

Demo 1: Task Definition Advanced

Introduction

Following are the learning objectives of this demonstration:

- Set ECS task environment variables
- Attach IAM roles to tasks
- Set task placement strategies in the service definition

Note: Create an ECS cluster in EC2 mode with two instances

Environment Variables

- Go to the ECS Task definition.
- Create a PSQL Task by mentioning Username and password through Key: Value
- During the demo, SME used Password from AWS Secrets.

Environment Files

Source

Location

S3 ARN ▾

s3 <ARN>|

✕

+

Environment variables

You may also designate AWS Systems Manager Parameter Store keys or ARNs using the 'valueFrom' field. ECS will inject the value into containers at run-time.

Key

VAR

Value ▾

HELLO LEARNERS

✕

VAR_SECRET

ValueF... ▾

arn:aws:secretsmanager:region:a

✕

Add key

Value ▾

Add value

- Connect to Psql through Username and Password

IAM Roles

SSH to any of the EC2 instance and enter inside the container by **docker exec** command

- Try running aws s3 command from the container.
- After failing, Attach an IAM role to (task)container providing s3 access.

Create new Task Definition

[Step 1: Select launch type compatibility](#)

Step 2: Configure task and container definitions

Configure task and container definitions

A task definition specifies which containers are included in your task and how they interact with each other. You can also specify data volumes for your containers to use. [Learn more](#)

Task Definition Name* ⓘ

Requires Compatibilities* EC2

Task Role ⓘ

Optional IAM role that tasks can use to make API requests to authorized AWS services. Create an Amazon Elastic Container Service Task Role in the [IAM Console](#)

Network Mode ⓘ

If you choose <default>, ECS will start your container using Docker's default networking mode, which is Bridge on Linux and NAT on Windows. <default> is the only supported mode on Windows.

- Try command again

Placement Constraints

- Create a Service in EC2 Mode
- Deploy Vote Service with AZ BinPack

☐ Blue/green deployment (powered by AWS CodeDeploy) ⓘ

This sets AWS CodeDeploy as the deployment controller for the service. A CodeDeploy application and deployment group are created automatically with [default settings](#) for the service. To change to the rolling update deployment type after the service has been created, you must re-create the service and select the "rolling update" deployment type.

Task Placement

Lets you customize how tasks are placed on instances within your cluster. Different placement strategies are available to optimize for availability and efficiency.

Placement Templates

AZ Balanced Spread

AZ Balanced Spread

AZ Balanced BinPack

BinPack

One Task Per Host

Custom

Edit

availability zone

Task tagging configuration

☒ Enable ECS managed tags ⓘ

Propagate tags from

Do not propagate

ⓘ

Tags

Key

Value

Add key

Add value

*Required

Cancel

Next step

- Deploy Worker Service with AZ Balancer
- Deploy Result Service with One Task per EC2
- Understand how the placement of tasks happened just because of placement strategies

