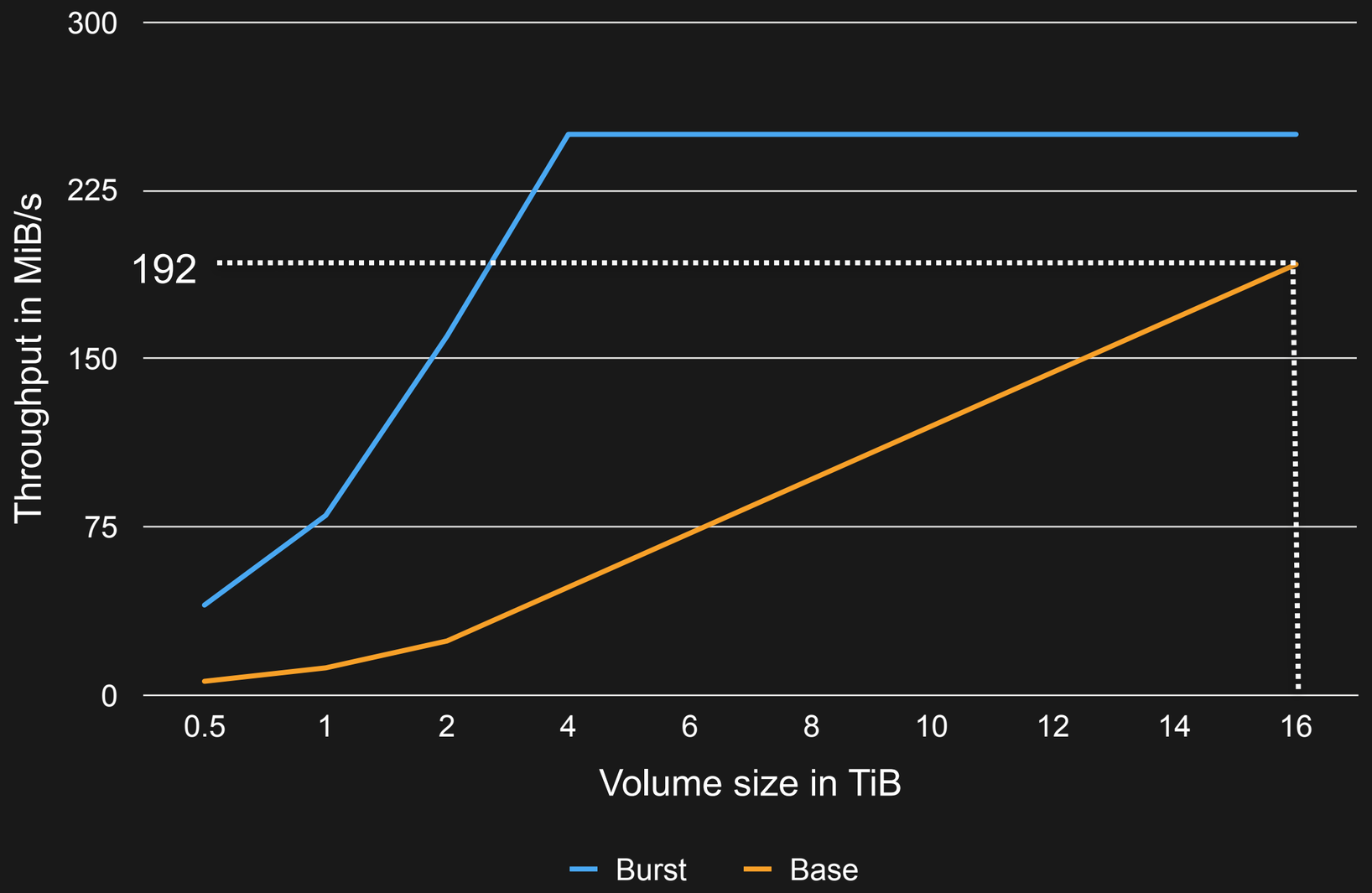
Baseline: 12 MiB/s per TB up to 192 MiB/s

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

Capacity: 500 GiB to 16 TiB

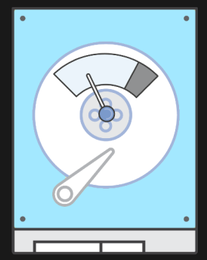
Ideal for sequential throughput workloads, such as logging and backup

Cold HDD – burst and base



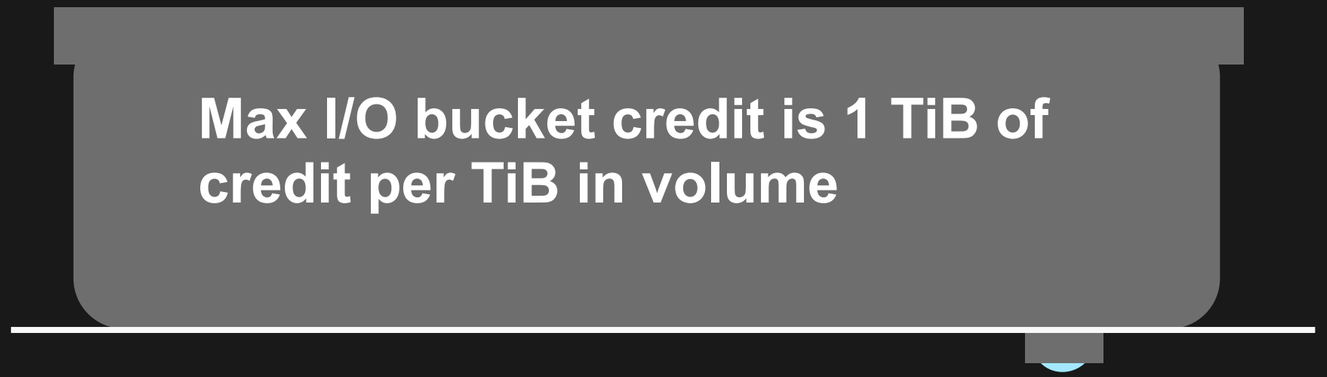
SC1

Burst bucket: Cold HDD (sc1)



sc1

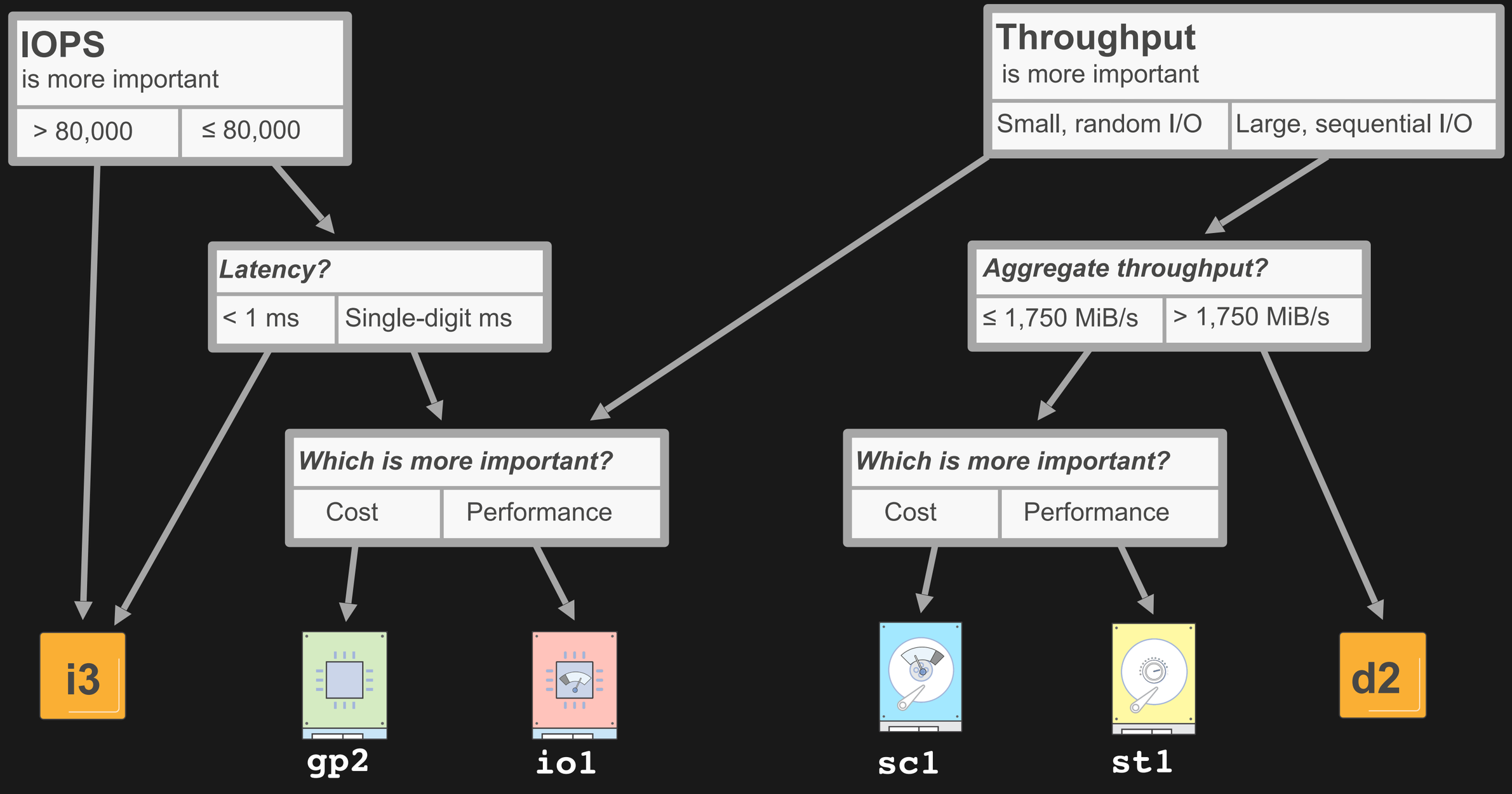
Always accumulating 12 MiB/s per TiB



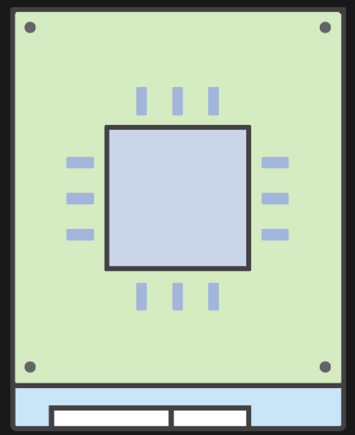
You can spend up to 80 MiB/s per TiB

Baseline performance = 12 MiB/s per TiB

Choosing an EBS volume type

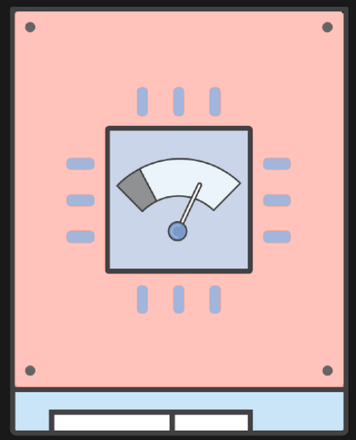


I/O Provisioned Volumes



gp2

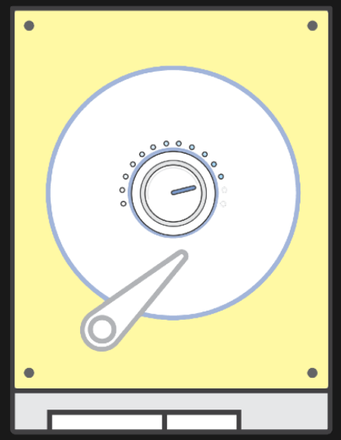
$0.10 per GiB



io1

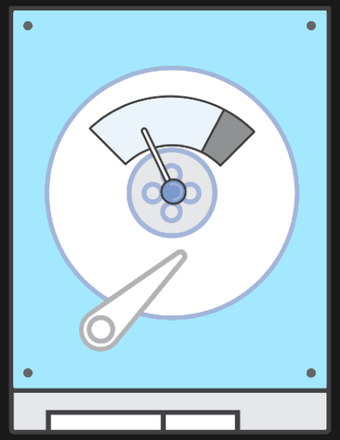
$0.125 per GiB
$0.065 per PIOPS

Throughput Provisioned Volumes



st1

$0.045 per GiB



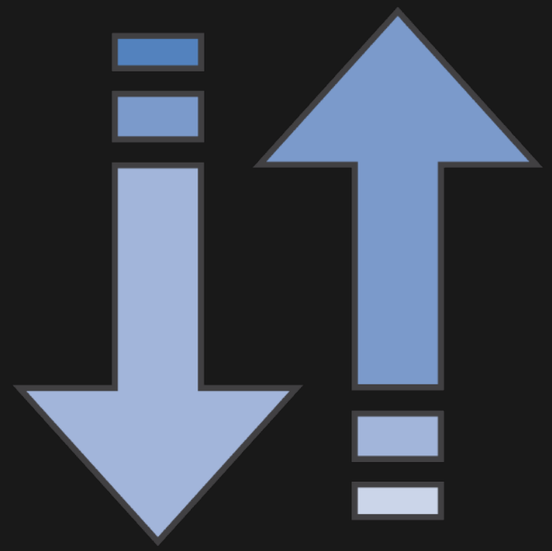
sc1

$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 hour and from the us-west-2 Region as of August 2017

EBS Elastic Volumes



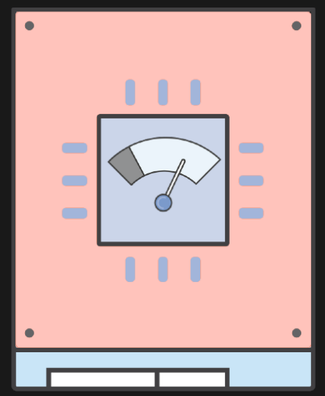
Elastic Volumes – features

- Increase volume size
- Change volume type
- Increase / decrease provisioned IOPS

*Volume modifications must be valid for the target volume type

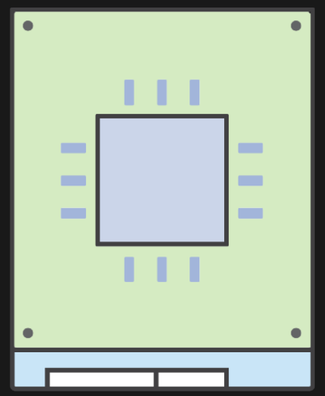
Elastic Volumes – overview

Elastic Volumes – supported volume types



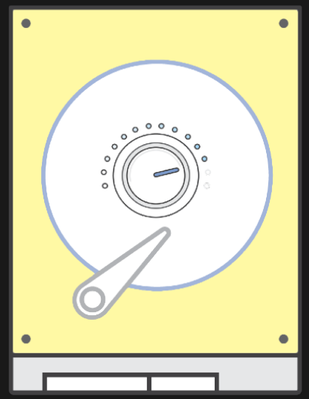
io1

Provisioned IOPS
SSD



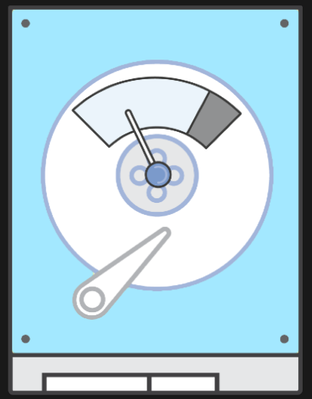
gp2

General Purpose
SSD



st1

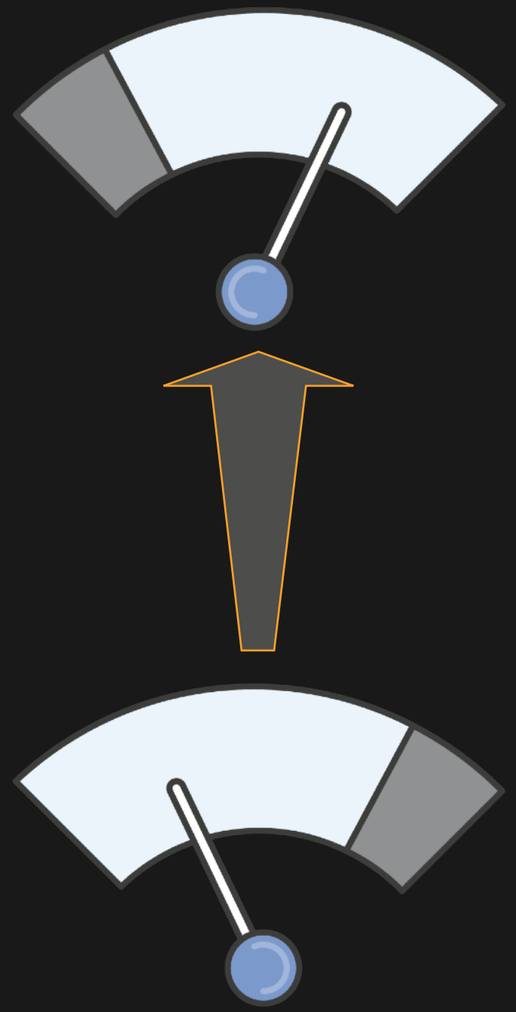
Throughput Optimized
HDD



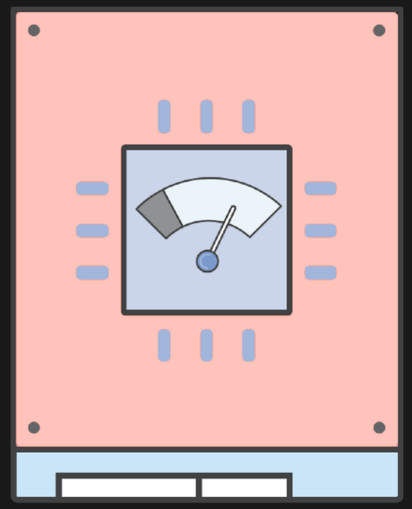
sc1

Cold
HDD

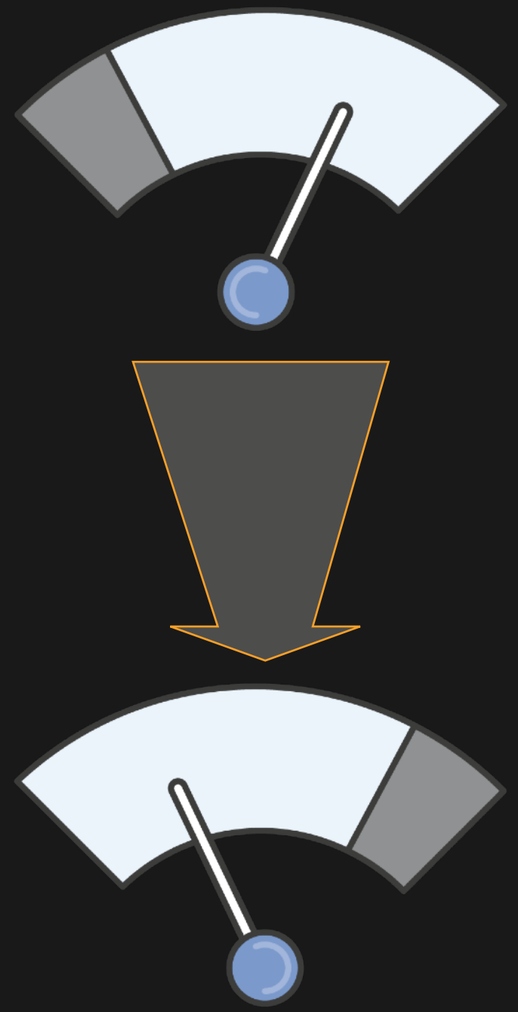
Elastic Volumes - modify provisioned IOPS



