

# ECS

## → SETUP A HIGHLY AVAILABLE ECS CLUSTER WITH LOAD BALANCER AND DYNAMIC PORT MAPPING.

- NOTE:

Your cluster should maintain atleast 6 tasks at any point of time and should be highly available across multiple AZ's.  
Use the below image from deployment:  
sabair0509/hiring-app:works

### 1. PREREQUISITES:

- AWS CLI INSTALLED AND CONFIGURED.
- DOCKER INSTALLED.
- ECS CLI INSTALLED.
- A VPC (VIRTUAL PRIVATE CLOUD) WITH MULTIPLE SUBNETS ACROSS DIFFERENT AZS

## → CREATE A VPC WITH MULTIPLE SUBNETS AND INTERNET GATEWAY ATTACHED TO IT

### VPC settings

**Resources to create** [Info](#)

Create only the VPC resource or the VPC and other networking resources.

☐ VPC only ☒ VPC and more

**Name tag auto-generation** [Info](#)

Enter a value for the Name tag. This value will be used to auto-generate Name tags for all resources in the VPC.

☒ Auto-generate

ecs-cluster

**IPv4 CIDR block** [Info](#)

Determine the starting IP and the size of your VPC using CIDR notation.

182.168.0.0/16 65,536 IPs

CIDR block size must be between /16 and /28.

**IPv6 CIDR block** [Info](#)

☒ No IPv6 CIDR block

☐ Amazon-provided IPv6 CIDR block

**Tenancy** [Info](#)

Default

**Number of Availability Zones (AZs)** [Info](#)

Choose the number of AZs in which to provision subnets. We recommend at least two AZs for high availability.

1 2 3

**Number of public subnets** [Info](#)

The number of public subnets to add to your VPC. Use public subnets for web applications that need to be publicly accessible over the internet.

0 3

**Number of private subnets** [Info](#)

The number of private subnets to add to your VPC. Use private subnets to secure backend resources that don't need public access.

0 3 6

► **Customize subnets CIDR blocks**

**NAT gateways (\$)** [Info](#)

Choose the number of Availability Zones (AZs) in which to create NAT gateways. Note that there is a charge for each NAT gateway.

None In 1 AZ 1 per AZ

**VPC endpoints** [Info](#)

Endpoints can help reduce NAT gateway charges and improve security by accessing S3 directly from the VPC. By default, full access policy is used. You can customize this policy at any time.

None S3 Gateway

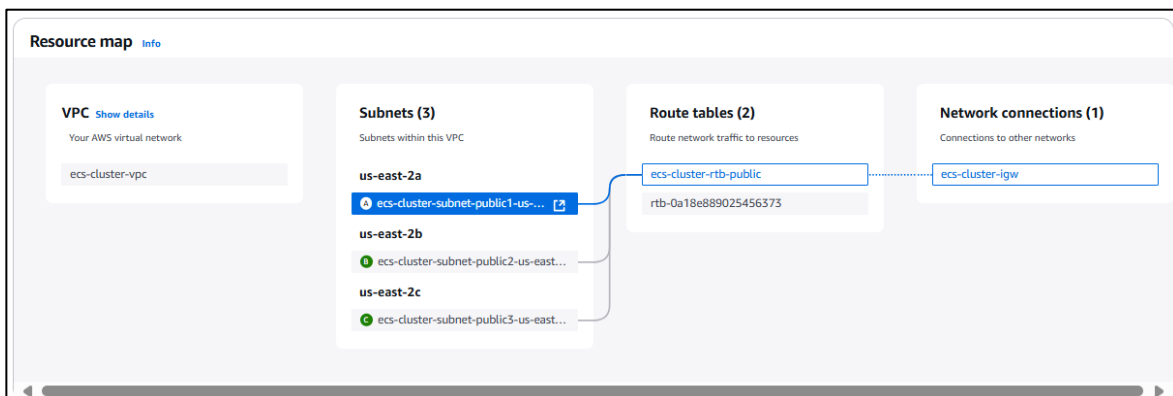
**DNS options** [Info](#)

