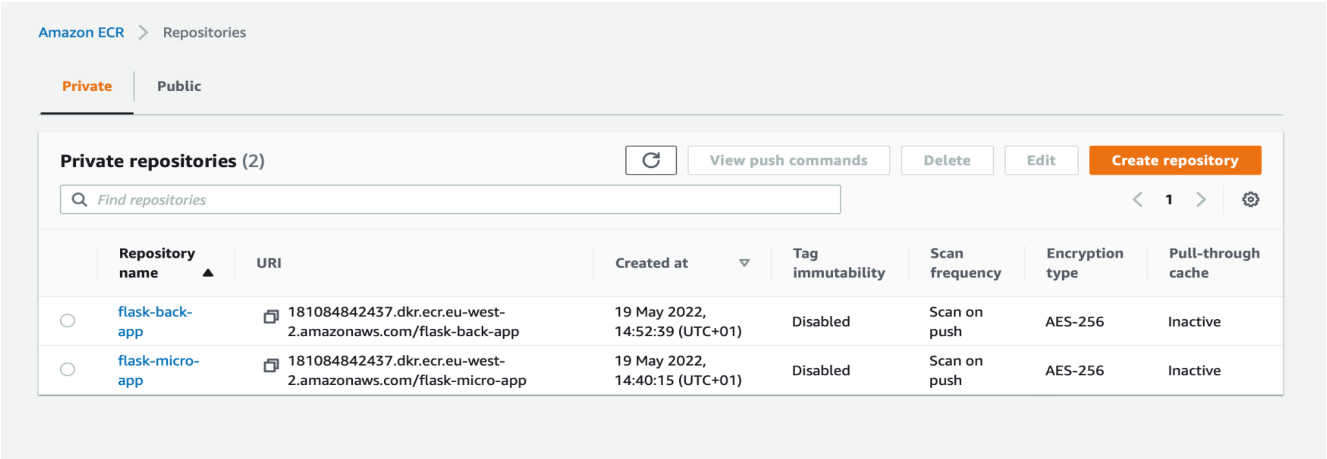
Screenshots

```
j@Js-MacBook-Pro ~ % aws configure
AWS Access Key ID [****************QNUX]:
AWS Secret Access Key [****************mikG]:
Default region name [eu-west-2]:
Default output format [json]:
```

```
j@Js-MacBook-Pro frontend % docker build -t flask-micro-app .
[+] Building 13.6s (11/11) FINISHED
 => [internal] load build definition from Dockerfile                    0.0s
 => => transferring dockerfile: 292B                                    0.0s
 => [internal] load .dockerignore                                       0.0s
 => => transferring context: 2B                                         0.0s
 => [internal] load metadata for docker.io/library/python:3.8-slim-buster  7.9s
 => [auth] library/python:pull token for registry-1.docker.io           0.0s
```

```
j@Js-MacBook-Pro backend % docker push 181084842437.dkr.ecr.eu-west-2.amazonaws.com/flask-back-app:latest
The push refers to repository [181084842437.dkr.ecr.eu-west-2.amazonaws.com/flask-back-app]
54a58bb976dc: Pushed
9e880bd3628c: Pushed
899c40b6f316: Pushed
048bd28b82b2: Pushed
3ed0c4ea6f65: Pushed
82ee592ec424: Pushed
```



