# Which Cloud for my data?

## Overview

**Teaching:** 10 min
**Exercises:** 20 min
**Questions**

- What cloud resources are available?
- How do I choose my resources?

**Objectives**

- Describe the advantages and disadvantages of various cloud computing platforms.
- Determine what resources a project will require
- Match required resources to computing platforms

## Cloud platform choices

There are several cloud providers to choose from. Some scientific clouds may either be free or allocate resources competitively. Commercial clouds can be very powerful, but choice can be overwhelming. We will cover as much as we can, but you will ultimately need to learn things not covered here so see the documentation below:

The major tradeoff between platforms is between flexibility and cost. Generally speaking, services that allow you more flexibility and autonomy, will be more expensive than highly managed services.

### University/Corporate Computing Clusters

Many universities and businesses operate their own computing clusters that are available to students and staff at low or no cost. If your employer maintains a computing cluster, this will almost always be the least expensive option.

However, most HPCCs (High Performance Computing Clusters) put limits on:

- The number of processors a user can utilize at once
- The amount of disk storage per user
- The amount of time a single process can run
- What programs can be installed, and by whom

- Who can have accounts and access data

HPCCs are also a shared resource, so even when you have access, your programs are unlikely to run immediately. Most HPCCs run some kind of scheduler, that you submit your processing jobs to, and it runs them as resources become available. In order to submit a job, you generally will need to know not only what program you want to run, but with how many processors and for how long. While interacting with the scheduler is no more difficult than interacting with the shell, it will have its own set of commands and syntax that you'll need to learn; and these vary widely among HPCCs.

There are also many upsides to using an HPCC. As previously mentioned, they're generally the least expensive option, but they often come with more perks. For instance, many HPCCs offer free or low-cost training, storage space, back-up options, and technical support. If your HPCC has a scheduler, you can also queue up many sequential jobs all at once, and you don't have to worry about racking up fees on instances that are sitting idle. It's often also much easier to pay for HPCC use than to pay for Amazon using grant money, however universities are getting better about AWS payments.

## Open Science Clouds

### XSEDE (https://www.xsede.org/)

The Extreme Science and Engineering Discovery Environment (XSEDE) is an NSF funded HPCC, so it is open to any US-based researcher, and shares most of the same benefits and drawbacks of a university or corporate HPCC. If your university or corporation doesn't have it's own HPCC resources, XSEDE will likely be your cheapest option.

Although any US-based researcher can use XSEDE, first they'll need an account (https://portal.xsede.org/#/guest). Like the HPCC options described above, XSEDE uses a scheduler to start jobs, and puts limits on how many resources any one user can utilize at once.

XSEDE can also be a bit intimidating at first because you will need to know what resources you need, and for how long, before you get started. XSEDE runs like a mini version of the NSF grant system. In order to qualify to submit large jobs, you'll have to submit a allocation request (https://portal.xsede.org/allocations/research), in the form of a short proposal. Also like an NSF grant, if your proposal is accepted, that means you have access to whatever resources you were approved for, for the time frame you requested.

Don't let that paragraph scare you off though. XSEDE has two different allocation tracks. If you aren't sure exactly what you'll need for your big project, you can request a startup allocation (https://portal.xsede.org/allocations/startup) which only requires an abstract rather than a proposal, and grants you a year to try out your new pipeline or analysis. These are usually granted in a week or so, and are intended for you to test your pipeline so you know what to ask for in your allocation proposal.

If that still sounds a little too daunting, XSEDE also has trial allocations (https://iujetstream.atlassian.net/wiki/spaces/JWT/pages/76149919/Jetstream+Trial+Access+Allocation) which give you access to only a tiny fraction of XSEDES power, but are plenty large enough to test your code and see if a larger allocation is worth pursuing. These allocations are granted more or less immediately by simply filling in a form and agreeing to the usage rules.

If you're interested in using XSEDE, check to see if your workplace has a Campus Champion (https://www.xsede.org/community-engagement/campus-champions). These are people who have had extensive training on both the XSEDE system and the allocation program, and can help you figure out how to apply and what you need.

Open Science Grid (https://opensciencegrid.org)

The Open Science Grid (OSG) is an NSF-funded national network of computing centers that have pooled their resources together and made them available to various research groups. The OSG is usable by any researcher based at a US institution, and is accessible for free without an allocation. It can provide millions of computing hours for researchers who have problems that fit well on its setup.

Certain projects and universities have direct access to the Open Science Grid, but any researcher can access it through the OSG Connect (https://osgconnect.net/) entry point. If you apply for OSG access through that website, you will have a consultation with someone who can help you determine if your analysis is a good fit for and how to get started.

