COMPX527- Week 2: Lecture 1 & 2

AWS Overview

Overview

THE UNIVERSITY OF WAIKATO
Te Whare Wānanga o Waikato

- Amazon Web Services (AWS)
- Basic building blocks of a cloud service
- Overlay Services
- Operations, Administration & Management (OAM) Services
- Accessing AWS services
- AWS Regions
- AWS CLI

Amazon Web Services (AWS)

- A collection of IT infrastructure services and resources
  - Provided by Amazon
  - Available
    - Publicly
    - Over the Internet
    - On demand
  - Scalable
  - Pay as you go pricing
  - Founded in 2006
    - Initially conceived to scale Amazon's e-commerce website
  - Has the largest market share of all cloud computing service providers

Basic Components of a Cloud

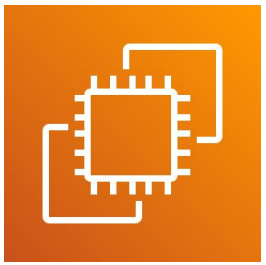
- There are four service components required for a functional cloud service:
  - Compute Service
  - Storage Service
  - Databases
  - Network

- All other services provided by a cloud service provider are built on top of these four services.

- AWS provides all these services and more.

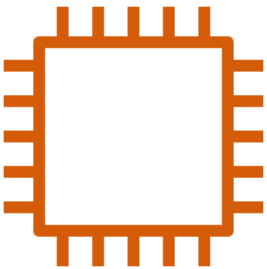
Compute Service

- Amazon Elastic Compute Cloud (EC2)
  - Provision virtual machines
    - On-demand
    - Different types and capacities available
    - Capacity can be quickly scaled up or down
    - Terminate your instances when you're not using them!



Amazon Elastic Compute Cloud (Amazon EC2)

Amazon Machine Images (AMI)



Instance

Compute Service

- Amazon Lambda

  - Serverless Compute functions
    - On-demand
    - Runs code in response to events
    - Automatic scaling

  - Bring your own code
    - C#, Java, Node.js, and many more

Storage Service

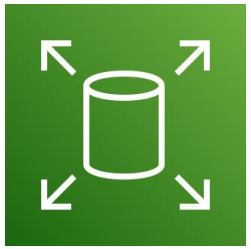
- Amazon Simple Storage Service (S3)
  - Distributed Object Storage
  - Store and Retrieve data anywhere, anytime
  - Everyone has access of your bucket

Amazon Simple Storage Service(Amazon EBS)

Bucket

Object

- Amazon Elastic Block Store (EBS)
  - Persistent block storage for EC2 instances
  - Think additional drives for servers
  - Persistent storage volume that is not tied to an EC2 instance.



Amazon Elastic Block Store (Amazon EBS)

Multiple Volumes

Snapshot

Volume

Database Service

- Amazon Aurora
  - Compatible with MySQL and PostgreSQL DB engines
- Amazon Relational Database Service (RDS)
  - Similar to MySQL in terms of functionality
- Amazon DynamoDB
  - Serverless
  - NoSQL database

They are:

- Scalable
- Administration tasks are abstracted away from end-users

Amazon Aurora

Amazon RDS

Amazon DynamoDB

Network Service

- Amazon Virtual Private Cloud (VPC)
  - Define a virtual network topology that closely resembles a traditional network in a datacenter
  - Logically isolated
  - End-user has total control over network

Amazon VPC

- Amazon Elastic Load Balancing (ELB)
  - Distributes incoming traffic to defined targets
    - EC2 instances
    - IP addresses
  - Can load balance traffic at:
    - Application Level - Layer 7 (HTTP, HTTPS)
    - Network Level - Layer 4 (TCP, UDP)

Amazon ELB

- Amazon Route 53
  - DNS service
  - Route user traffic to applications and services

Amazon Route 53

Overlay Services

- These are services that are built on top of the basic services
  - Amazon Simple Notification Service (SNS)
    - Messaging services for distributed systems and applications
  - Amazon Simple Email Service (SES)
    - Email sending service for transactional emails
  - Amazon CloudFront
    - Content Delivery Network Service
      - Effectively delivers data, video etc., close to the end-user
      - Requires compute, network and storage

Amazon SNS

Amazon SES

Amazon CloudFront

Operations, Administration & Mgt

- These are services required to securely deploy, operate and manage the different services available on AWS
  - AWS Calculator
    - Calculate the price of using AWS services
    - Compare cost of using different settings

  - AWS Identity, Access and Management (IAM)
    - Securely control access to AWS services
    - Create users and groups
    - Manage access and privileges for users and services

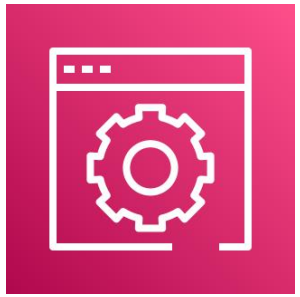
  - AWS CloudWatch
    - Monitor and analyse logs for AWS resources and services

  - AWS CloudTrail
    - Monitor and analyse logs for AWS API calls

  - AWS CloudFormation
    - Provision and manage AWS resources using templates

Accessing AWS Services

- AWS Management Console
  - Web interface
  - Good for performing simple operations

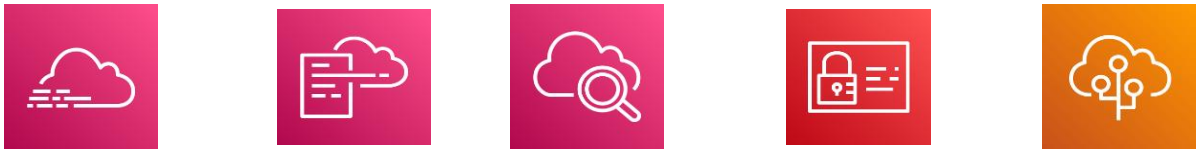


- AWS Command Line Interface
  - Recommended
  - Enables Automation
  - Repeatable and verifiable actions
  - Enables Infrastructure as code

- AWS Software Development Kits (SDKs)
  - Tailored for use with programming languages
  - Applications can interact with AWS services using SDKs
  - Also enables infrastructure as code

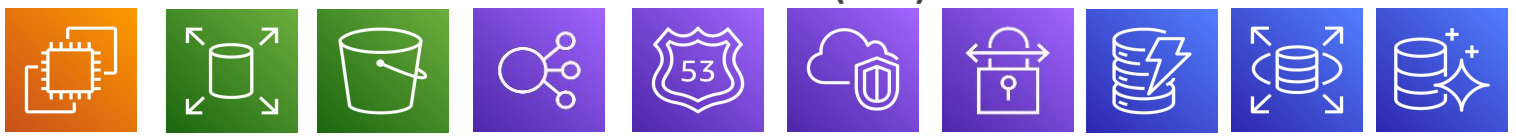
Tools

OAM Services



Overlay Services (PaaS)

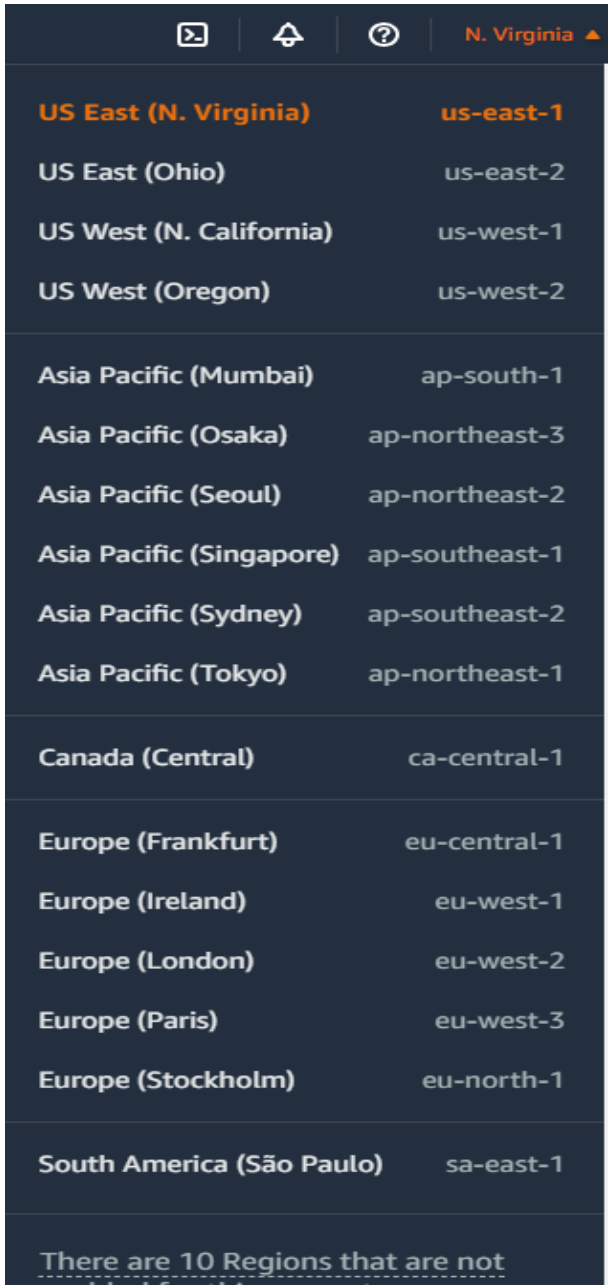
Basic Core Services (IaaS)



