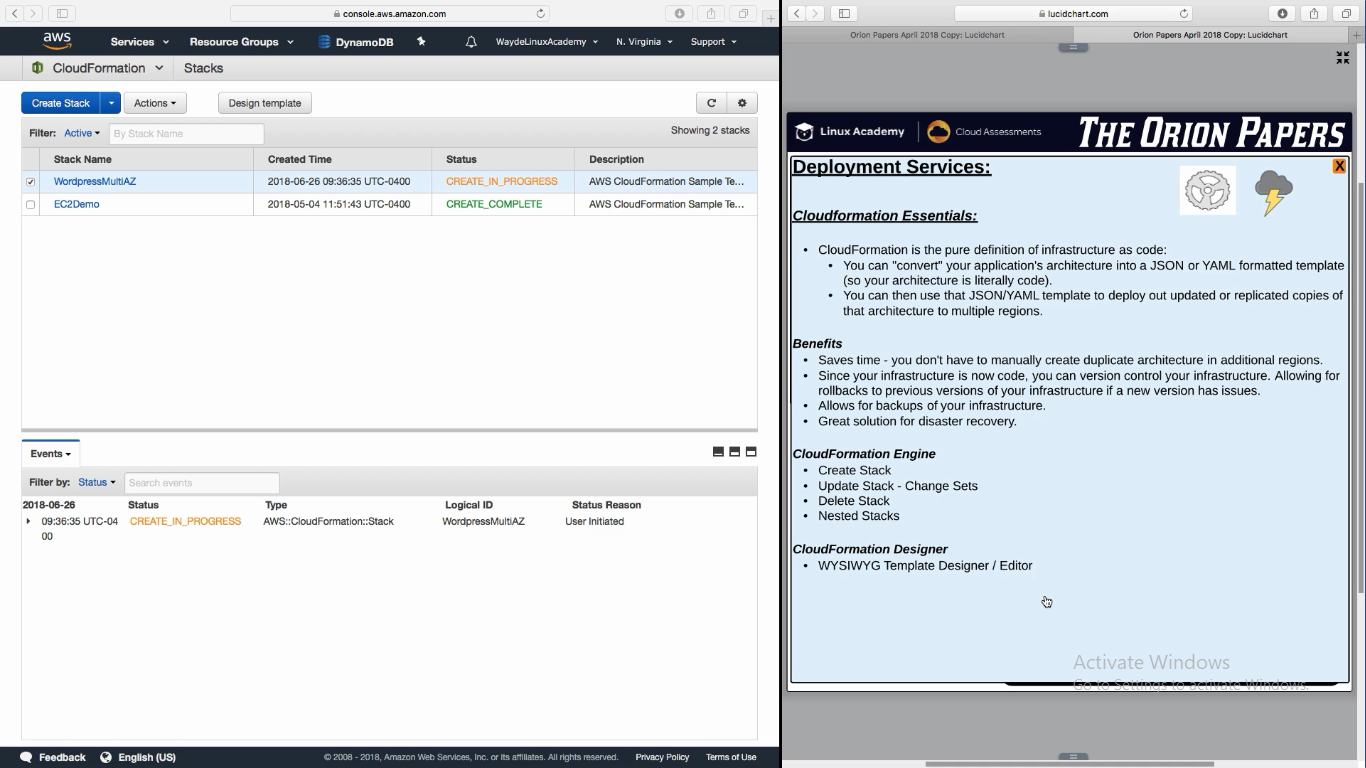
**Deployment Service**

1. CloudFormation
2. Container Services
3. Elastic Bean Stalk

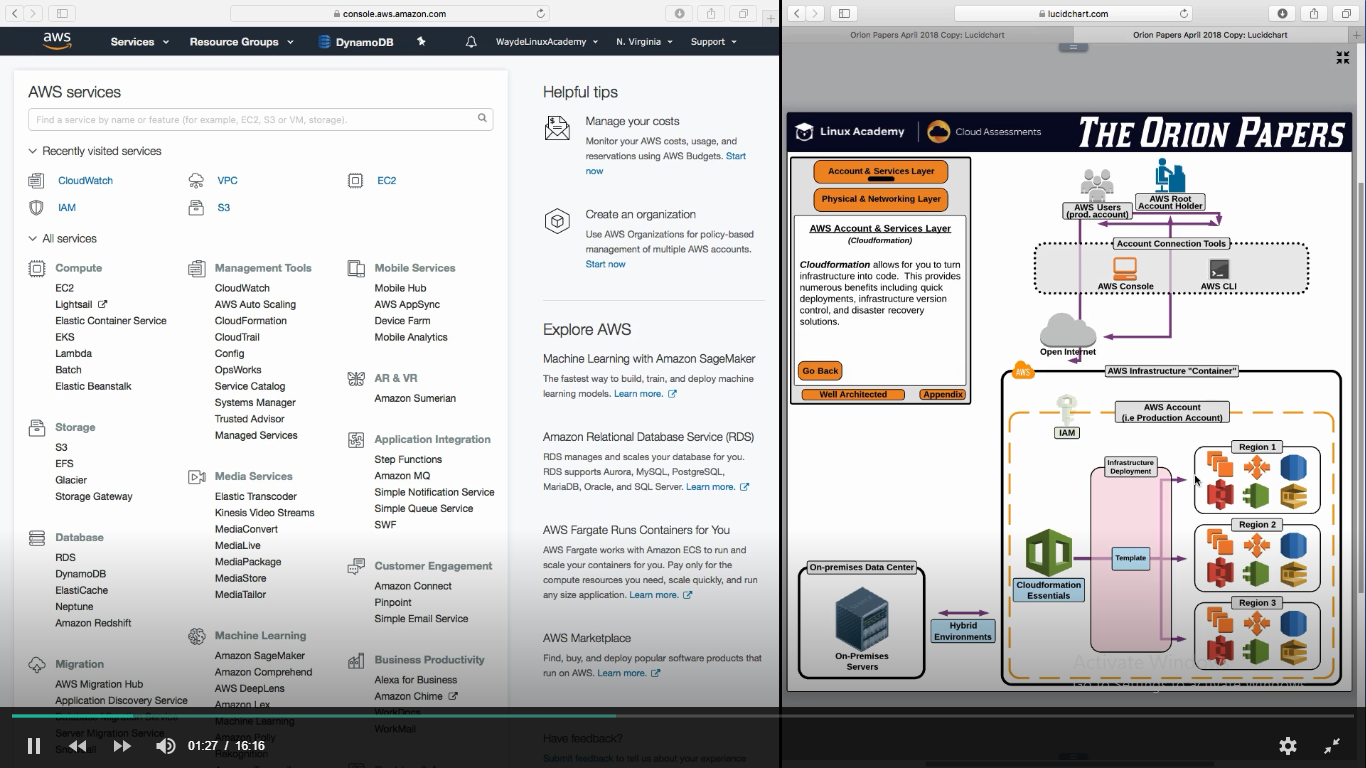
**Cloud Formation**

1. It gives infrastructure as a code and we can deploy our code in it
2. Instead of creating our own script for deploying (which shall have numerous API calls/functions) or creating a VPC etc for deployment, using cloud formation we can define a template (JSON or YAML). Cloud formation engine will read the template create a stack (infrastructure).
3. One template can create multiple stack
4. Using **Visual Designer** we can drag down resources (VPC, subnet) and create a workflow and it creates a cloud formation template