Demo 2: Service Definition Advanced

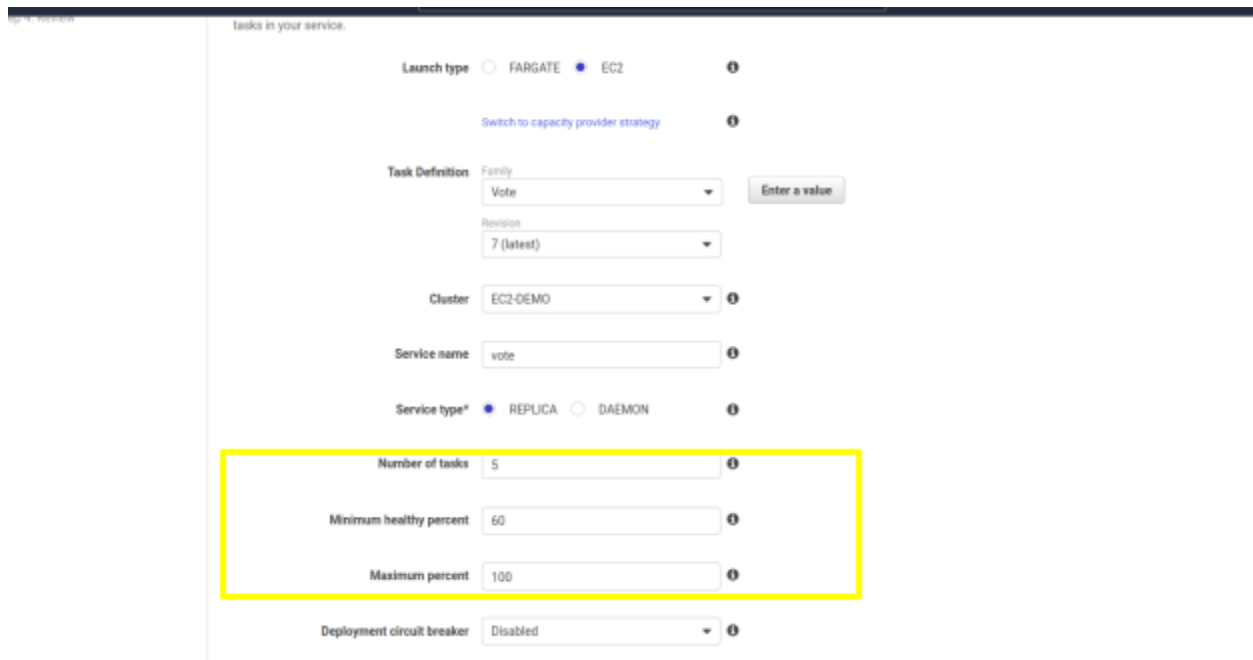
Introduction

Following are the learning objectives of this demonstration:

- Deployment using rolling updates

ECS Service Advanced

- Rolling Updates
 - Set the number of replica of vote service 5
 - MinHealthPercentage: 60
 - MaxHealthPercentage: 150



The screenshot shows the 'Tasks in your service' configuration page in the AWS ECS console. The configuration includes:

- Launch type:** EC2 (selected), FARGATE
- Task Definition:** Family: Vote, Revision: 7 (latest), Cluster: EC2-DEMO
- Service name:** vote
- Service type:** REPLICAS (selected), DAEMON
- Number of tasks:** 5
- Minimum healthy percent:** 60
- Maximum percent:** 100
- Deployment circuit breaker:** Disabled

A yellow box highlights the 'Number of tasks', 'Minimum healthy percent', and 'Maximum percent' fields.

- Change the version of the image in the task definition
- Update the Service and see the deployment happening without affecting end-user

Demo 3: Setup docker voting application on ECS cluster in ECS mode

Introduction

Following are the learning objectives of this demonstration:

- Create a production-ready infrastructure on AWS
- Deploy Docker voting application on EC2 based ECS cluster

Use following docker images:

