Scalable Machine Learning and Cloud Computing

Xianyuan Liu

xianyuan.liu@sheffield.ac.uk

HPC = High-performance computing cluster
AWS = Amazon Web Services

In Today’s Lecture:

Today we will,

- introduce local, HPC and Cloud Computing.
- look at **whether** to distribute a given problem
- have a quick run through of **services** and **tools** from the field of Cloud Computing.
- have several **walk-throughs** (AWS **EC2** and **lambda**).

In the Lab on Friday:

On Friday you will,

- **create** and **connect** to an **instance** on AWS
- set up an **Elastic Map Reduce cluster**
- use it to run **Apache Spark** to solve a matrix factorisation problem
- set up a **Lambda function**

HPC = High-performance computing cluster

Local vs HPC vs Cloud Computing

- Local Computing
  - Physically and directly managed by users, normally in the same geographic location as the user.

HPC = High-performance computing cluster

Local vs HPC vs Cloud Computing

- Local Computing
  - Physically and directly managed by users, normally in the same geographic location as the user.
- HPC
  - Clusters of computers combined to form a powerful computing environment (Stanage - Manchester, Bessemer - Leeds)
- Cloud Computing

The old Iceberg system server racks.

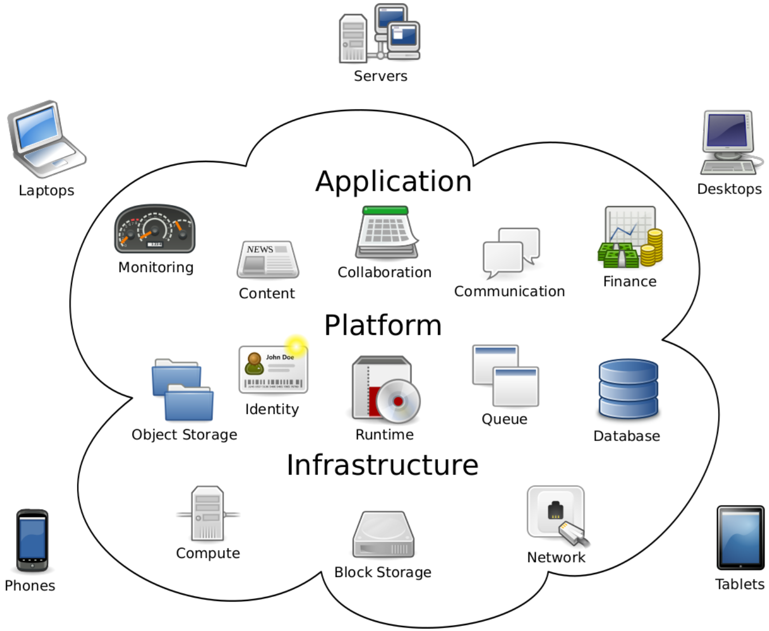
Image from: https://docs.hpc.shef.ac.uk/en/latest/hpc/what-is-hpc.html#gsc.tab=0

What is Cloud Computing

- Cloud computing is the **on-demand** availability of computer system resources, without direct active management by the user.

- Cloud computing relies on the **sharing of resources** to achieve coherence and typically uses a **pay-as-you-go** model.

- Large clouds often have functions **distributed** over multiple locations.



Content and image from: https://en.wikipedia.org/wiki/Cloud_computing

HPC vs Cloud?

Until now we've used the university's own HPC system.

HPC:

- Often provided for **free** by institution
- highly **interconnected** nodes - vital for many problems
- Most HPCs use SLURM (includes tools like job array, which let you run embarrassingly parallel arrays of jobs).
- Provide **support**

HPC vs Cloud?

HPC:

- Often provided for **free** by institution
- highly **interconnected** nodes - vital for many problems
- Most HPCs use SLURM (includes tools like job array, which let you run embarrassingly parallel arrays of jobs).
- Provide **support**

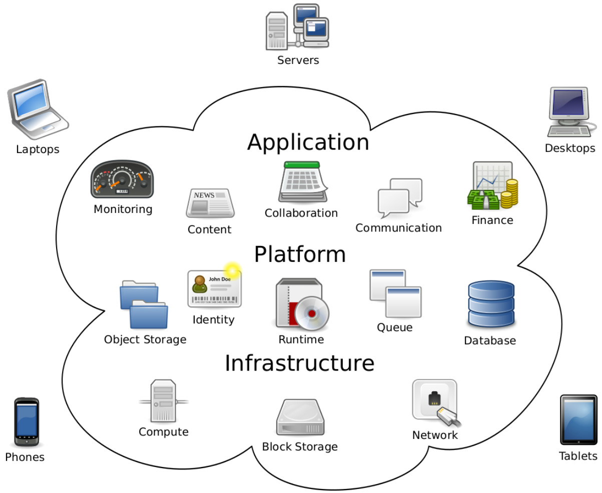
Cloud:

- **No queuing**
- **Quick** to set up
- Often **only option** outside of universities
- Well suited for **embarrassingly parallel** problems
- Almost all **platforms** and features **supported**
- Appropriate if **hosting a front-end server**
- Can be very expensive (+ accidental spend!)

Often ‘rent’ (virtual) hardware while you need it. Large clusters can be created briefly.

Why Cloud Computing for Machine Learning

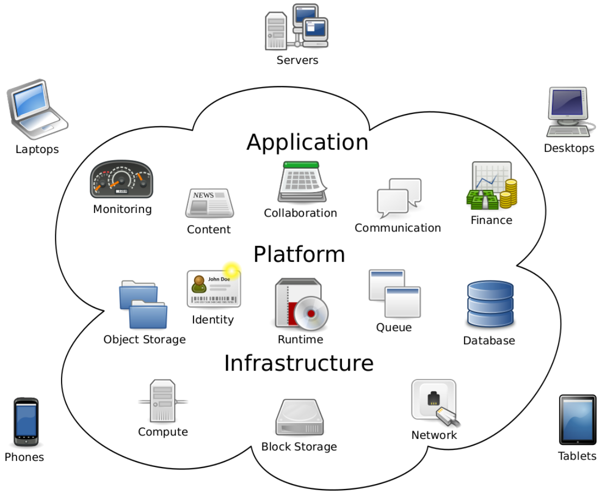
- Powerful Hardware
  - **On-demand** access to state-of-the-art hardware resources, instead of expensive investments in physical hardware.



Content and image from: https://en.wikipedia.org/wiki/Cloud_computing

Why Cloud Computing for Machine Learning

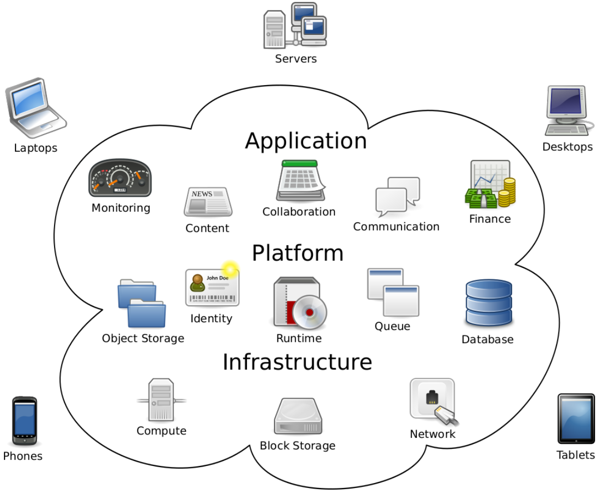
- Powerful Hardware
  - **On-demand** access to state-of-the-art hardware resources, instead of expensive investments in physical hardware.
- Scalability
  - **Dynamically** adjust resources (e.g., hardware, memory, storage) as your needs change.



Content and image from: https://en.wikipedia.org/wiki/Cloud_computing

Why Cloud Computing for Machine Learning

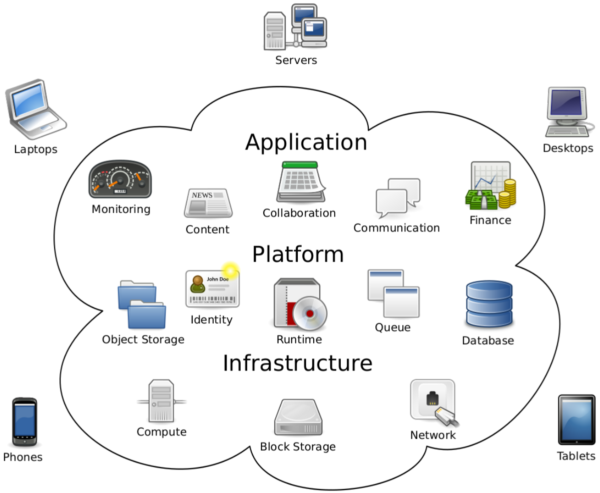
- Powerful Hardware
  - **On-demand** access to state-of-the-art hardware resources, instead of expensive investments in physical hardware.
- Scalability
  - **Dynamically** adjust resources (e.g., hardware, memory, storage) as your needs change.
- Pay-as-you-go Model
  - Pay **only** for the compute time and storage used.
  - Achieve significant savings on energy consumption, facility management, and ongoing hardware maintenance.