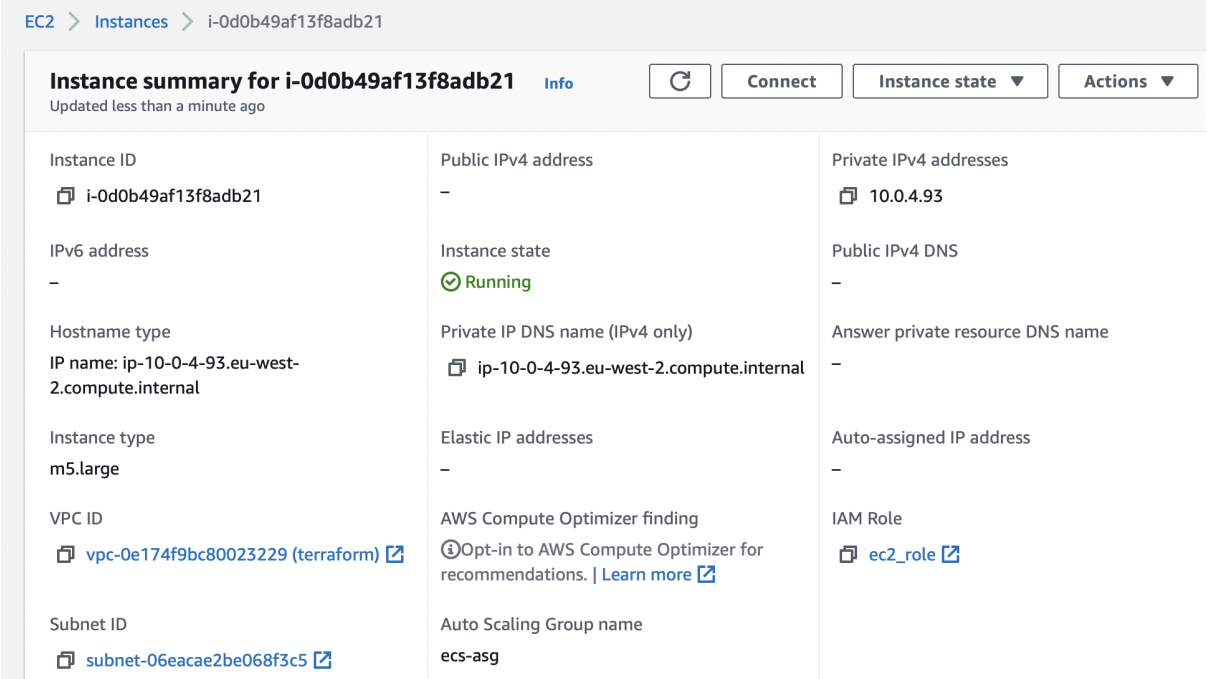
```
j@Js-MacBook-Pro terraform % terraform init

Initializing the backend...

Initializing provider plugins...
- Reusing previous version of hashicorp/aws from the dependency lock file
- Using previously-installed hashicorp/aws v4.14.0

Terraform has been successfully initialized!

You may now begin working with Terraform. Try running "terraform plan" to see
any changes that are required for your infrastructure. All Terraform commands
should now work.

If you ever set or change modules or backend configuration for Terraform,
rerun this command to reinitialize your working directory. If you forget, other
commands will detect it and remind you to do so if necessary.
j@Js-MacBook-Pro terraform %
```

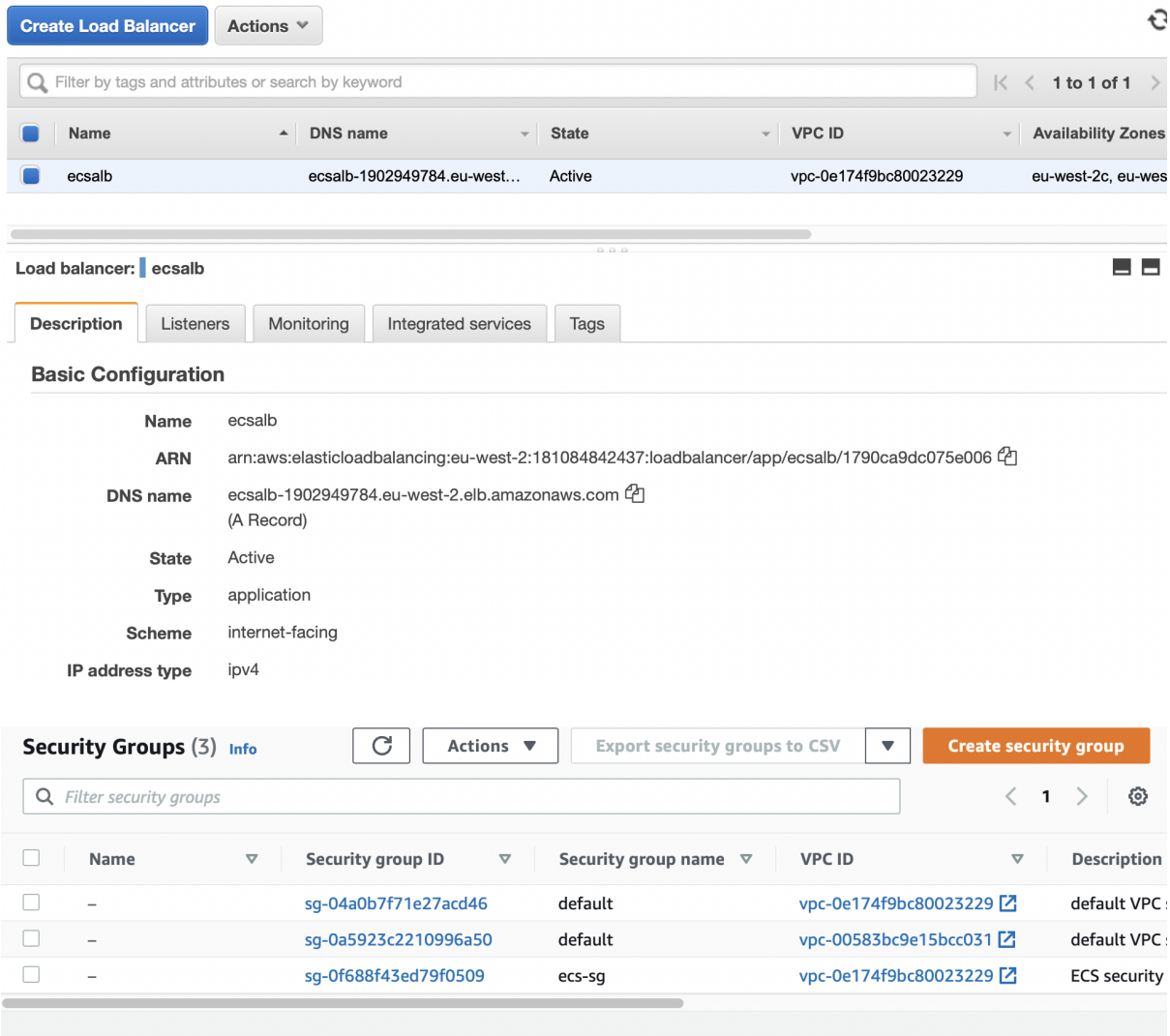
```
j@Js-MacBook-Pro terraform % terraform plan
var.ecs_image_url
  The desired ECR image URL.

  Enter a value: 181084842437.dkr.ecr.eu-west-2.amazonaws.com/flask-micro-app
```