increase

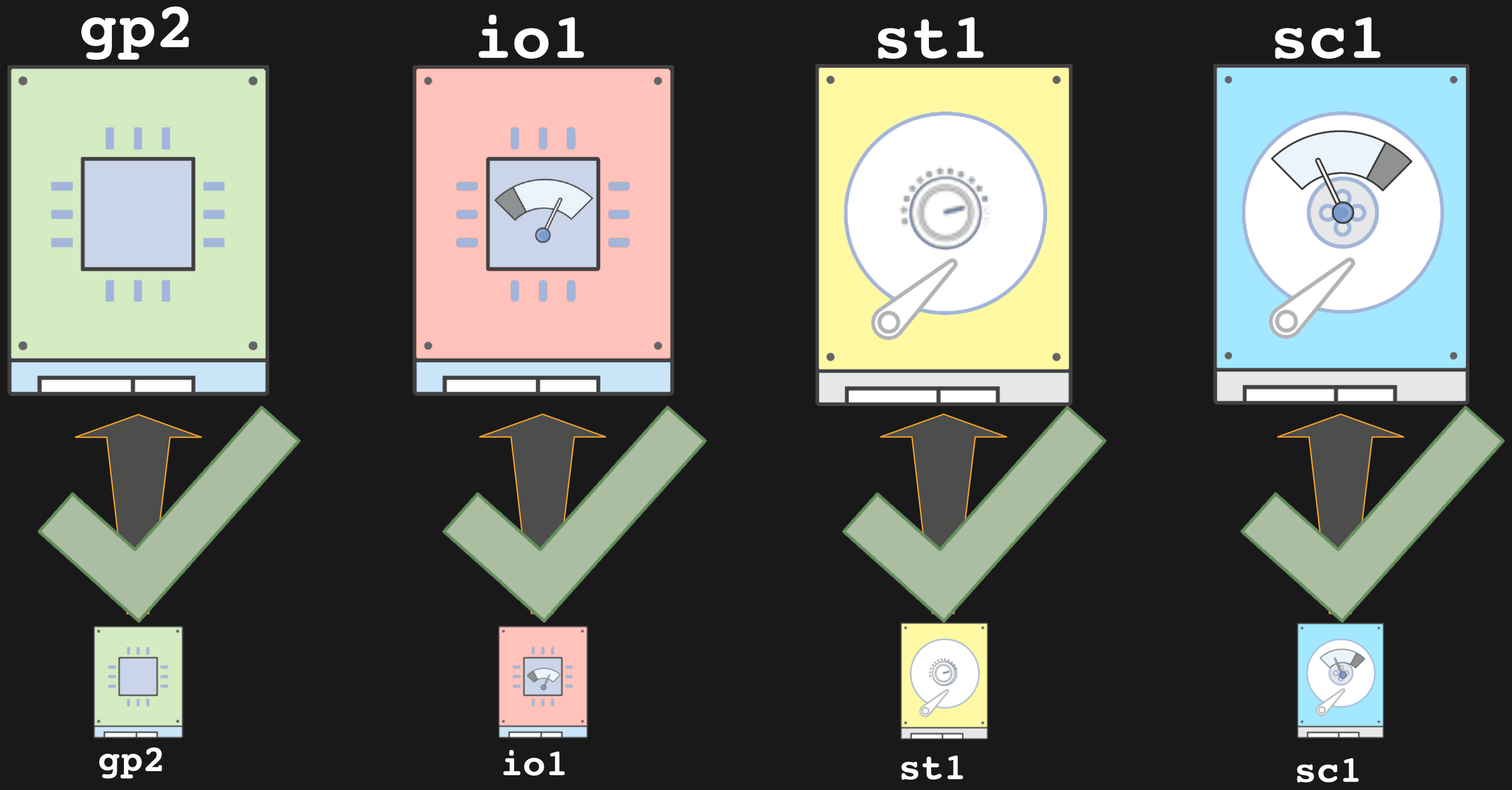


io1

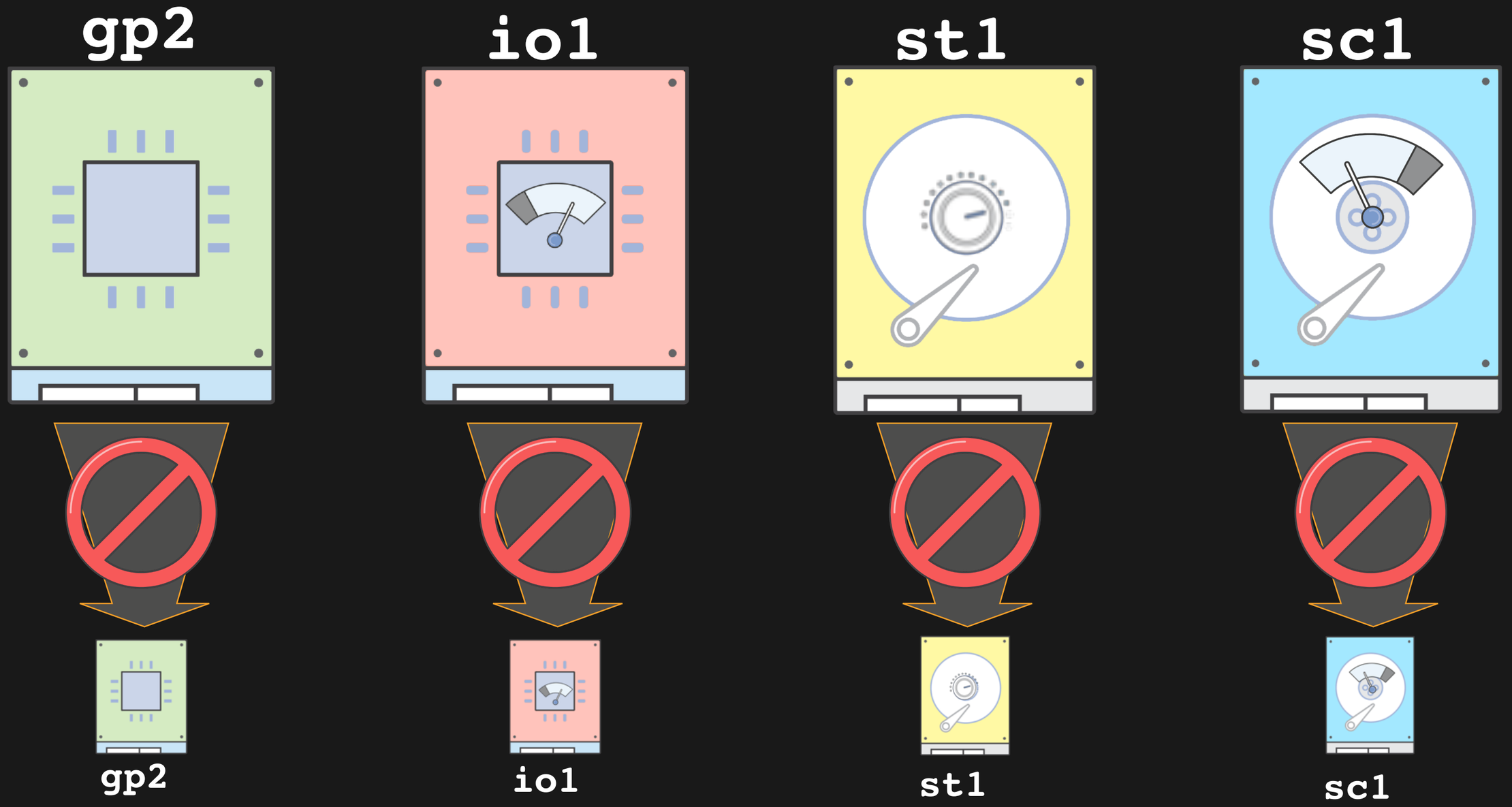


decrease

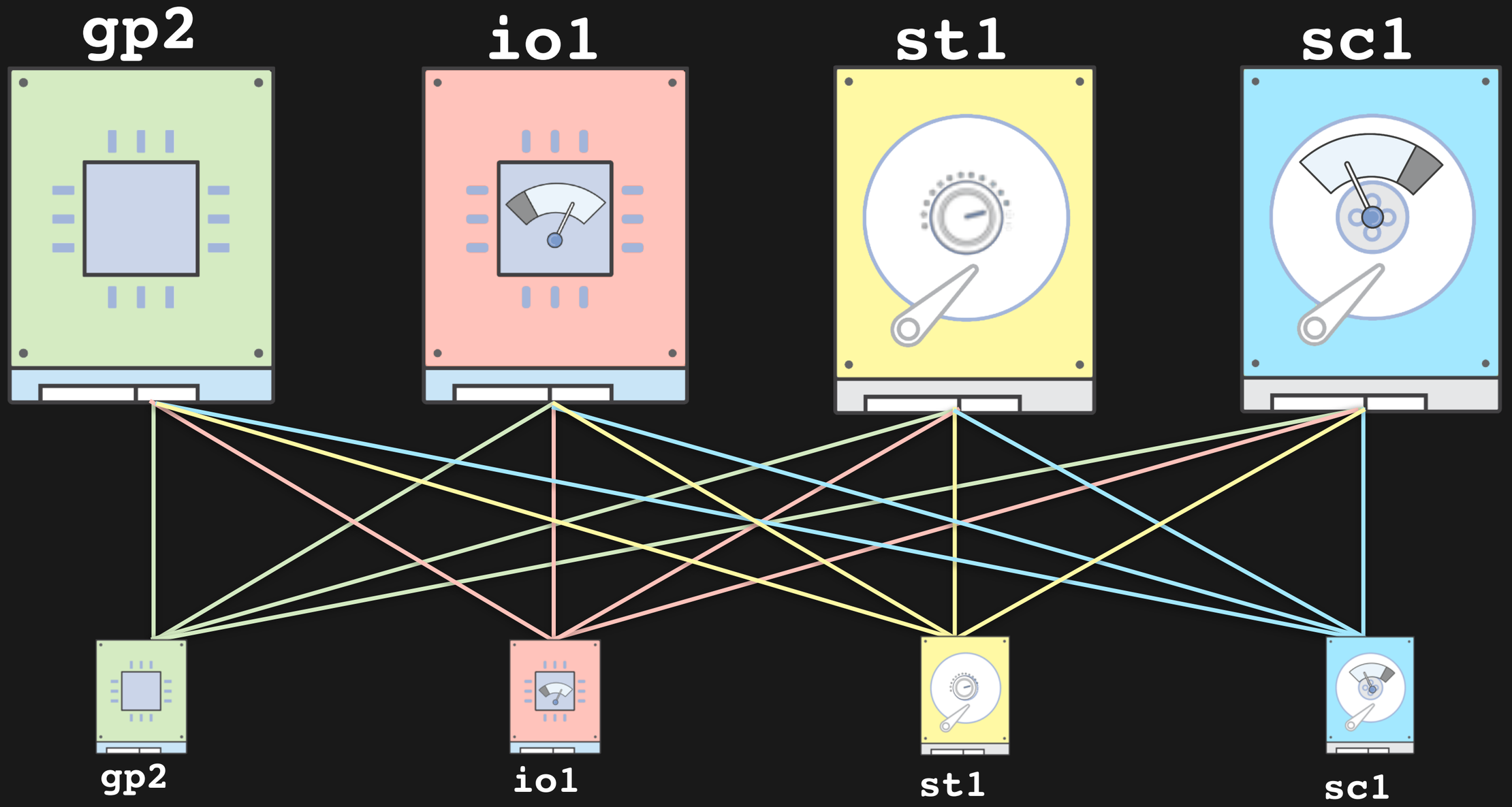
Elastic Volumes - increase volume size



Decrease volume size?



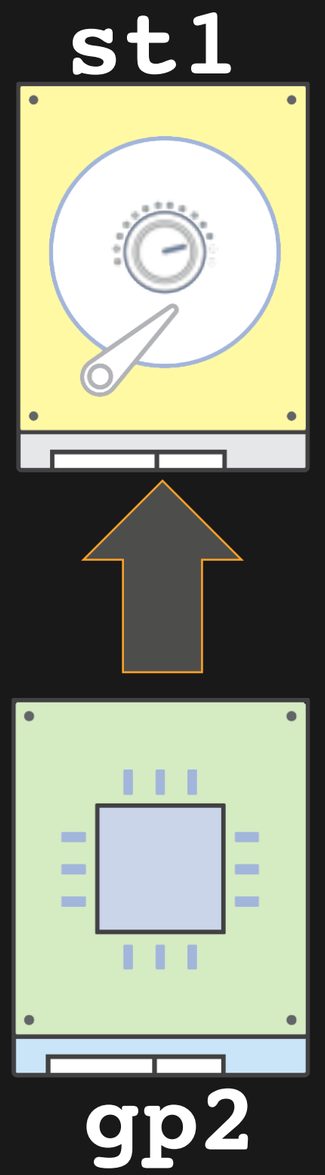
Change volume type and increase size?



Considerations

Volume modifications must be valid for the target volume type

Example

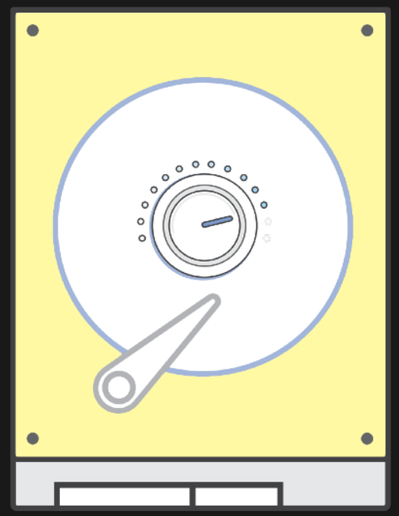


400 GiB

?

400 GiB

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

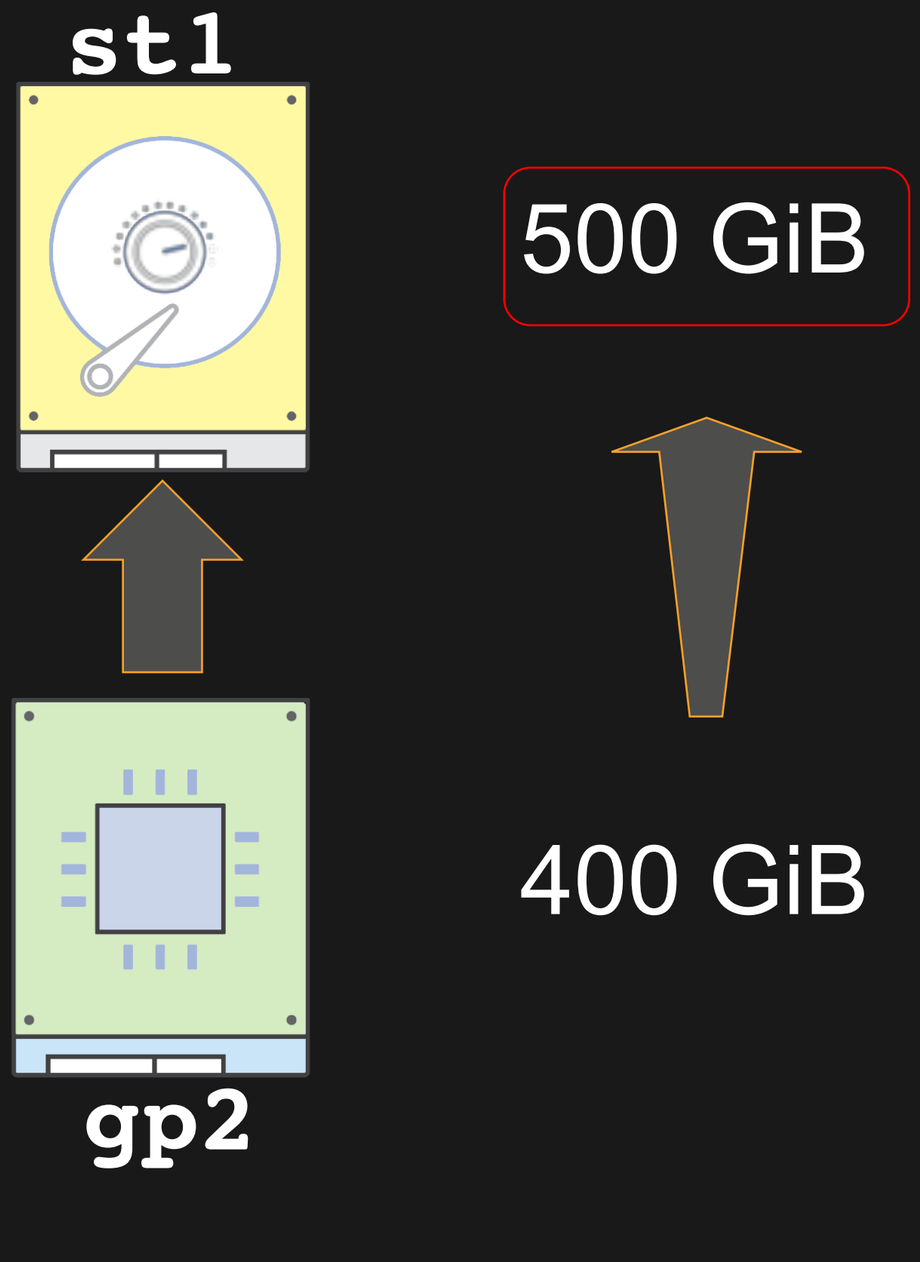
Capacity: 500 GiB to 16 TiB

Ideal for large-block, high-throughput sequential workloads

Considerations

Volume modifications must be valid for the target volume type

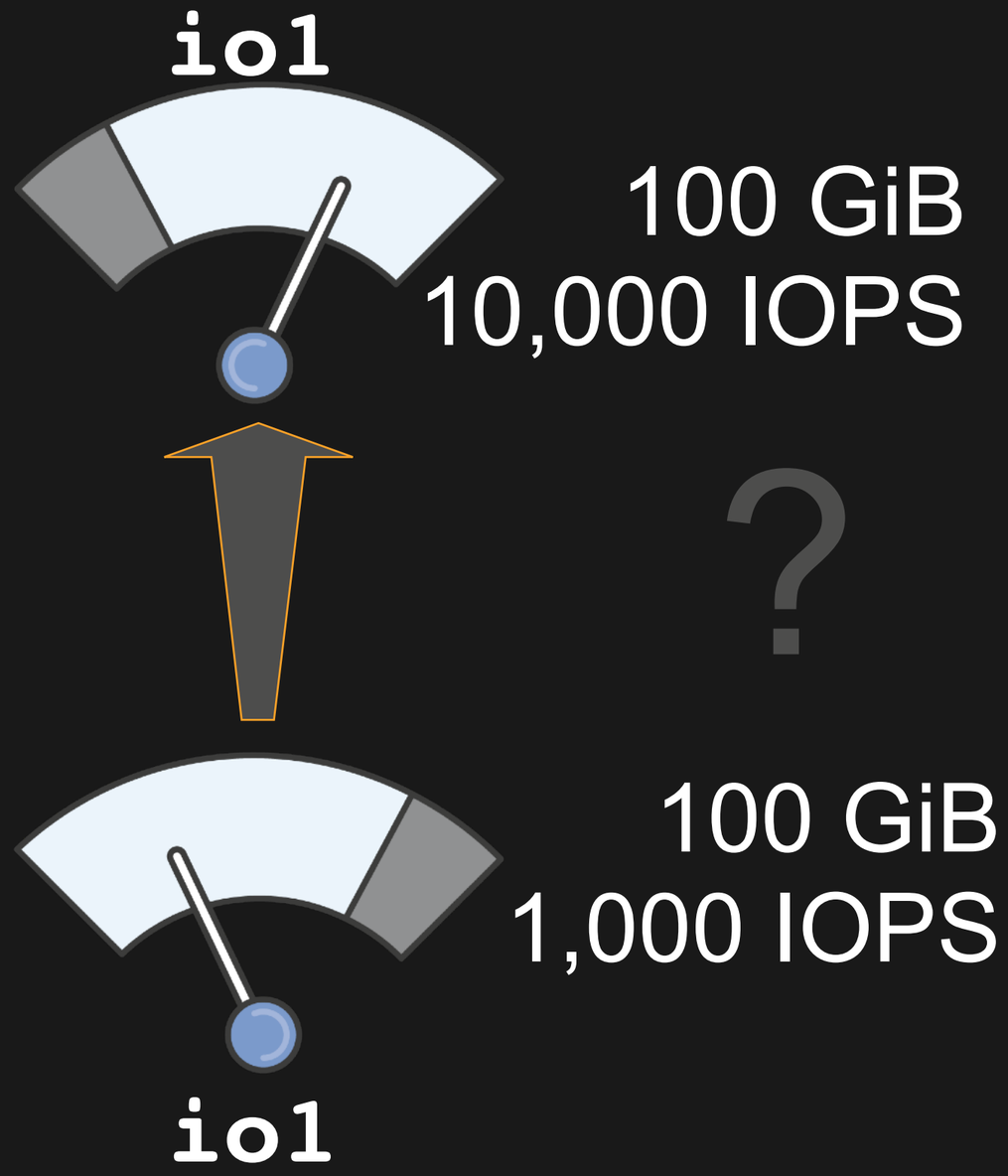
Example



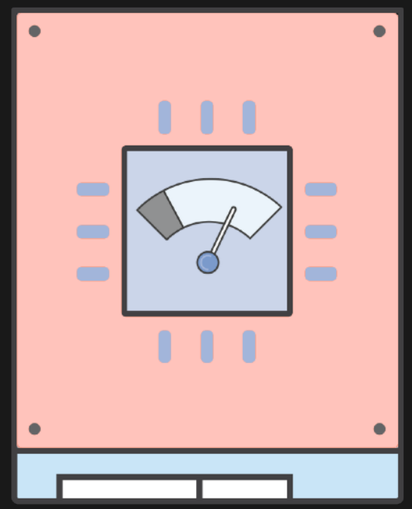
Considerations

Volume modifications must be valid for the target volume type

Example



EBS volume types: I/O Provisioned



`io1`

Provisioned IOPS SSD

Baseline: 100 to 20,000 IOPS

Throughput: 320 MiB/s

Latency: Single-digit ms

Capacity: 4 GiB to 16 TiB

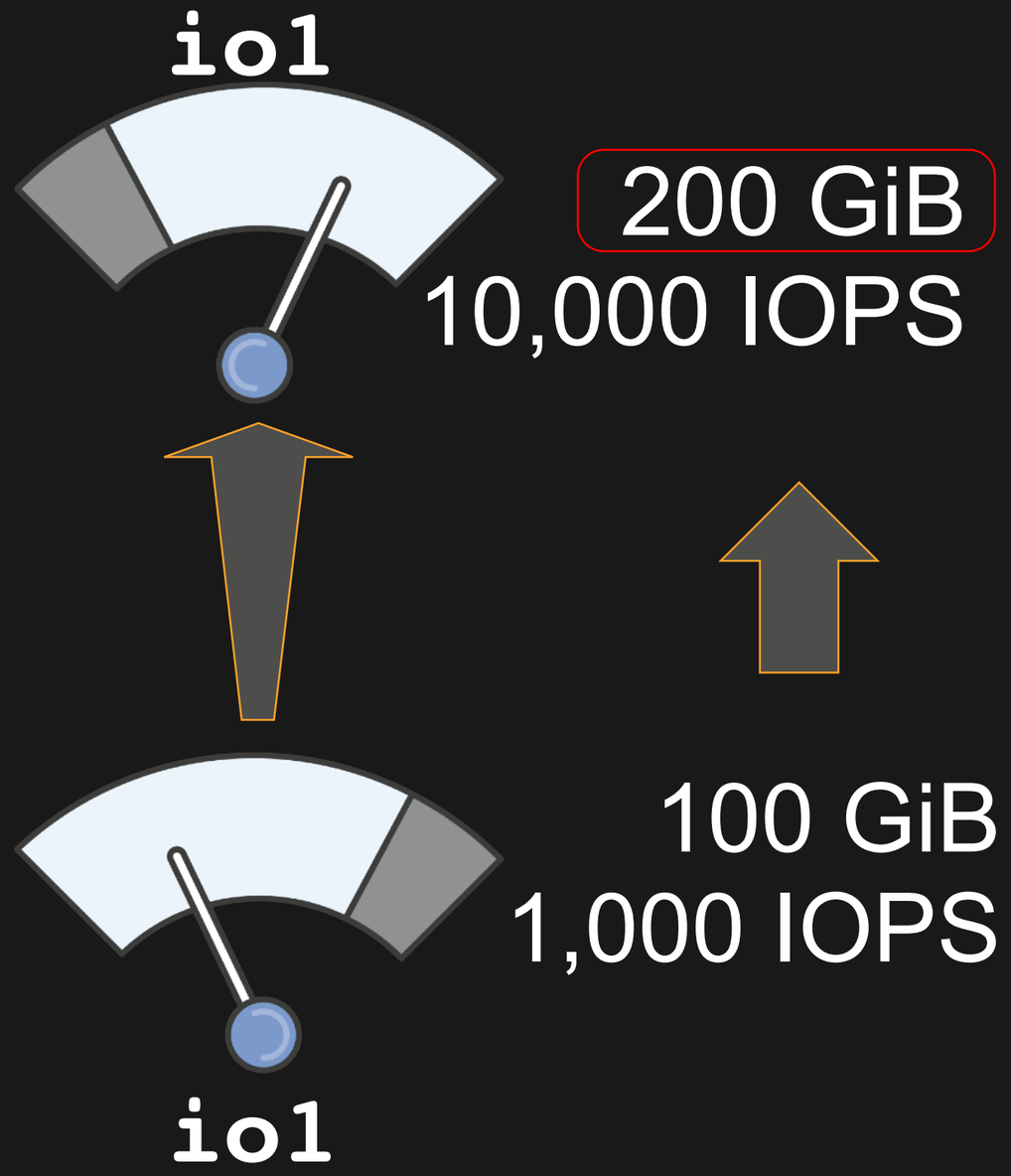
Max ratio: 50:1 IOPS to size

Ideal for critical applications and databases with sustained IOPS

Considerations

Volume modifications must be valid for the target volume type

example:



Elastic Volumes - automation ideas

Elastic Volumes

Amazon CloudWatch

Elastic Volumes - automation ideas

Right-sizing:

- Use a CloudWatch alarm to watch for a volume that is running at or near its IOPS limit.
- Initiate workflow and approval process to provision additional IOPS
- Publish a “free space” metric to CloudWatch and use a similar approval process to resize the volume and the filesystem.