Content and image from: https://en.wikipedia.org/wiki/Cloud_computing

Why Cloud Computing for Machine Learning

- Powerful Hardware
  - **On-demand** access to state-of-the-art hardware resources, instead of expensive investments in physical hardware.
- Scalability
  - **Dynamically** adjust resources (e.g., hardware, memory, storage) as your needs change.
- Pay-as-you-go Model
  - Pay **only** for the compute time and storage used.
  - Achieve significant savings on energy consumption, facility management, and ongoing hardware maintenance.
- Flexibility
  - Access from **anywhere** with an internet connection.
  - Share resources with **multiple** users.



Content and image from: https://en.wikipedia.org/wiki/Cloud_computing

Local vs HPC vs Cloud Computing

- Local Computing
  - Physically and directly managed by users, normally in the same geographic location as the user.

- HPC
  - Clusters of computers combined to form a powerful computing environment (Stanage - Manchester, Bessemer - Leeds)

- Cloud Computing

Distributed system - Parallel computing

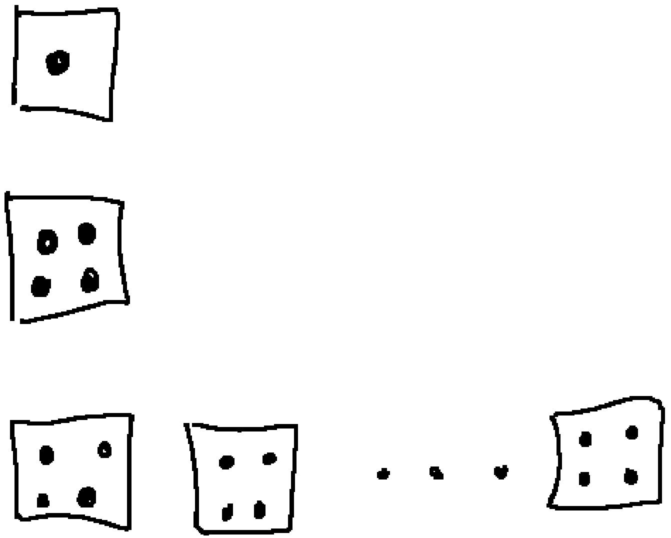
The old Iceberg system server racks.

Image from: https://docs.hpc.shef.ac.uk/en/latest/hpc/what-is-hpc.html#gsc.tab=0

Distributed system - Parallel computing

Different levels of parallelism:

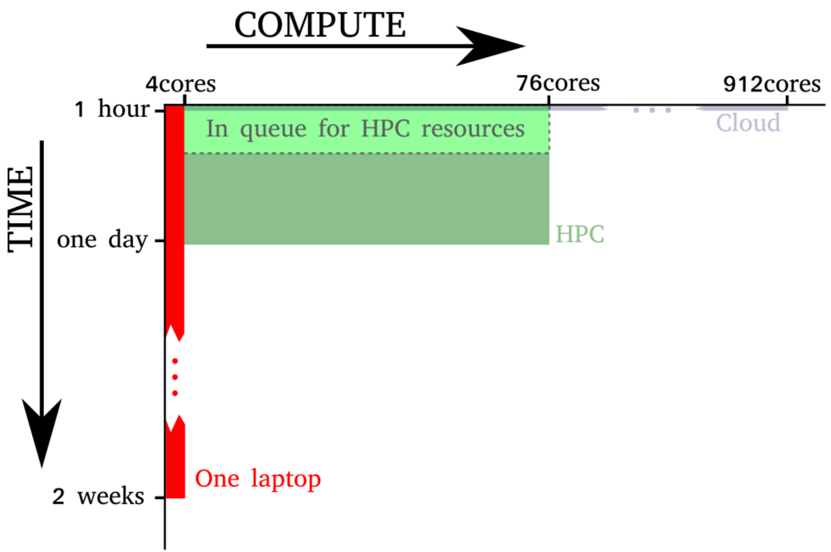
- **Single process** (e.g. python GIL code)
- Well parallelised multiprocess code (might be good to deploy on a single instance with lots of cores on AWS or HPC)
- Distributed over **multiple nodes** (either on AWS or HPC).



Distributed system - Parallel computing

Different levels of parallelism:

- **Single process** (e.g. python GIL code)
- Well parallelised multiprocess code (might be good to deploy on a single instance with lots of cores on AWS or HPC)
- Distributed over **multiple nodes** (either on AWS or HPC).



Local vs Distributed

- It's not always worth parallelising your application!

Local vs Distributed

- It's not always worth parallelising your application!
- **Human time** is worth more than **computer time**

Local vs Distributed

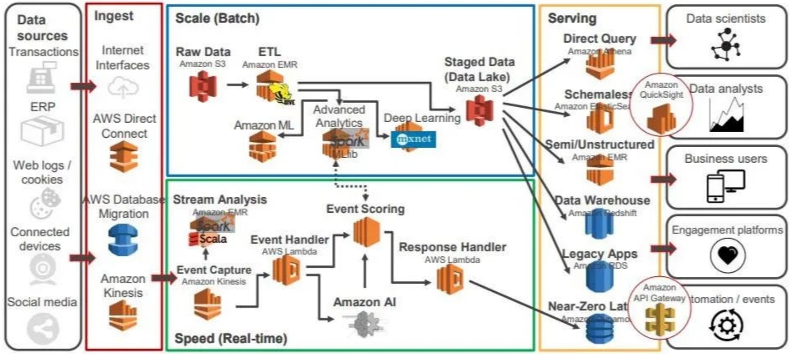
- It's not always worth parallelising your application!
- **Human time** is worth more than **computer time**, so if you can do something else and be patient it might be best not to bother with rewriting it to be parallel.

Why stay Local…

- May need to **redesign** algorithm

Why stay Local…

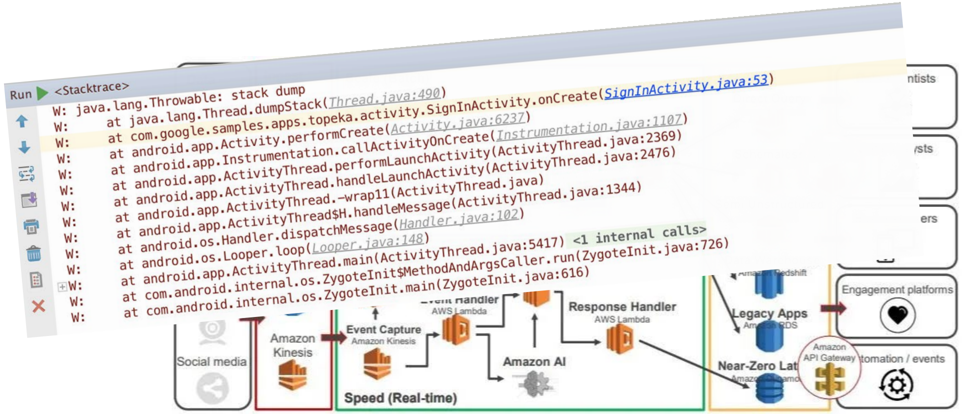
- May need to **redesign** algorithm
- Managing the **pipeline**.



Images from:
https://frogssavethequeen.files.wordpress.com/2014/08/surveillance-graffiti1-web-version.jpg
https://developer.android.com/studio/debug/stacktraces
https://medium.com/rock-your-data/getting-started-with-apache-kafka-efc616bd6dd5

Why stay Local…

- May need to **redesign** algorithm
- Managing the **pipeline**.
- Typically such environments are less **reliable** and less **predictable**.
  - Far more difficult to **debug**.

