



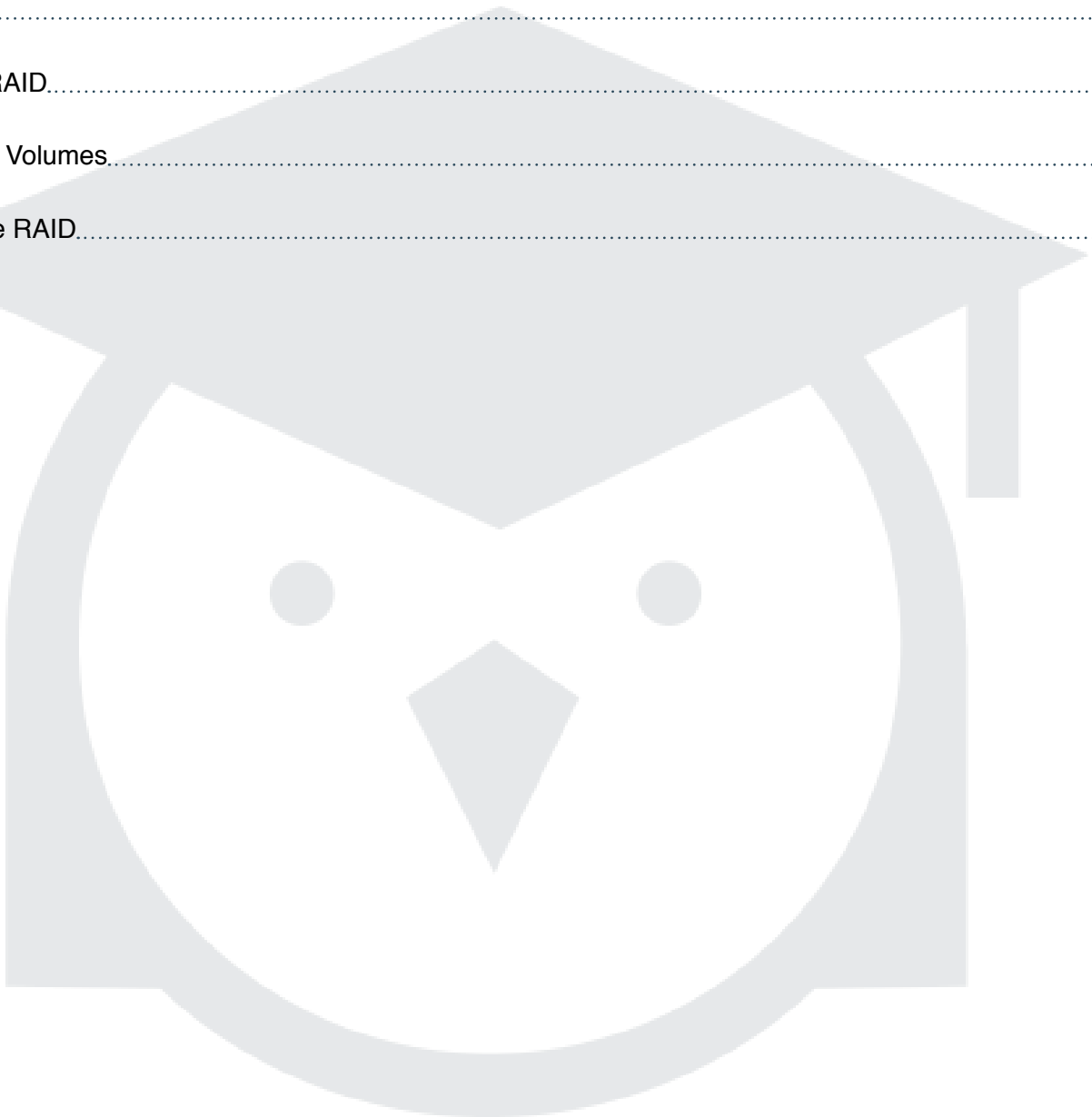
Linux Academy

Hands-On Training

Configuring
EBS Volume
RAID 0

Contents

Introduction.....	1
Setup.....	1
Note.....	1
Configure RAID.....	2
Add EBS Volumes.....	2
Configure RAID.....	3



Introduction

In this lab, we are going to configure an **EBS** (Elastic Block Store) volume with RAID configurations. There are two types of RAID volumes that can be used with EBS: RAID 0 and RAID 1. RAID 0 volumes add increased throughput performance for provisioned IOPS. RAID 1 creates redundancy — it will replicate information from the primary RAID device to the RAID 1 device.

