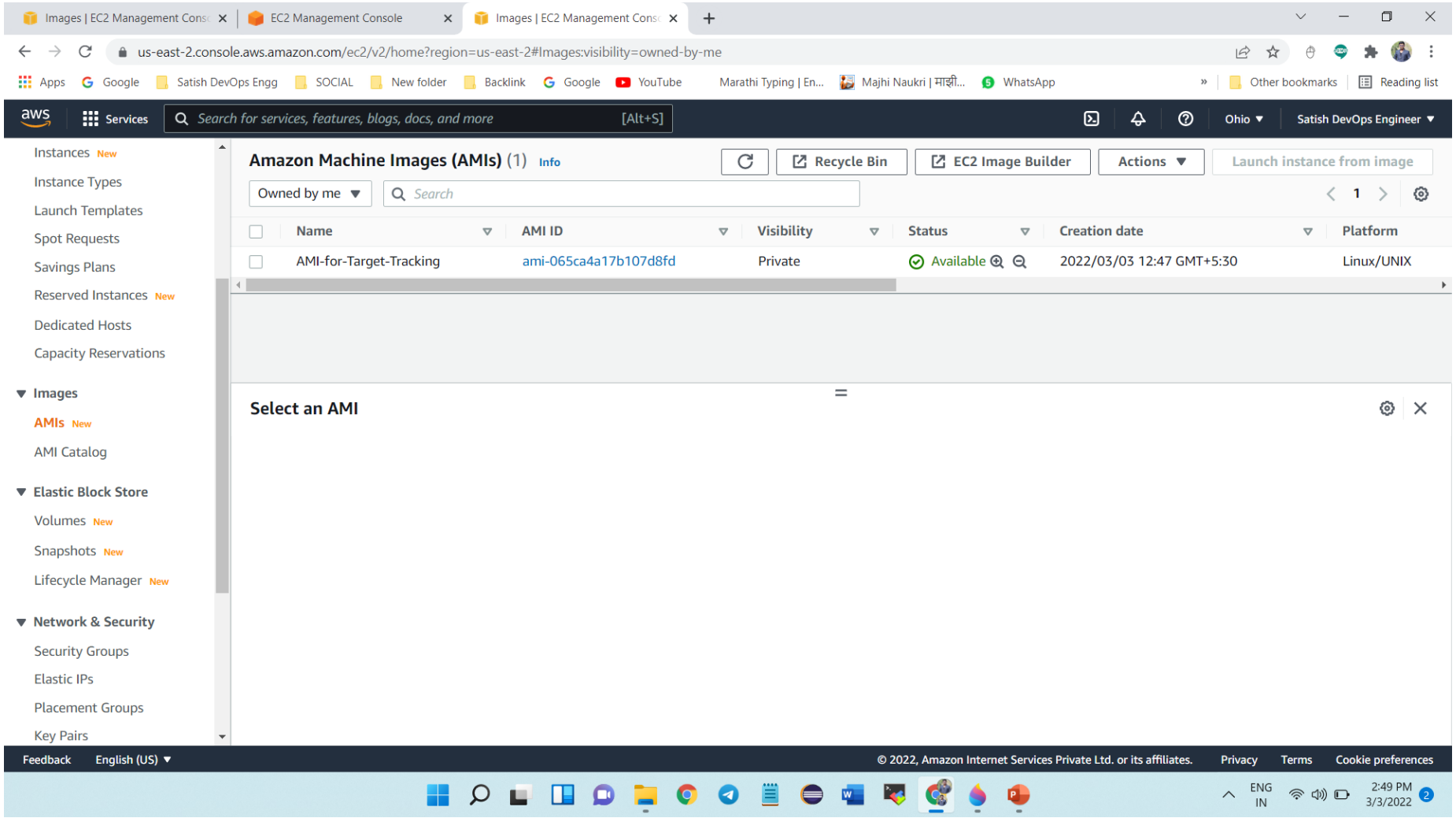
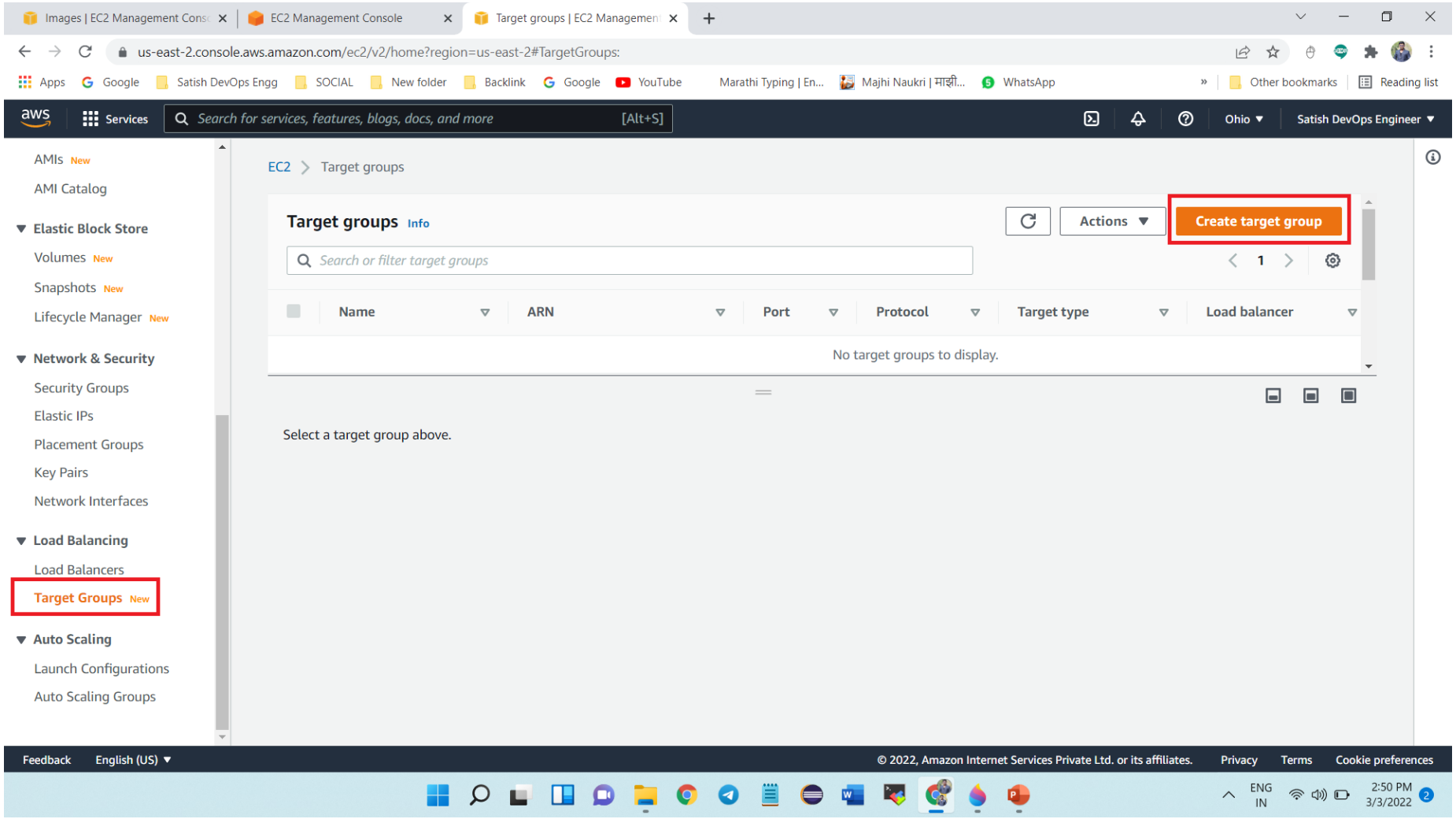
018 Practical - Auto Scaling with Target Tracking Policy - 02 Mar 2022

- We have to create one instance using AMAZON Linux AMI.
- After launching Instance we have installed httpd by using **yum install httpd –y** command
- Start the server by using **service httpd start** command
- Now go to **/var/www/html** directory by using **cd** command
- Add one **index.html** file into html directory
- Use **chkconfig httpd on command**

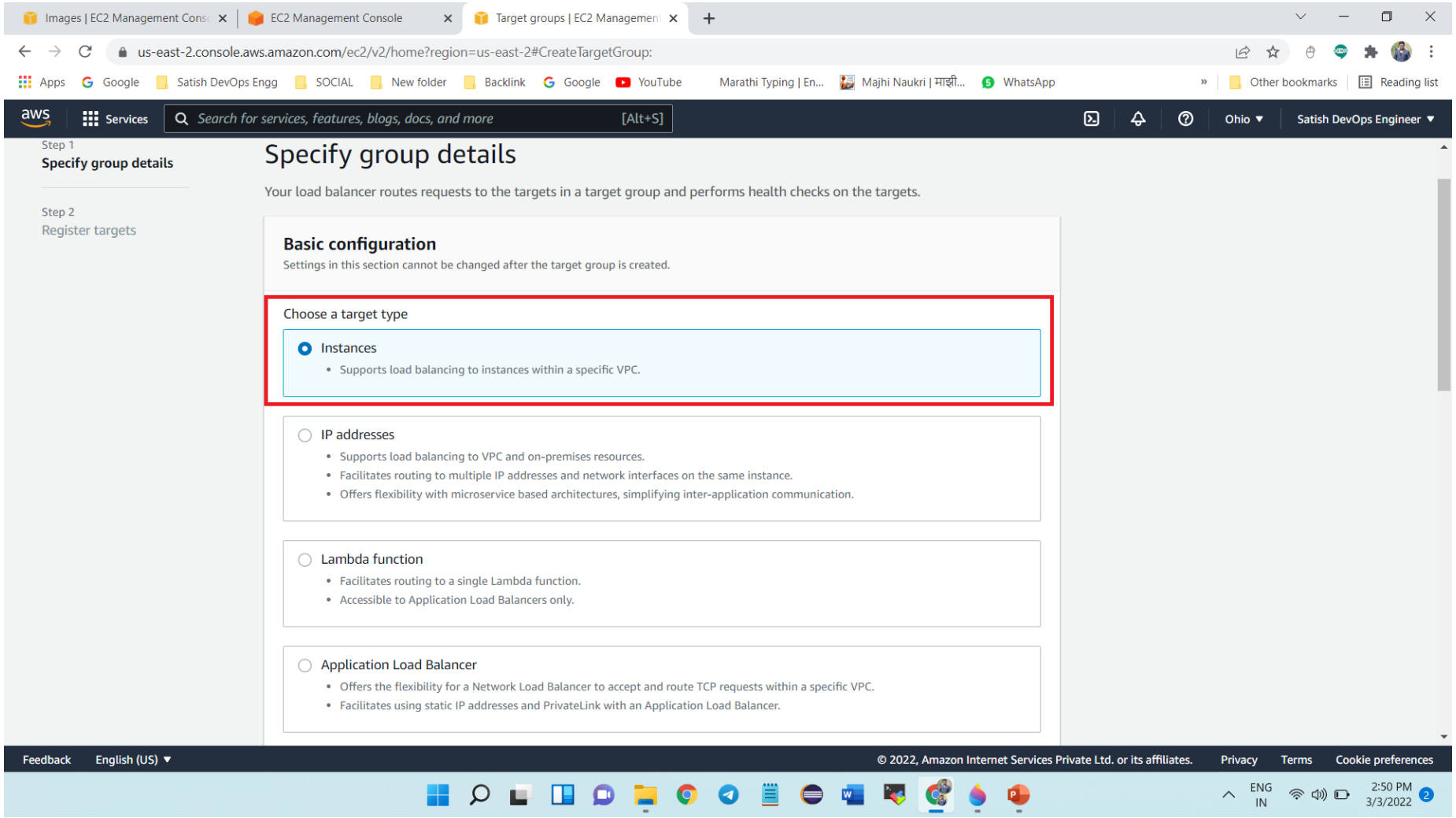
Now go to AWS Console and create AMI of that instance



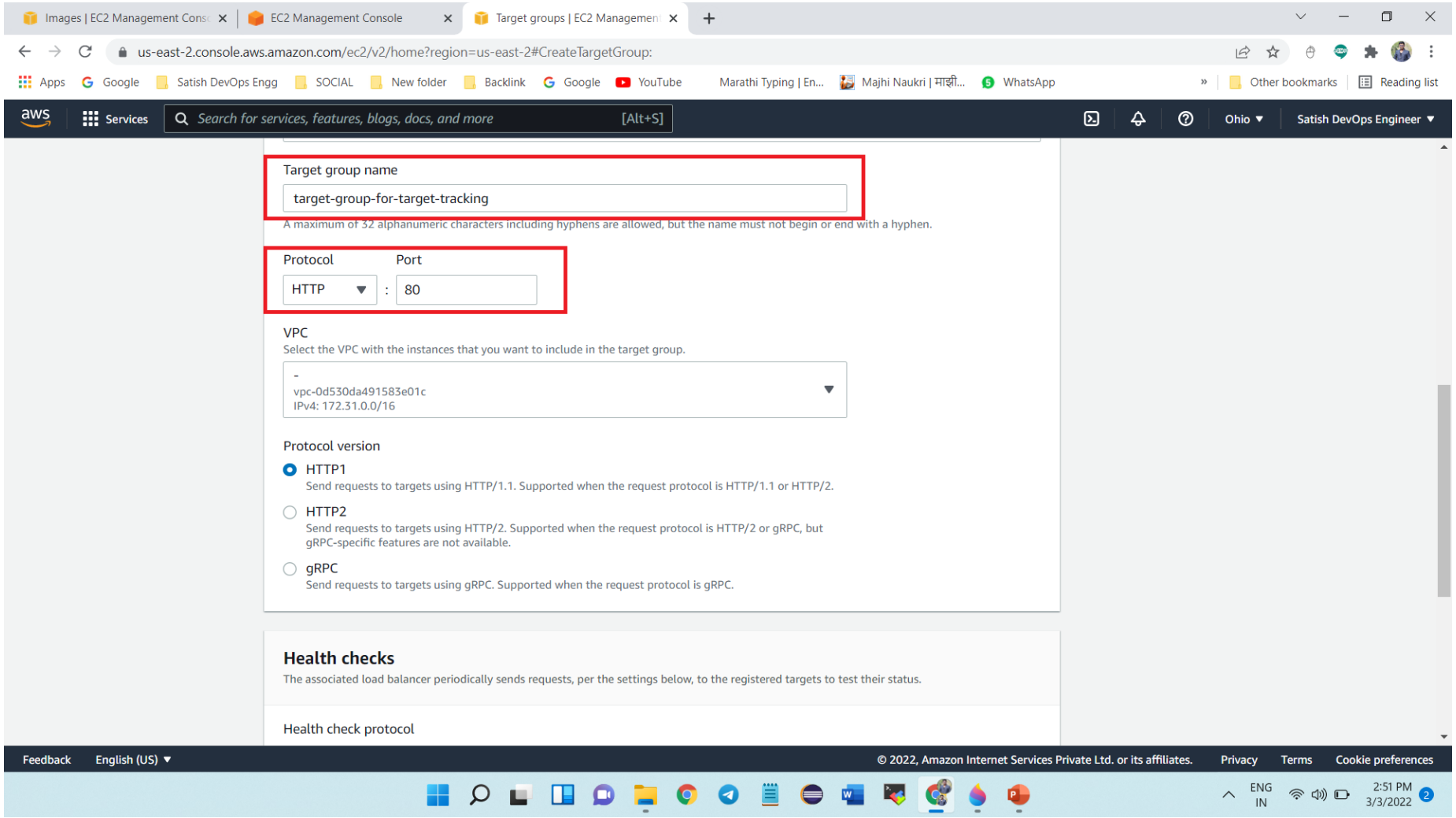
- Here we have created one AMI



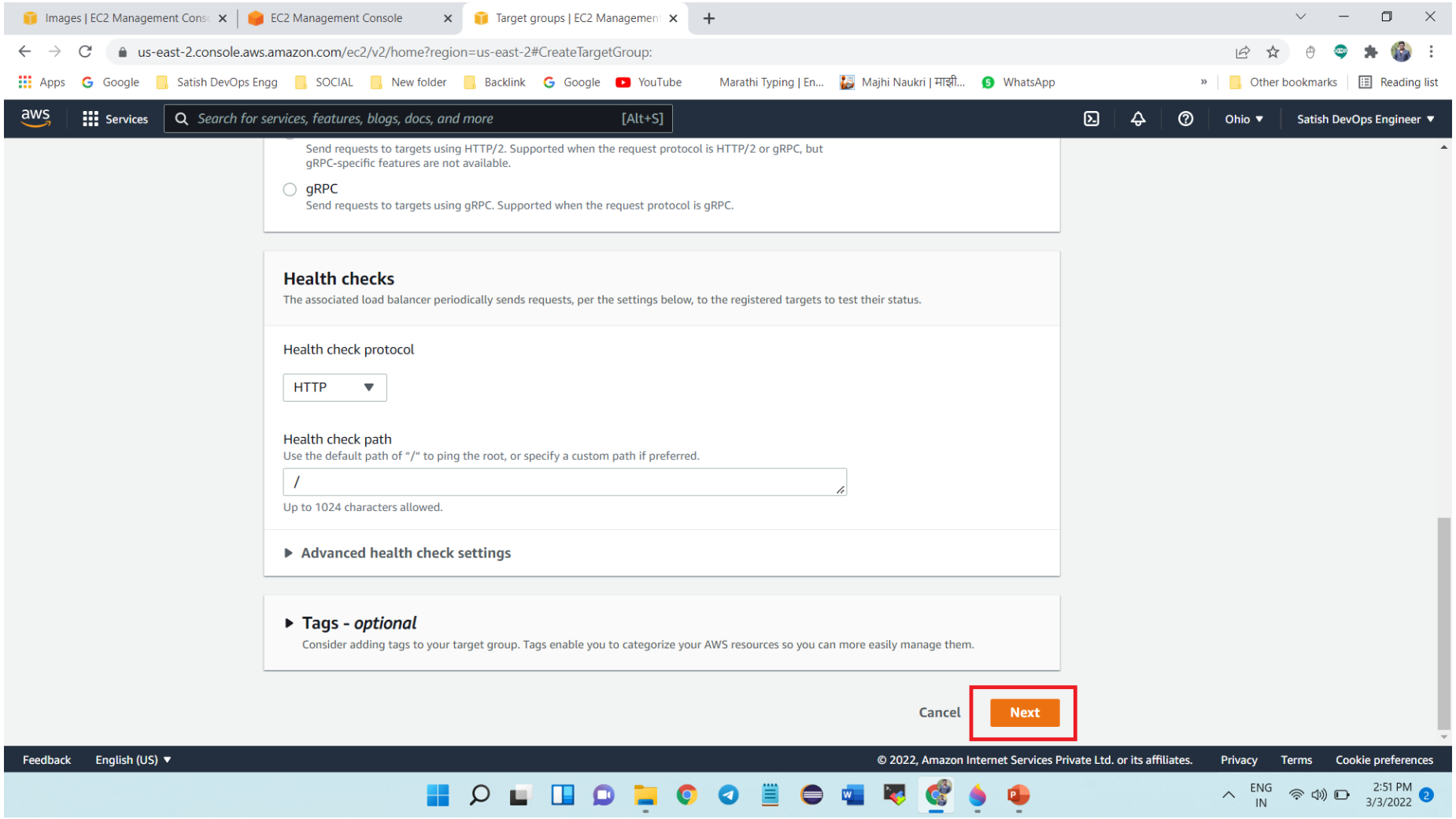
- Now go to Target Group
- Click on Create Target Group