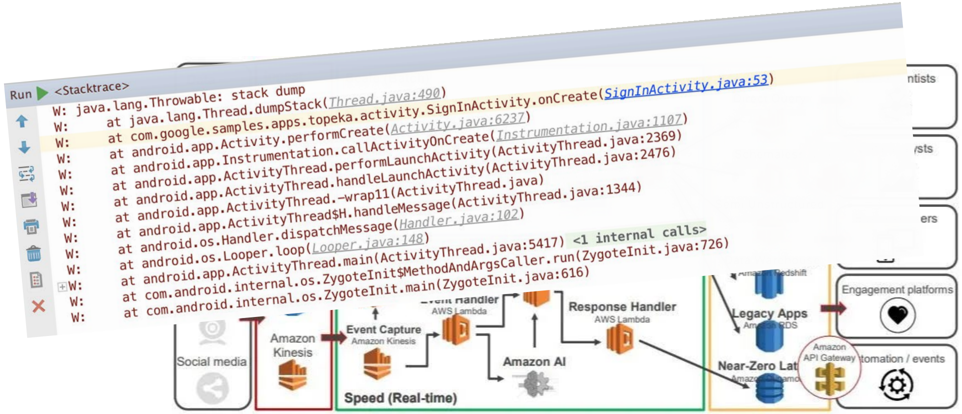


```
Run <Stacktrace>
W: java.lang.Throwable: stack dump
W:     at java.lang.Thread.dumpStack(Thread.java:490)
W:     at com.google.samples.apps.topeka.activity.SignInActivity.onCreate(SignInActivity.java:53)
W:     at android.app.Activity.performCreate(Activity.java:6237)
W:     at android.app.Instrumentation.callActivityOnCreate(Instrumentation.java:1107)
W:     at android.app.ActivityThread.performLaunchActivity(ActivityThread.java:2369)
W:     at android.app.ActivityThread.handleLaunchActivity(ActivityThread.java:2476)
W:     at android.app.ActivityThread.-wrap11(ActivityThread.java)
W:     at android.app.ActivityThread$H.handleMessage(ActivityThread.java:1344)
W:     at android.os.Handler.dispatchMessage(Handler.java:102)
W:     at android.os.Looper.loop(Looper.java:148)
W:     at android.app.ActivityThread.main(ActivityThread.java:5417) <1 internal calls>
W:     at com.android.internal.os.ZygoteInit$MethodAndArgsCaller.run(ZygoteInit.java:726)
W:     at com.android.internal.os.ZygoteInit.main(ZygoteInit.java:616)
W:
```

Images from:
https://frogssavethequeen.files.wordpress.com/2014/08/surveillance-graffiti1-web-version.jpg
https://developer.android.com/studio/debug/stacktraces
https://medium.com/rock-your-data/getting-started-with-apache-kafka-efc616bd6dd5

Why stay Local…

- May need to **redesign** algorithm
- Managing the **pipeline**.
- Typically such environments are less **reliable** and less **predictable**.
    - Far more difficult to **debug**.
- **Shifting data** to/from the servers might be a bottleneck.

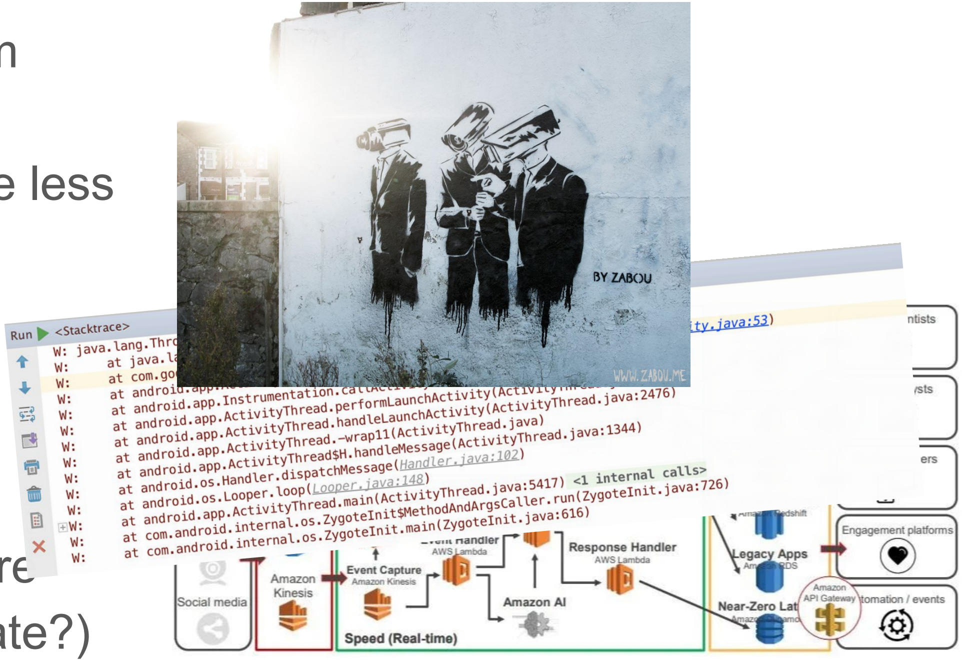


```
Run ▶ <Stacktrace>
W: java.lang.Throwable: stack dump
W:     at java.lang.Thread.dumpStack(Thread.java:490)
W:     at com.google.samples.apps.topeka.activity.SignInActivity.onCreate(SignInActivity.java:53)
W:     at android.app.Activity.performCreate(Activity.java:6237)
W:     at android.app.Instrumentation.callActivityOnCreate(Instrumentation.java:1107)
W:     at android.app.ActivityThread.performLaunchActivity(ActivityThread.java:2369)
W:     at android.app.ActivityThread.handleLaunchActivity(ActivityThread.java:2476)
W:     at android.app.ActivityThread.-wrap11(ActivityThread.java)
W:     at android.app.ActivityThread$H.handleMessage(ActivityThread.java:1344)
W:     at android.os.Handler.dispatchMessage(Handler.java:102)
W:     at android.os.Looper.loop(Looper.java:148)
W:     at android.app.ActivityThread.main(ActivityThread.java:5417) <1 internal calls>
W:     at com.android.internal.os.ZygoteInit$MethodAndArgsCaller.run(ZygoteInit.java:726)
W:     at com.android.internal.os.ZygoteInit.main(ZygoteInit.java:616)
W:
```

Images from:
https://frogssavethequeen.files.wordpress.com/2014/08/surveillance-graffiti1-web-version.jpg
https://developer.android.com/studio/debug/stacktraces
https://medium.com/rock-your-data/getting-started-with-apache-kafka-efc616bd6dd5

Why stay Local…

- May need to **redesign** algorithm
- Managing the **pipeline**.
- Typically such environments are less **reliable** and less **predictable**.
  - Far more difficult to **debug**.
- **Shifting data** to/from the servers might be a bottleneck.
- **Security** and **privacy** (where are you sending the data? Is it private?)



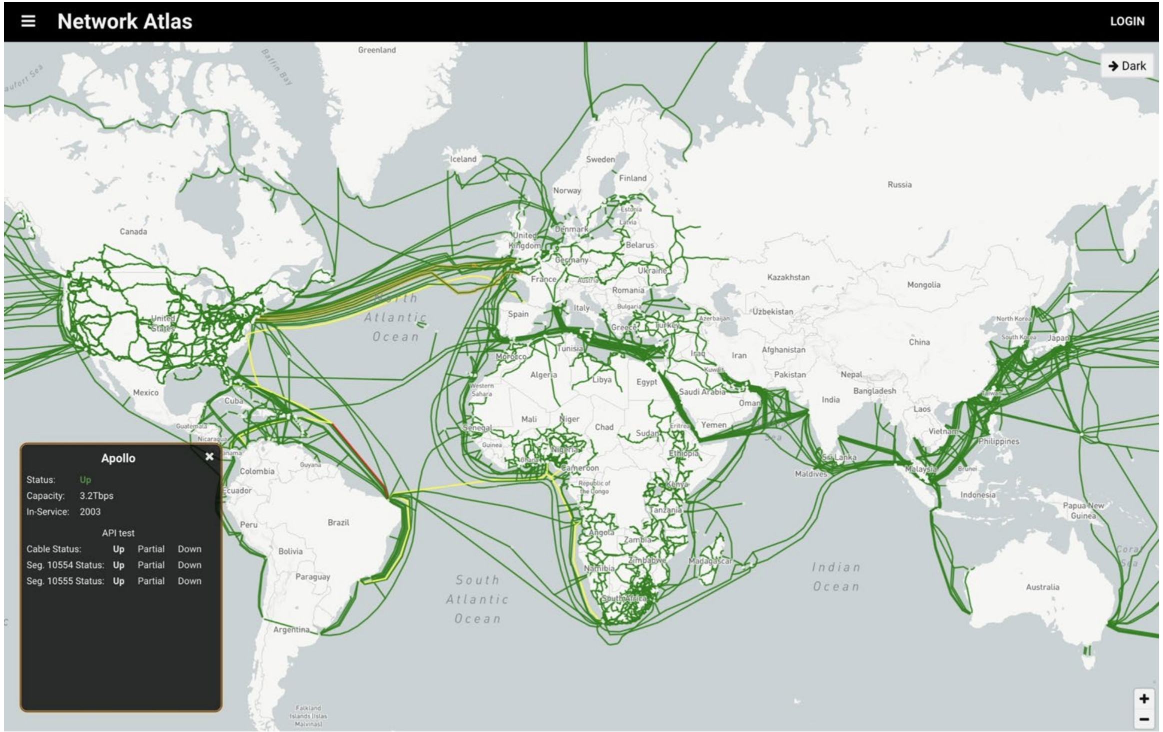
Run <Stacktrace>

```
W: java.lang.Thro
W:      at java.la
W:      at com.goo
W:      at android.app.                                           ity.java:53)
W:      at android.app.Instrumentation.callActivity
W:      at android.app.ActivityThread.performLaunchActivity(ActivityThrea
W:      at android.app.ActivityThread.handleLaunchActivity(ActivityThread.java:2476)
W:      at android.app.ActivityThread.-wrap11(ActivityThread.java)
W:      at android.app.ActivityThread$H.handleMessage(ActivityThread.java:1344)
W:      at android.os.Handler.dispatchMessage(Handler.java:102)
W:      at android.os.Looper.loop(Looper.java:148)
W:      at android.app.ActivityThread.main(ActivityThread.java:5417) <1 internal calls>
W:      at com.android.internal.os.ZygoteInit$MethodAndArgsCaller.run(ZygoteInit.java:726)
W:      at com.android.internal.os.ZygoteInit.main(ZygoteInit.java:616)
W:
```

