



Cloud



What do clouds have to do with computers?

- Computers “in the cloud”
 - Do a lot of the heavy lifting for us
 - Are easier to maintain
 - Make computing more affordable
 - etc.



What makes the cloud so great?

As a developer

- Easy to maintain
- Scalable
- Affordable
- Data retention
- Eliminates hardware bottlenecking
- Streamline security practices

As a user

- Faster services
- Globally accessible services
- Improved security
- More uptime

Cloud Providers



Top 3 Cloud Providers





Amazon Web Services



- #1 Cloud provider
- Hundreds of Services

Cloud Computing



Why cloud computing?

- No more hardware limitations
- Scalability
- Can be cheaper than maintaining hardware
- Ease of access
 - Ability to stand up a server in minutes



Cloud Computing with AWS

[Amazon Web Services: Elastic Compute Cloud \(EC2\)](#)

* Note there are other (more advanced) computing services within AWS.





Examples of EC2 usage

According to Intricately, the top ten AWS users based on EC2 monthly spend are:

- Netflix: \$19 million.
- Twitch: \$15 million.
- LinkedIn: \$13 million.
- Facebook: \$11 million.
- Turner Broadcasting: \$10 million.
- BBC: \$9 million.
- Baidu: \$9 million.
- ESPN: \$8 million.

[More items...](#) • Jan 28, 2020

<https://www.contino.io> › insights › whos-using-aws

Who's Using Amazon Web Services? [2020 Update] - Contino

Cloud Storage



Why cloud storage?

- Scalability
 - No need to worry about running out of storage
- Simplified replication
- Affordable
 - no need to use a computer to host static files



Examples of cloud storage

- Facebook
- Apple
- Amazon
- Netflix
- Google





Cloud Storage with AWS

[Amazon Web Services: Simple Storage Service \(S3\)](#)





Cloud Storage with AWS

[Amazon Web Services: Relational Database Services \(RDS\)](#)



Amazon RDS



Cloud Storage with AWS

[Amazon Web Services: DynamoDB - A NoSQL database](#)



amazon
DynamoDB

Infrastructure as Code (IaC)



Why use IaC?

- Easy to use
- Removes manual work of provisioning resources
- Control your infrastructure as you would your codebase
- Make replicating applications much easier
- Gives better oversight of infrastructure



Infrastructure as Code with AWS

[Amazon Web Services: Cloudformation](#)



Cloudformation Stack

AWS Resources provisioned together

- Defined together in a template



Hands-on experience



Hands-on experience

Assume we've created our frontend / backend. Pretend we haven't deployed it yet, or maybe we're trying to deploy it to another environment (i.e., we have been using a “development” environment, but we want an isolated “production” environment).

Using Infrastructure as Code deploy a 402 project including the following

- Static frontend (website)
- Backend API

Template: <https://github.com/christianlisle/COM-S-402-example-project>