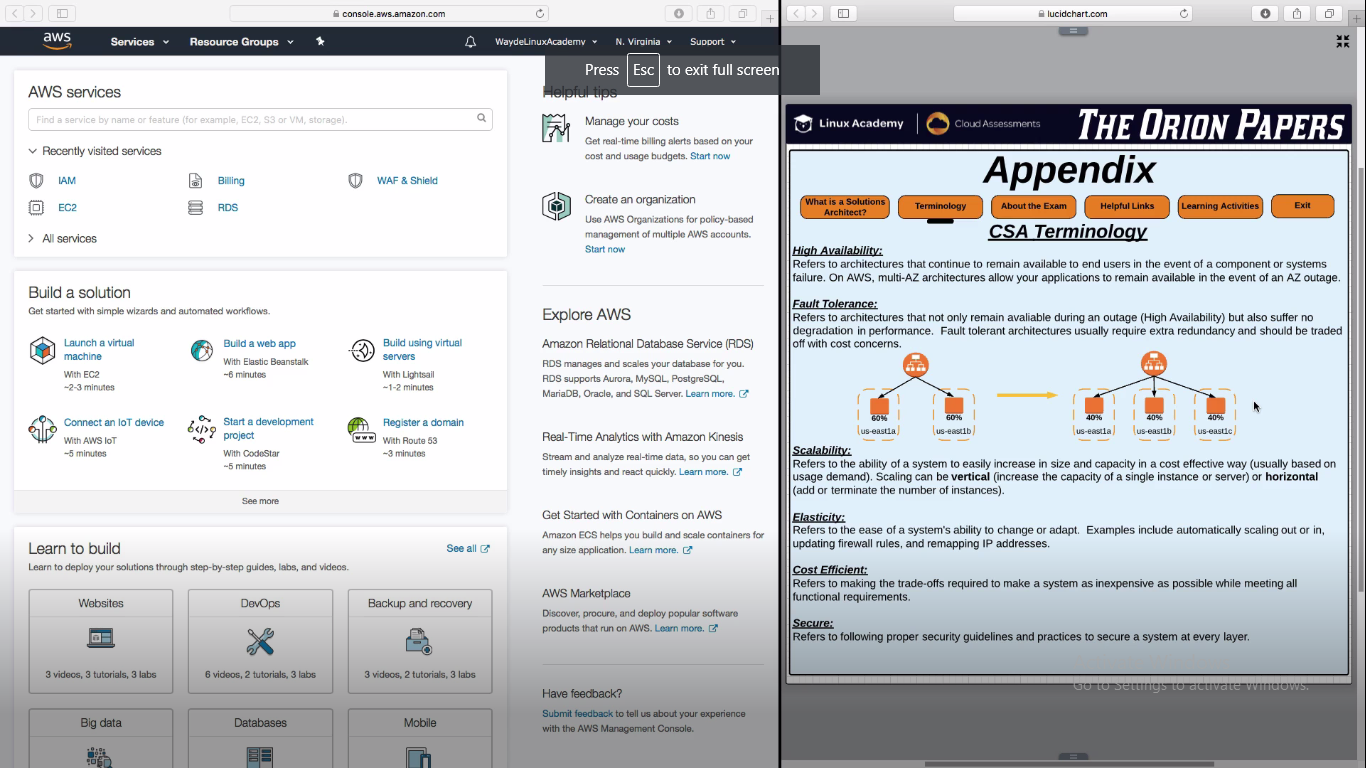
**Features for Well Architected Framework**

Secure can be in application level, OS level, networking level, access control, encryption etc

**High Availability**

1. Architecture your application in two availability zones in a region. So even if one zone goes down, application will be up and running in the other zone

**Fault Tolerance**

1. Architecture your application that when we have the high availability and an application is made to run in one zone (considering other zone is down), it should be capable of supporting the full load (as we are running with one zone)