Images from:
https://frogssavethequeen.files.wordpress.com/2014/08/surveillance-graffiti1-web-version.jpg
https://developer.android.com/studio/debug/stacktraces
https://medium.com/rock-your-data/getting-started-with-apache-kafka-efc616bd6dd5

Privacy and the GDPR…

The **EU GDPR** (General Data Protection Regulation) restricts the transfer of **personal data** to countries outside of the EU.



Apollo

Status:	Up		
Capacity:	3.2Tbps		
In-Service:	2003		

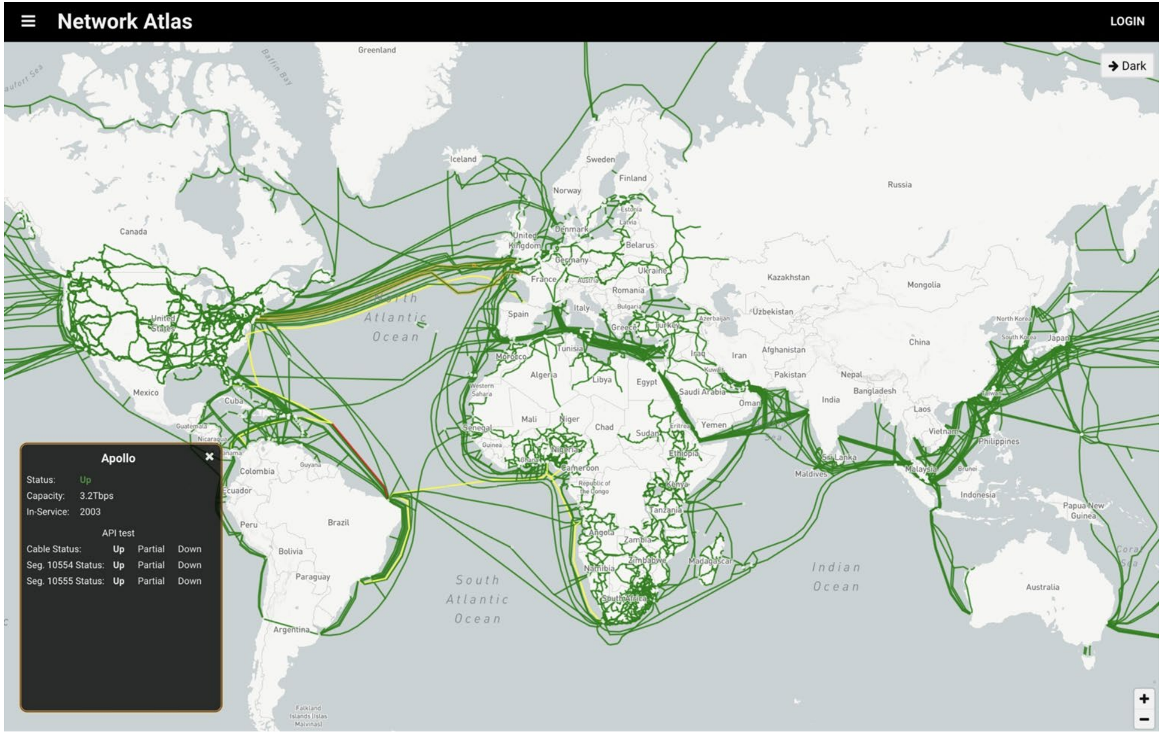
API test

Cable Status:	Up	Partial	Down
Seg. 10554 Status:	Up	Partial	Down
Seg. 10555 Status:	Up	Partial	Down

Privacy and the GDPR…

Post-brexit: UK GDPR is now in law, and is basically the same as the EU GDPR for now.

The **EU GDPR** (General Data Protection Regulation) restricts the transfer of **personal data** to countries outside of the EU.



Apollo ✕

Status: Up
Capacity: 3.2Tbps
In-Service: 2003

API test

Cable Status:	Up	Partial	Down
Seg. 10554 Status:	Up	Partial	Down
Seg. 10555 Status:	Up	Partial	Down

Privacy and the GDPR…

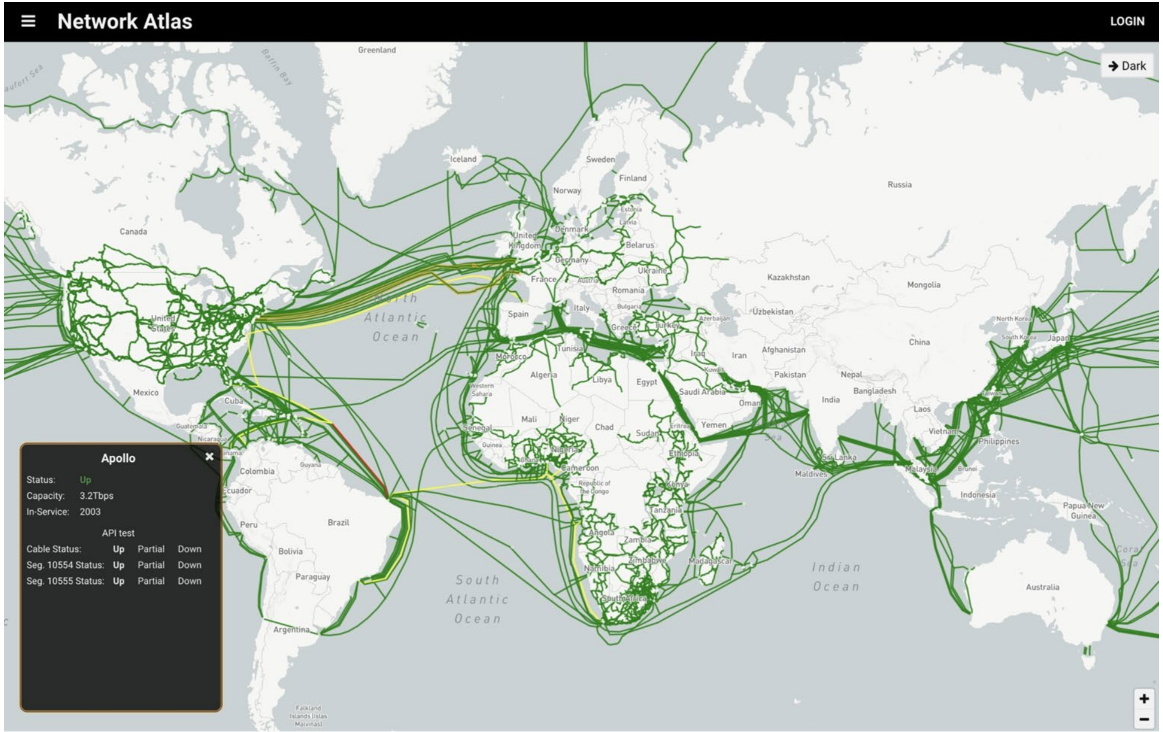
Post-brexit: UK GDPR is now in law, and is basically the same as the EU GDPR for now.

The **EU GDPR** (General Data Protection Regulation) restricts the transfer of **personal data** to countries outside of the EU.

