Chapter 26

Deploying Mass Spectrometric Data Analysis in the Amazon AWS Cloud Computing Environment

Jonathan E. Katz

Abstract

There are many advantages for deploying a mass spectrometry workflow to the cloud. While "cloud computing" can have many meanings, in this case, I am simply referring to a virtual computer that is remotely accessible over the Internet. This "computer" can have as many or few resources (CPU, RAM, disk space, etc.) as your demands require and those resources can be changed as you need without requiring complete reinstalls. Systems can be easily "checkpointed" and restored. I will describe how to deploy virtualized, remotely accessible computers on which you can perform your basic mass spectrometry data analysis. This use is a quite restricted microcosm of what is available under the umbrella of "cloud computing" but it is also the (useful!) niche use for which straightforward how-to documentation is lacking.

This chapter is intended for people with little or no experience in creating cloud computing instances. Executing the steps in this chapter, will empower you to instantiate a computer with the performance of your choosing with preconfigured software already installed using the Amazon Web Service (AWS) suite of tools. You can use this for use cases that span when you need limited access to high end computing thru when you give your collaborators access to preconfigured computers to look at their data.

Key words Cloud computing, Virtual computers, Mass spectrometry, AWS, EC2, Systems management

1 Introduction

For the academic or other smaller scale mass spectrometry laboratory, it is quite typical that the control and analysis computer are one in the same. This is a suboptimal workflow. Data analysis during acquisition runs the increased risk of poorly timed computer crashes. User access to control instrumentation invites unintended configuration management issues. Similarly, analysis tasks run the range of demands for computational resources——a simple view of the TIC and manual interrogation of some spectra requires considerably less than what is required when performing proteomic searches or metabolomic feature extraction. Further, which workflows are being utilized from day to day can vary tremendously——

Arnaud Delobel (ed.), *Mass Spectrometry of Glycoproteins: Methods and Protocols*, Methods in Molecular Biology, vol. 2271, https://doi.org/10.1007/978-1-0716-1241-5_26, © Springer Science+Business Media, LLC, part of Springer Nature 2021

for those that facilitate users this can create a burdensome administration load making sure the right computers with the right resources installed with the right software are available to the myriad of users and collaborators that want to see their data.

Currently, there are several computation resource vendors that provide Internet based access to user-defined remote-accessible virtual computers. In this example, I will focus on solutions provided by Amazon thru their Amazon Web Services (AWS) offering, however similar solutions exist from Google, Oracle, and so on. As alluded above, the range of resources offered by these providers runs from specialized storage and database applications, thru load balancing distributed processing and massive data analytics. My goals for this document naturally limit my discussions to those most commonly used to augment the mass spectrometry data analysis workflow.

In this chapter, I describe how to create a virtual computer using Amazon's EC2 service. I describe virtual hard drives that you can easily attach/detach from your new virtual computer allowing for easy backups/recovery and cloning. I provide a limited discussion on security and, finally, I provide some sample use cases.

In this case, the virtual computer will be physical hardware in Amazon's computational centers that is partitioned and configured, at your guidance, to appear as any other network attached computer. Typically, usage charges will be quoted on an hourly basis but prorated to the nearest second—Amazon does have a free tier which you can use to explore the features of cloud computing. This tier is only available for 1 year on an account and it does have a number of limitations (machine configurations, etc.). The charges will vary based on the resources that you are using. More memory, more powerful CPUs, more disk space—these will all increase your hourly charges. Depending on your workflow, there can be a significant cost difference between when the computer is "turned-on" and when it is off. While you will continue to be billed for hard drives (storage) you are using, you will no longer be billed for hourly CPU costs when the computer is shutdown. As an example, as of this writing, using Amazon's Cost Calculator [1], a t3.medium instance (Intel Xeon Platinum 8000 series processor, 2 cores, 4 GB RAM) with a 30 GB SSD "primary" drive and a 500 GB magnetic "data" drive will cost ~$70/month. Of this cost, ~$3/month are for the 30 GB SSD Drive, ~$22.50/month is for the 500 GB magnetic drive, and ~$44 is for the CPU ($30) and Microsoft software licenses ($14). Thus, when the computer is "off" the CPU and licensing costs stop and you are only being billed $25.50 for the persistent data storage.

While I, for the most part, will not write about configuration optimization, it is worth noting that the above described a la carte approach to configuration means that you can choose an expensive CPU with lots of memory when doing your proteomic search but then shutting down your computer and relaunching it with less resources when you are browsing your results. Additionally, I will not be talking about licensing enforcement methodologies. For commercial software that uses network based digital restriction management this is often transparent. For commercial software that uses USB dongles or network card "MAC address" based restrictions, these can be less trivial to overcome.

Here is the general outline we will follow.

1. Get an AWS account.
2. Select and instantiate your first cloud computer.
3. Install software you will need to perform analysis.
4. Copy over your data.
5. Enjoy your new cloud computer as we explore advanced workflows such as backups and data sharing.

2 Materials

1. *Payment Method*. By far the easiest method to pay for AWS services is via major credit card. Additional payment methods are listed on Amazon's AWS website [2].
2. *Computer/Network Connection*. Your computer will be acting as a terminal into a remote computer. The computational "heavy lifting" is being performed by your AWS resources and the RDP protocol is very efficient (*see* **Note 1**). Unfortunately, there is no way to virtualize screen resolution—I recommend that any computer you use have at least a true screen resolution of 1920 × 1080 or as your mass spectrometry software suggests.
3. *Web Browser*. To access and configure AWS resources.
4. *RDP Client*. RDP is the protocol that you will need to support in order to connect to your AWS Windows instances. There are many options available depending on your operating system, here are the ones I most commonly use:
   (a) Mac/Windows: **Microsoft Remote Desktop** client.
   (b) Chromebook/Android: Microsoft's **Remote Desktop (RD Client)** client.
   (c) Linux: **remmina**.

3 Methods

3.1 Creation of an Amazon AWS Account

Create an Amazon AWS account (or log in if you have one). You may have an amazon.com account for retail purchases, this is a different account that is created to manage AWS services.

1. Navigate your browser to https://console.aws.amazon.com/. If you already have an account, proceed to Subheading 3.2, otherwise continue to create an account.
2. After you enter your **email** and **password** (twice to prove you can), there is an option to add an **AWS Account Name**. This optional account name will be an alias for your 12-digit account number. When you (or a sub-account you create) log in, either the account number or the **AWS Account Name** will need to be provided.
3. Next you are asked to provide contact information. In this dialog, there is a question if this is a **Personal** or **Professional** account. While the choice you make will change some of the requested information (most notably, the request for a *Company name*) there is no difference in features between the two accounts.
4. You will next be asked for payment information. There are alternatives beyond the scope of this document, for this example I am assuming that you will be entering credit card information.
5. Validate your contact telephone number (by SMS or call) and choose a support plan (I choose the free **Basic** plan).
6. You will now be directed back to the dialogue you originally were presented with.

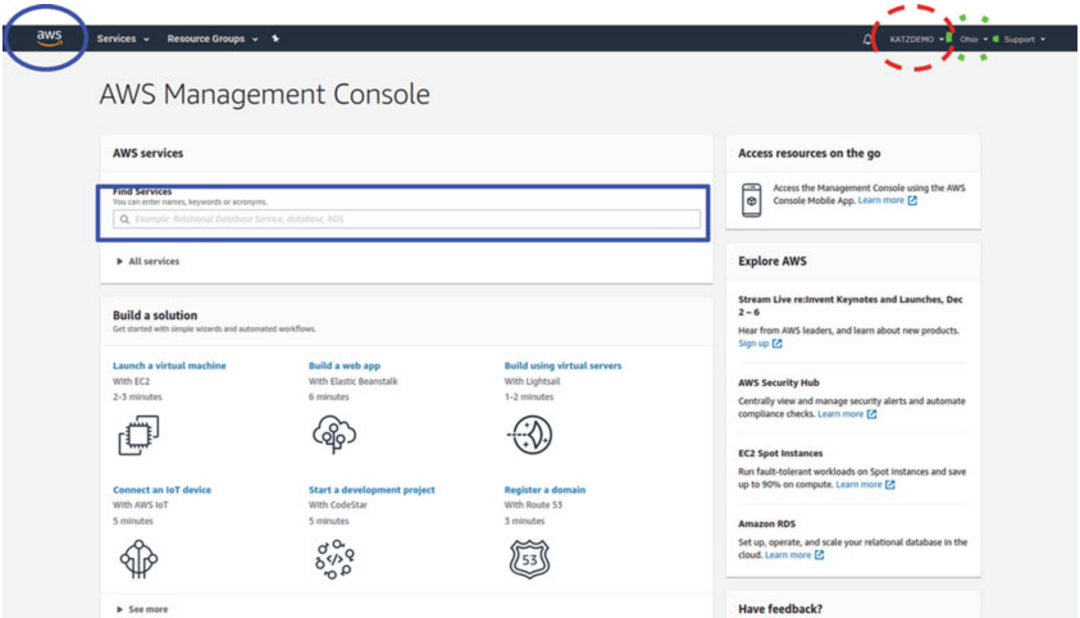
3.2 Log into Your Account

1. At this point, (assuming you have just followed the above instructions) you do not have any named subaccounts so you will need to log in with the email and password you previously provided. Amazon refers to this as **root account credentials**.
2. Please *see* Fig. 1 for an example of the interface after you have logged in.