Amazon CloudWatch

Elastic Volumes - automation ideas

Cost Reduction:

- Use metrics or schedules to reduce IOPS or to change the type of a volume

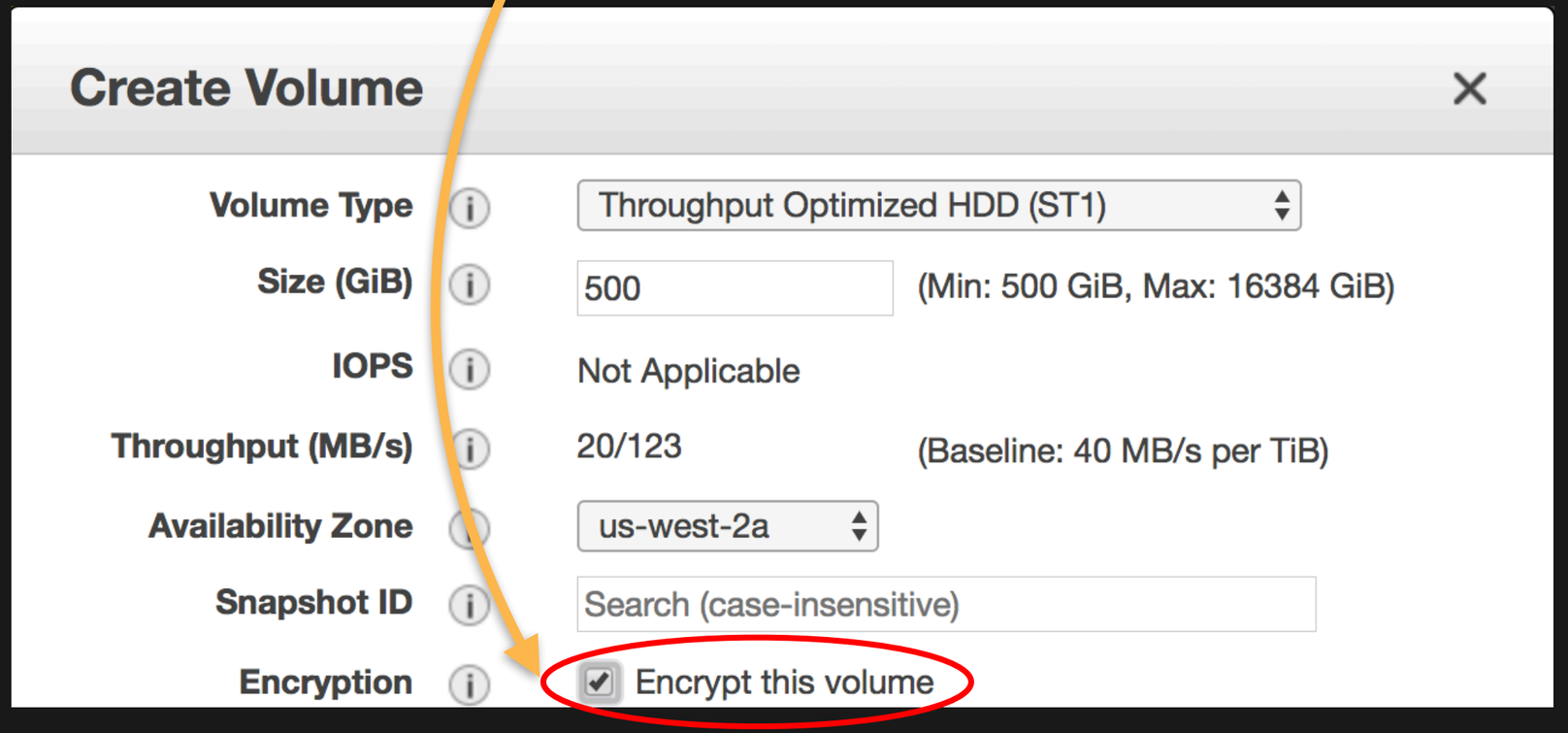
Amazon CloudWatch

https://github.com/awslabs/aws-elastic-volumes

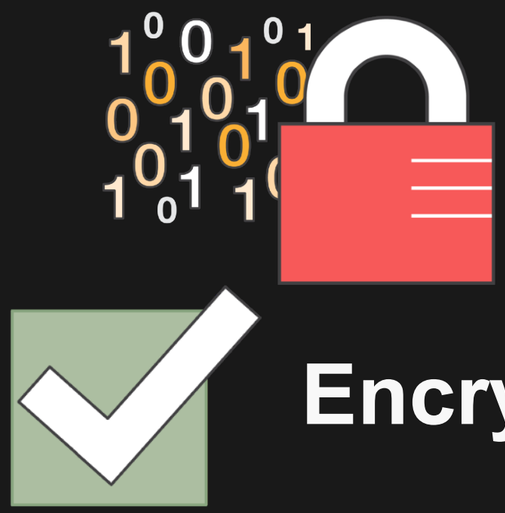
EBS encryption

What is EBS encryption?

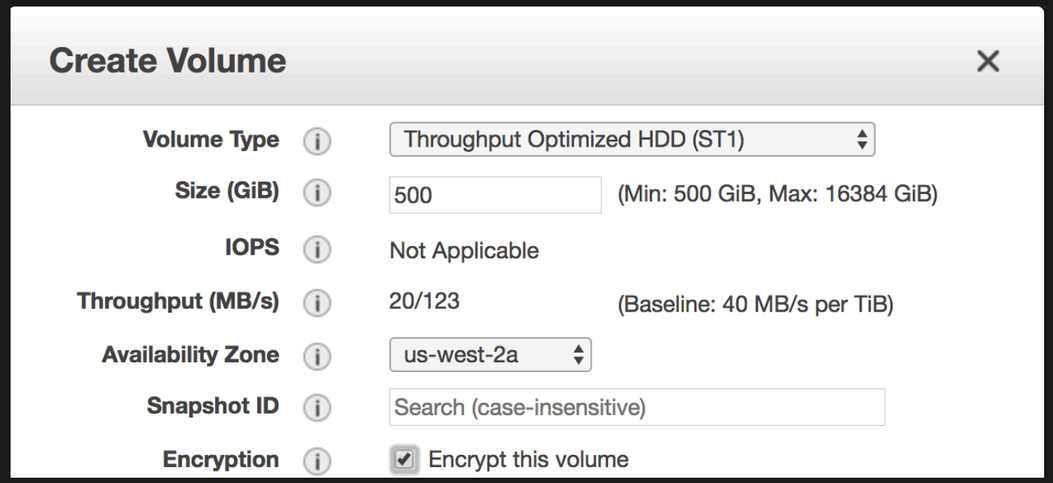
Encryption



What is EBS encryption?

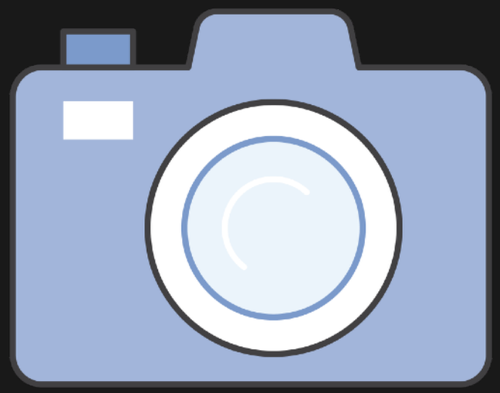


Encryption

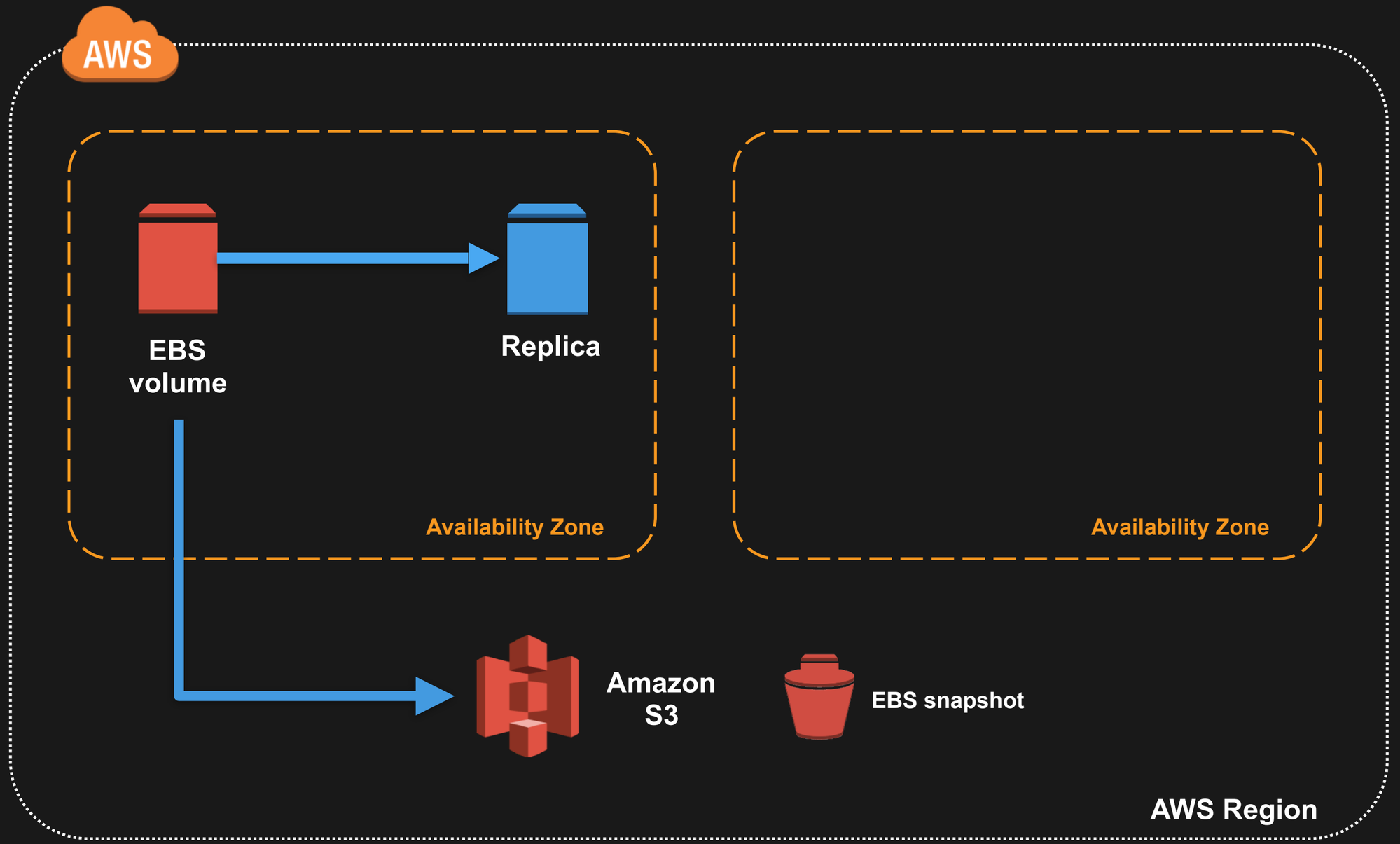


- Boot and data volumes can be encrypted
- Attach both encrypted and unencrypted
- Any current generation instance
- No volume performance impact
- Supported by all EBS volume types
- Snapshots also encrypted
- No extra cost

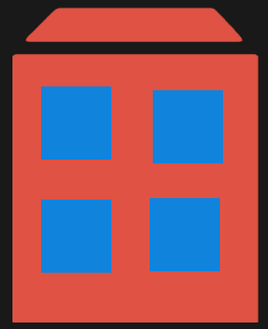
EBS snapshots



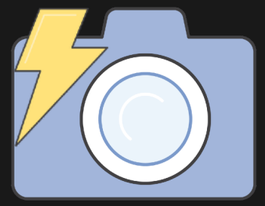
What is an EBS snapshot?

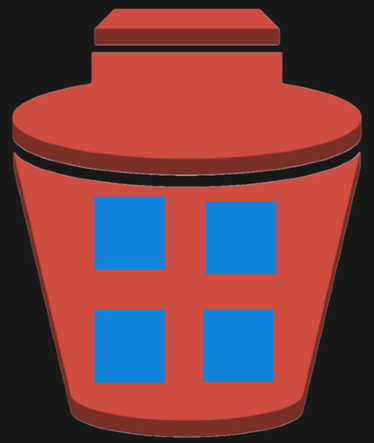


How does an EBS snapshot work?



EBS volume





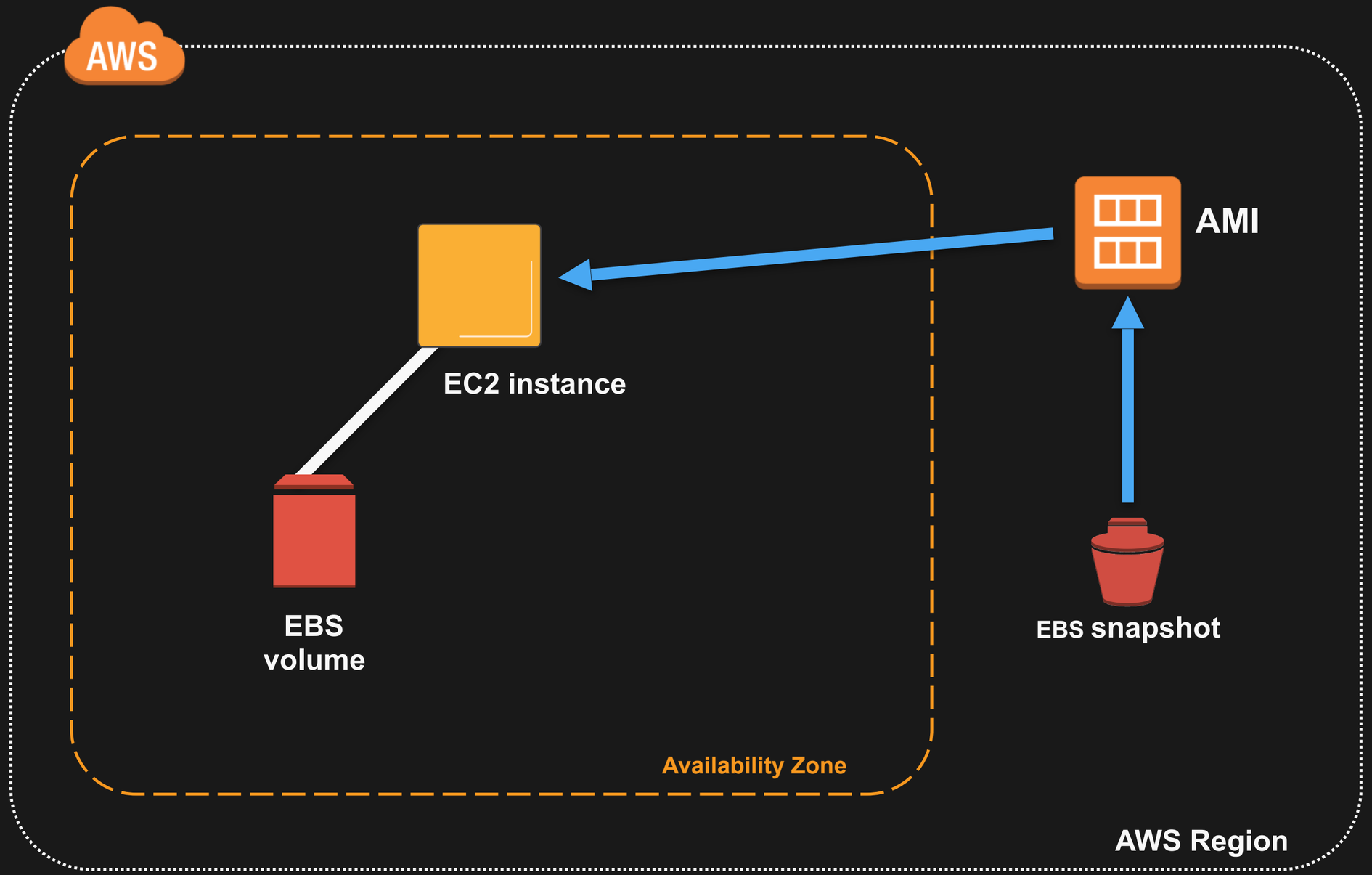
EBS snapshot

- Point-in-time backup of modified blocks
- Stored in S3, accessed via EBS APIs
- Subsequent snapshots are incremental
- Deleting a snapshot will only remove data exclusive to that snapshot