The OSG is a great fit for problems that can be broken into lots of independent pieces. One good example is read alignment: the starting read data can be broken into several pieces, each of them aligned, and then the results combined. Another good problem type for the OSG are multi-start simulations or statistical analyses where you need to run the same model or simulation many, many times. The payoff of using this approach is being able to run on many hundreds (sometimes thousands!) of computers at once, accelerating your analysis.

Note that you don't access a specific computing center through OSG – unlike XSEDE, where you apply for time and then run on a specific HPCC resource, the OSG sits on top of many resources and when you submit your work, it could run almost anywhere in the overall system.

Atmosphere (https://pods.iplantcollaborative.org/wiki/display/atmman/Getting+Started)

CyVerse (iPlant Collaborative) Atmosphere (http://www.cyverse.org/atmosphere)

JetStream (http://jetstream-cloud.org/)

## Commercial Clouds

Computing architecture is moving (albeit at a slow pace) to the Model-to-Data paradigm. This means that scientists should be encouraged to bring their compute to where the data is stored, instead of the the other way around. The following outlines the general differences between the three major commercial cloud providers: Amazon Web Services (AWS), Google Cloud Platform (GCP) and Microsoft Azure.

Essentially all cloud providers provide extremely similar computing and storage options; you can "rent" or provision computing infrastructure with very similar specifications across all three cloud vendors. Even the costs are highly comparable. What governs how to choose the right cloud computing vendor is highly opportunistic: (1)funding options, (2)solidarity with collaborating/similar scientific groups, (3)location of datasets that a particular research group works with and (4)familiarity with cloud vendor services.

1. Funding options: Does your grant stipulate where you should build your computing pipeline? For example, the NIH often partners with specific cloud vendors to provide cloud credits that allow researchers to compute for free. Some cloud vendors also provide research credits.

2. Solidarity with collaborating/similar scientific groups: Are other research groups in your field drawn to a specific cloud vendor? It might make sense to utilize the same cloud service to minimize transfer (egress) costs especially if you are sharing large datasets. You may also be able to make use of existing pipelines without reinventing the wheel if you choose the same cloud provider that your collaborators are using.
3. Location of datasets that a particular research group works with: Again, thinking of bringing your models to the where the data is stored helps minimize costs and saves you time in having to download and store data separately.
4. Services here refer to cloud vendor add-ons that take away the need for a user to set up their own computing infrastructure. A fully managed database (e.g. AWS RDS, GCP CloudSQL, Azure SQL DB) is an example of a service. If you are used to SQL Server, you may want to look into options provided by Azure. Are you more familiar with Postgres SQL? Then AWS and GCP might provide cheaper options for you.

## Amazon EC2 (http://aws.amazon.com/ec2/)

The Amazon Web Service (AWS) that you've been using is the Elastic Compute (EC2) cloud. There are actually lots of other cloud and storage solutions under the AWS umbrella, but when most data scientists say AWS, they mean EC2 (https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/EC2_GetStarted.html). With EC2, you can rent access to a cloud computing resource as small as your laptop, or as large as a 64 processor machine with 488GB of memory, and with a number of different operating systems. These instances can be optimized for jobs that are memory intensive, or require a lot of bandwidth, or almost any other specific need (https://aws.amazon.com/ec2/instance-types/). There are so many options that we can't cover them all here, but these are a few popular ones:

### On-Demand

All this variety and optimization makes EC2 much more expensive than an average HPCC, however, depending on your needs it can be quite affordable (https://aws.amazon.com/ec2/pricing/). If you want to start an EC2 instance whenever you want and have instant access, you can rent a quite large on-demand instance with 8 processors and 32 GB of memory for ~40 cents an hour, and tiny instances are only about half a cent per hour.

### Spot-Instances

If your program can tolerate pauses, and you don't need the analysis done as fast as possible, you can request a spot instance. Essentially, whenever Amazon has computing capacity that no one is paying them for, they lower the prices for renting some systems. If you request a spot-instance, that means you specify the size and parameters of your machine rental, and set a limit on how much you're willing to spend per hour. Then, whenever the rental rate dips below your maximum, your instance turns on and runs until the price goes back up. If it's not an important shopping season, and you aren't in a hurry, you can often run spot-instances for less than half their normal cost.

### Free Tier

There are also free options (https://aws.amazon.com/free/), which allow you to test out the interface and make sure it will meet your needs before you start renting.

Just remember that with EC2 and all other commercial services, you're paying for renting the computer, whether you're using it or not. If you leave an instance on idly for months after your pipeline has finished, you'll still have to pay for that time.