Vote: dipesh017/demo:vote

Redis: redis:5.0-alpine3.10

Worker: dipesh017/demo:worker

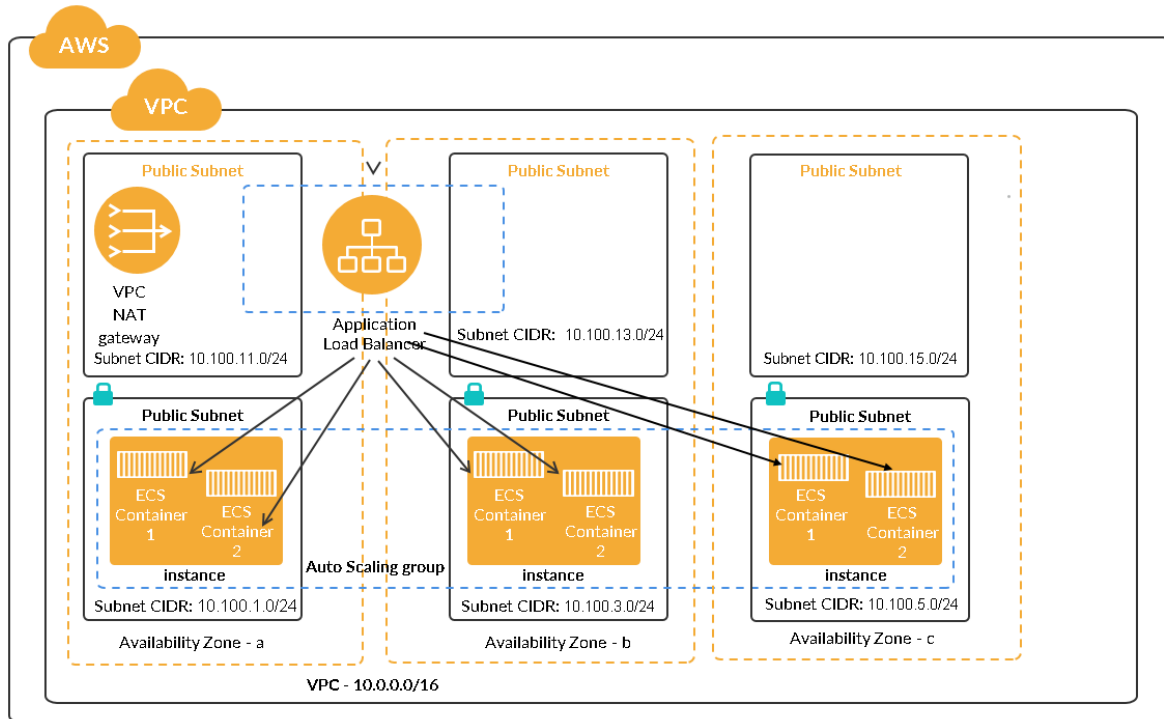
Db: postgres:9.4

POSTGRES_USER : postgres

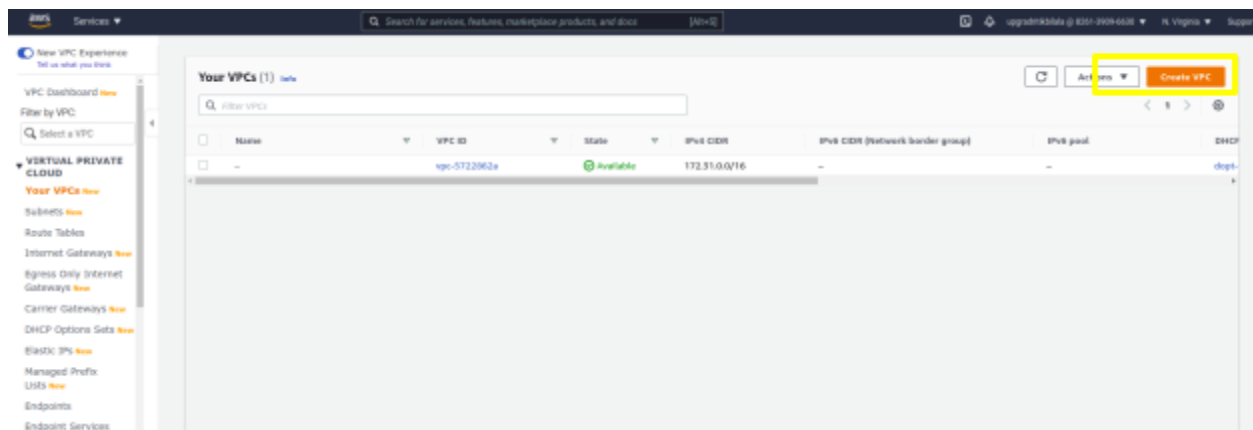
POSTGRES_PASSWORD : postgres

Result: dipesh017/demo:result

Let us set up a custom VPC with 3 private and 3 public subnets. Create a NAT gateway and an Internet gateway in public subnets. Launch ECS cluster in private subnets.



1. Go to VPC main page of AWS console
2. Click on create VPC



3. Create VPC of range 10.100.0.0/16

A VPC is an isolated portion of the AWS cloud populated by AWS objects, such as Amazon EC2 instances.

VPC settings

Name tag - optional
Creates a tag with a key of 'Name' and a value that you specify.

upgrad-demo

IPv4 CIDR block [Info](#)

10.100.0.0/16

IPv6 CIDR block [Info](#)

☒ No IPv6 CIDR block

☐ Amazon-provided IPv6 CIDR block

☐ IPv6 CIDR owned by me

Tenancy [Info](#)

Default

Tags

4. Create three public and 3 private subnets in new VPC

Subnets (5) [Info](#)

Filter by subnets

	Name	Subnet ID	State	VPC	IPv4 CIDR	IPv6 CIDR	Available IPv4 addresses
	-	subnet-d1ae568c	Available	vpc-5722882a	172.31.66.0/20	-	4096
	-	subnet-9d7089b5	Available	vpc-5722882a	172.31.0.0/20	-	4096
	-	subnet-46fa2177	Available	vpc-5722882a	172.31.48.0/20	-	4096
	-	subnet-ea0f5ae7	Available	vpc-5722882a	172.31.16.0/20	-	4096
	-	subnet-d47b4803	Available	vpc-5722882a	172.31.64.0/20	-	4096
	-	subnet-fb4347e4	Available	vpc-5722882a	172.31.32.0/20	-	4096

Select a subnet

- Public: 10.100.1.0/24 10.100.3.0/24 10.100.5.0/24

ii. Private: 10.100.11.0/24 10.100.13.0/24 10.100.15.0/24

Subnets (6) [Info](#) Refresh Actions Create subnet

< 1 > Settings

search: upgrad Clear filters

<input type="checkbox"/>	Name	Subnet ID	State	VPC	IPv4 CIDR
<input type="checkbox"/>	upgrad-private3	subnet-071e523e434b4d50d	Available	vpc-073cc7fd8379b34b3 upgrad-demo	10.100.11.0/24
<input type="checkbox"/>	upgrad-private1	subnet-044c7bb79e6879c54	Available	vpc-073cc7fd8379b34b3 upgrad-demo	10.100.13.0/24
<input type="checkbox"/>	upgrad-public3	subnet-0ae9036852a38d1cf	Available	vpc-073cc7fd8379b34b3 upgrad-demo	10.100.15.0/24
<input type="checkbox"/>	upgrad-public1	subnet-06b8bac8de60b5b38	Available	vpc-073cc7fd8379b34b3 upgrad-demo	10.100.17.0/24
<input type="checkbox"/>	upgrad-public2	subnet-06dad0ff3f063bf1a	Available	vpc-073cc7fd8379b34b3 upgrad-demo	10.100.19.0/24

Select a subnet

b. Create a Nat Gateway and Internet Gateway in one of the public subnet

✓ NAT gateway nat-027989b69a0568333 | upgrad-demo was created successfully. ✕

[VPC](#) > [NAT gateways](#) > nat-027989b69a0568333

nat-027989b69a0568333 / upgrad-demo Delete

Details [Info](#)

NAT gateway ID nat-027989b69a0568333	State Pending	State message Info -	Elastic IP address -
Private IP address -	Network Interface ID -	VPC vpc-073cc7fd8379b34b3 / upgrad-demo	Subnet subnet-06b8bac8de60b5b38 / upgrad-public1
Created 2021/03/21 22:26 GMT+5:30	Deleted -		

[Monitoring](#) [Tags](#)

VPC > Internet gateways > igw-04134eda070880264

igw-04134eda070880264 / upgrad-demo

Actions ▼

Details Info

Internet gateway ID igw-04134eda070880264	State Detached	VPC ID -	Owner 826139096630
--	-------------------	-------------	-----------------------

Tags


Manage tags

Search tags

< 1 > ⚙

Key	Value
Name	upgrad-demo

c. Attach IGW to VPC


 Internet gateway igw-04134eda070880264 successfully attached to vpc-073cc7fd8379b34b3
 ✕

VPC > Internet gateways > igw-04134eda070880264

igw-04134eda070880264 / upgrad-demo

Actions ▼

Details Info

Internet gateway ID igw-04134eda070880264	State Attached	VPC ID vpc-073cc7fd8379b34b3 upgrad-demo	Owner 826139096630
--	-------------------	---	-----------------------

Tags

Manage tags

Search tags

< 1 > ⚙

Key	Value
-----	-------

d. Map 0.0.0.0/0 route to the NAT Gateway in the Route tables of private subnets.

Private route tables::

Route table: rtb-0cc088146f5ec1e8c / upgrad-private		Edit route table association
Routes (2) <input type="text" value="Filter routes"/>		
Destination	Target	
10.100.0.0/16	local	
0.0.0.0/0	nat-0db5bbd61f8c31143	

Public route tables:

Route table: rtb-00370eed531824716 / upgrad-public		Edit route table association
Routes (2) <input type="text" value="Filter routes"/>		
Destination	Target	
10.100.0.0/16	local	
0.0.0.0/0	lgw-04134eda070880264	

5. ECS Cluster

- Create an EC2 mode ECS cluster using the above VPC. Launch two or three EC2 instances with type t2.medium

Networking

Configure the VPC for your container instances to use. A VPC is an isolated portion of the AWS cloud populated by AWS objects, such as Amazon EC2 instances. You can choose an existing VPC, or create a new one with this wizard.