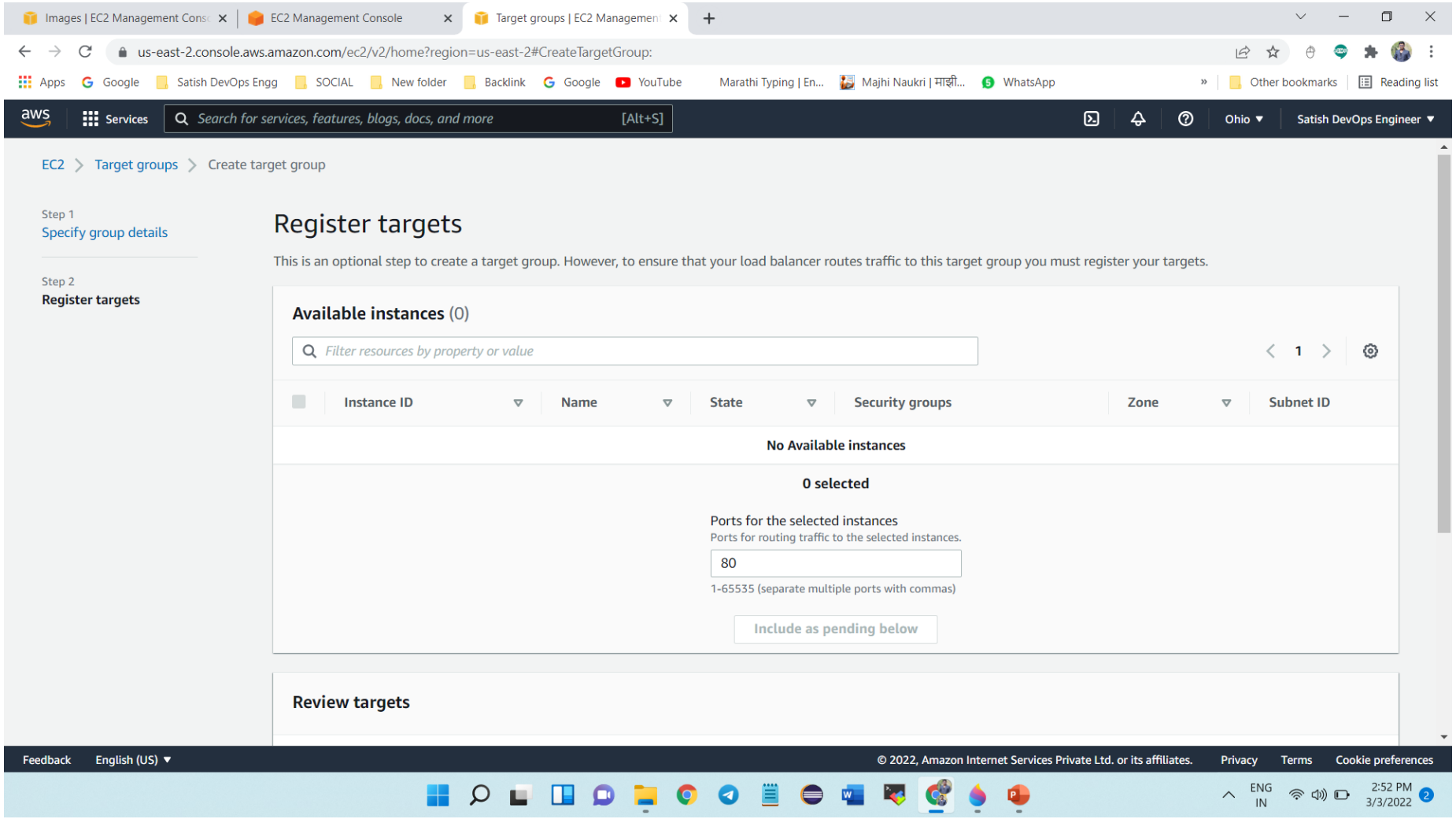
- Click on Instances



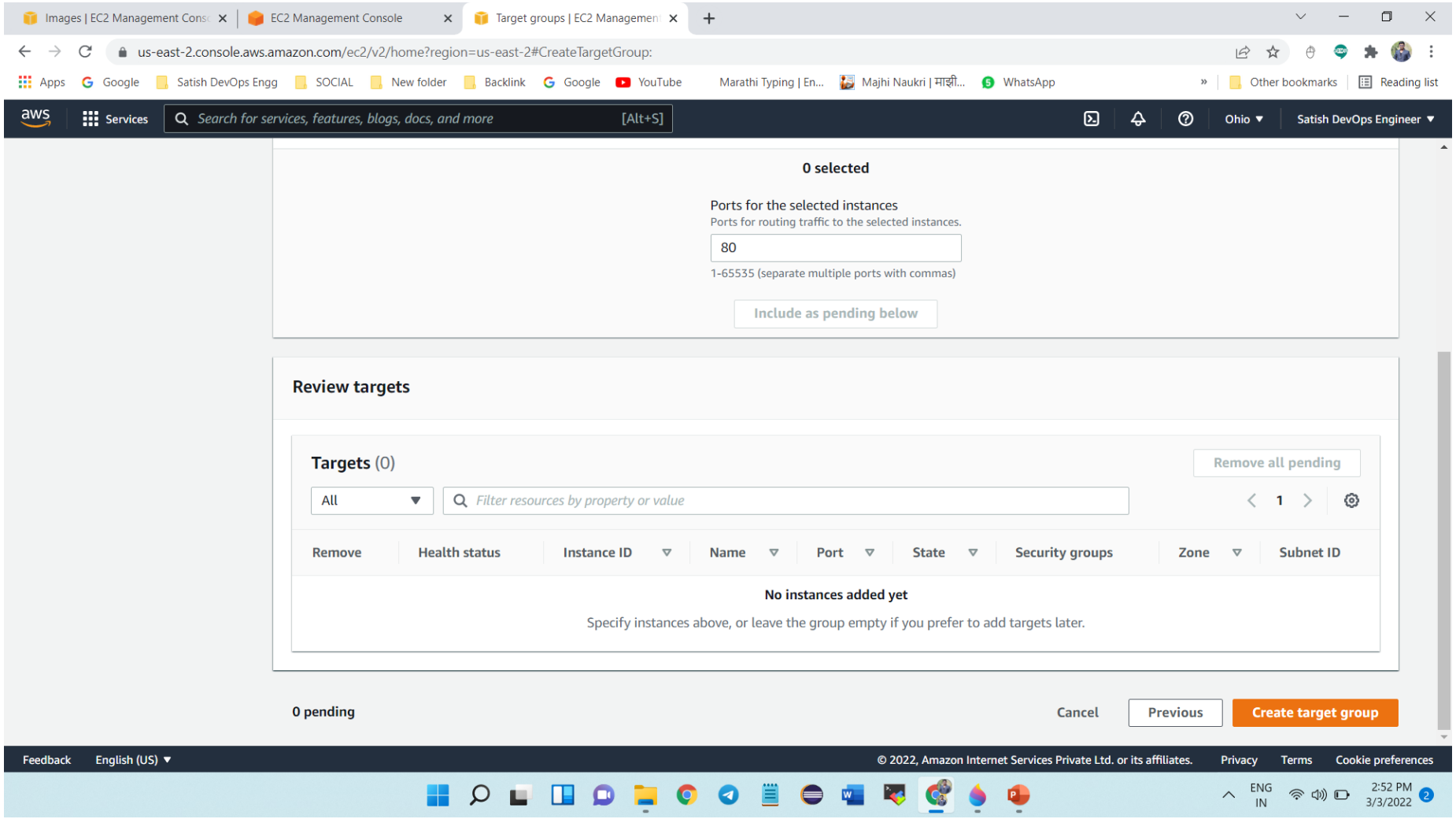
- Give Name to Target group
- Select Http Protocol and Port 80