AWS Regions

- Regions
  - Multiple locations worldwide
    - 37 Geographic Regions
- Each region is isolated from the other regions
    - Availability Zones
      - Multiple, isolated locations in a region
      - 117 Availability Zones

- Local Zones
      - An extension of the regio- Places services closer to end-users
      - 43 Local Zones



AWS CLI

AWS CLI

THE UNIVERSITY OF WAIKATO
Te Whare Wānanga o Waikato

- Used to interact with AWS Services using the command line.
- For example:
  - Create new EC2 instances
  - Get details about instances
  - Run commands on instances remotely
  - Terminate instances

  https://awscli.amazonaws.com/v2/documentation/api/latest/reference/index.html

Installing AWS CLI

- Can install on Linux, Windows and Mac
- https://docs.aws.amazon.com/cli/latest/userguide/getting-started-install.html

AWS Security Credentials

THE UNIVERSITY OF WAIKATO
Te Whare Wānanga o Waikato

- Account Id, username and password:
  - For logging into the online management console
- AWS Access Keys:
  - For running commands with AWS CLI
  - Access key ID and Secret access key
  - Created in the IAM Management Console
    - Select Users page
    - Select the user to see more details
    - Go to security credentials tab
    - Create new access key
    - Save keys somewhere safe

AWS configuration

THE UNIVERSITY OF WAIKATO
Te Whare Wānanga o Waikato

- Configure AWS CLI with AWS credentials

```
> aws configure
AWS Access Key ID [None]: **********************
AWS Secret Access Key [None]: ****************************************
Default region name [None]: us-east-1
Default output format [None]: json
```

Using AWS CLI

THE UNIVERSITY OF WAIKATO
Te Whare Wānanga o Waikato

- Test your configuration by creating an S3 bucket:

```
> aws s3 mb s3://my-first-COMPX527
```

- This makes a new bucket called my-first-COMPX527
- Bucket names are unique!
- To list all your buckets:

```
> aws s3 ls
```

- To copy a file into a bucket:

```
> aws s3 cp test.txt s3:// my-first-COMPX527
```

- To see the files in a bucket:

```
> aws s3 ls s3:// my-first-COMPX527
```

https://docs.aws.amazon.com/cli/latest/userguide/cli_code_examples.html

More examples

THE UNIVERSITY OF WAIKATO
Te Whare Wānanga o Waikato

- List EC2 instances by instance type

```
> aws ec2 describe-instances
--query "Reservations[*].Instances[*].{Instance:InstanceId,
                Subnet:SubnetId,PublicIP:PublicIpAddress,
                VolumeInfo:BlockDeviceMappings,Status:State.Name,
                SecurityGroups:SecurityGroups[*],
                Type:InstanceType,AZ:Placement.AvailabilityZone}"
--filters Name=instance-type,Values=t2.micro
--output table
```

More examples

THE UNIVERSITY OF WAIKATO
Te Whare Wānanga o Waikato

- List EC2 instances by account id

```
> aws ec2 describe-instances
--query "Reservations[*].Instances[*].{Instance:InstanceId,
            Subnet:SubnetId,PublicIP:PublicIpAddress,
            VolumeInfo:BlockDeviceMappings,Status:State.Name,
            SecurityGroups:SecurityGroups[*],
            Type:InstanceType,AZ:Placement.AvailabilityZone}"
--filters Name=owner-id,Values=285446995906
--output table
```

More examples

- Create a new DynamoDB table

```
> aws dynamodb create-table
--table-name MusicCollection
--attribute-definitions
                        AttributeName=Artist,AttributeType=S
            AttributeName=SongTitle,AttributeType=S
--key-schema        AttributeName=Artist,KeyType=HASH
            AttributeName=SongTitle,KeyType=RANGE
--provisioned-throughput ReadCapacityUnits=1,WriteCapacityUnits=1
```

AWS SDK

- Access AWS services directly from your code
- Javascript, Python, Java, .NET, Node.js, C++, PHP, Ruby, Go

```
import boto3

s3 = boto3.resource('s3')
bucket = s3.Bucket('my-bucket')
for obj in bucket.objects.all():
    print(obj.key)
```

Questions?

- Reminder: Terminate your EC2 instances when you're done using them