3.3 Optional Additional Account Configurations

The following two steps describe mechanisms that allow more granular control over billing and account access from individual entities. This can be done at any time, so, if unimportant for now, proceed to Subheading 3.4.

1. An **organization** allows you to have several administratively distinct AWS accounts that are tied together with shared billing and subject to organization wide policy controls. The organization managers can see the breakdown of the charges by account number which can be used for accounting or chargeback purposes. An example use case would be a chemistry department that wants individual labs to be able to manage their own access and resource deployment.
2. In the upper right menu, click the tab that matches your account name (Fig. 1, **red dashed circle**). In the pull down, select “My Organization.” There will be a large dialog displaced that is asking you to “create organization.” Click this.



AWS Management Console

AWS services

Find Services

You can enter names, keywords or acronyms.

Example: Relational Database Service, database, RDS

▸ All services

Build a solution

Get started with simple wizards and automated workflows.

Launch a virtual machine
With EC2
2-3 minutes

Build a web app
With Elastic Beanstalk
6 minutes

Build using virtual servers
With Lightsail
1-2 minutes

Connect an IoT device
With AWS IoT
5 minutes

Start a development project
With CodeStar
5 minutes

Register a domain
With Route 53
3 minutes

▸ See more

Access resources on the go

Access the Management Console using the AWS Console Mobile App. Learn more

Explore AWS

Stream Live re:Invent Keynotes and Launches, Dec 2 – 6

Hear from AWS leaders, and learn about new products. Sign up

AWS Security Hub

Centrally view and manage security alerts and automate compliance checks. Learn more

EC2 Spot Instances

Run fault-tolerant workloads on Spot Instances and save up to 90% on compute. Learn more

Amazon RDS

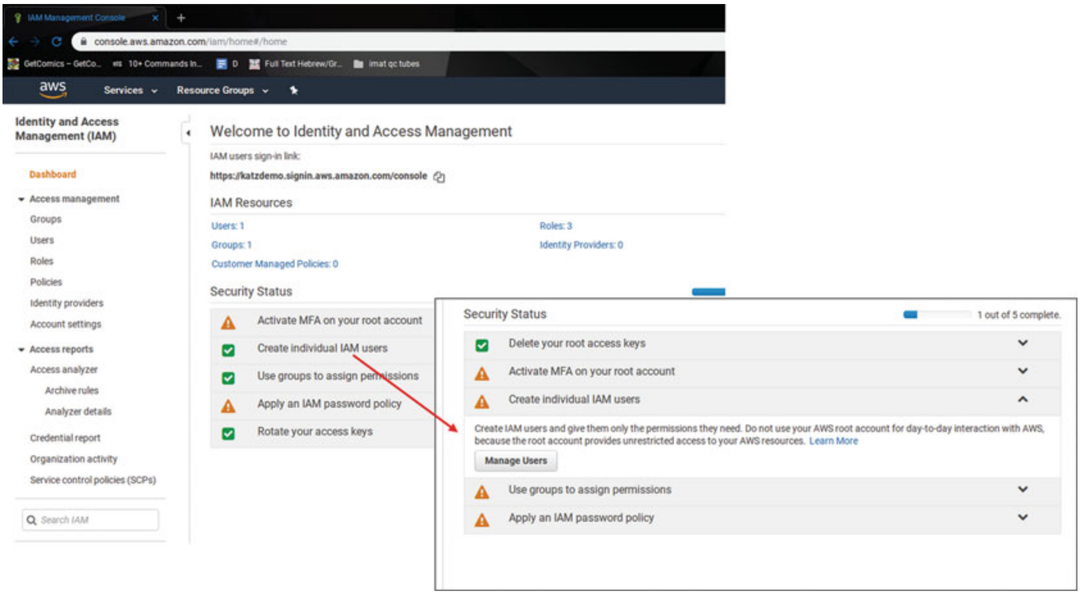
Set up, operate, and scale your relational database in the cloud. Learn more

Have feedback?

Fig. 1 The AWS Primary Console Screen. The AWS primary console screen has several dynamic portions including the main body and the side bars. Consistent will be the title bar along the top. You can always return to this view by clicking the AWS logo in the top left indicated by the **blue solid circle**. From this screen you can quickly navigate to other functional views by searching for the appropriate subsystem in the search box indicated by the **blue solid rectangle**. Other features of interest include your account management pulldown indicated by the **red dashed circle** and your current region indicated by the **green dotted circle**

3. After you have validated your email and your billing information you will then be able to either invite existing AWS members into your organization or create new accounts within your organization.
4. These are distinct administrative groups each of which can control its own access controls, set their own policies, etc. They are subject to the global organization policies.
5. The Amazon **Identity and Access Management (IAM)** service allows you to create additional accounts (subject to differential security policies) under a single AWS account. The login credentials used to create the AWS account are the “root” credentials for that account. It is a recommended best practice to only use the root credentials for specific administrative tasks—for general use, it is recommended to use IAM to create an alternate login to your AWS account. These IAM logins can be governed by defined policies on a group or individual basis. An example use case would be a chemistry laboratory that wants individual research scientists to be able to instantiate their own compute resources subject to the policies of the laboratories AWS account manager.

6. If you are not already logged in, navigate to https://console.aws.amazon.com/ and log in with your AWS root credentials (the email/password you used to create the AWS account).
7. If you have not already done so, you will need to enable IAM access to billing data. This will be required for the creation of IAM accounts with administrative access.
8. In the top right of the menu bar is your account name (Fig. 1, **red dashed circle**), click on that and in the pull-down menu, choose **My Account**.
9. In the plethoric catalog of choices, about half-way down, will be a section titled **IAM User and Role Access to Billing Information**. Click **edit**.
10. Click the checkbox to enable access and then click **Update**.
11. Now navigate to https://console.aws.amazon.com/iam/.
12. In the center dialog box, click the inverted chevron associated with "**Create individual IAM users**" then click **manage users** (*see* Fig. 2).
13. Click on the "**add user**" button. You will be asked to provide account authentication information. Commonly, and especially if this is your first **IAM** user on this account, you will probably want to select the option for **Access Type** as AWS **Management Console Access**.



Welcome to Identity and Access Management

IAM users sign-in link:
https://katzdemo.signin.aws.amazon.com/console

IAM Resources

Users: 1
Groups: 1
Customer Managed Policies: 0
Roles: 3
Identity Providers: 0

Security Status

- ⚠ Activate MFA on your root account
- ☑ Create individual IAM users
- ☑ Use groups to assign permissions
- ⚠ Apply an IAM password policy
- ☑ Rotate your access keys

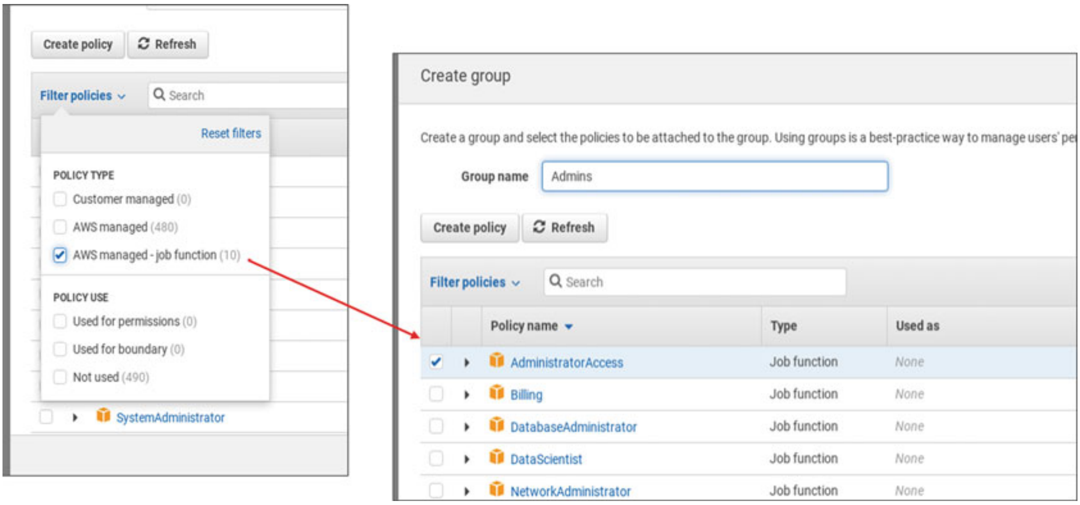
Security Status 1 out of 5 complete.