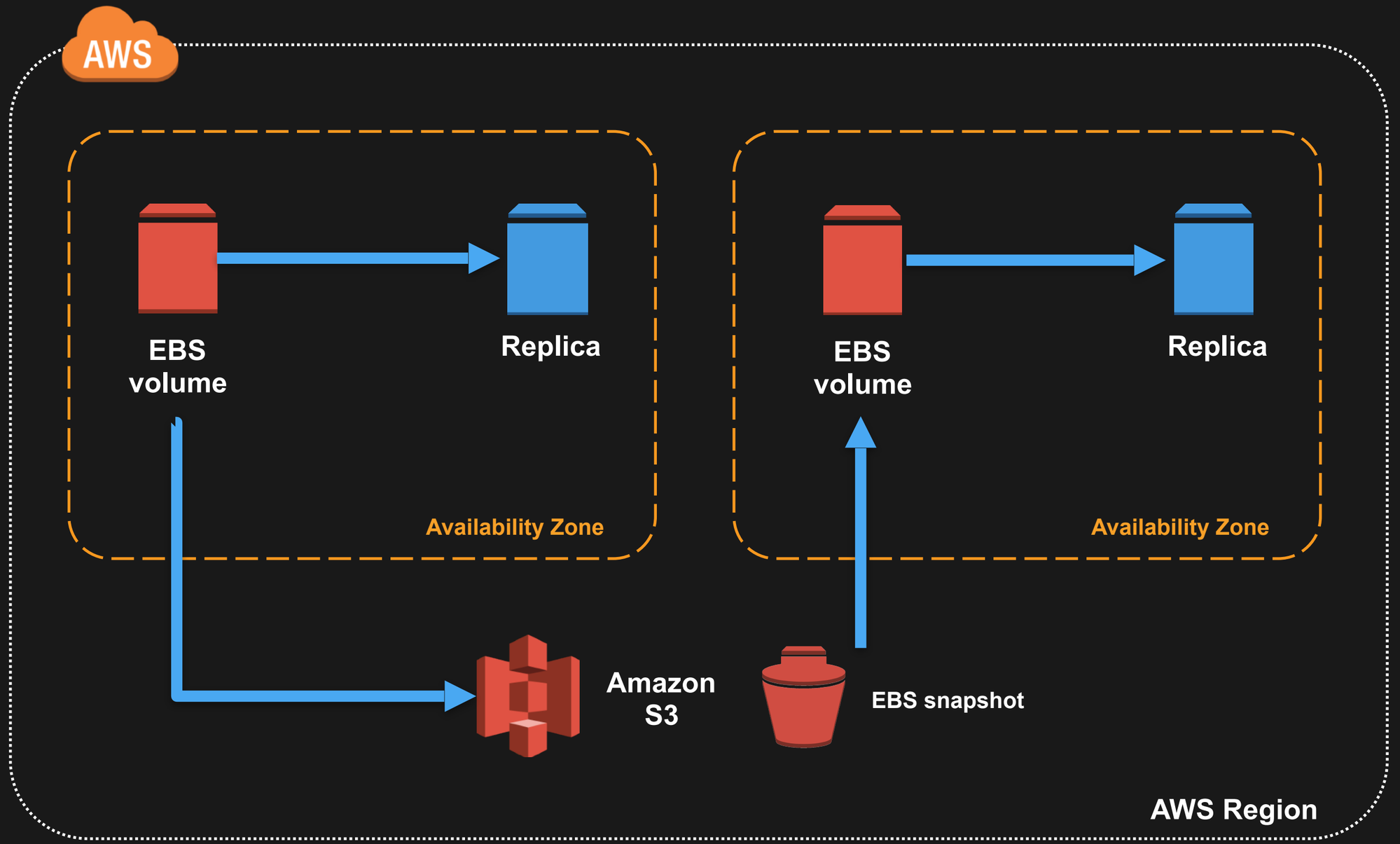
Creating an EBS snapshot

```
[ec2-user@summit ~]$ aws ec2 create-snapshot --volume-id vol-0f8e54c9346aca182
--description "Summit Snapshot"

{
    "Description": "Summit Snapshot",
    "Encrypted": true,
    "VolumeId": "vol-0f8e54c9346aca182",
    "State": "pending",
    "VolumeSize": 8,
    "Progress": "",
    "StartTime": "2017-08-14T04:48:50.000Z",
    "SnapshotId": "snap-039cc569ea4cb40f6",
    "OwnerId": "517977164187770"
}

[ec2-user@summit ~]$
```

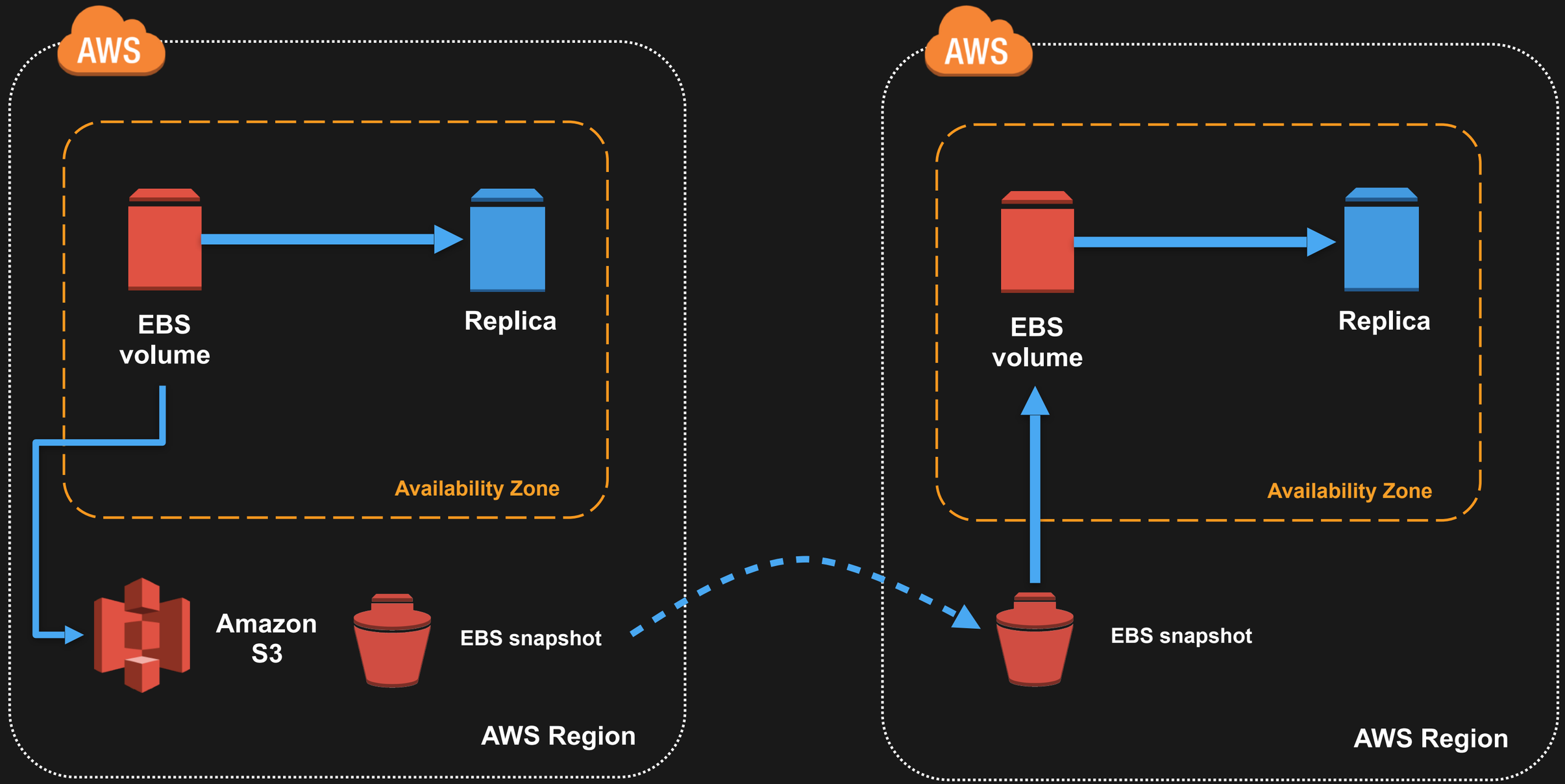
What can you do with a snapshot?



What can you do with a snapshot?

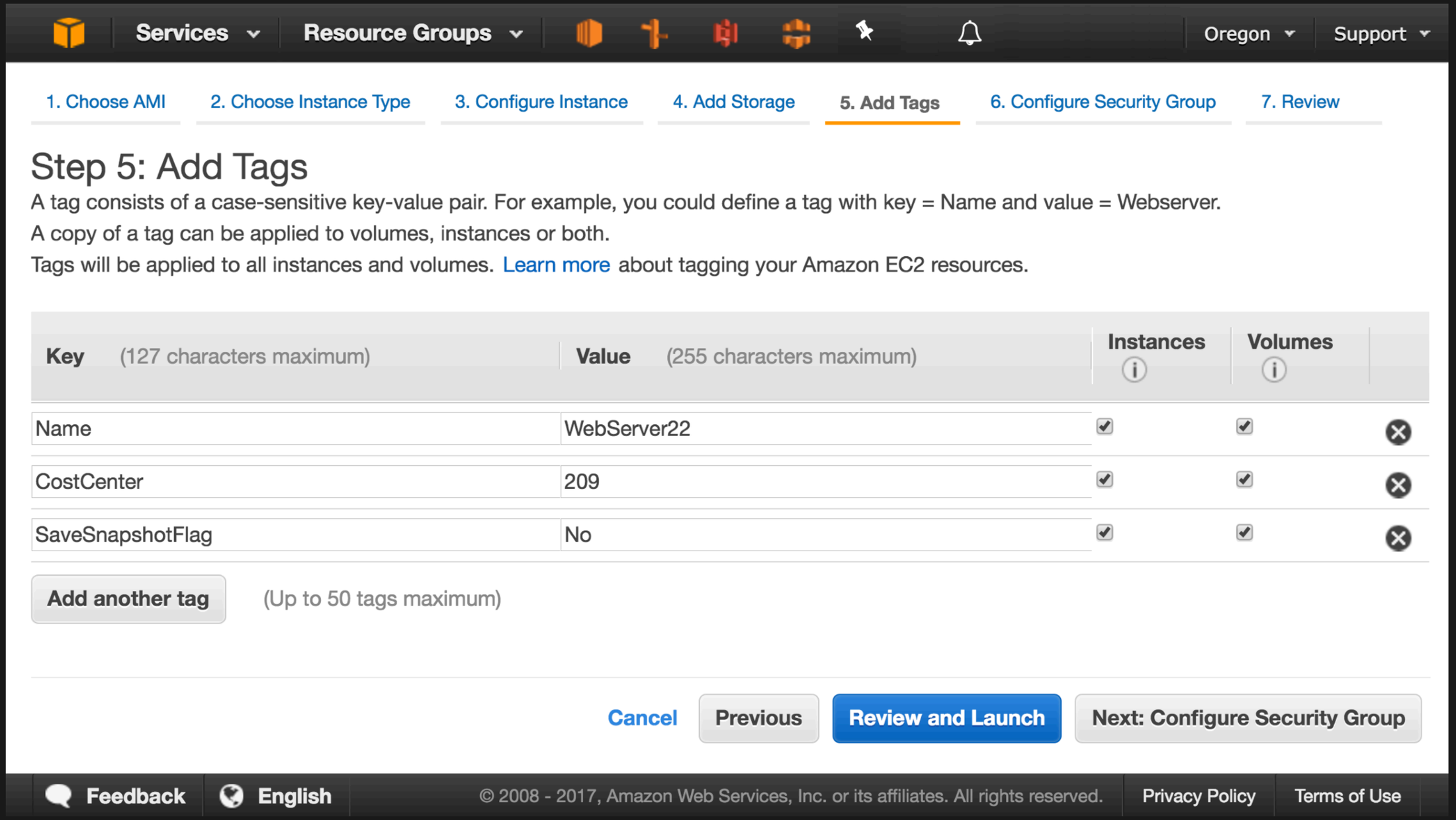


What can you do with a snapshot?



EBS tagging

EBS tagging - tag on creation



EBS tagging - enforced tag usage

- Enforce the use of specific tags with IAM
- Block the deletion of tags
- Control access to resources, ownership, and cost allocation.

EBS tagging - resource-level permissions

- `CreateTags` and `DeleteTags`
- `RunInstances` and `CreateVolume`
- Control users and groups that can tag resources on creation

EBS tagging - enforced volume encryption

- `RunInstances` and `CreateVolume`
- Mandate use of encryption for any volume created

https://aws.amazon.com/blogs/aws/new-tag-ec2-instances-ebs-volumes-on-creation/

EBS use case: Hadoop (big data)

Use Case

- EBS GP2 (SSD), ST1 and SC1 Hard Disk Drive (HDD) volumes
- Cloudera Hadoop to analyze 15TB of data / day so their customers can run advertising campaigns

Value Proposition

- **Flexibility**: Right size instance - CPU and Memory, not storage. With EBS, choose any instance
- **Cost**: More capacity and less processing at a lower price per request
- **Elastic:** increase capacity, tune performance, and change volume types live (no downtime)
- **Data Persistence**: Detach and reattach volumes to resize your cluster
- **Operational Efficiency**: Easy to expand, no more babysitting cluster

Videology Example

- Chose M4.10xlarge + **ST1 HDD Throughput Optimized EBS**

“We were able to save $15,000 per month, increase our available storage by 5%, and turn off eight server nodes by moving to Amazon EBS ST1 volumes to support our Hadoop cluster”

EBS use case: Cassandra (NoSQL)

Use Case

- Elasticity on demand at a cost-effective endpoint to analyze petabytes of network traffic
- Process, store, and analyze billions of network packets (metadata) and petabytes of raw packet data

Value Proposition

- **Flexibility**: Right size instance - CPU and Memory, not storage. With EBS, choose any instance
- **Cost**: Less expensive than on-prem, saved 30% costs by leveraging multi-tiered storage system with EBS
- **Performance**: Search petabytes of stored data with 1-3 second response time
- **Data Persistence**: Detach and reattach volumes to resize your cluster, minutes to move data that used to take days or weeks

Protectwise Example

- Chose I2 EC2 (hot data) + **GP2, ST1 EBS (warm data),** and S3 (cold data**)**

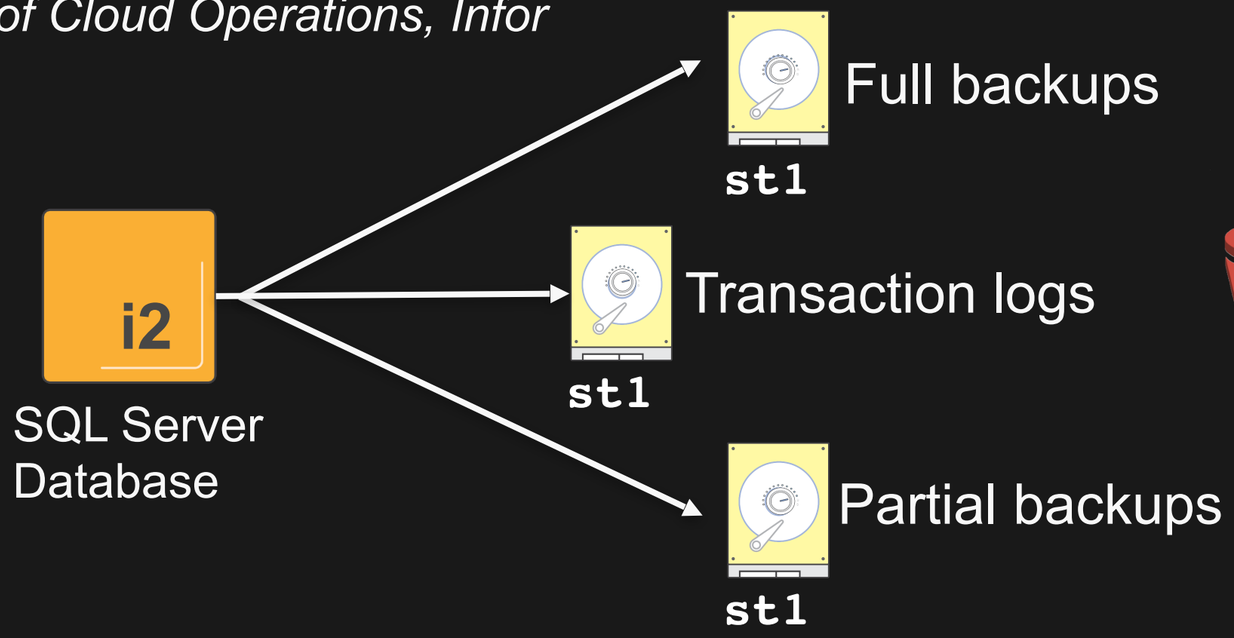
“Using AWS, we have reduced our storage costs by 95%, meaning we are spending $1 for every $20 we would have spent on a traditional system.”

EBS use case: hybrid volumes

Case Study:
https://aws.amazon.com/solutions/case-studies/infor-ebs/

*"We've seen much stronger performance for our database backup workloads with the Amazon **EBS ST1** volumes, and we're also saving **75 percent** on our monthly backup costs."*

~ Randy Young, Director of Cloud Operations, Infor



EBS snapshots

EBS deep dive

EBS deep dive

Performance

How do we count I/Os for GP2 and IO1?

When possible, we logically **merge** sequential I/Os (up to **256 KiB** in size)

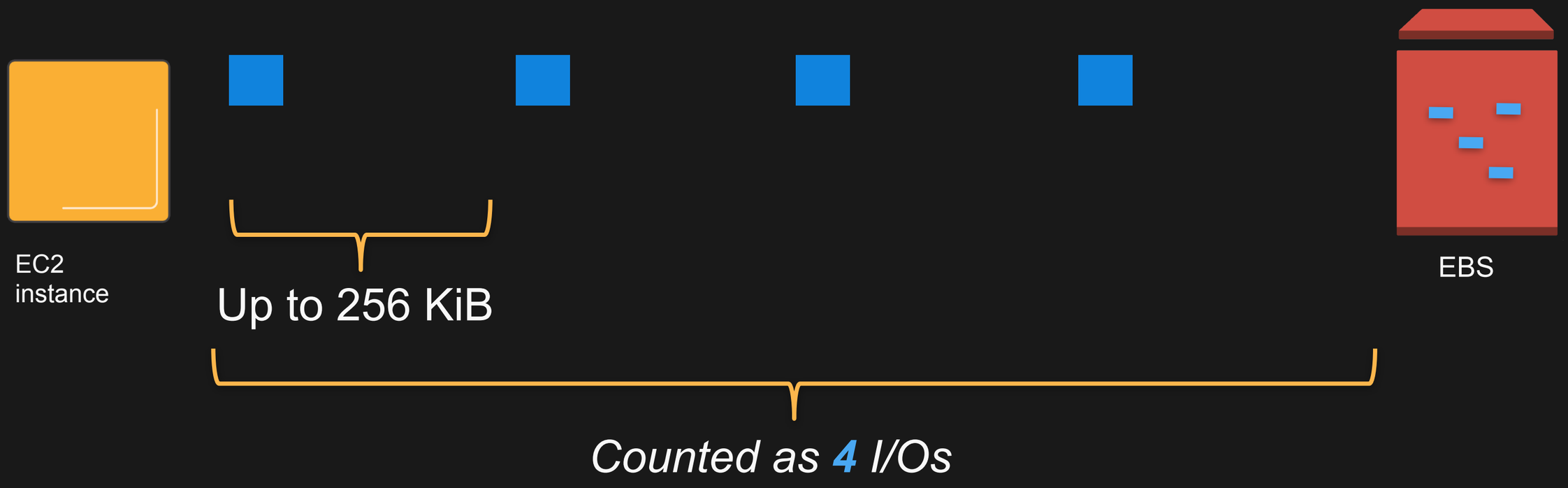
...To *minimize* I/O charges on IO1

and *maximize* burst on GP2

How do we count I/Os for GP2 and IO1?

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 for GP2 and IO1?

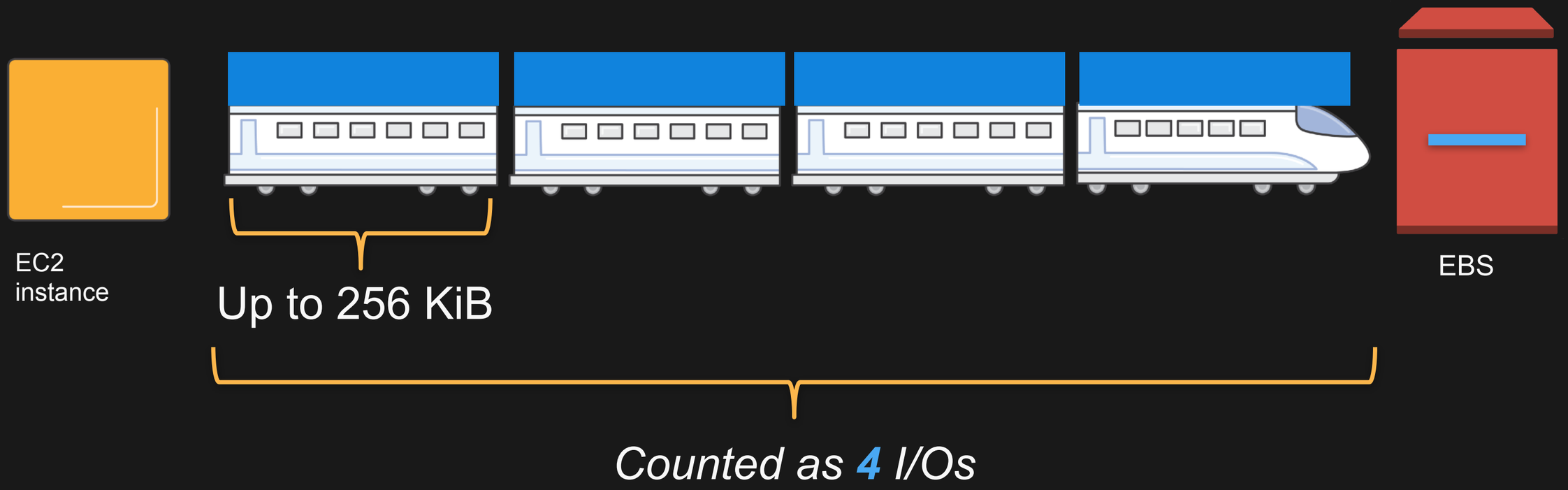
Example 2: Sequential I/O

- 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
- **1024 KiB**



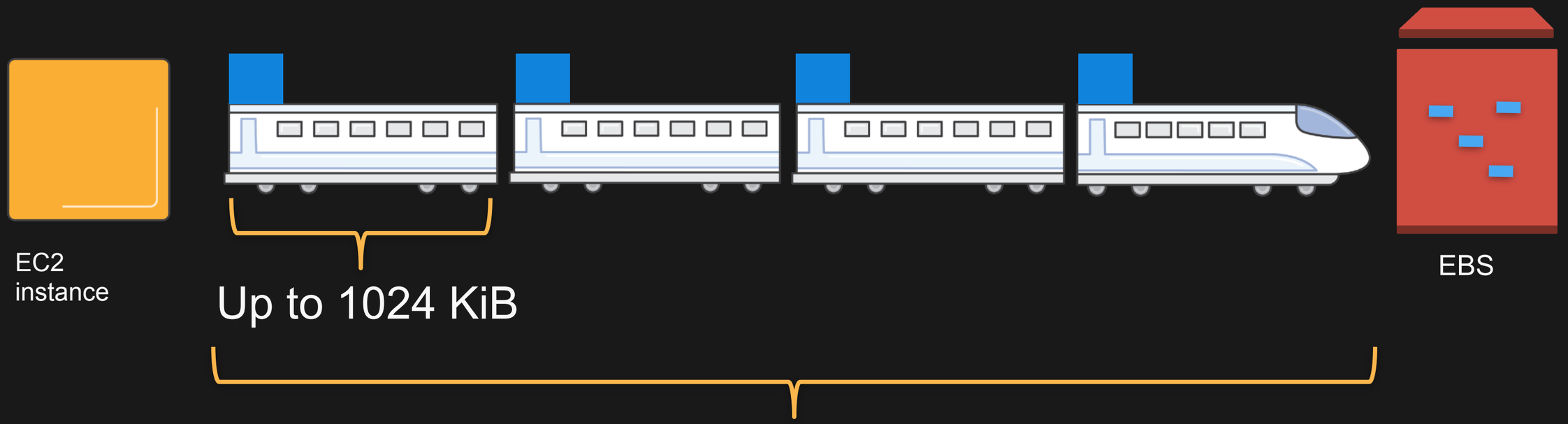
How do we count I/Os for ST1 and SC1?

