Docker Compose

Aws CodeDeploy Blue/Green Deployments

Core point of this talk

- *Migrate/Shift Your Application to CONTAINERS*

Build AMI Only Once

Required Packages

1. Docker
2. Docker-Compose
3. Aws CloudWatch Agent
4. Aws CodeDeploy Agent
5. Jq
6. Netcat/telnet
7. Net-tools
8. Awscli

Aws Infrastructure Setup

Terraform

Cloudformation

Manual Setup

Secure VPC, ALB, ASG, S3 for artifacts

Why Blue/Green Deployments?

- *Zero Downtime Deployments of our applications to New Version Switching*
- *Enable Auto Rollbacks (Incase of application health check failures)*
- *Blue/Green Deployment Strategy is Highly Adopted for Stable Deployments*

Why Docker?

- *Instead of Building AMI & Install Required Packages Again & Again, main focus is to build our Application Image with Dockerfile & Pushed to ECR.*

- *Easily switch between different version of application.*

- *Easy Deployments of Application by Simply pulling from ECR.*

Why Docker Compose?

- ***Complete application stack or Split application stack***

- *Example:*
- *Frontend in Single EC2 with docker-compose.*

- *Backend in Single EC2 with docker-compose.*

Service Discovery

- ***How containers talk to each other****?*
- *Application In Single Ec2*
- *Application in Separate Ec2*

- *Simple Solution*
- *Docker switch HOST*
- *Aws Private Hosted Zone*

Monitoring, Alerting & Logging

- ***Monitoring***
- *Ec2 AWS Cloudwatch Agent /Grafana/Zabbix*

(ASG Configure Cloudwatch Policies)

- ***Alerting***
- *Aws SNS /Grafana/Zabbix*

- ***Logging***
- *Docker Cloudwatch Agent*

Application Scaling

- *ASG Configured with CloudWatch Policies*

- *Eg: Step Scaling*
- *At CPU >60% Add 1 ec2*
- *At CPU <60% Destroy 1 ec2*

- *At Mem >70% Add 1 ec2*
- *At Mem <70% Destroy 1 ec2*

Application Load Balancing

- *External Load Balancing via ALB*

- *Internal Load Balancing via Nginx*

(Docker-Compose allows to RUN multiple containers of application in Single EC2 Machine)