**Scalability**

1. It is the feature to extend the decrease the system resources based on demand (high load and less load)

**Vertical Scaling:**

This is traditional scalling. Adding additional CPU, NAS, RAM to a server

**Horizontal Scaling:**

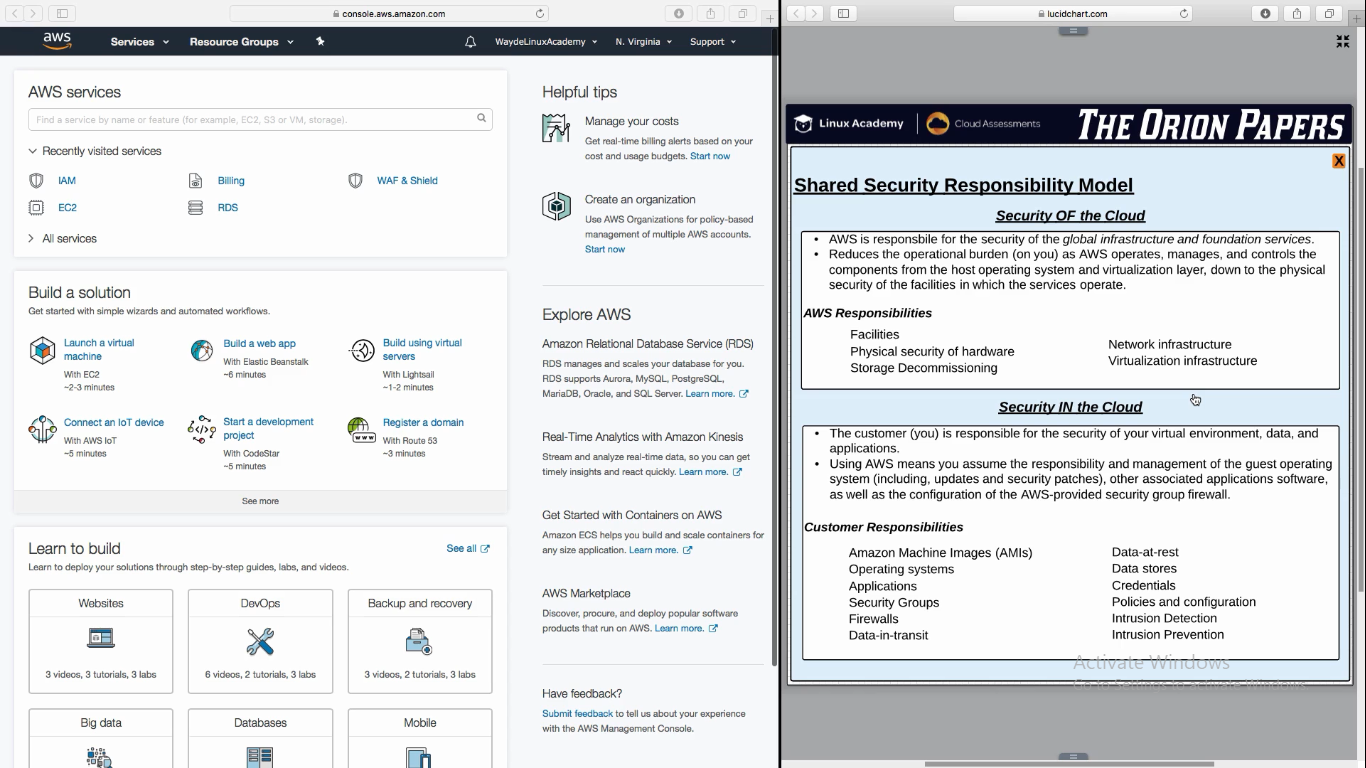
To spin new instance of an application when ever needed

**Elasticity**

1. Elasticity refers to how flexible our system is
2. Eligibility to adopt or change
3. If system automatically scales, on the fly should be able to update firewall, blacklist ips, remap ips of underlying instance etc system/app should adopt to it

AWS API services are software based. Whenever we launch a instance, load balancer or DB it interacts with a software layer. So any requirement can be scripted for automation

