

Terraform Init Log :

```
mahesh.singh@del1-lmc-n82481 capstone-aws % terraform init
Initializing the backend...
Initializing provider plugins...
- Reusing previous version of hashicorp/aws from the dependency lock file
- Using previously-installed hashicorp/aws v6.21.0
```

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.

Terraform plan Log :

```
mahesh.singh@del1-lmc-n82481 capstone-aws % terraform plan
data.aws_partition.current: Reading...
data.aws_caller_identity.current: Reading...
data.aws_availability_zones.available: Reading...
data.aws_partition.current: Read complete after 0s [id=aws]
data.aws_caller_identity.current: Read complete after 0s [id=550101108440]
data.aws_availability_zones.available: Read complete after 1s [id=us-east-1]
```

Terraform used the selected providers to generate the following execution plan. Resource actions are indicated with the following symbols:

+ create

Terraform will perform the following actions:

```
# aws_cloudwatch_log_group.ecs_logs will be created
+ resource "aws_cloudwatch_log_group" "ecs_logs" {
  + arn          = (known after apply)
  + id           = (known after apply)
  + log_group_class = (known after apply)
  + name         = "/ecs/flask-service-dev"
  + name_prefix   = (known after apply)
  + region       = "us-east-1"
  + retention_in_days = 7
  + skip_destroy   = false
  + tags_all      = (known after apply)
}
```

```
# aws_ecs_cluster.main will be created
+ resource "aws_ecs_cluster" "main" {
```

```
+ arn      = (known after apply)
+ id       = (known after apply)
+ name     = "flask-service-dev"
+ region   = "us-east-1"
+ tags_all = (known after apply)
```

```
+ setting (known after apply)
}
```

aws_ecs_service.service will be created

```
+ resource "aws_ecs_service" "service" {
  + arn                        = (known after apply)
  + availability_zone_rebalancing = (known after apply)
  + cluster                  = (known after apply)
  + deployment_maximum_percent = 200
  + deployment_minimum_healthy_percent = 100
  + desired_count             = 2
  + enable_ecs_managed_tags   = false
  + enable_execute_command    = false
  + health_check_grace_period_seconds = 120
  + iam_role                  = (known after apply)
  + id                       = (known after apply)
  + launch_type              = "FARGATE"
  + name                     = "flask-service-service"
  + platform_version         = (known after apply)
  + region                   = "us-east-1"
  + scheduling_strategy       = "REPLICA"
  + tags_all                 = (known after apply)
  + task_definition          = (known after apply)
  + triggers                 = (known after apply)
  + wait_for_steady_state    = false
```

```
+ deployment_configuration (known after apply)
```

```
+ load_balancer {
  + container_name = "flask"
  + container_port = 80
  + target_group_arn = (known after apply)
  # (1 unchanged attribute hidden)
}
```

```
+ network_configuration {
  + assign_public_ip = false
  + security_groups = (known after apply)
  + subnets       = (known after apply)
}
}
```

aws_ecs_task_definition.task will be created

```
+ resource "aws_ecs_task_definition" "task" {
```

```

+ arn = (known after apply)
+ arn_without_revision = (known after apply)
+ container_definitions = jsonencode(
  [
    + {
      + image = "550101108440.dkr.ecr.us-east-1.amazonaws.com/
flask-app:latest"
      + logConfiguration = {
        + logDriver = "awslogs"
        + options = {
          + awslogs-group = "/ecs/flask-service-dev"
          + awslogs-region = "us-east-1"
          + awslogs-stream-prefix = "ecs"
        }
      }
      + name = "flask"
      + portMappings = [
        + {
          + containerPort = 80
          + hostPort = 80
          + protocol = "tcp"
        },
      ]
    },
  ]
)
+ cpu = "256"
+ enable_fault_injection = (known after apply)
+ execution_role_arn = (known after apply)
+ family = "flask-task"
+ id = (known after apply)
+ memory = "512"
+ network_mode = "awsvpc"
+ region = "us-east-1"
+ requires_compatibilities = [
  + "FARGATE",
]
+ revision = (known after apply)
+ skip_destroy = false
+ tags_all = (known after apply)
+ task_role_arn = (known after apply)
+ track_latest = false

+ runtime_platform {
  + cpu_architecture = "ARM64"
  + operating_system_family = "LINUX"
}
}

```

aws_eip.nat will be created

```

+ resource "aws_eip" "nat" {
  + allocation_id    = (known after apply)
  + arn              = (known after apply)
  + association_id   = (known after apply)
  + carrier_ip       = (known after apply)
  + customer_owned_ip = (known after apply)
  + domain           = (known after apply)
  + id               = (known after apply)
  + instance         = (known after apply)
  + ipam_pool_id     = (known after apply)
  + network_border_group = (known after apply)
  + network_interface = (known after apply)
  + private_dns      = (known after apply)
  + private_ip       = (known after apply)
  + ptr_record       = (known after apply)
  + public_dns       = (known after apply)
  + public_ip        = (known after apply)
  + public_ipv4_pool = (known after apply)
  + region           = "us-east-1"
  + tags_all         = (known after apply)
}

# aws_iam_role.ecs_task_execution will be created
+ resource "aws_iam_role" "ecs_task_execution" {
  + arn              = (known after apply)
  + assume_role_policy = jsonencode(
    {
      + Statement = [
        + {
          + Action   = "sts:AssumeRole"
          + Effect    = "Allow"
          + Principal = {
            + Service = "ecs-tasks.amazonaws.com"
          }
        },
      ]
      + Version = "2012-10-17"
    }
  )
  + create_date      = (known after apply)
  + force_detach_policies = false
  + id               = (known after apply)
  + managed_policy_arns = (known after apply)
  + max_session_duration = 3600
  + name             = "ecsTaskExecutionRole"
  + name_prefix      = (known after apply)
  + path             = "/"
  + tags_all         = (known after apply)
  + unique_id        = (known after apply)

```

```

+ inline_policy (known after apply)
}

# aws_iam_role_policy_attachment.ecs_task_execution_policy will be
created
+ resource "aws_iam_role_policy_attachment" "ecs_task_execution_policy" {
  + id      = (known after apply)
  + policy_arn = "arn:aws:iam::aws:policy/service-role/
AmazonECSTaskExecutionRolePolicy"
  + role     = "ecsTaskExecutionRole"
}

# aws_internet_gateway.igw will be created
+ resource "aws_internet_gateway" "igw" {
  + arn      = (known after apply)
  + id       = (known after apply)
  + owner_id = (known after apply)
  + region   = "us-east-1"
  + tags_all = (known after apply)
  + vpc_id   = (known after apply)
}

# aws_lb.app_alb will be created
+ resource "aws_lb" "app_alb" {
  + arn                        = (known after apply)
  + arn_suffix                = (known after apply)
  + client_keep_alive         = 3600
  + desync_mitigation_mode    = "defensive"
  + dns_name                  = (known after apply)
  + drop_invalid_header_fields = false
  + enable_deletion_protection = false
  + enable_http2              = true
  + enable_tls_version_and_cipher_suite_headers = false
  + enable_waf_fail_open      = false
  + enable_xff_client_port     = false
  + enable_zonal_shift        = false
  + enforce_security_group_inbound_rules_on_private_link_traffic = (known
after apply)
  + id                        = (known after apply)
  + idle_timeout              = 60
  + internal                  = (known after apply)
  + ip_address_type           = (known after apply)
  + load_balancer_type        = "application"
  + name                      = "flask-alb"
  + name_prefix               = (known after apply)
  + preserve_host_header      = false
  + region                   = "us-east-1"
  + secondary_ips_auto_assigned_per_subnet = (known after
apply)
  + security_groups           = (known after apply)

```

```

+ subnets                                = (known after apply)
+ tags_all                                = (known after apply)
+ vpc_id                                  = (known after apply)
+ xff_header_processing_mode                = "append"
+ zone_id                                  = (known after apply)

+ subnet_mapping (known after apply)
}

# aws_lb_listener.http will be created
+ resource "aws_lb_listener" "http" {
  + arn                                = (known after apply)
  + id                                = (known after apply)
  + load_balancer_arn                  = (known after apply)
  + port                                = 80
  + protocol                            = "HTTP"
  + region                              = "us-east-1"
  + routing_http_request_x_amzn_mtls_clientcert_header_name =
(known after apply)
  + routing_http_request_x_amzn_mtls_clientcert_issuer_header_name
= (known after apply)
  + routing_http_request_x_amzn_mtls_clientcert_leaf_header_name =
(known after apply)
  +
routing_http_request_x_amzn_mtls_clientcert_serial_number_header_name =
(known after apply)
  + routing_http_request_x_amzn_mtls_clientcert_subject_header_name
= (known after apply)
  + routing_http_request_x_amzn_mtls_clientcert_validity_header_name
= (known after apply)
  + routing_http_request_x_amzn_tls_cipher_suite_header_name      =
(known after apply)
  + routing_http_request_x_amzn_tls_version_header_name          =
(known after apply)
  + routing_http_response_access_control_allow_credentials_header_value
= (known after apply)
  + routing_http_response_access_control_allow_headers_header_value
= (known after apply)
  + routing_http_response_access_control_allow_methods_header_value
= (known after apply)
  + routing_http_response_access_control_allow_origin_header_value =
(known after apply)
  + routing_http_response_access_control_expose_headers_header_value
= (known after apply)
  + routing_http_response_access_control_max_age_header_value      =
(known after apply)
  + routing_http_response_content_security_policy_header_value      =
(known after apply)
  + routing_http_response_server_enabled                            = (known after
apply)

```

```

+ routing_http_response_strict_transport_security_header_value      =
(known after apply)
+ routing_http_response_x_content_type_options_header_value        =
(known after apply)
+ routing_http_response_x_frame_options_header_value                =
(known after apply)
+ ssl_policy                                                         = (known after apply)
+ tags_all                                                           = (known after apply)
+ tcp_idle_timeout_seconds                                          = (known after
apply)

```

```

+ default_action {
+ order          = (known after apply)
+ target_group_arn = (known after apply)
+ type           = "forward"
}

```

```

+ mutual_authentication (known after apply)
}

```

aws_lb_target_group.app_tg will be created

```

+ resource "aws_lb_target_group" "app_tg" {
+ arn                = (known after apply)
+ arn_suffix         = (known after apply)
+ connection_termination = (known after apply)
+ deregistration_delay = "300"
+ id                 = (known after apply)
+ ip_address_type     = (known after apply)
+ lambda_multi_value_headers_enabled = false
+ load_balancer_arns  = (known after apply)
+ load_balancing_algorithm_type = (known after apply)
+ load_balancing_anomaly_mitigation = (known after apply)
+ load_balancing_cross_zone_enabled = (known after apply)
+ name               = "flask-service-tg"
+ name_prefix        = (known after apply)
+ port               = 80
+ preserve_client_ip  = (known after apply)
+ protocol            = "HTTP"
+ protocol_version    = (known after apply)
+ proxy_protocol_v2    = false
+ region             = "us-east-1"
+ slow_start          = 0
+ tags_all            = (known after apply)
+ target_type         = "ip"
+ vpc_id              = (known after apply)

```

```

+ health_check {
+ enabled          = true
+ healthy_threshold = 3
+ interval         = 30

```

```

+ matcher          = "200-399"
+ path             = "/"
+ port             = "traffic-port"
+ protocol         = "HTTP"
+ timeout          = (known after apply)
+ unhealthy_threshold = 3
}

+ stickiness (known after apply)

+ target_failover (known after apply)

+ target_group_health (known after apply)

+ target_health_state (known after apply)
}

# aws_nat_gateway.nat will be created
+ resource "aws_nat_gateway" "nat" {
+ allocation_id      = (known after apply)
+ association_id     = (known after apply)
+ connectivity_type  = "public"
+ id                 = (known after apply)
+ network_interface_id = (known after apply)
+ private_ip         = (known after apply)
+ public_ip          = (known after apply)
+ region             = "us-east-1"
+ secondary_allocation_ids = (known after apply)
+ secondary_private_ip_address_count = (known after apply)
+ secondary_private_ip_addresses = (known after apply)
+ subnet_id          = (known after apply)
+ tags_all           = (known after apply)
}

# aws_route.private_nat_access will be created
+ resource "aws_route" "private_nat_access" {
+ destination_cidr_block = "0.0.0.0/0"
+ id                     = (known after apply)
+ instance_id            = (known after apply)
+ instance_owner_id      = (known after apply)
+ nat_gateway_id         = (known after apply)
+ network_interface_id   = (known after apply)
+ origin                 = (known after apply)
+ region                 = "us-east-1"
+ route_table_id         = (known after apply)
+ state                  = (known after apply)
}

# aws_route.public_internet_access will be created
+ resource "aws_route" "public_internet_access" {