One can process the data if it’s done in a country covered by an EU/UK “**adequacy decision**”

The UK and EU consider each other to have this.

Otherwise one needs a **standard contractual clause** (these have begun to diverge between the EU and UK).



Apollo

Status: Up
Capacity: 3.2Tbps
In-Service: 2003

API test

```
Cable Status:        Up   Partial   Down
Seg. 10554 Status:   Up   Partial   Down
Seg. 10555 Status:   Up   Partial   Down
```

Privacy and the GDPR…

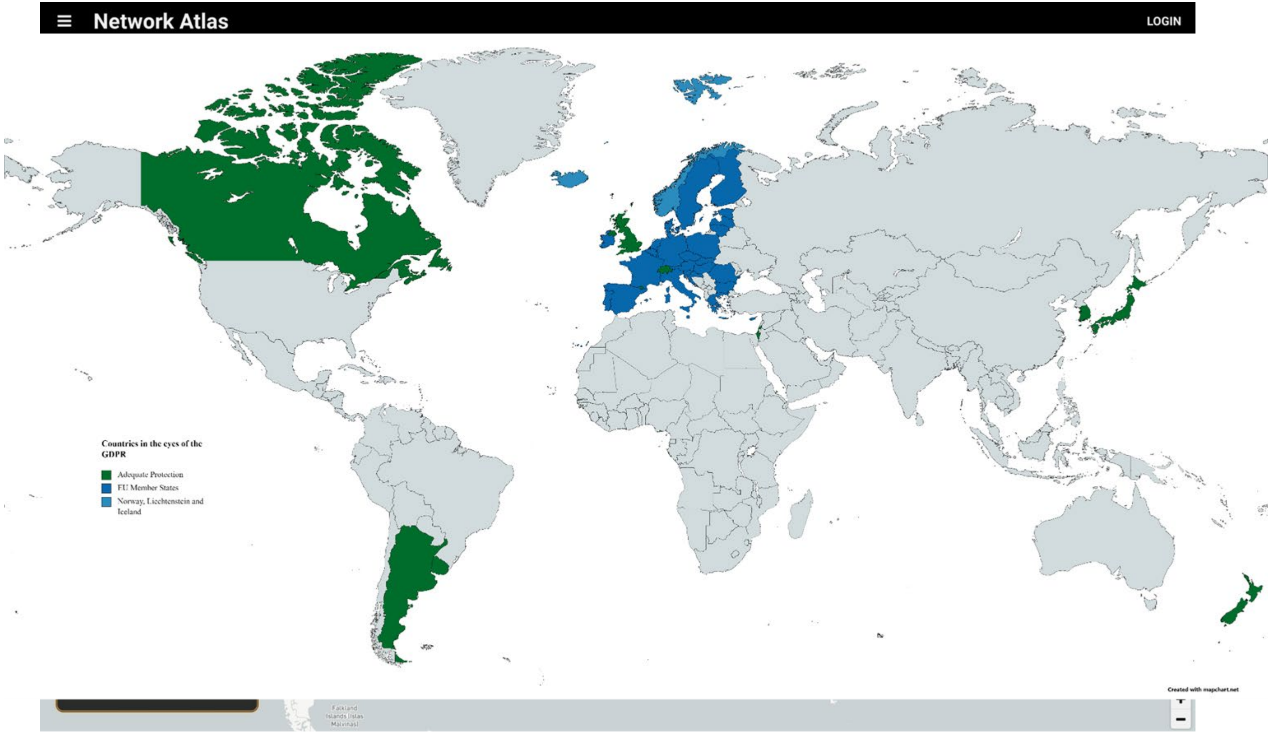
Post-brexit: UK GDPR is now in law, and is basically the same as the EU GDPR for now.

The **EU GDPR** (General Data Protection Regulation) restricts the transfer of **personal data** to countries outside of the EU.

One can process the data if it’s done in a country covered by an EU/UK “**adequacy decision**”

The UK and EU consider each other to have this.

Otherwise one needs a **standard contractual clause** (these have begun to diverge between the EU and UK).


≡ Network Atlas
LOGIN
Countries in the eyes of the GDPR
Adequate Protection
EU Member States
Norway, Liechtenstein and Iceland
Created with mapchart.net
Falkland Islands (Islas Malvinas)

Privacy and the GDPR…

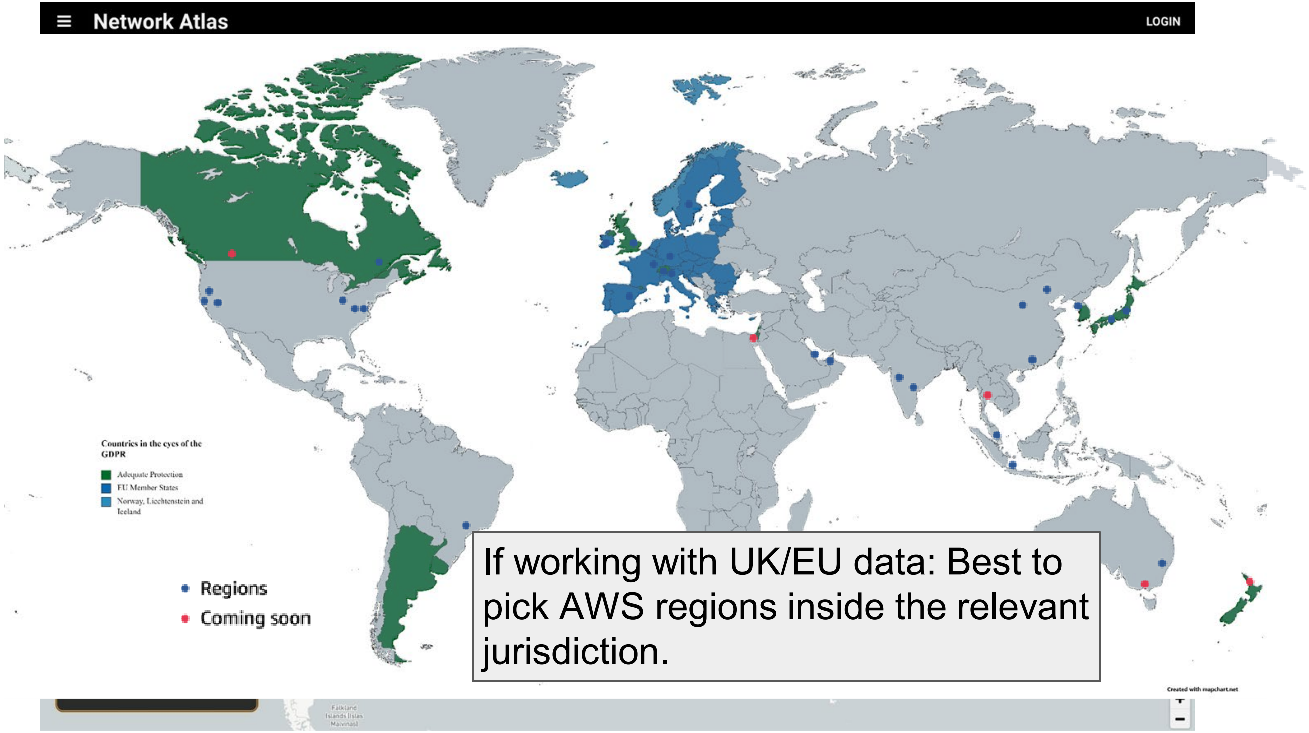
Post-brexit: UK GDPR is now in law, and is basically the same as the EU GDPR for now.

The **EU GDPR** (General Data Protection Regulation) restricts the transfer of **personal data** to countries outside of the EU.

One can process the data if it's done in a country covered by an EU/UK "**adequacy decision**"

The UK and EU consider each other to have this.

