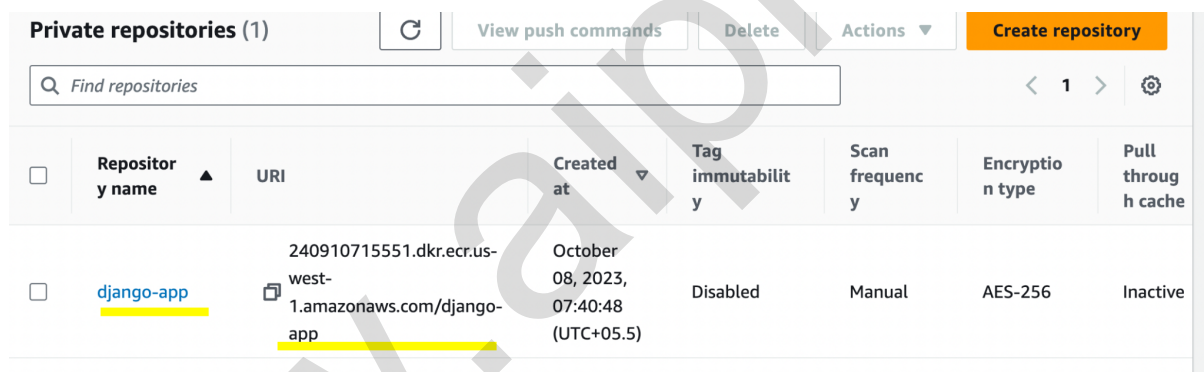


AWS INFRA CREATION WITH PYTHON DJANGO APP DEPLOYMENT [LOCAL]

Step1 - Clone the repository and set the aws configure in local terminal

https://github.com/praveen1994dec/Terraform_withcontainers.git

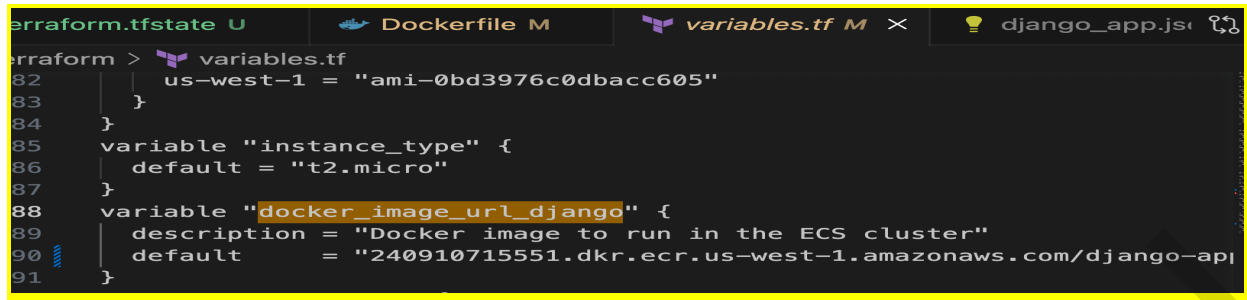
Step2 - Create a repo in aws ECR with name - > django-app



The screenshot shows the AWS ECR Private repositories interface. At the top, there's a header with 'Private repositories (1)', a refresh button, and buttons for 'View push commands', 'Delete', 'Actions', and 'Create repository'. Below the header is a search bar labeled 'Find repositories'. The main content is a table with columns: Repository name, URI, Created at, Tag immutability, Scan frequency, Encryption type, and Pull through cache. One repository is listed: 'django-app' with URI '240910715551.dkr.ecr.us-west-1.amazonaws.com/django-app', created on 'October 08, 2023, 07:40:48 (UTC+05.5)', with 'Disabled' tag immutability, 'Manual' scan frequency, 'AES-256' encryption type, and 'Inactive' pull through cache.

	Repository name	URI	Created at	Tag immutability	Scan frequency	Encryption type	Pull through cache
<input type="checkbox"/>	django-app	240910715551.dkr.ecr.us-west-1.amazonaws.com/django-app	October 08, 2023, 07:40:48 (UTC+05.5)	Disabled	Manual	AES-256	Inactive

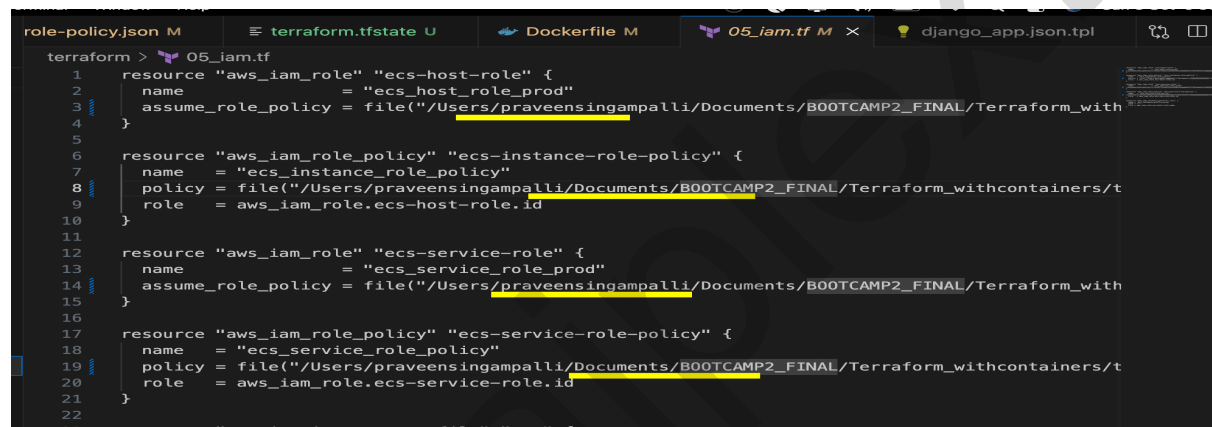
Step3 - Once the repo is created copy the repo url and add it in the `variables.tf` file in place of variable `docker_image_url_django`



```

erraform.tfstate U  Dockerfile M  variables.tf M  x  django_app.js
erraform > variables.tf
82     us-west-1 = "ami-0bd3976c0dbacc605"
83   }
84 }
85 variable "instance_type" {
86   default = "t2.micro"
87 }
88 variable "docker_image_url_django" {
89   description = "Docker image to run in the ECS cluster"
90   default     = "240910715551.dkr.ecr.us-west-1.amazonaws.com/django-api
91 }
  