```



```
+ destination_cidr_block = "0.0.0.0/0"
+ gateway_id             = (known after apply)
+ id                     = (known after apply)
+ instance_id            = (known after apply)
+ instance_owner_id      = (known after apply)
+ network_interface_id   = (known after apply)
+ origin                 = (known after apply)
+ region                 = "us-east-1"
+ route_table_id         = (known after apply)
+ state                  = (known after apply)
}
```

aws_route_table.private will be created

```
+ resource "aws_route_table" "private" {
+   arn          = (known after apply)
+   id           = (known after apply)
+   owner_id      = (known after apply)
+   propagating_vgws = (known after apply)
+   region       = "us-east-1"
+   route         = (known after apply)
+   tags_all      = (known after apply)
+   vpc_id        = (known after apply)
}
```

aws_route_table.public will be created

```
+ resource "aws_route_table" "public" {
+   arn          = (known after apply)
+   id           = (known after apply)
+   owner_id      = (known after apply)
+   propagating_vgws = (known after apply)
+   region       = "us-east-1"
+   route         = (known after apply)
+   tags_all      = (known after apply)
+   vpc_id        = (known after apply)
}
```

aws_route_table_association.private[0] will be created

```
+ resource "aws_route_table_association" "private" {
+   id          = (known after apply)
+   region      = "us-east-1"
+   route_table_id = (known after apply)
+   subnet_id    = (known after apply)
}
```

aws_route_table_association.private[1] will be created

```
+ resource "aws_route_table_association" "private" {
+   id          = (known after apply)
+   region      = "us-east-1"
+   route_table_id = (known after apply)
+   subnet_id    = (known after apply)
}
```

```
}
```

```
# aws_route_table_association.public[0] will be created
```

```
+ resource "aws_route_table_association" "public" {  
  + id          = (known after apply)  
  + region      = "us-east-1"  
  + route_table_id = (known after apply)  
  + subnet_id   = (known after apply)  
}
```

```
# aws_route_table_association.public[1] will be created
```

```
+ resource "aws_route_table_association" "public" {  
  + id          = (known after apply)  
  + region      = "us-east-1"  
  + route_table_id = (known after apply)  
  + subnet_id   = (known after apply)  
}
```

```
# aws_security_group.alb_sg will be created
```

```
+ resource "aws_security_group" "alb_sg" {  
  + arn          = (known after apply)  
  + description   = "ALB security group"  
  + egress        = [  
    + {  
      + cidr_blocks = [  
        + "0.0.0.0/0",  
      ]  
      + description = "Allow all outbound"  
      + from_port   = 0  
      + ipv6_cidr_blocks = []  
      + prefix_list_ids = []  
      + protocol     = "-1"  
      + security_groups = []  
      + self         = false  
      + to_port      = 0  
    },  
  ]  
  + id          = (known after apply)  
  + ingress      = [  
    + {  
      + cidr_blocks = [  
        + "0.0.0.0/0",  
      ]  
      + description = "Allow HTTP inbound"  
      + from_port   = 80  
      + ipv6_cidr_blocks = []  
      + prefix_list_ids = []  
      + protocol     = "tcp"  
      + security_groups = []  
      + self         = false  
    },  
  ]  
}
```

```

        + to_port      = 80
    },
]
+ name                = "alb-sg"
+ name_prefix         = (known after apply)
+ owner_id            = (known after apply)
+ region              = "us-east-1"
+ revoke_rules_on_delete = false
+ tags_all            = (known after apply)
+ vpc_id              = (known after apply)
}

```

aws_security_group.ecs_sg will be created

```

+ resource "aws_security_group" "ecs_sg" {
  + arn                = (known after apply)
  + description        = "ECS tasks security group"
  + egress              = [
    + {
      + cidr_blocks      = [
        + "0.0.0.0/0",
      ]
      + description      = "Allow all outbound"
      + from_port        = 0
      + ipv6_cidr_blocks = []
      + prefix_list_ids  = []
      + protocol         = "-1"
      + security_groups  = []
      + self             = false
      + to_port          = 0
    },
  ]
  + id                = (known after apply)
  + ingress            = [
    + {
      + cidr_blocks      = []
      + description      = "Allow ALB to reach ECS tasks on port 80"
      + from_port        = 80
      + ipv6_cidr_blocks = []
      + prefix_list_ids  = []
      + protocol         = "tcp"
      + security_groups  = (known after apply)
      + self             = false
      + to_port          = 80
    },
  ]
  + name                = "ecs-sg"
  + name_prefix         = (known after apply)
  + owner_id            = (known after apply)
  + region              = "us-east-1"
  + revoke_rules_on_delete = false
}

```

```
+ tags_all          = (known after apply)
+ vpc_id            = (known after apply)
}
```

aws_subnet.private[0] will be created

```
+ resource "aws_subnet" "private" {
+   arn                  = (known after apply)
+   assign_ipv6_address_on_creation = false
+   availability_zone     = "us-east-1a"
+   availability_zone_id   = (known after apply)
+   cidr_block            = "10.0.160.0/20"
+   enable_dns64          = false
+   enable_resource_name_dns_a_record_on_launch = false
+   enable_resource_name_dns_aaaa_record_on_launch = false
+   id                   = (known after apply)
+   ipv6_cidr_block_association_id = (known after apply)
+   ipv6_native          = false
+   map_public_ip_on_launch = false
+   owner_id             = (known after apply)
+   private_dns_hostname_type_on_launch = (known after apply)
+   region               = "us-east-1"
+   tags_all             = (known after apply)
+   vpc_id               = (known after apply)
}
```

aws_subnet.private[1] will be created

```
+ resource "aws_subnet" "private" {
+   arn                  = (known after apply)
+   assign_ipv6_address_on_creation = false
+   availability_zone     = "us-east-1b"
+   availability_zone_id   = (known after apply)
+   cidr_block            = "10.0.176.0/20"
+   enable_dns64          = false
+   enable_resource_name_dns_a_record_on_launch = false
+   enable_resource_name_dns_aaaa_record_on_launch = false
+   id                   = (known after apply)
+   ipv6_cidr_block_association_id = (known after apply)
+   ipv6_native          = false
+   map_public_ip_on_launch = false
+   owner_id             = (known after apply)
+   private_dns_hostname_type_on_launch = (known after apply)
+   region               = "us-east-1"
+   tags_all             = (known after apply)
+   vpc_id               = (known after apply)
}
```

aws_subnet.public[0] will be created

```
+ resource "aws_subnet" "public" {
+   arn                  = (known after apply)
+   assign_ipv6_address_on_creation = false
```

```

+ availability_zone           = "us-east-1a"
+ availability_zone_id       = (known after apply)
+ cidr_block                 = "10.0.0.0/20"
+ enable_dns64               = false
+ enable_resource_name_dns_a_record_on_launch = false
+ enable_resource_name_dns_aaaa_record_on_launch = false
+ id                         = (known after apply)
+ ipv6_cidr_block_association_id = (known after apply)
+ ipv6_native                = false
+ map_public_ip_on_launch    = true
+ owner_id                   = (known after apply)
+ private_dns_hostname_type_on_launch = (known after apply)
+ region                     = "us-east-1"
+ tags_all                   = (known after apply)
+ vpc_id                     = (known after apply)
}

```

aws_subnet.public[1] will be created

```

+ resource "aws_subnet" "public" {
+   arn                     = (known after apply)
+   assign_ipv6_address_on_creation = false
+   availability_zone       = "us-east-1b"
+   availability_zone_id    = (known after apply)
+   cidr_block              = "10.0.16.0/20"
+   enable_dns64            = false
+   enable_resource_name_dns_a_record_on_launch = false
+   enable_resource_name_dns_aaaa_record_on_launch = false
+   id                      = (known after apply)
+   ipv6_cidr_block_association_id = (known after apply)
+   ipv6_native             = false
+   map_public_ip_on_launch = true
+   owner_id                = (known after apply)
+   private_dns_hostname_type_on_launch = (known after apply)
+   region                  = "us-east-1"
+   tags_all                = (known after apply)
+   vpc_id                  = (known after apply)
}

```

aws_vpc.main will be created

```

+ resource "aws_vpc" "main" {
+   arn                     = (known after apply)
+   cidr_block              = "10.0.0.0/16"
+   default_network_acl_id  = (known after apply)
+   default_route_table_id = (known after apply)
+   default_security_group_id = (known after apply)
+   dhcp_options_id         = (known after apply)
+   enable_dns_hostnames    = (known after apply)
+   enable_dns_support      = true
+   enable_network_address_usage_metrics = (known after apply)
+   id                      = (known after apply)
}

```

```

+ instance_tenancy      = "default"
+ ipv6_association_id   = (known after apply)
+ ipv6_cidr_block       = (known after apply)
+ ipv6_cidr_block_network_border_group = (known after apply)
+ main_route_table_id   = (known after apply)
+ owner_id              = (known after apply)
+ region                = "us-east-1"
+ tags_all              = (known after apply)
}

```

Plan: 27 to add, 0 to change, 0 to destroy.

Changes to Outputs:

```

+ alb_dns_name    = (known after apply)
+ ecs_cluster_name = "flask-service-dev"
+ running_tasks   = 2

```

Terraform Apply Log:

```

mahesh.singh@del1-lmc-n82481 capstone-aws % terraform apply --
auto-approve
data.aws_availability_zones.available: Reading...
data.aws_caller_identity.current: Reading...
data.aws_partition.current: Reading...
data.aws_partition.current: Read complete after 0s [id=aws]
data.aws_caller_identity.current: Read complete after 1s
[id=550101108440]
data.aws_availability_zones.available: Read complete after 1s
[id=us-east-1]

```

Terraform used the selected providers to generate the following execution plan. Resource actions are indicated with the following symbols:

+ create

Terraform will perform the following actions:

```

# aws_cloudwatch_log_group.ecs_logs will be created
+ resource "aws_cloudwatch_log_group" "ecs_logs" {
+   arn          = (known after apply)
+   id           = (known after apply)
+   log_group_class = (known after apply)

```

```
+ name          = "/ecs/flask-service-dev"
+ name_prefix    = (known after apply)
+ region         = "us-east-1"
+ retention_in_days = 7
+ skip_destroy    = false
+ tags_all       = (known after apply)
}
```

aws_ecs_cluster.main will be created

```
+ resource "aws_ecs_cluster" "main" {
  + arn      = (known after apply)
  + id       = (known after apply)
  + name     = "flask-service-dev"
  + region   = "us-east-1"
  + tags_all = (known after apply)

  + setting (known after apply)
}
```

aws_ecs_service.service will be created

```
+ resource "aws_ecs_service" "service" {
  + arn                  = (known after apply)
  + availability_zone_rebalancing = (known after apply)
  + cluster              = (known after apply)
  + deployment_maximum_percent    = 200
  + deployment_minimum_healthy_percent = 100
  + desired_count                = 2
  + enable_ecs_managed_tags      = false
  + enable_execute_command       = false
  + health_check_grace_period_seconds = 120
  + iam_role                    = (known after apply)
  + id                        = (known after apply)
  + launch_type               = "FARGATE"
  + name                     = "flask-service-service"
  + platform_version         = (known after apply)
  + region                   = "us-east-1"
  + scheduling_strategy      = "REPLICA"
  + tags_all                 = (known after apply)
  + task_definition          = (known after apply)
  + triggers                 = (known after apply)
  + wait_for_steady_state    = false
}
```

```

+ deployment_configuration (known after apply)

+ load_balancer {
  + container_name = "flask"
  + container_port = 80
  + target_group_arn = (known after apply)
  # (1 unchanged attribute hidden)
}

+ network_configuration {
  + assign_public_ip = false
  + security_groups = (known after apply)
  + subnets        = (known after apply)
}
}

# aws_ecs_task_definition.task will be created
+ resource "aws_ecs_task_definition" "task" {
  + arn                  = (known after apply)
  + arn_without_revision = (known after apply)
  + container_definitions = jsonencode(
    [
      + {
        + image          = "550101108440.dkr.ecr.us-
east-1.amazonaws.com/flask-app:latest"
        + logConfiguration = {
          + logDriver = "awslogs"
          + options   = {
            + awslogs-group      = "/ecs/flask-service-dev"
            + awslogs-region     = "us-east-1"
            + awslogs-stream-prefix = "ecs"
          }
        }
        + name          = "flask"
        + portMappings  = [
          + {
            + containerPort = 80
            + hostPort      = 80
            + protocol      = "tcp"
          },