VPC

vpc-073cc7fd8379b34b...

Check the structure for vpc-073cc7fd8379b34b3 in the Amazon EC2 console.

Subnets

subnet-044c7bb79e6879c54

(10.100.11.0/24) | upgrad-private1 - us-east-1a

assign ipv6 on creation: Disabled

subnet-071e523e434b4d50d

(10.100.15.0/24) | upgrad-private3 - us-east-1a

assign ipv6 on creation: Disabled

subnet-0156d95bdac6e42d8

(10.100.13.0/24) | upgrad-private2 - us-east-1b

assign ipv6 on creation: Disabled

Select a subnet...

Auto assign public IP

Use subnet setting

aws

Services

Search for services, features, marketplace products, and docs

[Alt+S]

upgradtrikbilala @ 8261-3909-6630

N. Virginia

Support

New ECS Experience

Tell us what you think

Amazon ECS

Clusters

Task Definitions

Account Settings

Amazon EKS

Clusters

Amazon ECR

Repositories

AWS Marketplace

Discover software

Subscriptions

Clusters > upgrad-demo

Cluster : upgrad-demo

Update Cluster

Delete Cluster

Get a detailed view of the resources on your cluster.

Cluster ARN

arn:aws:ecs:us-east-1:826139096630:cluster/upgrad-demo

Status

ACTIVE

Registered container instances

1

Pending tasks count

0 Fargate, 0 EC2

Running tasks count

0 Fargate, 0 EC2

Active service count

0 Fargate, 0 EC2

Draining service count

0 Fargate, 0 EC2

Services

Tasks

ECS Instances

Metrics

Scheduled Tasks

Tags

Capacity Providers

Create

Update

Delete

Actions

Last updated on March 21, 2021 10:48:21 PM (0m ago)

Filter in this page

Launch type ALL

Service type ALL

b. Create a capacity provider group of 4 EC2 Machines in Private subnets

ServicesTasksECS InstancesMetricsScheduled TasksTagsCapacity Providers

CreateUpdateDelete

Last updated on March 21, 2021 10:53:42 PM (0m ago)

Filter in this page

<input type="checkbox"/>	Capacity ...	Type	ASG	Managed...	Managed...	Current ...	Desired ...	Min Size ...	Max Size...	Update S...
<input type="checkbox"/>	upgrad-de...	ASGProvi...	EC2Cont...	Yes	No	2	2	2	2	

c. Create the same 5 task definitions similar to as done for Fargate mode in the last session (using EC2 mode this time)

d. Launch task using services in EC2 Machines

Services

Tasks

ECS Instances

Metrics

Scheduled Tasks

Tags

Capacity Providers

Run new Task

Stop

Stop All

Actions

Last updated on March 22, 2021 3:11:24 PM (0m ago)

Desired task status:

Running

Stopped

Filter in this page

Launch type

ALL

< 1-5 > Page size 50

<input type="checkbox"/>	Task	Task defi...	Containe...	Last stat...	Desired ...	Started a...	Started ...	Group	Launch t...	Platform ...
<input type="checkbox"/>	1a1d1c6a...	worker:4	d221507d...	RUNNING	RUNNING	2021-03-2...	ecs-svc/9...	service:w...	EC2	--
<input type="checkbox"/>	a9b6a1df...	result:6	293d12bf...	RUNNING	RUNNING	2021-03-2...	ecs-svc/1...	service:re...	EC2	--
<input type="checkbox"/>	d3ce66ce...	vote:8	3f6f748f2...	RUNNING	RUNNING	2021-03-2...	ecs-svc/3...	service:vote	EC2	--
<input type="checkbox"/>	d77f8e2b...	redis:5	4c5e4f75...	RUNNING	RUNNING	2021-03-2...	ecs-svc/1...	service:re...	EC2	--
<input type="checkbox"/>	f78b21fd5...	postgres:3	d221507d...	RUNNING	RUNNING	2021-03-2...	ecs-svc/2...	service:db	EC2	--