- When possible, we **merge** sequential I/Os (up to **1 MiB** in size)
  - Workloads with primarily large, sequential I/Os perform best on ST1 and SC1
  - Ex: Big data / EMR, Hadoop, Kafka, log processing, data warehouses

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

Example 1: Random I/Os

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



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

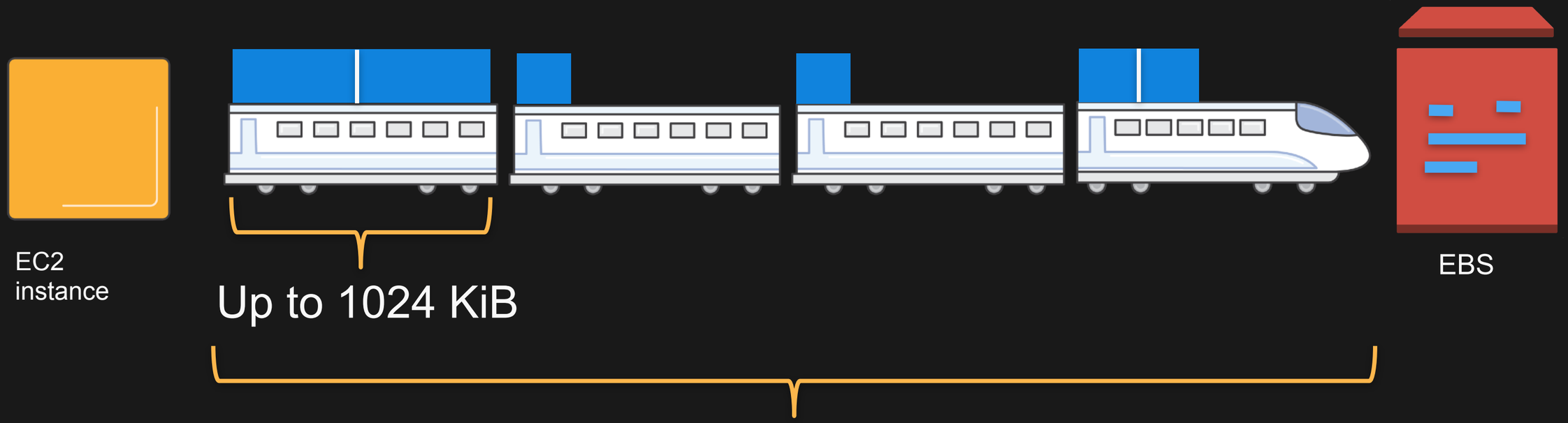
Example 2: Sequential I/O

- 4 sequential I/Os
- Each I/O **1024 KiB**

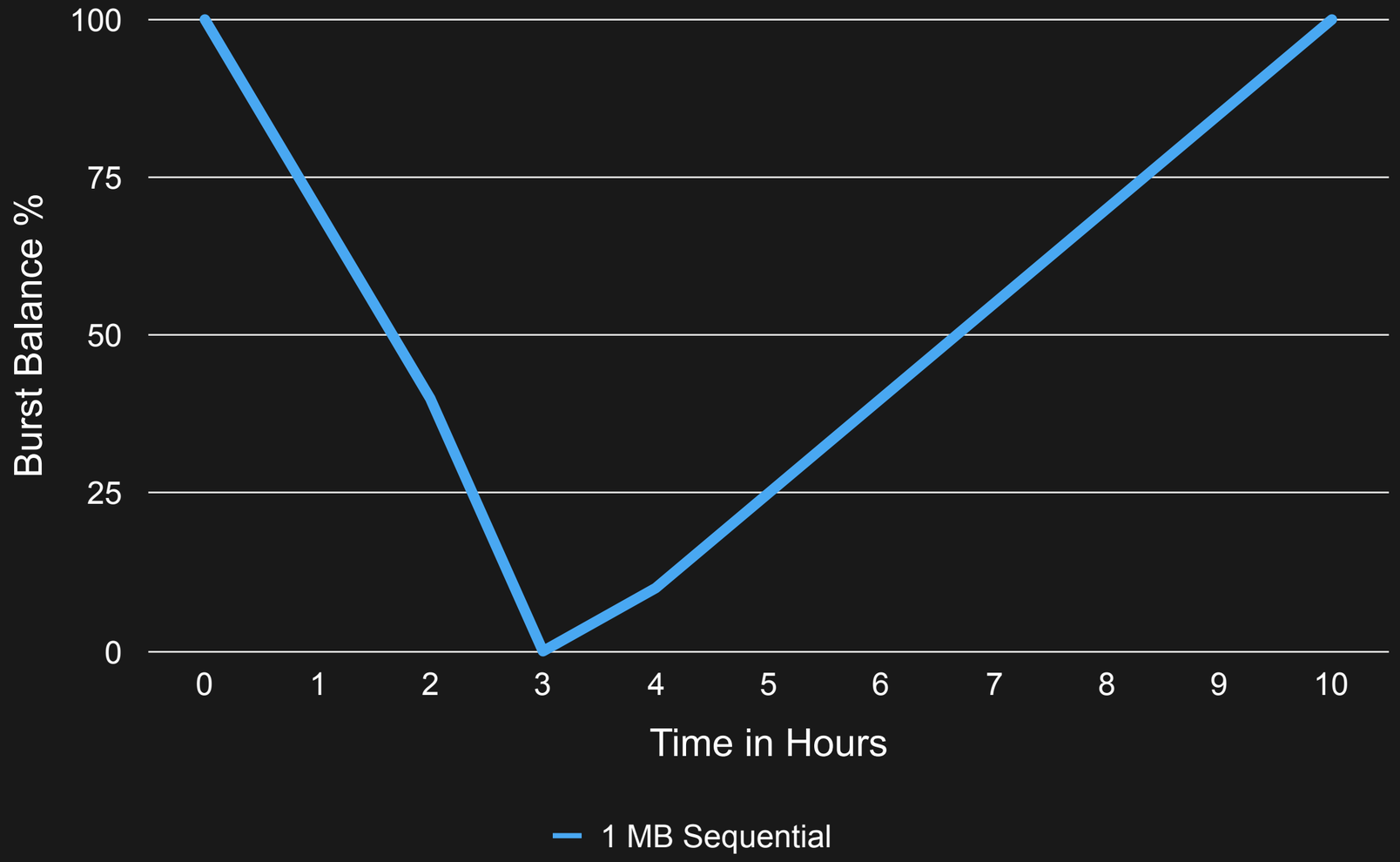
How do we count I/Os for ST1 and SC1?

Example 3: Mixed I/O

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



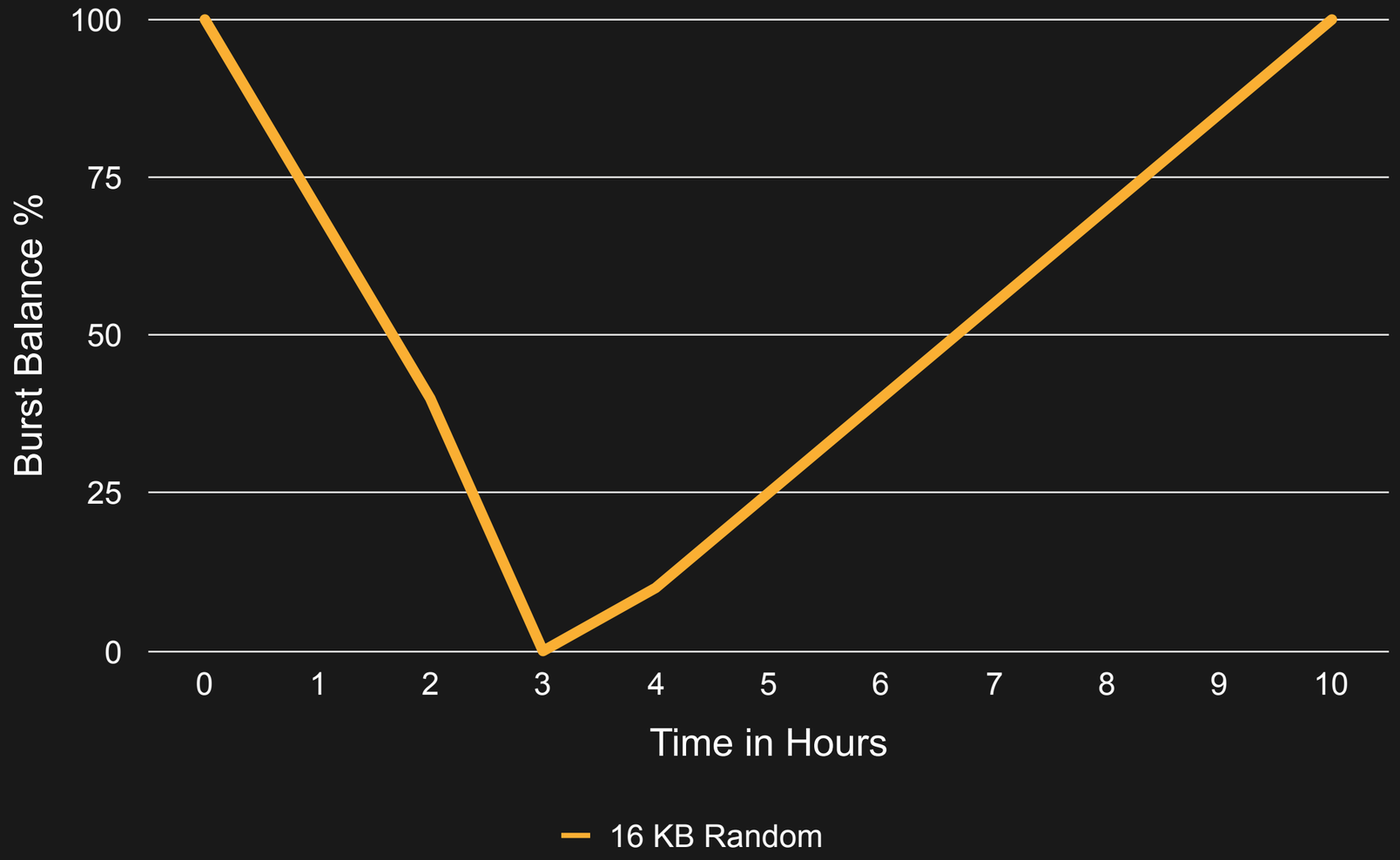
Burst balance for ST1 and SC1 - sequential



