TCSS 562: Software Engineering for Cloud Computing
Fall 2019
http://faculty.washington.edu/wlloyd/courses/tcss562

School of Engineering and Technology
University of Washington – Tacoma

Tutorial 3 – Introduction to Amazon EC2 and VM Storage Performance

Disclaimer: Subject to updates as corrections are found

Version 0.14 – **Groups up to 2 are allowed...**

Scoring: 20 pts maximum

The purpose of this tutorial is to introduce creating Amazon spot instances (Virtual Machines) on Amazon EC2, and then to walk through testing the disk performance of EBS-backed and instance-store backed disk volumes using Bonnie++, a disk and file system performance benchmarking tool. To complete this tutorial, the use of Amazon Cloud Credits is required as creating EC2 spot instances is not free.

Amazon EC2 instances, also known as virtual machines or VMs have a letter designating the family, and a number identifying the generation. Presently there are 5 generations (1-5). Most families do not have 5 generations of instances (only m). Instances combine a family letter and generation number followed by a period and a size (small, large, xlarge, 2xlarge, 4xlarge, 8xlarge, etc.) The sizes correspond to the same configuration, but an increasing quantity of available resources (e.g. CPU cores, memory, network capacity, storage capacity). The most common instance families include:

c	Compute Optimized Instances	Typically fast CPUs, but less memory
m	General Purpose Instances	More memory than C-family, slower CPUs
r	High Memory Instances	More memory than C or M
t	Burstable instances (cpu-time limited)	Lower-cost instances with CPU quotas

Less common instances include:

f	FPGA Instances	Instances with an on-board programmable FPGA
h/hs	Storage-optimized instances	Instances with enhanced disk capacity (HDDs)
i	Storage-optimized instances	Instances with enhanced disk capacity (SSDs)
p/g	GPU Instances	Instances with on-board GPU(s)
x	Extra-high memory	Instances with extreme memory

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).

It is recommended to use the **us-east-2 Ohio** region for cloud resources throughout TCSS 562. Currently (Fall 2019) the Ohio region has the lowest spot instances prices. Consolidating resources in one region reduces duplication of data and also the likelihood of accidentally creating resources (e.g. VMs) in another region and forgetting to terminate them resulting in accidental charges.

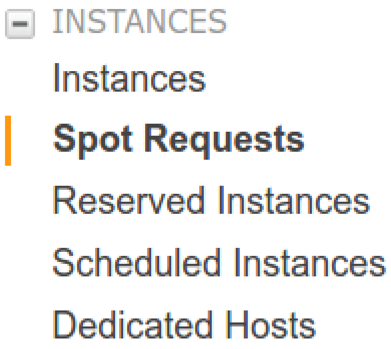
The AWS region can be changed using the dropdown list in the upper right-hand corner of the GUI. Select “**Ohio**”.

EBS, elastic block store, are disk volumes that are attached to Amazon EC2 virtual machines. EBS volumes are replicated disk volumes which are served to the VMs over the network in contrast to ephemeral instance-store volumes which are hosted using the local hardware where the VM itself is hosted. Here is a feature comparison:

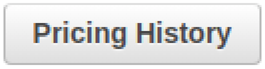
EC2 Instance Store	Elastic Block Store
- Disk hosted local to VM (on same physical server) - Non-persistent volume (no backup) - No automatic replication of data - Old days: snapshots only supported for VMs in XEN “pv” paravirtual mode (not hvm) - Nowadays: snapshots generally not supported - SSD or HDD, also NVMe SSD - IOPS based on HW capabilities and sharing	- Persistent block store volume - 99.999% availability - Automatic replication within availability zone - Snapshot support - Can modify volume as needed (vertical scaling) - SSD or HDD - Auto recovery - common types: gp2 and io1 - scalable IOPS (I/O operations per second)