- ☑ Delete your root access keys
- ⚠ Activate MFA on your root account
- ⚠ Create individual IAM users

Create IAM users and give them only the permissions they need. Do not use your AWS root account for day-to-day interaction with AWS, because the root account provides unrestricted access to your AWS resources. Learn More

- ⚠ Use groups to assign permissions
- ⚠ Apply an IAM password policy

Fig. 2 Adding IAM Users. From the main IAM console, clicking on the chevron next to the header Create individual IAM users will prompt you with a dialogue asking if you want to manage users



POLICY TYPE

□ Customer managed (0)

□ AWS managed (480)

☑ AWS managed - job function (10)

POLICY USE

□ Used for permissions (0)

□ Used for boundary (0)

□ Not used (490)

□ SystemAdministrator

Create group

Create a group and select the policies to be attached to the group. Using groups is a best-practice way to manage users' pe

Group name: Admins

	Policy name	Type	Used as
☑	AdministratorAccess	Job function	None
□	Billing	Job function	None
□	DatabaseAdministrator	Job function	None
□	DataScientist	Job function	None
□	NetworkAdministrator	Job function	None

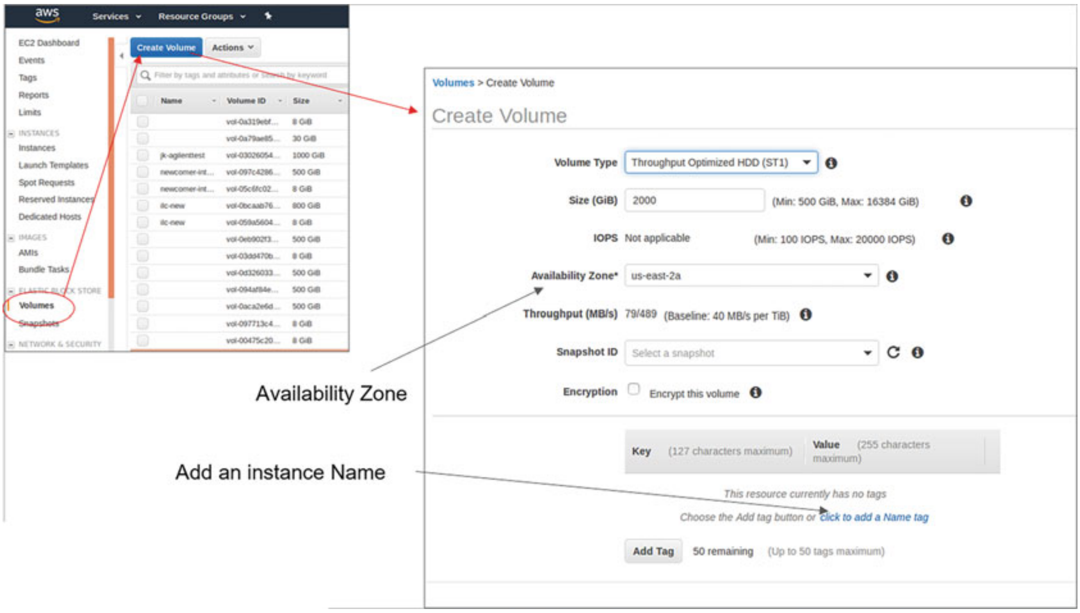
Fig. 3 Configuring permission credential groups for IAM users. When you create a permission group for IAM users, there is an exhaustively detailed collection of individual permissions that can be allowed or denied. To simplify administrative choices, common collections of permissions are group together. Show is the selection of AWS managed policies by job function and then they selection of the AdministratorAccess job. This particular selection will create a group with full permissions

14. You will now be asked to set the permissions for the user. The default case creates a user with no permissions at all which can be stifling. We will create our first account with full control. Start by clicking **Create Group**.
15. If you click on the chevron next to **Filter Polices** a small dialog will open, from there select **AWS managed - job function**. *See* Fig. 3.
16. There will now be about 10 listings. Select the checkbox next to **AdministratorAccess**, add a name in the text box next to **Group name** and then click on the **Create group** button. You will want to sure the group you have created is selected before proceeding. As you learn EC2 you will want to explore restrictions you may want to put in place.
17. Next step asks if you want to assign **tags** to the account. Tags are for your benefit and can be used for internal auditing or clustering of your accounts. It is ok to leave these blank.
18. Finally confirm all your settings and your new **IAM** account has been created.
19. Before leaving the **IAM** console note that near the top there is a direct link to log into your account. There is an option to the right of that link that allows you to **customize** that link. It is an alias for the IAM login.
20. Try logging out of your **root credential** account and try logging into your **IAM** account. Navigate to https://console.aws.amazon.com/ and enter your account ID or account alias to continue to the IAM login.

3.4 Create Your First Data Drive

With hard drives, there is an obvious logical divide between the drive one uses for operating system and installed software and the drive one uses for actual “data” files on which you are going to operate. For a lot of workflows, you might want to have completely different operating system and software configurations, but, typically, your data will remain a constant. In keeping with this mindset, while, perhaps, less intuitive, I will first describe the creation of a virtual hard drive (*see* **Note 2**) to hold your data rather than on the OS drive. Note that creating additional storage is not part of Amazon’s free tier, it is ok to skip this step and perform the creation and the attachment later.

1. An easy way to navigate the AWS console is to click on the **AWS** logo in the upper left of the title bar (Fig. 1, **blue circle**) and then type the desired service in the search box (Fig. 1, **blue rectangle**). In this case, we are going to type **EC2** and press enter.
2. In the upper right of the title bar, a location (“availability zone,” Fig. 1, **dotted green circle**) will be listed. When you provision services they will be instantiated in a data center in that locale. There are some optimization choices that can be made here (e.g., some services are not available in all locales and the same service can be priced *modestly* differently in different locales), but a good rule of thumb is to choose a location that is closest to your physical location—Amazon has not quite solved the “speed of light” issue and so there will be an imperceptible timing difference if you choose a more distal data center. What *is* important is that your data drives and your virtual computers are created in the same availability zones (i.e., the same functional portion of the data center—I will be walking through this).
3. There is a menu on the left-hand side of your screen. In that menu, there is a section **Elastic Block Storage**. Under that heading, select **Volumes** (think “virtual hard drive”).
4. Click on the button **Create Volume.**
5. You will be presented with several options; let us walk through a few (which I have also reflected in Fig. 4).
6. For a data drive, a spinning physical disk will be cheaper for the same amount of storage but have less performance than an SSD drive. I typically select **Throughput optimized HDD**.
7. Now set the size at **500** GiB (feel free to go higher or lower as your needs require.
8. **Important:** note your availability zone .... I choose **us-east-1a** for this example, but that does not matter .... What *does* matter is making sure you choose the same zone when creating your virtual computer.