```



```

    ]
  },
]
)
+ cpu = "256"
+ enable_fault_injection = (known after apply)
+ execution_role_arn = (known after apply)
+ family = "flask-task"
+ id = (known after apply)
+ memory = "512"
+ network_mode = "awsvpc"
+ region = "us-east-1"
+ requires_compatibilities = [
  + "FARGATE",
]
+ revision = (known after apply)
+ skip_destroy = false
+ tags_all = (known after apply)
+ task_role_arn = (known after apply)
+ track_latest = false

+ runtime_platform {
  + cpu_architecture = "ARM64"
  + operating_system_family = "LINUX"
}
}

```

aws_eip.nat will be created

```

+ resource "aws_eip" "nat" {
  + allocation_id = (known after apply)
  + arn = (known after apply)
  + association_id = (known after apply)
  + carrier_ip = (known after apply)
  + customer_owned_ip = (known after apply)
  + domain = (known after apply)
  + id = (known after apply)
  + instance = (known after apply)
  + ipam_pool_id = (known after apply)
  + network_border_group = (known after apply)
  + network_interface = (known after apply)
  + private_dns = (known after apply)
}

```

```

+ private_ip      = (known after apply)
+ ptr_record      = (known after apply)
+ public_dns      = (known after apply)
+ public_ip       = (known after apply)
+ public_ipv4_pool = (known after apply)
+ region          = "us-east-1"
+ tags_all        = (known after apply)
}

```

aws_iam_role.ecs_task_execution will be created

```

+ resource "aws_iam_role" "ecs_task_execution" {
  + arn              = (known after apply)
  + assume_role_policy = jsonencode(
    {
      + Statement = [
        + {
          + Action   = "sts:AssumeRole"
          + Effect    = "Allow"
          + Principal = {
            + Service = "ecs-tasks.amazonaws.com"
          }
        },
      ]
      + Version = "2012-10-17"
    }
  )
  + create_date      = (known after apply)
  + force_detach_policies = false
  + id               = (known after apply)
  + managed_policy_arns = (known after apply)
  + max_session_duration = 3600
  + name              = "ecsTaskExecutionRole"
  + name_prefix       = (known after apply)
  + path              = "/"
  + tags_all          = (known after apply)
  + unique_id         = (known after apply)

  + inline_policy (known after apply)
}

```

aws_iam_role_policy_attachment.ecs_task_execution_policy

will be created

```
+ resource "aws_iam_role_policy_attachment"
"ecs_task_execution_policy" {
  + id      = (known after apply)
  + policy_arn = "arn:aws:iam::aws:policy/service-role/
AmazonECSTaskExecutionRolePolicy"
  + role     = "ecsTaskExecutionRole"
}
```

aws_internet_gateway.igw will be created

```
+ resource "aws_internet_gateway" "igw" {
  + arn      = (known after apply)
  + id       = (known after apply)
  + owner_id = (known after apply)
  + region   = "us-east-1"
  + tags_all = (known after apply)
  + vpc_id   = (known after apply)
}
```

aws_lb.app_alb will be created

```
+ resource "aws_lb" "app_alb" {
  + arn                        = (known after apply)
  + arn_suffix                 = (known after apply)
  + client_keep_alive         = 3600
  + desync_mitigation_mode     = "defensive"
  + dns_name                   = (known after
apply)
  + drop_invalid_header_fields = false
  + enable_deletion_protection = false
  + enable_http2               = true
  + enable_tls_version_and_cipher_suite_headers =
false
  + enable_waf_fail_open       = false
  + enable_xff_client_port     = false
  + enable_zonal_shift         = false
  +
enforce_security_group_inbound_rules_on_private_link_traffic =
(known after apply)
  + id                        = (known after apply)
  + idle_timeout              = 60
  + internal                  = (known after apply)
```

```

    + ip_address_type                = (known after
apply)
    + load_balancer_type              = "application"
    + name                            = "flask-alb"
    + name_prefix                     = (known after
apply)
    + preserve_host_header            = false
    + region                          = "us-east-1"
    + secondary_ips_auto_assigned_per_subnet =
(known after apply)
    + security_groups                 = (known after
apply)
    + subnets                        = (known after apply)
    + tags_all                        = (known after apply)
    + vpc_id                          = (known after apply)
    + xff_header_processing_mode      = "append"
    + zone_id                         = (known after apply)

    + subnet_mapping (known after apply)
}

```

```

# aws_lb_listener.http will be created
+ resource "aws_lb_listener" "http" {
    + arn                            = (known after
apply)
    + id                             = (known after
apply)
    + load_balancer_arn              = (known
after apply)
    + port                           = 80
    + protocol                       = "HTTP"
    + region                         = "us-east-1"
    + routing_http_request_x_amzn_mtls_clientcert_header_name
= (known after apply)
    +
routing_http_request_x_amzn_mtls_clientcert_issuer_header_nam
e      = (known after apply)
    +
routing_http_request_x_amzn_mtls_clientcert_leaf_header_name
= (known after apply)
    +

```

routing_http_request_x_amzn_mtls_clientcert_serial_number_header_name = (known after apply)
 +
 routing_http_request_x_amzn_mtls_clientcert_subject_header_name = (known after apply)
 +
 routing_http_request_x_amzn_mtls_clientcert_validity_header_name = (known after apply)
 + routing_http_request_x_amzn_tls_cipher_suite_header_name = (known after apply)
 + routing_http_request_x_amzn_tls_version_header_name = (known after apply)
 +
 routing_http_response_access_control_allow_credentials_header_value = (known after apply)
 +
 routing_http_response_access_control_allow_headers_header_value = (known after apply)
 +
 routing_http_response_access_control_allow_methods_header_value = (known after apply)
 +
 routing_http_response_access_control_allow_origin_header_value = (known after apply)
 +
 routing_http_response_access_control_expose_headers_header_value = (known after apply)
 +
 routing_http_response_access_control_max_age_header_value = (known after apply)
 +
 routing_http_response_content_security_policy_header_value = (known after apply)
 + routing_http_response_server_enabled = (known after apply)
 +
 routing_http_response_strict_transport_security_header_value = (known after apply)
 +
 routing_http_response_x_content_type_options_header_value = (known after apply)

```

+ routing_http_response_x_frame_options_header_value
= (known after apply)
+ ssl_policy = (known after
apply)
+ tags_all = (known after
apply)
+ tcp_idle_timeout_seconds =
(known after apply)

```

```

+ default_action {
+ order = (known after apply)
+ target_group_arn = (known after apply)
+ type = "forward"
}

```

```

+ mutual_authentication (known after apply)
}

```

aws_lb_target_group.app_tg will be created

```

+ resource "aws_lb_target_group" "app_tg" {
+ arn = (known after apply)
+ arn_suffix = (known after apply)
+ connection_termination = (known after apply)
+ deregistration_delay = "300"
+ id = (known after apply)
+ ip_address_type = (known after apply)
+ lambda_multi_value_headers_enabled = false
+ load_balancer_arns = (known after apply)
+ load_balancing_algorithm_type = (known after apply)
+ load_balancing_anomaly_mitigation = (known after apply)
+ load_balancing_cross_zone_enabled = (known after apply)
+ name = "flask-service-tg"
+ name_prefix = (known after apply)
+ port = 80
+ preserve_client_ip = (known after apply)
+ protocol = "HTTP"
+ protocol_version = (known after apply)
+ proxy_protocol_v2 = false
+ region = "us-east-1"
+ slow_start = 0
+ tags_all = (known after apply)
}

```

```
+ target_type          = "ip"
+ vpc_id               = (known after apply)
```

```
+ health_check {
  + enabled            = true
  + healthy_threshold  = 3
  + interval           = 30
  + matcher            = "200-399"
  + path               = "/"
  + port               = "traffic-port"
  + protocol           = "HTTP"
  + timeout            = (known after apply)
  + unhealthy_threshold = 3
}
```

```
+ stickiness (known after apply)
```

```
+ target_failover (known after apply)
```

```
+ target_group_health (known after apply)
```

```
+ target_health_state (known after apply)
```

```
}
```

```
# aws_nat_gateway.nat will be created
```

```
+ resource "aws_nat_gateway" "nat" {
  + allocation_id          = (known after apply)
  + association_id         = (known after apply)
  + connectivity_type       = "public"
  + id                     = (known after apply)
  + network_interface_id    = (known after apply)
  + private_ip             = (known after apply)
  + public_ip              = (known after apply)
  + region                 = "us-east-1"
  + secondary_allocation_ids = (known after apply)
  + secondary_private_ip_address_count = (known after apply)
  + secondary_private_ip_addresses = (known after apply)
  + subnet_id              = (known after apply)
  + tags_all                = (known after apply)
}
```

```
# aws_route.private_nat_access will be created
+ resource "aws_route" "private_nat_access" {
  + destination_cidr_block = "0.0.0.0/0"
  + id                     = (known after apply)
  + instance_id           = (known after apply)
  + instance_owner_id     = (known after apply)
  + nat_gateway_id        = (known after apply)
  + network_interface_id   = (known after apply)
  + origin                 = (known after apply)
  + region                 = "us-east-1"
  + route_table_id        = (known after apply)
  + state                 = (known after apply)
}
```

```
# aws_route.public_internet_access will be created
+ resource "aws_route" "public_internet_access" {
  + destination_cidr_block = "0.0.0.0/0"
  + gateway_id             = (known after apply)
  + id                     = (known after apply)
  + instance_id           = (known after apply)
  + instance_owner_id     = (known after apply)
  + network_interface_id   = (known after apply)
  + origin                 = (known after apply)
  + region                 = "us-east-1"
  + route_table_id        = (known after apply)
  + state                 = (known after apply)
}
```

```
# aws_route_table.private will be created
+ resource "aws_route_table" "private" {
  + arn                 = (known after apply)
  + id                 = (known after apply)
  + owner_id           = (known after apply)
  + propagating_vgws   = (known after apply)
  + region             = "us-east-1"
  + route              = (known after apply)
  + tags_all           = (known after apply)
  + vpc_id             = (known after apply)
}
```

```
# aws_route_table.public will be created
```



```
+ resource "aws_route_table" "public" {  
  + arn          = (known after apply)  
  + id           = (known after apply)  
  + owner_id     = (known after apply)  
  + propagating_vgws = (known after apply)  
  + region       = "us-east-1"  
  + route        = (known after apply)  
  + tags_all     = (known after apply)  
  + vpc_id       = (known after apply)  
}
```

aws_route_table_association.private[0] will be created

```
+ resource "aws_route_table_association" "private" {  
  + id          = (known after apply)  
  + region      = "us-east-1"  
  + route_table_id = (known after apply)  
  + subnet_id   = (known after apply)  
}
```

aws_route_table_association.private[1] will be created

```
+ resource "aws_route_table_association" "private" {  
  + id          = (known after apply)  
  + region      = "us-east-1"  
  + route_table_id = (known after apply)  
  + subnet_id   = (known after apply)  
}
```

aws_route_table_association.public[0] will be created

```
+ resource "aws_route_table_association" "public" {  
  + id          = (known after apply)  
  + region      = "us-east-1"  
  + route_table_id = (known after apply)  
  + subnet_id   = (known after apply)  
}
```

aws_route_table_association.public[1] will be created

```
+ resource "aws_route_table_association" "public" {  
  + id          = (known after apply)  
  + region      = "us-east-1"  
  + route_table_id = (known after apply)  
  + subnet_id   = (known after apply)  
}
```

```
}
```

```
# aws_security_group.alb_sg will be created
+ resource "aws_security_group" "alb_sg" {
  + arn                = (known after apply)
  + description        = "ALB security group"
  + egress              = [
    + {
      + cidr_blocks      = [
        + "0.0.0.0/0",
      ]
      + description      = "Allow all outbound"
      + from_port         = 0
      + ipv6_cidr_blocks = []
      + prefix_list_ids  = []
      + protocol          = "-1"
      + security_groups  = []
      + self              = false
      + to_port           = 0
    },
  ]
  + id                  = (known after apply)
  + ingress              = [
    + {
      + cidr_blocks      = [
        + "0.0.0.0/0",
      ]
      + description      = "Allow HTTP inbound"
      + from_port         = 80
      + ipv6_cidr_blocks = []
      + prefix_list_ids  = []
      + protocol          = "tcp"
      + security_groups  = []
      + self              = false
      + to_port           = 80
    },
  ]
  + name                = "alb-sg"
  + name_prefix          = (known after apply)
  + owner_id             = (known after apply)
  + region               = "us-east-1"
```

```

+ revoke_rules_on_delete = false
+ tags_all                = (known after apply)
+ vpc_id                  = (known after apply)
}

