Voting Application Deployment

Pre-requisites:

* AWS CLI
* Terraform
* Storage Account (Tf state management)
* Git
* Docker Compose
* AWS programmatic access & keys

Steps:

* Configure AWS CLI using “aws configure”command
* Initialize terraform configuration using “terraform init”command
* Validate code using “terraform validate” command
* Apply the terraform code using “terraform apply –auto-approve” command
* Resources getting created:
  + VPC
  + Subnet
  + IGW
  + Route Tables
  + Route table association
  + EIP
  + TLS key
  + Public/Private Key
  + Security Groups
  + Inbound/Outbound Rules
  + EC2
* Get the private key using “terraform output -raw private\_key”
* Store pvt key in voting.pem file. Change the permissions to **READ ONLY** using “chmod 400 voting.pem”
* SSH into the server using “ssh -i voting.pem ubuntu@IP”
* Clone the github repo using “git clone https://github.com/dockersamples/example-voting-app.git”
* Install Docker and Docker compose.

# Add Docker's official GPG key:

sudo apt-get update

sudo apt-get install ca-certificates curl

sudo install -m 0755 -d /etc/apt/keyrings

sudo curl -fsSL https://download.docker.com/linux/ubuntu/gpg -o /etc/apt/keyrings/docker.asc

sudo chmod a+r /etc/apt/keyrings/docker.asc

# Add the repository to Apt sources:

echo \

"deb [arch=$(dpkg --print-architecture) signed-by=/etc/apt/keyrings/docker.asc] https://download.docker.com/linux/ubuntu \

$(. /etc/os-release && echo "$VERSION\_CODENAME") stable" | \

sudo tee /etc/apt/sources.list.d/docker.list > /dev/null

sudo apt-get update

sudo apt-get install docker-ce docker-ce-cli containerd.io docker-buildx-plugin docker-compose-plugin -y

* Change dir to **example-voting-app** repo and run command to “docker compose up -d”.This will run the docker application in background

Approaches

* Terraform
  + Open Source
  + Uses HCL (Hashicorp Language)
  + Can be used Multi-Cloud/Hybrid Cloud, Cross Region deployment
  + Reusable modules
  + Remote State management for tracking deployed resources
* AWS CloudFormation:
  + AWS native service
  + Uses JSON or YAML templates.
  + Only for AWS relative resources
  + Paid Service (Operations > 30s )
* AWS CLI:
  + AWS specific
  + No State management