Cloud Computing

 (https://github.com/learn-co-curriculum/dsc-cloud-computing) **(https://github.com/learn-co-curriculum/dsc-cloud-computing/issues/new)**

Introduction

For most of this curriculum, we have focused on techniques that you can apply on your own computer. In a production setting, you will likely need to go beyond your computer and utilize cloud services! This lesson gives an overview of what that means and what your options are.

Objectives

You will be able to:

- Explain the general concept of "the cloud"
- Identify some of the most popular cloud platforms
- Identify the key use cases of cloud platforms

The Cloud

(https://commons.wikimedia.org/wiki/File:Cloud_computing_icon.svg)

Cloud computing is originally an IT infrastructure term, which is contrasted with *on-premises* computing. In traditional IT infrastructure, specialized computers (servers) would be purchased and set up at the company to do things like store customer data in a database, or serve the company's website. The problem with this setup is that it's less flexible (leaving resources unused at low-traffic times, running the risk of servers being overloaded at high-traffic times) and requires on-site server maintenance expertise. Some small or niche organizations still use 100% on-premises computing (e.g. because of special security needs), but nowadays most have moved to using some kind of cloud platform.

On a cloud platform, server resources are typically available "on-demand", meaning that the amount of storage space or CPU time scales based on user needs, and the user pays accordingly. This

allows for better agility, performance, and monitoring than traditional on-premises setups.

Popular Cloud Platforms

Amazon Web Services (AWS) emerged in the early 21st Century as the first modern cloud platform. Previously there had been time-sharing approaches and virtual private network (VPN) services, but AWS also introduced the concept of virtual private servers. Two early AWS products, Simple Storage Service (S3) and Elastic Compute Cloud (EC2), were released in 2006 and are still very popular today.

Other major tech companies soon followed with their own cloud platforms, to compete with AWS and also offer novel products and services. The two major competitors to AWS are Microsoft's **Azure** and **Google Cloud**. For most typical cloud computing use cases, all three of these companies offer comparable services, although one particular offering might be best for your company's needs in terms of pricing or specifications.
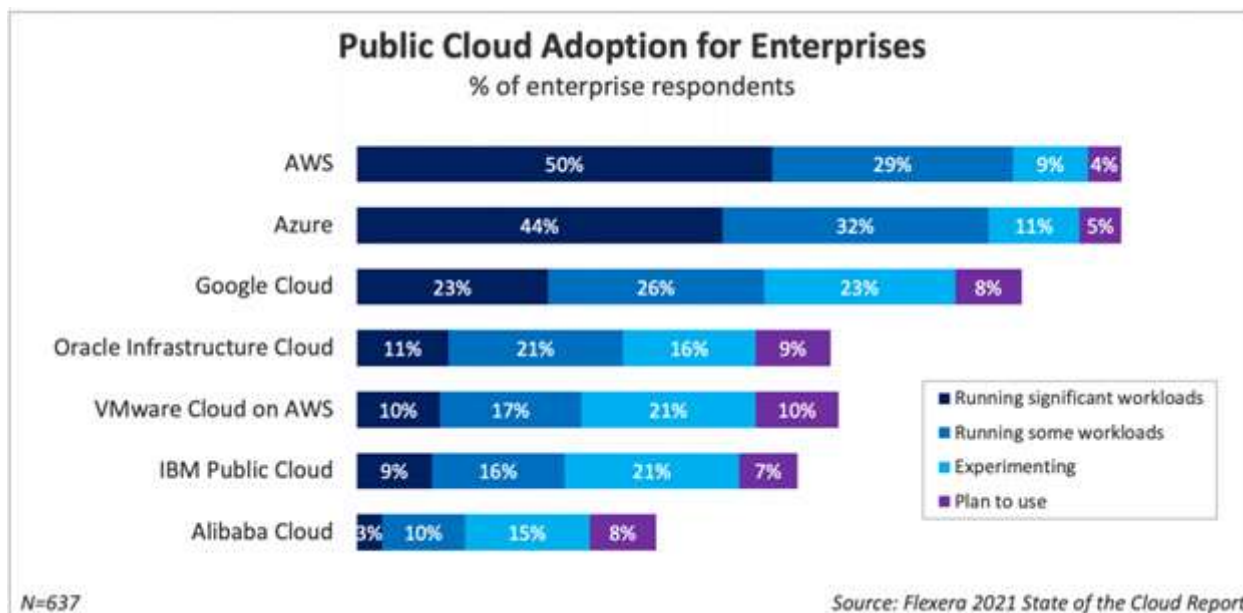
From **_Cloud Computing Trends: 2021 State of the Cloud Report_** ⏍ **(https://www.flexera.com/blog/cloud-computing-trends-2021-state-of-the-cloud-report/)**

Because AWS is the most popular cloud platform, we'll take some time later to discuss the AWS ecosystem and specific AWS services that you may find useful in your projects or future career.