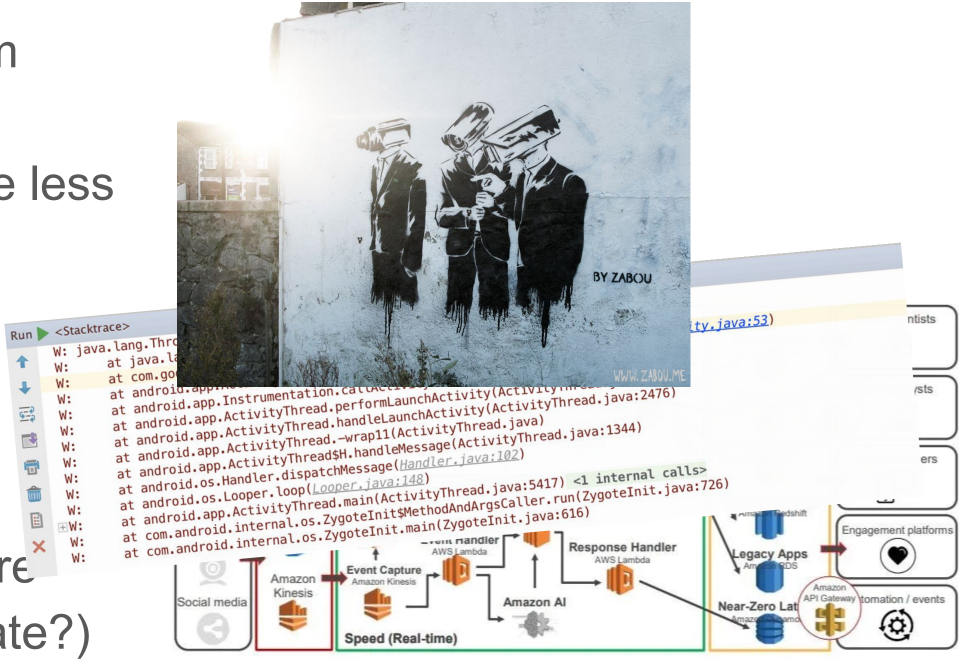
Otherwise one needs a **standard contractual clause** (these have begun to diverge between the EU and UK).


≡ Network Atlas
LOGIN
Countries in the eyes of the GDPR
🟩 Adequate Protection
🟦 EU Member States
🟦 Norway, Liechtenstein and Iceland
● Regions
● Coming soon
Created with mapchart.net
Falkland Islands (Islas Malvinas)

If working with UK/EU data: Best to pick AWS regions inside the relevant jurisdiction.

Why stay Local…

- May need to **redesign** algorithm
- Managing the **pipeline**.
- Typically such environments are less **reliable** and less **predictable**.
  - Far more difficult to **debug**.
- **Shifting data** to/from the servers might be a bottleneck.
- **Security** and **privacy** (where are you sending the data? Is it private?)
- **Cheaper**.



```
W: java.lang.Thr
W:     at java.la
W:     at com.go
W:     at android.app.Act
W:     at android.app.Instrumentation.callActiv
W:     at android.app.ActivityThread.performLaunchActivity(ActivityThread
W:     at android.app.ActivityThread.handleLaunchActivity(ActivityThread.java:2476)
W:     at android.app.ActivityThread.-wrap11(ActivityThread.java)
W:     at android.app.ActivityThread$H.handleMessage(ActivityThread.java:1344)
W:     at android.os.Handler.dispatchMessage(Handler.java:102)
W:     at android.os.Looper.loop(Looper.java:148)
W:     at android.app.ActivityThread.main(ActivityThread.java:5417) <1 internal calls>
W:     at com.android.internal.os.ZygoteInit$MethodAndArgsCaller.run(ZygoteInit.java:726)
W:     at com.android.internal.os.ZygoteInit.main(ZygoteInit.java:616)
W:
```

ity.java:53)

Images from:
https://frogssavethequeen.files.wordpress.com/2014/08/surveillance-graffiti1-web-version.jpg
https://developer.android.com/studio/debug/stacktraces
https://medium.com/rock-your-data/getting-started-with-apache-kafka-efc616bd6dd5

Service Models

There are different **service models** available: the trade off is typically complexity vs flexibility:

- **Infrastructure** as a Service (IaaS) - high level of control
  - Provides virtual servers, storage, networking resources, and other essential elements.
  - E.g. **AWS EC2**, Microsoft Azure Virtual Machines, Google Compute Engine.

Service Models

There are different **service models** available: the trade off is typically complexity vs flexibility:

- **Infrastructure** as a Service (IaaS) - high level of control
    - Provides virtual servers, storage, networking resources, and other essential elements.
    - E.g. **AWS EC2**, Microsoft Azure Virtual Machines, Google Compute Engine.

- **Platform** as a Service (PaaS) [always on] - medium
- **Function** as a Service (FaaS) [event driven] - low
    - Offers a development and deployment platform on top of the infrastructure.
    - You provide software/code, but the service starts up virtual machines, allocates storage, runs databases, etc.
    - E.g. website with database / **AWS Lambda**, Azure Functions, Google Cloud Functions.

Service Models

There are different **service models** available: the trade off is typically complexity vs flexibility:

- **Infrastructure** as a Service (IaaS) - high level of control
    - Provides virtual servers, storage, networking resources, and other essential elements.
    - E.g. **AWS EC2**, Microsoft Azure Virtual Machines, Google Compute Engine.

- **Platform** as a Service (PaaS) [always on] - medium
- **Function** as a Service (FaaS) [event driven] - low
    - Offers a development and deployment platform on top of the infrastructure.
    - You provide software/code, but the service starts up virtual machines, allocates storage, runs databases, etc.
    - E.g. website with database / **AWS Lambda**, Azure Functions, Google Cloud Functions.

- **Software** as a Service (SaaS) - low
    - Provides ready-to-use software applications over the internet
    - E.g. Google Docs

Service Models

There are different **service models** available: the trade off is typically complexity vs flexibility:

- **Infrastructure** as a Service (IaaS) - high level of control
    - Provides virtual servers, storage, networking resources, and other essential elements.
    - E.g. **AWS EC2**, Microsoft Azure Virtual Machines, Google Compute Engine.

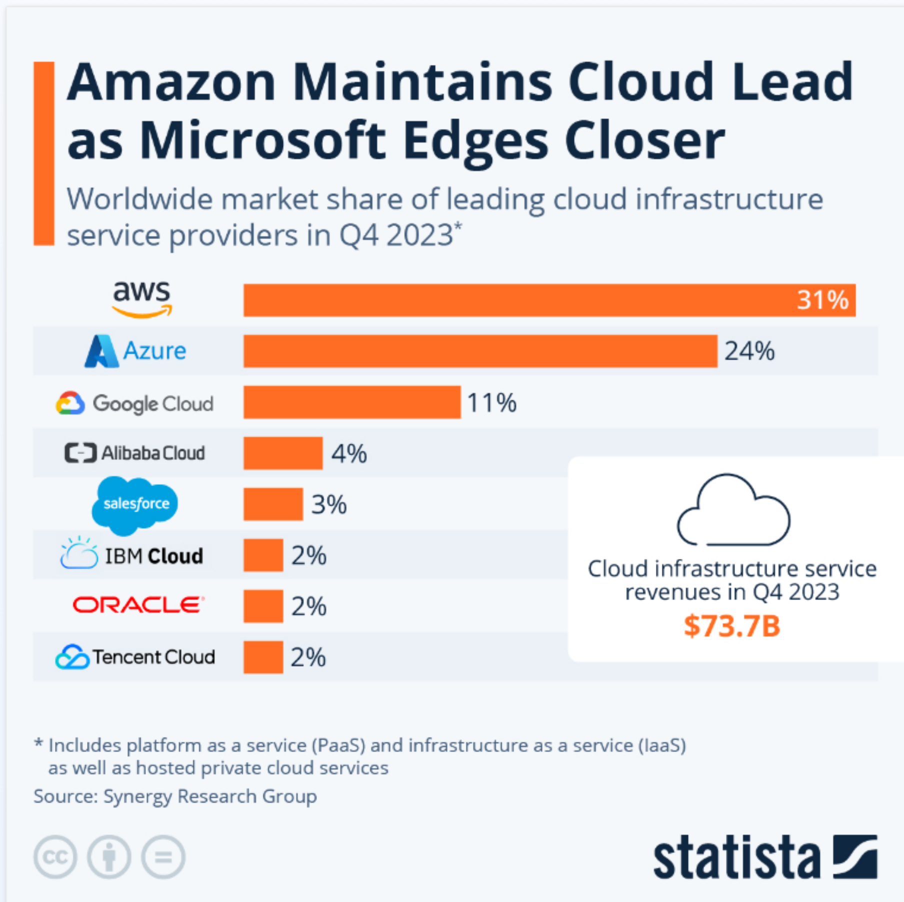
- **Platform** as a Service (PaaS) [always on] - medium
- **Function** as a Service (FaaS) [event driven] - low
    - Offers a development and deployment platform on top of the infrastructure.
    - You provide software/code, but the service starts up virtual machines, allocates storage, runs databases, etc.
    - E.g. website with database / **AWS Lambda**, Azure Functions, Google Cloud Functions.

- **Software** as a Service (SaaS) - low
    - Provides ready-to-use software applications over the internet
    - E.g. Google Docs

Infrastructure as Code (IaC) - cloud configuration specified in code.

Cloud Computing Platforms

We will focus on **AWS** (others similar).