Name	Volume ID	Size
	vol-0a319ebf...	8 GiB
	vol-0a79ae85...	30 GiB
jk-agilenttest	vol-03026054...	1000 GiB
newcomer-int...	vol-097c4286...	500 GiB
newcomer-int...	vol-05c6fc02...	8 GiB
ilc-new	vol-0bcaab76...	800 GiB
ilc-new	vol-059a5604...	8 GiB
	vol-0eb902f3...	500 GiB
	vol-03dd470b...	8 GiB
	vol-0d326033...	500 GiB
	vol-094af84e...	500 GiB
	vol-0aca2e6d...	500 GiB
	vol-097713c4...	8 GiB
	vol-00475c20...	8 GiB

Volumes > Create Volume

Create Volume

Volume Type: Throughput Optimized HDD (ST1)

Size (GiB): 2000 (Min: 500 GiB, Max: 16384 GiB)

IOPS: Not applicable (Min: 100 IOPS, Max: 20000 IOPS)

Availability Zone*: us-east-2a

Throughput (MB/s): 79/489 (Baseline: 40 MB/s per TiB)

Snapshot ID: Select a snapshot

Encryption: □ Encrypt this volume

Key (127 characters maximum) | Value (255 characters maximum)

This resource currently has no tags

Choose the Add tag button or click to add a Name tag

Add Tag 50 remaining (Up to 50 tags maximum)

Availability Zone

Add an instance Name

Fig. 4 Creating an EBS Data Volume. This figure shows key features of the AWS user interface with which you will interact when creating an EBS data volume. Please note the locations where you select the availability zone as well as where you can provide a name to your virtual hard drive

9. **Snapshot ID** is useful, but not for your first hard drive. In the future, you might create a data drive filled with lots of good resources. If you take a snapshot of that drive, you can then create new "hard drives" based on these snapshots. I will be going through this workflow a bit later.
10. **Key/Value** pairs are arbitrary tags that you can use to organize your volumes. The same mechanism is used for your virtual computers. I *strongly* urge you to name all of your virtual hard drives and computers. If you create a **Name** key with a value descriptive to you, that name will appear in the listings of resources. It is so useful to add this tag that you will note, below the list of entries, a link that says "**click here to add a Name tag**" prompting you to do so if you have not already. For this example, I added the tag "MS Data Disk 001" and then clicked **Create Volume** for which I was rewarded with a "Volume created successfully" notice and a volume ID which I did not bother recording since I remembered to add a name tag. The drive you created will appear as unformatted when you attach it to your virtual computer. This is easy to remedy and we will walk through this later, but if you do not recognize that the drive starts as unformatted, it can be difficult to troubleshoot why your new drive is not working.
11. You can now click the **Close** button. You will now be back at the "volumes" screen and should see your new volume. Note

that the name tag content appears in the name column for the volume. If you had not named the volume this can get quite confusing when you have more than one of them.

12. To delete your volume (volumes cost money!), make sure you are on the volumes management screen (if not: click **AWS** in the title bar (Fig. 1, **blue circle**), search **EC2** (Fig. 1, **blue rectangle**), select **Volumes** under the **Elastic Block Storage** of the left menu). Now select your volume and select **Actions** → **Detach Volume** (assume it is attached to an instance), then select **Actions** → **Delete Volume**. Please note that this is an irrevocable action. Note that any data on the volume reflected in a snapshot will be retained, so this will not reduce your storage costs unless you delete snapshots you created as well.

3.5 Creation of Computer Credentials

Beyond your AWS login credentials, you need to create credentials to log into the computer instances you create. AWS credentials allow you to log into the AWS control panel. Computer credentials allow you to log into the computer you created. AWS delivers computer login credentials to you via a public/private key mechanism. This workflow is shown in Fig. 5.

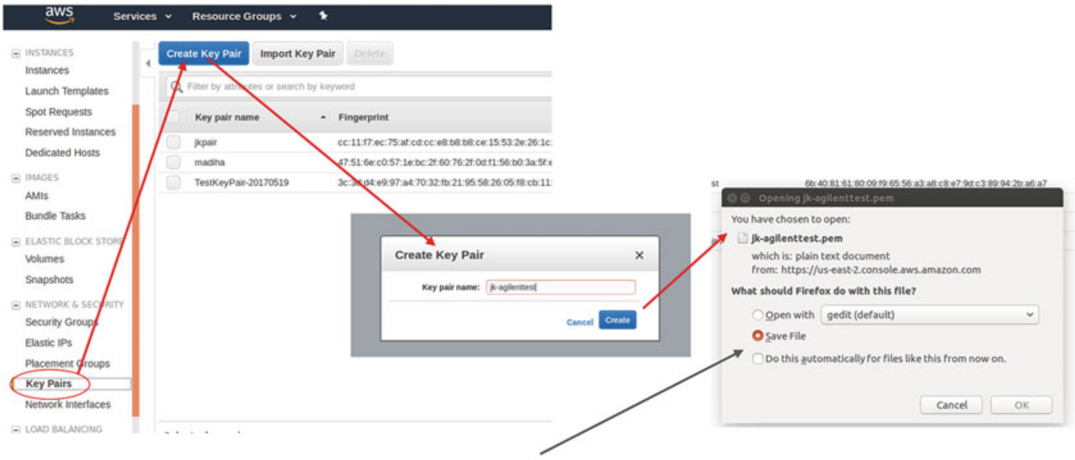
1. If not already on the EC2 page, click the **AWS** logo in the title bar (Fig. 1, **blue circle**), then search for **EC2** (Fig. 1, **blue rectangle**), then on the left hand side menu, scroll down until you hit the **Network and Security** section, in there, click **Key Pairs**.
2. Click **Create Key Pair**. Name your pair informatively like "keypair1" then click **Create.** Your browser will automatically download a file that matches the key pair you just created—in this case, "keypair1.pem" should have been downloaded.
3. Now, when you create your compute instances, or other resources that require authentication you can reference this keypair (*see* **Note 3**).

3.6 "Provision" Your First Compute Instance

You will now "provision" your first compute instance. This workflow is shown in Fig. 6.

Click on the **Launch Instance** button, if you do not see the **Launch Instance** button, click on the **AWS** in the left side of the title bar (Fig. 1, **blue circle**), search for **EC2** (Fig. 1, **blue rectangle**), press enter. You will either see the **Launch Instance** button, or you can click the **Instances** item in the menu that is on along the left side of your dashboard (Fig. 6).

1. After you have started the launch dialog, your first choice will now be which operating system you want to have "installed" on your new instance. I have had success with Microsoft **Windows Server 2008 R2 Base** (64bit). All the Microsoft



Key pair name	Fingerprint
jkpair	cc:11:f7:ec:75:af:cd:cc:e8:b8:b8:ce:15:53:2e:26:1c:
madiha	47:51:6e:c0:57:1e:bc:2f:60:76:2f:0d:f1:56:b0:3a:5f:
TestKeyPair-20170519	3c:3f:d4:e9:97:a4:70:32:fb:21:95:58:26:05:f8:cb:11:

Create Key Pair

Key pair name: jk-agilenttest

Cancel Create

Opening jk-agilenttest.pem

You have chosen to open:

jk-agilenttest.pem

which is: plain text document

from: https://us-east-2.console.aws.amazon.com

What should Firefox do with this file?

○ Open with gedit (default)

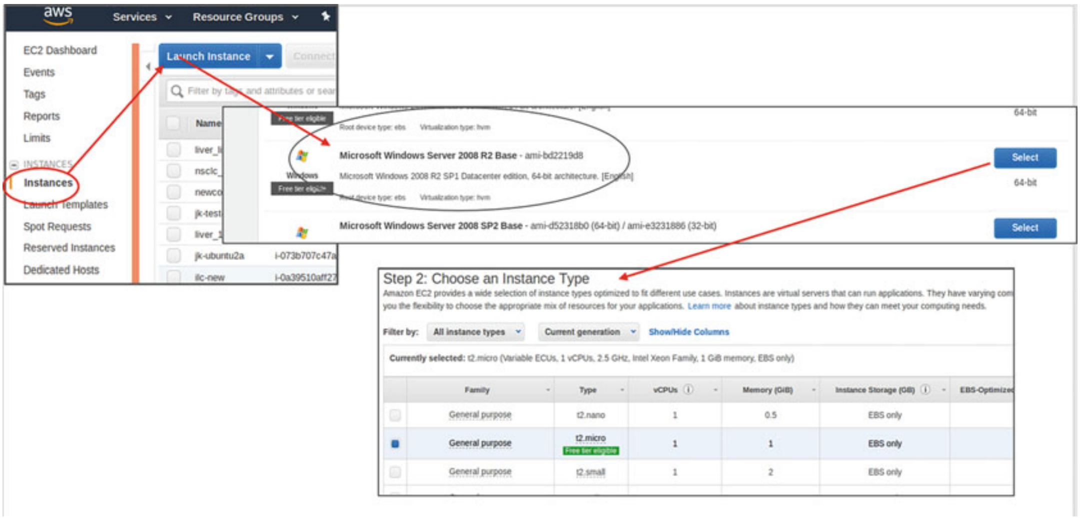
● Save File

□ Do this automatically for files like this from now on.

