# AWS

# Identity Access Management (IAM)

## Terms

* Users – End users (people)
* Groups – a collection of users under one set of permissions. For example: HR, Marketing etc. each of them requiring different types of permissions.
* Roles – you create roles (e.g. S3-ReadOnlyAccess) and then you can assign them to users or other AWS resources (e.g. EC2 )
* Policies – A document that defines one (or more) permissions. The policy can then be attached to a user/group/role.

## What is IAM

Allow you to manage users and their level of access to the AWS console.

IAM gives you:

* Centralized control of your AWS account
* Shared access to your AWS account
* Granular Permissions
* Identity Federation (allowing users to login to your site using their Active Directory, Facebook or LinkedIn etc accounts)
* Multifactor Authentication
* Provides temporary access for users/devices and services, as necessary.  
  For example, allowing users of your app to retrieve data from your S2 database.
* Allows you to set up you own password rotations policy
* Integrated with many AWS services
* Supports PCI DSS Compliance for applications associated with the payments industry.
* IAM is universal. It does not apply to specific regions.

## IAM Lab

1. Create a new AWS account: <https://aws.amazon.com/premiumsupport/knowledge-center/create-and-activate-aws-account/>
2. Create user accounts. Use your root account only when you absolutely don’t have another choice. For everything else – use a user account.  
   NOTE: When you create the account, this is the only time that you can see the user’s Secret Access Key which is required to run CLI commands. So, **Download the .csv and keep it safe. Otherwise, you won’t be able to access this account anymore!! Or even better, save it in keypass!!**
   1. User & paswd – required for logging into the Console
   2. Access Key ID & Secret Access Key – required for using the CLI
   3. New users don’t have any permissions. You need to give them permissions.
3. Create Roles: for example: if we want to allow EC2 instances to write to our database (S3 buckets).

Elastic Compute Cloud

Elastic Compute Cloud

# Elastic Compute Cloud - EC2

* Secure, resizable compute capacity in the cloud.
* Like a virtual machine in the cloud (hosted in AWS)
* Allow super-easy scaling
* The capacity you want when you need it
* You have complete access to your instances (root)
* Pay only for what you use, when you use it.
* You can select your capacity and then grow/shrink it as you need.
* Pricing models:
  + On demand – you pay by the hour/second depending on the type of instance you run
    - Good for application that are short-term, spiky or unpredictable workloads that cannot be interrupted.
    - For testing the waters – new application in development
  + Reserved Instance (RI) – you reserve the capacity for yourself for 1-3 years. This is regional and allow you to get up to 72% discount (as long as you’re in the same region)
    - Suitable when you know in advance that you need capacity X long term (years)
    - Require you to pay up front
    - Standard RI – a static capacity (you can’t change it) – up to 72% off
    - Convertible RIs – you can change the capacity for equal/greater value – up to 54%
    - Scheduled RIs – when you don’t need the capacity all the time but instead only X days every month so you can pay only for these re-occuring times.
  + Spot – Purchase unused capacity at discounts of up to 90% (fluctuate according to supply and demand). Once the capacity drop below your ask or the price above it, you will use your instance.
    - Suitable when you need to run something occasionally and you can spin it up from scratch easily (no long installations etc needed) and when it’s not time-critical.
  + Dedicated – a physical EC2 server dedicated for your use. This is the most expensive option.
    - This is suitable for when you have licenses that are tied to a physical HW or you have some regulations that forces you to use this option.

# For the exam

* See the Exam Blueprint pdf