Amazon Maintains Cloud Lead as Microsoft Edges Closer

Worldwide market share of leading cloud infrastructure service providers in Q4 2023*

Cloud infrastructure service revenues in Q4 2023
$73.7B

* Includes platform as a service (PaaS) and infrastructure as a service (IaaS) as well as hosted private cloud services

Source: Synergy Research Group

Cloud Computing Platforms

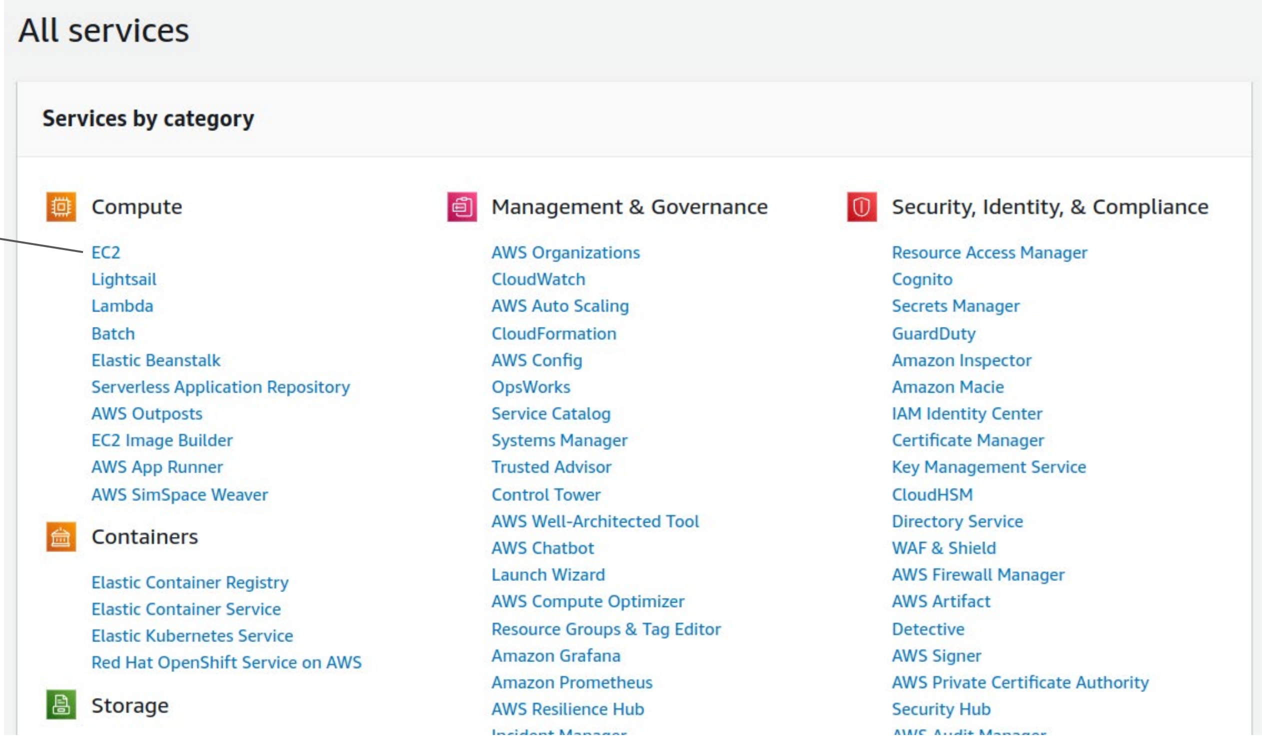
We will focus on **AWS** (others similar).

Huge range of services.

Amazon Elastic Compute Cloud (**EC2**) allows users to rent virtual machines.

Amazon Maintains Cloud Lead as Microsoft Edges Closer

Worldwide market share of leading cloud infrastructure service providers in Q4 2023*



All services

Services by category

Compute

EC2
Lightsail
Lambda
Batch
Elastic Beanstalk
Serverless Application Repository
AWS Outposts
EC2 Image Builder
AWS App Runner
AWS SimSpace Weaver

Containers

Elastic Container Registry
Elastic Container Service
Elastic Kubernetes Service
Red Hat OpenShift Service on AWS

Storage

Management & Governance

AWS Organizations
CloudWatch
AWS Auto Scaling
CloudFormation
AWS Config
OpsWorks
Service Catalog
Systems Manager
Trusted Advisor
Control Tower
AWS Well-Architected Tool
AWS Chatbot
Launch Wizard
AWS Compute Optimizer
Resource Groups & Tag Editor
Amazon Grafana
Amazon Prometheus
AWS Resilience Hub
Incident Manager

Security, Identity, & Compliance

Resource Access Manager
Cognito
Secrets Manager
GuardDuty
Amazon Inspector
Amazon Macie
IAM Identity Center
Certificate Manager
Key Management Service
CloudHSM
Directory Service
WAF & Shield
AWS Firewall Manager
AWS Artifact
Detective
AWS Signer
AWS Private Certificate Authority
Security Hub
AWS Audit Manager

AWS: Examples of Instance Types

On-demand (pay per hour) or Reserved

AWS: Examples of Instance Types

On-demand (pay per hour) or Reserved

Cheapest: t4g.nano [burstable, smallest instance] for 3 years $41 (ex. VAT).

- Also need to pay a few pence for storage, data transfer, etc.

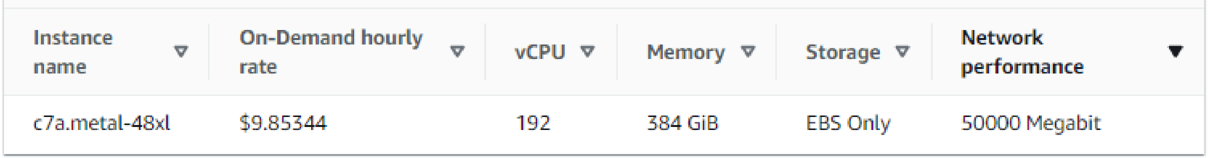
AWS: Examples of Instance Types

On-demand (pay per hour) or Reserved

Cheapest: t4g.nano [burstable, smallest instance] for 3 years $41 (ex. VAT).

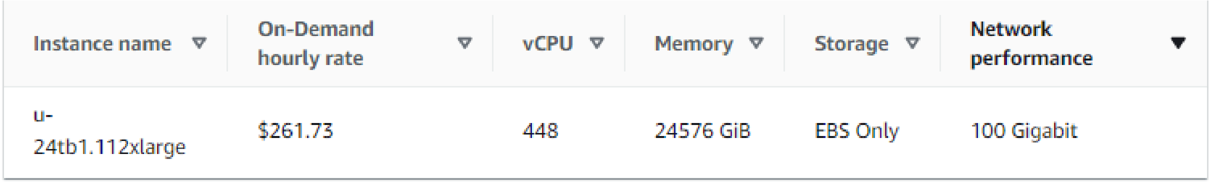
- Also need to pay a few pence for storage, data transfer, etc.

Compute optimised



Instance name	On-Demand hourly rate	vCPU	Memory	Storage	Network performance
c7a.metal-48xl	$9.85344	192	384 GiB	EBS Only	50000 Megabit

Memory optimised



Instance name	On-Demand hourly rate	vCPU	Memory	Storage	Network performance
u-24tb1.112xlarge	$261.73	448	24576 GiB	EBS Only	100 Gigabit

AWS: Examples of Instance Types

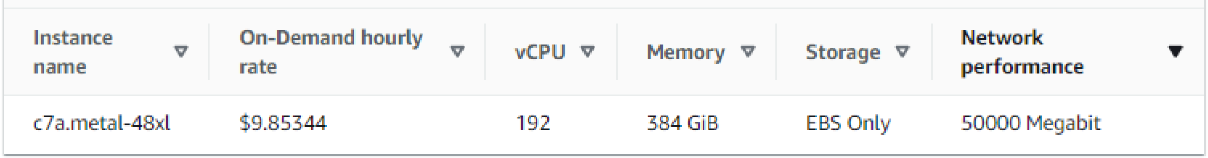
Other decisions: Storage type (e.g. HDD vs SSD), what OS will you install, etc. We will explore this in more detail in the lab.

On-demand (pay per hour) or Reserved

Cheapest: t4g.nano [burstable, smallest instance] for 3 years $41 (ex. VAT).