## Google Cloud (https://cloud.google.com/): getting started (https://cloud.google.com/compute/docs/quickstart)

GCP offers very competitive prices for compute and storage (as of July 2019, their compute pricing is lower than that of AWS and Azure for instances of comparable specifications). If you are looking to dabble in cloud computing but do not need a vast catalog of services, GCP would be a good place to start looking.

Their version of “Spot Intances” are known as pre-emptible instances and offer very competitive pricing. GCP also has TPUs.

Microsoft Azure (https://azure.microsoft.com/en-us/)

If your software requires Microsoft Windows, it may be cheaper to use MS Azure due to licensing issues. Azure’s computing instances are known as Azure Virtual Machines and often come at a slightly higher cost than other cloud computing vendors’ offerings. If a lot of your computing pipeling is Windows dependent, it may make sense to build everything on MS Azure from the get go.

# How to Choose

As you can see, highly managed systems (HPCCs, XSEDE, etc) usually are free or cheap, but relatively inflexible. There may be certain programs you can’t install, or there may be long wait times. Commercial systems are generally more flexible because you can make them look however you want, but they can be quite expensive, especially if you run for a long time, or have a lot of data. However, there are other things to consider.

Another way to think about this is not *whether* you want to spend your time and money, but *where* you want to spend them. AWS will let you install anything you want, but that also means that you’ll have to spend some time up-front installing programs and testing configurations. Your HPCC jobs might take a week to start, but you don’t have to do any of the systems administration, and you can work on other projects while your job sits in the queue.

Your familiarity with the pipeline can also be a factor. Let’s say you want to run a program you’ve never run before. If you have no way to estimate how long your job will take, a service like AWS might save you money, because you can run your program until its done, no matter how long that is. Running it on an HPCC can be really frustrating, because you’ll need to submit your job with the amount of time and resources it will use. An HPCC will only let your job run the amount of time you requested, so if you underestimate the amount of time it will take, even by one minute, the system kills your program and you need to resubmit it. On the other hand, if you want to run that same program, but you can easily estimate the runtime, an HPCC is going to be a much cheaper choice.

In my work, I often use both commercial and non-commercial services. I tend to use AWS for testing, with small amounts of data, until I know how the program behaves. Then I port the pipeline to my university HPCC for running the full dataset.

## ✎ Discussion

In small groups or on your own, plot out your next bioinformatics project. With guidance from your instructors and the above references, try to determine not only what types of resources you'll need, but what platform will best suit your project.

Some things to consider:

- How much data do you have?
- What computational steps will it need?
  - What is the *largest* computational step?
  - Can any steps be done in parallel?
- What is your timeframe?
- Who will be doing most of the computational work?
  - What computational skills do they have?
  - Do you need to share the data across many labs?
- How many times will you need to run this pipeline?

## 📌 Human genomic data & Security

Note that if you are working with human genomics data there might be ethical and legal considerations that affect your choice of cloud resources to use. The terms of use, and/or the legislation under which you are handling the genomic data, might impose heightened information security measures for the computing environment in which you intend to process it. This is a too broad topic to discuss in detail here, but in general terms you should think through the technical and procedural measures needed to ensure that the confidentiality and integrity of the human data you work with is not breached. If there are laws that govern these issues in the jurisdiction in which you work, be sure that the cloud service provider you use can certify that they support the necessary measures. Also note that there might exist restrictions for use of cloud service providers that operate in other jurisdictions than your own, either by how the data was consented by the research subjects or by the jurisdiction under which you operate. Do consult the legal office of your institution for guidance when processing human genomic data.

## Other Resources:

Learn more about cloud computing in bioinformatics:

Fusaro VA, Patil P, Gafni E, Wall DP, Tonellato PJ (2011) **Biomedical Cloud Computing With Amazon Web Services**. PLoS Comput Biol 7(8): e1002147. doi: 10.1371/journal.pcbi.1002147 (http://journals.plos.org/ploscompbiol/article?id=10.1371/journal.pcbi.1002147)

## ❗ Key Points

- Choose your cloud resource to best fit your budget and flexibility requirements

‹ (../03-verifying-

^ (../)

instance/index.html)

Edit on GitHub (https://github.com/datacarpentry/cloud-genomics/edit/gh-pages/_episodes/04-which-cloud.md) / Contributing (https://github.com/datacarpentry/cloud-genomics/blob/gh-pages/CONTRIBUTING.md) / Source (https://github.com/datacarpentry/cloud-genomics/) / Cite (https://github.com/datacarpentry/cloud-genomics/blob/gh-pages/CITATION) / Contact (mailto:team@carpentries.org)

Using The Carpentries theme (https://github.com/carpentries/carpentries-theme/) — Site last built on: 2019-08-27 17:14:54 +0000.