```

aws_security_group.ecs_sg will be created

```

+ resource "aws_security_group" "ecs_sg" {
+   arn                = (known after apply)
+   description        = "ECS tasks security group"
+   egress              = [
+     {
+       + cidr_blocks    = [
+         + "0.0.0.0/0",
+       ]
+       + description    = "Allow all outbound"
+       + from_port      = 0
+       + ipv6_cidr_blocks = []
+       + prefix_list_ids = []
+       + protocol        = "-1"
+       + security_groups = []
+       + self            = false
+       + to_port         = 0
+     },
+   ]
+   id                  = (known after apply)
+   ingress              = [
+     {
+       + cidr_blocks    = []
+       + description    = "Allow ALB to reach ECS tasks on port
80"
+       + from_port      = 80
+       + ipv6_cidr_blocks = []
+       + prefix_list_ids = []
+       + protocol        = "tcp"
+       + security_groups = (known after apply)
+       + self            = false
+       + to_port         = 80
+     },
+   ]
+   name                = "ecs-sg"
+   name_prefix          = (known after apply)

```

```
+ owner_id          = (known after apply)
+ region            = "us-east-1"
+ revoke_rules_on_delete = false
+ tags_all          = (known after apply)
+ vpc_id            = (known after apply)
}
```

aws_subnet.private[0] will be created

```
+ resource "aws_subnet" "private" {
  + arn                  = (known after apply)
  + assign_ipv6_address_on_creation = false
  + availability_zone     = "us-east-1a"
  + availability_zone_id   = (known after apply)
  + cidr_block            = "10.0.160.0/20"
  + enable_dns64          = false
  + enable_resource_name_dns_a_record_on_launch = false
  + enable_resource_name_dns_aaaa_record_on_launch = false
  + id                   = (known after apply)
  + ipv6_cidr_block_association_id = (known after
apply)
  + ipv6_native          = false
  + map_public_ip_on_launch = false
  + owner_id            = (known after apply)
  + private_dns_hostname_type_on_launch = (known after
apply)
  + region              = "us-east-1"
  + tags_all            = (known after apply)
  + vpc_id              = (known after apply)
}
```

aws_subnet.private[1] will be created

```
+ resource "aws_subnet" "private" {
  + arn                  = (known after apply)
  + assign_ipv6_address_on_creation = false
  + availability_zone     = "us-east-1b"
  + availability_zone_id   = (known after apply)
  + cidr_block            = "10.0.176.0/20"
  + enable_dns64          = false
  + enable_resource_name_dns_a_record_on_launch = false
  + enable_resource_name_dns_aaaa_record_on_launch = false
  + id                   = (known after apply)
}
```

```

    + ipv6_cidr_block_association_id      = (known after
apply)
    + ipv6_native                         = false
    + map_public_ip_on_launch             = false
    + owner_id                           = (known after apply)
    + private_dns_hostname_type_on_launch = (known after
apply)
    + region                             = "us-east-1"
    + tags_all                           = (known after apply)
    + vpc_id                             = (known after apply)
  }

```

aws_subnet.public[0] will be created

```

+ resource "aws_subnet" "public" {
  + arn                        = (known after apply)
  + assign_ipv6_address_on_creation = false
  + availability_zone          = "us-east-1a"
  + availability_zone_id       = (known after apply)
  + cidr_block                 = "10.0.0.0/20"
  + enable_dns64               = false
  + enable_resource_name_dns_a_record_on_launch = false
  + enable_resource_name_dns_aaaa_record_on_launch = false
  + id                        = (known after apply)
  + ipv6_cidr_block_association_id = (known after
apply)
  + ipv6_native                = false
  + map_public_ip_on_launch    = true
  + owner_id                   = (known after apply)
  + private_dns_hostname_type_on_launch = (known after
apply)
  + region                     = "us-east-1"
  + tags_all                   = (known after apply)
  + vpc_id                     = (known after apply)
}

```

aws_subnet.public[1] will be created

```

+ resource "aws_subnet" "public" {
  + arn                        = (known after apply)
  + assign_ipv6_address_on_creation = false
  + availability_zone          = "us-east-1b"
  + availability_zone_id       = (known after apply)

```

```

+ cidr_block                = "10.0.16.0/20"
+ enable_dns64              = false
+ enable_resource_name_dns_a_record_on_launch = false
+ enable_resource_name_dns_aaaa_record_on_launch = false
+ id                        = (known after apply)
+ ipv6_cidr_block_association_id = (known after
apply)
+ ipv6_native              = false
+ map_public_ip_on_launch  = true
+ owner_id                 = (known after apply)
+ private_dns_hostname_type_on_launch = (known after
apply)
+ region                  = "us-east-1"
+ tags_all                 = (known after apply)
+ vpc_id                  = (known after apply)
}

```

aws_vpc.main will be created

```

+ resource "aws_vpc" "main" {
+ arn                      = (known after apply)
+ cidr_block              = "10.0.0.0/16"
+ default_network_acl_id  = (known after apply)
+ default_route_table_id  = (known after apply)
+ default_security_group_id = (known after apply)
+ dhcp_options_id         = (known after apply)
+ enable_dns_hostnames    = (known after apply)
+ enable_dns_support      = true
+ enable_network_address_usage_metrics = (known after
apply)
+ id                      = (known after apply)
+ instance_tenancy        = "default"
+ ipv6_association_id     = (known after apply)
+ ipv6_cidr_block         = (known after apply)
+ ipv6_cidr_block_network_border_group = (known after apply)
+ main_route_table_id     = (known after apply)
+ owner_id                = (known after apply)
+ region                  = "us-east-1"
+ tags_all                = (known after apply)
}

```

Plan: 27 to add, 0 to change, 0 to destroy.

Changes to Outputs:

- + alb_dns_name = (known after apply)
- + ecs_cluster_name = "flask-service-dev"
- + running_tasks = 2

aws_ecs_cluster.main: Creating...

aws_cloudwatch_log_group.ecs_logs: Creating...

aws_iam_role.ecs_task_execution: Creating...

aws_vpc.main: Creating...

aws_eip.nat: Creating...

aws_cloudwatch_log_group.ecs_logs: Creation complete after 1s
[id=/ecs/flask-service-dev]

aws_iam_role.ecs_task_execution: Creation complete after 2s
[id=ecsTaskExecutionRole]

aws_iam_role_policy_attachment.ecs_task_execution_policy:
Creating...

aws_ecs_task_definition.task: Creating...

aws_eip.nat: Creation complete after 2s
[id=eipalloc-077f1ed9391ad2db0]

aws_ecs_task_definition.task: Creation complete after 0s [id=flask-task]

aws_iam_role_policy_attachment.ecs_task_execution_policy:
Creation complete after 0s [id=ecsTaskExecutionRole/
arn:aws:iam::aws:policy/service-role/
AmazonECSTaskExecutionRolePolicy]

aws_vpc.main: Creation complete after 3s
[id=vpc-03c55fc904140637c]

aws_subnet.private[1]: Creating...

aws_route_table.public: Creating...

aws_internet_gateway.igw: Creating...

aws_subnet.public[0]: Creating...

aws_subnet.private[0]: Creating...

aws_lb_target_group.app_tg: Creating...

aws_route_table.private: Creating...

aws_subnet.public[1]: Creating...

aws_security_group.alb_sg: Creating...

aws_subnet.private[1]: Creation complete after 2s
[id=subnet-082b24bc961d63241]

aws_route_table.public: Creation complete after 2s
[id=rtb-0d4315bdacdd240ce]

aws_subnet.private[0]: Creation complete after 2s

[id=subnet-084d5c30067102c75]
aws_internet_gateway.igw: Creation complete after 2s
[id=igw-0b687ce9c38a957a0]
aws_route.public_internet_access: Creating...
aws_route_table.private: Creation complete after 2s
[id=rtb-0644c490fd39d7c6d]
aws_route_table_association.private[0]: Creating...
aws_route_table_association.private[1]: Creating...
aws_route_table_association.private[0]: Creation complete after 1s
[id=rtbassoc-0f2650cbdf752d539]
aws_route_table_association.private[1]: Creation complete after 1s
[id=rtbassoc-0dfc057e362081f99]
aws_route.public_internet_access: Creation complete after 2s
[id=r-rtb-0d4315bdacdd240ce1080289494]
aws_lb_target_group.app_tg: Creation complete after 4s
[id=arn:aws:elasticloadbalancing:us-east-1:550101108440:targetgroup/flask-service-tg/c6b5b66a04cf4b42]
aws_security_group.alb_sg: Creation complete after 5s
[id=sg-01d0854f620d31526]
aws_security_group.ecs_sg: Creating...
aws_ecs_cluster.main: Still creating... [00m10s elapsed]
aws_security_group.ecs_sg: Creation complete after 4s
[id=sg-08000d7164421d98f]
aws_ecs_cluster.main: Creation complete after 12s
[id=arn:aws:ecs:us-east-1:550101108440:cluster/flask-service-dev]
aws_subnet.public[0]: Still creating... [00m10s elapsed]
aws_subnet.public[1]: Still creating... [00m10s elapsed]
aws_subnet.public[0]: Creation complete after 12s
[id=subnet-09d4602e9f6aef303]
aws_nat_gateway.nat: Creating...
aws_subnet.public[1]: Creation complete after 13s
[id=subnet-0cc6b8aa68c55d73c]
aws_route_table_association.public[0]: Creating...
aws_route_table_association.public[1]: Creating...
aws_lb.app_alb: Creating...
aws_route_table_association.public[0]: Creation complete after 1s
[id=rtbassoc-02124e61f12298822]
aws_route_table_association.public[1]: Creation complete after 1s
[id=rtbassoc-0d341595af0a213a4]
aws_nat_gateway.nat: Still creating... [00m10s elapsed]

aws_lb.app_alb: Still creating... [00m10s elapsed]
aws_nat_gateway.nat: Still creating... [00m20s elapsed]
aws_lb.app_alb: Still creating... [00m20s elapsed]
aws_nat_gateway.nat: Still creating... [00m31s elapsed]
aws_lb.app_alb: Still creating... [00m30s elapsed]
aws_nat_gateway.nat: Still creating... [00m41s elapsed]
aws_lb.app_alb: Still creating... [00m40s elapsed]
aws_nat_gateway.nat: Still creating... [00m51s elapsed]
aws_lb.app_alb: Still creating... [00m50s elapsed]
aws_nat_gateway.nat: Still creating... [01m01s elapsed]
aws_lb.app_alb: Still creating... [01m00s elapsed]
aws_nat_gateway.nat: Still creating... [01m11s elapsed]
aws_lb.app_alb: Still creating... [01m10s elapsed]
aws_nat_gateway.nat: Still creating... [01m21s elapsed]
aws_lb.app_alb: Still creating... [01m20s elapsed]
aws_nat_gateway.nat: Still creating... [01m31s elapsed]
aws_lb.app_alb: Still creating... [01m30s elapsed]
aws_nat_gateway.nat: Creation complete after 1m38s
[id=nat-05a58cac315d8f7e5]
aws_route.private_nat_access: Creating...
aws_route.private_nat_access: Creation complete after 2s [id=r-
rtb-0644c490fd39d7c6d1080289494]
aws_lb.app_alb: Still creating... [01m40s elapsed]
aws_lb.app_alb: Still creating... [01m50s elapsed]
aws_lb.app_alb: Still creating... [02m00s elapsed]
aws_lb.app_alb: Still creating... [02m10s elapsed]
aws_lb.app_alb: Still creating... [02m20s elapsed]
aws_lb.app_alb: Still creating... [02m30s elapsed]
aws_lb.app_alb: Still creating... [02m40s elapsed]
aws_lb.app_alb: Still creating... [02m50s elapsed]
aws_lb.app_alb: Still creating... [03m00s elapsed]
aws_lb.app_alb: Creation complete after 3m7s
[id=arn:aws:elasticloadbalancing:us-
east-1:550101108440:loadbalancer/app/flask-alb/6923f38c0b839406]
aws_lb_listener.http: Creating...
aws_lb_listener.http: Creation complete after 1s
[id=arn:aws:elasticloadbalancing:us-east-1:550101108440:listener/
app/flask-alb/6923f38c0b839406/b8c81681259aed28]
aws_ecs_service.service: Creating...
aws_ecs_service.service: Creation complete after 3s
[id=arn:aws:ecs:us-east-1:550101108440:service/flask-service-dev/

flask-service-service]

Apply complete! Resources: 27 added, 0 changed, 0 destroyed.