☒ Enable DNS hostnames

☒ Enable DNS resolution

► **Additional tags**

Cancel Preview code Create VPC



## → CREATE A ROUTE TABLE AND ATTACHED INTERNET GATEWAY ALONG WITH SUBNETS

Route tables



Create route table

rtb-0d8dff60c6b19a86 / ecs-rt

Actions

**Details** Info

Route table ID  
rtb-0d8dff60c6b19a86

VPC  
vpc-0d761b75e5f5660a9 | ecs-cluster-vpc

Main  
No

Owner ID  
390403872402

Explicit subnet associations  
3 subnets

Edge associations  
-

Routes Subnet associations Edge associations Route propagation Tags

**Routes (2)**

Both Edit routes

Filter routes

Destination	Target	Status	Propagated
0.0.0.0/0	igw-0d7a05dc6e6e4587e	Active	No
182.168.0.0/16	local	Active	No

rtb-0d8dff60c6b19a86 / ecs-rt

Actions

**Details** Info

Route table ID  
rtb-0d8dff60c6b19a86

VPC  
vpc-0d761b75e5f5660a9 | ecs-cluster-vpc

Main  
No

Owner ID  
390403872402

Explicit subnet associations  
3 subnets

Edge associations  
-

Routes Subnet associations Edge associations Route propagation Tags

**Explicit subnet associations (3)**

Edit subnet associations

Find subnet association

Name	Subnet ID	IPv4 CIDR	IPv6 CIDR
ecs-cluster-subnet-public2-us-east-2b	subnet-0c761fa0f118dc7bf	182.168.16.0/20	-
ecs-cluster-subnet-public1-us-east-2a	subnet-07ec9ff067531245f	182.168.0.0/20	-
ecs-cluster-subnet-public3-us-east-2c	subnet-0715788edd70b706d	182.168.32.0/20	-

## → CREATE SECURITY GROUP AND ADD ROUTES IN SG FOR PORT 8080 AND 22

Security group (sg-0d98d6669d2193c61 | ecs-sg) was created successfully

Details

sg-0d98d6669d2193c61 - ecs-sg

Actions

**Details**

Security group name  
ecs-sg

Owner  
390403872402

Security group ID  
sg-0d98d6669d2193c61

Inbound rules count  
2 Permission entries

Description  
for ecs high available cluster

Outbound rules count  
3 Permission entries

VPC ID  
vpc-0d761b75e5f5660a9

Inbound rules Outbound rules Sharing - new VPC associations - new Tags

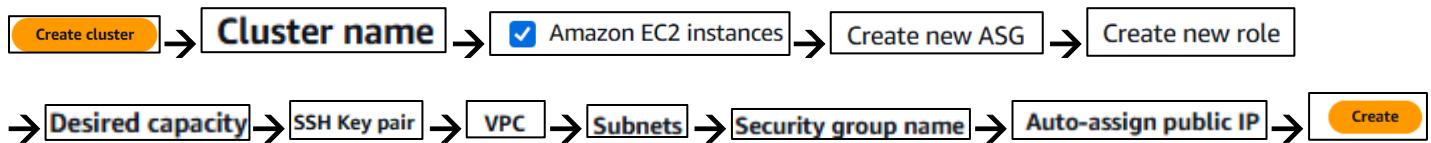
**Inbound rules (2)**

Manage tags Edit inbound rules

Search

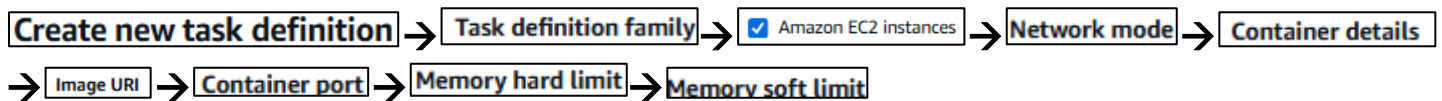
	Name	Security group rule ID	IP version	Type	Protocol	Port ra...	Source	Description
<input type="checkbox"/>	-	sg-0f3c1ff7a355afcf1	IPv4	SSH	TCP	22	0.0.0.0/0	-
<input type="checkbox"/>	-	sg-0748a9c78c6490715	IPv4	Custom TCP	TCP	8080	0.0.0.0/0	-