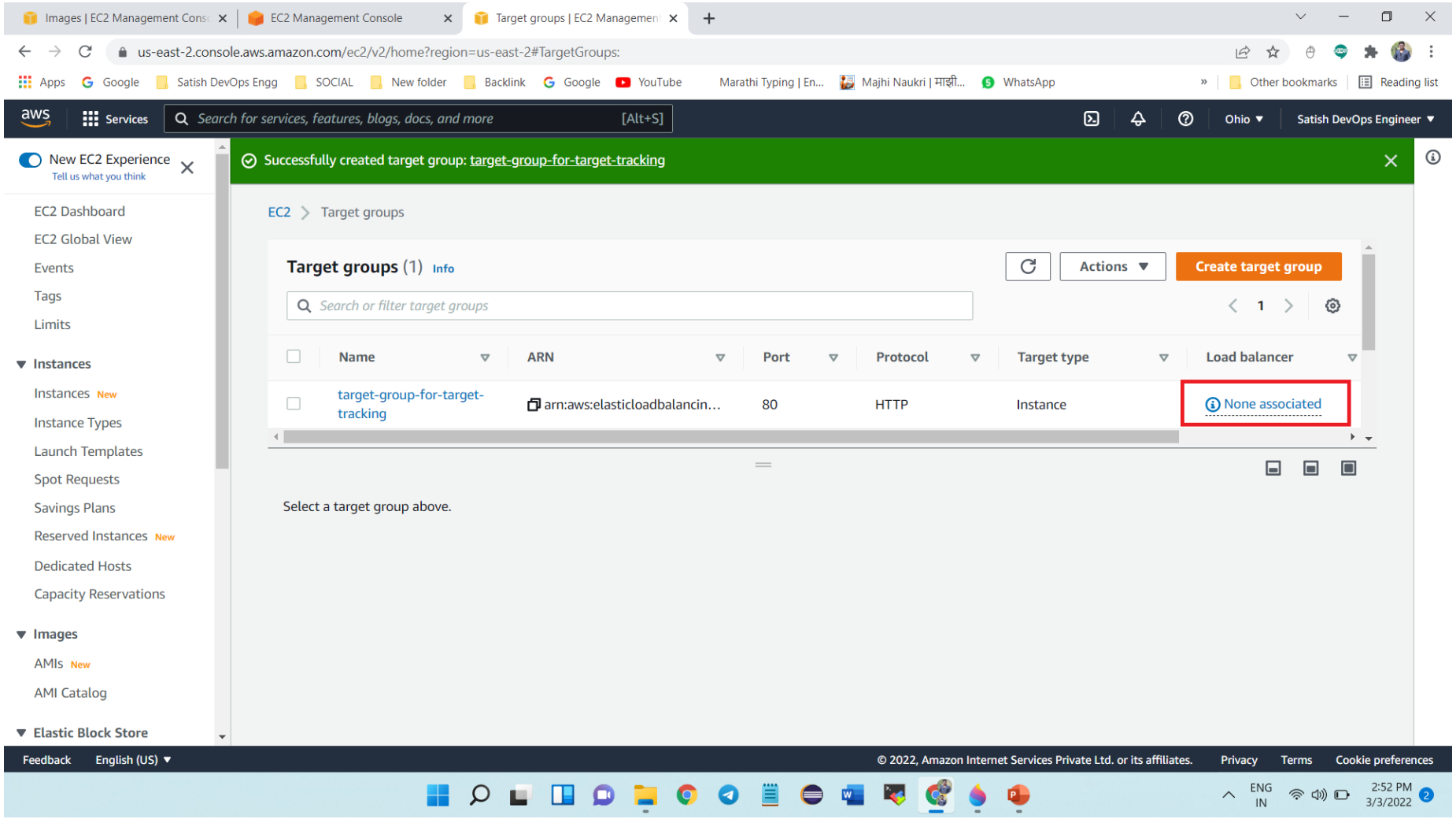
- Scroll Down
- Click on Next



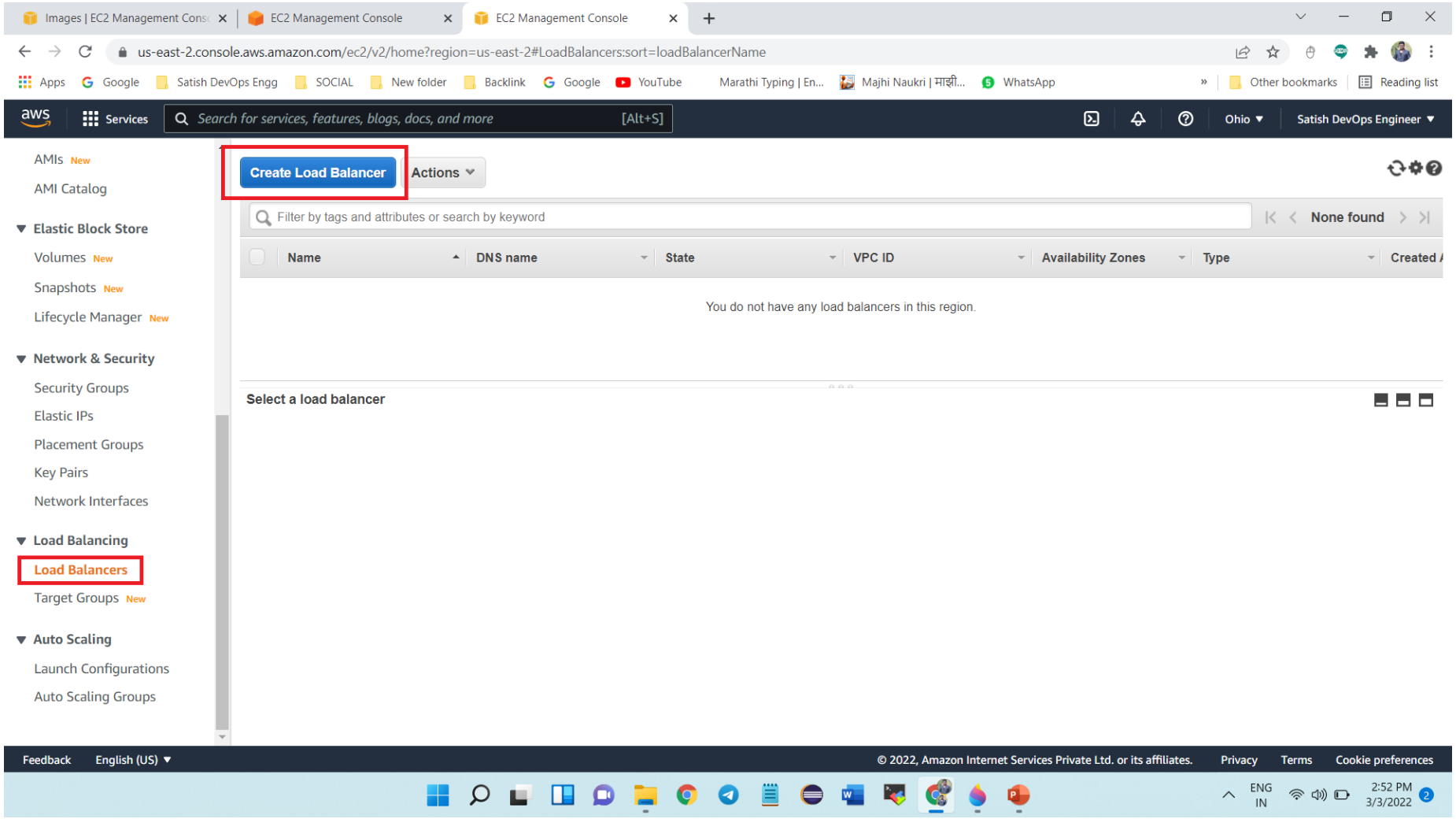
- Then Again Scroll Down



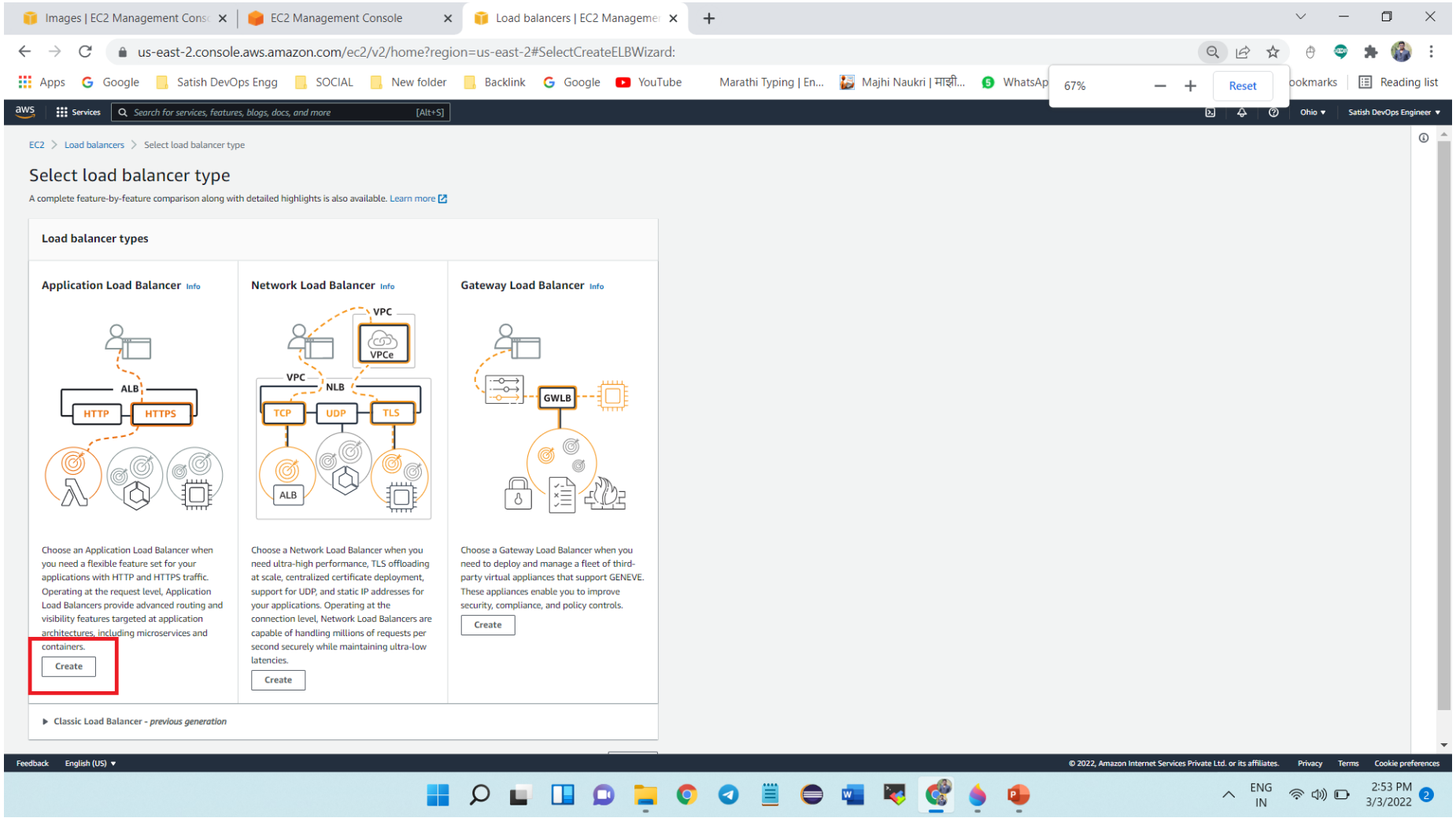
- Click on Create Target Group



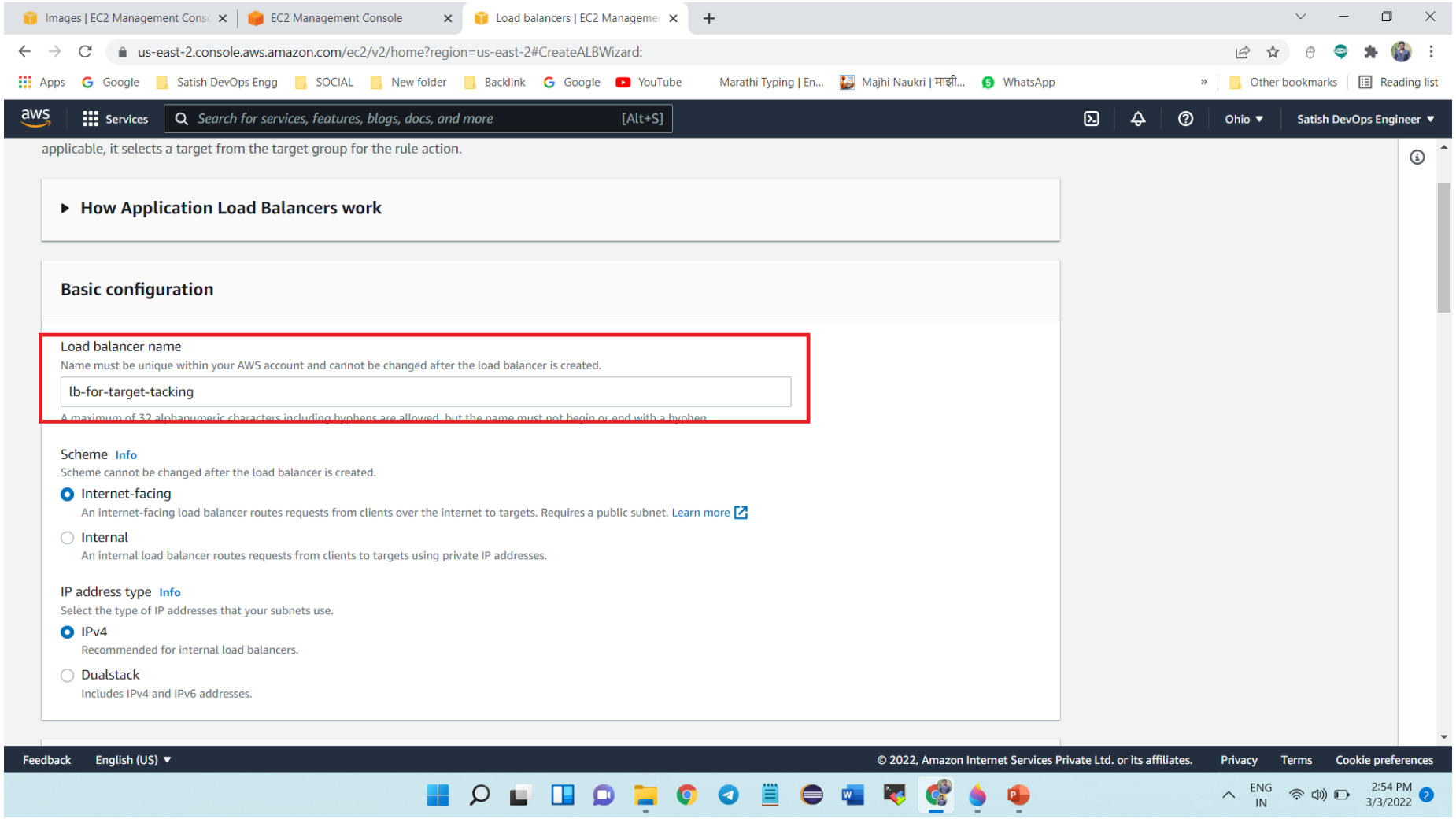
- Here our target group is created and there is no instance inside that target group.
- Also it is not connected to any Load Balancer



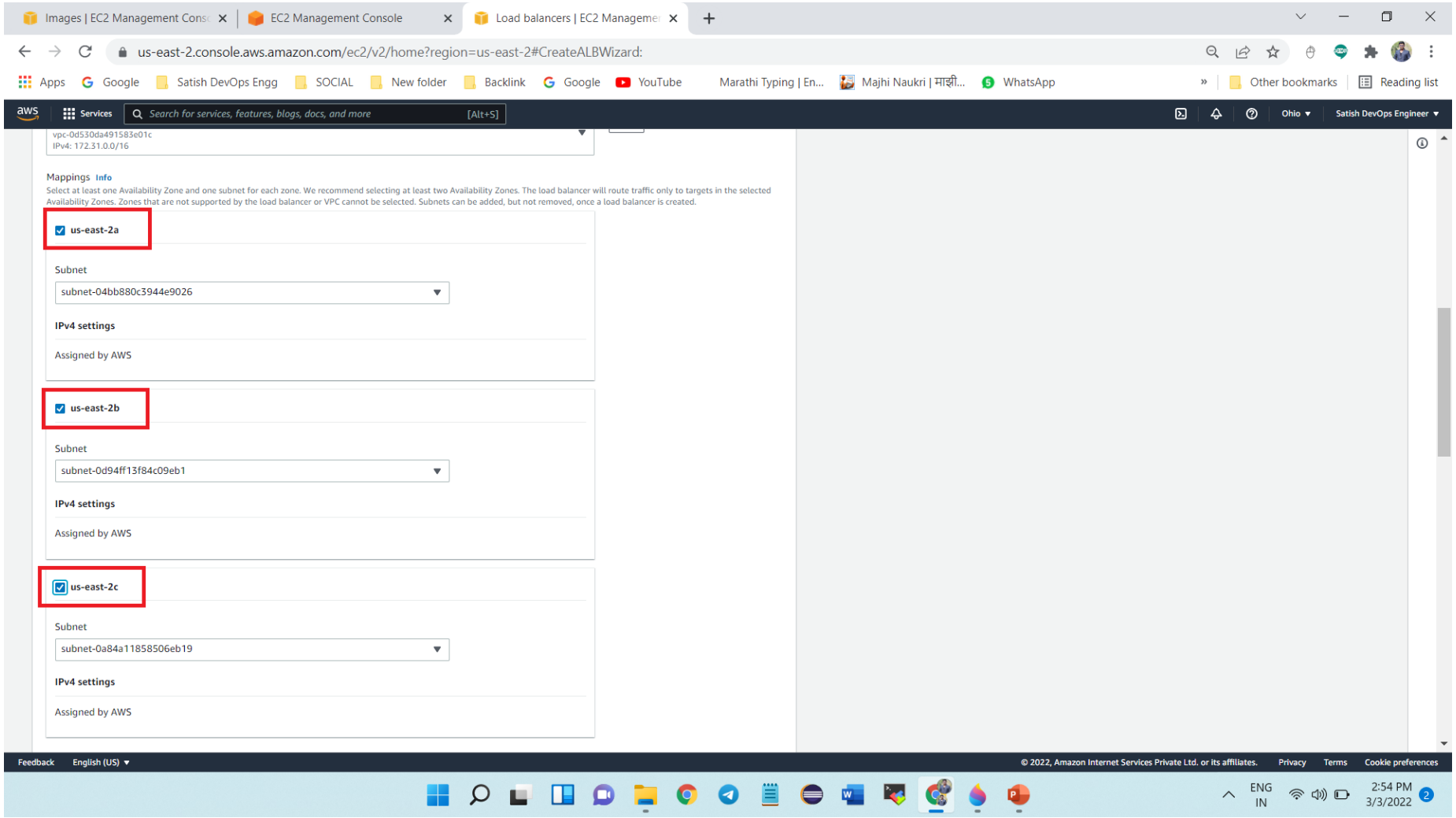
- Now go to Load Balancers
- Click on Create Load Balancer



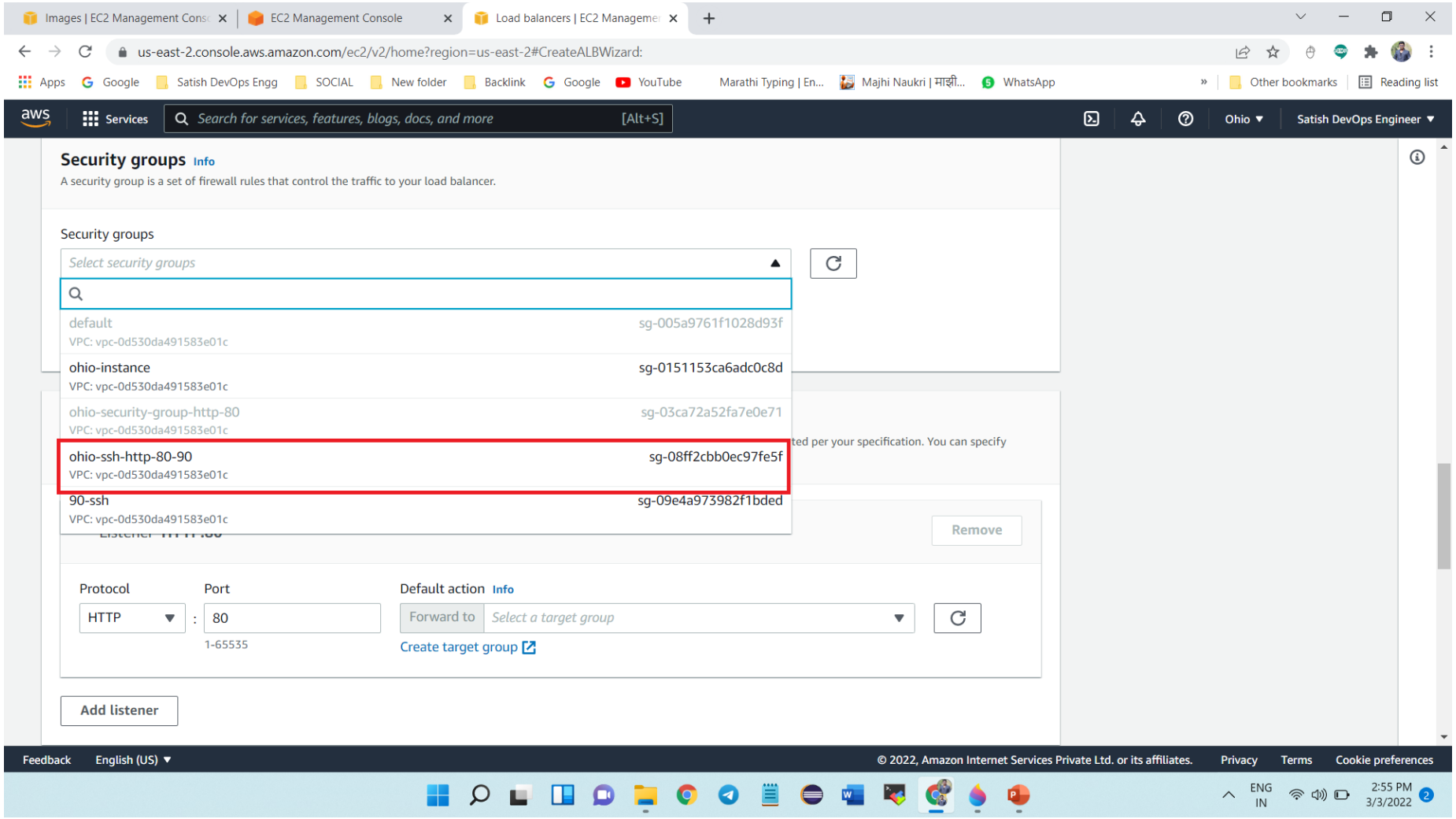
- Select Application Load Balancer



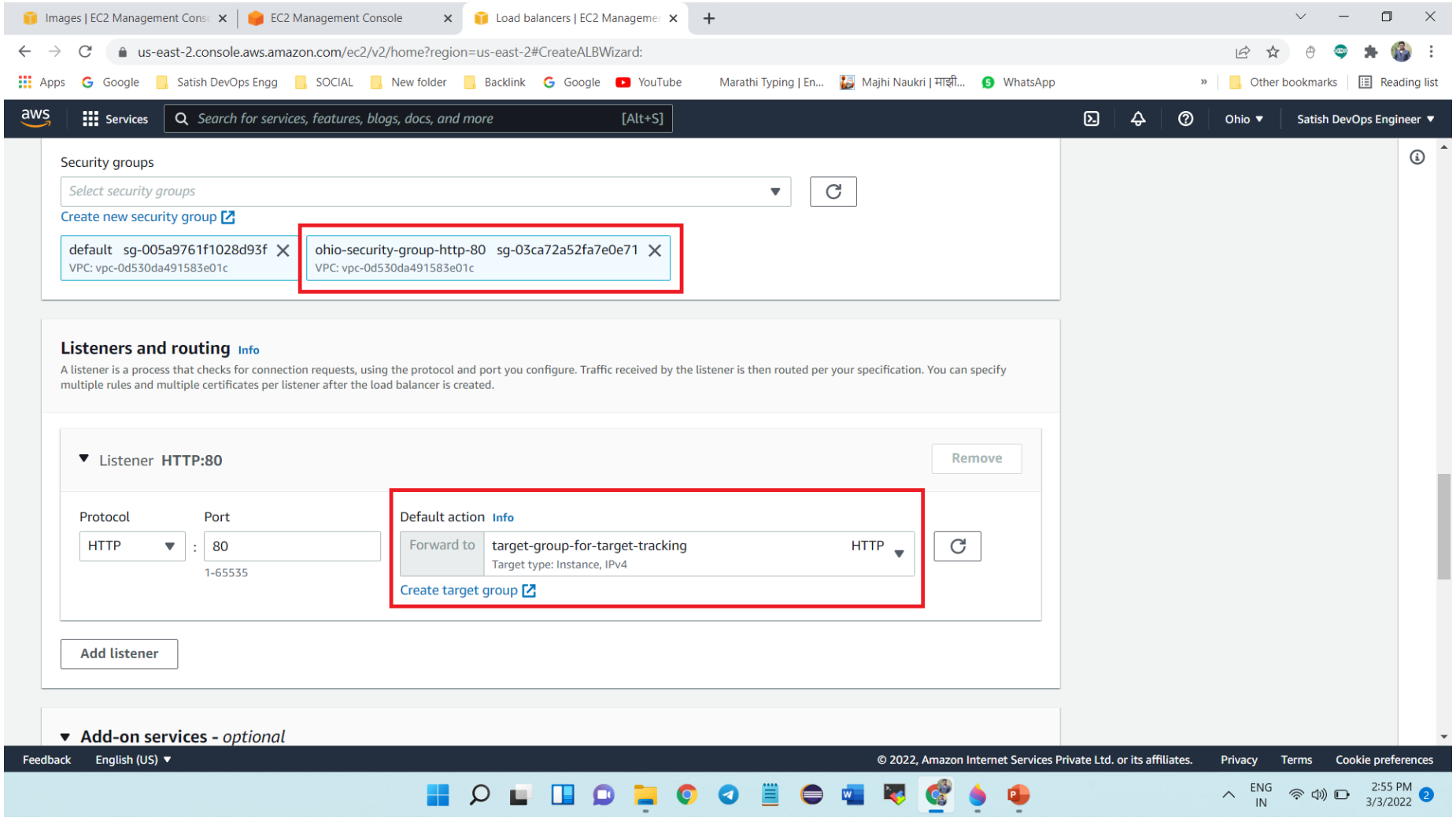
- Give name to Load Balancer



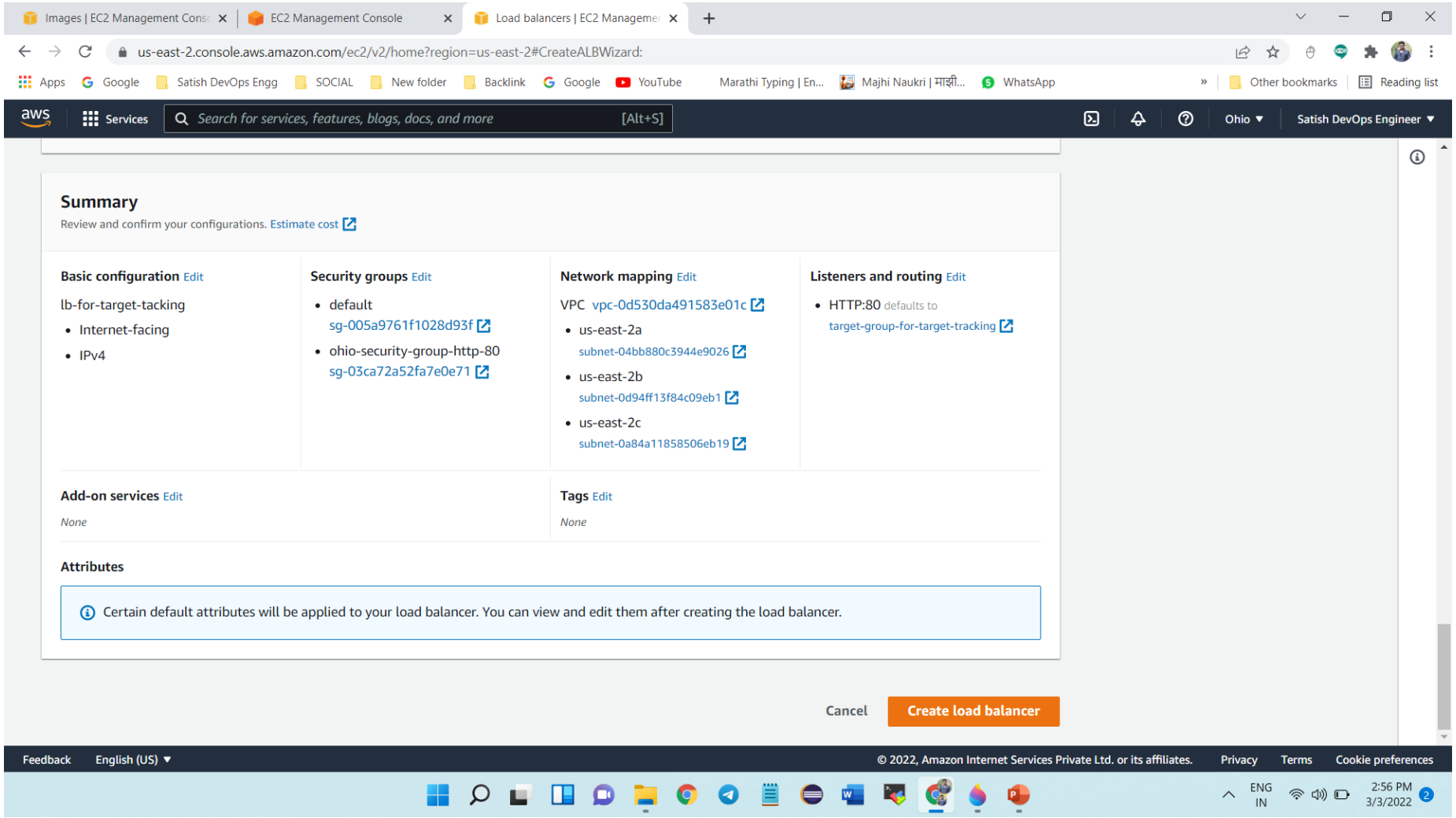
- Select all three zones.



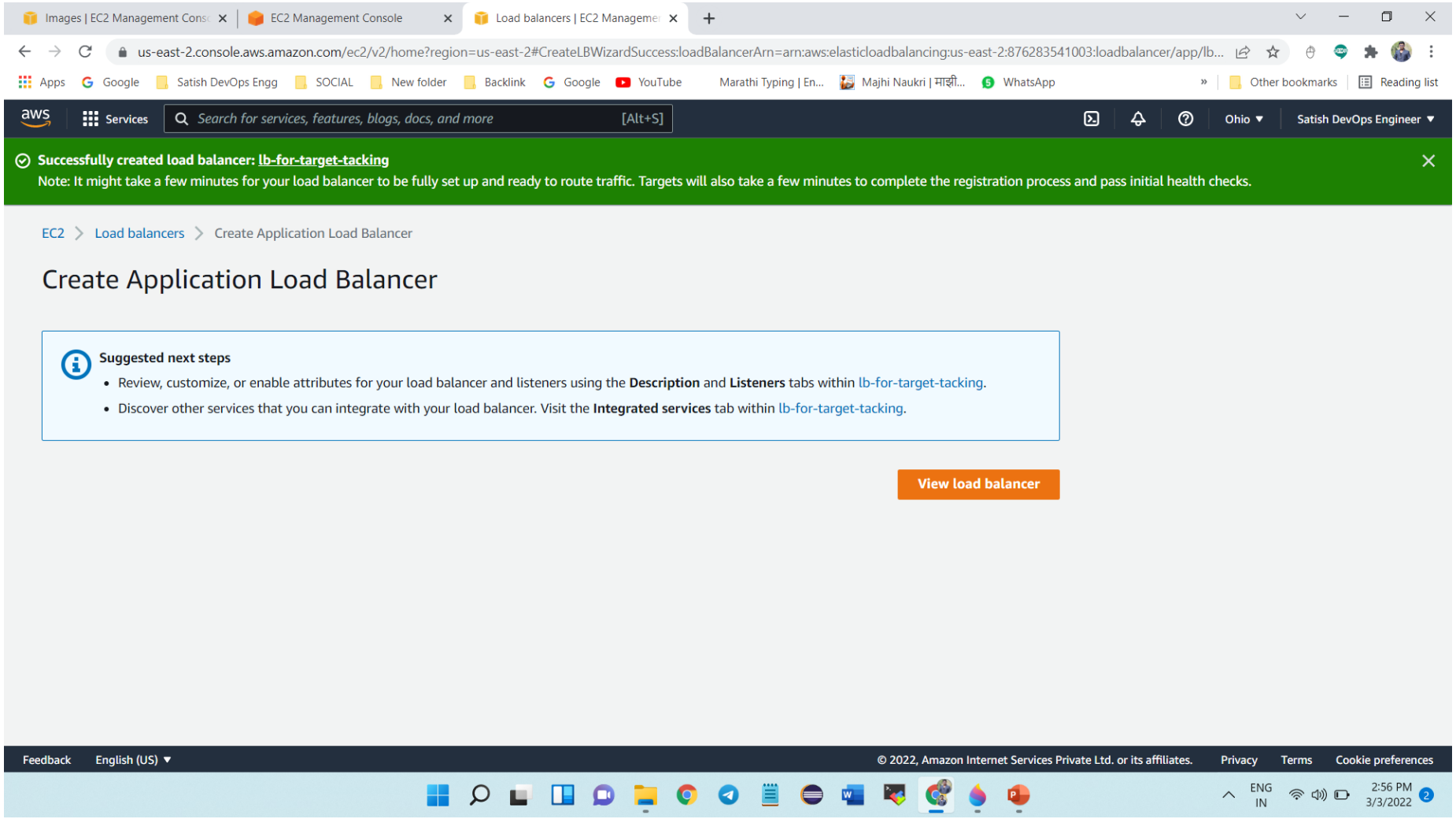
- Select security group in which **HTTP** port 80 is opened



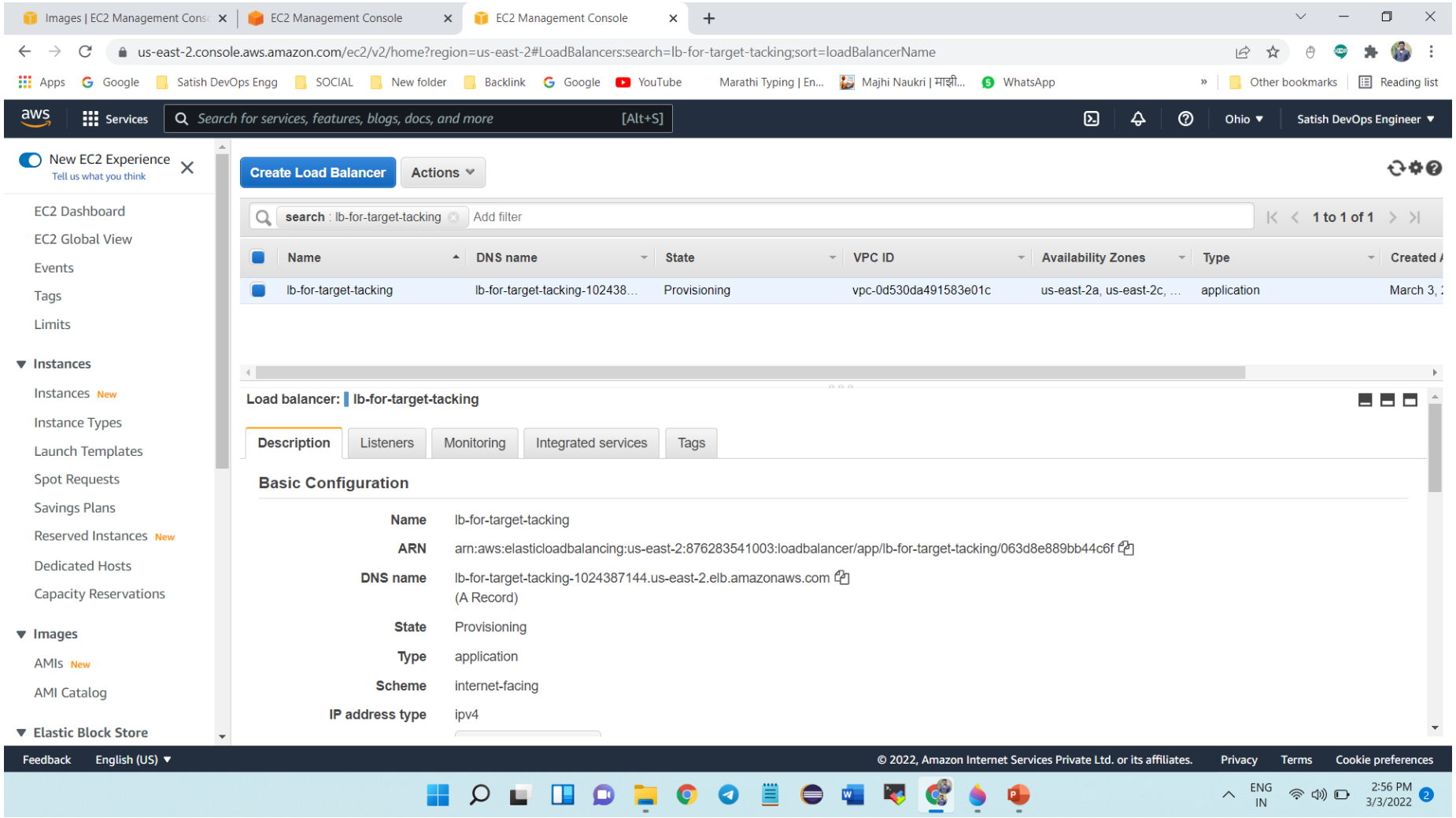
- Now in the listener select that the created target group

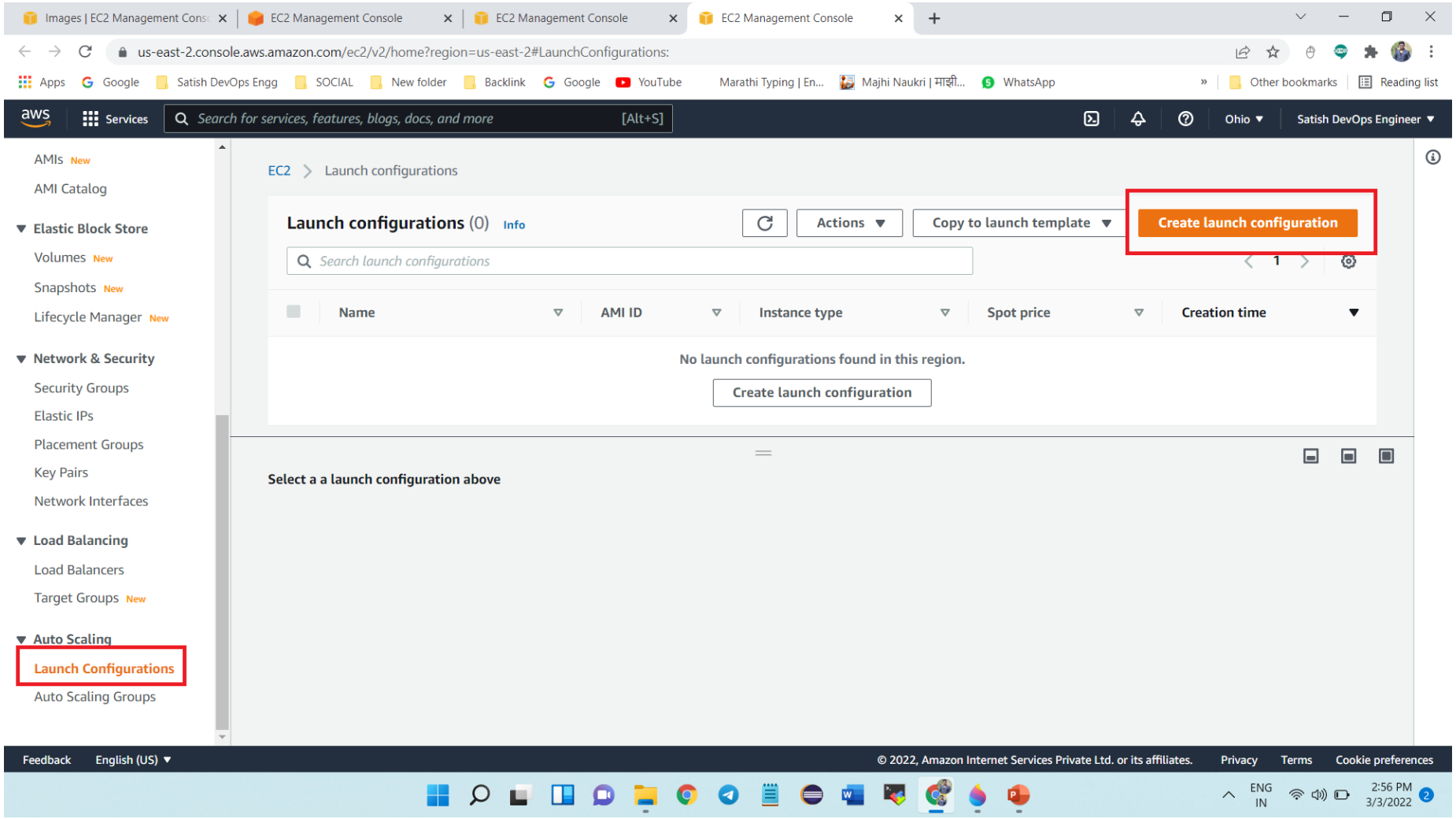


- Now click on Create Load Balancer

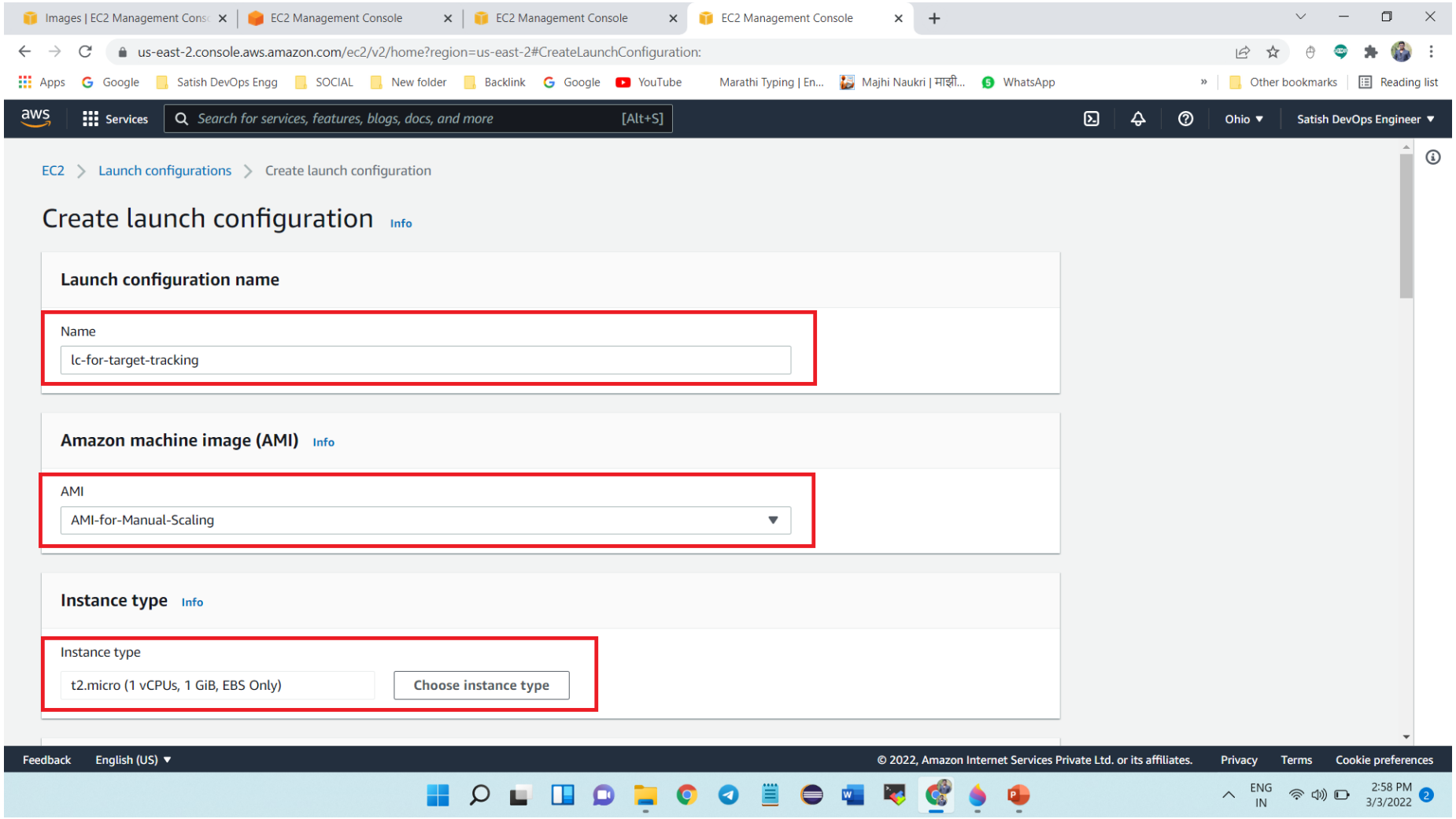


- Our Load Balancer is created successfully
- Click on View Load Balancer

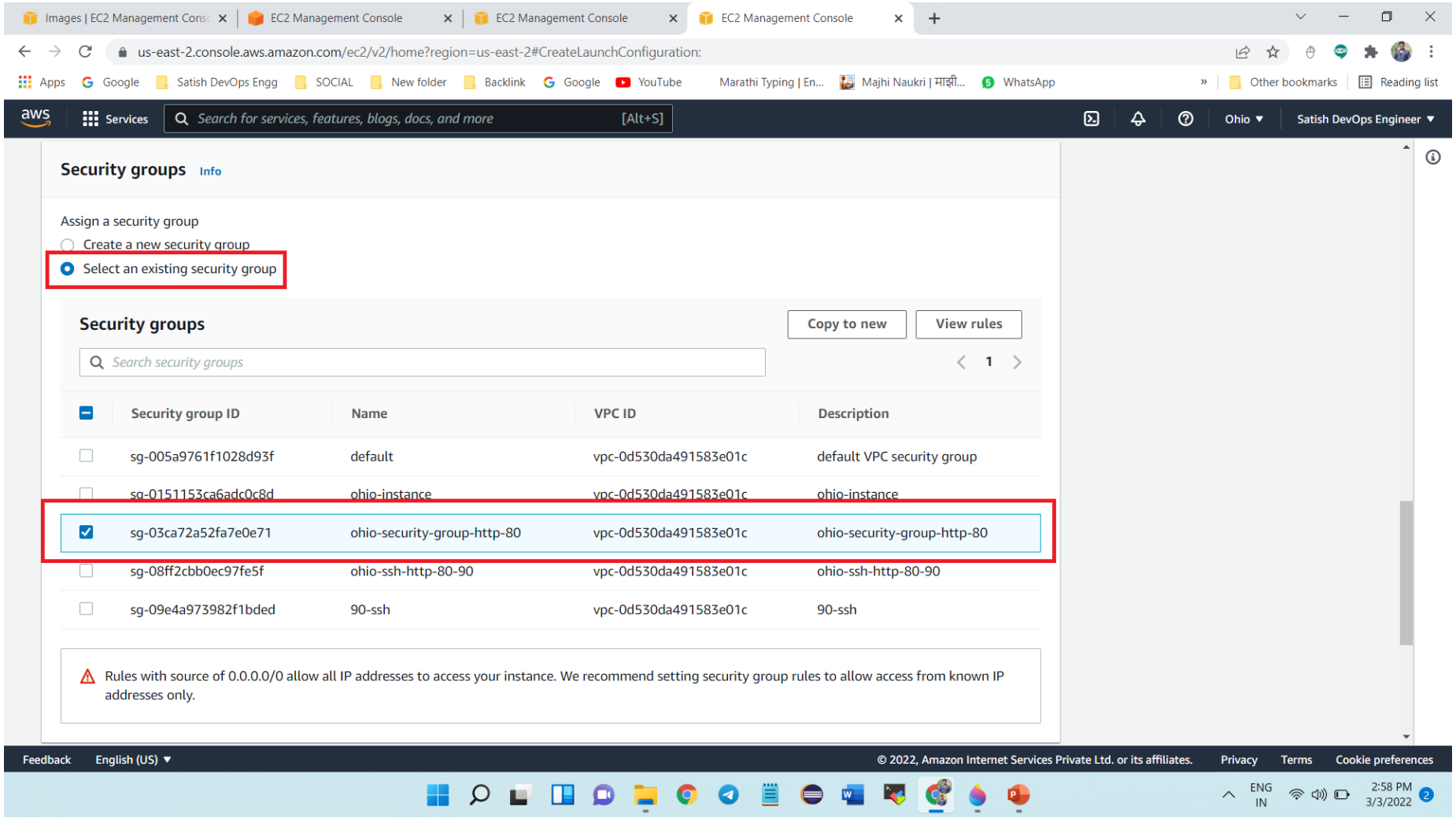

Images | EC2 Management Cons
EC2 Management Console
EC2 Management Console
us-east-2.console.aws.amazon.com/ec2/v2/home?region=us-east-2#LoadBalancers:search=lb-for-target-tacking;sort=loadBalancerName
Apps
Google
Satish DevOps Engg
SOCIAL
New folder
Backlink
Google
YouTube
Marathi Typing | En...
Majhi Naukri | माझी...
WhatsApp
Other bookmarks
Reading list
aws
Services
Search for services, features, blogs, docs, and more
[Alt+S]
Ohio
Satish DevOps Engineer
New EC2 Experience
Tell us what you think
EC2 Dashboard
EC2 Global View
Events
Tags
Limits
Instances
Instances New
Instance Types
Launch Templates
Spot Requests
Savings Plans
Reserved Instances New
Dedicated Hosts
Capacity Reservations
Images
AMIs New
AMI Catalog
Elastic Block Store
Create Load Balancer
Actions
search : lb-for-target-tacking
Add filter
1 to 1 of 1
Name
DNS name
State
VPC ID
Availability Zones
Type
Created
lb-for-target-tacking
lb-for-target-tacking-102438...
Provisioning
vpc-0d530da491583e01c
us-east-2a, us-east-2c, ...
application
March 3,
Load balancer: lb-for-target-tacking
Description
Listeners
Monitoring
Integrated services
Tags
Basic Configuration
Name
lb-for-target-tacking
ARN
arn:aws:elasticloadbalancing:us-east-2:876283541003:loadbalancer/app/lb-for-target-tacking/063d8e889bb44c6f
DNS name
lb-for-target-tacking-1024387144.us-east-2.elb.amazonaws.com
(A Record)
State
Provisioning
Type
application
Scheme
internet-facing
IP address type
ipv4
Feedback
English (US)
© 2022, Amazon Internet Services Private Ltd. or its affiliates.
Privacy
Terms
Cookie preferences
ENG
IN
2:56 PM
3/3/2022



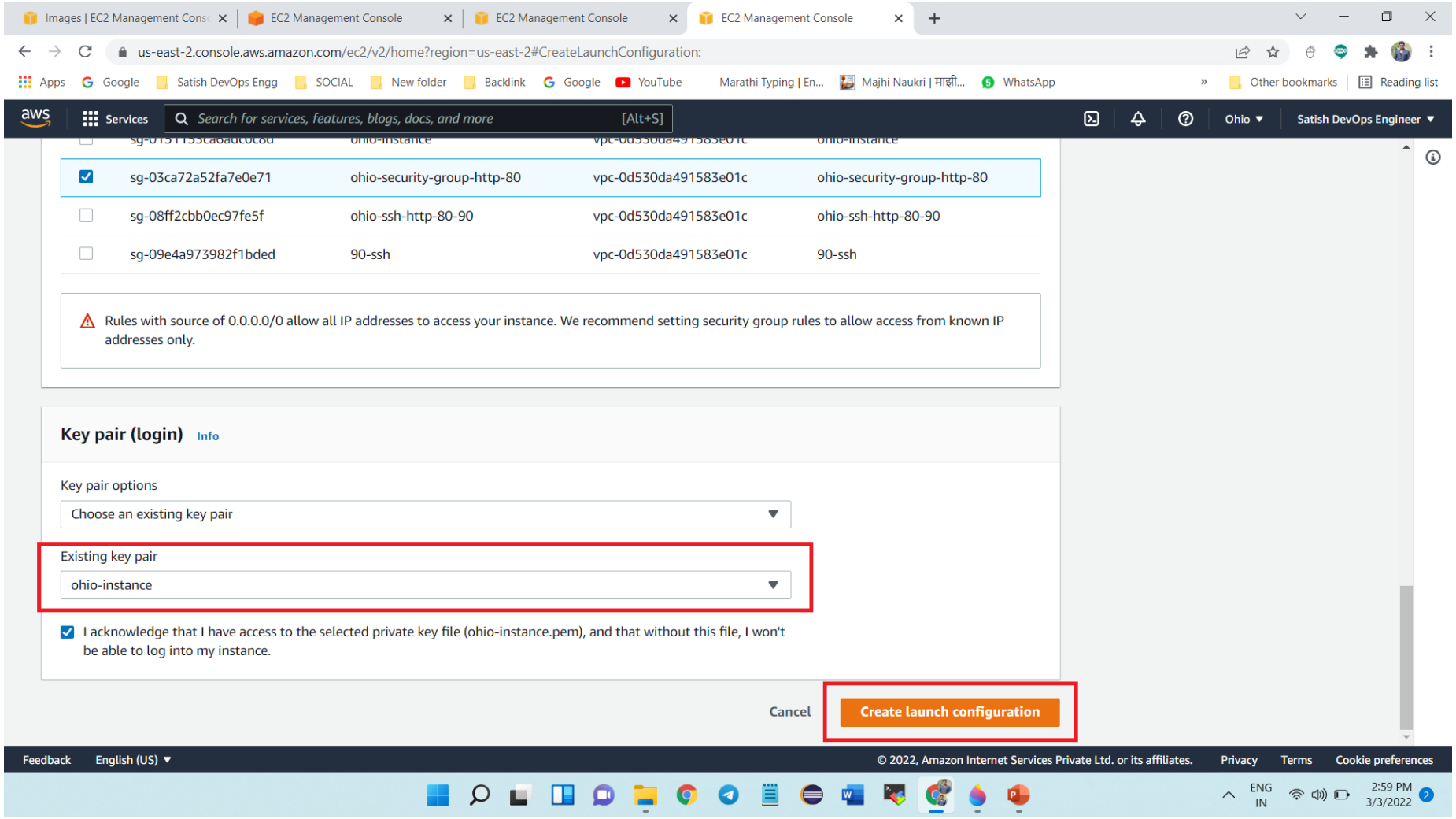
- Now go to Launch Configuration
- Click on Create Launch Configuration



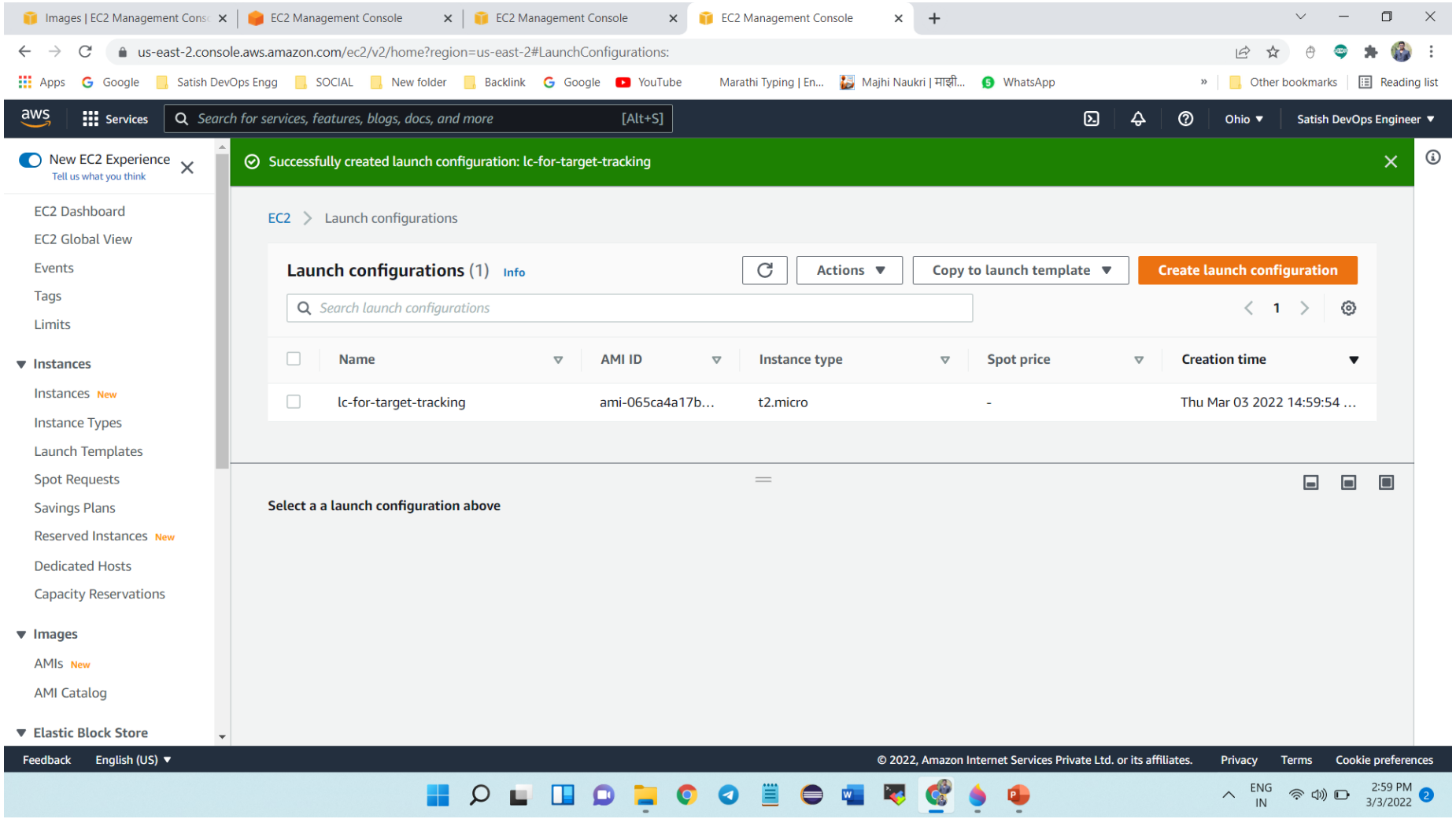
- Give name to Launch Configuration
- Select Particular AMI
- Select Instance Type as t2.micro



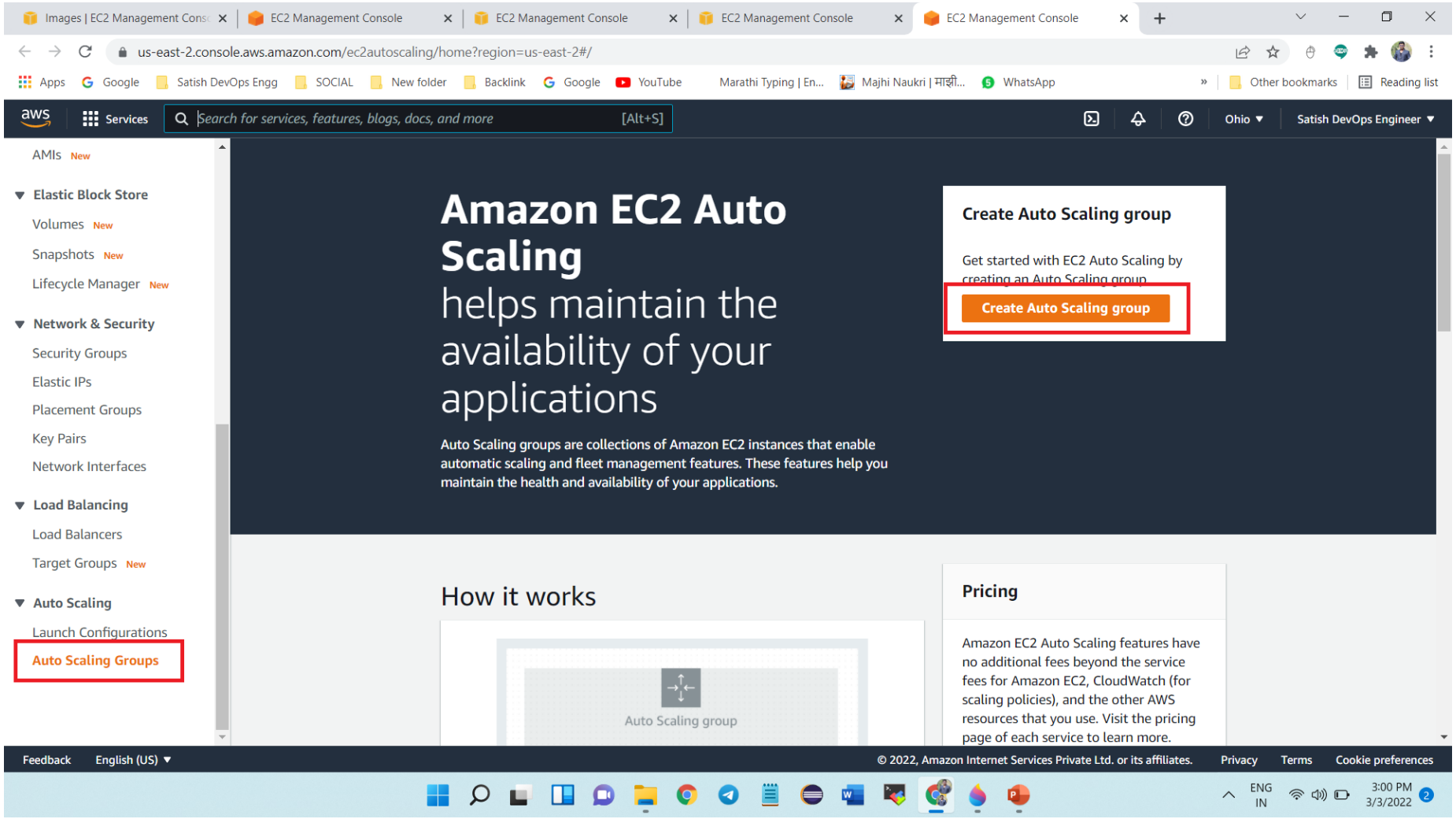
- Select the same security group in which HTTP port 80 is opened



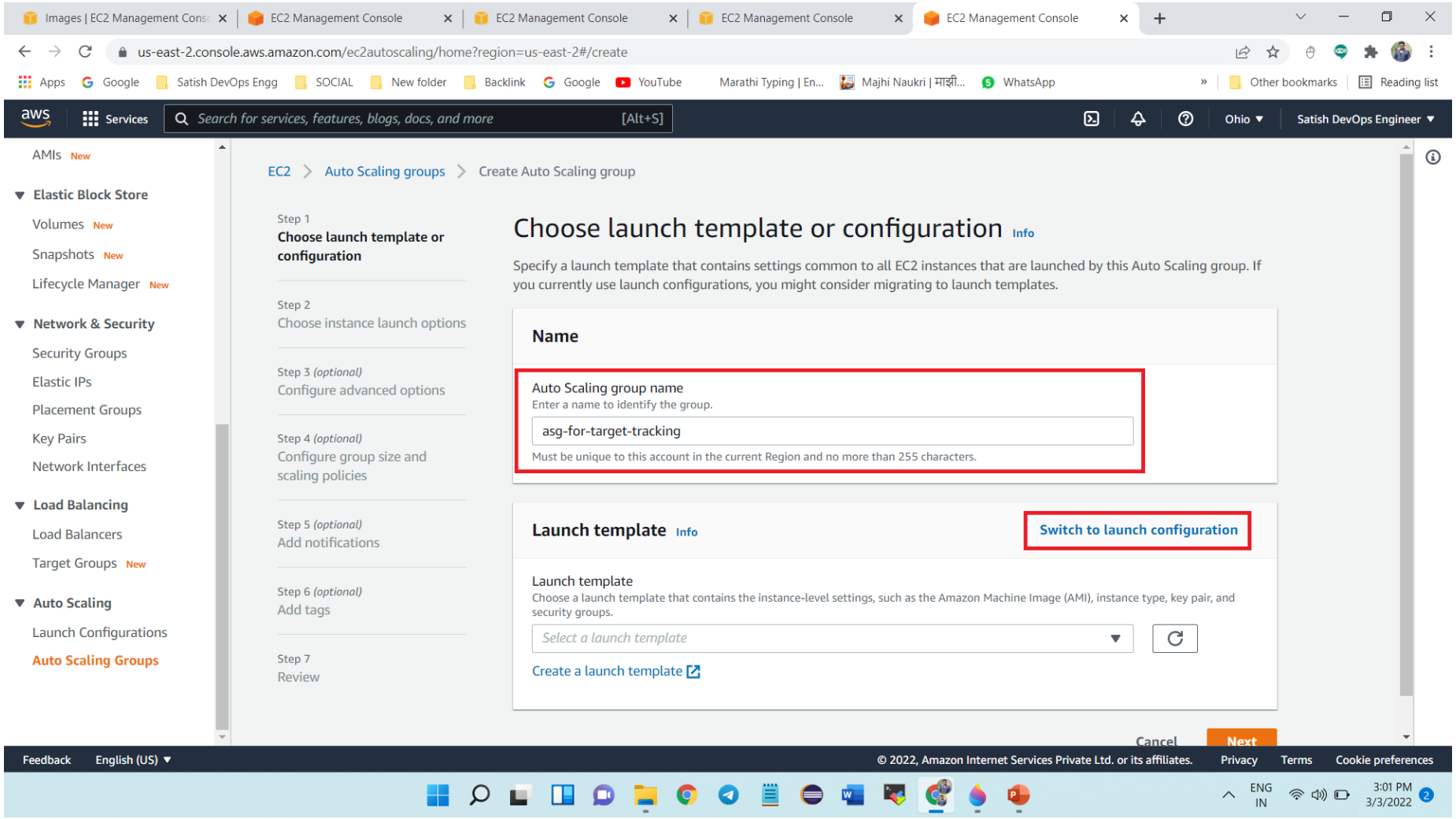
- Select key pair also
- Then click on Create Launch Configuration



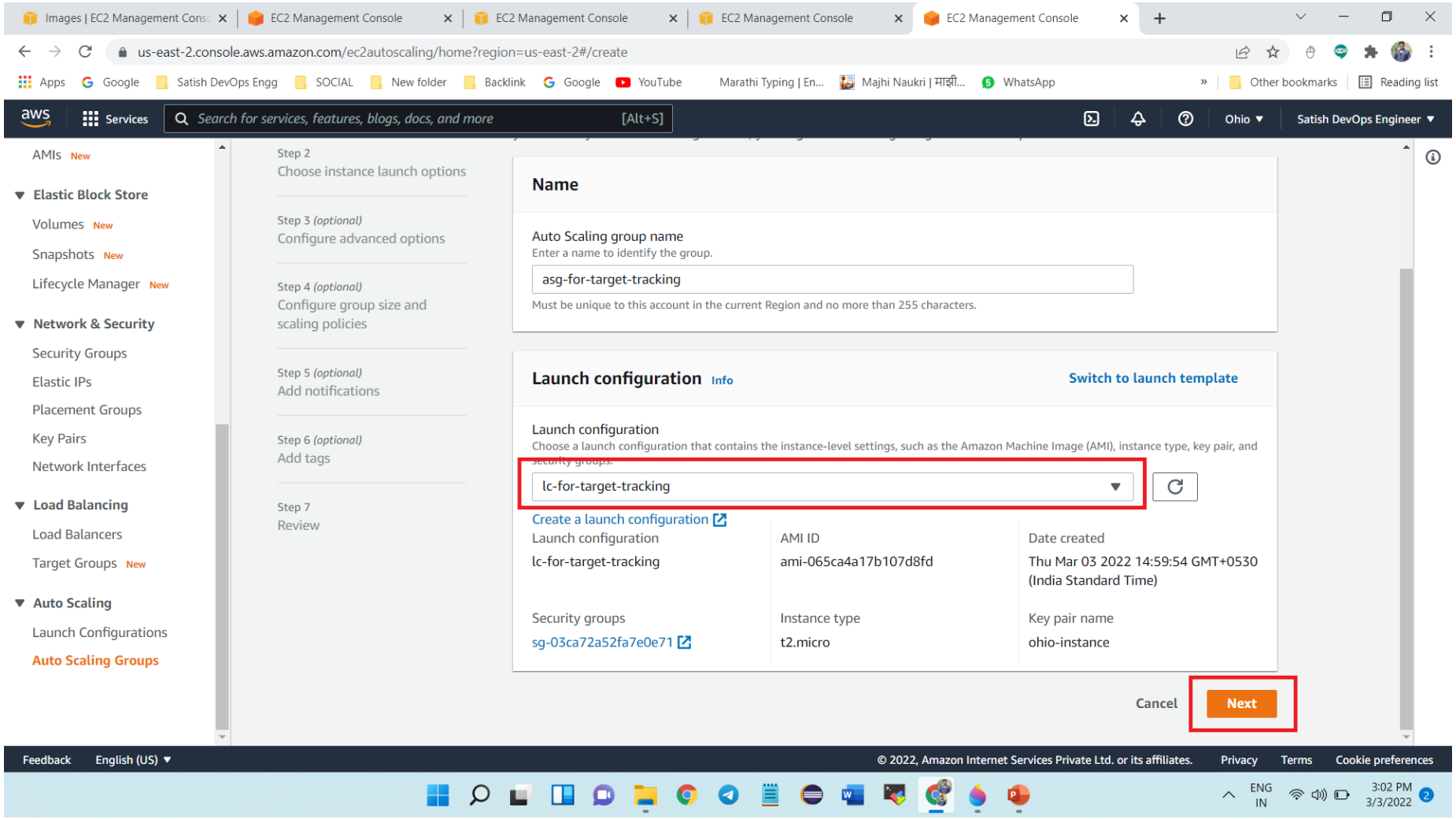
- Our Launch Configuration is created successfully



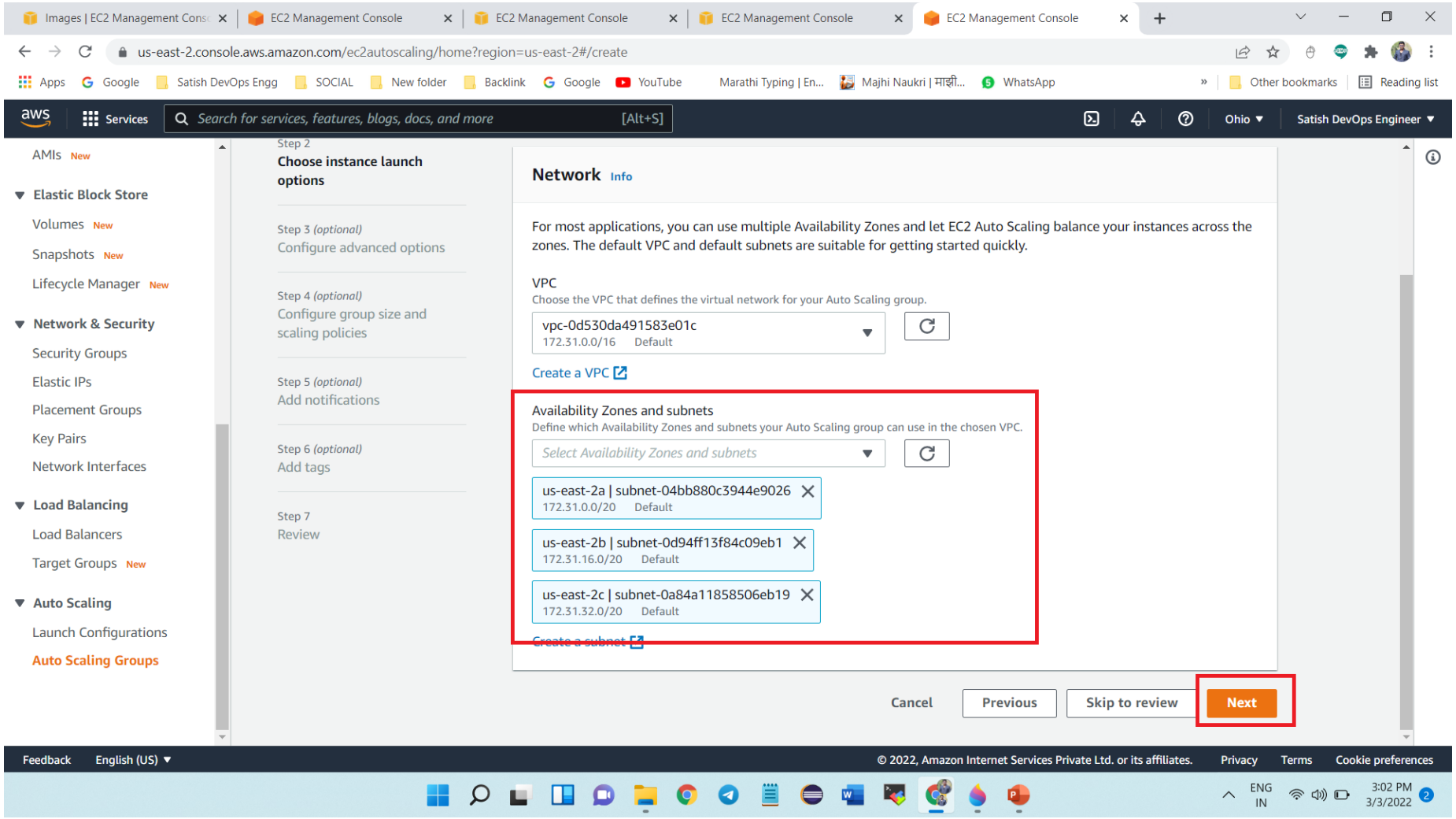
- Now go to Auto Scaling Group
- Click on Create Auto Scaling Group



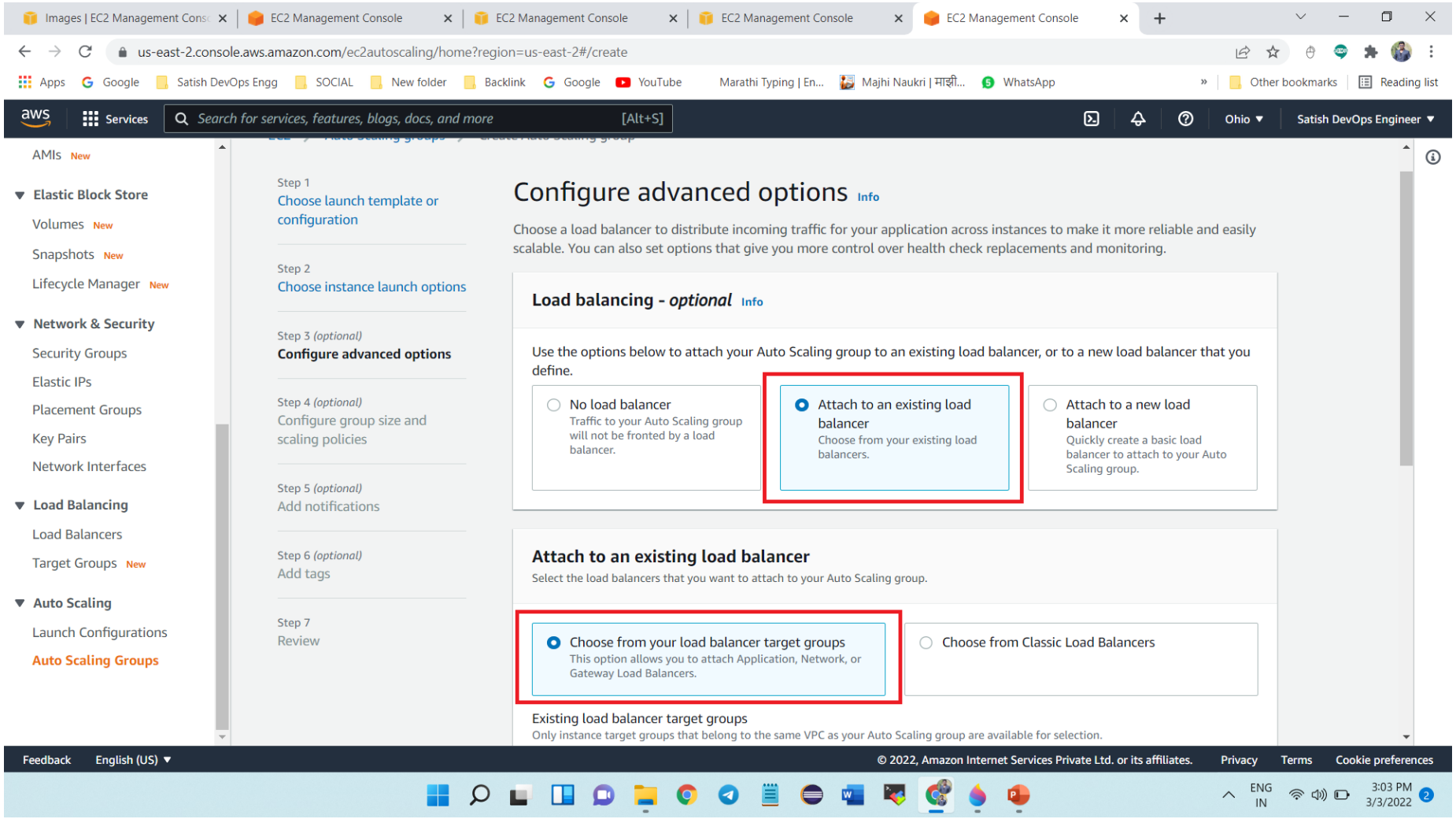
- Give name to ASG
- Then click on Switch to Launch Configuration



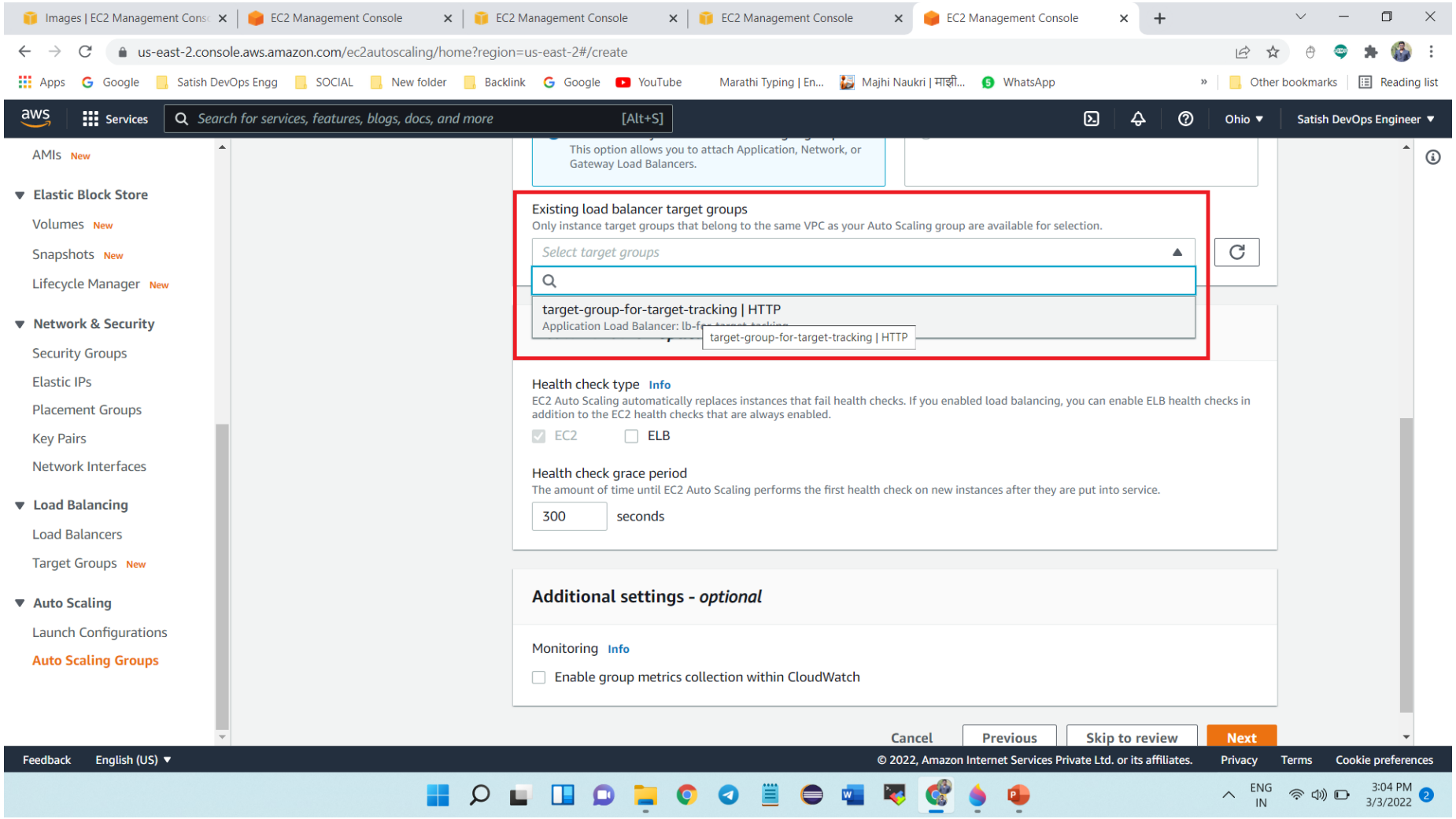
- Select launch configuration
- Then click on next



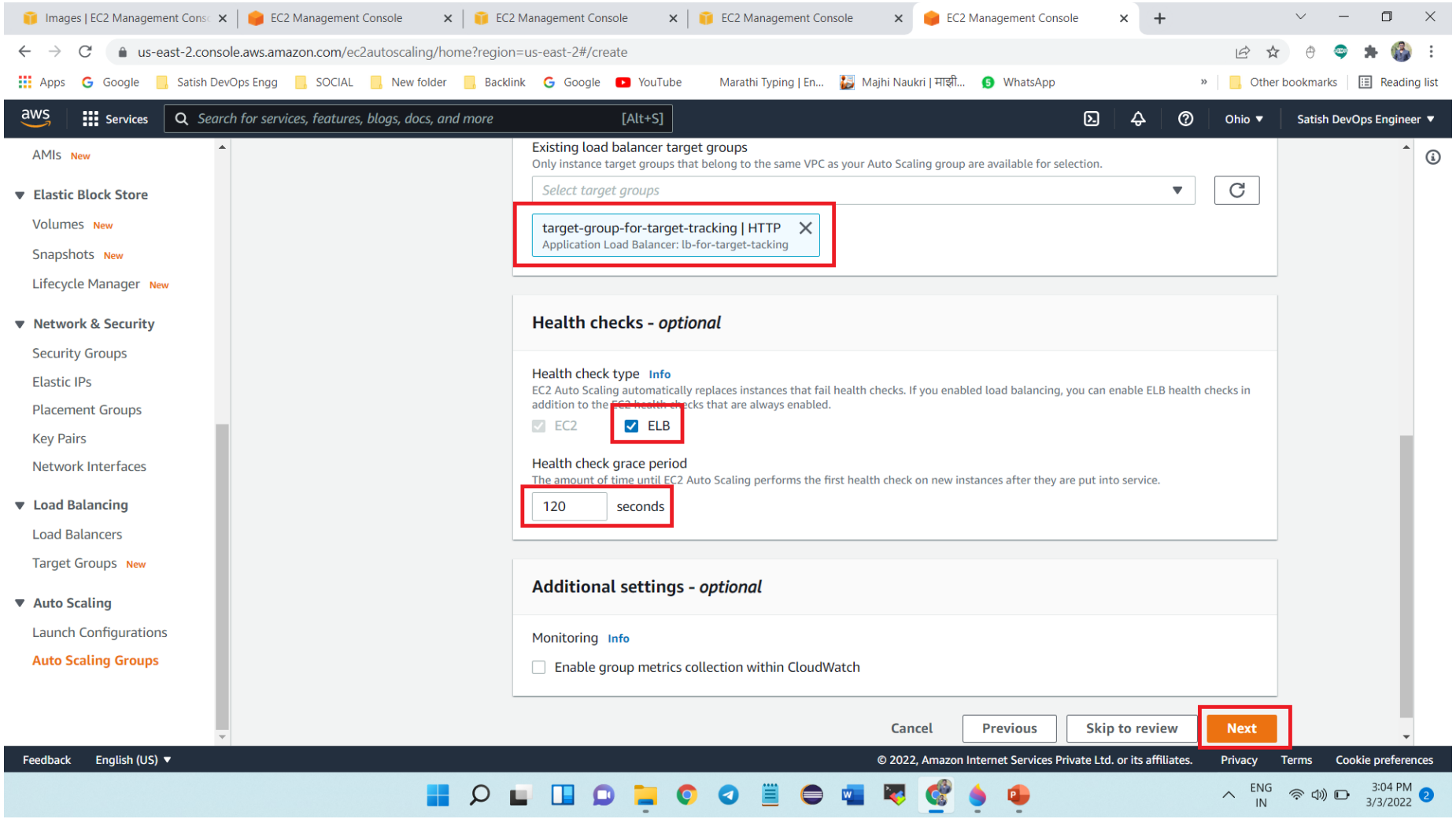
- Select all availability zones
- Then click on Next



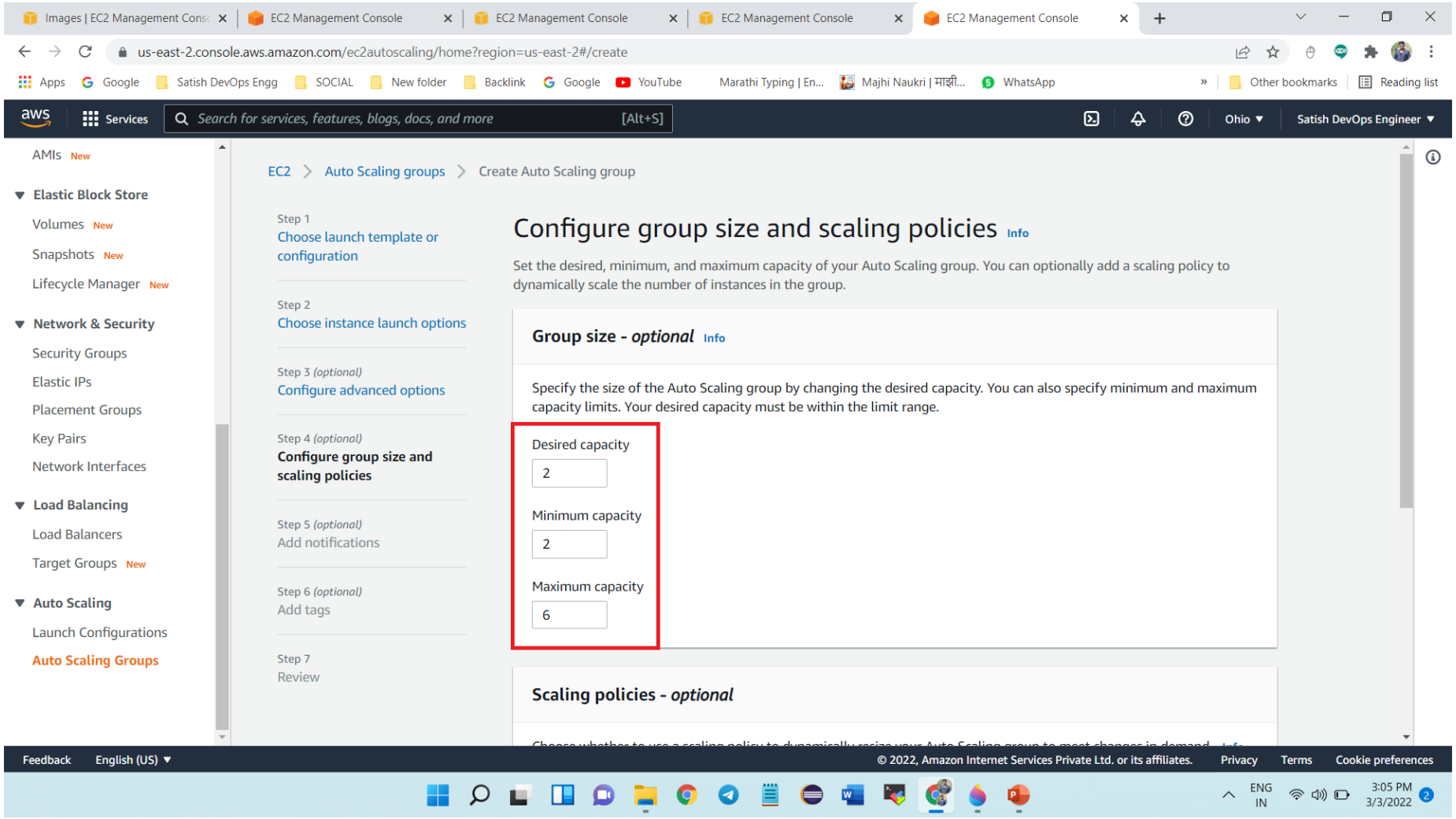
- Select **Attach to an Existing Load Balancer**
- Also select **Choose from your load balancer target groups**



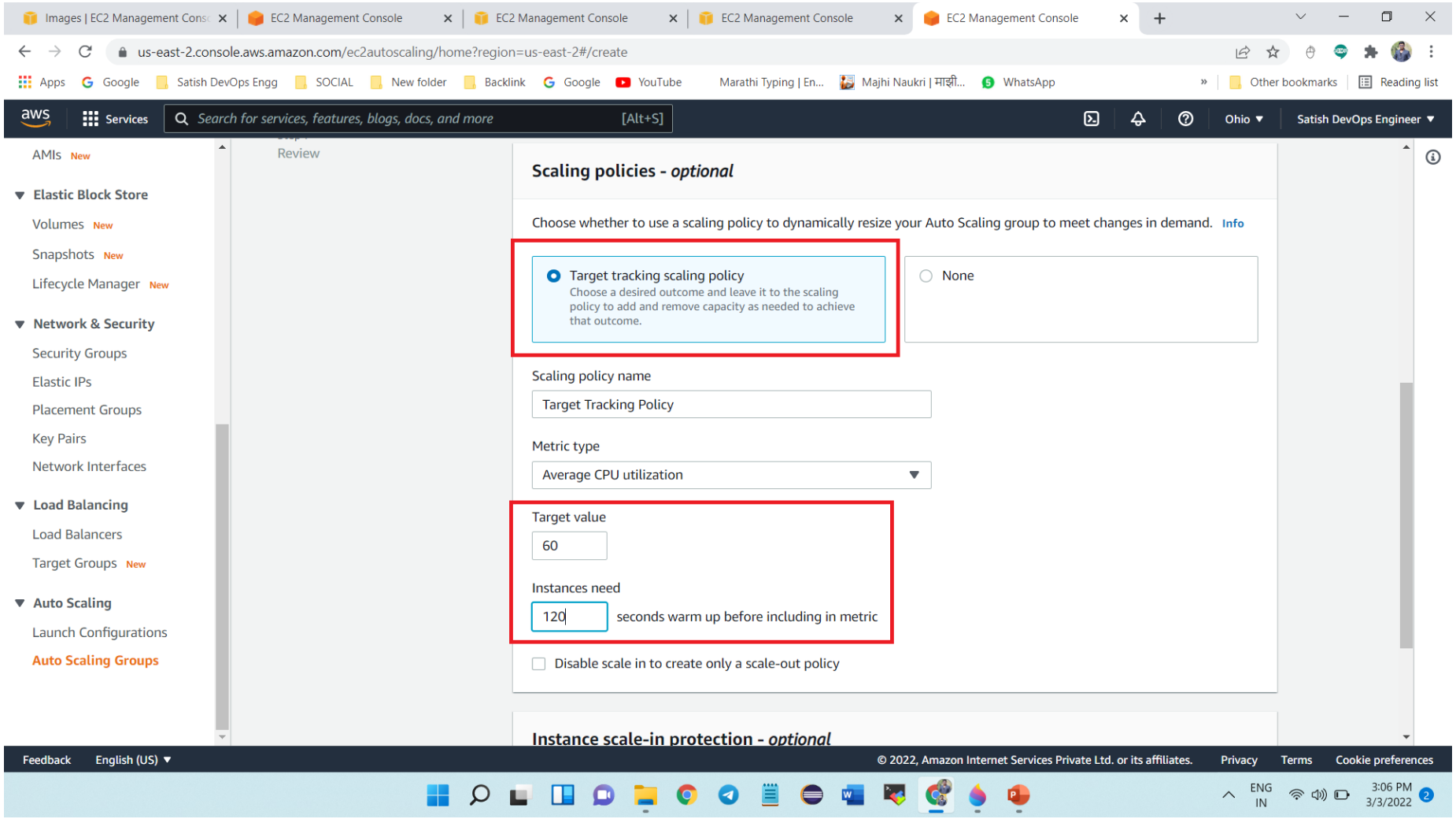
- Select that existing load balancer target group



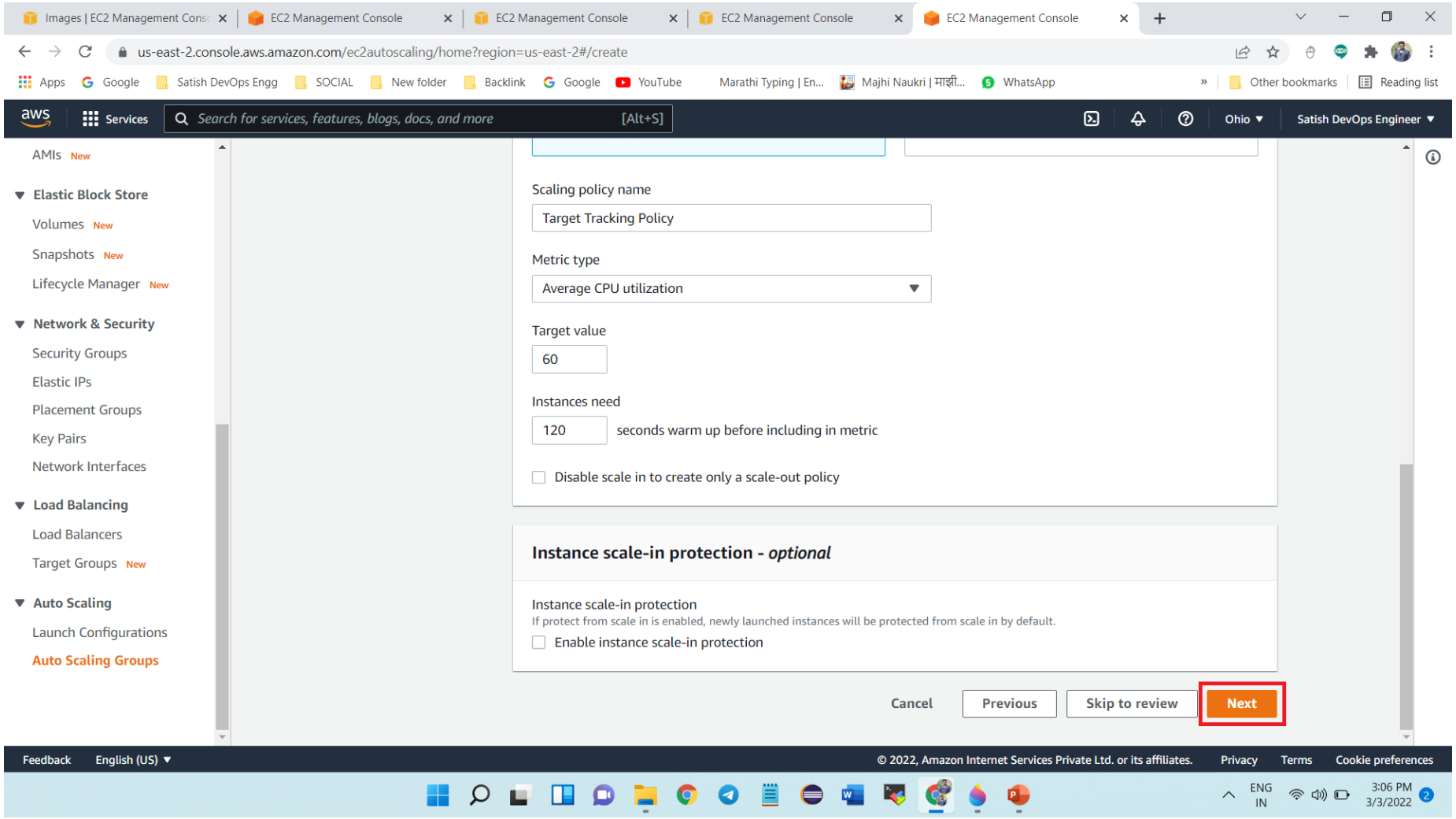
- Click on **ELB**
- Change Grace period to 120 Seconds
- Then click on Next