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

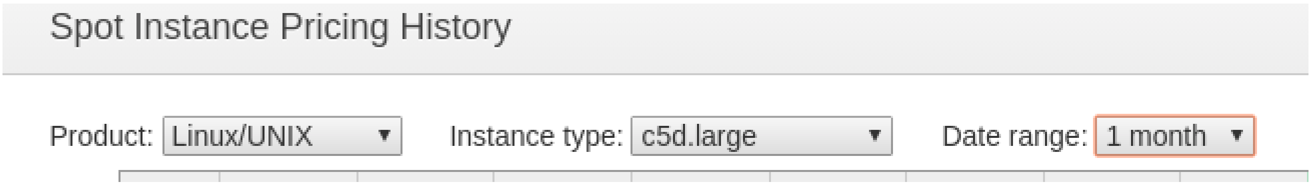
First, log into the AWS management console, and navigate to the EC2 Dashboard.
Click on the “Spot Requests” option on the left-hand menu:



Before launching a auction-based“spot instance” that offers EC2 VMs at a fraction of the full price, check the going rate for VMs in your region. Select the “Pricing History” button:



Filter the graph as follows:



Spot Instance Pricing History

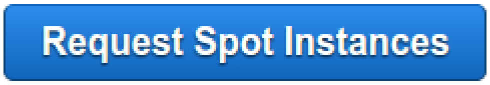
Product: Linux/UNIX   Instance type: c5d.large   Date range: 1 month

Now, *write down* the least expensive “availability zone” within the Ohio region. The options include: us-east-2a, us-east-2b, and us-east-2c. The price shown is the price per hour to rent a c5d.large VM.

Check the capabilities of the c5d.large VM using this website:
https://www.ec2instances.info/?filter=c5d

Adjust or remove the filter to inspect other types of VMs.

Next, Select the “Request Spot Instances” button:



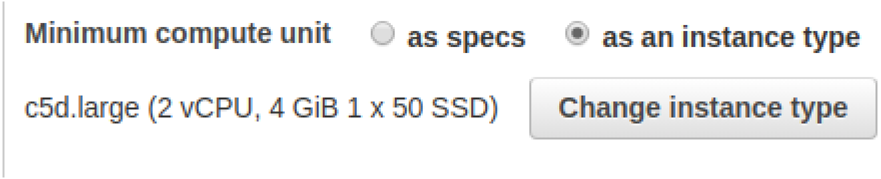
As in the screenshot, configure the spot request as follows.
If the parameter is not specified, keep the default value.

Request type: * Request (one-time)

Target capacity: 1

AMI: (select from the dropdown) “Ubuntu Server 18.04 LTS (HVM)”

Now, select the **instance type** by clicking **“Change instance type”** and selecting **c5d.large** from the list:


Minimum compute unit ○ as specs ● as an instance type

c5d.large (2 vCPU, 4 GiB 1 x 50 SSD)

Network: vpc-???????? (172.?.0.0/16) (default)

Availability Zone: In the drop down list select “Select a specific zone/subnet…”.

Now, check the least expensive availability zone that you’ve written down from above: either *us-east-2a, us-east-2b, or us-east-2c*

*** *Selecting the availability zone restricts our VM to be created in a specific data center within the region. Data centers are called “availability zones” (AZs). Different AZs have different demand and pricing for spot instances. Every AWS account relabels AZs so that not everyone has the same mapping for us-east-2a. This spreads out requests for resources better. Your label may not match your friend’s AWS account.* ***

Now set the **Key pair name**.

If this is your first time creating a VM, create a new keypair by clicking on the link:

Create new key pair

This opens a new window. 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/

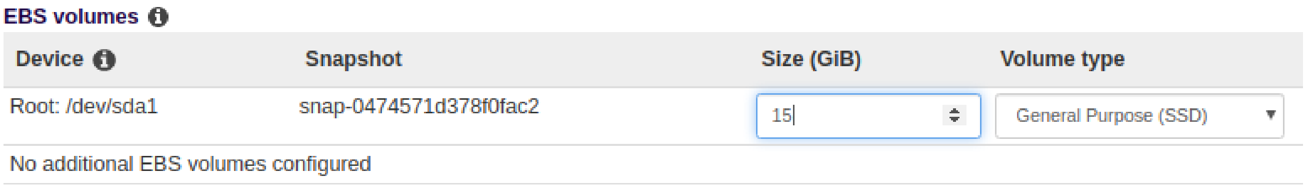
Now, go back to the spot instance wizard and **REFRESH** this list of keypairs. Select your new keypair.