## → CREATE A ECS CLUSTER



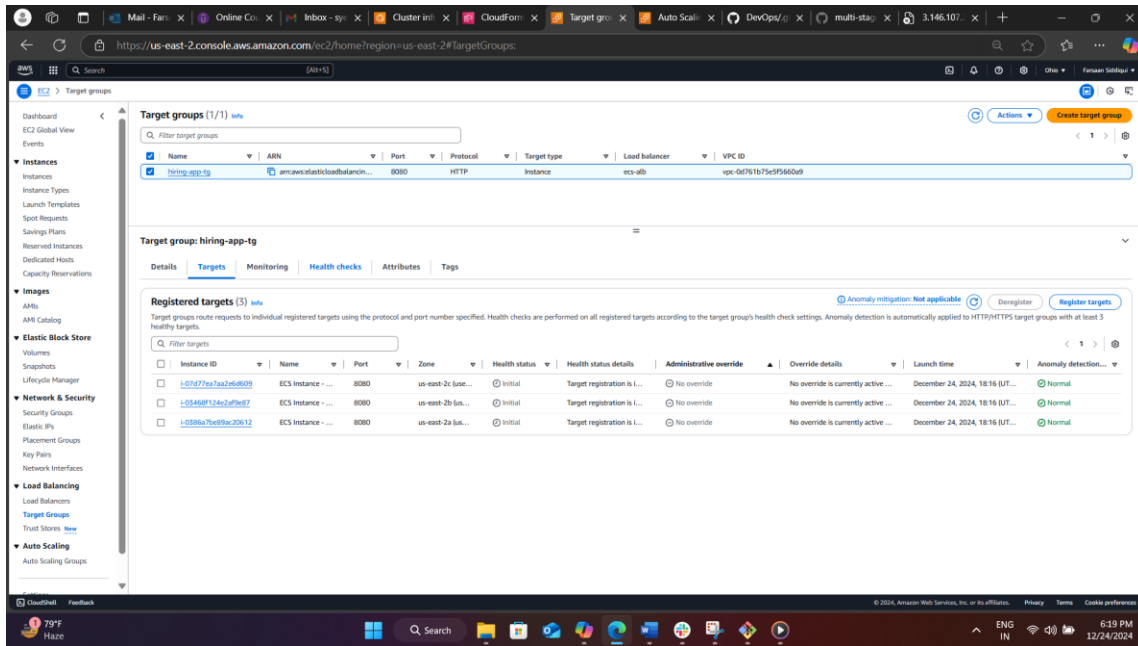
The screenshot shows the AWS ECS console interface. The left sidebar contains navigation links for Clusters, Namespaces, Task definitions, Account settings, Install AWS Copilot, Amazon ECR, Repositories, AWS Batch, Documentation, Discover products, and Subscriptions. The main content area is titled 'Cluster overview' for 'cluster-dev'. It displays the cluster's ARN, status (Active), services (Draining: Active), encryption (Managed storage: Fargate ephemeral storage), and CloudWatch monitoring (Default). Below this, the 'Infrastructure' tab is selected, showing 'Capacity providers (3)' and 'Container instances (3)'. The capacity providers table lists FARGATE, FARGATE\_SPOT, and Infra-ECS-Cluster-dev-1. The container instances table lists three active instances with their IDs, capacity providers, availability zones, and resource usage.