Outputs:

alb_dns_name = "flask-alb-1803430498.us-east-1.elb.amazonaws.com"
ecs_cluster_name = "flask-service-dev"
running_tasks = 2
maresh.singh@del1-lmc-n82481 capstone-aws %

Terraform Destroy Log :

maresh.singh@del1-lmc-n82481 capstone-aws % terraform destroy
data.aws_availability_zones.available: Reading...
aws_ecs_cluster.main: Refreshing state... [id=arn:aws:ecs:us-east-1:550101108440:cluster/flask-service-dev]
data.aws_caller_identity.current: Reading...
data.aws_partition.current: Reading...
aws_cloudwatch_log_group.ecs_logs: Refreshing state... [id=/ecs/flask-service-dev]
aws_eip.nat: Refreshing state... [id=eipalloc-077f1ed9391ad2db0]
aws_vpc.main: Refreshing state... [id=vpc-03c55fc904140637c]
aws_iam_role.ecs_task_execution: Refreshing state...
[id=ecsTaskExecutionRole]
data.aws_partition.current: Read complete after 0s [id=aws]
data.aws_caller_identity.current: Read complete after 0s
[id=550101108440]
data.aws_availability_zones.available: Read complete after 1s
[id=us-east-1]
aws_iam_role_policy_attachment.ecs_task_execution_policy: Refreshing state... [id=ecsTaskExecutionRole/arn:aws:iam::aws:policy/service-role/AmazonECSTaskExecutionRolePolicy]
aws_ecs_task_definition.task: Refreshing state... [id=flask-task]
aws_route_table.private: Refreshing state...
[id=rtb-0644c490fd39d7c6d]
aws_internet_gateway.igw: Refreshing state...
[id=igw-0b687ce9c38a957a0]

aws_route_table.public: Refreshing state...
[id=rtb-0d4315bdacdd240ce]
aws_subnet.private[0]: Refreshing state...
[id=subnet-084d5c30067102c75]
aws_subnet.private[1]: Refreshing state...
[id=subnet-082b24bc961d63241]
aws_security_group.alb_sg: Refreshing state...
[id=sg-01d0854f620d31526]
aws_subnet.public[1]: Refreshing state...
[id=subnet-0cc6b8aa68c55d73c]
aws_lb_target_group.app_tg: Refreshing state...
[id=arn:aws:elasticloadbalancing:us-east-1:550101108440:targetgroup/flask-service-tg/c6b5b66a04cf4b42]
aws_subnet.public[0]: Refreshing state...
[id=subnet-09d4602e9f6aef303]
aws_route.public_internet_access: Refreshing state... [id=r-rtb-0d4315bdacdd240ce1080289494]
aws_route_table_association.private[0]: Refreshing state...
[id=rtbassoc-0f2650cbdf752d539]
aws_route_table_association.private[1]: Refreshing state...
[id=rtbassoc-0dfc057e362081f99]
aws_security_group.ecs_sg: Refreshing state...
[id=sg-08000d7164421d98f]
aws_route_table_association.public[0]: Refreshing state...
[id=rtbassoc-02124e61f12298822]
aws_route_table_association.public[1]: Refreshing state...
[id=rtbassoc-0d341595af0a213a4]
aws_nat_gateway.nat: Refreshing state...
[id=nat-05a58cac315d8f7e5]
aws_lb.app_alb: Refreshing state...
[id=arn:aws:elasticloadbalancing:us-east-1:550101108440:loadbalancer/app/flask-alb/6923f38c0b839406]
aws_route.private_nat_access: Refreshing state... [id=r-rtb-0644c490fd39d7c6d1080289494]
aws_lb_listener.http: Refreshing state...
[id=arn:aws:elasticloadbalancing:us-east-1:550101108440:listener/app/flask-alb/6923f38c0b839406/b8c81681259aed28]
aws_ecs_service.service: Refreshing state... [id=arn:aws:ecs:us-east-1:550101108440:service/flask-service-dev/flask-service-service]

Terraform used the selected providers to generate the following execution plan. Resource actions are indicated with the following symbols:

- destroy

Terraform will perform the following actions:

```
# aws_cloudwatch_log_group.ecs_logs will be destroyed
- resource "aws_cloudwatch_log_group" "ecs_logs" {
  - arn          = "arn:aws:logs:us-east-1:550101108440:log-
group:/ecs/flask-service-dev" -> null
  - id          = "/ecs/flask-service-dev" -> null
  - log_group_class = "STANDARD" -> null
  - name        = "/ecs/flask-service-dev" -> null
  - region      = "us-east-1" -> null
  - retention_in_days = 7 -> null
  - skip_destroy   = false -> null
  - tags         = {} -> null
  - tags_all      = {} -> null
  # (2 unchanged attributes hidden)
}

# aws_ecs_cluster.main will be destroyed
- resource "aws_ecs_cluster" "main" {
  - arn    = "arn:aws:ecs:us-east-1:550101108440:cluster/flask-
service-dev" -> null
  - id     = "arn:aws:ecs:us-east-1:550101108440:cluster/flask-
service-dev" -> null
  - name   = "flask-service-dev" -> null
  - region = "us-east-1" -> null
  - tags   = {} -> null
  - tags_all = {} -> null

  - setting {
    - name = "containerInsights" -> null
    - value = "disabled" -> null
  }
}

# aws_ecs_service.service will be destroyed
```

```

- resource "aws_ecs_service" "service" {
  - arn = "arn:aws:ecs:us-east-1:550101108440:service/flask-service-dev/flask-service-service" -> null
  - availability_zone_rebalancing = "ENABLED" -> null
  - cluster = "arn:aws:ecs:us-east-1:550101108440:cluster/flask-service-dev" -> null
  - deployment_maximum_percent = 200 -> null
  - deployment_minimum_healthy_percent = 100 -> null
  - desired_count = 2 -> null
  - enable_ecs_managed_tags = false -> null
  - enable_execute_command = false -> null
  - health_check_grace_period_seconds = 120 -> null
  - iam_role = "/aws-service-role/ecs.amazonaws.com/AWSServiceRoleForECS" -> null
  - id = "arn:aws:ecs:us-east-1:550101108440:service/flask-service-dev/flask-service-service" -> null
  - launch_type = "FARGATE" -> null
  - name = "flask-service-service" -> null
  - platform_version = "LATEST" -> null
  - propagate_tags = "NONE" -> null
  - region = "us-east-1" -> null
  - scheduling_strategy = "REPLICA" -> null
  - tags = {} -> null
  - tags_all = {} -> null
  - task_definition = "arn:aws:ecs:us-east-1:550101108440:task-definition/flask-task:22" -> null
  - triggers = {} -> null
  - wait_for_steady_state = false -> null

  - deployment_circuit_breaker {
    - enable = false -> null
    - rollback = false -> null
  }

  - deployment_configuration {
    - bake_time_in_minutes = "0" -> null
    - strategy = "ROLLING" -> null
  }

```

```

- deployment_controller {
  - type = "ECS" -> null
}

- load_balancer {
  - container_name = "flask" -> null
  - container_port = 80 -> null
  - target_group_arn = "arn:aws:elasticloadbalancing:us-
east-1:550101108440:targetgroup/flask-service-tg/
c6b5b66a04cf4b42" -> null
  # (1 unchanged attribute hidden)
}

- network_configuration {
  - assign_public_ip = false -> null
  - security_groups = [
    - "sg-08000d7164421d98f",
  ] -> null
  - subnets = [
    - "subnet-082b24bc961d63241",
    - "subnet-084d5c30067102c75",
  ] -> null
}
}

# aws_ecs_task_definition.task will be destroyed
- resource "aws_ecs_task_definition" "task" {
  - arn = "arn:aws:ecs:us-east-1:550101108440:task-
definition/flask-task:22" -> null
  - arn_without_revision = "arn:aws:ecs:us-
east-1:550101108440:task-definition/flask-task" -> null
  - container_definitions = jsonencode(
    [
      - {
        - environment = []
        - essential = true
        - image = "550101108440.dkr.ecr.us-
east-1.amazonaws.com/flask-app:latest"
        - logConfiguration = {
          - logDriver = "awslogs"
          - options = {

```

```

        - awslogs-group      = "/ecs/flask-service-dev"
        - awslogs-region    = "us-east-1"
        - awslogs-stream-prefix = "ecs"
    }
}
- mountPoints    = []
- name          = "flask"
- portMappings  = [
    - {
        - containerPort = 80
        - hostPort      = 80
        - protocol       = "tcp"
    },
]
- systemControls = []
- volumesFrom    = []
},
]
) -> null
- cpu              = "256" -> null
- enable_fault_injection = false -> null
- execution_role_arn  = "arn:aws:iam::550101108440:role/
ecsTaskExecutionRole" -> null
- family           = "flask-task" -> null
- id               = "flask-task" -> null
- memory           = "512" -> null
- network_mode     = "awsvpc" -> null
- region           = "us-east-1" -> null
- requires_compatibilities = [
    - "FARGATE",
] -> null
- revision         = 22 -> null
- skip_destroy     = false -> null
- tags             = {} -> null
- tags_all         = {} -> null
- task_role_arn    = "arn:aws:iam::550101108440:role/
ecsTaskExecutionRole" -> null
- track_latest     = false -> null
# (2 unchanged attributes hidden)

- runtime_platform {

```

```

    - cpu_architecture      = "ARM64" -> null
    - operating_system_family = "LINUX" -> null
  }
}

# aws_eip.nat will be destroyed
- resource "aws_eip" "nat" {
  - allocation_id      = "eipalloc-077f1ed9391ad2db0" -> null
  - arn                = "arn:aws:ec2:us-
east-1:550101108440:elastic-ip/eipalloc-077f1ed9391ad2db0" -> null
  - association_id      = "eipassoc-0327226e9b9b05569" -> null
  - domain              = "vpc" -> null
  - id                  = "eipalloc-077f1ed9391ad2db0" -> null
  - network_border_group = "us-east-1" -> null
  - network_interface    = "eni-0e87503f7f865f74a" -> null
  - private_dns          = "ip-10-0-11-21.ec2.internal" -> null
  - private_ip           = "10.0.11.21" -> null
  - public_dns           =
"ec2-23-23-135-18.compute-1.amazonaws.com" -> null
  - public_ip            = "23.23.135.18" -> null
  - public_ipv4_pool      = "amazon" -> null
  - region                = "us-east-1" -> null
  - tags                  = {} -> null
  - tags_all              = {} -> null
  # (5 unchanged attributes hidden)
}

# aws_iam_role.ecs_task_execution will be destroyed
- resource "aws_iam_role" "ecs_task_execution" {
  - arn                = "arn:aws:iam::550101108440:role/
ecsTaskExecutionRole" -> null
  - assume_role_policy = jsonencode(
    {
      - Statement = [
        - {
          - Action   = "sts:AssumeRole"
          - Effect    = "Allow"
          - Principal = {
            - Service = "ecs-tasks.amazonaws.com"
          }
        }
      ],

```



```

    ]
    - Version = "2012-10-17"
  }
) -> null
- create_date = "2025-11-22T14:22:51Z" -> null
- force_detach_policies = false -> null
- id = "ecsTaskExecutionRole" -> null
- managed_policy_arns = [
  - "arn:aws:iam::aws:policy/service-role/
AmazonECSTaskExecutionRolePolicy",
] -> null
- max_session_duration = 3600 -> null
- name = "ecsTaskExecutionRole" -> null
- path = "/" -> null
- tags = {} -> null
- tags_all = {} -> null
- unique_id = "AROYAFEUY3MB7UA7BVKJ" -> null
# (3 unchanged attributes hidden)
}

```

aws_iam_role_policy_attachment.ecs_task_execution_policy will be destroyed

```

- resource "aws_iam_role_policy_attachment"
"ecs_task_execution_policy" {
  - id = "ecsTaskExecutionRole/arn:aws:iam::aws:policy/
service-role/AmazonECSTaskExecutionRolePolicy" -> null
  - policy_arn = "arn:aws:iam::aws:policy/service-role/
AmazonECSTaskExecutionRolePolicy" -> null
  - role = "ecsTaskExecutionRole" -> null
}

```

aws_internet_gateway.igw will be destroyed

```

- resource "aws_internet_gateway" "igw" {
  - arn = "arn:aws:ec2:us-east-1:550101108440:internet-
gateway/igw-0b687ce9c38a957a0" -> null
  - id = "igw-0b687ce9c38a957a0" -> null
  - owner_id = "550101108440" -> null
  - region = "us-east-1" -> null
  - tags = {} -> null
  - tags_all = {} -> null
  - vpc_id = "vpc-03c55fc904140637c" -> null
}