6. (Application Load Balancer) ALB Routing:

a. Create an ALB sitting in any two public subnets and attach two target groups, **vote**, and **result**. Ensure that result and vote service get registered with ALB

EC2 > Target groups > ecs

Target groups (2/2) Info

Search or filter target groups

search: ecs X Clear filters

<input checked="" type="checkbox"/>	Name	ARN	Port	Protocol	Target type
<input checked="" type="checkbox"/>	ecs-result	arn:aws:elasticloadbalancin...	80	HTTP	IP
<input checked="" type="checkbox"/>	ecs-vote	arn:aws:elasticloadbalancin...	80	HTTP	IP

aws Services Search for services, features, marketplace products [Alt+S]

upgrad-admin-ss0-access/v.dipesh.garg@upgrad.com @ acadview N. Virginia Support

demo-voting | HTTPS:443

To edit, select a mode above.

Rule limits for condition values, wildcards, and total rules.

1	arn...86337	IF	THEN
		✓ Host is result.upgrad.dev	Forward to ecs-result: 1 (100%) Group-level stickiness: Off
2	arn...a881a	IF	THEN
		✓ Host is vote.upgrad.dev	Forward to ecs-vote: 1 (100%) Group-level stickiness: Off
last	HTTPS 443: default action <i>This rule cannot be moved or deleted</i>	IF	THEN
		✓ Requests otherwise not routed	Forward to ecs-vote: 1 (100%) Group-level stickiness: Off

- One for Vote app and Second for Result app
- Host upgradvote.com path to the vote target group
- Host upgradresult.com path to the result target group
- As the domain is not registered, we have to resolve it locally, make this entry in the /etc/hosts file to resolve DNS at your pc.

```
127.0.0.1      localhost
127.0.1.1      ritik-HP-Notebook
192.168.0.100  master
192.168.0.108  client
3.82.44.122    upgradresult.com
3.82.44.122    upgradvoting.com

# The following lines are desirable for IPv6 capable hosts
::1          ip6-localhost ip6-loopback
fe00::0      ip6-localnet
ff00::0      ip6-mcastprefix
ff02::1      ip6-allnodes
ff02::2      ip6-allrouters
```

f. Show the outputs on Browser.

Route 53 > Hosted zones > upgrad.dev

Records (313) | DNSSEC signing | Hosted zone tags (4)

Records (313) Info

Automatic mode is the current search behavior optimized for best filter results. To change modes go to settings.

Type ▼
Routing policy ▼
Alias ▼
1 match < 1 >

X

<input type="checkbox"/>	Record name ▼	Type ▼	Routin... ▼	Differ... ▼	Value/Route traffic to ▼
<input type="checkbox"/>	vote.upgrad.dev	CNAME	Simple	-	demo-voting-2130952745.us-east-1.elb.amazonaws.com