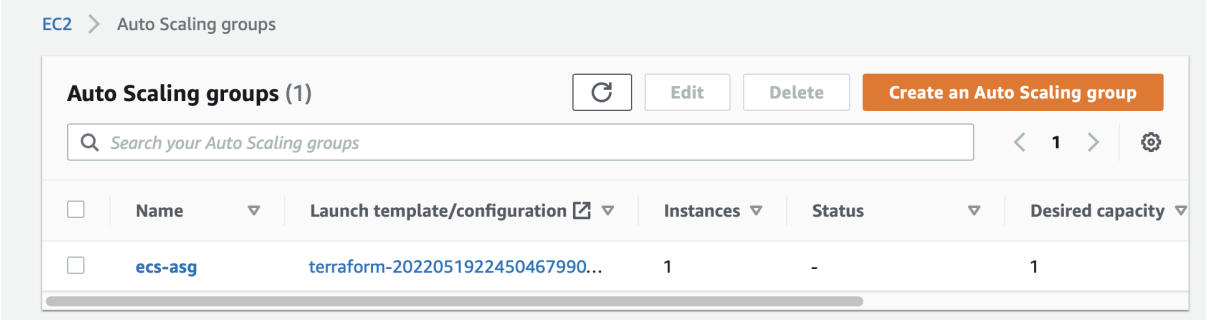
```
j@Js-MacBook-Pro terraform % terraform apply
var.ecs_image_url
  The desired ECR image URL.

  Enter a value:
```