If using a Ubuntu/Linux machine to access AWS, no key conversion is required.

Now click the "**Additional configurations"** tree list to reveal more options:

For the EBS volume:

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.

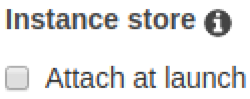


EBS volumes

Device	Snapshot	Size (GiB)	Volume type
Root: /dev/sda1	snap-0474571d378f0fac2	15	General Purpose (SSD)
No additional EBS volumes configured			

By default, the root volume that includes the Linux operating system, called the boot drive, with be created using the EBS volume.

Now, tell the wizard we also want an "Instance store" local disk.
In the wizard, please select the "Instance store" check box:



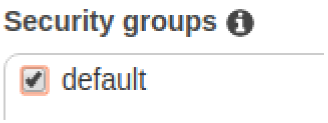
Instance store

□ Attach at launch

This will automatically format and mount a data drive, as a secondary disk.
This disk will be an instance store type disk volume.

This way, your VM will have both types of disks. A network disk (EBS) and a local disk (instance store).

Next, specify that you'd like to use the default security group by checking the box:

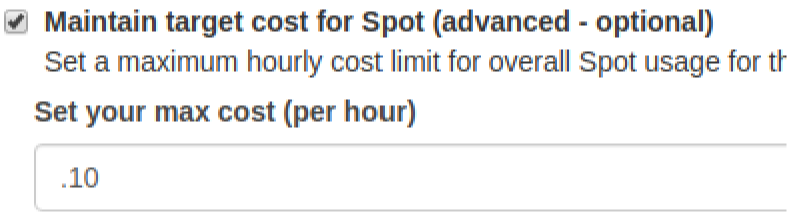


Security groups

☑ default

You will see that by default “Auto-assign IPv4 public IP” is set to **Enable**. This means your VM will automatically receive a public network IP address to facilitate connecting via SSH.

Then, scroll down quite a ways to specify a maximum bid price for your VM. Check the box, and enter 10 cents an hour as the maximum you’re willing to pay to rent this VM:



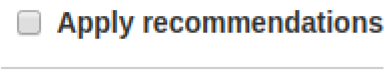
☑ Maintain target cost for Spot (advanced - optional)

Set a maximum hourly cost limit for overall Spot usage for th

Set your max cost (per hour)

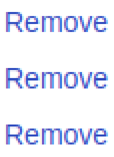
.10

Next, there is a grey box called “Fleet request”.
Uncheck the “Apply recommendations” checkbox:



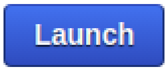
□ Apply recommendations

Next, remove **ALL** recommendations, except c5d.large, by click Remove for each. The link appears on the right-hand side of the recommended instance types. These recommendations provide similar alternatives. **We don’t want them for tutorial #3:**



Before launching, be sure that you’ve checked the “Attach at launch” checkbox for Instance Store. If not, go back and check the box.

Next, click on the blue “Launch” button at the bottom of the wizard:



2. Log into your Amazon EC2 Spot Instance

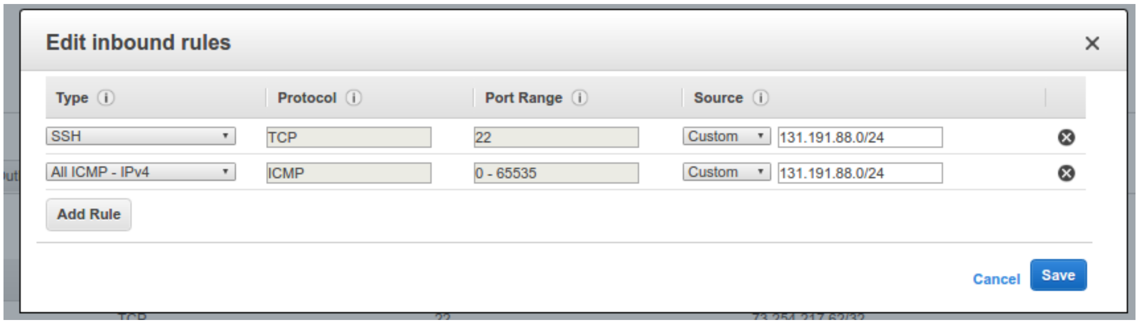
Once the spot request has been submitted, let’s check what is your IP address.
In your web browser, open Google search, and type in “What is my IP?”.

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”:



Edit inbound rules

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

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

The motivation for adding your CIDR block is that if you commonly use a buliding’s WiFi network you may receive various addresses in the same block from day-to-day. If you add ranges vs. individual IPs you’ll hopefully not need to update the security as often.

Overtime you can accumulate access permissions in your default security group for common places you tend to work. For example: at UW-Tacoma, at home, at Starbucks, etc.

Another Security Group option is to add the “All ICMP – IPv4” permission. This enables you to “ping” VM instances.

For every VM that is launched, you will associate 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 (VM), 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.

On Linux, verify the permissions of your keypair:
From the command prompt, navigate to the directory including your key
$ls -alt *.pem

-rw-rw-r-- 1 wlloyd wlloyd   1696 Oct 8 20:03 uw_wlloyd_oh1.pem

Sometimes when first downloading the key, the persmissions are too open. Here the key is world-readable. Owner, group, and other all have the “r” (read) permission.

Set owner read write permission with Linux “chmod”:

chmod 0600 <keypair-name>

Now recheck the permissions:
$ls -alt *.pem
-rw------- 1 wlloyd wlloyd 1696 Oct 8 20:03 uw_wlloyd_oh1.pem

If your key has any other permissions than shown above it will likely not work.

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. Partition and format the NVMe SSD emphermal disk

By default, the local NVMe SSD is not mounted when booting up a c5d.large VM.

Check your mounted disk partitions with “df -h”:

```
$df -h
```

The 15 GB device that is mounted on “/” (as root) is the EBS volume.
We would like to prepare the local disk and mount this at “/mnt”.

Follow the command sequence below to **partition** your local NVMe SSD virtual disk, then **format the disk** with the Linux ext4 filesystem. Then **mount the volume** so it is accessible:

```
$sudo parted /dev/nvme0n1 mklabel msdos

$ sudo parted -a optimal /dev/nvme0n1 mkpart primary ext4 0%  46.6GB
```

```
Information: You may need to update /etc/fstab.

$ sudo mkfs.ext4 /dev/nvme0n1p1
mke2fs 1.44.1 (24-Mar-2018)
Discarding device blocks: done
Creating filesystem with 11376640 4k blocks and 2845248 inodes
Filesystem UUID: 7c363c9a-a3f6-43ec-8770-72e1c59c8da2
Superblock backups stored on blocks:
        32768, 98304, 163840, 229376, 294912, 819200, 884736, 1605632, 2654208,
        4096000, 7962624, 11239424

Allocating group tables: done
Writing inode tables: done
Creating journal (65536 blocks): done
Writing superblocks and filesystem accounting information: done

$ sudo mount /dev/nvme0n1p1 /mnt -t ext4
$ df -h
Filesystem      Size  Used Avail Use% Mounted on
udev            1.8G     0  1.8G   0% /dev
tmpfs           371M  720K  370M   1% /run
/dev/nvme1n1p1   15G  1.1G   14G   8% /
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/loop0       88M   88M     0 100% /snap/core/5328
/dev/loop1       13M   13M     0 100% /snap/amazon-ssm-agent/495
tmpfs           371M     0  371M   0% /run/user/1000
/dev/nvme0n1p1   43G   53M   41G   1% /mnt
```

At this stage, the EBS volume is mounted at "/", and the local NVMe SSD disk is mounted at "/mnt". We can profile performance of both types of cloud disks using the same VM.

4. Install the Bonnie++ disk benchmark utility

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

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

We've mounted your instance store volume under "/mnt". The instance store volume represents space on the local hardware that hosts the VM. For the c5d.large instance type, this is a local NVMe 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.

5. 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++ wants to create files that are twice the size of the RAM of the VM.
Here you should have increased the root volume size to 15 GB or else the EBS volume won’t have enough space on a c5d.large. Available disk space can be checked with: “df -h”

```
$sudo chmod a+rwx /mnt/tmp
$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.
We usethe same file size on the instance store volume for comparison purposes.

```
$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. If submitting with a partner, include both names at the top of the PDF as they appear in CANVAS.

Only one person in the team should submit the assignment!

Include your HTML table output (or CSV if unavailable) from all Bonnie tests at the bottom of the PDF.

1. (4 points) What EC2 instance type did you run Bonnie++ on?

2. (3 points) Which disk (EBS or Instance Store) provided faster sequential output block reads throughput (in K/sec)?

3. (3 points) Which disk (EBS or Instance Store) required more CPU capacity for sequential output block reads?

4. (3 points) Now evaluate performance for Sequential output block rewrites and Sequential input block reads. Which disk performs better EBS or Instance Store?

5. (3 points) For the operations: "Sequential output block rewrites" and "Sequential input block reads", which disk (EBS or Instance Store) requires more CPU capacity to perform the I/O?

6. (2 point) Now consider performance of random seeks. Which disk (EBS or Instance Store) provides faster random seek disk performance?

7. (2 point) If your EBS volume test provided faster performance compared to the Instance Store volume *for the random seek test*, 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...)

6. Bonus Activity: Benchmark a "Provisioned IOPS" EBS Volume

(Optional / Non-graded) 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.

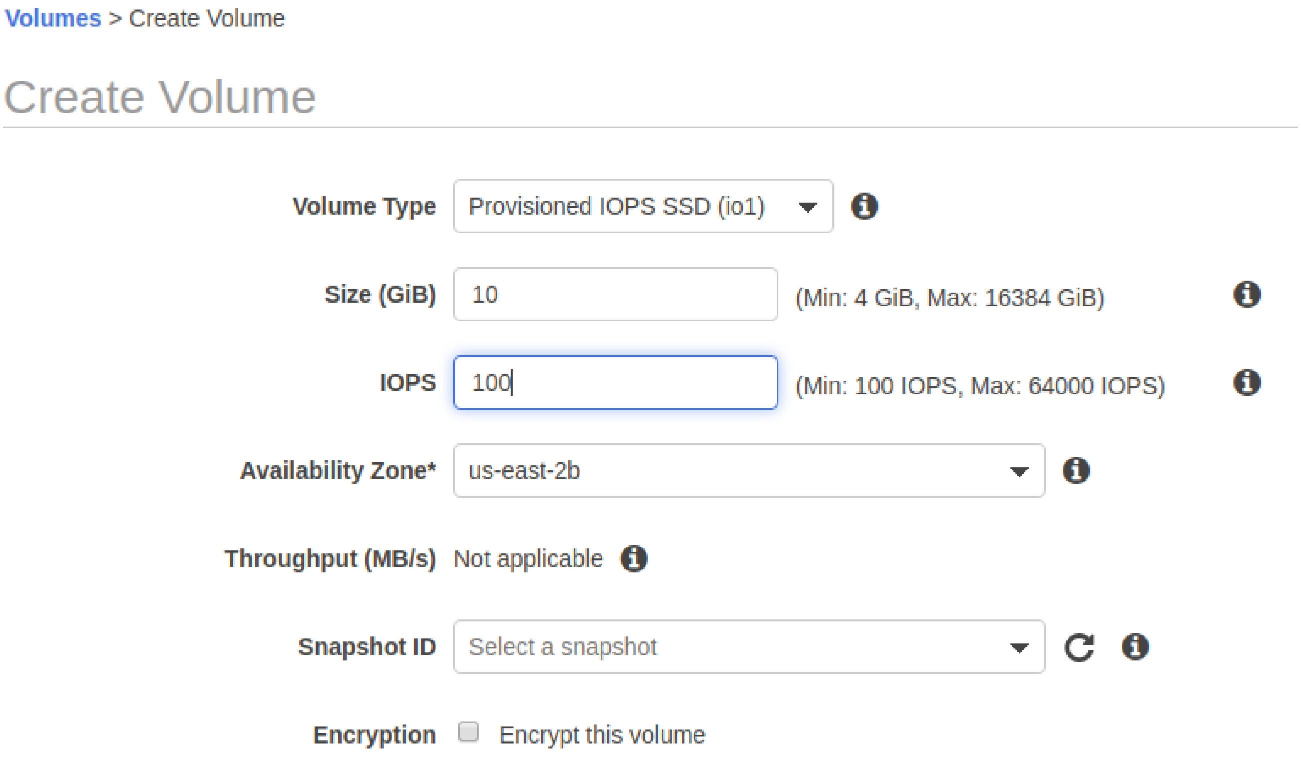
Let’s create a new IOPS EBS volume, and attach it to our currently running spot instance:

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…



Volumes > Create Volume

Create Volume

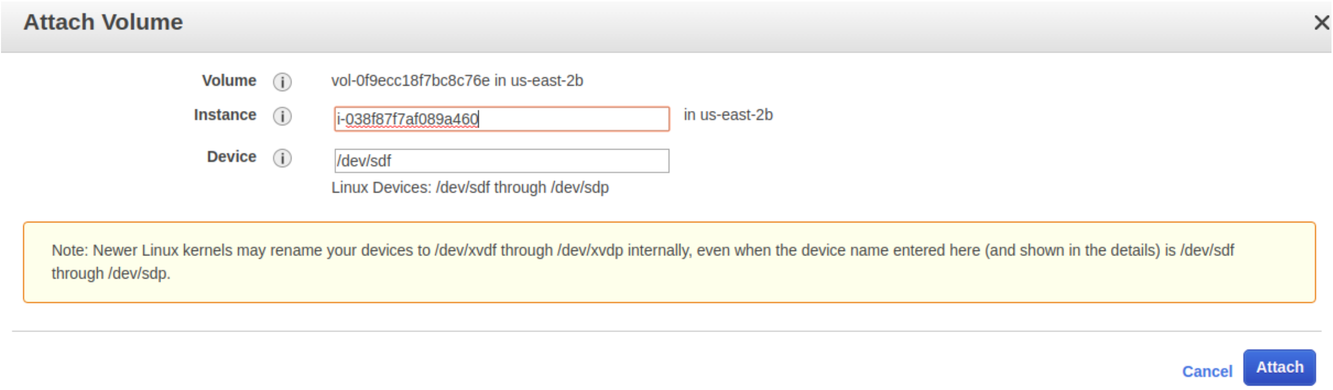
Volume Type	Provisioned IOPS SSD (io1)
Size (GiB)	10 (Min: 4 GiB, Max: 16384 GiB)
IOPS	100 (Min: 100 IOPS, Max: 64000 IOPS)
Availability Zone*	us-east-2b
Throughput (MB/s)	Not applicable
Snapshot ID	Select a snapshot
Encryption	☐ Encrypt this volume

<<< <u>WARNING</u> - BE SURE TO DELETE THIS VOLUME AFTER COMPLETING TESTS. IT IS EXPENSIVE TO KEEP IN YOUR ACCOUNT !!! >>>

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”



Attach Volume

Volume: vol-0f9ecc18f7bc8c76e in us-east-2b

Instance: i-038f87f7af089a460 in us-east-2b

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.

Please note the disclaimer in yellow.

As of this writing the disclaimer is out of date for 5th gen SSD enabled resources.

When attaching to your c5d instance the device name will be mapped to /dev/nvme2n1.

Next you’ll need to partition, format, and mount the disk:

```
$sudo parted /dev/nvme2n1 mklabel msdos

$sudo parted -a optimal /dev/nvme2n1 mkpart primary ext4 0% 10.0GB

$sudo mkfs.ext4 /dev/nvme2n1p1

$ sudo mkdir /mnt2

$ sudo mount /dev/nvme2n1p1 /mnt2 -t ext4
```

Then check that the disk is available at /mnt2:

```
$ df -h
```

Create a local tmp directory and grant world read write execute permission:

```
$sudo mkdir /mnt2/tmp_ebsp

$sudo chmod a+rwx /mnt2/tmp_ebsp
```

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

```
$bonnie++ -d /mnt2/tmp_ebsp -s 7520M -n 0 -m TEST-EBSP -f -b -q >> bonnie.csv
```

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?

7. Cleanup

At the end of the tutorial:

Be sure to:

#1 - **TERMINATE** all EC2 instances

#2 - **DELETE** all EBS volumes

Failing to clean up could result in loss of AWS credits and/or AWS charges to a credit card.