

## **Tutorial 1 - Benchmarking Amazon EC2 Disk Performance**

*Disclaimer: Subject to updates as corrections are found*

Version 0.10 – **Groups up to 2 are allowed...**

Scoring: up to 10 pts credit on the midterm exam\*

The purpose of this tutorial is to introduce creating Amazon spot instances on Amazon EC2 to then walk through testing the performance of EBS-backed and instance-store backed disk volumes using Bonnie++, a disk and file system performance benchmarking tool. To complete this tutorial, you will need to have Amazon Cloud Credits as the tutorial is not free. The tutorial may be done using a free tier t2.micro instance, but this only allows for benchmarking of a GP2 EBS volume.

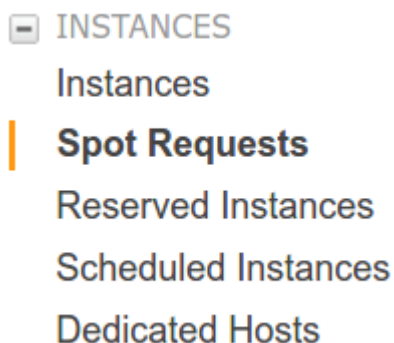
EBS, elastic block store, provide disk volumes which are mounted to Amazon EC2 virtual machines. EBS volumes are replicated disk volumes which are served to the VMs over the network in contrast to instance-store volumes which are hosted using local hardware.

It is recommended to complete the tutorial using a web browser from the same operating system as your Putty or SSH client (e.g. Ubuntu terminal).

Complete the tutorial using the US-EAST-1A Virginia Region.

### **1. Launch an EBS-backed Amazon EC2 Ubuntu 16.04 instance**

First, log into the AWS management console, and navigate to the EC2 Dashboard. Click on the “Spot Requests” option on the left-hand menu:



Next, Select the “Request Spot Instances” button:

**Request Spot Instances**

\* - 10 points are provided as replacement points for missed questions on the midterm exam

As in the screenshot, configure the spot request as follows.  
**If the parameter is not specified, keep the default value.**

Request type: \* Request (one-time)

Target capacity: 1

AMI: (select from the dropdown) Ubuntu Server 16.04 LTS (HVM)

Instance type: c3.large

Network: default

Availability Zone: Select specific zone/subnet...

Check, us-east-1e, default subnet

Check the pricing history (click the button)

Set your max price (per instance/hour)

Select a value at your discretion. At least 10 cents is recommended.

Click on the <Next> button

The screenshot shows the AWS Spot Instance Launch Wizard interface. The top navigation bar includes 'Services', 'Resource Groups', and user information. The wizard title is 'Spot instance launch wizard'. The left sidebar shows the progress: 'Step 1: Find instance types' (active), 'Step 2: Configure', and 'Step 3: Review'. The main content area is titled 'Select request type' and contains several sections:

- Request type:** Three radio buttons are shown: 'Request' (selected), 'Request and Maintain', and 'Reserve for duration'. Descriptions are provided for each.
- Target capacity:** A text input field with the value '1' and a button labeled 'instances'.
- AMI:** A dropdown menu showing 'Ubuntu Server 16.04 LTS (HVM)' and a 'Search for AMI' button.
- Instance type(s):** A dropdown menu showing 'c3.large (2 vCPU, 3.75 GiB, 2 x 16)' and a 'Select' button.
- Allocation strategy:** Two radio buttons: 'Lowest price' (selected) and 'Diversified'.
- Network:** A dropdown menu showing 'vpc-4f8db929 (172.31.0.0/16) (default)' and a 'Create new VPC' button.
- Availability Zone:** A dropdown menu showing 'Select specific zone/subnet...'. Below it, a list of zones is shown: 'us-east-1a', 'us-east-1b', 'us-east-1c', 'us-east-1d', and 'us-east-1e' (selected). A 'Create new subnet' button is also present.
- Subnet:** A dropdown menu showing 'subnet-1a82f326 (172.31.4)'.
- Maximum price:** Two radio buttons: 'Use automated bidding (recommended)' and 'Set your max price (per instance/hour)'. The latter is selected, and a text input field shows '\$ .20'. A 'Pricing History' button is located to the right.

At the bottom, there is a footer message: 'Not finding what you are looking for? Use the old Spot Instance Launch Wizard.' and two buttons: 'Cancel' and 'Next'.

Now, the next screen is “Step 2 – Configure”.

Check the box for Instance Store, “Attach at launch” ...

EBS volumes:

Change the capacity from 8GB to 15GB. The default amount of 8GB does not provide us with enough free space for the Bonnie++ disk benchmark. 7,520 MB is required.

Key pair name:

If this is the first time you’ve launched a VM, you’ll need to create a new keypair. When generating a keypair, you’ll download the key file to your machine and use in place of a plaintext password to log into the cloud VM.

If using the Windows Putty client to access the VM, follow these instructions to convert the keypair for use:

<http://www.cnx-software.com/2012/07/20/how-use-putty-with-an-ssh-private-key-generated-by-openssh/>

If using Ubuntu/Linux, no conversion is required.

Security groups: default

Auto-assign IPv4 Public IP: Use subnet setting (Enable)

Next, click on the review button.

Be sure that “Attach instance store” says “yes”... If not, go back and check the box.

## **2. Log into your Amazon EC2 Spot Instance**

Once the spot request has been submitted, let’s check what is your IP address. In your web browser, open Google search, and type in “What is my IP?”.

Your IP address should appear.

Note the first 3 numbers.

Let’s add SSH permission for your CIDR network block.

If your IP is for example 120.118.53.108, then your 24-bit CIDR block which would include all 255 addresses on the local subnet will be 120.118.53.0/24.

Go to EC2 – Instances, and select your instance once it appears.

In the instance description pane, find “Security Groups”.

Click on “default”...

This takes you into the Security Groups editor.

Click on the “Inbound” tab.

Click on “Edit”.

Click “Add Rule”

For the Rule Type, select “SSH”

Type	Protocol	Port Range	Source
SSH	TCP	22	Custom 131.191.88.0/24
All ICMP - IPv4	ICMP	0 - 65535	Custom 131.191.88.0/24

Buttons: Add Rule, Cancel, Save

For the Source, add your CIDR block, e.g. 120.118.53.0/24

Optionally, you can add the “All ICMP - IPv4” permission. This enables you to “ping” VM instances.

Every VM that is launched will have a security group.

To access VMs from your local network(s), security settings will need to be adjusted accordingly.

Now, navigate back to “Instances”, find your instance, and select it.

Note your instance’s “IPv4 Public IP”.

If you’ve configured the “All ICMP - IPv4” permission, try pinging your instance.

```
$ ping 54.165.102.178
PING 54.165.102.178 (54.165.102.178) 56(84) bytes of data.
64 bytes from 54.165.102.178: icmp_seq=1 ttl=41 time=79.2 ms
64 bytes from 54.165.102.178: icmp_seq=2 ttl=41 time=71.7 ms
```

CTRL-C will exit.

Next, using your putty or ssh terminal client, and your keypair, SSH into the instance as follows. The first time you’re connecting to a new IP, you may need to acknowledge the host’s authenticity.

```
ssh -i <your_key_file_name> ubuntu@<the IPv4 Public IP>
```

Example Output:

```
$ssh -i <your_key_file_name> ubuntu@54.165.102.178
The authenticity of host '54.165.102.178 (54.165.102.178)' can't be established.
ECDSA key fingerprint is SHA256:35893chQunQ2eVt908X8jxyvJpYJb0NdxOQjmi6U3OQ.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added '54.165.102.178' (ECDSA) to the list of known hosts.
Welcome to Ubuntu 16.04.2 LTS (GNU/Linux 4.4.0-64-generic x86_64)
```

- \* Documentation: <https://help.ubuntu.com>
- \* Management: <https://landscape.canonical.com>
- \* Support: <https://ubuntu.com/advantage>

Get cloud support with Ubuntu Advantage Cloud Guest:  
<http://www.ubuntu.com/business/services/cloud>

0 packages can be updated.  
0 updates are security updates.

The programs included with the Ubuntu system are free software;  
the exact distribution terms for each program are described in the  
individual files in /usr/share/doc/\*/copyright.

Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by  
applicable law.

To run a command as administrator (user "root"), use "sudo <command>".  
See "man sudo\_root" for details.

[ubuntu@ip-172-31-56-31:~\\$](#)

### 3. **Install the Bonnie++ disk benchmark utility**

It is easy to install bonnie++.  
From your VM's command line simply type:

```
$sudo apt install bonnie++
```

Next, let's check out the EBS and instance store disks on the VM.  
Type the "df" command to inspect disk file system space:

```
$ df -h
Filesystem      Size  Used Avail Use% Mounted on
udev            1.9G   0 1.9G   0% /dev
tmpfs           377M  5.3M  371M   2% /run
```

/dev/xvda1	15G	948M	14G	7%	/
tmpfs	1.9G	0	1.9G	0%	/dev/shm
tmpfs	5.0M	0	5.0M	0%	/run/lock
tmpfs	1.9G	0	1.9G	0%	/sys/fs/cgroup
/dev/xvda	15G	39M	15G	1%	/mnt
tmpfs	377M	0	377M	0%	/run/user/1000

Based on our launch parameters, your instance store volume has been mounted under “/mnt”. The instance store volume represents space on the local hardware that hosts the VM. For the c3.large instance type, this is a local SSD drive.

Your root partition is an EBS volume, mounted under “/”.  
By default, the EBS volume is created as a GP2 – General Purpose 2 SSD EBS volume. This volume is granted a baseline of 100 IOPS (I/O operations per second), and is burstable to 3,000 using a credit-based allocation approach.

For a detailed description of how Amazon manages IOPS for EBS volumes see this article: <http://docs.aws.amazon.com/AWSEC2/latest/UserGuide/EBSVolumeTypes.html>

Amazon allows a certain number of IOPS relative to the overall disk size. This is approximated based on the typical number of IOPS for a hard disk based on different usage scenarios.

#### 4. **Run the Bonnie++ disk benchmark utility to test EBS and Instance Store disk performance**

To run our disk benchmark, we will use ~ 7,520 MB of free disk space.

From the command line:

Create a instance store temp directory and set the permissions:

```
$sudo mkdir /mnt/tmpi
$sudo chmod a+rwX /mnt/tmpi
```

Now run bonnie on the EBS volume:

The /tmp directory is on the “/” root partition.  
Using /tmp as the bonnie directory will test the EBS volume performance.

```
$bonnie++ -d /tmp -s 7520M -n 0 -m TEST-EBS -f -b -q > bonnie.csv
```

This will capture the output to a text file.

Next, run bonnie on the Instance store volume:

The /mnt/tmpi directory is on the “/mnt” partition, which is the auto-mounted instance store volume which is hosted local to the VM.

Using /mnt/tmpi tests the instance store volume performance.

```
$bonnie++ -d /mnt/tmpi -s 7520M -n 0 -m TEST-IS -f -b -q >> bonnie.csv
```

Bonnie provides formatting utilities which take the CSV output and convert the output to either text (txt) or html.

These utilities are:

bon\_csv2html

bon\_csv2text

To use the utilities to generate formatted output, simply redirect your bonnie.csv output file into the utility:

```
$bon_csv2txt < bonnie.csv
```

AND

```
$bon_csv2html < bonnie.csv
```

Now, using the clipboard, copy the contents of the bon\_csv2html output, and save this as a local .html file on your laptop. Try opening the file in a web browser by navigating from “[file:///](#)”

---

## Tutorial Questions:

Submit written answers as a PDF file on canvas.

Include your CSV output from all bonnie tests at the bottom of the PDF.

1. (2 points) Which disk (EBS or Instance Store) provided faster sequential output block reads throughput (in K/sec)?
2. (2 points) Which disk (EBS or Instance Store) required more CPU capacity for sequential output block reads?
3. (2 points) Now evaluate performance for Sequential output block rewrites and Sequential input block reads. Which disk performs better EBS or Instance Store?
4. (2 points) For Sequential output block rewrites and Sequential input block reads, which disk requires more CPU capacity?
5. (1 point) Now consider performance of random seeks. Which disk (EBS or Instance Store) provides faster random seek disk performance?
6. (1 point) If your EBS volume test provided faster performance compared to the Instance Store volume, can you suggest why? What is unique about how amazon hosts EBS volumes that could provide this performance improvement versus our local instance store volume? (Hint: the unique characteristic is mentioned previously in the tutorial...)

## 5. Bonus: Benchmark a “Provisioned IOPS” EBS Volume

Provisioned IOPS EBS volumes forgo the credit based approach to provide consistent, guaranteed performance of EBS volumes within +/- 10%. The catch, this performance costs more, and is not offered as a FREE tier resource. The free tier offers up to 30GBs of general purpose EBS storage a month.

Let's create a new IOPS EBS volume, and attach it to our currently running spot instance:

Go to EC2 | Elastic Block Store | Volumes

And then click “Create Volume”.

Specify as follows:

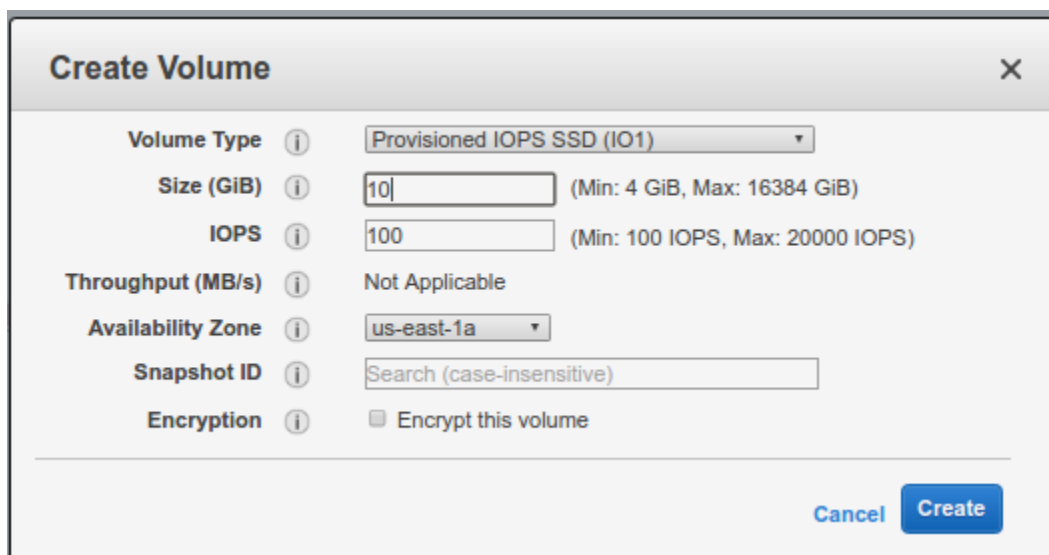
Volume Type: Provisioned IOPS SSD

Size (GiB): 10

IOPS: 100

Availability Zone: us-east-1e

The availability zone must match where your EC2 spot instances is running...



The screenshot shows the 'Create Volume' dialog box with the following settings:

- Volume Type: Provisioned IOPS SSD (IO1)
- Size (GiB): 10 (Min: 4 GiB, Max: 16384 GiB)
- IOPS: 100 (Min: 100 IOPS, Max: 20000 IOPS)
- Throughput (MB/s): Not Applicable
- Availability Zone: us-east-1a
- Snapshot ID: Search (case-insensitive)
- Encryption: ☐ Encrypt this volume

Buttons: Cancel, Create

According to Amazon, an IOPS EBS volume with these settings will cost 12.5 cents per GB compared to 10 cents per GB for a general-purpose (GP2) EBS volume. Additionally Amazon charges for the guaranteed IOPS. The charge for 1-month is \$.065 x 100 IOPS or \$6.50/month. The total monthly cost of a 10GB volume with a 100 IOPS guarantee is \$7.75/month.

**\*\* There would be no practical reason to ever create this EBS volume since the guaranteed IOPS is no better than what GP2 provides. \*\***

Once the volume has been created (try clicking refresh a few times), select “Action” and then “Attach Volume”

When attaching, note the device name. Something like /dev/sdf

When attaching a brand new EBS volume to an EC2 instance the disk will not initially be formatted. We must format the disk before it can be used:

Complete the following command sequence to partition the disk before formatting:

```
$ sudo fdisk /dev/xvdf
```

Welcome to fdisk (util-linux 2.27.1).

Changes will remain in memory only, until you decide to write them.

Be careful before using the write command.

Device does not contain a recognized partition table.

Created a new DOS disklabel with disk identifier 0xd40d55bd.

Command (m for help): n

Partition type

p primary (0 primary, 0 extended, 4 free)

e extended (container for logical partitions)

Select (default p): p

Partition number (1-4, default 1): 1

First sector (2048-20971519, default 2048):

Last sector, +sectors or +size{K,M,G,T,P} (2048-20971519, default 20971519):

Created a new partition 1 of type 'Linux' and of size 10 GiB.

Command (m for help): t

Selected partition 1

Partition type (type L to list all types): 83

Changed type of partition 'Linux' to 'Linux'.

Command (m for help): w

The partition table has been altered.

Calling ioctl() to re-read partition table.

Syncing disks.

Now, let's format the disk:

<pre>\$ sudo mkfs.ext4 /dev/xvdf1 mke2fs 1.42.13 (17-May-2015) Creating filesystem with 2621184 4k blocks and 655360 inodes Filesystem UUID: ff9843da-af26-4c91-ac82-cc6856f61e11 Superblock backups stored on blocks: 32768, 98304, 163840, 229376, 294912, 819200, 884736, 1605632  Allocating group tables: done Writing inode tables: done Creating journal (32768 blocks): done Writing superblocks and filesystem accounting information: done</pre>	
--	--

Now, let's create a new mount point and tmp directory to run bonnie++. We'll call the temp directory tmp\_esp for "provisioned" ebs storage.

<pre>\$sudo mkdir /mnt2</pre>
-------------------------------

Now, let's mount the new EBS volume to the new mount point:

<pre>\$sudo mount /dev/xvdf1 /mnt2 \$sudo mkdir /mnt2/tmp_esp \$sudo chmod a+rwX /mnt2/tmp_esp</pre>
--

Now, let's test this volume and append our results to the bonnie.csv output file:

<pre>\$bonnie++ -d /mnt2/tmp_esp -s 7520M -n 0 -m TEST-EBSP -f -b -q &gt;&gt; bonnie.csv</pre>
--

Note, the test may be slower because of the very low IOPS of the EBS volume.

Once the test completes, capture your bon\_csv2html output, and copy and paste the HTML to a html file on your laptop, save it, and view in a browser.

Or, alternatively, use bon\_csv2text and view using the command line.  
How does the provisioned IOPS EBS volume perform?

## 6. Cleanup

At the end of the tutorial, be sure to **TERMINATE** all EC2 instances, and **DELETE** all EBS volumes. Failing to do so, could result in loss of AWS credits or AWS charges to a credit card.