Cancel OK

It is **IMPORTANT** to save this file.

Fig. 5 Creating a Key Pair. This figure shows key features of the AWS user interface with which you will interact when creating a Key Pair. It is important that you save the produced .pem file (your file save dialogue may be different than that shown)



Root device type: ebs Virtualization type: hvm

Microsoft Windows Server 2008 R2 Base - ami-bd2219d8

Microsoft Windows 2008 R2 SP1 Datacenter edition, 64-bit architecture. [English]

Root device type: ebs Virtualization type: hvm

Microsoft Windows Server 2008 SP2 Base - ami-d52318b0 (64-bit) / ami-e3231886 (32-bit)

Step 2: Choose an Instance Type

Amazon EC2 provides a wide selection of instance types optimized to fit different use cases. Instances are virtual servers that can run applications. They have varying com you the flexibility to choose the appropriate mix of resources for your applications. Learn more about instance types and how they can meet your computing needs.

Filter by: All instance types Current generation Show/Hide Columns

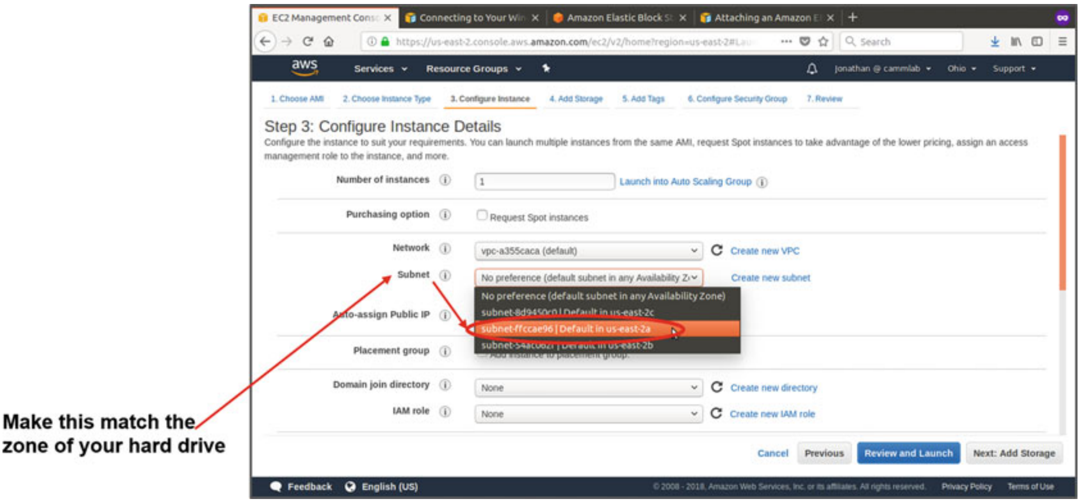
Currently selected: t2.micro (Variable ECUs, 1 vCPUs, 2.5 GHz, Intel Xeon Family, 1 GiB memory, EBS only)

	Family	Type	vCPUs	Memory (GiB)	Instance Storage (GB)	EBS-Optimized
□	General purpose	t2.nano	1	0.5	EBS only	
■	General purpose	t2.micro Free tier eligible	1	1	EBS only	
□	General purpose	t2.small	1	2	EBS only	

Fig. 6 Instantiating your Compute Instance - Feature Selection. This figure shows key features of the AWS user interface with which you will interact when selecting the operating system and compute hardware for your compute instance

instances will be of the server variety, this one does have the feel of Windows 7 which works well with many of the mass spectrometry software needs (which tend to be a generation behind in OS). If you need Windows 10, you can select an appropriate alternative.

2. Your next choice will be the hardware resources available to your instance. For this example, I will choose **t3.large** with its 2 vCPUs and 8 GB of ram; you should choose a configuration



Step 3: Configure Instance Details

Configure the instance to suit your requirements. You can launch multiple instances from the same AMI, request Spot instances to take advantage of the lower pricing, assign an access management role to the instance, and more.

Number of instances: 1

Purchasing option: □ Request Spot instances

Network: vpc-a355caca (default)

Subnet: No preference (default subnet in any Availability Z

- No preference (default subnet in any Availability Zone)
- subnet-8d9450c0 | Default in us-east-2c
- subnet-ffccae96 | Default in us-east-2a
- subnet-54ac062f | Default in us-east-2b

Auto-assign Public IP

Placement group: Add instance to placement group.

Domain join directory: None

IAM role: None

Make this match the zone of your hard drive

Fig. 7 Instantiating your Compute Instance—Availability Zone Selection. In step 3, make sure that you choose the subnet that matches the Availability Zone of any storage resources that this instance will need to connect to

that matches your needs. Please *see* **Note 4**. After selecting, click **Next: Configure Instance Details**.

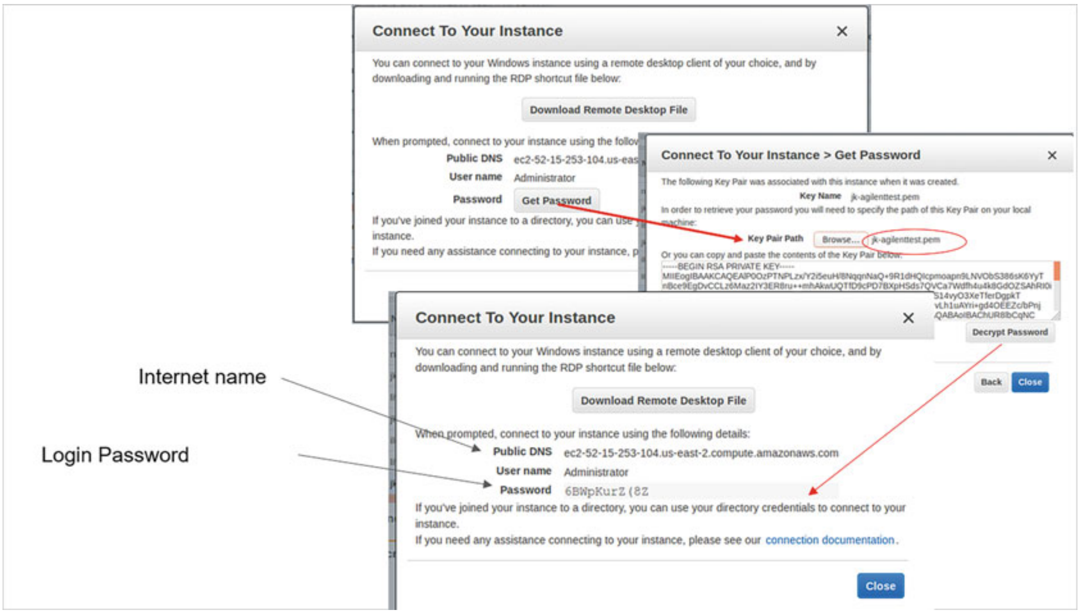
3. On the instance details page, for your purposes, the only crucial parameter is the **Subnet**, click on that and choose the availability zone that matches the data drive you created above (*see* Fig. 7). I will choose **us-east-1a** to match the choice I made for the hard drive storage. Do not feel compelled to choose this same zone—choose the one near you—just make sure it matches the zone in which your data drive resides. Now click **Next: add storage**.
4. You are now asked to create the initial drives for your instance. I tend to think of these as the operating system volumes. You are creating the C: drive which will hold the Windows operating system and probably and software you install. I will be choosing the default **30 GB SSD** as my option and then clicking **Next: Add Tags**. By default, this drive will have the same name as the "instance name" which means if you do not name your instance this will be blank. Regarding size, later on I will show you how to expand the size of your OS volume, however, if you know ahead of time that you will need more storage, feel free to select that now. Realistically, I typically run my root volume as **100 GB**.
5. As with the hard drive storage, Key/Value pairs are arbitrary tags that you can use to organize your compute instances. Click on "**click here to add a Name tag**" prompting you to do so if you have not already. For this example, I added the tag "**MS_Mach_001**" and then clicked **Next: Configure Security Group**.