```

```

}

# aws_lb.app_alb will be destroyed
- resource "aws_lb" "app_alb" {
  - arn =
"arn:aws:elasticloadbalancing:us-
east-1:550101108440:loadbalancer/app/flask-alb/
6923f38c0b839406" -> null
  - arn_suffix = "app/flask-alb/
6923f38c0b839406" -> null
  - client_keep_alive = 3600 -> null
  - desync_mitigation_mode = "defensive"
-> null
  - dns_name = "flask-
alb-1803430498.us-east-1.elb.amazonaws.com" -> null
  - drop_invalid_header_fields = false -> null
  - enable_cross_zone_load_balancing = true ->
null
  - enable_deletion_protection = false -> null
  - enable_http2 = true -> null
  - enable_tls_version_and_cipher_suite_headers =
false -> null
  - enable_waf_fail_open = false -> null
  - enable_xff_client_port = false -> null
  - enable_zonal_shift = false -> null
  - id =
"arn:aws:elasticloadbalancing:us-
east-1:550101108440:loadbalancer/app/flask-alb/
6923f38c0b839406" -> null
  - idle_timeout = 60 -> null
  - internal = false -> null
  - ip_address_type = "ipv4" -> null
  - load_balancer_type = "application" ->
null
  - name = "flask-alb" -> null
  - preserve_host_header = false -> null
  - region = "us-east-1" -> null
  - security_groups = [
    - "sg-01d0854f620d31526",
  ] -> null
  - subnets = [

```

```

    - "subnet-09d4602e9f6aef303",
    - "subnet-0cc6b8aa68c55d73c",
  ] -> null
  - tags                                = {} -> null
  - tags_all                            = {} -> null
  - vpc_id                              =
"vpc-03c55fc904140637c" -> null
  - xff_header_processing_mode          = "append"
-> null
  - zone_id                             = "Z35SXDOTRQ7X7K"
-> null
  # (3 unchanged attributes hidden)

  - access_logs {
    - enabled = false -> null
    # (2 unchanged attributes hidden)
  }

  - connection_logs {
    - enabled = false -> null
    # (2 unchanged attributes hidden)
  }

  - subnet_mapping {
    - subnet_id      = "subnet-09d4602e9f6aef303" -> null
    # (4 unchanged attributes hidden)
  }
  - subnet_mapping {
    - subnet_id      = "subnet-0cc6b8aa68c55d73c" -> null
    # (4 unchanged attributes hidden)
  }
}

# aws_lb_listener.http will be destroyed
- resource "aws_lb_listener" "http" {
  - arn                                =
"arn:aws:elasticloadbalancing:us-east-1:550101108440:listener/
app/flask-alb/6923f38c0b839406/b8c81681259aed28" -> null
  - id                                =
"arn:aws:elasticloadbalancing:us-east-1:550101108440:listener/
app/flask-alb/6923f38c0b839406/b8c81681259aed28" -> null

```

```

- load_balancer_arn =
"arn:aws:elasticloadbalancing:us-
east-1:550101108440:loadbalancer/app/flask-alb/
6923f38c0b839406" -> null
- port = 80 -> null
- protocol = "HTTP" -> null
- region = "us-east-1" -> null
- routing_http_response_server_enabled =
true -> null
- tags = {} -> null
- tags_all = {} -> null
# (11 unchanged attributes hidden)

- default_action {
  - order = 1 -> null
  - target_group_arn = "arn:aws:elasticloadbalancing:us-
east-1:550101108440:targetgroup/flask-service-tg/
c6b5b66a04cf4b42" -> null
  - type = "forward" -> null
}
}

# aws_lb_target_group.app_tg will be destroyed
- resource "aws_lb_target_group" "app_tg" {
  - arn = "arn:aws:elasticloadbalancing:us-
east-1:550101108440:targetgroup/flask-service-tg/
c6b5b66a04cf4b42" -> null
  - arn_suffix = "targetgroup/flask-service-tg/
c6b5b66a04cf4b42" -> null
  - deregistration_delay = "300" -> null
  - id = "arn:aws:elasticloadbalancing:us-
east-1:550101108440:targetgroup/flask-service-tg/
c6b5b66a04cf4b42" -> null
  - ip_address_type = "ipv4" -> null
  - lambda_multi_value_headers_enabled = false -> null
  - load_balancer_arns = [
    - "arn:aws:elasticloadbalancing:us-
east-1:550101108440:loadbalancer/app/flask-alb/
6923f38c0b839406",
  ] -> null
  - load_balancing_algorithm_type = "round_robin" -> null

```

- load_balancing_anomaly_mitigation = "off" -> null
- load_balancing_cross_zone_enabled =
- "use_load_balancer_configuration" -> null
- name = "flask-service-tg" -> null
- port = 80 -> null
- protocol = "HTTP" -> null
- protocol_version = "HTTP1" -> null
- proxy_protocol_v2 = false -> null
- region = "us-east-1" -> null
- slow_start = 0 -> null
- tags = {} -> null
- tags_all = {} -> null
- target_type = "ip" -> null
- vpc_id = "vpc-03c55fc904140637c" -> null
- # (1 unchanged attribute hidden)
- health_check {
 - enabled = true -> null
 - healthy_threshold = 3 -> null
 - interval = 30 -> null
 - matcher = "200-399" -> null
 - path = "/" -> null
 - port = "traffic-port" -> null
 - protocol = "HTTP" -> null
 - timeout = 5 -> null
 - unhealthy_threshold = 3 -> null
- }
- stickiness {
 - cookie_duration = 86400 -> null
 - enabled = false -> null
 - type = "lb_cookie" -> null
 - # (1 unchanged attribute hidden)
- }
- target_failover {}
- target_group_health {
 - dns_failover {
 - minimum_healthy_targets_count = "1" -> null
 - minimum_healthy_targets_percentage = "off" -> null

```

    }
    - unhealthy_state_routing {
      - minimum_healthy_targets_count    = 1 -> null
      - minimum_healthy_targets_percentage = "off" -> null
    }
  }

  - target_health_state {}
}

# aws_nat_gateway.nat will be destroyed
- resource "aws_nat_gateway" "nat" {
  - allocation_id          = "eipalloc-077f1ed9391ad2db0" ->
null
  - association_id         = "eipassoc-0327226e9b9b05569"
-> null
  - connectivity_type      = "public" -> null
  - id                     = "nat-05a58cac315d8f7e5" -> null
  - network_interface_id   = "eni-0e87503f7f865f74a" -> null
  - private_ip             = "10.0.11.21" -> null
  - public_ip              = "23.23.135.18" -> null
  - region                 = "us-east-1" -> null
  - secondary_allocation_ids = [] -> null
  - secondary_private_ip_address_count = 0 -> null
  - secondary_private_ip_addresses = [] -> null
  - subnet_id              = "subnet-09d4602e9f6aef303" -> null
  - tags                   = {} -> null
  - tags_all                = {} -> null
}

# aws_route.private_nat_access will be destroyed
- resource "aws_route" "private_nat_access" {
  - destination_cidr_block = "0.0.0.0/0" -> null
  - id                     = "r-rtb-0644c490fd39d7c6d1080289494" ->
null
  - nat_gateway_id        = "nat-05a58cac315d8f7e5" -> null
  - origin                 = "CreateRoute" -> null
  - region                 = "us-east-1" -> null
  - route_table_id        = "rtb-0644c490fd39d7c6d" -> null
  - state                  = "active" -> null
  # (13 unchanged attributes hidden)

```

```

}

# aws_route.public_internet_access will be destroyed
- resource "aws_route" "public_internet_access" {
  - destination_cidr_block    = "0.0.0.0/0" -> null
  - gateway_id                = "igw-0b687ce9c38a957a0" -> null
  - id                        = "r-rtb-0d4315bdacdd240ce1080289494" ->
null
  - origin                    = "CreateRoute" -> null
  - region                    = "us-east-1" -> null
  - route_table_id            = "rtb-0d4315bdacdd240ce" -> null
  - state                      = "active" -> null
  # (13 unchanged attributes hidden)
}

```

```

# aws_route_table.private will be destroyed
- resource "aws_route_table" "private" {
  - arn                      = "arn:aws:ec2:us-east-1:550101108440:route-
table/rtb-0644c490fd39d7c6d" -> null
  - id                      = "rtb-0644c490fd39d7c6d" -> null
  - owner_id                = "550101108440" -> null
  - propagating_vgws        = [] -> null
  - region                  = "us-east-1" -> null
  - route                    = [
    - {
      - cidr_block            = "0.0.0.0/0"
      - nat_gateway_id        = "nat-05a58cac315d8f7e5"
      # (11 unchanged attributes hidden)
    },
  ] -> null
  - tags                    = {} -> null
  - tags_all                = {} -> null
  - vpc_id                  = "vpc-03c55fc904140637c" -> null
}

```

```

# aws_route_table.public will be destroyed
- resource "aws_route_table" "public" {
  - arn                      = "arn:aws:ec2:us-east-1:550101108440:route-
table/rtb-0d4315bdacdd240ce" -> null
  - id                      = "rtb-0d4315bdacdd240ce" -> null
  - owner_id                = "550101108440" -> null
}

```

```

- propagating_vgws = [] -> null
- region           = "us-east-1" -> null
- route            = [
  - {
    - cidr_block      = "0.0.0.0/0"
    - gateway_id      = "igw-0b687ce9c38a957a0"
    # (11 unchanged attributes hidden)
  },
] -> null
- tags              = {} -> null
- tags_all          = {} -> null
- vpc_id            = "vpc-03c55fc904140637c" -> null
}

```

aws_route_table_association.private[0] will be destroyed

```

- resource "aws_route_table_association" "private" {
  - id          = "rtbassoc-0f2650cbdf752d539" -> null
  - region      = "us-east-1" -> null
  - route_table_id = "rtb-0644c490fd39d7c6d" -> null
  - subnet_id    = "subnet-084d5c30067102c75" -> null
  # (1 unchanged attribute hidden)
}

```

aws_route_table_association.private[1] will be destroyed

```

- resource "aws_route_table_association" "private" {
  - id          = "rtbassoc-0dfc057e362081f99" -> null
  - region      = "us-east-1" -> null
  - route_table_id = "rtb-0644c490fd39d7c6d" -> null
  - subnet_id    = "subnet-082b24bc961d63241" -> null
  # (1 unchanged attribute hidden)
}

```

aws_route_table_association.public[0] will be destroyed

```

- resource "aws_route_table_association" "public" {
  - id          = "rtbassoc-02124e61f12298822" -> null
  - region      = "us-east-1" -> null
  - route_table_id = "rtb-0d4315bdacdd240ce" -> null
  - subnet_id    = "subnet-09d4602e9f6aef303" -> null
  # (1 unchanged attribute hidden)
}

```


aws_route_table_association.public[1] will be destroyed

```
- resource "aws_route_table_association" "public" {  
  - id          = "rtbassoc-0d341595af0a213a4" -> null  
  - region      = "us-east-1" -> null  
  - route_table_id = "rtb-0d4315bdacdd240ce" -> null  
  - subnet_id    = "subnet-0cc6b8aa68c55d73c" -> null  
  # (1 unchanged attribute hidden)  
}
```

aws_security_group.alb_sg will be destroyed

```
- resource "aws_security_group" "alb_sg" {  
  - arn          = "arn:aws:ec2:us-  
east-1:550101108440:security-group/sg-01d0854f620d31526" -> null  
  - description  = "ALB security group" -> null  
  - egress       = [  
    - {  
      - cidr_blocks = [  
        - "0.0.0.0/0",  
      ]  
      - description  = "Allow all outbound"  
      - from_port    = 0  
      - ipv6_cidr_blocks = []  
      - prefix_list_ids = []  
      - protocol      = "-1"  
      - security_groups = []  
      - self          = false  
      - to_port       = 0  
    },  
  ] -> null  
  - id           = "sg-01d0854f620d31526" -> null  
  - ingress      = [  
    - {  
      - cidr_blocks = [  
        - "0.0.0.0/0",  
      ]  
      - description  = "Allow HTTP inbound"  
      - from_port    = 80  
      - ipv6_cidr_blocks = []  
      - prefix_list_ids = []  
      - protocol      = "tcp"  
      - security_groups = []  
    }  
  ]  
}
```

```

        - self          = false
        - to_port       = 80
    },
] -> null
- name                = "alb-sg" -> null
- owner_id            = "550101108440" -> null
- region              = "us-east-1" -> null
- revoke_rules_on_delete = false -> null
- tags                = {} -> null
- tags_all            = {} -> null
- vpc_id              = "vpc-03c55fc904140637c" -> null
# (1 unchanged attribute hidden)
}

# aws_security_group.ecs_sg will be destroyed
- resource "aws_security_group" "ecs_sg" {
  - arn                = "arn:aws:ec2:us-
east-1:550101108440:security-group/sg-08000d7164421d98f" -> null
  - description        = "ECS tasks security group" -> null
  - egress              = [
    - {
      - cidr_blocks      = [
        - "0.0.0.0/0",
      ]
      - description      = "Allow all outbound"
      - from_port        = 0
      - ipv6_cidr_blocks = []
      - prefix_list_ids  = []
      - protocol         = "-1"
      - security_groups  = []
      - self             = false
      - to_port          = 0
    },
  ] -> null
  - id                  = "sg-08000d7164421d98f" -> null
  - ingress              = [
    - {
      - cidr_blocks      = []
      - description      = "Allow ALB to reach ECS tasks on port
80"
      - from_port        = 80