4 TiB ST1 volume

1 MiB sequential:
500 MB/s for 3 hours

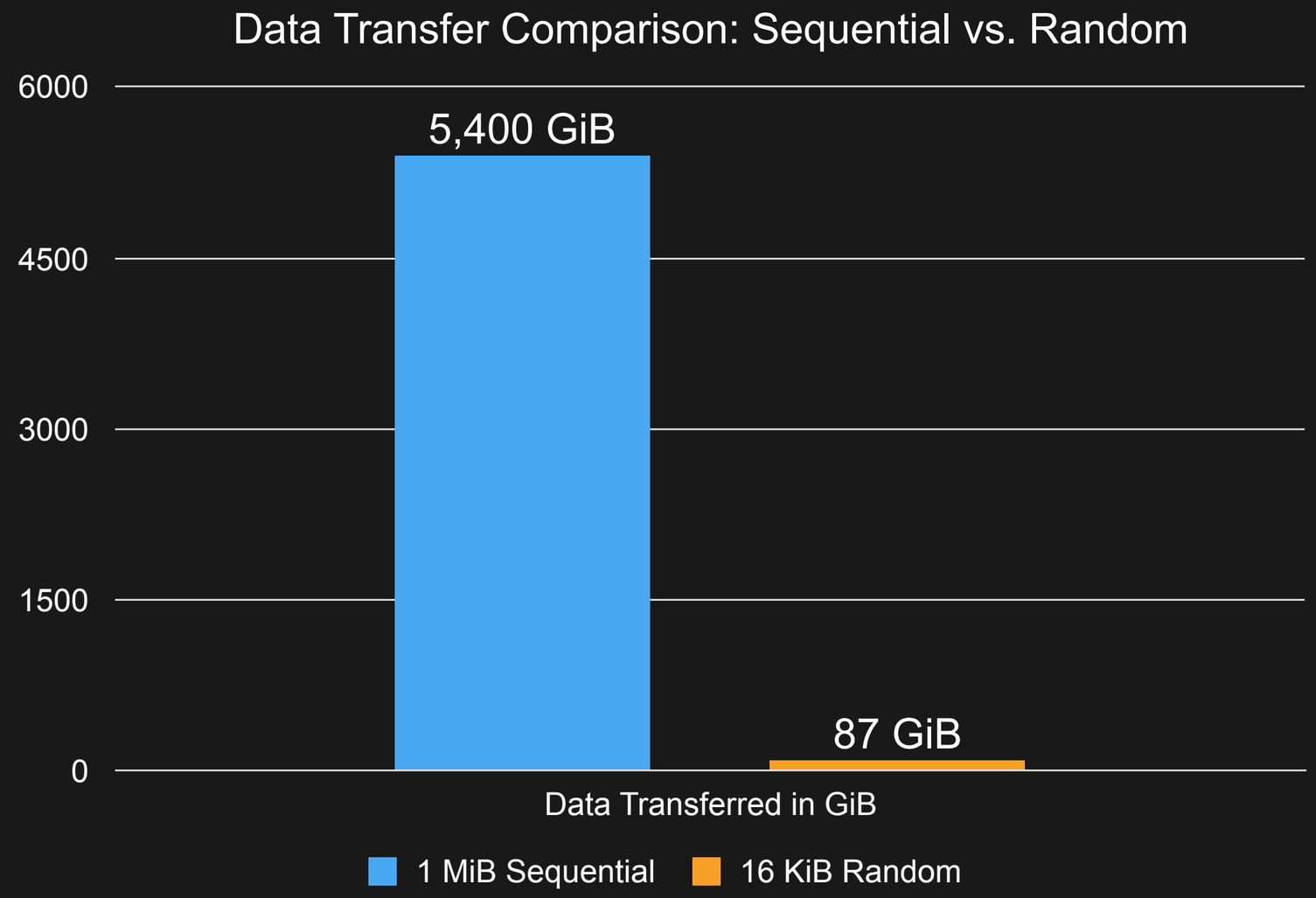
Burst balance for ST1 and SC1 - random



4 TiB ST1 volume

16 KiB random:
8 MB/s for 3 hours

Burst balance for ST1 and SC1



4 TiB ST1 volume

1 MiB sequential:
5.4 TiB transferred

16 KiB random:
87 GiB transferred

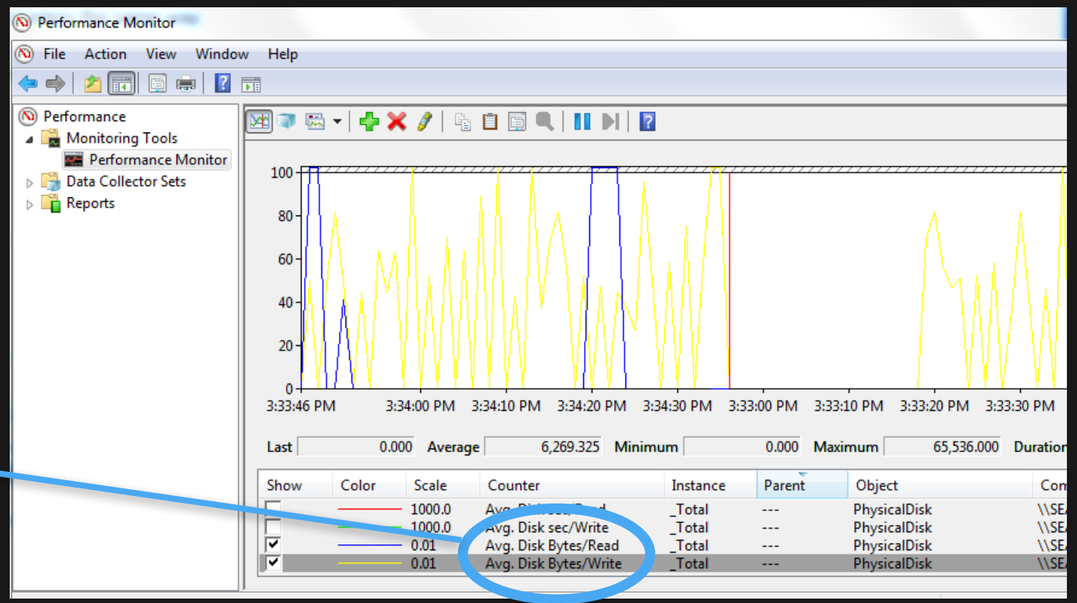
Verify workload I/O patterns

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
```

2046 sectors x 512 bytes/sector = ~1024 KiB

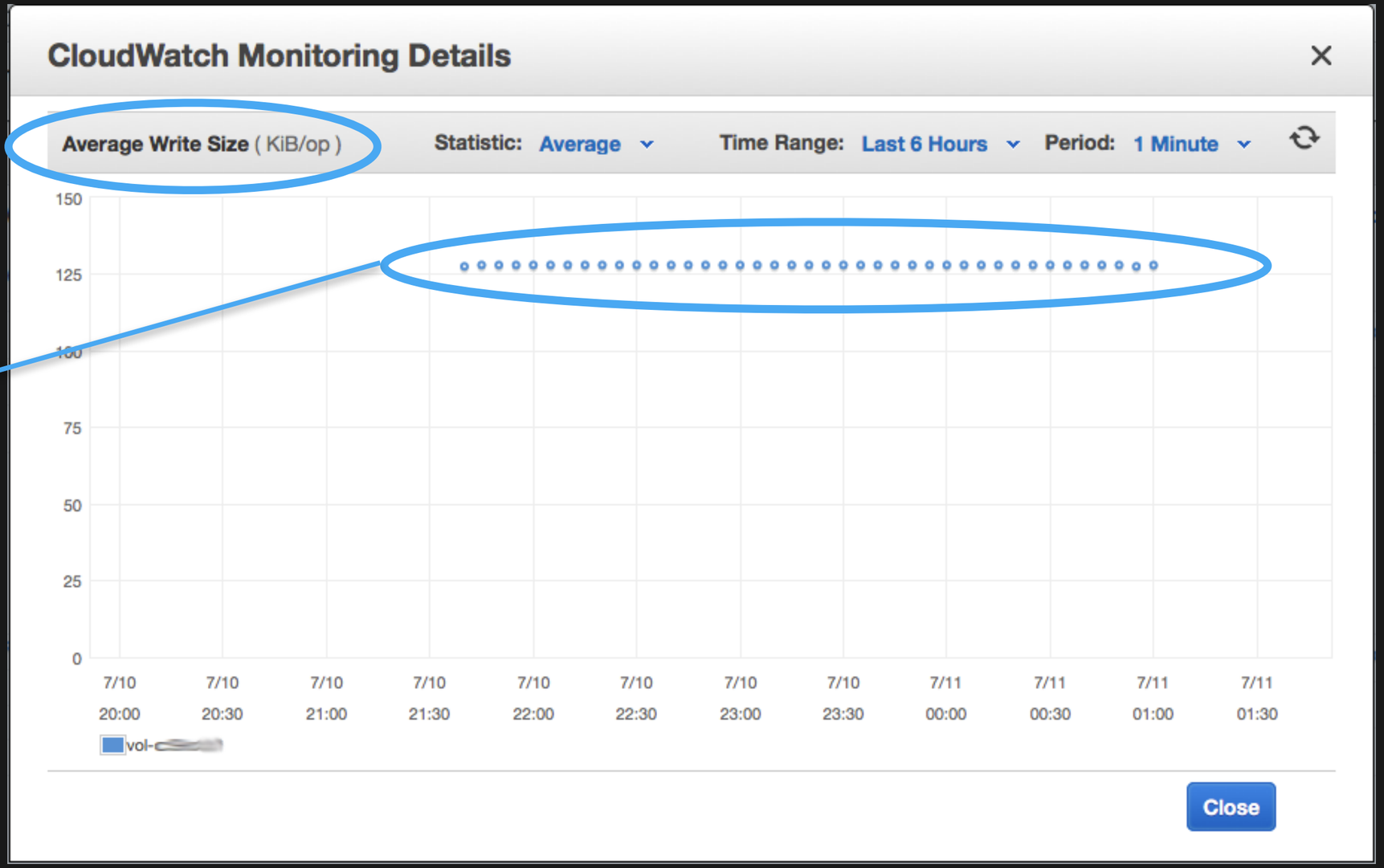
perfmon for Windows



Verify ST1 & SC1 workloads

Amazon CloudWatch console

128 KiB

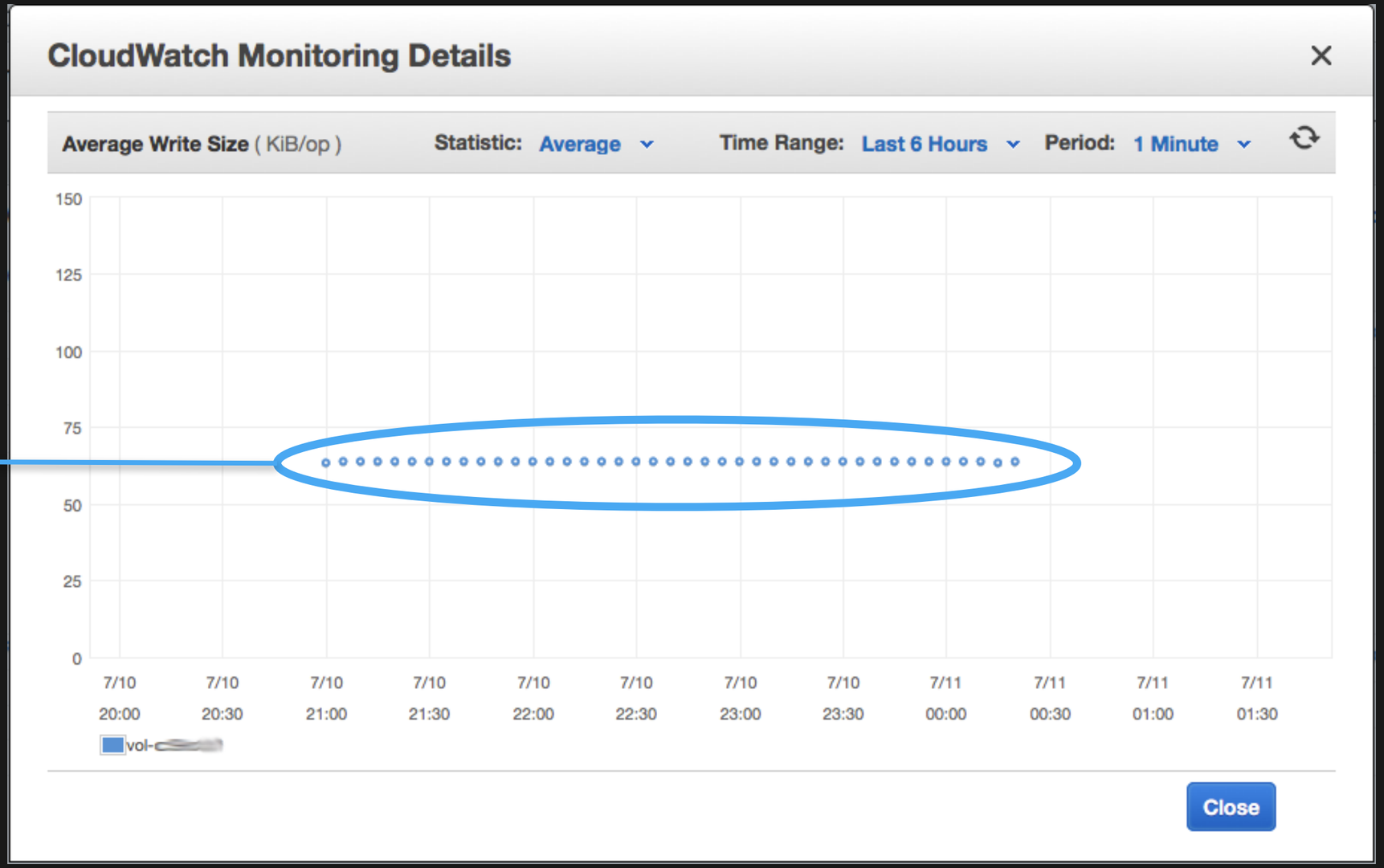


Verify ST1 & SC1 workloads

Amazon CloudWatch console

Under 64 KiB?

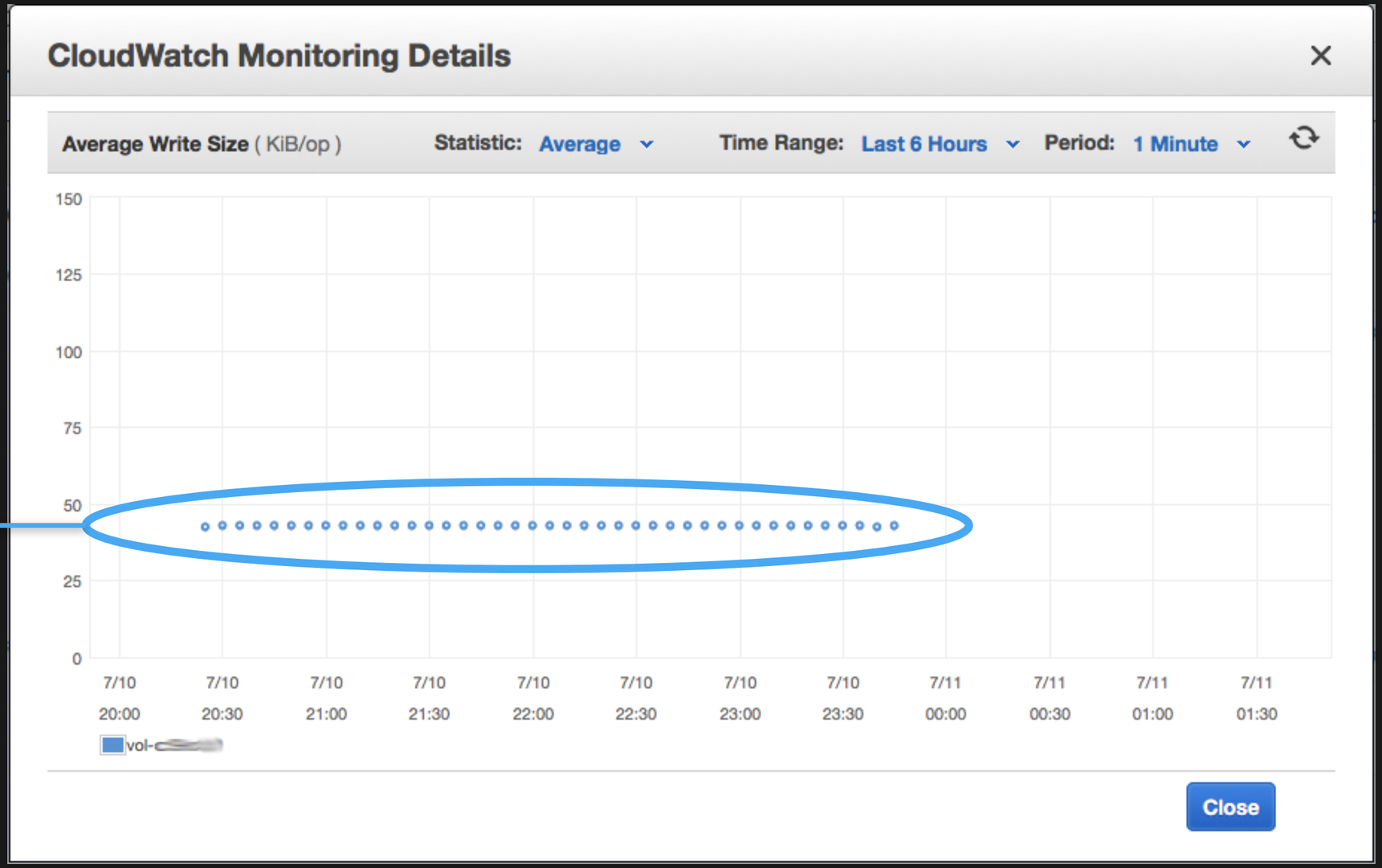
Small or random IOPS likely



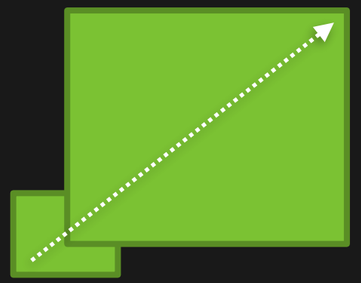
Verify ST1 & SC1 workloads

Amazon CloudWatch console

Stuck around 44 KiB?



ST1 & SC1: Linux performance tuning



Increase maximum request size

- Recommended ST1 and SC1 on a 4.2+ Linux kernel
- Memory allocated per device
- Default is 32, max for EC2 is 256
- OS boot command line configuration

For example with GRUB's `/boot/grub/menu.lst` configuration:

```
kernel /boot/vmlinuz-4.4.5-15.26.amzn1.x86_64 root=LABEL=/ console=ttyS0 xen_blkfront.max=256
```

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

ST1 & SC1: Linux performance tuning

Increase read-ahead buffer:

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

For example:

```
$ sudo blockdev –setra 2048 /dev/xvdf
```

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

IOPS vs. throughput

Example:
io1 volume
20,000 PIOPS

10,000 IOPS
32 KiB
320 MB/s

20,000 IOPS
16 KiB
320 MB/s

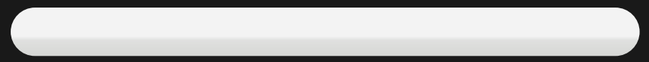
1,250 IOPS
256 KiB
320 MB/s

I/O request size

Performance: EBS-optimized

c4.large

Dedicated to EBS

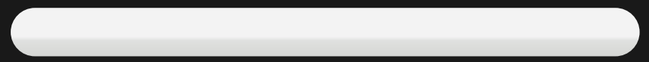


500 Mbps ~ 62.5 MB/s
4,000 16K IOPS

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

c4.2xlarge

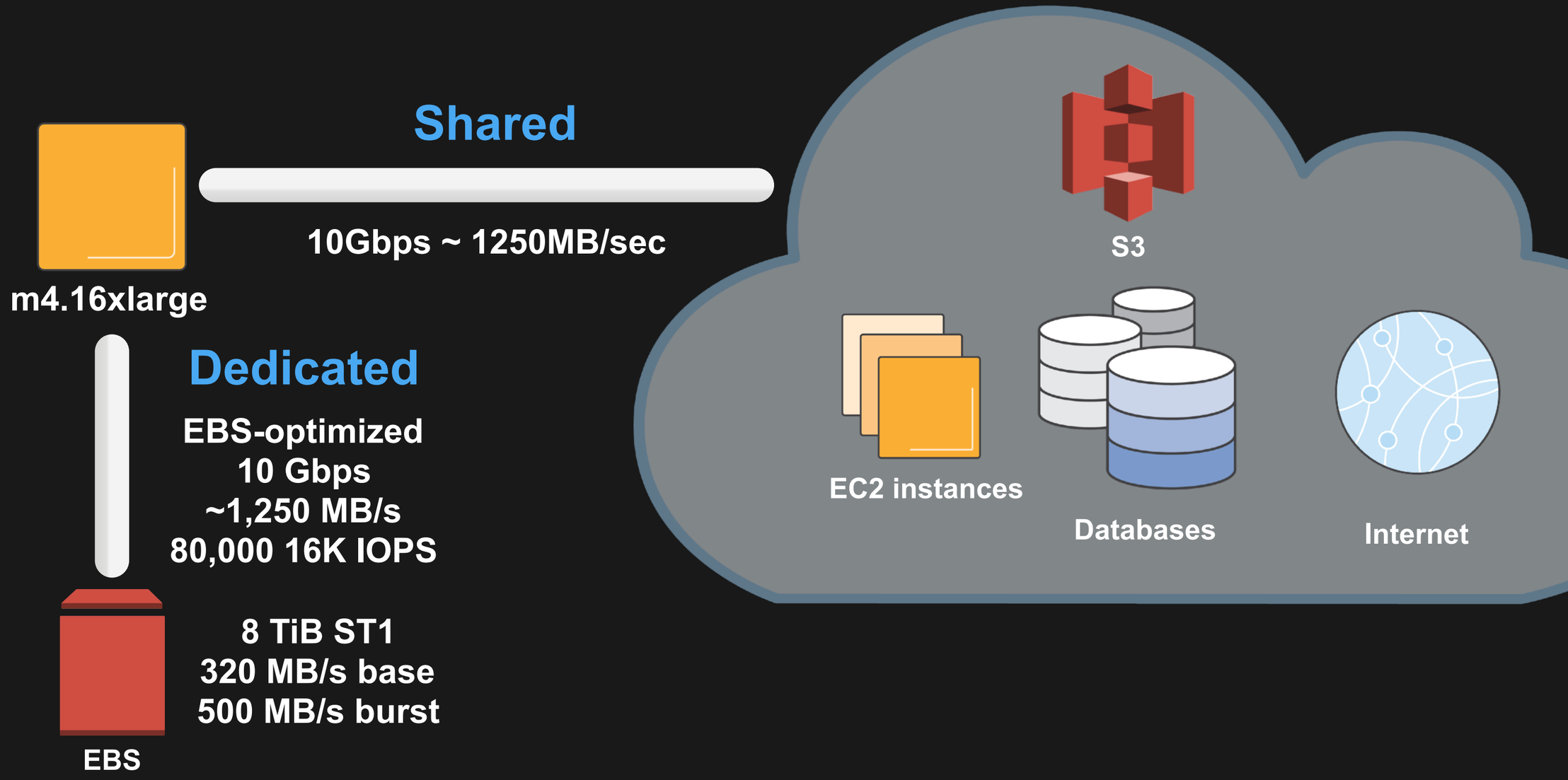
Dedicated to EBS



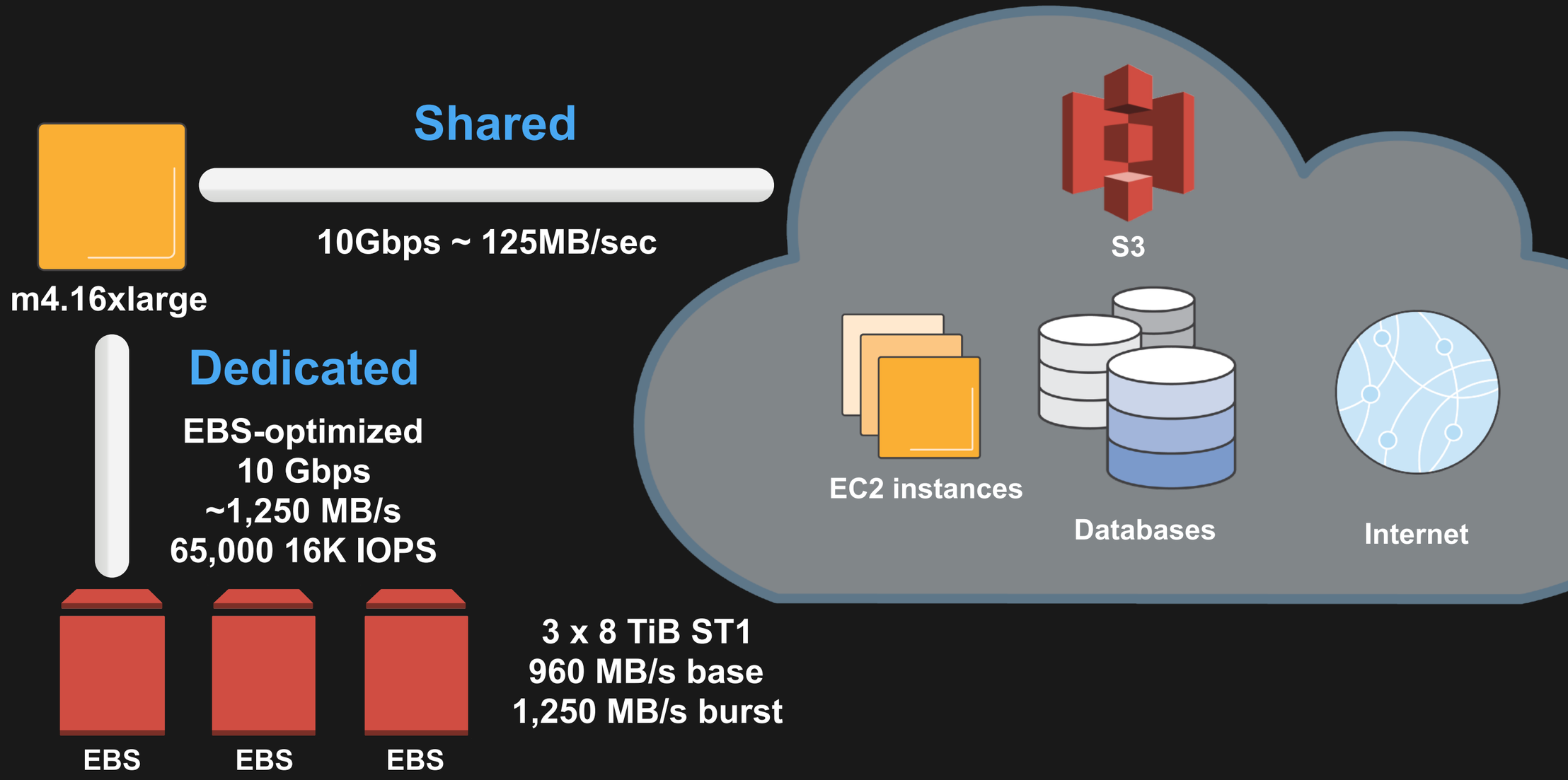
1 Gbps ~ 125 MB/s
8,000 16K IOPS

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

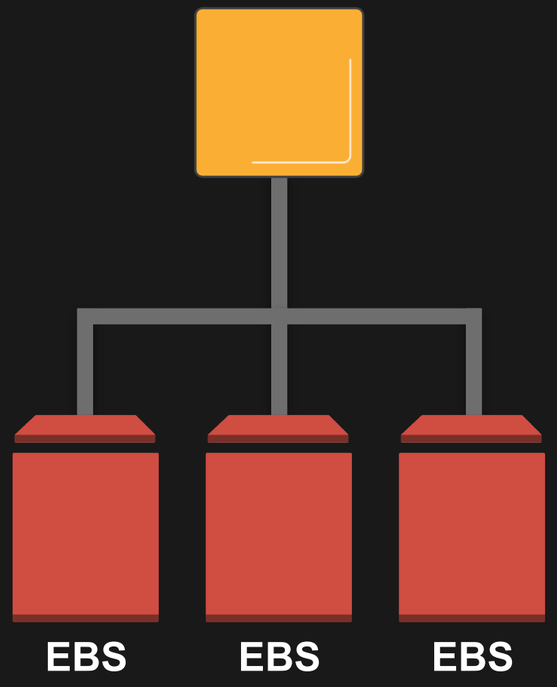
Performance: throughput workload



Performance: throughput workload



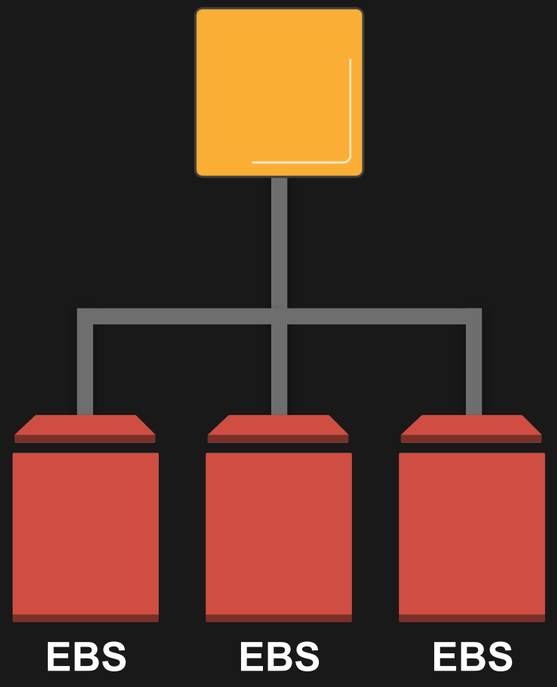
Best practice: RAID



When to RAID?

- Storage requirement > 16 TiB
- Throughput requirement > 500 MB/s
- IOPS requirement > 20,000 @ 16K

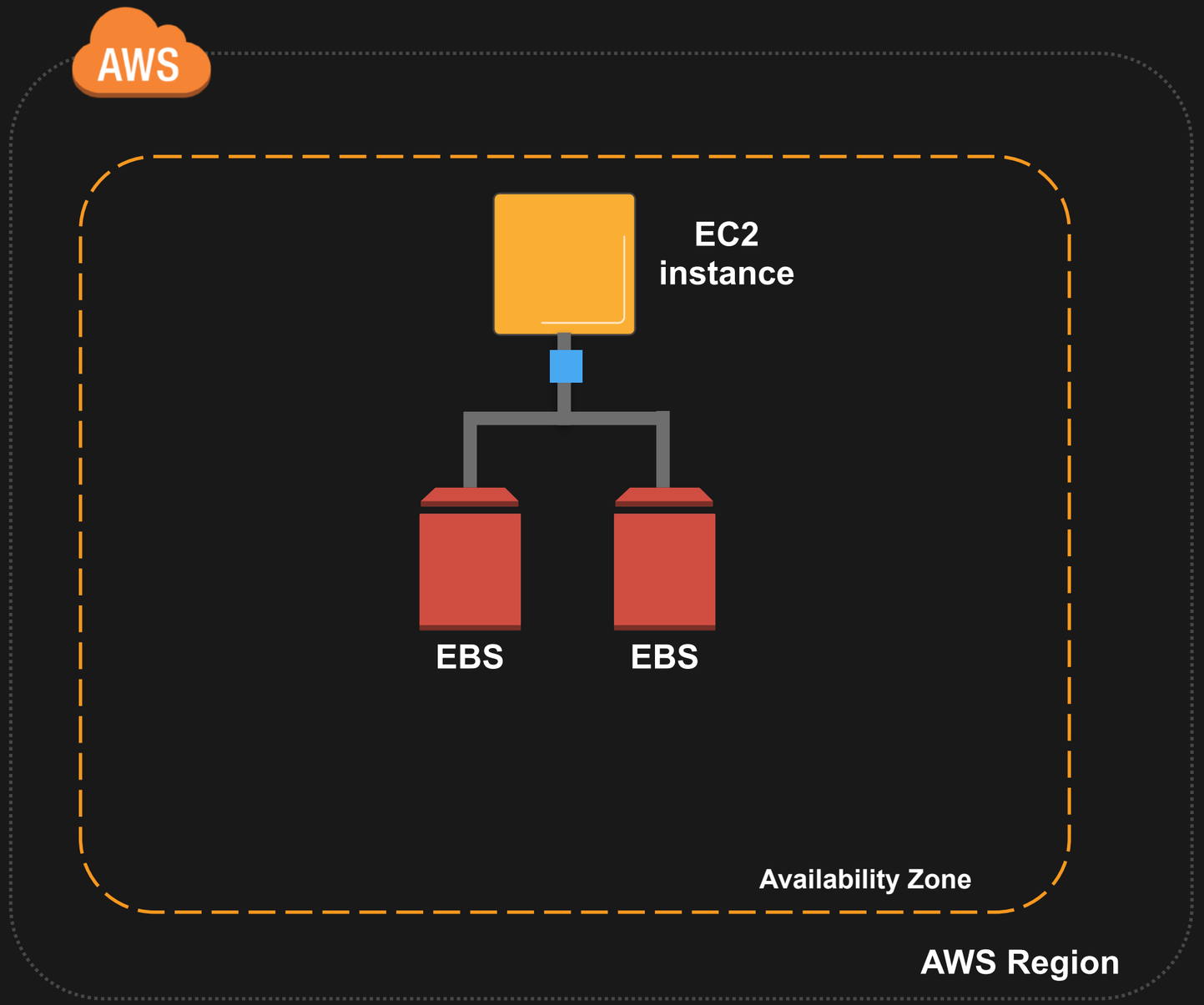
Best practice: RAID



Avoid RAID for redundancy

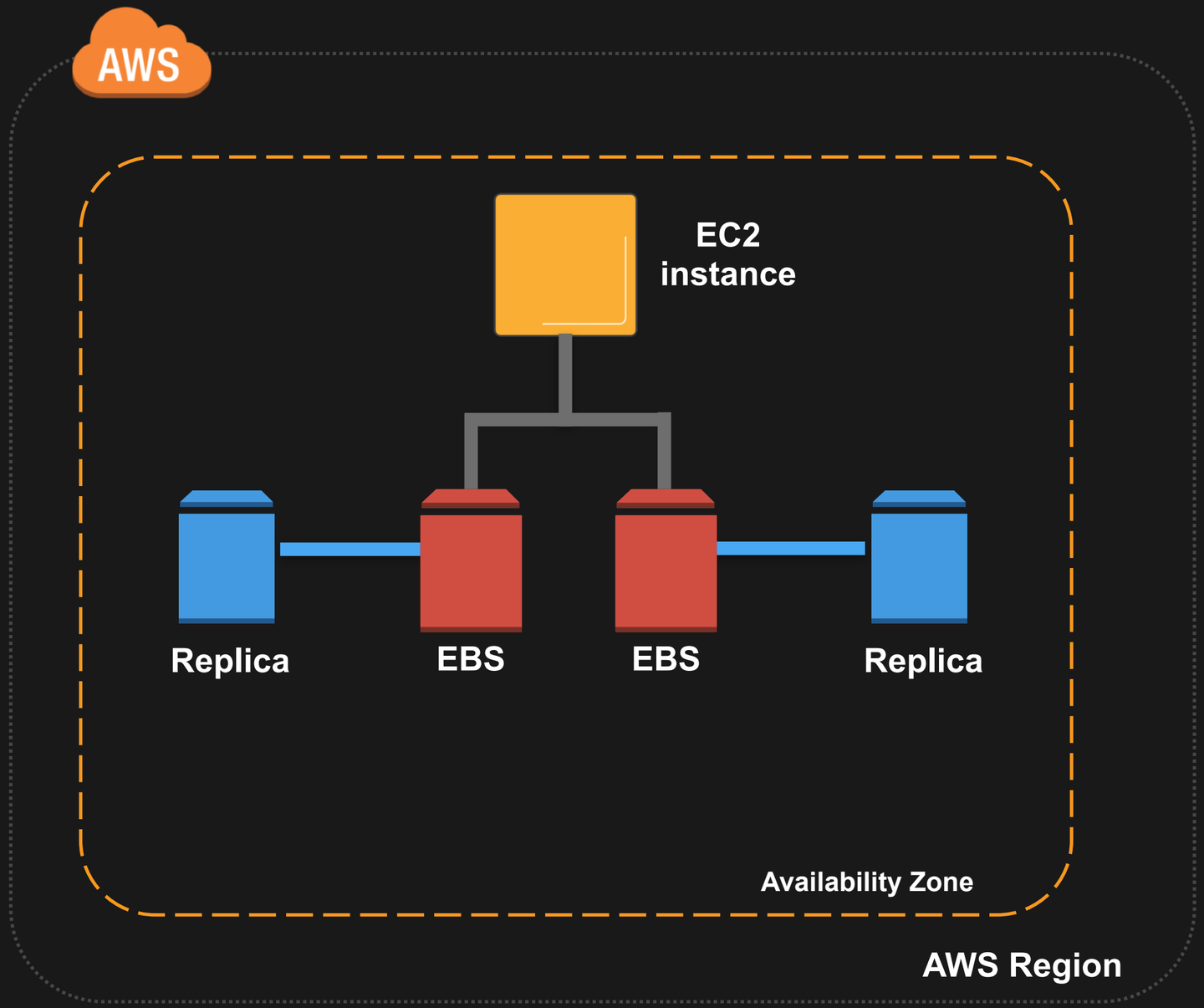
- EBS data is already replicated
- RAID1 halves available EBS bandwidth
- RAID5/6 loses 20–30% of usable I/O to parity

Best practice: RAID



RAID 0

Best practice: RAID



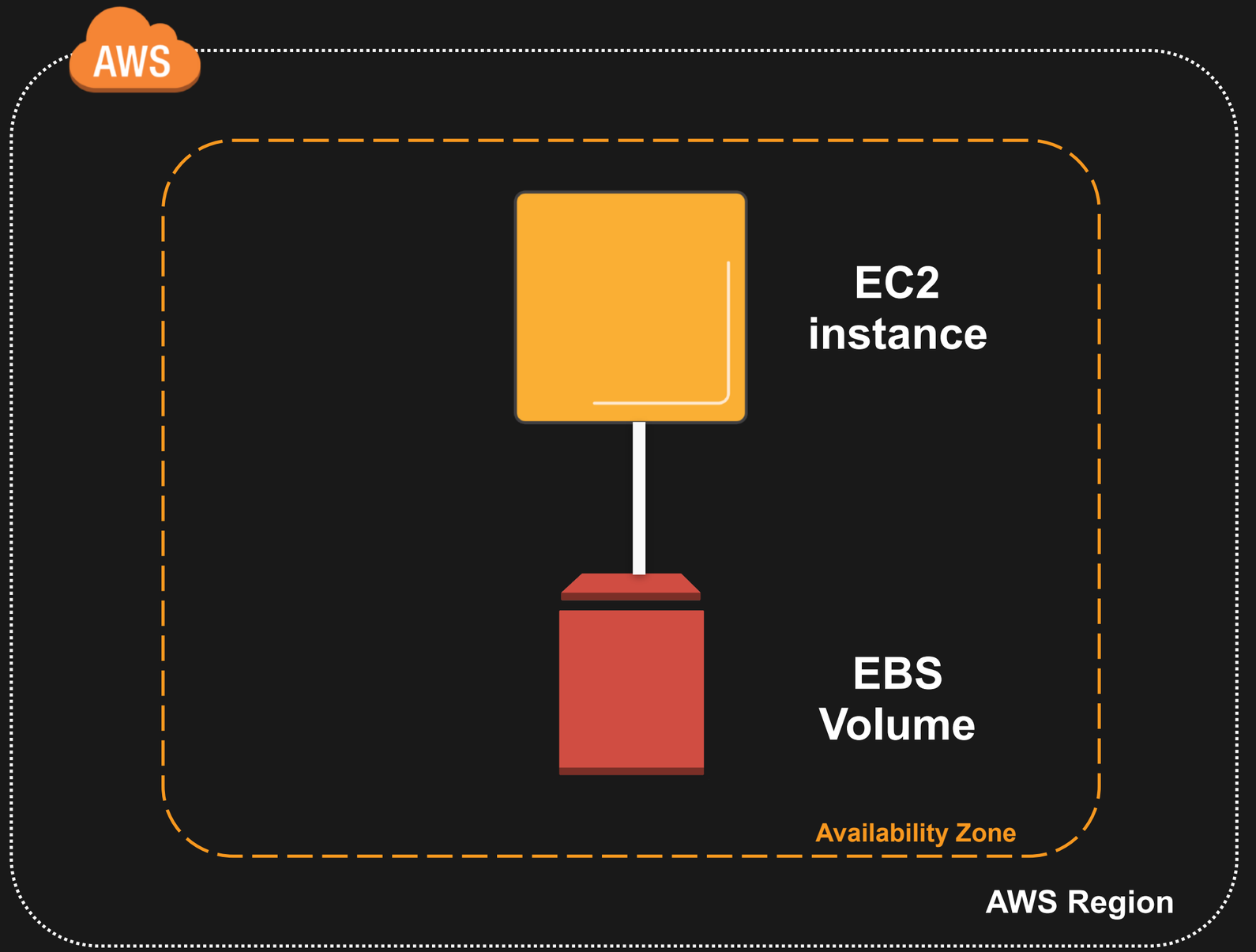
RAID 10

EBS deep dive

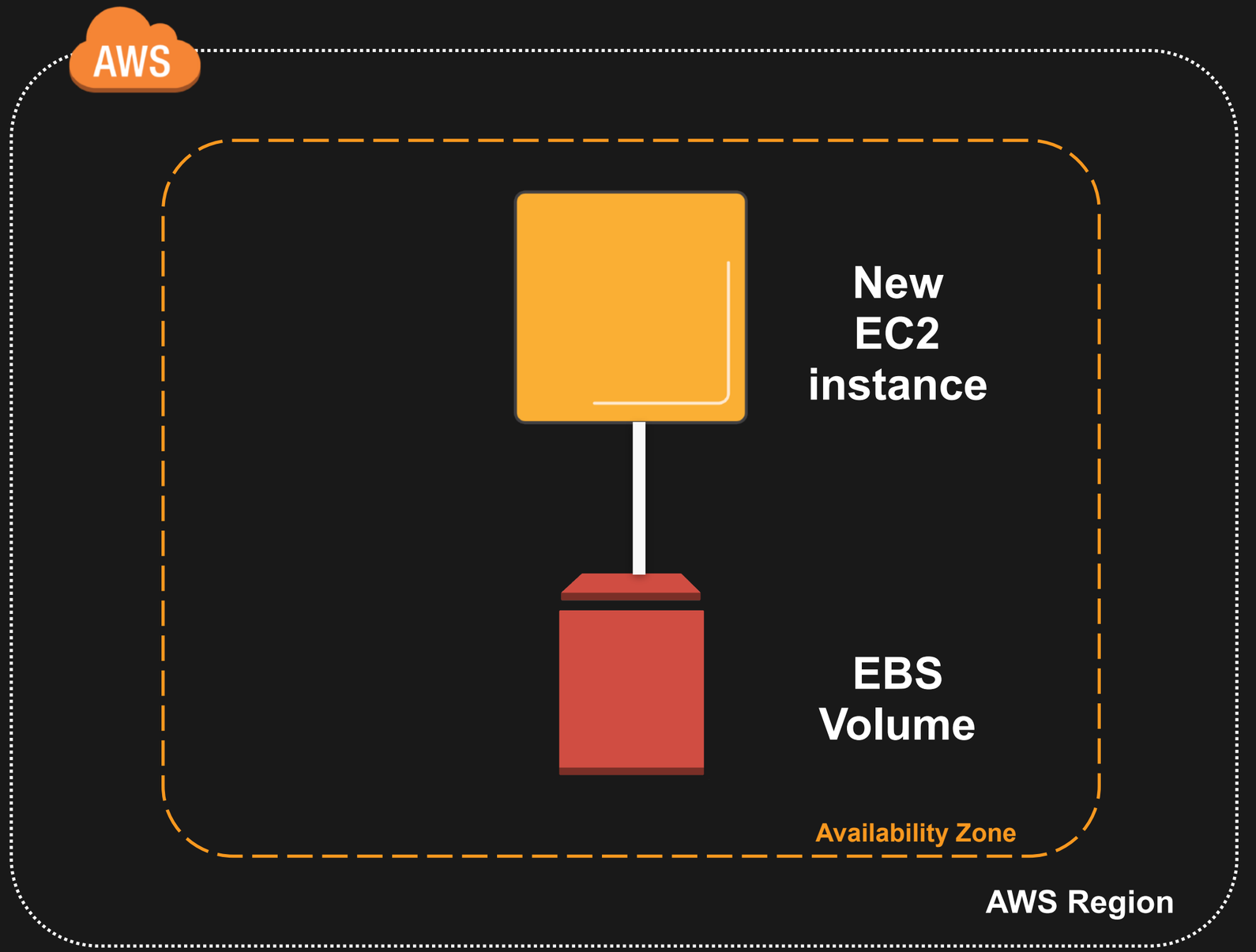
Performance

Reliability

What about EC2 instance failure?



What about EC2 instance failure?



EBS enables EC2 auto recovery

Amazon CloudWatch
per-instance metric alarm:

`StatusCheckFailed_System`

When alarm triggers?
RECOVER Instance

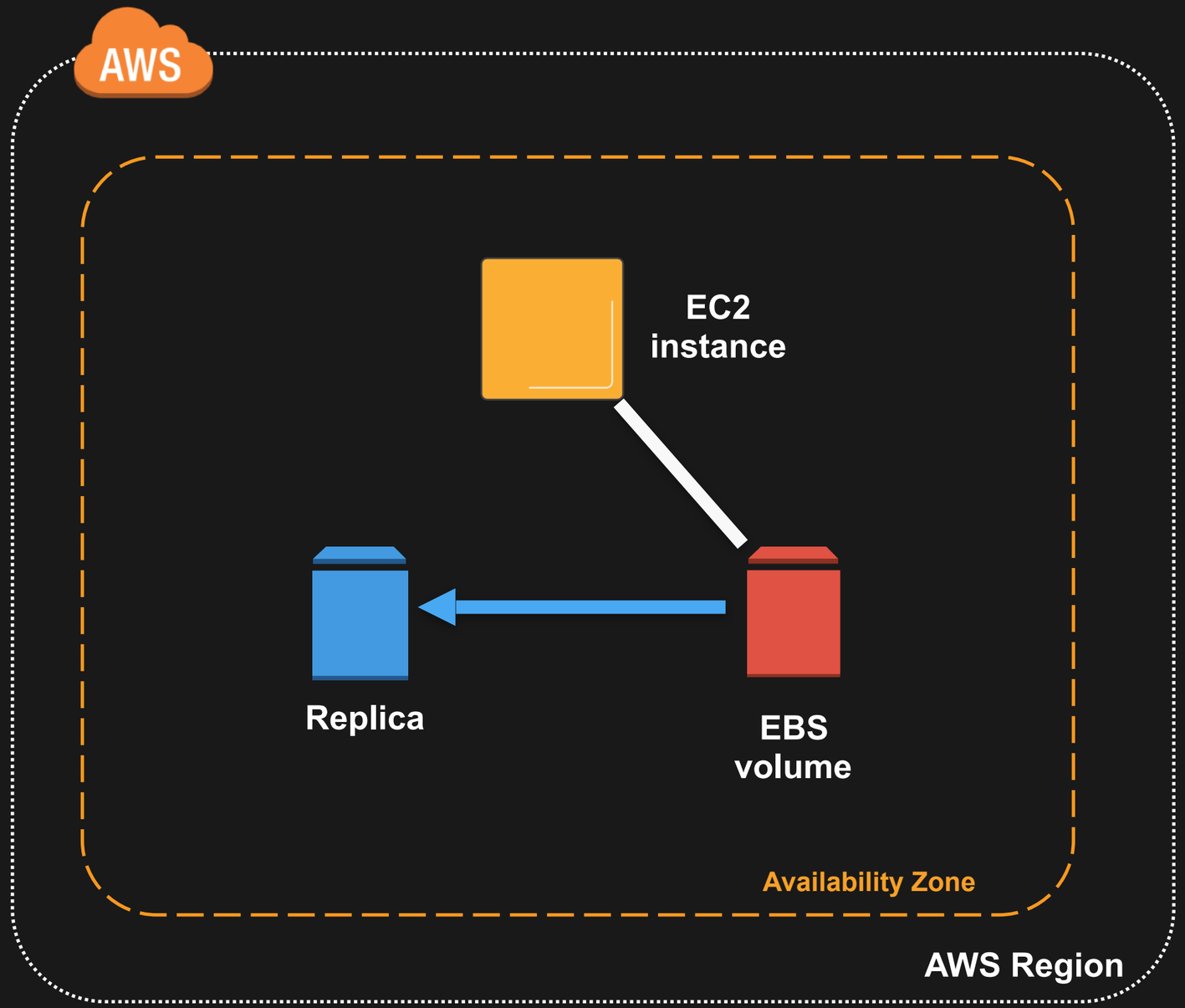
Instance retains:

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

** Supported on C3, C4, M3, M4, P2, R3, R4, T2, and X1 instance types with EBS-only storage*

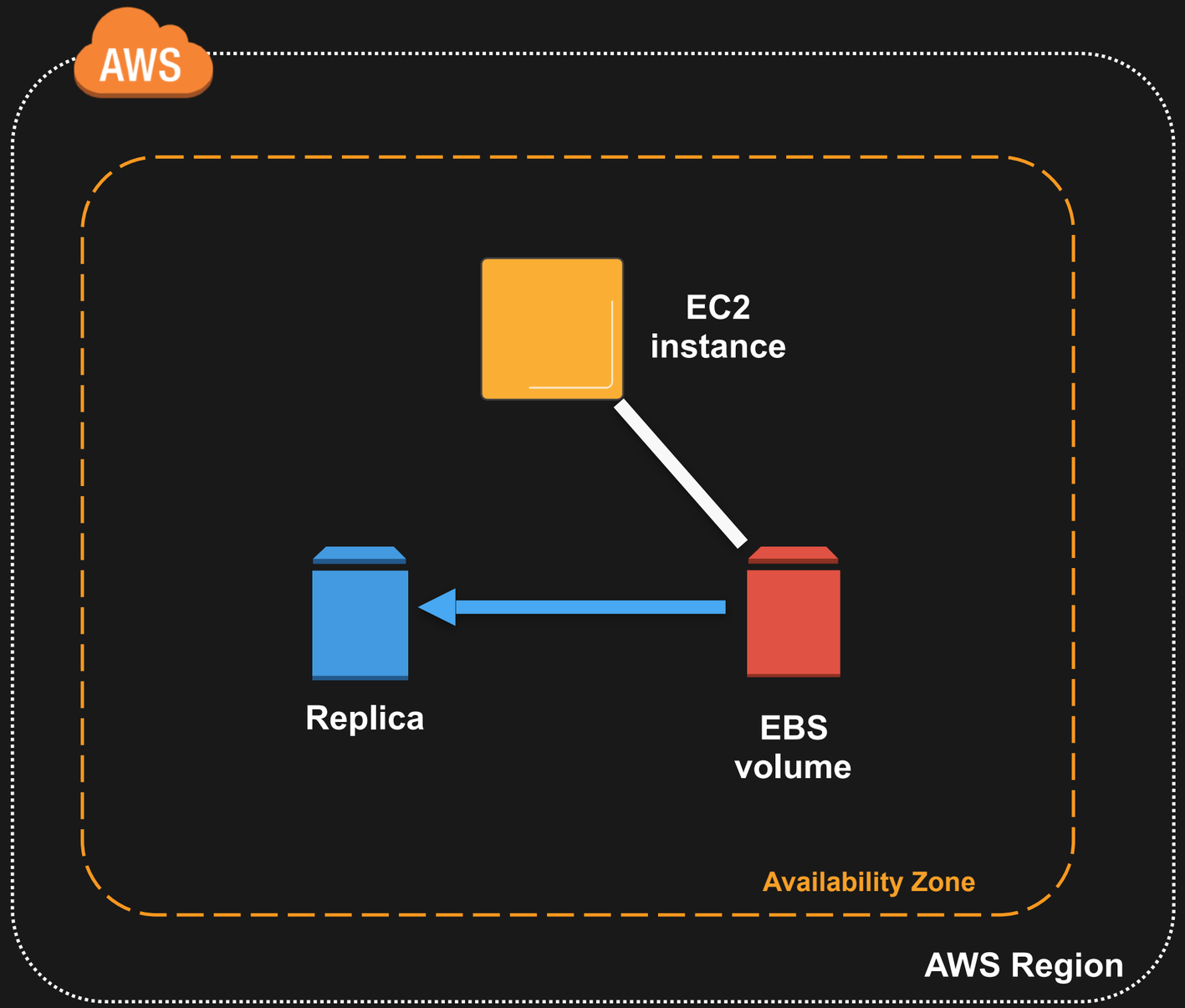
https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/ec2-instance-recover.html

What about EC2 instance termination?



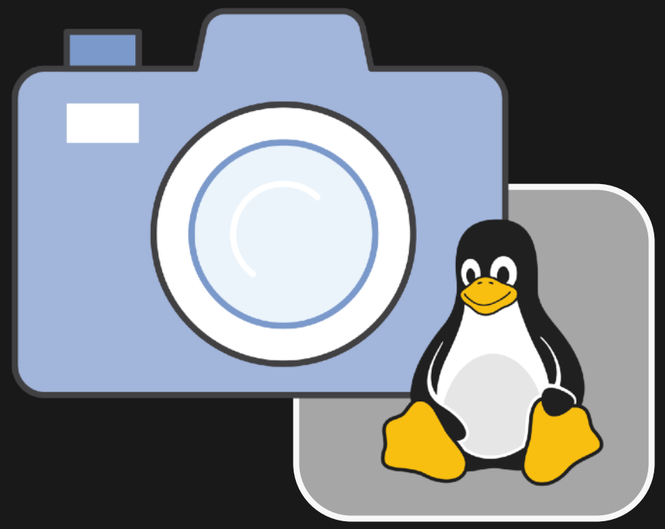
`DeleteOnTermination = False`

What about EC2 instance termination?



`DeleteOnTermination = True`

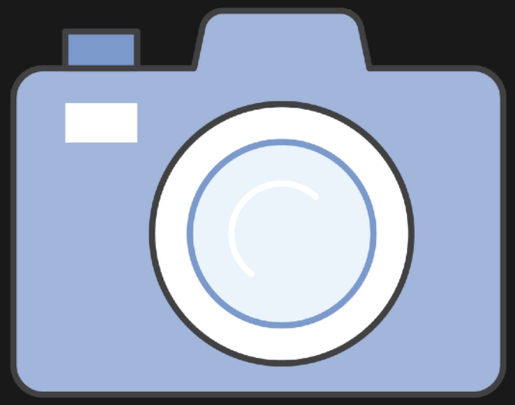
Best practice: taking snapshots from Linux



Quiesce I/O

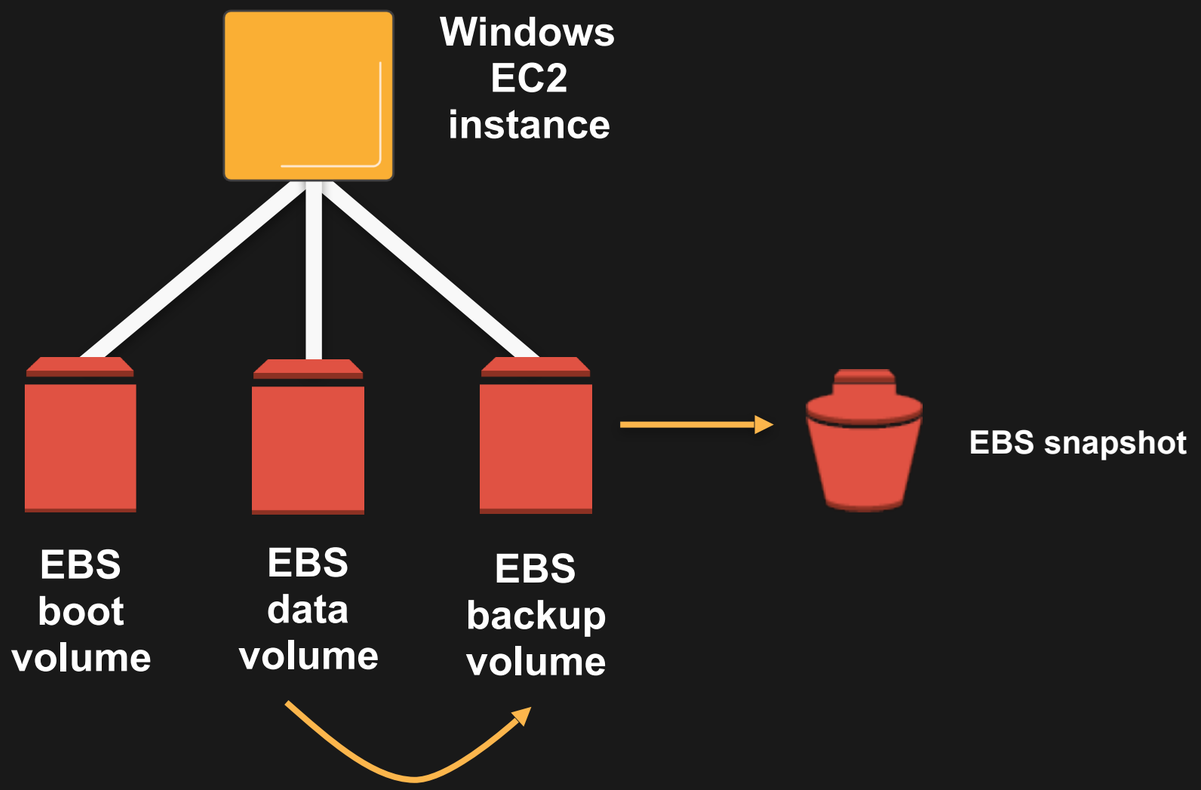
1. Database: `FLUSH` and `LOCK` tables
2. File system: `sync` and `fsfreeze`