```

Step4 - Change the policy file paths in `iam.tf` and `variables.tf` file



```

role-policy.json M  terraform.tfstate U  Dockerfile M  05_iam.tf M  x  django_app.json.tpl
terraform > 05_iam.tf
1  resource "aws_iam_role" "ecs-host-role" {
2    name = "ecs_host_role_prod"
3    assume_role_policy = file("/Users/praveensingampalli/Documents/BOOTCAMP2_FINAL/Terraform_with
4  }
5
6  resource "aws_iam_role_policy" "ecs-instance-role-policy" {
7    name = "ecs_instance_role_policy"
8    policy = file("/Users/praveensingampalli/Documents/BOOTCAMP2_FINAL/Terraform_withcontainers/t
9    role = aws_iam_role.ecs-host-role.id
10 }
11
12 resource "aws_iam_role" "ecs-service-role" {
13   name = "ecs_service_role_prod"
14   assume_role_policy = file("/Users/praveensingampalli/Documents/BOOTCAMP2_FINAL/Terraform_with
15 }
16
17 resource "aws_iam_role_policy" "ecs-service-role-policy" {
18   name = "ecs_service_role_policy"
19   policy = file("/Users/praveensingampalli/Documents/BOOTCAMP2_FINAL/Terraform_withcontainers/t
20   role = aws_iam_role.ecs-service-role.id
21 }
22
23 resource "aws_iam_instance_profile" "ecs" {
  
```

Step5 - Login into ECR by hitting the below command in terminal local

```

aws ecr get-login-password --region
us-west-1 | docker login --username AWS
--password-stdin <YOUR ECR REPO URL>
  
```

Step6 - Go to app folder
cd app/

Step7 - Build the docker image

docker build --platform=linux/amd64 -t <URL ECR REPO URL>:latest .

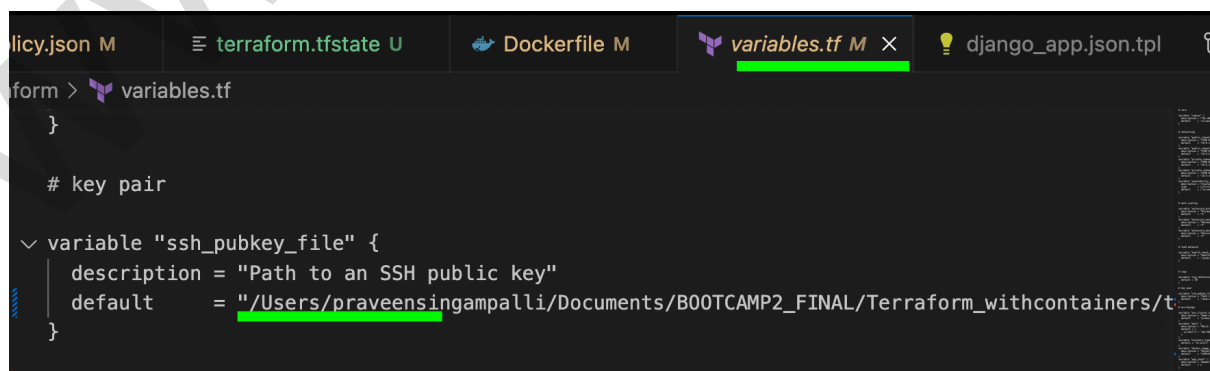
Step8 - Push the image to ECR

docker push <URL ECR REPO URL>:latest

Step9 - Go to **terraform folder and hit this below command to create the key pair**

ssh-keygen -f california-region-key-pair

Step10 - Add the above file key file path in **variable.tf**



```
form > variables.tf
}

# key pair

variable "ssh_pubkey_file" {
  description = "Path to an SSH public key"
  default     = "/Users/praveensingampalli/Documents/BOOTCAMP2_FINAL/Terraform_withcontainers/t"
}
```

Step 11 - Hit the below commands in terraform folder

```
terraform init
```

```
terraform plan -out terraform.out
```

```
terraform apply "terraform.out"
```

Step12 - Install Python BOTO3

```
pip install boto3 click
```

Step13 - Export the AWS access/Secret key and region

```
export AWS_ACCESS_KEY_ID=""
```

```
export AWS_SECRET_ACCESS_KEY=""
```

```
export AWS_DEFAULT_REGION="us-west-1"
```

Step14 - Go to deploy folder

```
python3 update-ecs.py --cluster=production-cluster  
--service=production-service
```

Step15 - Validate the ECS service and there should be 0 Tasks pending in the dashboard

Go to cloudwatch/VPC and check the data and logs stream

Step16 - > Go to EC2-> Loadbanacer -> Copy the DNS name -> Hit the below url in the browser

<DNS_NAME>/ping/

Step17 - Hit the command and delete the architecture

terraform destroy