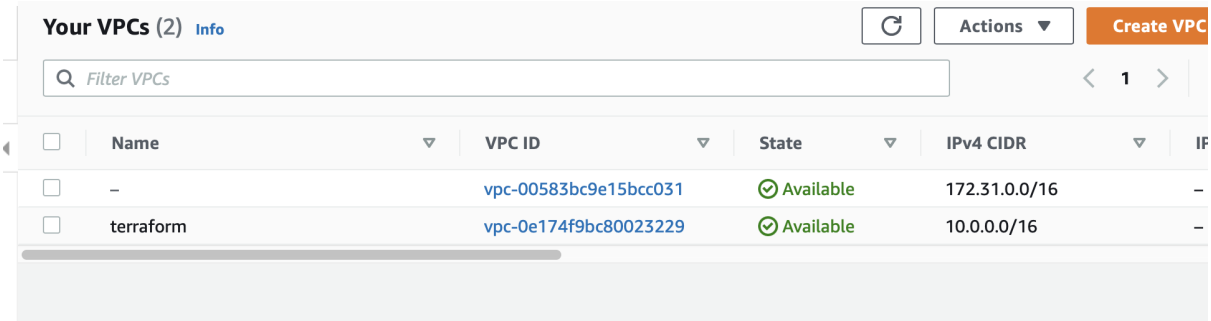
```
j@Js-MacBook-Pro terraform % terraform apply
var.ecs_image_url
  The desired ECR image URL.

  Enter a value: 181084842437.dkr.ecr.eu-west-2.amazonaws.com/flask-micro-app
```

```
salb/1790ca9dc075e006]
aws_lb_listener.alb_listener: Creating...
aws_lb_listener.alb_listener: Creation complete after 0s [id=arn:aws:elasticloadbalancing:eu-west-2:181084842437:listener/app/ecsalb/1790ca9dc075e006/c47cbe27c697cfec]
aws_ecs_service.service: Creating...
aws_ecs_service.service: Creation complete after 1s [id=arn:aws:ecs:eu-west-2:181084842437:service/ecs_cluster/flask-docker]
aws_appautoscaling_target.ecs_service_scaling_target: Creating...
aws_appautoscaling_target.ecs_service_scaling_target: Creation complete after 0s [id=service/ecs_cluster/flask-docker]
aws_appautoscaling_policy.ecs_service_memory_scale_out_policy: Creating...
aws_appautoscaling_policy.ecs_service_cpu_scale_out_policy: Creating...
aws_appautoscaling_policy.ecs_service_memory_scale_out_policy: Creation complete after 1s [id=memory-target-tracking-scaling-policy]
aws_appautoscaling_policy.ecs_service_cpu_scale_out_policy: Creation complete after 1s [id=cpu-target-tracking-scaling-policy]
aws_cloudwatch_metric_alarm.ecs_service_cpu_scale_out_alarm: Creating...
aws_cloudwatch_metric_alarm.ecs_service_cpu_scale_out_alarm: Creation complete after 0s [id=CPU utilization greater than 50%]

Apply complete! Resources: 47 added, 0 changed, 0 destroyed.

Outputs:

alb_dns_name = "ecsalb-1902949784.eu-west-2.elb.amazonaws.com"
j@Js-MacBook-Pro terraform %
```


EC2 > Instances > i-0d0b49af13f8adb21
Instance summary for i-0d0b49af13f8adb21 Info
Updated less than a minute ago
Connect
Instance state
Actions
Instance ID
i-0d0b49af13f8adb21
Public IPv4 address
–
Private IPv4 addresses
10.0.4.93
IPv6 address
–
Instance state
Running
Public IPv4 DNS
–
Hostname type
IP name: ip-10-0-4-93.eu-west-2.compute.internal
Private IP DNS name (IPv4 only)
ip-10-0-4-93.eu-west-2.compute.internal
Answer private resource DNS name
–
Instance type
m5.large
Elastic IP addresses
–
Auto-assigned IP address
–
VPC ID
vpc-0e174f9bc80023229 (terraform)
AWS Compute Optimizer finding
Opt-in to AWS Compute Optimizer for recommendations. | Learn more
IAM Role
ec2_role
Subnet ID
subnet-06eacae2be068f3c5
Auto Scaling Group name
ecs-asg


Create Load Balancer
Actions
Filter by tags and attributes or search by keyword
1 to 1 of 1
Name
DNS name
State
VPC ID
Availability Zones
ecsalb
ecsalb-1902949784.eu-west...
Active
vpc-0e174f9bc80023229
eu-west-2c, eu-wes
Load balancer: ecsalb
Description
Listeners
Monitoring
Integrated services
Tags
Basic Configuration
Name
ecsalb
ARN
arn:aws:elasticloadbalancing:eu-west-2:181084842437:loadbalancer/app/ecsalb/1790ca9dc075e006
DNS name
ecsalb-1902949784.eu-west-2.elb.amazonaws.com
(A Record)
State
Active
Type
application
Scheme
internet-facing
IP address type
ipv4
Security Groups (3) Info
Actions
Export security groups to CSV
Create security group
Filter security groups
1
Name
Security group ID
Security group name
VPC ID
Description
–
sg-04a0b7f71e27acd46
default
vpc-0e174f9bc80023229
default VPC
–
sg-0a5923c2210996a50
default
vpc-00583bc9e15bcc031
default VPC
–
sg-0f688f43ed79f0509
ecs-sg
vpc-0e174f9bc80023229
ECS security





DynamoDB Table: