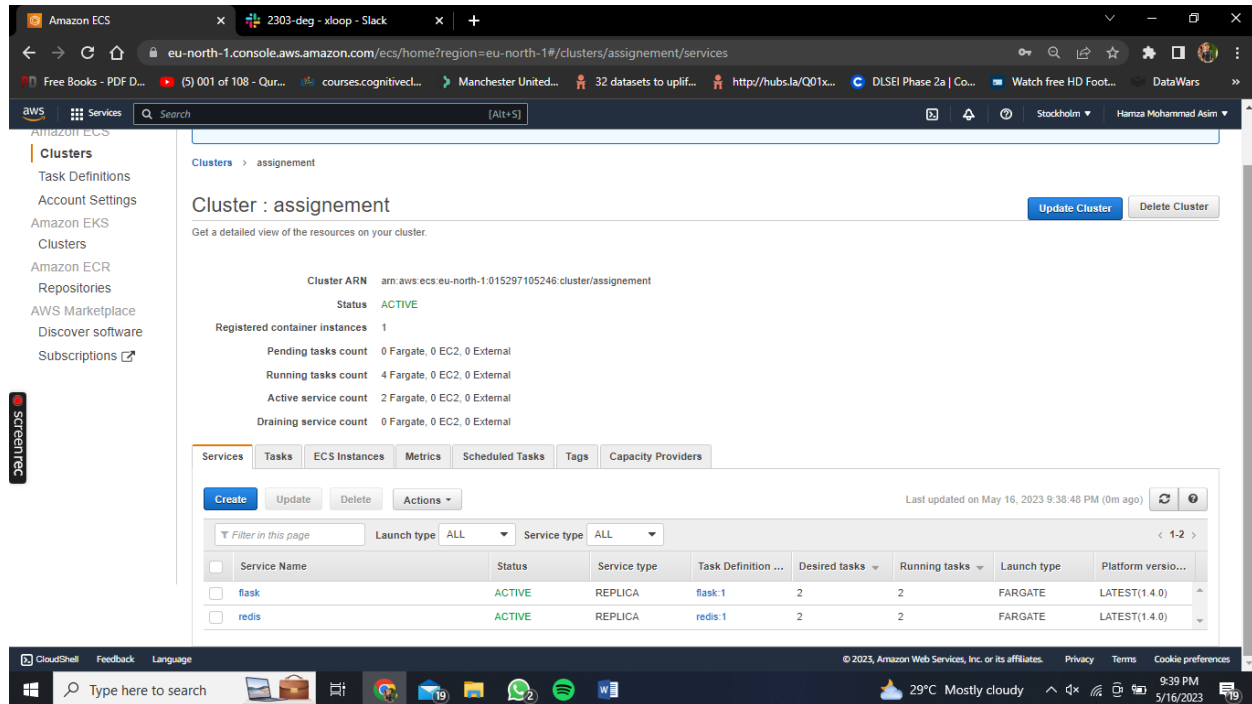


Graded Assignment 4.5

Name: Saad Sameer Khan

Employee#: 2303.KHI.DEG.034

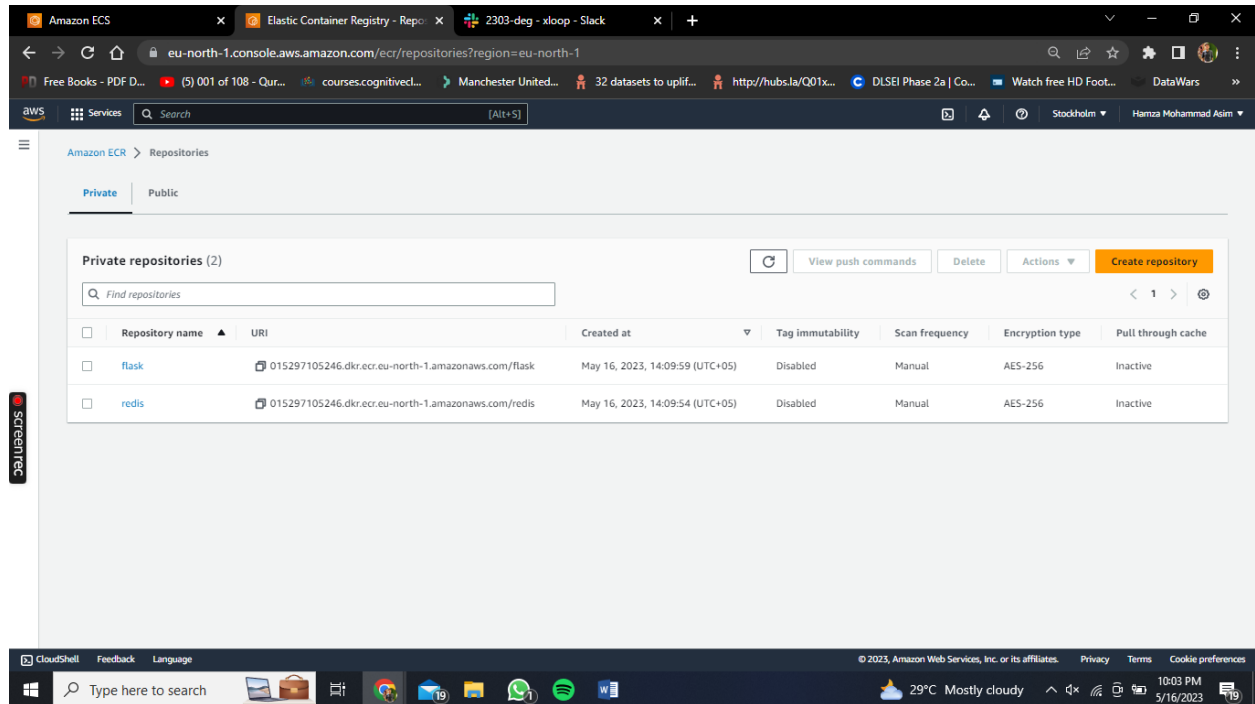
Collaborated with: Mohammad Hamza Asim (2303.KHI.DEG.014)



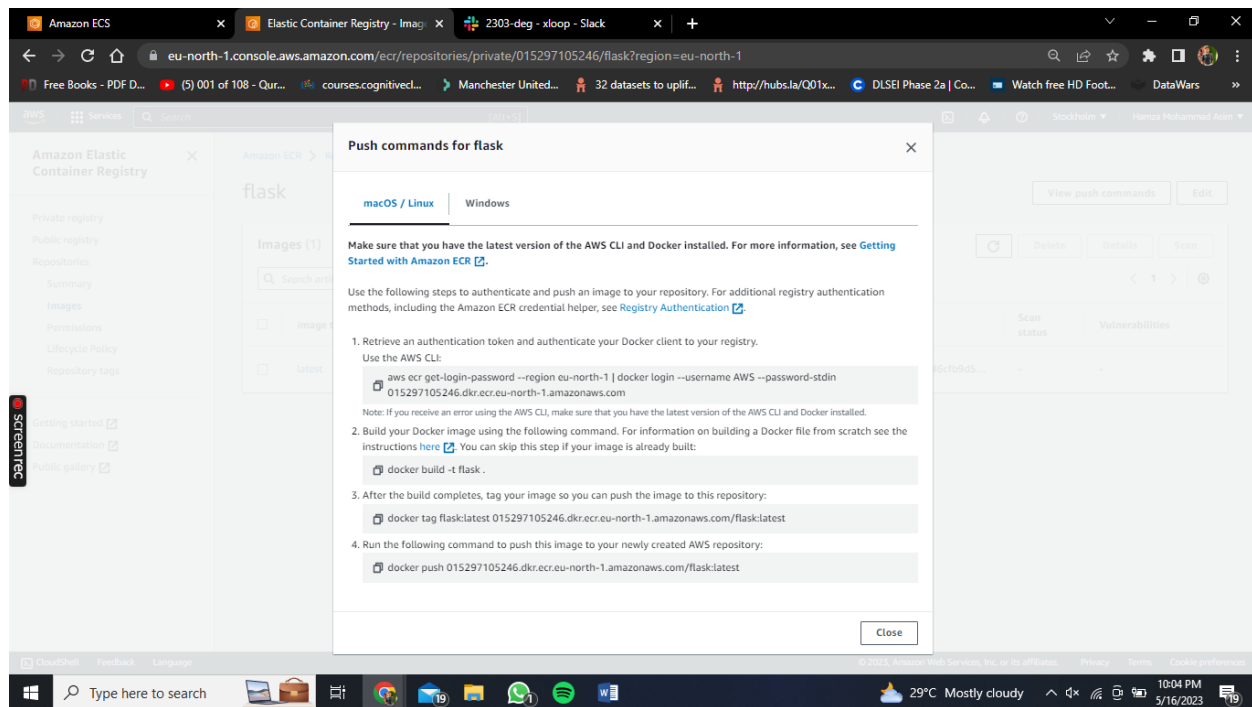
Created a cluster named 'assignment' and added 2 services in it. Clusters are basically a group where you are capable of managing and deploying the containers. Services are taking care of the containers where you can see whether it's in active or inactive state and provide a replica service type in case of failure, which version of Task Definition do you want to run on (which is mentioned later) and so on

The two services added were:

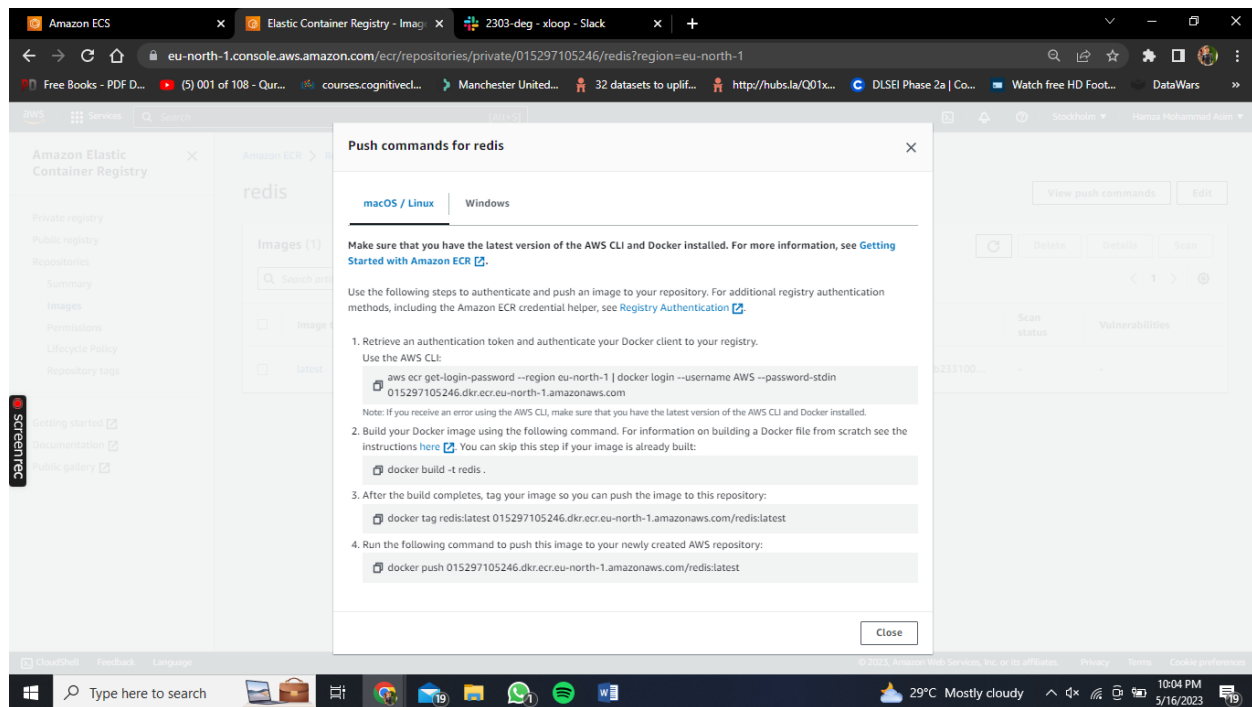
1. 'flask': the flask app
2. 'redis': redis image



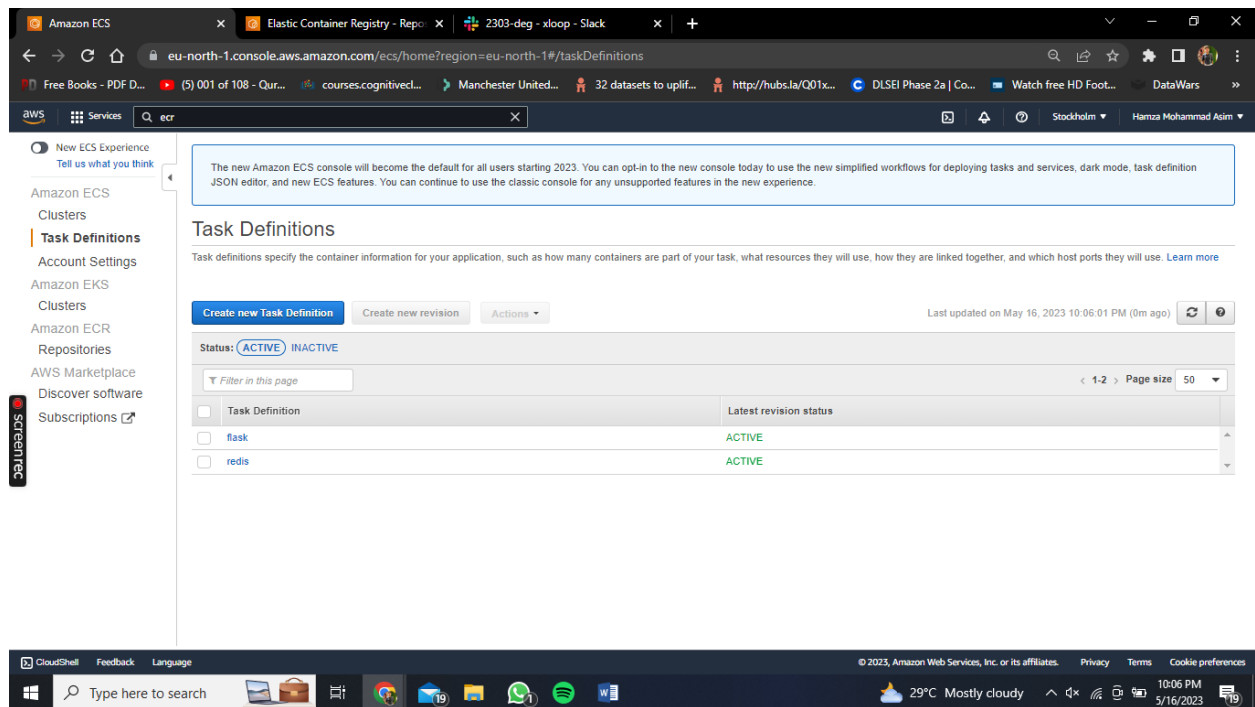
This is ECR (Elastic Container Registry) where we have a repository which allows us to push our image in this AWS registry for various use and purposes. As seen in the picture, two are added one is flask app and other is redis.



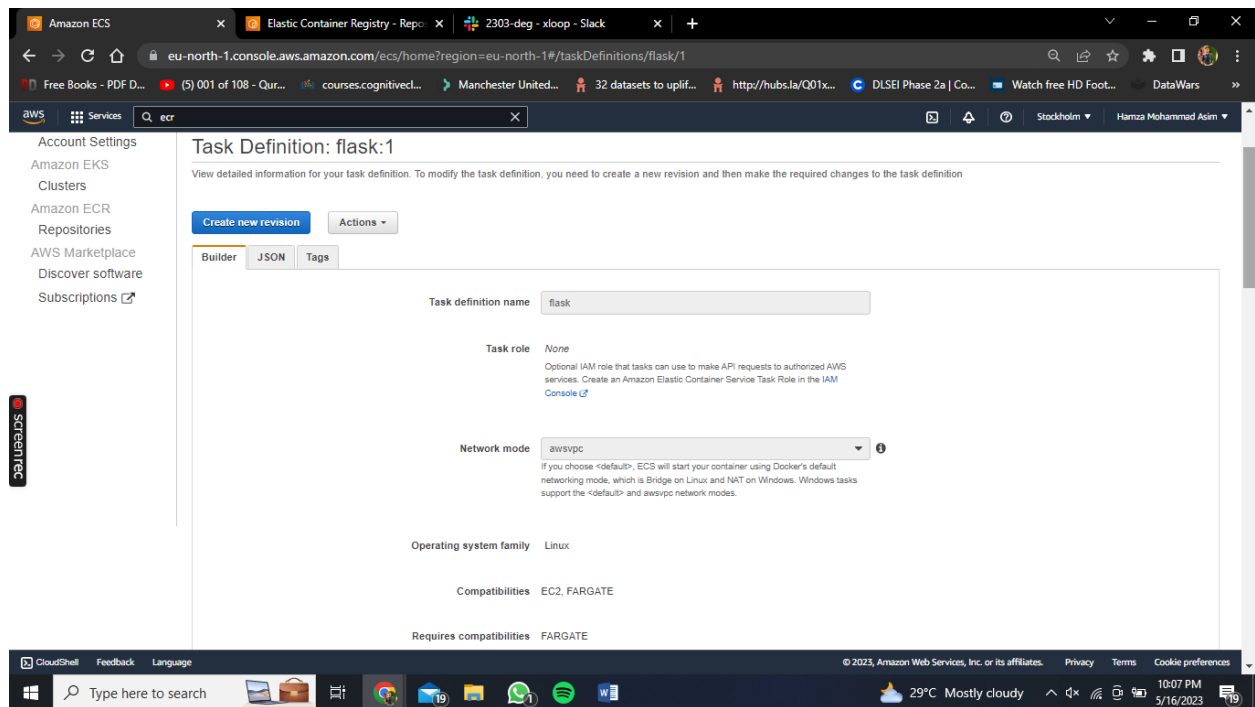
These commands were run in the same directory where the dockerfile was present along with the app.py and requirement.txt for pushing the flask app into the registry ECR



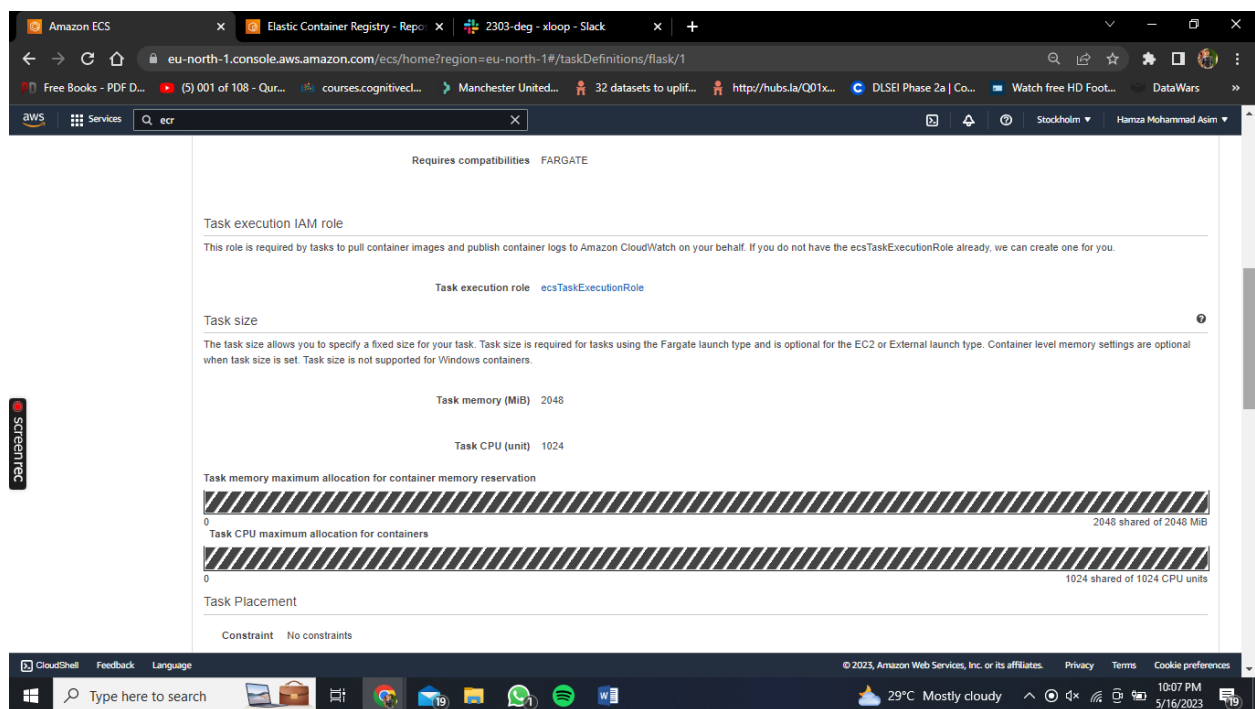
Likewise, same steps were performed for Redis image but the only slight difference in this was that instead of build command we ran 'docker pull redis' which allowed us to pull the redis image from dockerhub and then we simply push it in the ECR



Here we have created the Task Definitions. This section of ECS is very crucial and should be carefully looked at as we move on to specify important details such as the container image, CPU and memory requirements, networking settings. It's like a blueprint that guides ECS on how to create and manage your containers effectively.



Here is the task definition of flask. Fargate is chosen for compatibility as it makes our lives much easier when it comes to management of underlying compute resources which are required to run a container



Resource allocation was provided of 2GB RAM and 1vCPU

The screenshot displays the Amazon ECS console interface for a task definition named 'flask'. The browser address bar shows the URL: `eu-north-1.console.aws.amazon.com/ecs/home?region=eu-north-1#/taskDefinitions/flask/1`. The console shows the following configuration details:

Container Name	Image	CPU Units	GPU	Inference Accelerato...	Hard/Soft memory limits (MiB)	Essential
flask	015297105246.dkr.ecr...	0			--/--	true

Details

Port Mappings

Host Port	Container Port	Protocol
5000	5000	Tcp

Environment Variables

Key	Value/ValueFrom
SERVICE_DISCOVERY	redis.redis.com

Environment Files

Source	Location
No environment files	

Container Ordering

Container Name	Condition
No container ordering	

Container Timeouts

Mount Points

Container Path	Source Volume	Read only
No mount points		

Volumes from

Source Container	Read only
No volumes from	

Ulimits

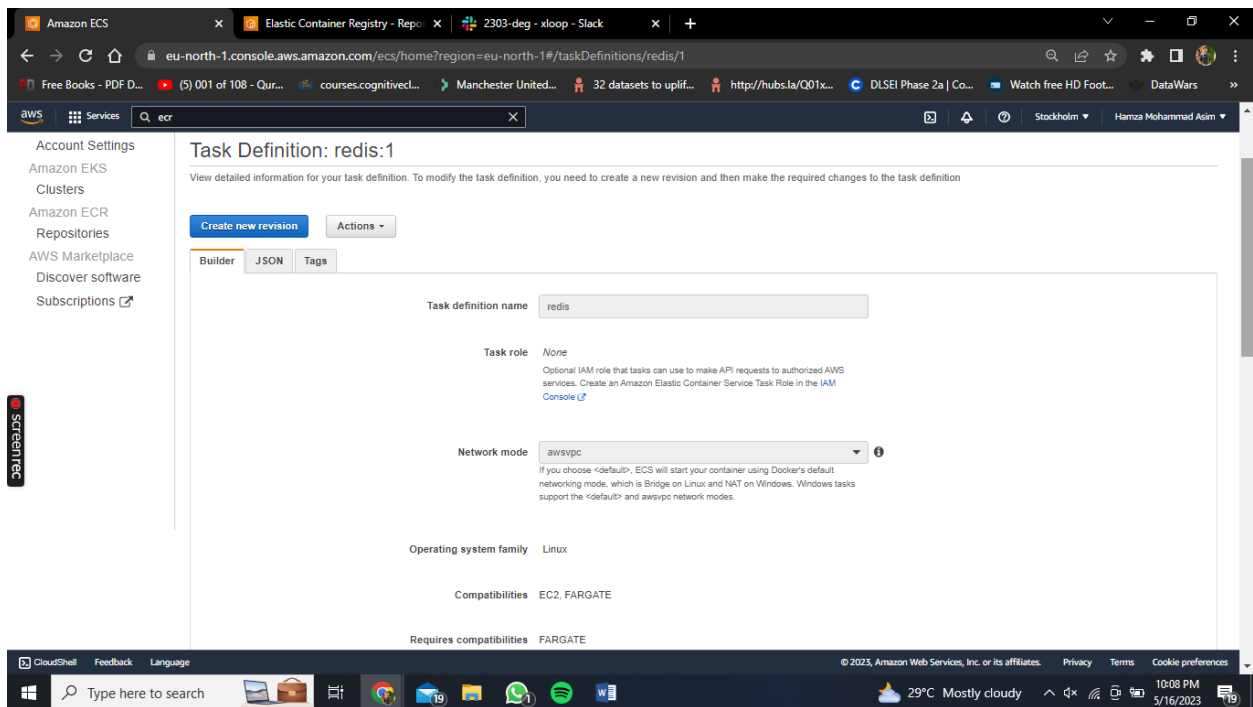
Name	Soft limit	Hard limit
No ulimit		

Log Configuration

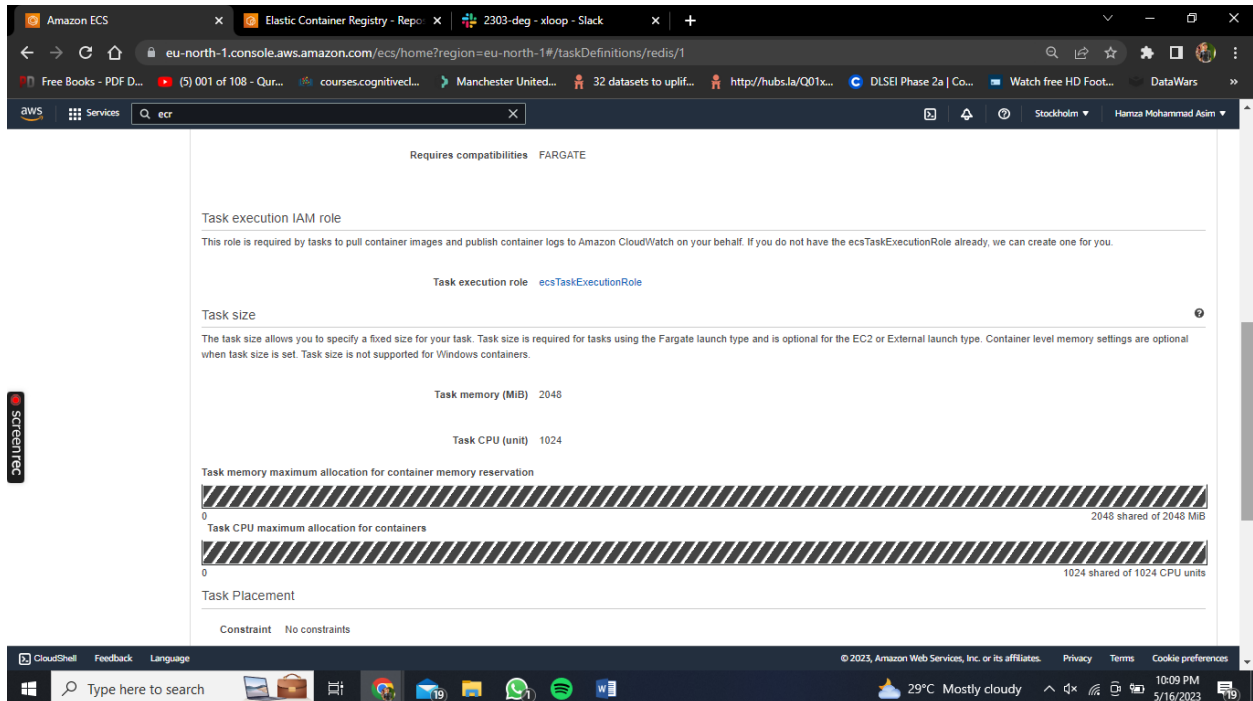
Log driver: awslogs

Key	Value
awslogs-group	/ecs/flask
awslogs-region	eu-north-1
awslogs-stream-prefix	ecs

This is the container which is added and the image url was provided the same from ECR where we recently pushed our app. Port was provided 5000. Environment Variables were provided with key 'SERVICE_DISCOVERY' which allows us to host our flask and easily locate the redis container.



Likewise, same settings were provided to Redis Task definition as well.



Amazon ECS Elastic Container Registry - Repo 2303-deg - xloop - Slack

eu-north-1.console.aws.amazon.com/ecs/home?region=eu-north-1#/taskDefinitions/redis/1

Free Books - PDF D... (5) 001 of 108 - Qur... courses.cognitived... Manchester United... 32 datasets to uplif... http://hubs.la/Q01x... DLSEI Phase 2a | Co... Watch free HD Foot... DataWars

Services ecr

Container definitions

Container Name	Image	CPU Units	GPU	Inference Accelerato...	Hard/Soft memory limits (MiB)	Essential
redis	015297105246.dkr.ecr...	0				true

Details

Port Mappings

Host Port	Container Port	Protocol
6379	6379	tcp

Environment Variables

Key	Value/ValueFrom
No environment variables	

Environment Files

Source	Location
No environment files	

Container Ordering

Container Name	Condition
No container ordering	

Container Timeouts

Mount Points

Container Path	Source Volume	Read only
No mount points		

Volumes from

Source Container	Read only
No volumes from	

Ulimits

Name	Soft limit	Hard limit
No ulimit		

Log Configuration

Log driver: awslogs

Key	Value
awslogs-group	/ecs/redis
awslogs-region	eu-north-1
awslogs-stream-prefix	ecs

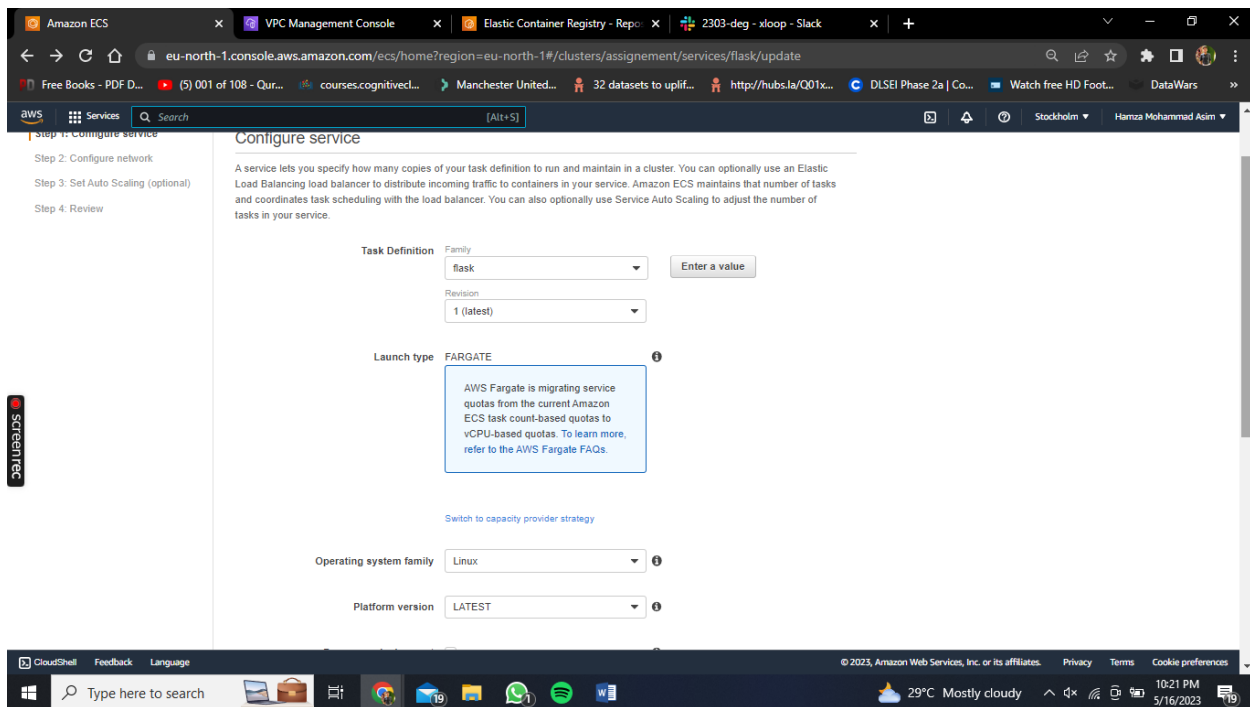
CloudShell Feedback Language

Type here to search

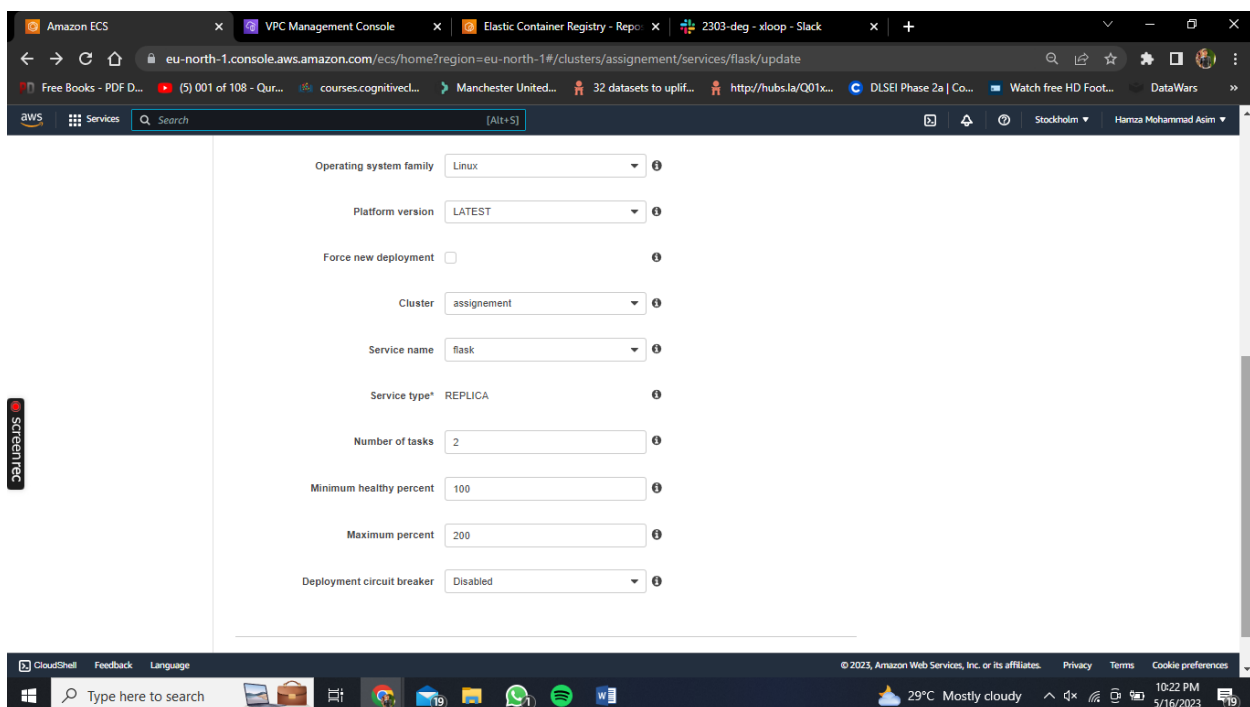
© 2023, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences

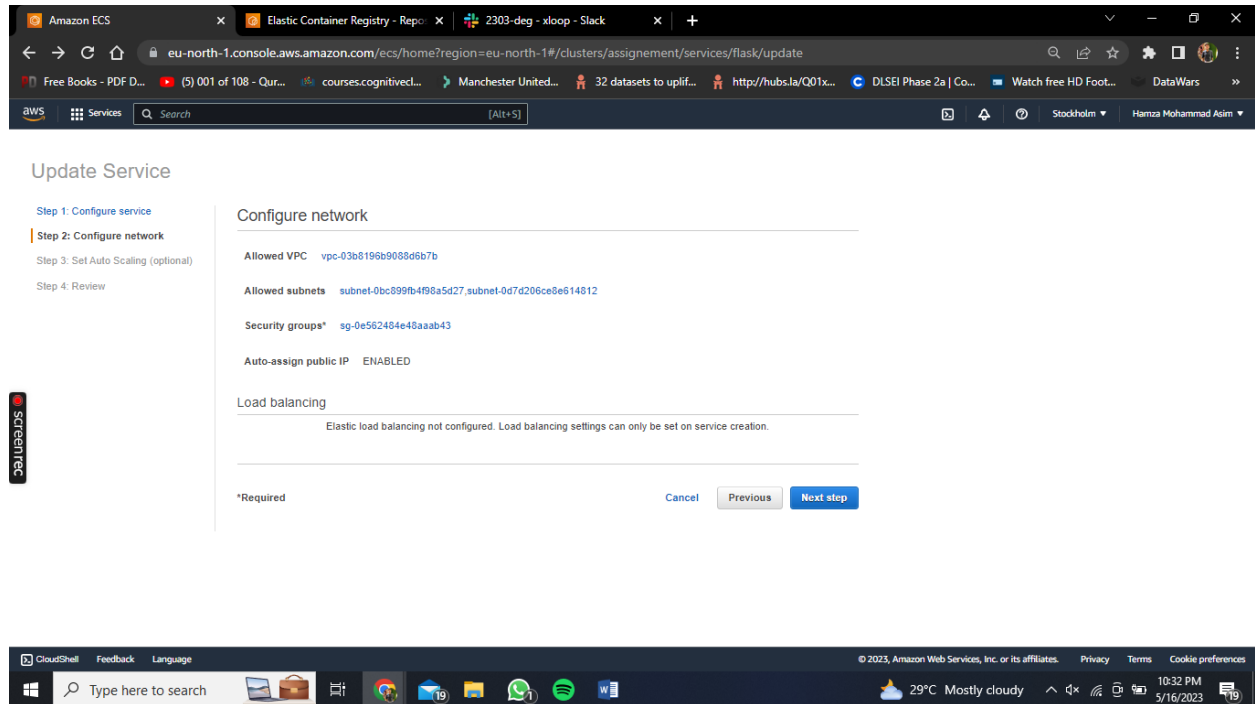
29°C Mostly cloudy 10:09 PM 5/16/2023

Port mentioned above: 6379

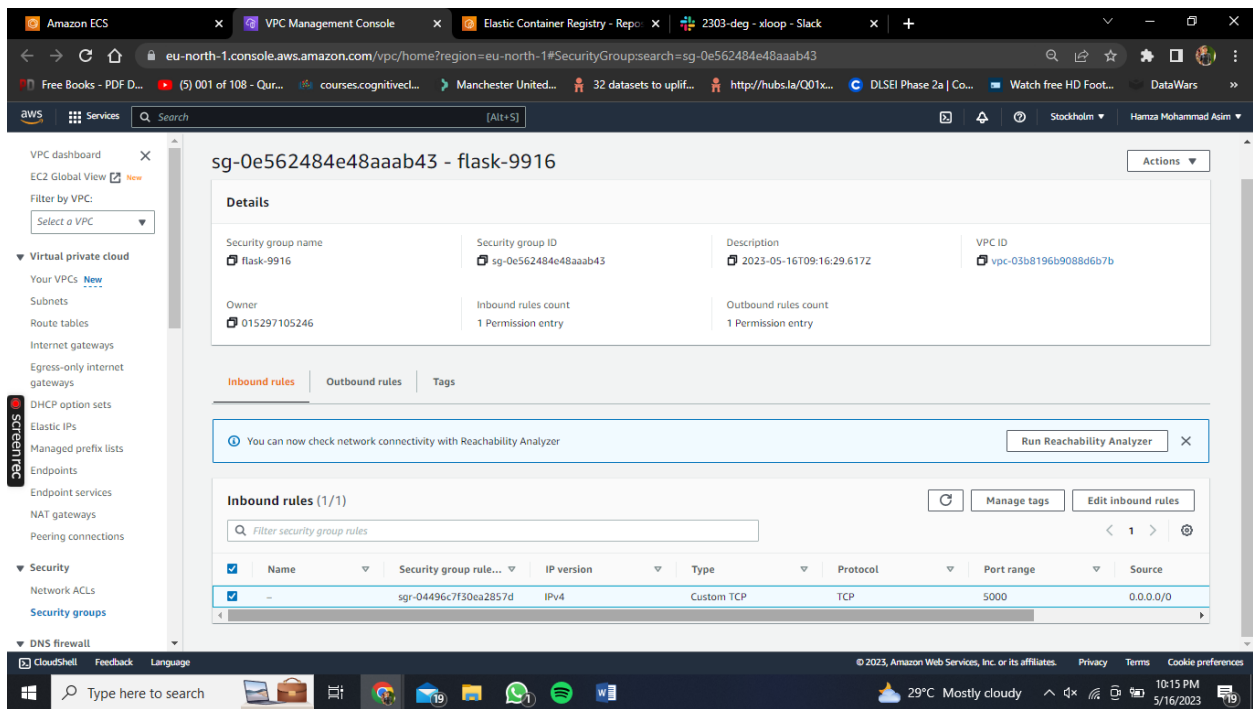


Now we're back in ECS to configure our services after creating Definition Tasks. The service that we created recently of both flask and redis, we'll specify those definition tasks while configuring service in both services respectively.

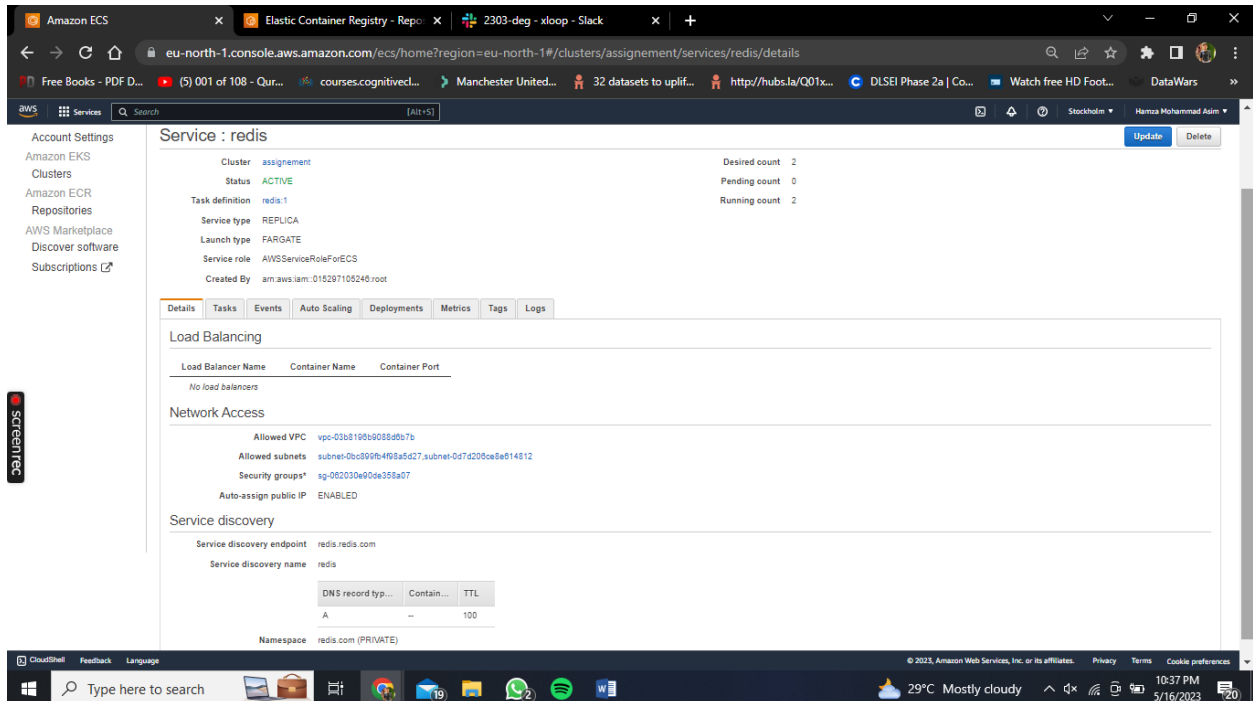




VPC is chosen same as cluster's. we chose all available subnets.



