# Seagate EXOS X for AWS Outposts Deployment Guide

AWS Outposts is a fully managed service that extends the AWS infrastructure, services, APIs, and tools to virtually any data center, colocation space, or on-premises facility for a truly consistent hybrid experience. AWS Outposts is ideal for workloads that require low latency access to on-premises systems or local data storage.

Seagate's Exos X Storage arrays offer block storage over the iSCSI protocol to EC2 instances running on AWS Outposts Rack and Servers. This is especially helpful for Outposts 1U and 2U Servers which only contain instance (boot) storage. The Exos X Arrays have the same controller modules whether installed in a 2U24 All-Flash or Hybrid Array or a 5U84 Hybrid or all HDD Array, with raw storage capacity over 2PB and is easily expandable to over 6PB. As such the configuration of the Array is identical except for the type and number of storage devices.

Seagate's Exos X Storage Arrays support traditional RAID levels 0,1,10,5,50,6 as well as Seagate's ADAPT distributed raid protection algorithm which provides greater durability and faster rebuilds than traditional RAID levels. Other features include host-side multipath, volume-based snapshots, and Windows VSS support, as well as replication and tiering. Most of these features are outside the scope of this guide but links are provided below to help with the deployment of these particular features.

Exos X Arrays with AWS Outposts provides a best of class hybrid-cloud storage experience; a cost efficient, massive-scale private block storage platform for applications that require profiles from high-performance All-flash to high-capacity HDDs and everything in-between.

A simple diagram of Seagate Exos X iSCSI Array with AWS Outposts is below.
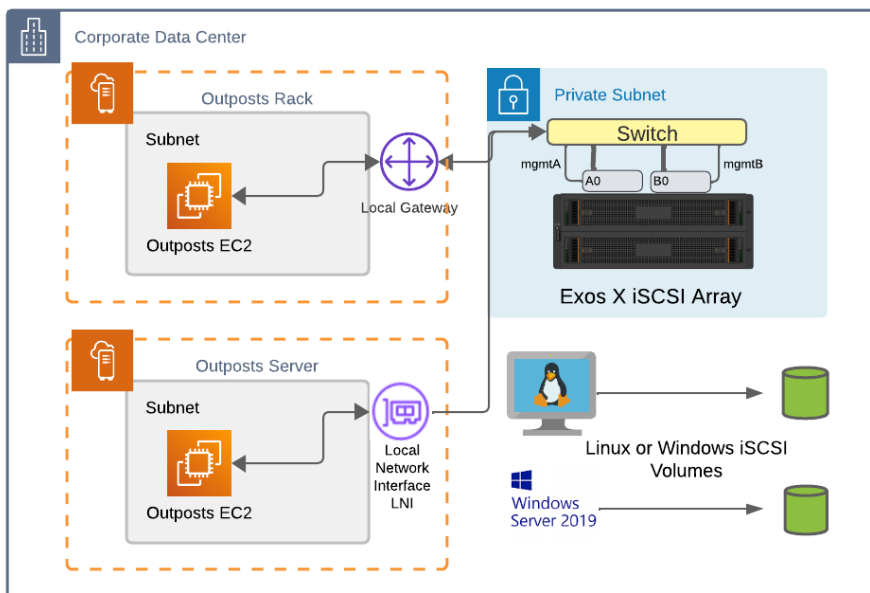
Requirements for deployment:

- An AWS Account with AWS Outposts configured and functional in the local DC.
- An Exos X iSCSI Storage Array reachable from the Outposts Server or Rack.
- EC2 instances on Outposts with routing to the on-premises network

Configuration of AWS Outposts to on-premises data sources can be found on this AWS BLOG post:

Deployment Steps:

The following steps provide the basic configuration for creating and mapping an iSCSI volume on Ubuntu Linux with an Exos X iSCSI Array deployed on-premises. Note that the initial Array configuration steps are assumed to be complete which includes:

1. Exos Management Controller hostnames or IP's (2, one for each controller)
2. One or more iSCSI service ports from each controller (up to 8 total)
3. Storage pools / disk groups created under each controller (dg01,dg02)

These steps can be found in the Getting Started Guide:
https://www.seagate.com/files/www-content/support-content/disk-arrays/exos-x-4006/_shared/files/83-00007887-10-01-c_exos_x_4006_getting_started.pdf

Additional configuration for Storage Pools and other servers can be found in the Storage Management Guide for 4006 Controllers:

https://www.seagate.com/content/dam/seagate/assets/support/disk-arrays/exos-x-4006-2u12/_shared/files/204468700-01-A_4006_SMG.pdf

The following example of CLI commands on the Exos X Storage Array show the steps to:

1. Create a 5TB volume using an existing disk-group (dg01)

2. Assign a host to a linux (ubuntu) initiator

3. Map the new volume and LUN (2) to iSCSI ports A0,1 and B0,1

Note: The Exos X CLI Reference Manual can be found [here](#)

Open a secure-shell, ssh, session to a Management Controller hostname or IP with sufficient privileges.

ssh manage@exos-array

```
Password:
SEAGATE Exos X 4426
System Name: 2U24-Hybrid-Array
System Location: Longmont, CO
Version: I200R005-02
```

create volume outpost-vol0 pool dg01 size 5TB

set initiator id iqn.2004-10.com.ubuntu:01:4f598a60af82 nickname outposts-linux

Note: iSCSI Initiator is host/os-specific- can be found in /etc/iscsi/initiatorname.iscsi on the linux ec2 instance

map volume ports 0-1 lun 2 initiator outposts-linux outposts-vol0

Verify that the volume and mappings are configured on the array:

show maps

```
Volume View [Serial Number (00c0ff43defd00002146f36501000000) Name (outposts-vol0) ] Mapping:
  Ports LUN   Access      Identifier                              Nickname Profile
  --------------------------------------------------------------------------------
  0-1   2     read-write   iqn.2004-10.com.ubuntu:01:4f598a60af82  outposts-linux  Standard
```

The following steps are executed on the linux host with elevated (sudo) privileges and assumes that all iSCSI and multipath packages have been installed and the required services have been enabled. iSCSI services port addresses in this example are running on 192.168.1.(11,13,19,21).

Specific instructions for setting up iSCSI on Ubuntu can be found here:
https://ubuntu.com/server/docs/iscsi-initiator-or-client

The following commands are used to discover and login to the ports on the storage controller and verify that the session is established.

iscsiadm --mode discovery -t sendtargets -p 192.168.1.13

```
192.168.1.11:3260,1 iqn.1992-09.com.seagate:01.array.00c0fff09a09

192.168.1.19:3260,2 iqn.1992-09.com.seagate:01.array.00c0fff09a09

192.168.1.13:3260,3 iqn.1992-09.com.seagate:01.array.00c0fff09a09

192.168.1.21:3260,4 iqn.1992-09.com.seagate:01.array.00c0fff09a09
```

Login to the iSCSI targets:

 # iscsiadm –m node login

```
192.168.1.11:3260,1 iqn.1992-09.com.seagate:01.array.00c0fff09a09

192.168.1.11:3260,1 iqn.1992-09.com.seagate:01.array.00c0fff09a09

192.168.1.11:3260,1 iqn.1992-09.com.seagate:01.array.00c0fff09a09
```

…….

Rescan / identify new LUNS

 # iscsiadm –m session –rescan

```
Rescanning session [sid: 1, target: iqn.1992-09.com.seagate:01.array.00c0fff09a09, portal:
192.168.1.11,3260]

Rescanning session [sid: 2, target: iqn.1992-09.com.seagate:01.array.00c0fff09a09, portal:
192.168.1.11,3260]

Rescanning session [sid: 3, target: iqn.1992-09.com.seagate:01.array.00c0fff09a09, portal:
192.168.1.11,3260]

Rescanning session [sid: 4, target: iqn.1992-09.com.seagate:01.array.00c0fff09a09, portal:
192.168.1.11,3260]
```

List all SCSI targets which will include the Storage array enclosure as well as the dual-path connections of the volumes.

Note: This can be a long list depending on the number of service ports and volumes discovered.

