Deploying An application in Amazon ECS with EC2

Resources Covered:

EC2| Docker | ECR | ECS | FARGATE | Application Load Balancer

Flow of Execution:

1. Launch EC2 Instance
2. Install Docker
3. Build docker Image
4. Creating ECR
5. Login into ECR
6. Tag existing image As AWS ECR repo
7. Push an Image to ECR
8. Creating Application Load Balancer
9. Create Task Definition
10. Create AWS ECS Cluster
11. Create Service
12. Validation

Steps:

1. Create an VPC with 3 availability zones make sure its only public
2. Make sure whether route tables and subnets are created along with igw – attached
3. Launch an EC2 Instance
   - Name = ec2-server
   - Create a key pair in the name of vockey
   - Auto assign public ip - enable
   - Create security group – name ec2-server-SG
   - Add SG rules – HTTP And anywhere in IP
   - Launch Instance

4.open GitBash/Amazon connect to install Docker image

- Copy the public ip address of the launched ec2 instance and try to connect to the instance in git ssh -i keyname.pem ec2-user@ip address or use the Amazon connect
- Follow the commands
  - sudo su
  - yum update -y
  - curl -fsSL https://get.docker.com -o get-docker.sh
  - yum install docker -y
  - docker –version
  - systemctl start docker
  - systemctl status docker

5.Building a Docker image and containerizing web Application

- mkdir project
- cd project
- vim Dockerfile
- copy the contents in the file and paste it in the terminal- link for code –
- to save and exit - :wq!
- vim index.html
- copy the contents in the file and paste it in the terminal- link for code –
- to save and exit - :wq!
- ls
- docker build -t project-image .

6. Create a repository in ECR & pushing the docker Image ECR launch

- Go to the console and type ECR
- Let the visibility setting be private as default
- Give a name for repository – myecr-repo
- Scroll down and Create Repository
- After creating a repository on top right corner u can see the push commands click on that
- You can see the commands to push a docker image in the ECR but u must have a role assigned to it

7. Creating a Role to access other services

- Go to the console and search of IAM
- Create a Role
- Trusted entity type must be – AWS service as default
- Use case must me – EC2 and click next
- Permission polices - AdministratorAccess and click next
- Role name – ecr-role1 and click role

8.Create Access Key in AWS

- Go to console and search for IAM
- Click on Users
- Go to Security Creditials tab and scroll down
- Create access Keys
- Click on CLI and Check the box and click next
- No need Description click Create and Done
- Save the access key and secret access key it is used for future, u can also download the csv.file

9. AWS Configuration

- Go to git bash and type the commands
- aws configure
- access key – paste it
- secret access key – paste it
- default region name – us-east-1
- output format – press enter
- for verifying open file explorer in you pc and select /users/admin/.aws/you can view the config open with notepad

10. Login to ECR from AWS CLI

- As we saw earlier push commands in the created ecr-repo
- Copy the first command and paste it – check login succeeded
- docker images – view the images we stored earlier
- copy the third command and paste it and change the file name as you can view in the prompt
- docker images – review as we pushed the image in the repo
- Copy the fourth command and paste it to push the image in to the ECR repo
- Verify In ECR > Repositories > Images > Latest image has been pushed

11. Create an Application Load Balancer In AWS

- Go to console - EC2
- Left side click on Load Balancer and Create LB
- Application Load Balancer and Create
- Name – project-alb
- Default VPC
- Select AZs – us-east-1a, us-east-1b, us-east-1c
- SG- along with default SG, the SG we created for EC2 instance also should be added
- Scroll down and create Target group
    - Scroll down target group name – TG
    - and in the health checks, click on advance health checks change the healthy threshold - 2
    - click next and create target group
- refresh In the target group section and select the created target group
- scroll down and create load balancer
- verify the created load balancers

12.Create Task Definition , Cluster and Service in Amazon ECS

- Go to the console type ECR
- Left side click on task definitions and click on create new task definition
- Family Name- ecr-task-def
- Container 1 – name- ecs-container1 and for image URI go to ECR repository and find the URI copy that paste it
- Scroll down and next
- App environment – default
- Task execution role –
    - Go to IAM
    - Left side roles and create a new role
    - All default and in the use case – type ECS – and check Elastic container service task
    - Permission policies – amazonecstaskexecutionrolepolicy and click on next
    - Role name – ECSTaskExecution-Role and click on create role
- Now refresh the task definition role and select the created role name - ECSTaskExecution-Role
- Scroll down and click next and create

13.Creating cluster go to ECR

- Left side go to clusters and create cluster
- Cluster name- ecs-cluster and scroll down
- Subnets only three is enough and remove the remaining and click on create

- Verify the created cluster in the clusters and in the services tab create a service
- Scroll down and in the task definition select the created task definition family
- Revision- 1(LATEST)
- Service name – ecs-service
- Desired task – 2
- Scroll down and in the network setting subnets must be only us-east-1a, us-east-1b, us-east-1c
- SG – create a new SG
    - Sg name – my-ecs-SG
    - Description - my-ecs-SG
    - Inbound rules – http > source -anywhere
- Load balancer type – choose ALB and choose the existing load balancer – project-alb
- Listener – if default doesn't work 80:HTTP use this
- Target group name – service-TG and Evaluation Order – 1 and path /*
- Scroll down and create
- Go to clusters to verify the clusters that is created

14.Validation

Go to ECS > Clusters > ecs-cluster > Services > ecs-service > Networking tab> copy the DNS name and paste it in the browser