**Shared Security Model**

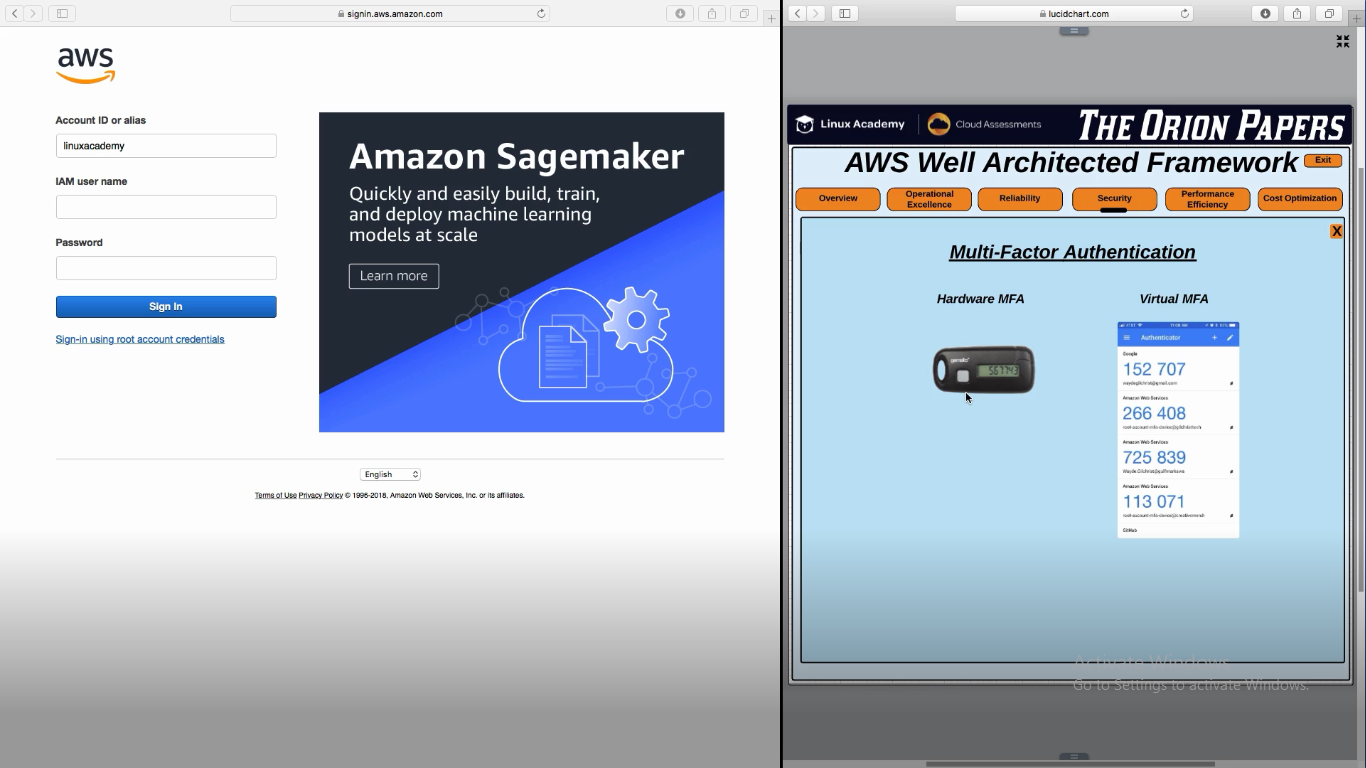
****

AWS Takes care of the API endpoint service for the AWS services (EC2, S3, RDB etc)

**Accessing AWS in console**

1. Login to browser
2. Aws.amazon.com
3. There are two types of users to login
   1. AWS Root user account
      1. Root user will login with a email id
      2. He/she is the user who created the AWS Account
      3. They have full privilege to do everything
   2. AWS Users account
      1. Dev Account
      2. Prod Account