Hardware Acceleration

You're already familiar with software libraries like NumPy and Spark that can help your computer perform at its best, but ultimately your personal computer (laptop or desktop) was probably not designed specifically for high-powered machine learning tasks.

Hardware acceleration is useful when you need to train models or make predictions more quickly. As a general concept, **hardware acceleration** ⏍ **(https://www.omnisci.com/learn/resources/technical-**

glossary/hardware-acceleration) means using purpose-built hardware rather than general-purpose hardware.

In the case of machine learning, this typically means running your code on a **GPU**, rather than a CPU. A CPU *can* do everything that a GPU can do, but it is not optimized for it, so it will likely take more time. **This blog** ⬏ **(https://towardsdatascience.com/maximize-your-gpu-dollars-a9133f4e546a)** argues that a CPU is to a GPU as a horse and buggy is to a car.

The easiest way to get started with utilizing a GPU is by using a **cloud notebook**.

Cloud Notebooks

A cloud notebook means that you can work in a familiar notebook interface, while at the same time using more-powerful computational resources. You usually don't have to install or configure anything -- you can just start coding in a super-powerful cloud environment. Sometimes these cloud environments even have built-in GitHub integration, so you can complete the entire development process in the cloud! Some past Flatiron students have found cloud notebooks to be extremely helpful for building the kinds of models they wanted to build in the time they had.

There are also some downsides to cloud notebooks to be aware of:

- Sometimes you don't have precise control over the available **Python packages** and their versions. So, for example, you might want to use the latest version of TensorFlow, but the service you're using might not support it yet.
- Paid tools (including the AWS and Google Cloud Platform tools listed below) can quickly get very **expensive**. In particular be very careful about completing tutorials or running other people's projects, because it's easy to use up a lot of compute resources. You will typically get some free credits when signing up for these services, and we recommend that you save them for your own portfolio projects. Also make sure you always shut down your notebook instance when you're not using it.
- Some of these notebooks don't work with a straightforward **file system** like you do on your everyday DS setup. Instead of simply being able to call `pd.read_csv` to open a file, you'll likely need to interact with a web API to pull in your data from a storage bucket or other virtual storage location.

Some popular cloud notebooks include:

- **AWS SageMaker** ⬏ **(https://docs.aws.amazon.com/sagemaker/latest/dg/nbi.html)**
  - Professional-grade cloud JupyterLab instances (paid service, some free credits may be available)
  - We'll also have more lessons on this later in the curriculum
- **Google Cloud Notebooks** ⬏ **(https://cloud.google.com/vertex-ai/docs/workbench/user-managed?hl=en_US)**

- Professional-grade cloud JupyterLab instances (paid service, some free credits may be available)
- **Google Colab ▶ (https://research.google.com/colaboratory/)**
  - More like the "Google Docs" of notebooks: not quite the same as a Jupyter notebook, and doesn't have access to a file system, but very fast to get started and 100% free
- **Kaggle Kernels ▶ (https://www.kaggle.com/code)**
  - Not exactly a Jupyter notebook, but a similar interface from Kaggle. Great if you're using a dataset from Kaggle
  - Free, although you're limited to **30 hours per week ▶ (https://www.kaggle.com/page/GPU-tips-and-tricks)** of GPU time
  - Go **here ▶ (https://www.kaggle.com/dansbecker/running-kaggle-kernels-with-a-gpu)** for instructions on how to set up a GPU, **here ▶ (https://www.kaggle.com/docs/notebooks)** for full documentation
- **Databricks Community Edition ▶ (https://databricks.com/product/faq/community-edition)**
  - Not exactly a Jupyter notebook, but a similar interface from Databricks
  - Free cloud Spark cluster functionality

Cloud Instances/Containers

Sometimes you want to be able to run a full-fledged cloud computer, not just a notebook. With cloud instances, you can provision the necessary compute resources for completing essentially any task! You get root access to the file system and can run long-running scripts without overheating your personal computer. With Docker, you can even develop the code locally then deploy it online to speed up processing times.

There are also some downsides of cloud instances to be aware of:

- Similar to cloud notebooks, paid cloud instances can get very **expensive**. Pay particular attention to the amount of storage you are using, since a storage-specific solution might be a lot cheaper than a general-purpose cloud instance.
- Cloud instances require some **systems administration** skills. You'll likely need to configure access permissions and ports. Most containers are Linux-based so you'll need to be comfortable navigating and installing things using the terminal.

Some popular cloud instances include:

- **AWS EC2 ▶ (https://aws.amazon.com/ec2/)**
  - Go **here ▶ (https://aws.amazon.com/blogs/machine-learning/train-deep-learning-models-on-gpus-using-amazon-ec2-spot-instances/)** for instructions on training deep learning models with GPUs on EC2
- **Google Cloud Compute Engine ▶ (https://cloud.google.com/compute/docs)**
  - Go **here ▶ (https://cloud.google.com/ai-platform/training/docs/using-gpus)** for instructions on using GPUs for training models with Google Cloud Platform

- **[Azure Virtual Machines](https://docs.microsoft.com/en-us/azure/virtual-machines/)** ⤳ **(https://docs.microsoft.com/en-us/azure/virtual-machines/)**
  - Because this is made by Microsoft, they're one of the few places that offers Windows virtual machines
- **[Oracle Cloud Infrastructure](https://www.oracle.com/cloud/)** ⤳ **(https://www.oracle.com/cloud/)**
  - Generous free tier

Cloud Storage

It's annoying to have huge data files taking up space on your laptop, and if you want to train your model in the cloud, your data also needs to be in the cloud. But for reasons related to hardware acceleration, it can get pretty expensive to store large datasets in general-purpose cloud services like an EC2 instance or a cloud VM. That's when cloud storage services become useful.

There are many different kinds of cloud storage tools, but they fall roughly into three categories: **file storage** systems, cloud storage **buckets**, and cloud **databases**.

Cloud File Storage

This type of cloud storage is probably most familiar to you already. Cloud file storage includes something like Dropbox or Google Drive, where files are stored in directories that can be navigated in a way that is similar to the file system on your local computer.

While there are professional-grade cloud tools for this such as **AWS Elastic File System (EFS)** ⤳ **(https://aws.amazon.com/efs/)** , **Azure Files** ⤳ **(https://azure.microsoft.com/en-us/services/storage/files/)** , and **Google Cloud Filestore** ⤳ **(https://cloud.google.com/filestore)** , the main takeaway you should have is the contrast between file storage and the storage approaches described below.

Cloud Storage Buckets

Also known as **object storage** services, cloud storage buckets are great for storing raw files, e.g. folders full of images, CSVs, or JSONs. Unlike cloud file storage, files are not stored in directories. Instead, all data is stored in a "flat" system where it can be retrieved by name. This lack of hierarchy makes cloud storage buckets faster to use and more cost efficient than traditional file storage.

Typically these services are not free, although they are fairly cheap (2-5 cents per GB per month) and typically come with some free credits.

Some major providers of cloud storage buckets include:

- **[AWS S3](https://aws.amazon.com/s3/getting-started/)** ⤳ **(https://aws.amazon.com/s3/getting-started/)**
- **[Google Cloud Storage](https://cloud.google.com/storage/)** ⤳ **(https://cloud.google.com/storage/)**
- **[Azure Storage](https://docs.microsoft.com/en-us/azure/storage/common/storage-introduction)** ⤳ **(https://docs.microsoft.com/en-us/azure/storage/common/storage-introduction)**

AWS S3 is the oldest and tends to have the most integration support with other platforms, although you may need to use Google storage if you're using other Google products or Azure storage if you're using other Azure products. Google Cloud Functions, for example, rely on source code being stored in Google Cloud Storage.

Cloud Databases

Databases are a familiar concept, but thus far we have mainly practiced using SQLite. If you're working with production-scale SQL data, it's best not to save it in a SQLite file but rather to store it on a database server.

Most likely your projects in this program will not require you to use a production-scale database, but this is an opportunity to practice if you want to prepare for on-the-job tasks.

Some major providers of cloud databases include:

- **Heroku Postgres** **(https://www.heroku.com/postgres)**
- **MongoDB Atlas** **(https://www.mongodb.com/cloud/atlas)**
- **AWS Aurora** **(https://aws.amazon.com/rds/aurora/)**
- **AWS RDS** **(https://aws.amazon.com/rds/)**

Summary

In this lesson, we discussed the definition of "cloud services" and walked through several use cases. In summary, most cloud services can be described as either being used for *hardware acceleration*, including cloud notebooks and cloud instances, and *cloud storage*, including cloud file storage, cloud storage buckets, and cloud databases.