## → CREATE A TASK DEFINITION (basically this is blueprint which will be used to create task and services)

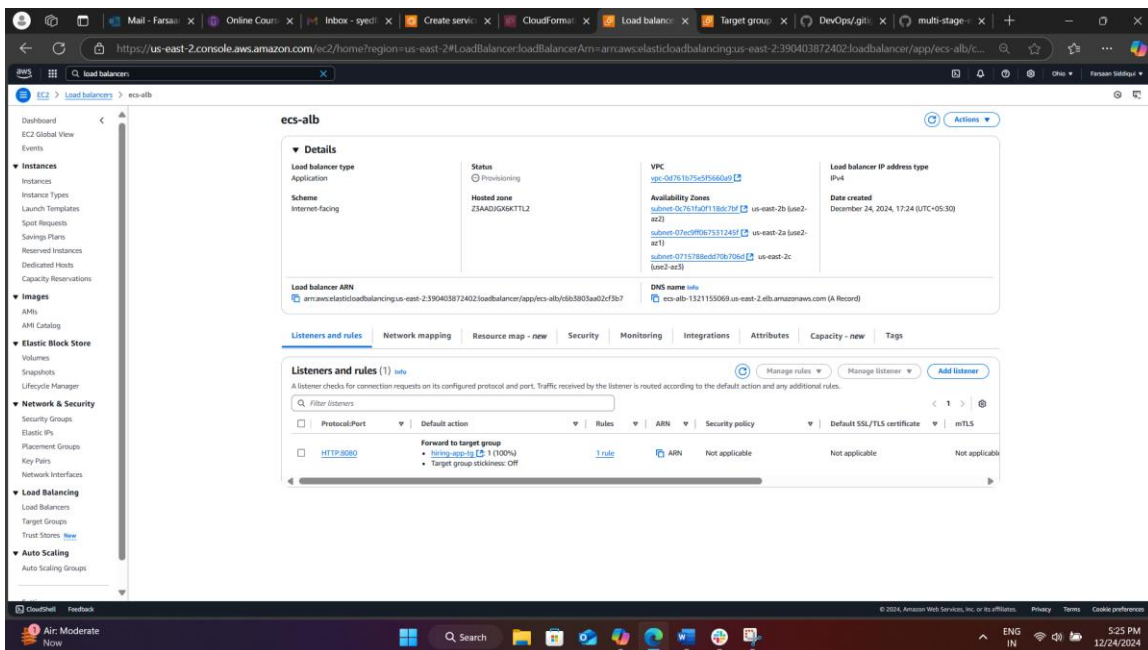
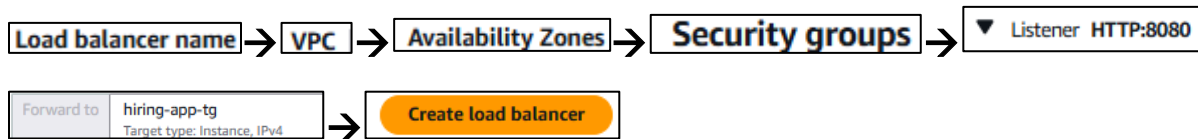


The screenshot shows the 'Task definition successfully created' notification for 'hiring-app-prod:1'. The task definition details are displayed, including its ARN, status (ACTIVE), task role, fault injection (turned off), time created, app environment (EC2), operating system/architecture (Linux/x86\_64), and network mode (bridge). The 'Containers' tab is selected, showing the 'Task size' section with CPU and memory allocation bars. The 'Containers' table lists the 'Hiring-app' container with its image URI, private registry status, essential status, CPU, memory limits, and GPU requirements.

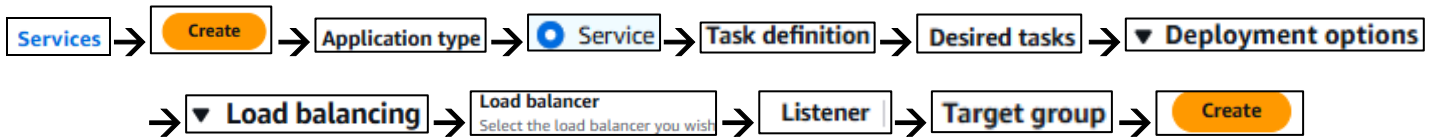
## → CREATE TARGET GROUP AND ADD ALL THE CREATED EC2 IN THAT GROUP



## → CREATE A APPLICATION LOAD BALANCER AND SET THE TARGET EC2 WHICH IS CREATED FOR ECS



## → CREATE A SERVICE IN CLUSTER



**\*NOTE\***

**MAKE SURE YOU ADD CORRECT VPC AND CORRECT INSTANCES IN TARGET GROUPS AND ALSO IN SECURITY GROUP  
THE REQUIRED PORTS SSH AND HTTP ARE ALLOWED**

The screenshot shows the AWS Management Console for the 'application' service in the 'cluster-dev' cluster. The service is in an 'Active' state. The 'Tasks' tab is selected, showing a list of tasks with columns for Task, Last status, Desired status, Task definition, Health status, Started by, Started at, Container instance, Launch type, and Platform version. The first task is '034cb469c6d6431cbb09fedb6dceacf5' with a status of 'Running'. Below the tasks, the 'Containers' tab is selected, showing a list of containers with columns for Container name, Container runtime ID, Image URI, Image Digest, Status, Health status, CPU, and Memory hard/soft limit. The first container is 'Hiring-app' with a status of 'Running'.

## → CHECK ON BROWSER WITH /hiring FOLLOWED LOAD BALANCER URL

**DNS name Info**  
**my-alb-48577490.us-east-2.elb.amazonaws.com (A Record)**

The screenshot shows a web browser with the URL 'my-alb-48577490.us-east-2.elb.amazonaws.com/hiring/'. The page displays a green message: 'Welcome to Techie Horizon'.