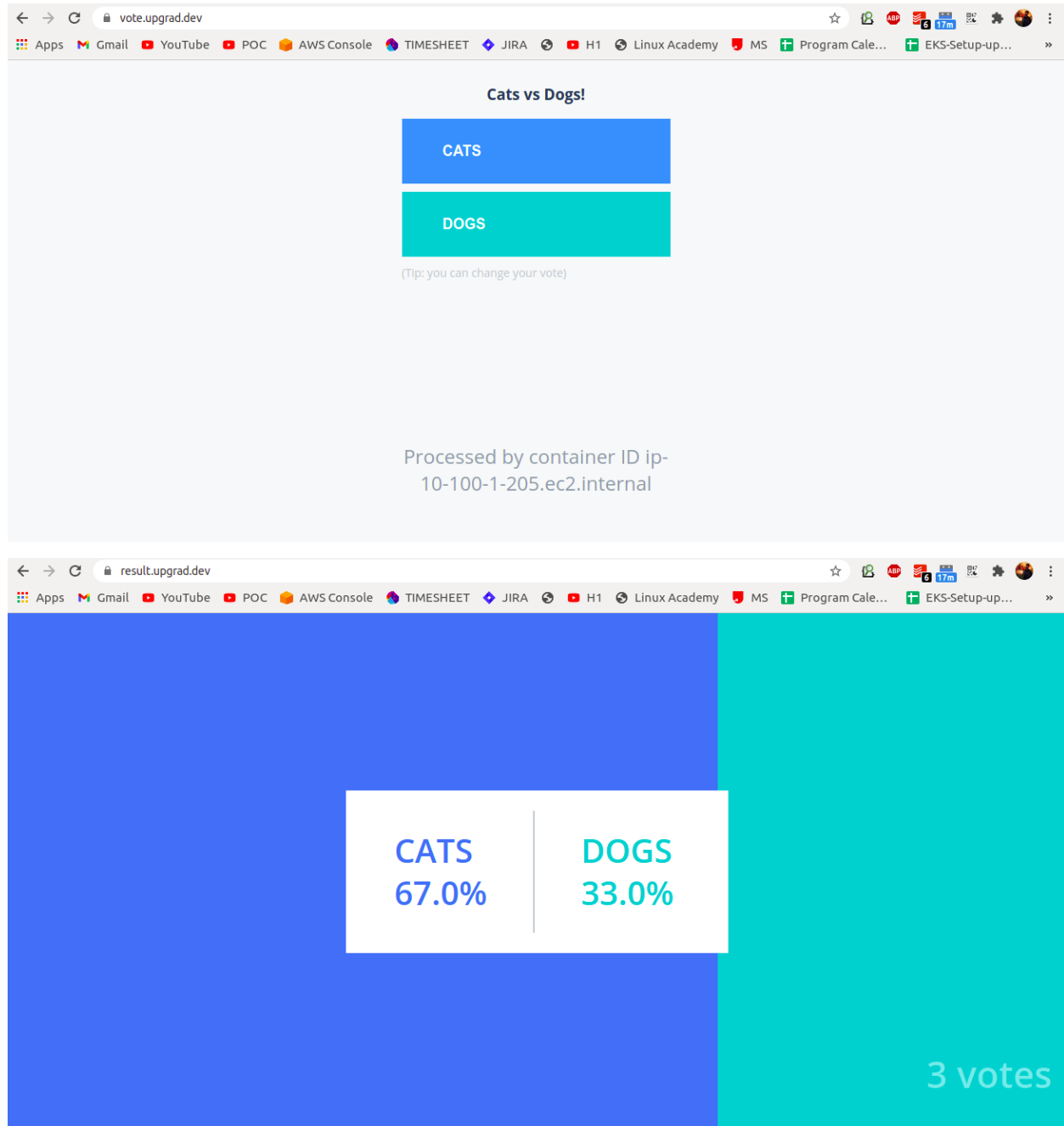
Records (313) Info

Automatic mode is the current search behavior optimized for best filter results. To change modes go to settings.

Type ▼
Routing policy ▼
Alias ▼
1 match < 1 >

X

<input type="checkbox"/>	Record name ▼	Type ▼	Routin... ▼	Differ... ▼	Value/Route traffic to ▼
<input type="checkbox"/>	result.upgrad.dev	CNAME	Simple	-	demo-voting-2130952745.us-east-1.elb.amazonaws.com



7. Service Auto Scaling

- Create Service autoscaler by specifying a rule to scale the system when the CPU reaches 50%

aws Services Search for services, features, marketplace products [Alt+S] upgrad-admin-ss0-access/v_dipesh.garg@upgrad.com @ acadview N. Virginia Support

Step 2: Configure network
Step 3: Set Auto Scaling (optional)
Step 4: Review

Service Auto Scaling (optional)

Automatically adjust your service's desired count up and down within a specified range in response to CloudWatch alarms. You can modify your Service Auto Scaling configuration at any time to meet the needs of your application.

Service Auto Scaling ☐ Do not adjust the service's desired count
☒ Configure Service Auto Scaling to adjust your service's desired count

Minimum number of tasks Automatic task scaling policies you set cannot reduce the number of tasks below this number.