 # lsscsi

```
[0:0:0:0]    disk    ATA      ST1000NM0055-1V4 TN05  /dev/sda

[1:0:0:0]    disk    ATA      ST1000NM0055-1V4 TN05  /dev/sdb

[15:0:0:0]   enclosu SEAGATE  4865             G280  –

[15:0:0:2]   disk    SEAGATE  4865             G280  /dev/sdc

[15:0:0:3]   disk    SEAGATE  4865             G280  /dev/sdg

[16:0:0:0]   enclosu SEAGATE  4865             G280  –

[16:0:0:2]   disk    SEAGATE  4865             G280  /dev/sdd

[16:0:0:3]   disk    SEAGATE  4865             G280  /dev/sdh

[17:0:0:0]   enclosu SEAGATE  4865             G280  –

[17:0:0:2]   disk    SEAGATE  4865             G280  /dev/sde

[17:0:0:3]   disk    SEAGATE  4865             G280  /dev/sdi

[18:0:0:0]   enclosu SEAGATE  4865             G280  –
```

```
[18:0:0:2]   disk    SEAGATE  4865            G280  /dev/sdf
[18:0:0:3]   disk    SEAGATE  4865            G280  /dev/sdj
```

Check Multipath configuration and policy. The output will identify the device mapper links to use for accessing each block device over multiple paths.

multipath -ll

```
3600c0ff00043defd2146f36501000000 dm-2 SEAGATE,4865

size=4.5T features='1 queue_if_no_path' hwhandler='1 alua' wp=rw

`-+- policy='service-time 0' prio=50 status=active

  |- 16:0:0:2 sdd 8:48  active ready running

  |- 15:0:0:2 sdc 8:32  active ready running

  |- 18:0:0:2 sdf 8:80  active ready running

  `- 17:0:0:2 sde 8:64  active ready running

3600c0ff00043e183cd62f06501000000 dm-3 SEAGATE,4865

size=4.5T features='1 queue_if_no_path' hwhandler='1 alua' wp=rw

`-+- policy='service-time 0' prio=10 status=active

  |- 16:0:0:3 sdh 8:112 active ready running

  |- 15:0:0:3 sdg 8:96  active ready running

  |- 17:0:0:3 sdi 8:128 active ready running

  `- 18:0:0:3 sdj 8:144 active ready running
```

Create an ext4 filesystem on one of the device-mapper links, in this case using /dev/dm-3

mkfs.ext4 /dev/dm-3

```
mke2fs 1.46.5 (30-Dec-2021)

Creating filesystem with 1220702208 4k blocks and 152588288 inodes

Filesystem UUID: aaf9c147-7417-43fb-814a-0a99ec97bb18

Superblock backups stored on blocks:

        32768, 98304, 163840, 229376, 294912, 819200, 884736, 1605632, 2654208,

        4096000, 7962624, 11239424, 20480000, 23887872, 71663616, 78675968,

        102400000, 214990848, 512000000, 550731776, 644972544


Allocating group tables: done

Writing inode tables: done

Creating journal (262144 blocks): done

Writing superblocks and filesystem accounting information: done
```

Create a mount point, mount the filesystem and check that is indeed mounted :

```
# mkdir /mnt/iscsi-dm3

# mount /dev/dm-3 /mnt/iscsi-dm3

# df -h /mnt/iscsi-dm3/

Filesystem                                Size  Used Avail Use% Mounted
on

/dev/mapper/3600c0ff00043e183cd62f06501000000  4.6T   28K  4.3T   1%
/mnt/iscsi-dm3
```

Considerations when configuring the AWS Outposts network:

- AWS Outposts deployments differ between Servers and Racks in the way they attach to an on-premises subnet and thus to the Exos X Array's iSCSI service address ports.
- Outpost Server deployments use a Local Network Interface (LNI), while Outposts Rack use a Local Gateway
- Verify connectivity to and from components in AWS Outposts instances to the on-premises subnet
- Local network interface, LNI, throughput on a single Outposts Server is limited to 500mbit. Higher performance can be achieved using multiple Outposts Servers or by using an Outposts Racks

Support for Exos X Arrays

Seagate provides Worldwide 24/7 support for Data Storage products here:

https://www.seagate.com/support/data-storage-systems/contact-support/

North American Direct Support Phone Number: 1-800 732 4283

This page also lists contact phone numbers for EMEA, APAC and Latin American customers.

Seagate's Exos X Arrays provide a full spectrum of storage options for AWS Outposts.

The AWS Service Ready program helps customers find AWS Technology Partner products that integrate directly with specific AWS services. One such example is that Outposts Ready Partners offer products that integrate with AWS Outposts deployments bringing demonstrated experience and success helping clients evaluate and use their technologies productively. As an AWS Outposts Ready Partner, Seagate provides the expertise, availability, and performance to customers that require all-flash, hybrid, and high-capacity HDD arrays at petabyte scale.