3. EBS: snapshot all volumes
4. When CreateSnapshot API returns success, it is safe to resume

Best practice: taking snapshots from Windows



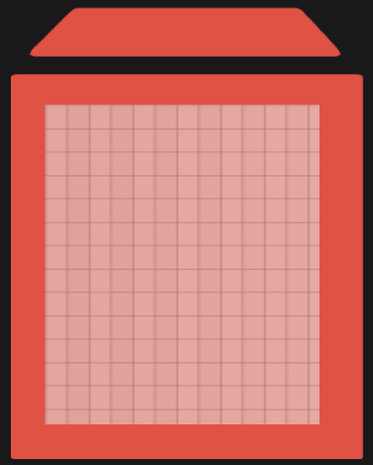
1. `sync` equivalent available
2. Use Volume Shadow Copy Service-(VSS) aware utilities for backups
3. EBS: backups on dedicated volume for snapshots

Best practice: taking snapshots from Windows



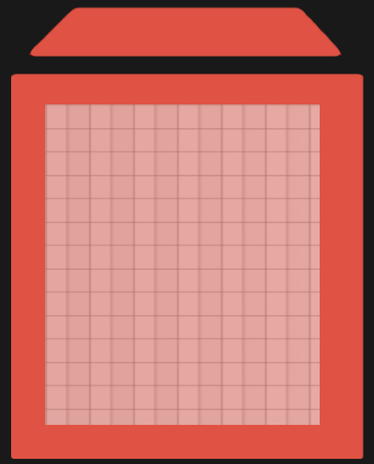
Windows Server Backup

EBS volume initialization



New EBS volume?

- Attach and it’s ready to go



New EBS volume from snapshot?

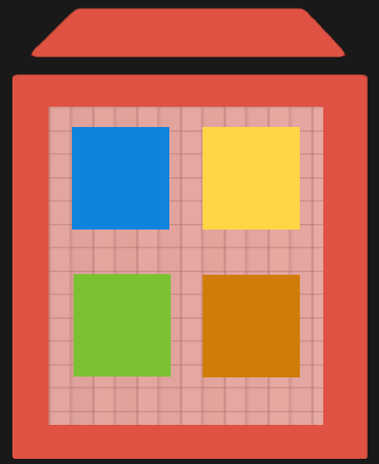
- Initialize for best performance
- Random read across volume

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

Best practice: automate snapshots

Key ingredients:

AWS Lambda

Amazon EC2
Run command

Tagging

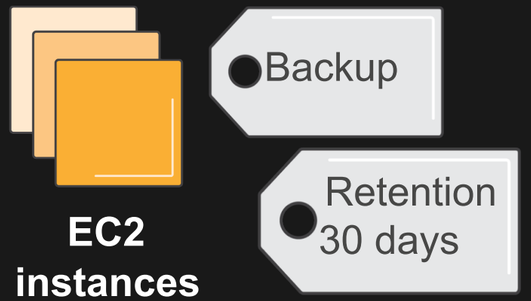
https://aws.amazon.com/ec2/run-command/

Best practice: automate snapshots

EC2 instances

Tagging

Best practice: automate snapshots



EC2 instances

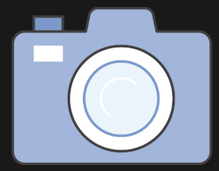
Lambda scheduled event: daily snapshots

1. 

Search for instances tagged “Backup”

2. 

EC2 Run commands to quiesce file system

3. 

Snapshot attached volumes

4. 

Tag snapshots with expire date

Best practice: automate snapshot expiration

EBS snapshots

Lambda scheduled event: daily expire

1.

Search for snapshots tagged to “Expire On” today

2.

Delete expired snapshots

Prototype for EBS Snapshot Custodian:

https://github.com/dR0ski/lambda-ebs-snapshot-custodian

EBS best practices

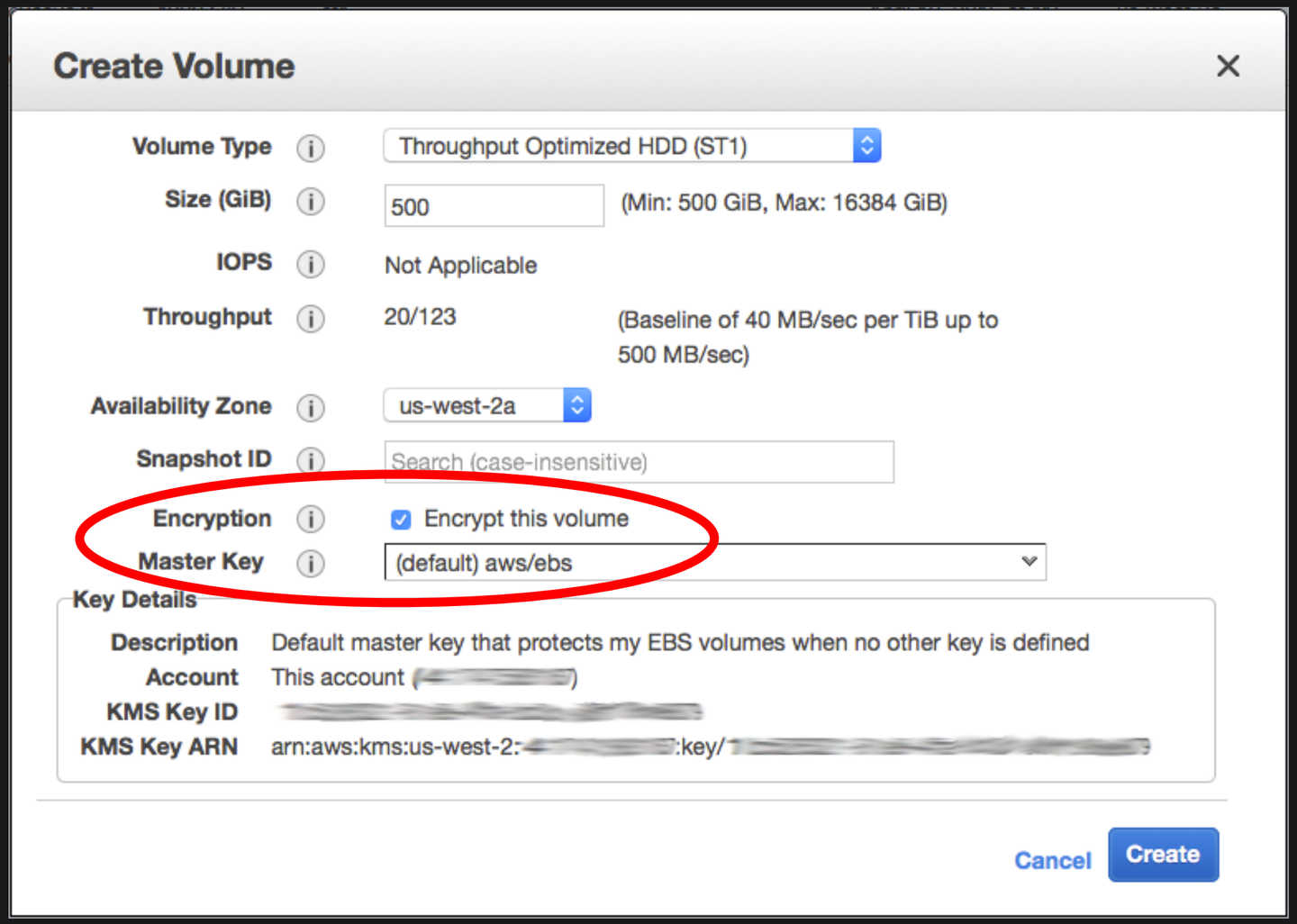
Performance

Reliability

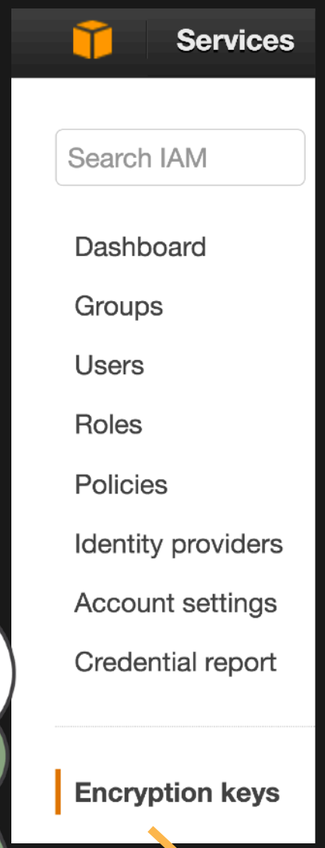
Security

Best practice: encryption

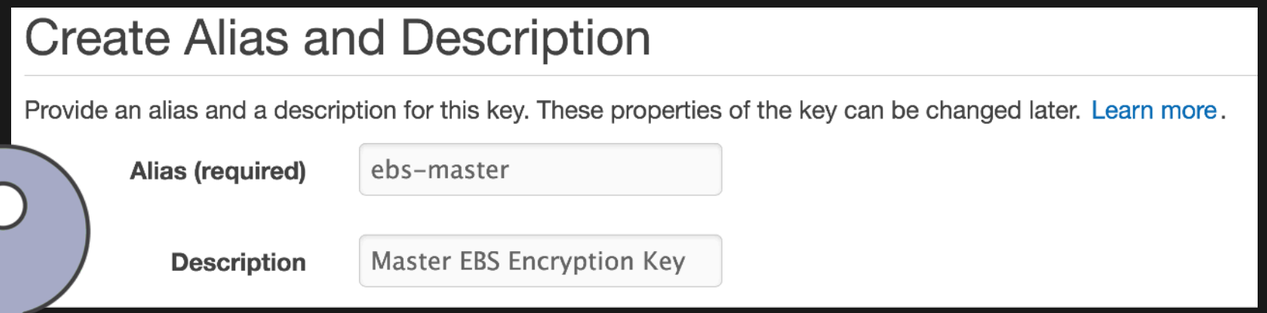
EBS encryption: data volumes



Best practice: encryption



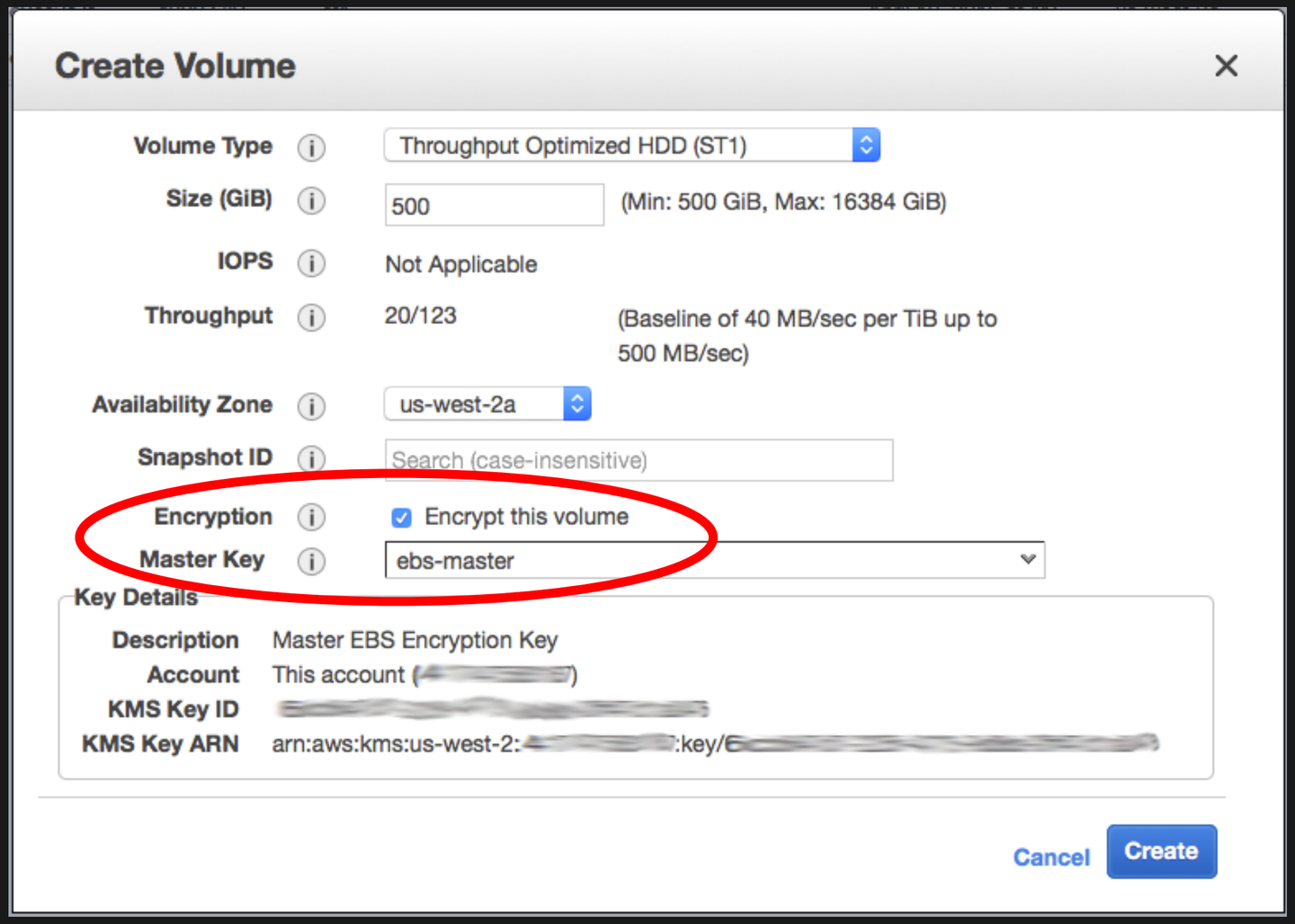
Create a new AWS KMS master key for EBS



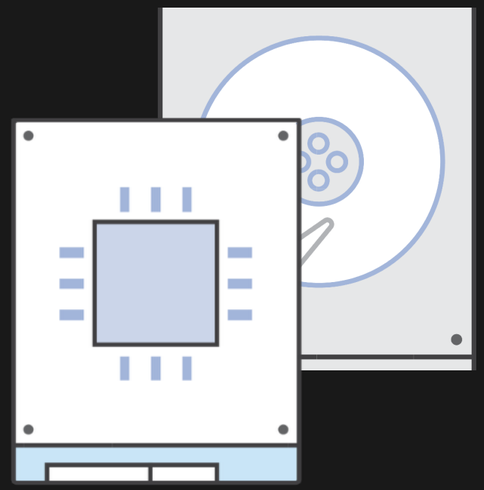
- Define key rotation policy
- Enable AWS CloudTrail auditing
- Control who can use key
- Control who can administer key

Best practice: encryption

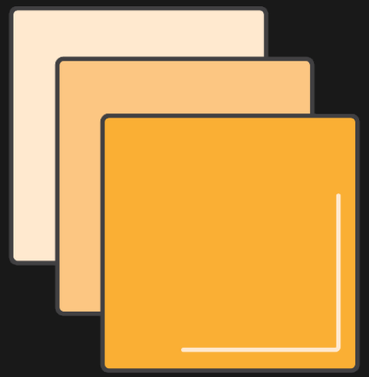
EBS encryption: data volumes



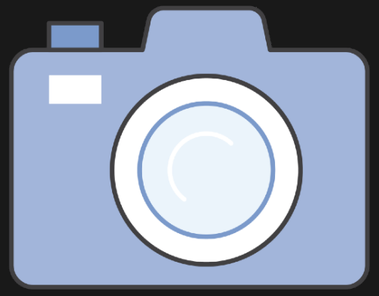
Summary



Select the right volume for your workload



Select the right instance for your workload



Take snapshots, tag snapshots

Use encryption if you need it

AWS

SUMMIT

Thank You

amazon web services