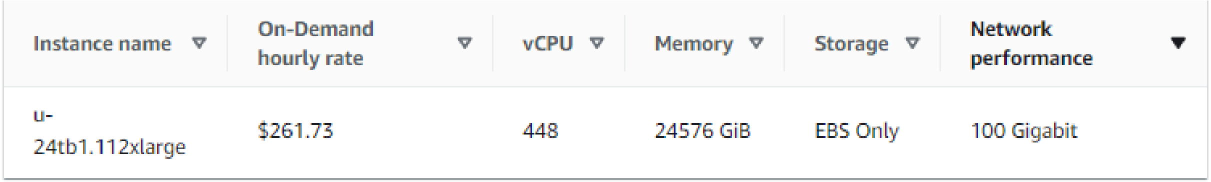
- Also need to pay a few pence for storage, data transfer, etc.

Compute optimised



Instance name ▽	On-Demand hourly rate ▽	vCPU ▽	Memory ▽	Storage ▽	Network performance ▼
c7a.metal-48xl	$9.85344	192	384 GiB	EBS Only	50000 Megabit

Memory optimised



Instance name ▽	On-Demand hourly rate ▽	vCPU ▽	Memory ▽	Storage ▽	Network performance ▼
u-24tb1.112xlarge	$261.73	448	24576 GiB	EBS Only	100 Gigabit

There are also instances that have one or more **GPUs**.

Some tools widely used…

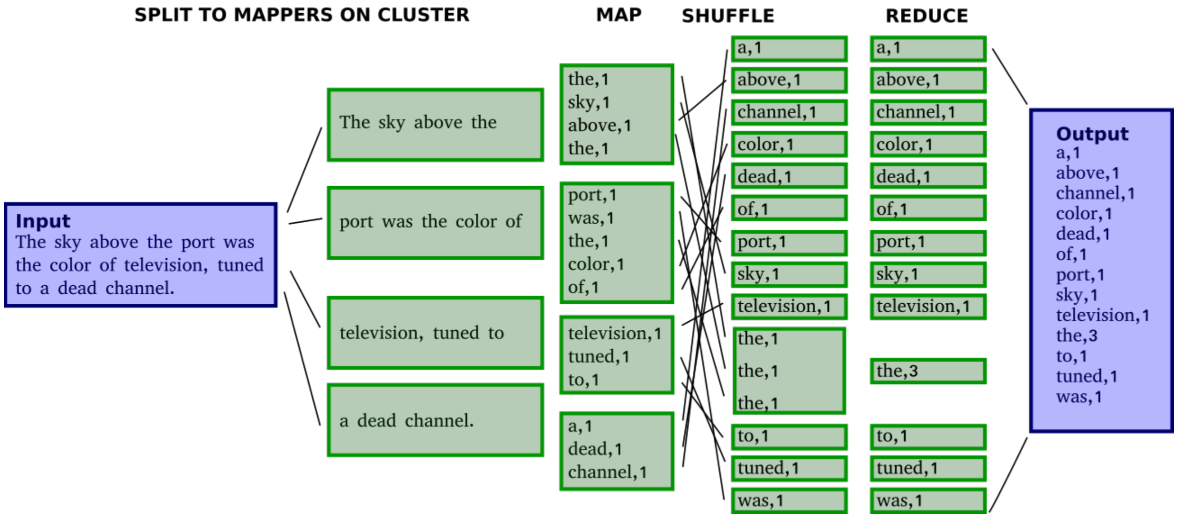
- MapReduce
- Apache Spark
- Terraform
- Docker

MapReduce

- A programming model for processing **large datasets** in parallel across clusters.

- First, **breaks down** a large task into smaller, independent subtasks (**map** phase)

- **Shuffles** the results to be in the right locations and then aggregates them (**reduce** phase).



SPLIT TO MAPPERS ON CLUSTER

MAP

SHUFFLE

REDUCE

Input
The sky above the port was the color of television, tuned to a dead channel.

The sky above the

port was the color of

television, tuned to

a dead channel.

the,1
sky,1
above,1
the,1

port,1
was,1
the,1
color,1
of,1

television,1
tuned,1
to,1

a,1
dead,1
channel,1

a,1

above,1

channel,1

color,1

dead,1

of,1

port,1

sky,1

television,1

the,1
the,1
the,1

to,1

tuned,1

was,1

a,1

above,1

channel,1

color,1

dead,1

of,1

port,1

sky,1

television,1

the,3

to,1

tuned,1

was,1

Output
a,1
above,1
channel,1
color,1
dead,1
of,1
port,1
sky,1
television,1
the,3
to,1
tuned,1
was,1

Example of a distributed word-count, using MapReduce

Apache Spark

- Apache SPARK is a **distributed cluster** computing framework.

- More flexible and faster than MapReduce
  - Functionalities for batch data processing (similar to MapReduce), real-time data processing, and machine learning.
  - In-memory processing rather than disk-based processing.

- Particularly useful for **iterative algorithms** (e.g. ML gradient descent).

We'll look at this in the lab in more detail.

Spark's RDDs function as a working set for distributed programs that offers a (deliberately) restricted form of distributed shared memory.

Inside Apache Spark the workflow is managed as a directed acyclic graph (DAG). Nodes represent RDDs while edges represent the operations on the RDDs.

- from wikipedia's article.

Terraform

- Defines the cloud infrastructure (virtual servers, storage, networking) as code files and can automate the creation and management across different cloud providers.

- Essential to manage in a scalable and automated way, especially for complex deployments.

- **Infrastructure** as code (IaC)

https://docs.aws.amazon.com/cli/latest/userguide/getting-started-install.html
https://app.terraform.io/app/getting-started/example

Docker

- A platform allows to put all the **software, libraries and configuration** files that you need to run an application into a **container**.

- It sort of replaces **virtual machines**, in a ‘lighter’ way, and makes it easier to **move data** between them.

- Simplifies development workflows, and ensures consistent application behavior across different environments.

- Mainly for linux, but typically not on HPC.

Other service… Lambda

AWS Lambda

- AWS Lambda allows you to put code on the cloud, **without having to deal with running a server**. The code is **triggered** by a variety of **events**, and can perform basic tasks in response. Typically **stateless**.

- **Function** as a Service (FaaS)

- Examples of where Lambda might be useful:

- IoT deployments	Parsing incoming data from sensors
- Automated backups	An hourly script run and checking the backups!
- ML deployment	Refresh air pollution map every 10 minutes
- Media conversion	Managing the deployment of infrastructure, Netflix
- Websites	Low traffic sites with scripts

Activities

- AWS EC2 - Infrastructure as a Service (IaaS)
- AWS Lambda - Function as a Service (FaaS)

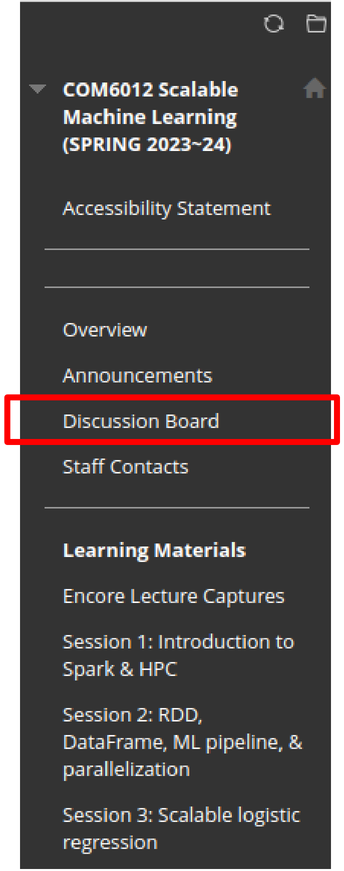
Take Home Messages

- Not always worth distributing your code (and lots of issues, including privacy).

- Service Models (Infrastructure-, Platform-/Function-, Software- as-a-Service).
    - EC2 is an example of Infrastructure-as-a-Service.
    - Lambda is an example of a Function-as-a-Service.

- Technology is changing quickly.
    - AWS EC2 is only about 15 years old.
    - Docker and Lambda are only about 10 years old.
    - Need to keep yourself updated if you enter this field.

Please note:


COM6012 Scalable Machine Learning (SPRING 2023~24)
Accessibility Statement
Overview
Announcements
Discussion Board
Staff Contacts
Learning Materials
Encore Lecture Captures
Session 1: Introduction to Spark & HPC
Session 2: RDD, DataFrame, ML pipeline, & parallelization
Session 3: Scalable logistic regression

Assignment Brief:
https://docs.google.com/document/d/1QSBkfnLLgf5qM0KWkeRayeAbrITRKY4JZ8aYZA8npro/edit

FAQ
https://docs.google.com/document/d/1QSBkfnLLgf5qM0KWkeRayeAbrITRKY4JZ8aYZA8npro/edit#bookmark=id.3ukytk47kosf

You are required to complete all assignment questions using **batch mode**.

Acknowledgement

The slides for this lecture are adapted from Dr Michael Smith's guest lecture slides in 2023.