6. Feel free to just click **Review and Launch**. For **step 6**, you are asked to create the security group. Imagine that you are configuring a virtual firewall. The default is all outgoing connections and incoming RDP connections are allowed. If you wish to explore this, for example, to restrict access to specific machines or to allow your computer to act as a web server, please *see* **Note 5**.
7. Finally, you will be asked to review your choices and click **Launch**. At this point, you will be prompted for they keypair you want to use. *Make sure you have a copy of the .pem file for the keypair you select*. You can also choose to create a new key pair. Acknowledge you have access to the .pem file and click **Launch Instances**.
8. You will be told the instance is being created and invited to click **View Instances**.
9. To delete your instance (instances cost money!), make sure you are on the instances management screen (if not: click **AWS** in the title bar, search **EC2**, select **Instances** under the **Instances** section of the left menu). Now select your instance and select **Actions** → **Instance State** → **Terminate**. Please note that this is an irrevocable action. Also note that hard drives that were created as part of creating this instance (e.g., the operating system hard drive) will also be destroyed. So if you want to preserve that for some reason, you should make a snapshot first.

3.7 Connecting to Your New Cloud Computer

Based on the default configuration of your new computer and the default security rules, you can only connect to your new Windows computer using the RDP protocol. You will also need the .pem file associated with the keypair used to create the instance.

1. If you are not on the page showing your instances, click **AWS** in the title bar, then search for **EC2** then, on the left menu, click **Instances** in the **Instances** section.
2. Click the checkbox next to your instance and then click the box labeled **Connect**.
3. You will now be presented with two options (Fig. 8). You are prompted to get an easy to use RDP file, and you are prompted to get the administrator password for your instance. Also, you are informed of the IP name/address of your new compute instance. If your RDP client can use the RDP file, that is great. If not, just write down the IP name of your computer. Now click **Get Password**.
4. You will now be asked for the .pem file that is associated with your key pair. **Choose File** allows you to select that file, then you can **Decrypt Password** to get the login password for your



Connect To Your Instance

You can connect to your Windows instance using a remote desktop client of your choice, and by downloading and running the RDP shortcut file below:

When prompted, connect to your instance using the follow

Public DNS ec2-52-15-253-104.us-eas

User name Administrator

Password Get Password

If you've joined your instance to a directory, you can use instance.

If you need any assistance connecting to your instance, p

Connect To Your Instance > Get Password

The following Key Pair was associated with this instance when it was created.

Key Name jk-agilenttest.pem

In order to retrieve your password you will need to specify the path of this Key Pair on your local machine:

Key Pair Path jk-agilenttest.pem

Or you can copy and paste the contents of the Key Pair below:

```
-----BEGIN RSA PRIVATE KEY-----
MIIEogIBAAKCAQEAlP0OzPTNPLzx/Y2i5euH/8NqqnNaQ+9R1dHQlcpmoapn9LNVObS386sK6YyT
nBce9EgDvCCLz6Maz2IY3ER8ru++mhAkwUQTfD9cPD7BXpHSds7QVCa7Wdfh4u4k8GdOZSAhRI0i
S14vyO3XeTferDgpkT
vLh1uAYri+gd4OEEZc/bPnj
QABAoIBAChUR8bCqNC
```

Connect To Your Instance

You can connect to your Windows instance using a remote desktop client of your choice, and by downloading and running the RDP shortcut file below:

When prompted, connect to your instance using the following details:

Public DNS ec2-52-15-253-104.us-east-2.compute.amazonaws.com

User name Administrator

Password 6BWpKurZ(8Z

If you've joined your instance to a directory, you can use your directory credentials to connect to your instance.

If you need any assistance connecting to your instance, please see our connection documentation.

Fig. 8 Connecting to your Instance. You will be using an RDP client to connect to your instance. In order to do this, you will need the IP hostname and the administrator password. This exemplar shows what that dialogue looks like and where that information may be obtained

new compute instance (Fig. 8). If your RDP client supports this, not that you can copy the password to the clipboard by clicking the icon to the right of the displayed password.

5. Open your RDP client and choose to connect to the IP address/name associated with instance. Log in as administrator and use the password you decrypted above. Woo Hoo!
6. I tend to do things at this point for my convenience and taste, like, download Firefox. Note that in your initial install, the security is set such that the Microsoft web browser will prompt you to go to any website that is not part of its trusted domains. Either lower the security setting, or just keep adding Internet addresses as prompted. Also, if you are not used to using Windows Server, you will encounter minor differences, such as the security mentioned above and be prompted for reasons when you do shutdowns.

3.8 Attach and Configure Your Data Drive

Note that you can attach volumes to compute instances that are not running.

1. Click on **Volumes** under the **Elastic Block Store** in the left hand menu—if it is not there, click the **AWS** logo in the title bar, search for **EC2** and then click enter. You should be presented with a menu of all your cloud based hard drives called *volumes*. If you have done nothing other than these

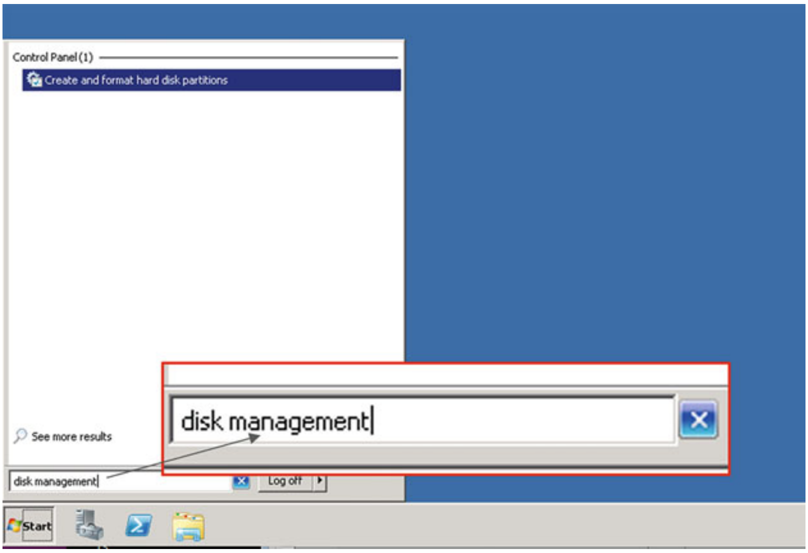
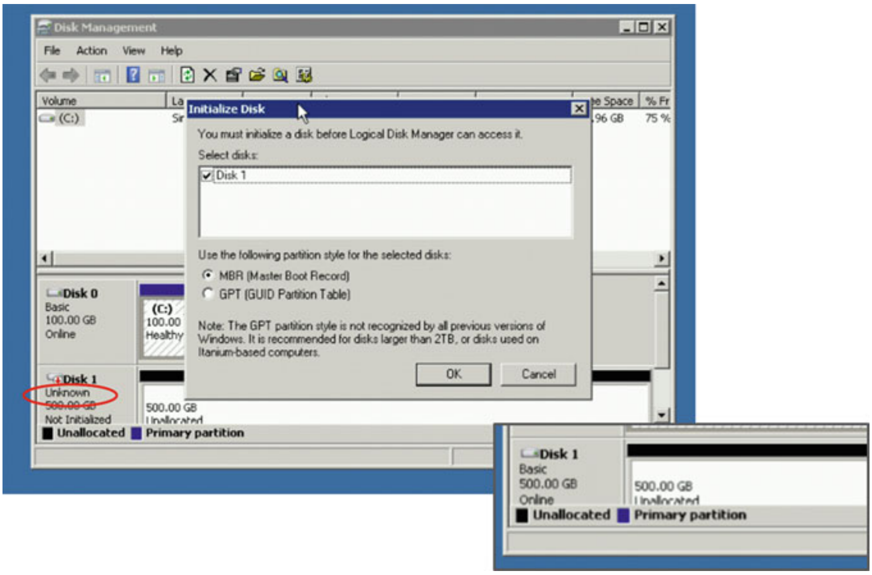


Fig. 9 Starting the Windows Disk Management Tool. Click the Windows button in the lower left, then type "disk management" and press enter

instructions, you should have one volume named after your compute instance that is shown as "attached" and you should have the MS-Data-01 drive you created marked as "available."

2. Check the **MS-Data-01** drive, click on **Actions** and then choose to **Attach Volume**. You now see the joy of naming everything you do descriptively! If you click on the instance, you can type the name of the machine you created or select it from the dialog it prompts you with. Then click **Attach**. You will be sent back to the volume page with a prompt showing that efforts are being undertook to attach your volume to your instance. Remember, if this device has not been configured, your compute instance will treat it as an attached, unformatted hard drive. **Important:** A volume can only be attached to one instance at a time and you will have to detach a volume from an instance before you can delete it.
3. Format your virtual hard drive. **Important:** this only needs to be done *once* when the drive is first attached and before you use it. Switch over to your windows instance, click on the windows start button in the lower left corner. Type **disk management** and press enter (*see* Fig. 9). You should be presented with the windows disk management dialog which will show your functional "C" drive (probably as Disk 0) and the "Unallocated" "D" drive (probably as Disk 1).
4. We now need to initialize (format) and partition the hard drive so Window can use it. When you start disk management, it may present you with the initialization dialog immediately. If not, right-click on the "Unknown" volume and select **Initialize Disk.**



You must initialize a disk before Logical Disk Manager can access it.

