General Immersion:

###Introduction###

AWS Accounts:

- They are not allowed to give us access to internal accounts.

- Get Access to an AWS account through TU Delft (re-immbursed)

- For the first phase we will use re-usable accounts (3 days at a time)

\* This would mean having to re-download zip files for training

\* Wouldn't be a price

\* Should use early in development

\* Avoid using for Testing, may be expensive

- Current test accounts don't have a lot of limitations, easy to use for development

- If we run into trouble with permission we should message Anastashia (She has access to Test accounts)

Today:

Go through 2 workshops

- 3 Tier web app

\* Introduction to core concerps

\* VPC: Perosnal part of the cloud

\* resources inside subnets

\* 2 instances

\* Autoscalling

- serverless web application

\* AWS Amplify, Amazon Cognito (Authentication), REST API, Lambda functions, DynamoDB

Each Workshop is 2 hours. Anastasia will be available throughout the day (excluding meetings)

###Presentation:###

Introduction:

- Foundational Blocks Architected in a Secure/Sustainable way.

- Things built on these blocks are your own responsibility

- Each Region -> Availability Zones -> Data-Center clusters

- Service availability differs from regions. Ireland has access to everything

- Services are cattegorized

AWS Well-Architected:

- List of best pracices on how workloads should be architected

- Operational Excellence, Security, Reliable, Percomance Efficient, Cost Oprimization, Sustainability

- Later in the procces these pillars will be evaluated

- We need to put things in multiple Availability Zones

- Lambda, by default, operates in multiple zones

AWS Compute:

- Amazon EC2: Virtual Machines

- Elastic Beanstalk: Deploys foundational elements we might need for a web applicaton. More managed way to deploy basic Architecture, It is a lttle limited in its capabilities. Can't launch serverless webapp

- EC2 Autoscaling:

- Lambda: Serverless functions

- Fargate: Serverless Containers while being serverless

- EKS: Takes care of control plane

- Batch: Run big batch jobs

Orchastrators: Services used

EKS vs ECS: 2 different services that orchastrate containers. ECS is the orchastrator itself without CPU or memory, you need to find where to run the them. Multiple instances within a cluster. Use Fargate to have a serverless way to run containers without having to assign EC2 instances.

We can also avoid using containers and use Lambda instead.

What Services we should use:

The labs will be similar to what we are creating (especially the second one)

We do however need a broad understanding of the foundations of AWS

We will create a services Diagram later

Processors:

- We will have to pick our own procesor (AMD, Intel, Amazon)

Storage:

EBS - Standard File system. Services that reside in 1 availability zone. File system storage attatched to EC2 instance

S3 - Object storage. Need to use S3 API to store objects or static website hosting.

Databases:

Relational, key-value

Security:

IAM - services allowing API calls to AWS services. Need IAM user/role with permissions.

Networking:

- VPC: logically isolated section of AWS cloud where you can launch AWS resources. You create a VPC within a region across Availability zones

- subnets exist within an availability zones and within a VPC

- Cloudfront: content delivery network. Works with S3 to have static content delivered quickly to end-users.

###Hands on Part###