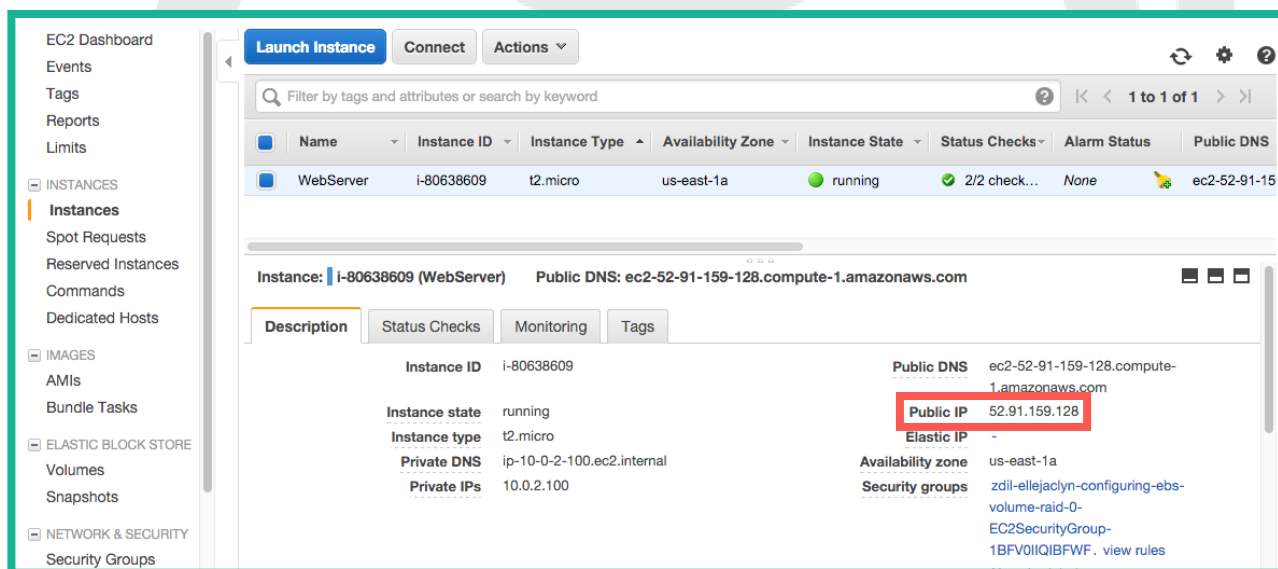
Note

There are limitations to RAID. While more RAID may seem like it would provide greater performance, because RAID communicates within the network, performance limitations may be on the side of the instance, network, or other associated parts.

Begin by logging into the provided AWS console with the given credentials.

Setup

From your **EC2 Dashboard**, view the provided instance, and retrieve the *public* IP address from the **Description** tag.



Log in through SSH, using the traditional Linux Academy credentials of a *linuxacademy* user and *123456* password.

From here, you have the option of working as the root user through the **su** command or adding **sudo** to the beginning of the lab's commands. The commands in this document will assume you are working as root.