```

```

- ipv6_cidr_blocks = []
- prefix_list_ids = []
- protocol        = "tcp"
- security_groups = [
  - "sg-01d0854f620d31526",
]
- self            = false
- to_port         = 80
},
] -> null
- name            = "ecs-sg" -> null
- owner_id        = "550101108440" -> null
- region          = "us-east-1" -> null
- revoke_rules_on_delete = false -> null
- tags            = {} -> null
- tags_all        = {} -> null
- vpc_id          = "vpc-03c55fc904140637c" -> null
# (1 unchanged attribute hidden)
}

```

aws_subnet.private[0] will be destroyed

```

- resource "aws_subnet" "private" {
  - arn              = "arn:aws:ec2:us-
east-1:550101108440:subnet/subnet-084d5c30067102c75" -> null
  - assign_ipv6_address_on_creation      = false -> null
  - availability_zone                    = "us-east-1a" -> null
  - availability_zone_id                 = "use1-az4" -> null
  - cidr_block                          = "10.0.160.0/20" -> null
  - enable_dns64                        = false -> null
  - enable_ini_at_device_index          = 0 -> null
  - enable_resource_name_dns_a_record_on_launch = false ->
null
  - enable_resource_name_dns_aaaa_record_on_launch = false
-> null
  - id                                  = "subnet-084d5c30067102c75" ->
null
  - ipv6_native                         = false -> null
  - map_customer_owned_ip_on_launch     = false -> null
  - map_public_ip_on_launch             = false -> null
  - owner_id                            = "550101108440" -> null
  - private_dns_hostname_type_on_launch = "ip-name" ->

```

```

null
  - region                = "us-east-1" -> null
  - tags                  = {} -> null
  - tags_all              = {} -> null
  - vpc_id                = "vpc-03c55fc904140637c" ->
null
  # (4 unchanged attributes hidden)
}

# aws_subnet.private[1] will be destroyed
- resource "aws_subnet" "private" {
  - arn                    = "arn:aws:ec2:us-
east-1:550101108440:subnet/subnet-082b24bc961d63241" -> null
  - assign_ipv6_address_on_creation = false -> null
  - availability_zone       = "us-east-1b" -> null
  - availability_zone_id    = "use1-az6" -> null
  - cidr_block              = "10.0.176.0/20" -> null
  - enable_dns64            = false -> null
  - enable_ini_at_device_index = 0 -> null
  - enable_resource_name_dns_a_record_on_launch = false ->
null
  - enable_resource_name_dns_aaaa_record_on_launch = false
-> null
  - id                     = "subnet-082b24bc961d63241" ->
null
  - ipv6_native            = false -> null
  - map_customer_owned_ip_on_launch = false -> null
  - map_public_ip_on_launch = false -> null
  - owner_id               = "550101108440" -> null
  - private_dns_hostname_type_on_launch = "ip-name" ->
null
  - region                = "us-east-1" -> null
  - tags                  = {} -> null
  - tags_all              = {} -> null
  - vpc_id                = "vpc-03c55fc904140637c" ->
null
  # (4 unchanged attributes hidden)
}

# aws_subnet.public[0] will be destroyed
- resource "aws_subnet" "public" {

```

```

- arn = "arn:aws:ec2:us-
east-1:550101108440:subnet/subnet-09d4602e9f6aef303" -> null
- assign_ipv6_address_on_creation = false -> null
- availability_zone = "us-east-1a" -> null
- availability_zone_id = "use1-az4" -> null
- cidr_block = "10.0.0.0/20" -> null
- enable_dns64 = false -> null
- enable_ini_at_device_index = 0 -> null
- enable_resource_name_dns_a_record_on_launch = false ->
null
- enable_resource_name_dns_aaaa_record_on_launch = false
-> null
- id = "subnet-09d4602e9f6aef303" ->
null
- ipv6_native = false -> null
- map_customer_owned_ip_on_launch = false -> null
- map_public_ip_on_launch = true -> null
- owner_id = "550101108440" -> null
- private_dns_hostname_type_on_launch = "ip-name" ->
null
- region = "us-east-1" -> null
- tags = {} -> null
- tags_all = {} -> null
- vpc_id = "vpc-03c55fc904140637c" ->
null
# (4 unchanged attributes hidden)
}

```

aws_subnet.public[1] will be destroyed

```

- resource "aws_subnet" "public" {
- arn = "arn:aws:ec2:us-
east-1:550101108440:subnet/subnet-0cc6b8aa68c55d73c" -> null
- assign_ipv6_address_on_creation = false -> null
- availability_zone = "us-east-1b" -> null
- availability_zone_id = "use1-az6" -> null
- cidr_block = "10.0.16.0/20" -> null
- enable_dns64 = false -> null
- enable_ini_at_device_index = 0 -> null
- enable_resource_name_dns_a_record_on_launch = false ->
null
- enable_resource_name_dns_aaaa_record_on_launch = false

```

```

-> null
  - id = "subnet-0cc6b8aa68c55d73c" ->
null
  - ipv6_native = false -> null
  - map_customer_owned_ip_on_launch = false -> null
  - map_public_ip_on_launch = true -> null
  - owner_id = "550101108440" -> null
  - private_dns_hostname_type_on_launch = "ip-name" ->
null
  - region = "us-east-1" -> null
  - tags = {} -> null
  - tags_all = {} -> null
  - vpc_id = "vpc-03c55fc904140637c" ->
null
  # (4 unchanged attributes hidden)
}

# aws_vpc.main will be destroyed
- resource "aws_vpc" "main" {
  - arn = "arn:aws:ec2:us-
east-1:550101108440:vpc/vpc-03c55fc904140637c" -> null
  - assign_generated_ipv6_cidr_block = false -> null
  - cidr_block = "10.0.0.0/16" -> null
  - default_network_acl_id = "acl-0b5102ecdad982d8f" ->
null
  - default_route_table_id = "rtb-02485c57d42735f80" ->
null
  - default_security_group_id = "sg-02134c55268a27f26" ->
null
  - dhcp_options_id = "dopt-0d04c3486383e33c7" ->
null
  - enable_dns_hostnames = false -> null
  - enable_dns_support = true -> null
  - enable_network_address_usage_metrics = false -> null
  - id = "vpc-03c55fc904140637c" -> null
  - instance_tenancy = "default" -> null
  - ipv6_netmask_length = 0 -> null
  - main_route_table_id = "rtb-02485c57d42735f80" ->
null
  - owner_id = "550101108440" -> null
  - region = "us-east-1" -> null

```

```

- tags                = {} -> null
- tags_all            = {} -> null
  # (4 unchanged attributes hidden)
}

```

Plan: 0 to add, 0 to change, 27 to destroy.

Changes to Outputs:

```

- alb_dns_name      = "flask-alb-1803430498.us-
east-1.elb.amazonaws.com" -> null
- ecs_cluster_name = "flask-service-dev" -> null
- running_tasks    = 2 -> null

```

Do you really want to destroy all resources?

Terraform will destroy all your managed infrastructure, as shown above.

There is no undo. Only 'yes' will be accepted to confirm.

Enter a value: yes

```

aws_iam_role_policy_attachment.ecs_task_execution_policy:
Destroying... [id=ecsTaskExecutionRole/arn:aws:iam::aws:policy/
service-role/AmazonECSTaskExecutionRolePolicy]
aws_route.public_internet_access: Destroying... [id=r-
rtb-0d4315bdacdd240ce1080289494]
aws_route_table_association.private[0]: Destroying...
[id=rtbassoc-0f2650cbdf752d539]
aws_route_table_association.public[1]: Destroying...
[id=rtbassoc-0d341595af0a213a4]
aws_route_table_association.private[1]: Destroying...
[id=rtbassoc-0dfc057e362081f99]
aws_route.private_nat_access: Destroying... [id=r-
rtb-0644c490fd39d7c6d1080289494]
aws_route_table_association.public[0]: Destroying...
[id=rtbassoc-02124e61f12298822]
aws_ecs_service.service: Destroying... [id=arn:aws:ecs:us-
east-1:550101108440:service/flask-service-dev/flask-service-
service]
aws_iam_role_policy_attachment.ecs_task_execution_policy:
Destruction complete after 1s
aws_route_table_association.public[0]: Destruction complete after

```

2s

aws_route_table_association.private[1]: Destruction complete after

2s

aws_route_table_association.public[1]: Destruction complete after

2s

aws_route_table_association.private[0]: Destruction complete after

2s

aws_route.public_internet_access: Destruction complete after 2s

aws_route.private_nat_access: Destruction complete after 2s

aws_internet_gateway.igw: Destroying...

[id=igw-0b687ce9c38a957a0]

aws_route_table.public: Destroying... [id=rtb-0d4315bdacdd240ce]

aws_route_table.private: Destroying... [id=rtb-0644c490fd39d7c6d]

aws_nat_gateway.nat: Destroying... [id=nat-05a58cac315d8f7e5]

aws_route_table.public: Destruction complete after 1s

aws_route_table.private: Destruction complete after 1s

aws_ecs_service.service: Still destroying... [id=arn:aws:ecs:us-east-1:550101108440:serv...lask-service-dev/flask-service-service, 00m10s elapsed]

aws_nat_gateway.nat: Still destroying...

[id=nat-05a58cac315d8f7e5, 00m10s elapsed]

aws_internet_gateway.igw: Still destroying...

[id=igw-0b687ce9c38a957a0, 00m10s elapsed]

aws_ecs_service.service: Still destroying... [id=arn:aws:ecs:us-east-1:550101108440:serv...lask-service-dev/flask-service-service, 00m20s elapsed]

aws_internet_gateway.igw: Still destroying...

[id=igw-0b687ce9c38a957a0, 00m20s elapsed]

aws_nat_gateway.nat: Still destroying...

[id=nat-05a58cac315d8f7e5, 00m20s elapsed]

aws_ecs_service.service: Still destroying... [id=arn:aws:ecs:us-east-1:550101108440:serv...lask-service-dev/flask-service-service, 00m30s elapsed]

aws_nat_gateway.nat: Still destroying...

[id=nat-05a58cac315d8f7e5, 00m30s elapsed]

aws_internet_gateway.igw: Still destroying...

[id=igw-0b687ce9c38a957a0, 00m30s elapsed]

aws_nat_gateway.nat: Destruction complete after 33s

aws_eip.nat: Destroying... [id=eipalloc-077f1ed9391ad2db0]

aws_eip.nat: Destruction complete after 2s

aws_ecs_service.service: Still destroying... [id=arn:aws:ecs:us-

east-1:550101108440:serv...lask-service-dev/flask-service-service,
00m40s elapsed]
aws_internet_gateway.igw: Still destroying...
[id=igw-0b687ce9c38a957a0, 00m40s elapsed]
aws_ecs_service.service: Still destroying... [id=arn:aws:ecs:us-
east-1:550101108440:serv...lask-service-dev/flask-service-service,
00m50s elapsed]
aws_internet_gateway.igw: Still destroying...
[id=igw-0b687ce9c38a957a0, 00m50s elapsed]
aws_ecs_service.service: Still destroying... [id=arn:aws:ecs:us-
east-1:550101108440:serv...lask-service-dev/flask-service-service,
01m00s elapsed]
aws_internet_gateway.igw: Still destroying...
[id=igw-0b687ce9c38a957a0, 01m00s elapsed]
aws_ecs_service.service: Still destroying... [id=arn:aws:ecs:us-
east-1:550101108440:serv...lask-service-dev/flask-service-service,
01m10s elapsed]
aws_internet_gateway.igw: Still destroying...
[id=igw-0b687ce9c38a957a0, 01m10s elapsed]
aws_ecs_service.service: Still destroying... [id=arn:aws:ecs:us-
east-1:550101108440:serv...lask-service-dev/flask-service-service,
01m20s elapsed]
aws_internet_gateway.igw: Still destroying...
[id=igw-0b687ce9c38a957a0, 01m20s elapsed]
aws_ecs_service.service: Still destroying... [id=arn:aws:ecs:us-
east-1:550101108440:serv...lask-service-dev/flask-service-service,
01m30s elapsed]
aws_internet_gateway.igw: Still destroying...
[id=igw-0b687ce9c38a957a0, 01m30s elapsed]
aws_ecs_service.service: Still destroying... [id=arn:aws:ecs:us-
east-1:550101108440:serv...lask-service-dev/flask-service-service,
01m40s elapsed]
aws_internet_gateway.igw: Still destroying...
[id=igw-0b687ce9c38a957a0, 01m40s elapsed]
aws_ecs_service.service: Still destroying... [id=arn:aws:ecs:us-
east-1:550101108440:serv...lask-service-dev/flask-service-service,
01m50s elapsed]
aws_internet_gateway.igw: Still destroying...
[id=igw-0b687ce9c38a957a0, 01m50s elapsed]
aws_ecs_service.service: Still destroying... [id=arn:aws:ecs:us-
east-1:550101108440:serv...lask-service-dev/flask-service-service,