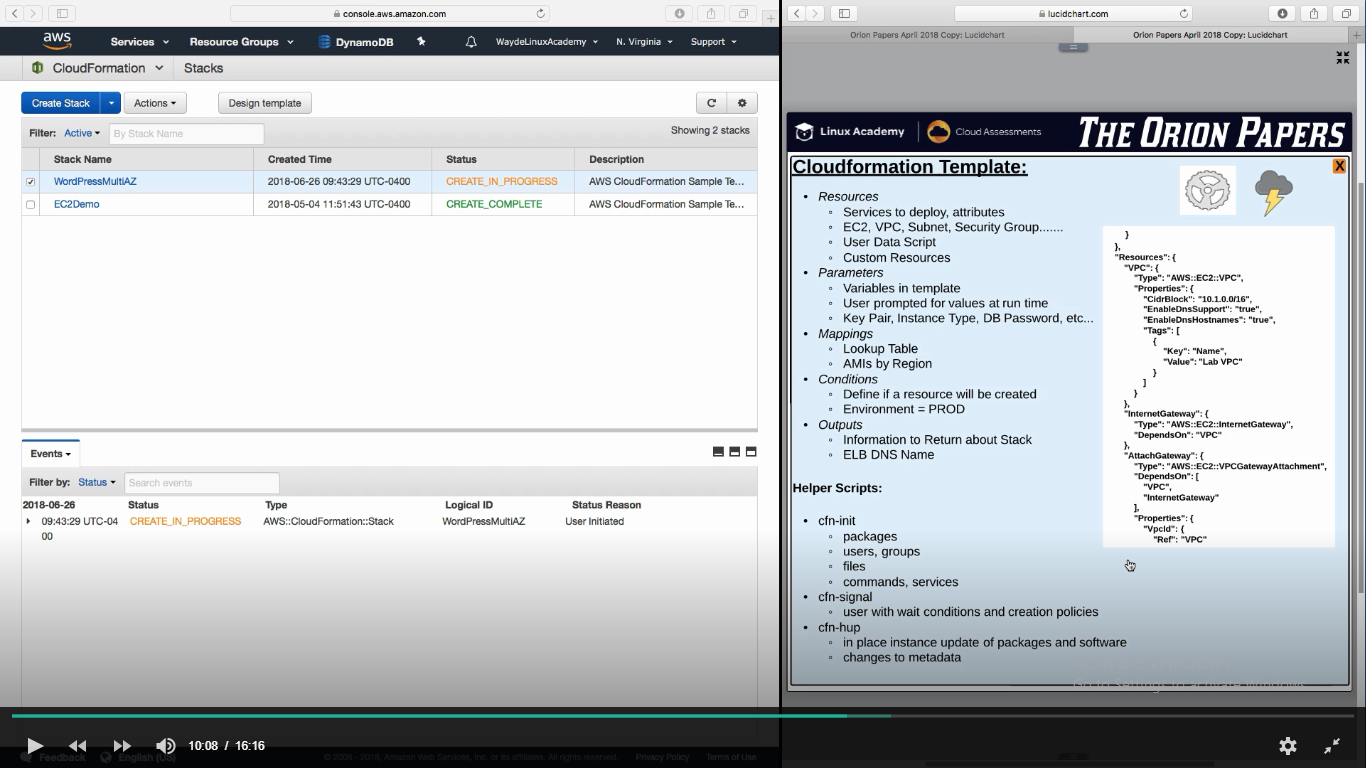


Benefits:

1. We can mimic prod changes by copying the prod template and test it in an environment and role it in prod



Cloud Formation Template – JSON



**Container Service**

Deploying our docker container in ECS (Elastic Container Service)

ECS is an AWS solution for launching docker solution for docker container orchestrations

AWS have a registry called Container Registry (ECR) which stores all docker images in that registry

**ECR:**

1. In this registry we will define the task definition

**Task Defnition:**

1. Task definition will point to one or more docker images that are going to run on that container and define how much cpu/memory to be assigned for this container. All containers will run on the instance (AWS resource - Host OS)
2. Task definition have “Launch Type” which ask whether we want to run the task in the EC2 which we manage (here we have to manage the ECS cluster) or in a service called “Far Gate”. Far gate is a managed serveless solution funning EC2 instance on our behalf (we do not want to manage the ECS cluster)

**ECS cluster** have EC2 container instances and **ECS agent** (installed on it) which communicates with ECS and launch the task in the instances

**Elastic Container Service for Kubernetes (EKS)**

1. Start/stop container
2. Schedule containers
3. Manage kubernetes control panel

We have to use ALB (Application Load Balancer) which will process extern API call (from users over internet) and will path based routing to different container based on their port number