To prepare our server to use RAID devices, we must first install the package used to manage RAID:

```
yum install mdadm
```

For more information on the `mdadm` package, view the man page:

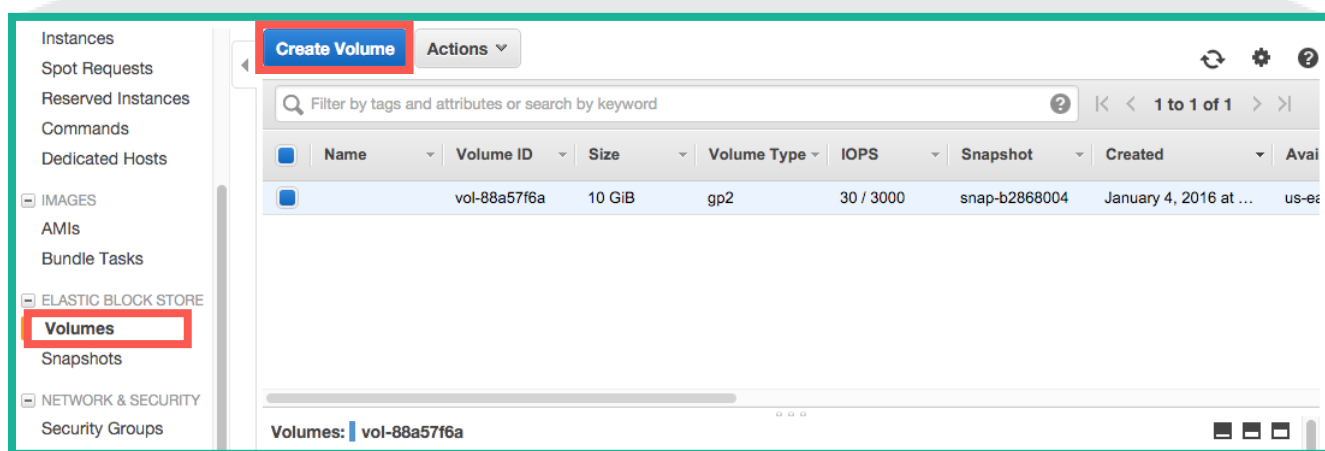
`man mdadm`

Configure RAID

We will configure two EBS volumes, 10 GiB in size, for our RAID. From our EC2 instance, we will configure and mount these volumes.

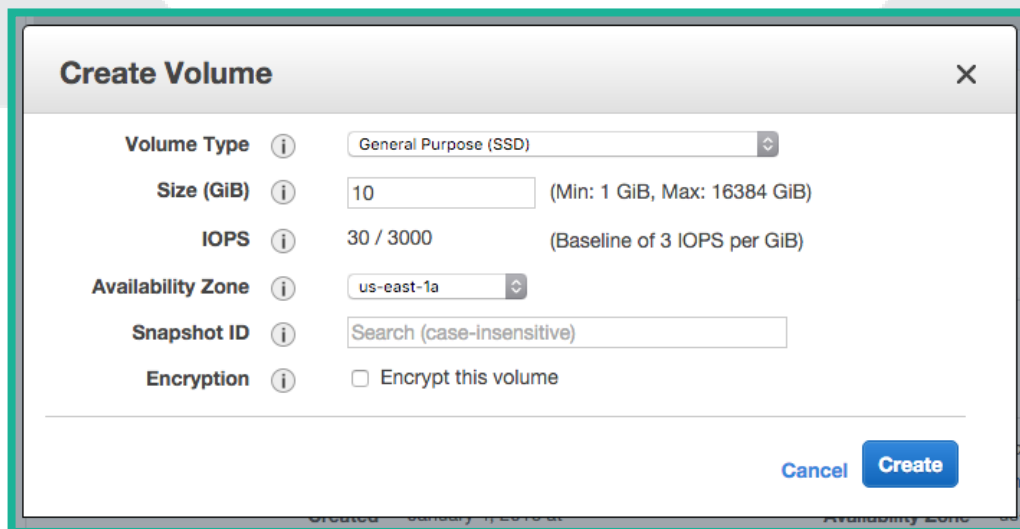
Add EBS Volumes

From the **EC2 Dashboard**, select **Volumes**, under **Elastic Block Store**. Press **Create Volume**.

