**CLOUDFORMATION OVERVIEW**

* CloudFormation is a declarative way of outlining your AWS Infrastructure, for any resources
* For ex, within a CloudFormation template, you say:
* I want a security group
* I want 2 EC2 instances using this security groups
* I want an S3 bucket
* I want a Load balancer (ELB) in front of these machines
* Then CloudFormation creates those for you, in the right order, with the exact configuration that you specify

**Benefits of CloudFormation:**

* All your infrastructure is defined as a code (**Infrastructure as Code – IaC**)
* You will never ever create your resources manually
* Changes to the Infrastructure are reviewed through code
* **Cost**:
* Each resources within the stack are tagged with an identifier so you can easily see how much a stack costs you
* You can estimate the costs of your resources using the CloudFormation Template
* Savings Strategy: In Dev, you could automate the deletion of a templates at 5 PM and recreated at 8 AM, safely. This saves your costs as there are no resources between 5 PM and 8 AM.
* **Productivity**:
* Ability to destroy and recreate an infrastructure on the cloud on the fly
* Automated generation of Diagrams for your templates
* Declarative programming – no need to figure out ordering and orchestration)
* Don’t need to re-create the templates:
* Leverage existing templates on the web
* Leverage the documentation
* CloudFormation supports all AWS resources

**HANDS-ON**

* We can either use an S3 URL or upload a *.yaml* file from our local while creating a CloudFormation template. (Other details like – name etc., need to be filled)
* Once a template is created, we can go to the stack and then to the *Templates* tab to see the uploaded yaml file.