

About Volumes

- EN
- FR

BSU volumes are virtual hard drives that you can attach to an FCU instance created in the same Availability Zone to store data. A volume is defined by its size and IOPS capacity.

The following topics are discussed:

- [Volumes and Instances](#)
- [Volumes Attachment and Device Names](#)
 - [Linux Instances](#)
 - [Windows Instances](#)
- [Data Persistence](#)
- [Volume Types and IOPS](#)
 - [*IOPS Credit and Burst Possibility for Performance Volumes](#)

Related Pages

- [About Instances](#)
- [Working with Snapshots](#)
- [Resource Isolation](#)

Volumes and Instances

When you create an instance, a volume is automatically created and attached to it to store all the data related to the operating system (OS). This volume is called a root device and appears in the **Volumes** tab of Cockpit. Root devices have a default size of 10 GiB for Linux instances (8 GiB for CentOS 7 instances), and 50 GiB for Windows instances. You can store data on a root device. However, depending on the block device mapping configuration of the OUTSCALE Machine Image (OMI) used to create the instance, this root device may be deleted when terminating this instance.

To add storage capacity to your instances or to separate your data from the OS files, you can create volumes with the storage capacity of your choice (within the limit of the quotas allocated to your account).

Volumes are placed in an Availability Zone (AZ). If you do not specify any AZ, the AZ A is used by default.

You can attach volumes to any instance created within the same AZ, in the public Cloud or in a Virtual Private Cloud (VPC). When attached to an instance, a volume works exactly like a physical hard drive and can be used in the same way, after initializing this volume as you would have done with a physical one. The instance can then interact with the volume and access your data. When you stop an instance, the attached volumes remain attached to it and can therefore be accessed after the instance is restarted.

A volume persists independently from the instance lifecycle: it can be easily detached from the instance it was previously attached to, and then attached to another instance. When terminating an instance, volumes attached to it are either deleted or detached, depending on the block device mapping of the instance. For more information, see [Defining Block Device Mappings](#)

You can attach up to 25 volumes to a same instance, alongside its root device. However, a volume can be attached to only one instance at the same time.

A volume can be in these different states:

- **creating**: The creation process of the volume is in progress.
- **available**: The volume is created but not attached to an instance. A volume is in the `available` state even if there is data stored on it.
- **in-use**: The volume is attached to an instance. A volume is in the `in-use` state even if the instance it is attached to is stopped.
- **deleting**: The deletion process of the volume is in progress.
- **error**: The creation of the volume failed.



To identify your resources more easily, you can add tags to them. For more information, see [Tagging Your Resources](#).

Volumes Attachment and Device Names

You must always attach the root device of an instance using the `/dev/sda` device name and mount it as `/dev/sda1` (bootdisk), even though it appears as `/dev/vda1` in the instance. For any other volume you attach to an instance, you must use a device name in the `/dev/xvdX` format.



Non official OMI must support the PCI Hotplug. Otherwise, the volume is not visible when attached to an instance. **You need to reboot the instance and add the following modules to the /etc/modules files:**

- pci-hotplug
- acpihp

Linux Instances

For Linux instances, the /dev/xvdX device name you choose for your volumes is transformed by Linux in a /dev/sdY device name, following the order in which you attach your volumes to the instance.

Two OUTSCALE packages for CentOS and Ubuntu instances enable to link the /dev/xvdX device name to the corresponding /dev/sdY Linux one. You can see these links when listing the devices from the Linux instance in the /dev directory:

Devices Listing Sample

```
$ ls -l /dev/[svx][vd]*
brw-rw---- 1 root disk 8, 0 30 mai 14:40 /dev/sda
brw-rw---- 1 root disk 8, 16 30 mai 14:41 /dev/sdb
brw-rw---- 1 root disk 253, 0 30 mai 14:16 /dev/vda
brw-rw---- 1 root disk 253, 1 30 mai 14:16 /dev/vda1
lrwxrwxrwx 1 root root 3 30 mai 14:41 /dev/xvdd -> sdb
lrwxrwxrwx 1 root root 3 30 mai 14:40 /dev/xvdh -> sda
```



The /etc/fstab file of your Linux instances must be configured with /dev/xvdX device names.

Windows Instances

Volumes attached to Windows instances are numbered according to the order in which you attach them to the instance, independently from the /dev/xvdX device name you choose when attaching them. The root device is numbered zero, the first volume you attach is numbered one, and so on. If you detach a volume, its associated number is released and reassigned to the next volume you attach to the instance.

3DS OUTSCALE provides the BSU Utility application to get the corresponding Windows disk number for each volume you attach to an instance. Windows official OMI released after June 06, 2016 contain this application, in C:\Windows\Outscale\. If you launched instances using OMI released before this date, you can download the application by clicking the following link: [bsu_utility.exe](#).

To get the corresponding Windows disk number for a volume attached to the instance, run the application from the instance in the directory where it is stored using the following command:

```
bsu_utility.exe DEVICE_NAME
```



The device name corresponds to the one you specified when attaching the volume to the instance (for example, /dev/xvdB).

This command returns the Windows disk number corresponding to the device name you specified, as in the following example:

BSU Utility Sample

```
bsu_utility.exe /dev/xvdh
1
```

3DS OUTSCALE also provides the following Powershell command that does the same action as BSU Utility:

```

BSU Utility Powershell Script

Get-WmiObject -class "Win32_DiskDrive" | Select-Object Name,SerialNumber |
Where-Object {$_.SerialNumber -eq "DEVICE_NAME"}
    
```

Data Persistence

Data stored on a volume are persistent as long as the volume exists and can be accessed when the volume is attached to an instance. If you delete a volume, data stored on it are deleted too.

When detaching a volume from an instance, all the data remain stored on it and this volume becomes available again. This volume can then be attached to another instance, enabling you to access and manage your data again.

You can backup your data stored on a volume using snapshots, that can be copied to another Region. Snapshots enable you to retrieve your data in case of failure of a volume or after you deleted a volume. For more information about snapshots and how to use them, see [Working with Snapshots](#).

Volume Types and IOPS

Volume performance is measured in IOPS (input output per second), that is, the number of read and write operations that the volume can perform in one second.

You can choose between three volume types, depending on the performance you need:

- Magnetic, with a fixed number of IOPS, no matter its size. Magnetic volumes also have a small burst possibility in case of a temporary increase of your needs.
- Performance, with a burst possibility for volumes below 1 TiB.
- Enterprise, enabling you to create a volume with the number of IOPS you need. Enterprise volumes are SSD volumes, and therefore have a reduced latency between the time the operation order is sent to the volume and the time the volume executes it.

The following table presents basic use cases for each volume type and describes their performance characteristics:

	Magnetic	Performance (SSD)	Enterprise (SSD)
API and AWS CLI volume name	standard	gp2	io1
Use cases	<ul style="list-style-type: none"> • Cold workloads where you do not need to access data frequently • Cases in which the lowest storage cost highly matters 	<ul style="list-style-type: none"> • Most workloads that require moderate performance with moderate costs • Applications that require high performance for a short period of time (for example, starting a file system) 	<ul style="list-style-type: none"> • Workloads where you must access data frequently (for example, a database) • Critical business applications that can be blocked by a low performance when accessing data stored on the volume
Volume size	1 GiB - 14.55 TiB (14901 GiB)	1 GiB - 14.55 TiB (14901 GiB)	4 GiB - 14.55 TiB (14901 GiB)
IOPS performance	<ul style="list-style-type: none"> • 250 read operations and 150 write operations in average • Burst possibility up to 450 IOPS 	<ul style="list-style-type: none"> • Between 1 GiB and 1 TiB: Baseline performance of 3 IOPS per gibibyte, with burst possibility* • Between 1 TiB and 3.33 TiB: 3 IOPS per gibibyte • Between 3.34 TiB and 14.55 TiB: 10000 IOPS 	Maximum performance ratio of 300 IOPS per gibibyte
I/O size	4 KiB	4 KiB	4 KiB

Minimum IOPS /volume	50 read operations and 250 write operations in average	100	100
Maximum IOPS /volume	50 read operations and 250 write operations in average	10000	13000
Maximum IOPS /instance	36000	36000	36000
Throughput performance	/	/	/
Maximum throughput /volume	40 MiB/s	160 MiB/s	200 MiB/s
Maximum throughput /instance	600 MiB/s	600 MiB/s	600 MiB/s

*IOPS Credit and Burst Possibility for Performance Volumes

The IOPS performance of Performance (gp2) volumes depends on their size. Between 1 GiB and 1 TiB, their performance is ruled by the principle of a leaky bucket, that is described below.

Each Performance volume between 1 GiB and 1 TiB receives an initial IOPS credit balance of 5.4 million IOPS, which enables the volume to burst up to 3000 IOPS per second for a maximum duration of 30 minutes. Every second, your IOPS credit balance is refilled with 3 IOPS per gibibyte. The more IOPS credit you have for the volume, the longer it can burst beyond its baseline performance of 3 IOPS per gibibyte.



Burst possibility and IOPS credits are only available for Performance volumes with a size below 1 TiB, as the baseline performance for volume sizes above 1 TiB exceeds the maximum burst performance of 3000 IOPS per second.

The burst duration depends on the following three elements:

- IOPS credit balance (5.4 million IOPS maximum)
- Burst performance (3000 IOPS maximum)
- Volume size

To calculate your maximum burst duration in seconds, use the following equation:

$$\text{Burst duration} = \text{IOPS credit balance} / (\text{Burst performance} - 3 \times \text{Volume size})$$

When your IOPS credit balance is empty, the volume cannot burst anymore. Its performance remains at its baseline performance level of 3 IOPS per gibibyte, until the IOPS demand drops below this baseline performance level and that IOPS credits you do not use are added to your IOPS credit balance.

AWS™ and **Amazon Web Services™** are trademarks of Amazon Technologies, Inc or its affiliates in the United States and/or other countries.

Windows® is a registered trademark of Microsoft Corporation in the United States and/or other countries.