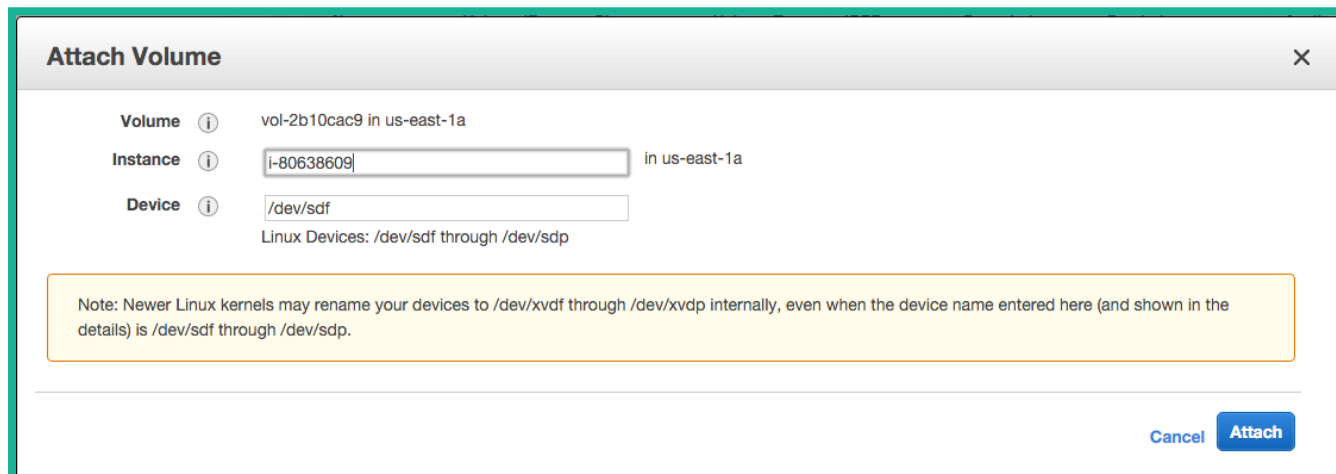


EBS RAID configurations need volumes to be similar in size and performance. You do not want to combine a General Purpose and a Provisioned IOPS volume, for example.

For this volume, we will be using a **Volume Type** of *General Purpose (SSD)*, **10 GiB** in **Size**. Ensure your availability zone is set to the same location as the instance. Press **Create**. Replicate this for a second volume.



These volumes will now need to attach to an instance. Right-click on the first volume you created or select **Actions** and select **Attach Volume**. Attach this to the available *WebServer* instance. You can leave the **Device** set to the default. Click **Attach**.



Attach Volume

Volume ⓘ vol-2b10cac9 in us-east-1a

Instance ⓘ i-80638609 in us-east-1a

Device ⓘ /dev/sdf
Linux Devices: /dev/sdf through /dev/sdp

Note: Newer Linux kernels may rename your devices to /dev/xvdf through /dev/xvdp internally, even when the device name entered here (and shown in the details) is /dev/sdf through /dev/sdp.

Cancel Attach

Repeat this for the second volume.

Configure RAID

Returning to our console, we can navigate to our */dev* directory and see the *xvdf* and *xvdg* volumes we just created:

```
cd /dev
ls
```

Before formatting or partitioning these volumes, we first need to configure them for RAID. This is where the **mdadm** command comes in:

```
mdadm --create --verbose /dev/md0 --level=0 --name=linuxacademy-raid --raid-devices=2 /dev/xvdf /dev/xvdg
```

The **--create** flag denotes that we are creating a new RAID configuration. The **--verbose** flag allows for verbose output. */dev/md0* is how we define the device, and **--level=0** specifies that we want RAID 0. We choose to **name** our's *linuxacademy-raid*. The **raid-devices** flag defines how many devices there are (in this instance, two) while the last two snippets show which volumes are used.

Next, we need to write a filesystem to our RAID device, across all 20 GiB. To do this run:

```
mkfs.xfs -L linuxacademy /dev/md0
```

The **-L** denotes a label (*linuxacademy*, in this example), and */dev/md0* calls to the RAID device just created.

To verify this works, run **lsblk** to view the two RAID devices called md0. We will then need to mount these:

```
mount /dev/md0 /mnt
```

Now, run **df -h** to view your mounted filesystems. You will see your md0 device.