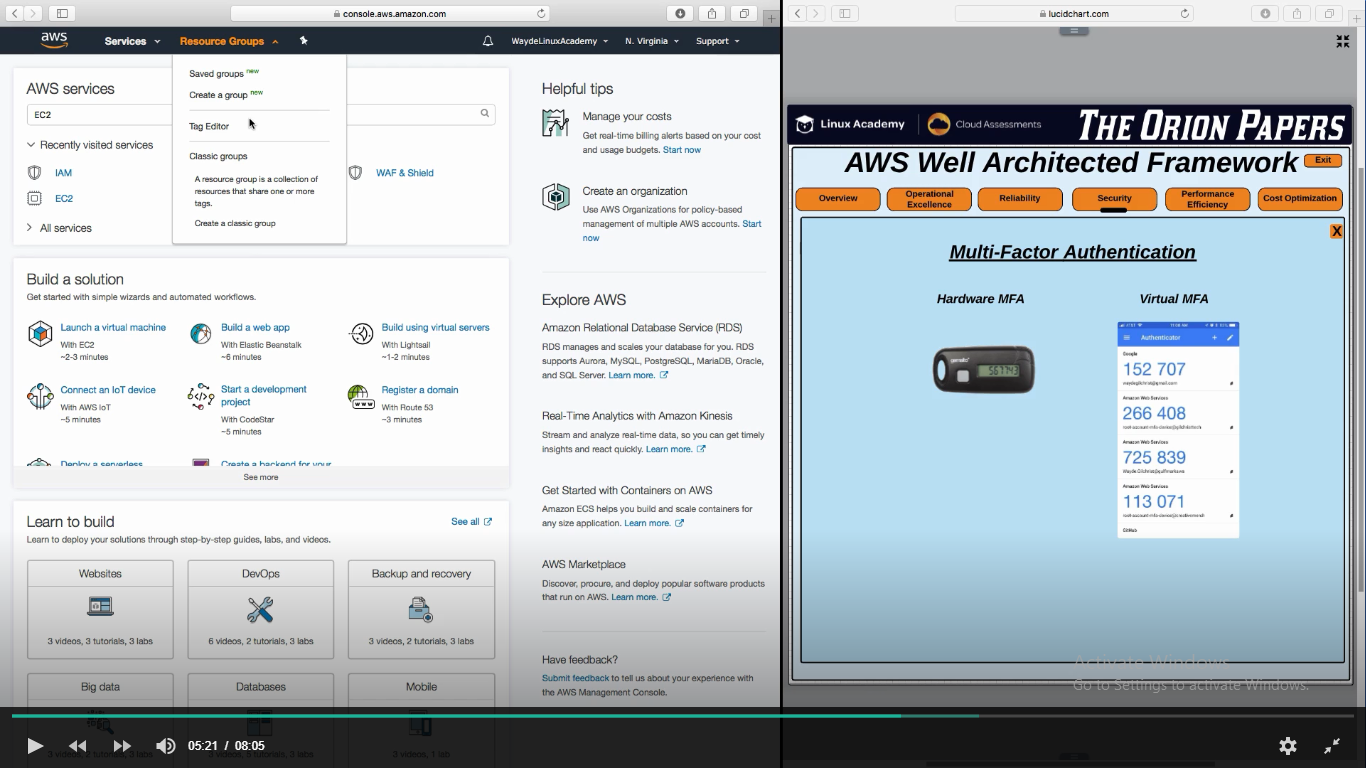
User account will ask for account ID, IAM UserName and password to login

****

**MFA – MultiFactor Authentication**

6 digit pin and changes every 30s

**Console Tour**

****

**Resource Group:**

Using tag editor we can group related instances/services (of one project) and give a tag and create a resource group. This group will be a collection of all resources needed for a project/application

Pin next to resource group is to create a shortcut button to any of the service which we wish to add to it

**Support**

Support -> My support case -> create case to raise a ticket for support people in AWS

Support -> Health Events -> AWS Outage will be reported here

Support -> AWS Documents, Trainings, Forums etc