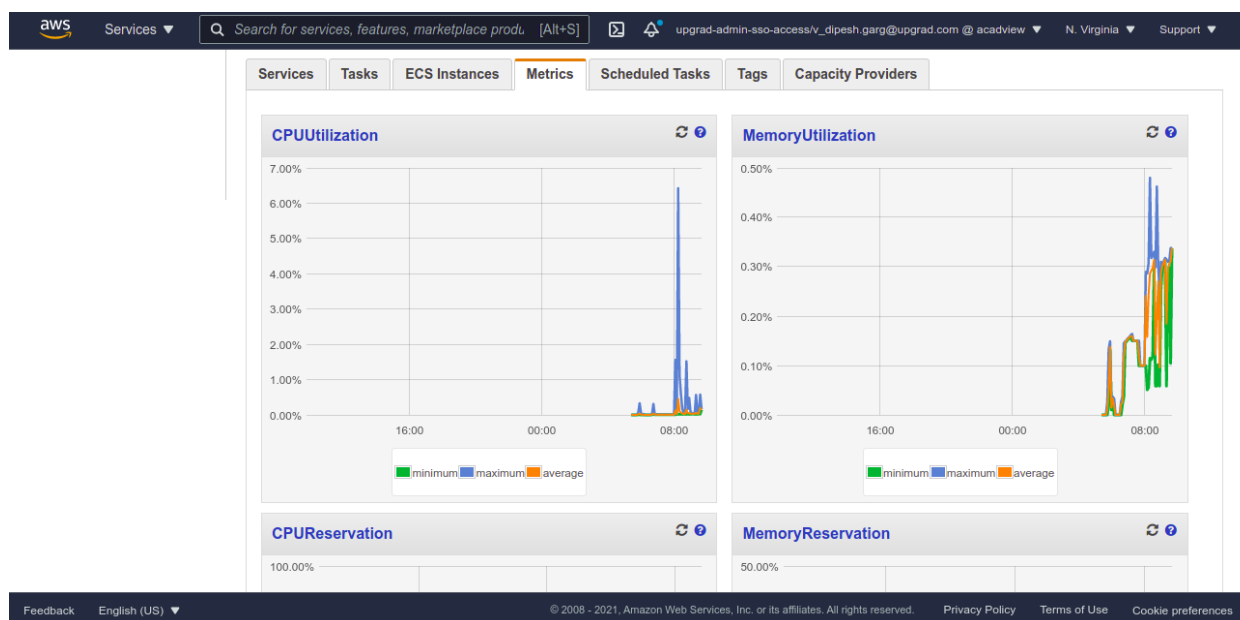
Desired number of tasks Automatic task scaling policies you set cannot increase the number of tasks above this number.

Maximum number of tasks Automatic task scaling policies you set cannot increase the number of tasks above this number.

IAM Automatic task scaling policies you set cannot increase the number of tasks above this number.

8. Logging and Monitoring

a. CPU and Memory of Each service in ECS Console



b. Cluster Stats

c. Logging using Cloudwatch

New ECS Experience
Tell us what you think

Amazon ECS

Clusters

Task Definitions

Account Settings

Amazon EKS

Clusters

Amazon ECR

Repositories

AWS Marketplace

Discover software

Subscriptions

Clusters > demo-voting-ec2 > Task: 1a1d1c6af40c4e599d1cd73e252672ee

Task : 1a1d1c6af40c4e599d1cd73e252672ee

Details

Tags

Logs

Filter logs

All

30s

5m

1h

6h

1d

< 1-100 >

1w

Last updated on March 22, 2021 3:15:25 PM (0m ago)

Timestamp (UTC+00:00)	Message
▶ 2021-03-22 13:38:04	Waiting for db
▶ 2021-03-22 13:38:03	Waiting for db
▶ 2021-03-22 13:38:02	Waiting for db
▶ 2021-03-22 13:38:01	Waiting for db
▶ 2021-03-22 13:38:00	Waiting for db
▶ 2021-03-22 13:37:59	Waiting for db
▶ 2021-03-22 13:37:58	Waiting for db
▶ 2021-03-22 13:37:57	Waiting for db

Run more like this

Stop

Disclaimer

- After completing your work, you must terminate all the AWS resources such as EC2 instances, RDS instances, S3 buckets, or any other AWS resource you will create during the hands-on activity.
- Delete the custom **VPC**, and **NAT** gateways after the work is completed. Stop running ECS tasks and update service definition to set desired tasks as 0 before leaving the ECS console. Delete the **ECS -EC2 cluster** and terminate all its EC2 instances when not in use.
- Do not forget to delete all the **load balancers** and **autoscaling groups** before logging out from your account.

Note: Please create an EC2 instance with less than t2.medium size to avoid budget overshoot and AWS account suspension.