Security group inbound rule was added with port range 5000



Likewise, same done for redis. Note that namespace was mentioned over here which was created in AWS cloudmap named : redis.com and service name : redis.
Service discovery endpoint: redis.redis.com

Amazon ECS VPC Management Console Elastic Container Registry - Repo 2303-deg - xloop - Slack

eu-north-1.console.aws.amazon.com/vpc/home?region=eu-north-1#SecurityGroup:search=sg-062030e90de358a07

Free Books - PDF D... (5) 001 of 108 - Qur... courses.cognitived... Manchester United... 32 datasets to uplif... http://hubs.la/Q01x... DLSEI Phase 2a | Co... Watch free HD Foot... DataWars

Services Search [Alt+S]

VPC dashboard EC2 Global View Filter by VPC: Select a VPC

Virtual private cloud Your VPCs Subnets Route tables Internet gateways Egress-only internet gateways DHCP option sets Elastic IPs Managed prefix lists Endpoints Endpoint services NAT gateways Peering connections

Security Network ACLs Security groups

DNS firewall

sg-062030e90de358a07 - redis-372

Details

Security group name redis-372	Security group ID sg-062030e90de358a07	Description 2023-05-16T09:17:30.009Z	VPC ID vpc-03b8196b9088d6b7b
Owner 015297105246	Inbound rules count 1 Permission entry	Outbound rules count 1 Permission entry	

Inbound rules Outbound rules Tags

You can now check network connectivity with Reachability Analyzer [Run Reachability Analyzer](#)

Inbound rules (1/1)

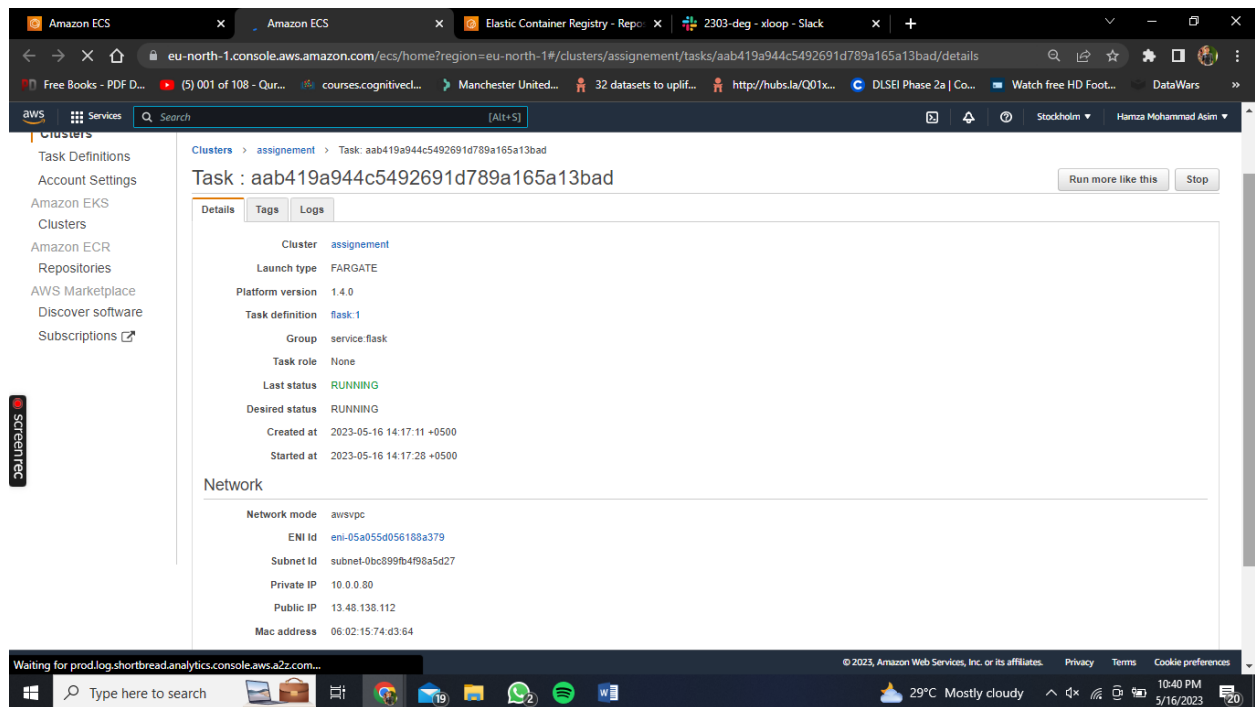
Filter security group rules

Name	Security group rule	IP version	Type	Protocol	Port range	Source
	sg-055032c412d209...	IPv4	Custom TCP	TCP	6379	0.0.0.0/0

© 2023, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences

29°C Mostly cloudy 10:39 PM 5/16/2023

Security group inbound rule was added with port range 6379



After setting up everything, In this step we will go to our cluster and in flask service. We'll open up a task and click on ENI ID

The screenshot shows the AWS Management Console for the 'eu-north-1' region. The 'Network interfaces' page is active, displaying a table with one network interface selected. The details for the selected network interface are shown below the table.

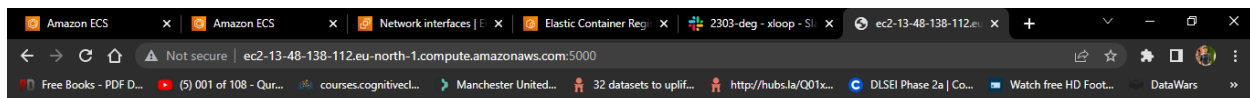
Name	Network interface ID	Subnet ID	VPC ID	Availability Zone	Security group n...	Security group IDs
-	eni-05a055d056188a379	subnet-0bc899fb4f98a5d27	vpc-03b8196b9088d6b7b	eu-north-1a	flask-9916	sg-0e562484e48aaa...

Network interface: eni-05a055d056188a379

Owner 015297105246	Requester ID 347722326666	Requester-managed True
Source/dest. check True		
IP addresses		
Private IPv4 address 10.0.0.80	Private IPv4 DNS ip-10-0-0-80.eu-north-1.compute.internal	Elastic Fabric Adapter False
Public IPv4 address 13.48.138.112	Public IPv4 DNS ec2-13-48-138-112.eu-north-1.compute.amazonaws.com	IPv6 addresses -
Secondary private IPv4 addresses -	Association ID -	Elastic IP address owner amazon

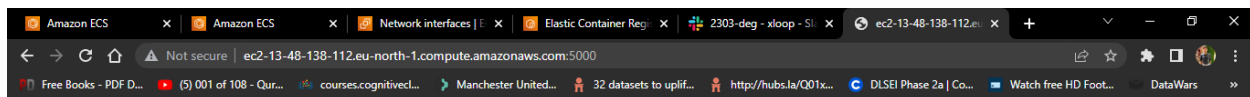
After ticking the Network interface above, a box will open up where we can copy our Public IPv4 DNS and copy on a new tab along with the port 5000. Results are attached below.

' http://ec2-13-48-138-112.eu-north-1.compute.amazonaws.com:5000/ '



Hello World! I have been seen 4 times.

screenrec



Hello World! I have been seen 4 times.

screenrec