02m00s elapsed]
aws_internet_gateway.igw: Still destroying...
[id=igw-0b687ce9c38a957a0, 02m00s elapsed]
aws_ecs_service.service: Still destroying... [id=arn:aws:ecs:us-east-1:550101108440:serv...lask-service-dev/flask-service-service, 02m10s elapsed]
aws_internet_gateway.igw: Still destroying...
[id=igw-0b687ce9c38a957a0, 02m10s elapsed]
aws_ecs_service.service: Still destroying... [id=arn:aws:ecs:us-east-1:550101108440:serv...lask-service-dev/flask-service-service, 02m20s elapsed]
aws_internet_gateway.igw: Still destroying...
[id=igw-0b687ce9c38a957a0, 02m20s elapsed]
aws_ecs_service.service: Still destroying... [id=arn:aws:ecs:us-east-1:550101108440:serv...lask-service-dev/flask-service-service, 02m30s elapsed]
aws_internet_gateway.igw: Still destroying...
[id=igw-0b687ce9c38a957a0, 02m30s elapsed]
aws_ecs_service.service: Still destroying... [id=arn:aws:ecs:us-east-1:550101108440:serv...lask-service-dev/flask-service-service, 02m40s elapsed]
aws_internet_gateway.igw: Still destroying...
[id=igw-0b687ce9c38a957a0, 02m40s elapsed]
aws_ecs_service.service: Still destroying... [id=arn:aws:ecs:us-east-1:550101108440:serv...lask-service-dev/flask-service-service, 02m50s elapsed]
aws_internet_gateway.igw: Still destroying...
[id=igw-0b687ce9c38a957a0, 02m50s elapsed]
aws_ecs_service.service: Still destroying... [id=arn:aws:ecs:us-east-1:550101108440:serv...lask-service-dev/flask-service-service, 03m00s elapsed]
aws_internet_gateway.igw: Still destroying...
[id=igw-0b687ce9c38a957a0, 03m00s elapsed]
aws_ecs_service.service: Still destroying... [id=arn:aws:ecs:us-east-1:550101108440:serv...lask-service-dev/flask-service-service, 03m10s elapsed]
aws_internet_gateway.igw: Still destroying...
[id=igw-0b687ce9c38a957a0, 03m10s elapsed]
aws_ecs_service.service: Still destroying... [id=arn:aws:ecs:us-east-1:550101108440:serv...lask-service-dev/flask-service-service, 03m20s elapsed]

**aws_internet_gateway.igw: Still destroying...
[id=igw-0b687ce9c38a957a0, 03m20s elapsed]**
aws_ecs_service.service: Still destroying... [id=arn:aws:ecs:us-east-1:550101108440:serv...lask-service-dev/flask-service-service, 03m30s elapsed]
**aws_internet_gateway.igw: Still destroying...
[id=igw-0b687ce9c38a957a0, 03m30s elapsed]**
aws_ecs_service.service: Still destroying... [id=arn:aws:ecs:us-east-1:550101108440:serv...lask-service-dev/flask-service-service, 03m40s elapsed]
**aws_internet_gateway.igw: Still destroying...
[id=igw-0b687ce9c38a957a0, 03m40s elapsed]**
aws_ecs_service.service: Still destroying... [id=arn:aws:ecs:us-east-1:550101108440:serv...lask-service-dev/flask-service-service, 03m50s elapsed]
**aws_internet_gateway.igw: Still destroying...
[id=igw-0b687ce9c38a957a0, 03m50s elapsed]**
aws_ecs_service.service: Still destroying... [id=arn:aws:ecs:us-east-1:550101108440:serv...lask-service-dev/flask-service-service, 04m00s elapsed]
**aws_internet_gateway.igw: Still destroying...
[id=igw-0b687ce9c38a957a0, 04m00s elapsed]**
aws_ecs_service.service: Still destroying... [id=arn:aws:ecs:us-east-1:550101108440:serv...lask-service-dev/flask-service-service, 04m10s elapsed]
**aws_internet_gateway.igw: Still destroying...
[id=igw-0b687ce9c38a957a0, 04m10s elapsed]**
aws_ecs_service.service: Still destroying... [id=arn:aws:ecs:us-east-1:550101108440:serv...lask-service-dev/flask-service-service, 04m20s elapsed]
**aws_internet_gateway.igw: Still destroying...
[id=igw-0b687ce9c38a957a0, 04m20s elapsed]**
aws_ecs_service.service: Still destroying... [id=arn:aws:ecs:us-east-1:550101108440:serv...lask-service-dev/flask-service-service, 04m30s elapsed]
**aws_internet_gateway.igw: Still destroying...
[id=igw-0b687ce9c38a957a0, 04m30s elapsed]**
aws_ecs_service.service: Still destroying... [id=arn:aws:ecs:us-east-1:550101108440:serv...lask-service-dev/flask-service-service, 04m40s elapsed]
aws_internet_gateway.igw: Still destroying...

[id=igw-0b687ce9c38a957a0, 04m40s elapsed]
aws_ecs_service.service: Still destroying... [id=arn:aws:ecs:us-east-1:550101108440:serv...lask-service-dev/flask-service-service, 04m50s elapsed]
aws_internet_gateway.igw: Still destroying...
[id=igw-0b687ce9c38a957a0, 04m50s elapsed]
aws_ecs_service.service: Still destroying... [id=arn:aws:ecs:us-east-1:550101108440:serv...lask-service-dev/flask-service-service, 05m00s elapsed]
aws_internet_gateway.igw: Still destroying...
[id=igw-0b687ce9c38a957a0, 05m00s elapsed]
aws_ecs_service.service: Still destroying... [id=arn:aws:ecs:us-east-1:550101108440:serv...lask-service-dev/flask-service-service, 05m10s elapsed]
aws_internet_gateway.igw: Still destroying...
[id=igw-0b687ce9c38a957a0, 05m10s elapsed]
aws_ecs_service.service: Still destroying... [id=arn:aws:ecs:us-east-1:550101108440:serv...lask-service-dev/flask-service-service, 05m20s elapsed]
aws_internet_gateway.igw: Still destroying...
[id=igw-0b687ce9c38a957a0, 05m20s elapsed]
aws_ecs_service.service: Still destroying... [id=arn:aws:ecs:us-east-1:550101108440:serv...lask-service-dev/flask-service-service, 05m30s elapsed]
aws_internet_gateway.igw: Still destroying...
[id=igw-0b687ce9c38a957a0, 05m30s elapsed]
aws_ecs_service.service: Still destroying... [id=arn:aws:ecs:us-east-1:550101108440:serv...lask-service-dev/flask-service-service, 05m40s elapsed]
aws_internet_gateway.igw: Still destroying...
[id=igw-0b687ce9c38a957a0, 05m40s elapsed]
aws_ecs_service.service: Still destroying... [id=arn:aws:ecs:us-east-1:550101108440:serv...lask-service-dev/flask-service-service, 05m50s elapsed]
aws_internet_gateway.igw: Still destroying...
[id=igw-0b687ce9c38a957a0, 05m50s elapsed]
aws_ecs_service.service: Still destroying... [id=arn:aws:ecs:us-east-1:550101108440:serv...lask-service-dev/flask-service-service, 06m00s elapsed]
aws_internet_gateway.igw: Still destroying...
[id=igw-0b687ce9c38a957a0, 06m00s elapsed]

aws_ecs_service.service: Still destroying... [id=arn:aws:ecs:us-east-1:550101108440:serv...lask-service-dev/flask-service-service, 06m10s elapsed]

aws_internet_gateway.igw: Still destroying... [id=igw-0b687ce9c38a957a0, 06m10s elapsed]

aws_ecs_service.service: Still destroying... [id=arn:aws:ecs:us-east-1:550101108440:serv...lask-service-dev/flask-service-service, 06m20s elapsed]

aws_internet_gateway.igw: Still destroying... [id=igw-0b687ce9c38a957a0, 06m20s elapsed]

aws_ecs_service.service: Still destroying... [id=arn:aws:ecs:us-east-1:550101108440:serv...lask-service-dev/flask-service-service, 06m30s elapsed]

aws_internet_gateway.igw: Still destroying... [id=igw-0b687ce9c38a957a0, 06m30s elapsed]

aws_ecs_service.service: Still destroying... [id=arn:aws:ecs:us-east-1:550101108440:serv...lask-service-dev/flask-service-service, 06m40s elapsed]

aws_internet_gateway.igw: Still destroying... [id=igw-0b687ce9c38a957a0, 06m40s elapsed]

aws_ecs_service.service: Still destroying... [id=arn:aws:ecs:us-east-1:550101108440:serv...lask-service-dev/flask-service-service, 06m50s elapsed]

aws_internet_gateway.igw: Still destroying... [id=igw-0b687ce9c38a957a0, 06m50s elapsed]

aws_ecs_service.service: Still destroying... [id=arn:aws:ecs:us-east-1:550101108440:serv...lask-service-dev/flask-service-service, 07m00s elapsed]

aws_internet_gateway.igw: Still destroying... [id=igw-0b687ce9c38a957a0, 07m00s elapsed]

aws_ecs_service.service: Still destroying... [id=arn:aws:ecs:us-east-1:550101108440:serv...lask-service-dev/flask-service-service, 07m10s elapsed]

aws_internet_gateway.igw: Still destroying... [id=igw-0b687ce9c38a957a0, 07m10s elapsed]

aws_ecs_service.service: Destruction complete after 7m16s

aws_subnet.private[0]: Destroying... [id=subnet-084d5c30067102c75]

aws_ecs_cluster.main: Destroying... [id=arn:aws:ecs:us-east-1:550101108440:cluster/flask-service-dev]

aws_lb_listener.http: Destroying...

[id=arn:aws:elasticloadbalancing:us-east-1:550101108440:listener/app/flask-alb/6923f38c0b839406/b8c81681259aed28]
aws_security_group.ecs_sg: Destroying...
[id=sg-08000d7164421d98f]
aws_subnet.private[1]: Destroying...
[id=subnet-082b24bc961d63241]
aws_ecs_task_definition.task: Destroying... [id=flask-task]
aws_ecs_task_definition.task: Destruction complete after 1s
aws_cloudwatch_log_group.ecs_logs: Destroying... [id=/ecs/flask-service-dev]
aws_iam_role.ecs_task_execution: Destroying...
[id=ecsTaskExecutionRole]
aws_ecs_cluster.main: Destruction complete after 1s
aws_lb_listener.http: Destruction complete after 1s
aws_lb.app_alb: Destroying... [id=arn:aws:elasticloadbalancing:us-east-1:550101108440:loadbalancer/app/flask-alb/6923f38c0b839406]
aws_lb_target_group.app_tg: Destroying...
[id=arn:aws:elasticloadbalancing:us-east-1:550101108440:targetgroup/flask-service-tg/c6b5b66a04cf4b42]
aws_lb_target_group.app_tg: Destruction complete after 1s
aws_cloudwatch_log_group.ecs_logs: Destruction complete after 1s
aws_subnet.private[0]: Destruction complete after 2s
aws_subnet.private[1]: Destruction complete after 2s
aws_security_group.ecs_sg: Destruction complete after 2s
aws_iam_role.ecs_task_execution: Destruction complete after 2s
aws_internet_gateway.igw: Still destroying...
[id=igw-0b687ce9c38a957a0, 07m20s elapsed]
aws_lb.app_alb: Still destroying...
[id=arn:aws:elasticloadbalancing:us-east-1:...alancer/app/flask-alb/6923f38c0b839406, 00m10s elapsed]
aws_internet_gateway.igw: Still destroying...
[id=igw-0b687ce9c38a957a0, 07m30s elapsed]
aws_internet_gateway.igw: Destruction complete after 7m31s
aws_lb.app_alb: Destruction complete after 19s
aws_subnet.public[1]: Destroying...
[id=subnet-0cc6b8aa68c55d73c]
aws_subnet.public[0]: Destroying... [id=subnet-09d4602e9f6aef303]
aws_security_group.alb_sg: Destroying...
[id=sg-01d0854f620d31526]

aws_subnet.public[0]: Destruction complete after 1s
aws_subnet.public[1]: Destruction complete after 2s
aws_security_group.alb_sg: Destruction complete after 2s
aws_vpc.main: Destroying... [id=vpc-03c55fc904140637c]
aws_vpc.main: Destruction complete after 1s

Destroy complete! Resources: 27 destroyed.
mahesh.singh@del1-lmc-n82481 capstone-aws %