Select disks:

☑ Disk 1

Use the following partition style for the selected disks:

● MBR (Master Boot Record)

○ GPT (GUID Partition Table)

Note: The GPT partition style is not recognized by all previous versions of Windows. It is recommended for disks larger than 2TB, or disks used on Itanium-based computers.

OK Cancel

Fig. 10 Initializing Your Storage Device. Typically, you will be presented with this dialog when you start "disk management" and there is an uninitialized storage device attached—these will be shown as "unknown" devices as shown by the red circle. If you cancel the "Initialize" dialog or otherwise want to open it, you can right-click on the "unknown" disk and select initialize

5. Select **MBR** and select ok (*see* Fig. 10); there may be cases when you should choose **GPT**, *see* **Note 6** for a discussion. The drive will switch status from "unknown" to "basic."
6. Now we will partition the drive and add filesystems. Right-click on the "Unallocated" drive and choose "**New Simple Volume**. . . .".
7. Follow the wizard choosing all default steps (unless you are wiser). You will end up with a single partition hard drive, formatted as NTFS and allocated as the D: drive (assuming that is what you assigned).
8. It is strongly recommended that you store all data and result files on this data disk. Unless you take active action, the C drive is ephemeral and will vanish if you ever terminate (vs. stop) your compute instance.

3.9 Transferring Data to Your New Computer (Including Vendor Software)

The general workflow that has worked for me is to use another vendor that has addressed the "data sharing" problem. Upload your data or install files from your control to desktop computer and then access them on your compute instance.

1. **Google File Stream** allows you to present your Google based storage as a separate drive device (e.g., "g:"). You can get the download here https://support.google.com/a/answer/7491144?hl=en. While this works incredibly well, I tend to

recommend that you change your cache drive from the default C: drive to your D: data drive as the C: drive will probably be too small (*see* **Note 7** for instructions).

2. **Dropbox** also works quite well. It presents your files as part of your home folder within Windows. Dropbox presents files as part of your file tree rather than as a separate device. I find this to be less to my liking with regards to keeping data/volumes and files cleanly managed.
3. At this point I will also install my mass spectrometry analysis software. Online resources (e.g., "R") I will get online. If I have install files from a vendor, I will use Dropbox or Google File-Stream to import the vendor install files.

3.10 Creating Snapshots

Think of a snapshot as a complete backup/clone of one of your hard drives in its current state. These can, of course, be used for data recovery, but they can also be used to build new cloned hard drives (possibly with different sizing). If you created an operating system volume when you created your instance, be aware it will be destroyed when your instance is destroyed. A snapshot of your operating system volume will allow you to clone your installation in the future as well as access data that may have been stored on that volume.

1. If you are cloning the hard drive that contains your operating system you will want to stop the instance first. In general, you only want to snapshot a volume that is not actively getting written to. For example, do not snapshot your data volume while you are doing a Mascot search.
   (a) Log into the instance, click the **windows button** and select **shutdown**.
   (b) -OR- From the AWS console, click **AWS** in the title bar, search for **EC2**, select **Instances** on the left menu in the instances section. Find your instance in the main panel (you did remember to use good names, yes?), select it, then click **Actions** → **Instance State** → **Stop**.
2. I tend to perform snapshots on the volume screen. If you are not there already, click **AWS** in the title bar, search **EC2**, select **Volumes** under the **Elastic Block Storage** of the left menu.
3. Now select the volume for which you want to create a snapshot and then click **Actions** → **Create Snapshot** (*see* **Note 8** regarding snapshot costs).
4. To Delete a snapshot, navigate to the snapshots management screen by clicking **AWS** in the title bar, search **EC2**, select **Snapshots** under the **Elastic Block Storage** of the left menu. Select the snapshot(s) you wish to delete, then click **Actions** → **Delete**.

3.11 Workflow Examples

In the following examples, I will describe some common tasks I have done that leverage the advantages of a workflow in the cloud putting together the processes I described above. These workflows take advantage of the fact that cloning hard drives, creating snapshots of hard drives and moving hard drives between computers of different configurations (beefy vs. not) are all *trivial* on the cloud and executed via simple mouse clicks.

3.11.1 Clone Your OS Drive and Create a New Compute Instance from it

1. Shutdown your compute instance (Subheading 3.10). **Important:** Note the password required to log into this instance – in these instructions, I will tell you how you can swap out operating system "hard drives." The password administration configuration files will remain associated with the snapshot of the operating system hard drive.
2. Create a snapshot of your OS drive (Subheading 3.10). **Important:** Write down or copy the snapshot ID of the snapshot you wish to use as the basis for your cloned OS drive (and you did save the login password, from above, yes?). Note that the administration password used during the creation of this instance will be the same password you use for any instance you create in the future for which you make this the operating system drive.
3. Create a new volume (Subheading 3.4—**Important**: when you get to Subheading 3.4, **step 9**, you will choose the snapshot you created above—This is where having written down the ID or copying it to your clipboard is really useful). If your volume is larger than your snapshot, we will have to address this in Subheading 3.11.2 below.
4. Create a new instance (Subheading 3.6). Often, I will create an instance and do software installs with less CPU and RAM than is required when doing analysis. When creating the new instance, you can scale the CPU or RAM as you see fit. When you create an instance, it comes with an attached new storage device we will have to remove.
5. Now shut the instance down (Subheading 3.10, **step 1**).
6. Detach (and delete) the volumes that were created when you created your new instance (Subheading 3.4, **step 12**).
7. Attach the volume you created in Subheading 3.11.1, **step 3** above. To do this, click **AWS** in the title bar, search **EC2**, select **Volumes** under the **Elastic Block Storage**. Select the drive you want to attach as the operating system drive and select **Actions** → **Attach Volume**. Choose your stopped instance by name or id. For device.... **IMPORTANT :** Type in **/dev/sda1**. Yes, this is documented, no not in an obvious places.

8. Now restart your instance. From the instances main panel (if not there, click **AWS** in the title bar, search for **EC2**, select **Instances** on the left menu in the instances section) select the instance, then choose **Actions** → **Instance State** → **Start** and connect as normal.

3.11.2 Increase the Size of Your OS Volume

In following the steps of Subheading 3.11.1 above, at **step 3**, it is possible you may have created your new instance with a larger volume size than that from which you created your snapshot. If this is the case, Windows will still think your volume is the size when the snapshot was originally created (as referenced in Subheading 3.11.1, **step 3**). Here is one way to resolve this.

1. Connect to your instance (Subheading 3.7).
2. Press the windows key, type **cmd**, right-click on the "cmd" app and select "run as administrator" (*see* Fig. 11).
3. Type **diskpart** in the command shell.
4. At the "diskpart" prompt type list volume, note the volume number of the volume you want to expand. If this is the system volume, it will most likely be volume 0.
5. Type select volume 0 and, if needed, replace 0 with the actual volume you want to expand.

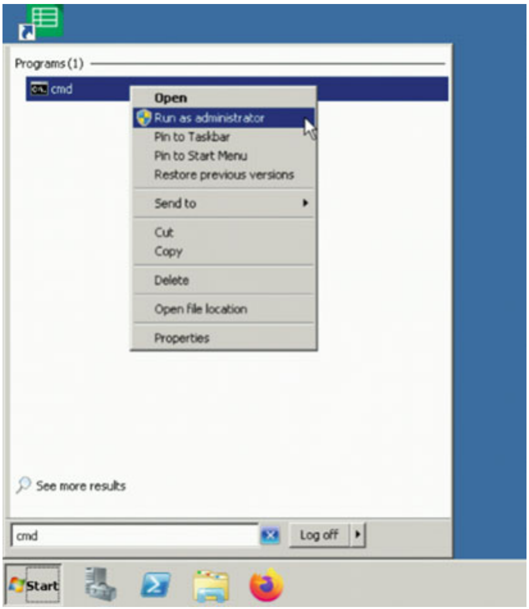
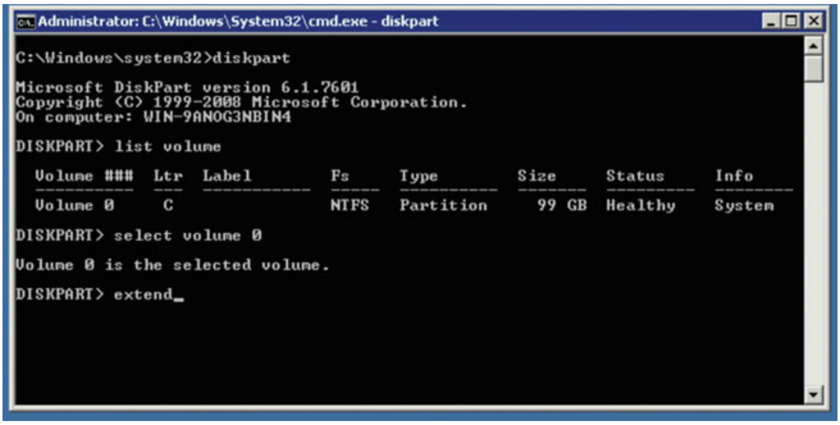


Fig. 11 Starting an "Administrator" Command Shell. Click the Windows button in the lower left, then type "cmd", **do not** press enter, but rather right-click on the "cmd" entry and choose "Run as Administrator"



```
C:\Windows\system32>diskpart

Microsoft DiskPart version 6.1.7601
Copyright (C) 1999-2008 Microsoft Corporation.
On computer: WIN-9ANOG3NBIN4

DISKPART> list volume

  Volume ###  Ltr  Label        Fs     Type        Size     Status     Info
  ----------  ---  -----------  -----  ----------  -------  ---------  --------
  Volume 0     C                NTFS   Partition     99 GB  Healthy    System

DISKPART> select volume 0

Volume 0 is the selected volume.

DISKPART> extend_
```

Fig. 12 Sample DiskPart Dialogue Expanding a Volume. This exemplar shows a sample interaction with the "diskpart" command to extend a volume

6. Type **extend**. You should be rewarded with a status message that diskpart was able to successfully extend the partition size (*see* Fig. 12).
7. You can now leave diskpart by typing **exit**.

3.11.3 Create a Data Analysis System for a Collaborator

If a collaborator wishes to perform some basic analysis on their data on their own this can easily be accomplished by creating a computing instance loaded with just their software and data.

1. Create an instance and install it with all the software your collaborator will need for their analysis.
2. Follow Subheading 3.11.1 to create a new instance based on a clone of your master image from the above step.
3. Attach a clone of their data drive to their instance or otherwise copy their data so that it is available on their instance.
4. If your collaborator has troubles with RDP, consider installing **Anydesk** [3] or **Teamviewer** [4] for them to use instead.
5. Important considerations.
   (a) Make sure that this "computer" you created for your collaborator has no personal information on it (e.g., account logins, browser caches, etc.).
   (b) Make sure that the "computer" does not have write access to any important resources. For example, the primary data repository. Assume that anything you give to your collaborator will unintentionally be loaded with every piece of malware you can think of—not every collaborator is this bad, but enough are.
   (c) Resources cost money! You will want to shut down the instances when you are not using them and free up and hard drive space you have allocated that you no longer are using.

3.11.4 Cheap Laptops and Cloud Resources Make for Safer and More Robust Bench-Side Computing

Within my laboratory, it is becoming more common for students and researchers to desire to use their laptops at the bench. This is very convenient for real time documentation as well as access to protocols and Internet based resources. However, to use the same laptop inside and outside of the laboratory presents safety concerns that need to be addressed. This can be addressed with cloud-based resources.

1. Acquire an inexpensive laptop. This laptop will be dedicated to use within the laboratory. We currently use Chromebooks for this task; if this works with your policies and workflows, it allows for people to easily replicate their environment and access resources across several laptops (e.g., one inside and one outside the lab). Further Chromebooks support Microsoft RDP.
2. As necessary, use the laptop to connect to an AWS instance to record or analyze data and protocols. Since the AWS instance is accessible from any other computer, it allows the user to have access to the same environment and data inside and outside the lab without the risk of carrying a contaminated device.

4 Notes

1. For perspective, I often perform remote administration via a low-end Chromebook (Intel Celeron, 4 GB Ram) connected over a 4G hotspot (1.5 Mbps up/10 Mbps down). While sufficient, improved network performance certainly does not hurt.
2. Throughout this chapter, for simplicity, I will often refer to “hard drives” when referring to the storage available to your cloud instance. Technology-wise, storage can either be SSD or magnetic based. For large bulk storage (e.g., data files) the magnetic drives will typically be cheaper. The computer “boot drives” in Amazon need to be SSD based. When you request a storage volume and attach it to your instance, it functionally behaves very much like a physical “hard drive”; thus, aside from performance choices (such as SSD vs magnetic) the underlying implementation details are not important. The suite of storage solutions is referred to as **EBS**, the virtual hard drives are referred to as **volumes**.
3. Superficially, a “keypair” is a cryptographic concept that references a mathematical system in which one number of the pair can encrypt messages to the holder of the second number of the pair and vice versa. When you create the keypair, Amazon keeps one and gives you the other. Now, when Amazon wants to send you information (like a password) it uses this system.

Since Amazon servers do not keep the half of the keypair they sent you, it can always be used as a way to authenticate yourself back to Amazon.

4. It is worth checking the AWS cost calculator [1] when configuring instances. You can also check https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/instance-types.html for a complete description of all the instances and what their nuances of difference are. As an example, costs can be different depending on what region you instantiate your images; the free tier eligible configurations are not always the cheapest options when you have to pay—for example, t2.micro is "Free tier" but t3 are often cheaper than t2 even though next generation.

5. The default configuration allows all outgoing connections and only allows incoming connections using a computer remote control protocol, RDP, from all addresses (denoted by the 0.0.0.0/0). If all connections to your resource will come from a specific computer or collection of computers, you might want to limit the incoming RDP connections to be from a specific machine or subnet on the Internet. As an example, if I am coming from a machine with a known IP address of 128.125.1.15, I could add a rule that restricts incoming access to 128.125.1.15/32. If I am coming from any one of the machines on the 128.125.0.0 subnet, I could specify that as 128.125.0.0/16. There are many subnet mask calculators online that will help you format these correctly for other use cases [5]. You can additionally add rules here to allow other types of incoming connections (e.g., if you were running a MaxQuant server).

6. Hard drives need to be initialized and partitioned. Partitioning allows you to make a single hard drive appear as multiple virtual hard drives. Initialization takes a raw device and adds the minimal data structure required for the operating system to start using it—this includes the definition of the partition table. You have two options when you initialize your hard drive—MBR and GPT. MBR is the older, and traditionally more universally recognized format. However, there are some limitations—MBR can only have 4 physical partitions, but, even worse, MBR can only operate, assuming some common assumptions, with "hard drives" that are 2 TB or smaller. I tend to select MBR unless my operational needs require GPT (e.g., larger volumes).

7. To change the Google FileStream cache location in Windows, you are looking to change the registry value of ContentCachePath the process of which is described here: https://support.google.com/a/answer/7644837?hl=en. I recommend creating **D:\GoogleFSCache** and setting ContentCachePath to

point to that directory. To edit the registry values, click on the **Windows Button**, type **regedit** then navigate through **HKEY_LOCAL_MACHINE**, **SOFTWARE**, **Google**, **DriveFS**. When you get to **DriveFS**, you will click on the name rather than expanding it. Then do **Edit** → **New** → **String Value** and create **ContentCachePath**. If you double click on this, you can set it to your new cache directory (in my case, D:\GoogleFSCache). You will now need to reboot your instance or restart Google File Stream. Reboots are easier to execute so I will assume you did that; in either case, when FileStream next comes online, you should see a confirmatory notice that your cache has moved.

8. With storage, you are billed for the disk space you consume—when you originally create a snapshot, no additional space is required (it is all part of the original volume). Only if you make changes to the volume, then the snapshot has to store the differences and your storage costs go up. If you delete the underlying volume, then the snapshot will consume as much space as that volume did.

Acknowledgments

I thank David B. Agus and the USC Center for Applied Molecular Medicine for generous institutional support and Naim Matasci for insightful discussions.

References

1. https://calculator.s3.amazonaws.com/index.html
2. https://aws.amazon.com/premiumsupport/knowledge-center/accepted-payment-methods/
3. https://anydesk.com
4. https://www.teamviewer.com
5. https://www.calculator.net/ip-subnet-calculator.html