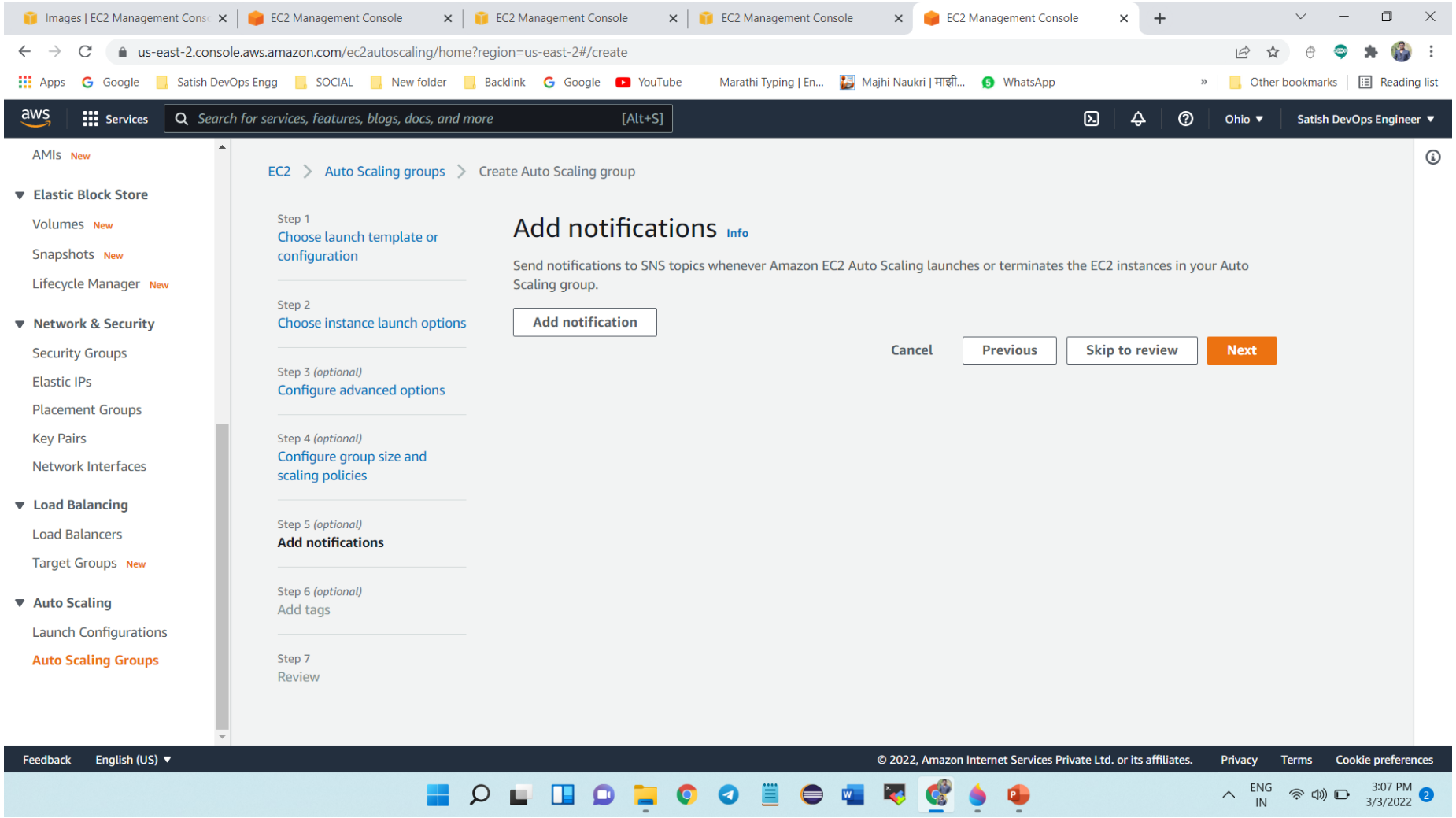
- Now set desired capacity as 2
- Minimum capacity as 2
- Maximum capacity as 6



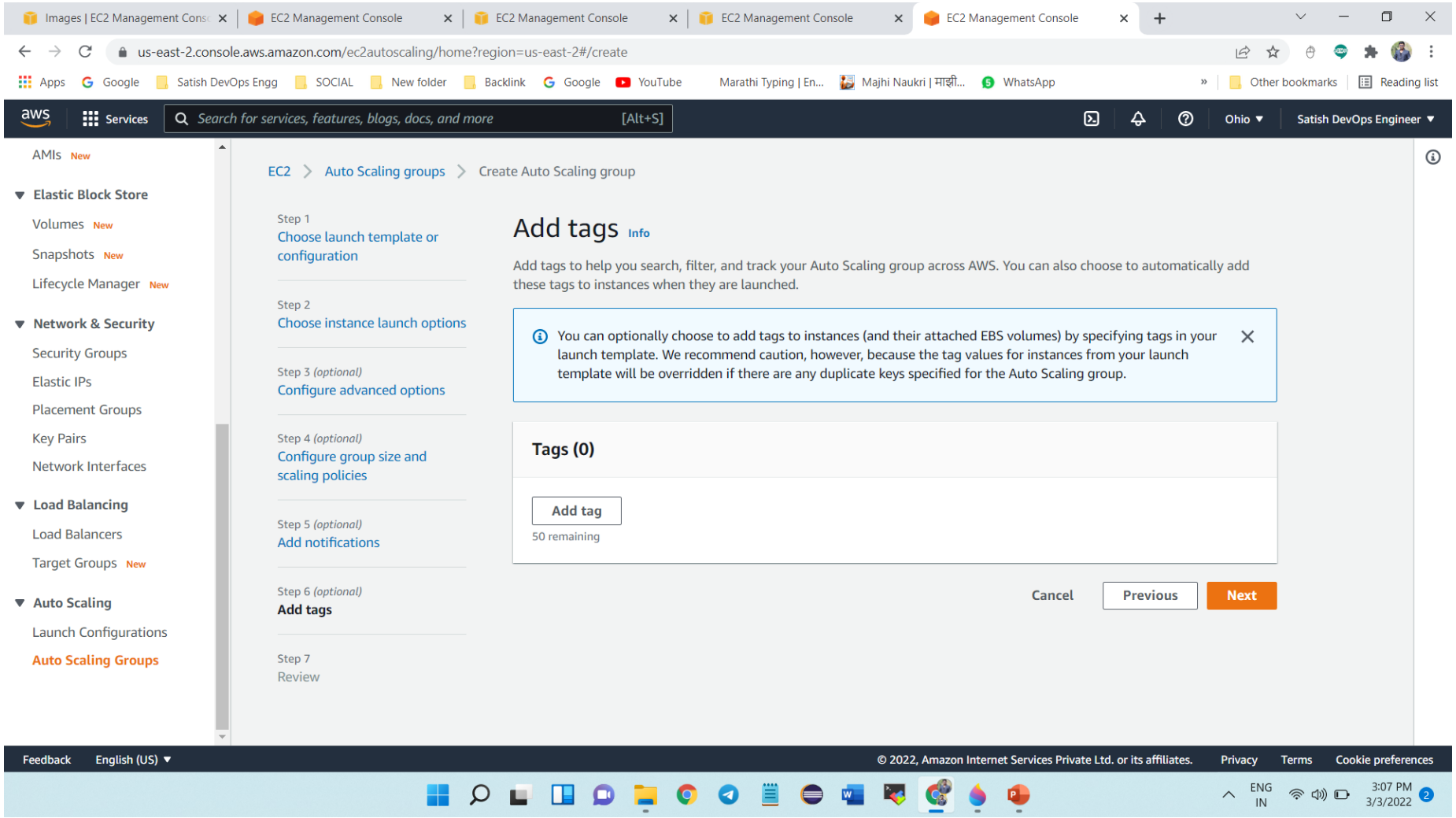
- Select Target Tracking Policy
- Use Target Value as 60
- Warm up time as 120 Seconds



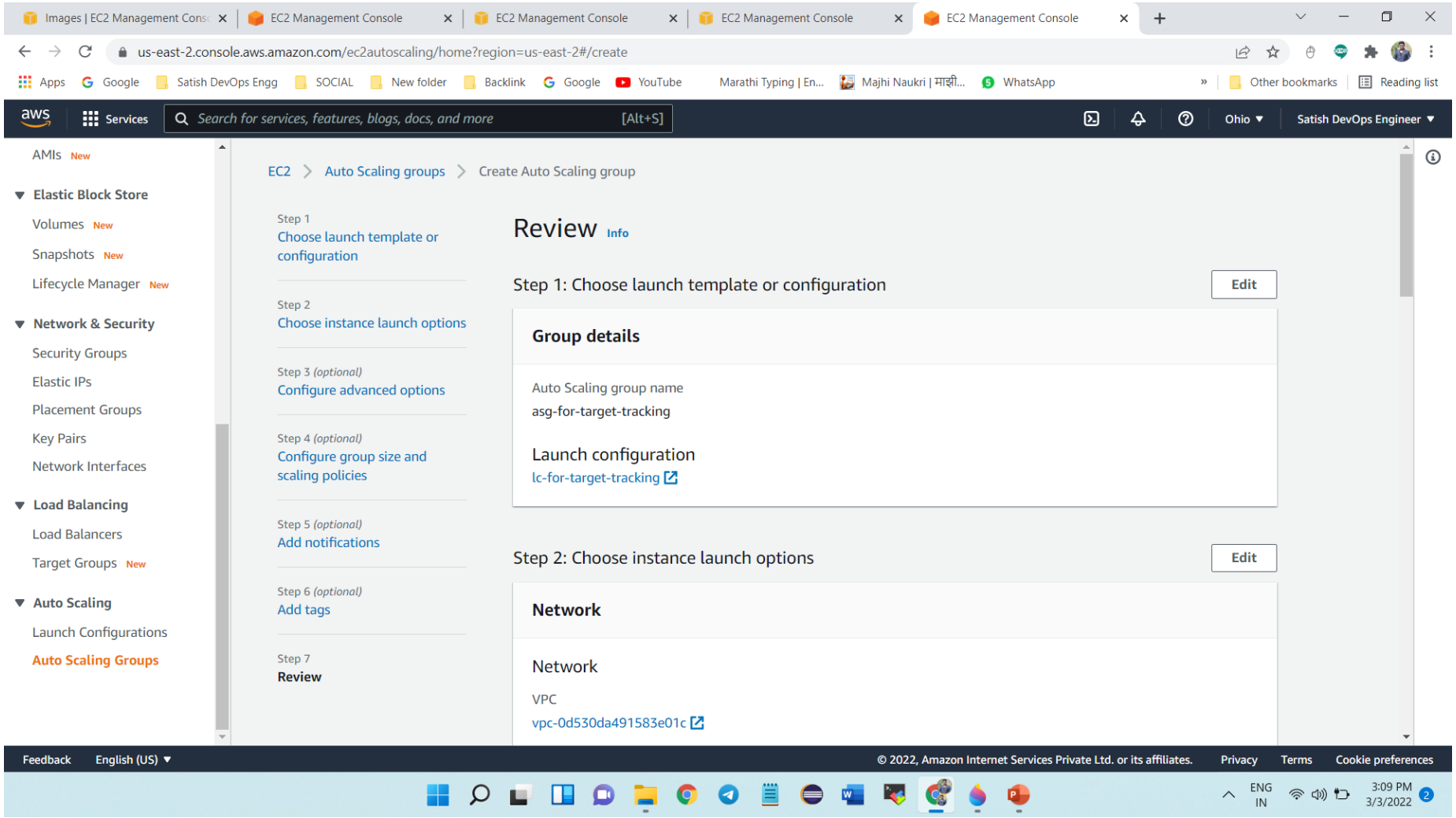
- Now click on Next



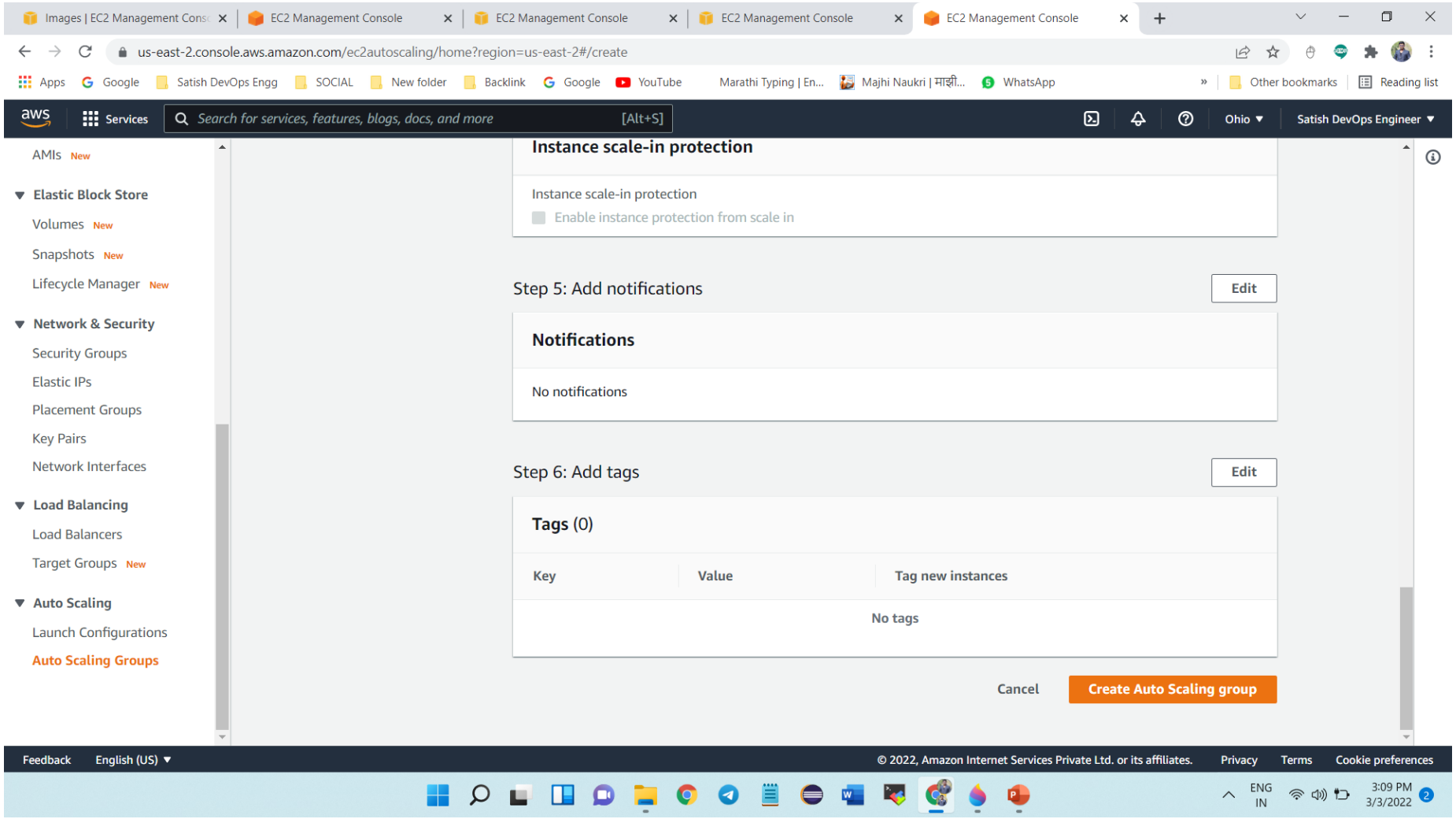
- Click on Next



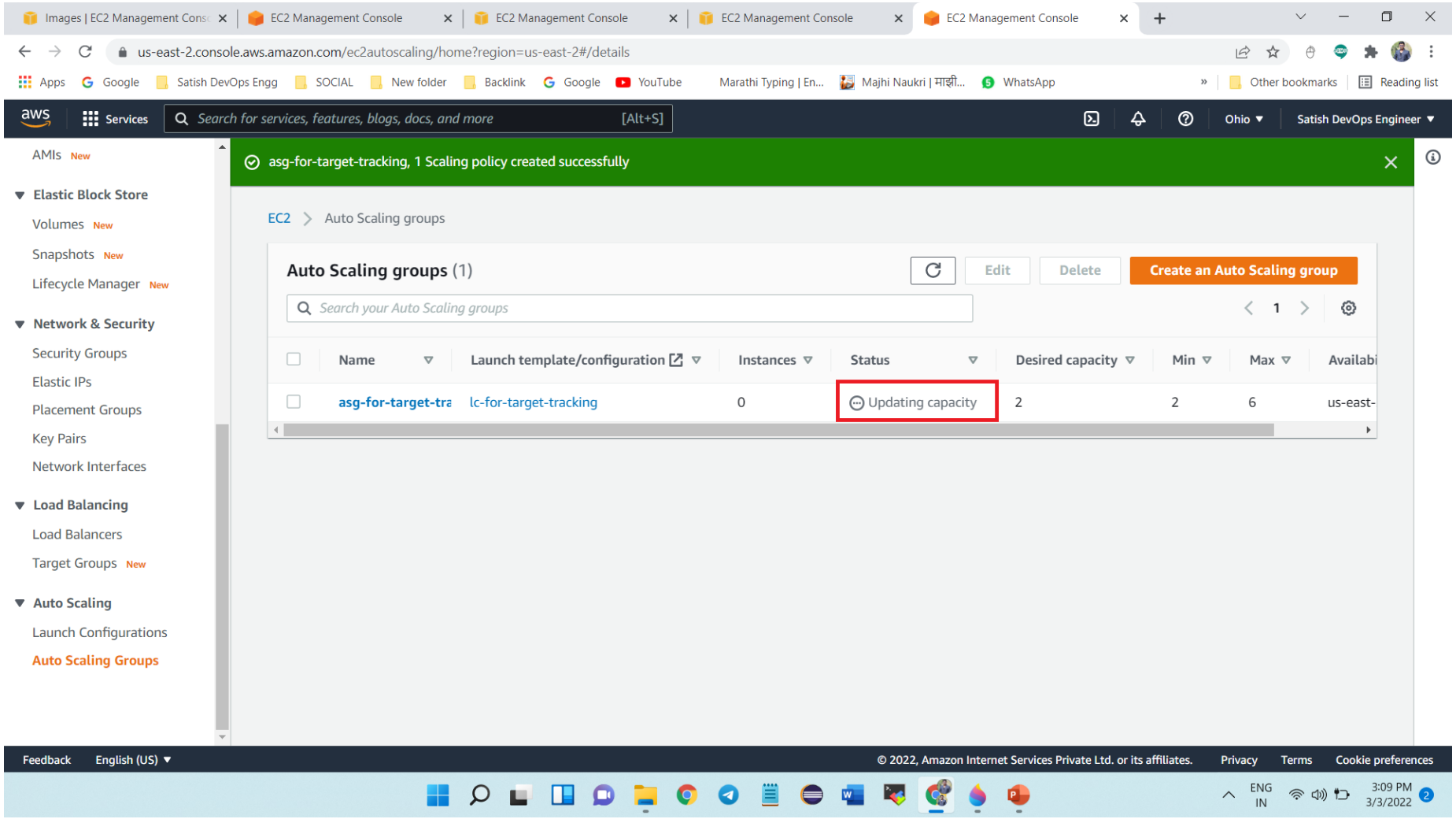
- Click on Next



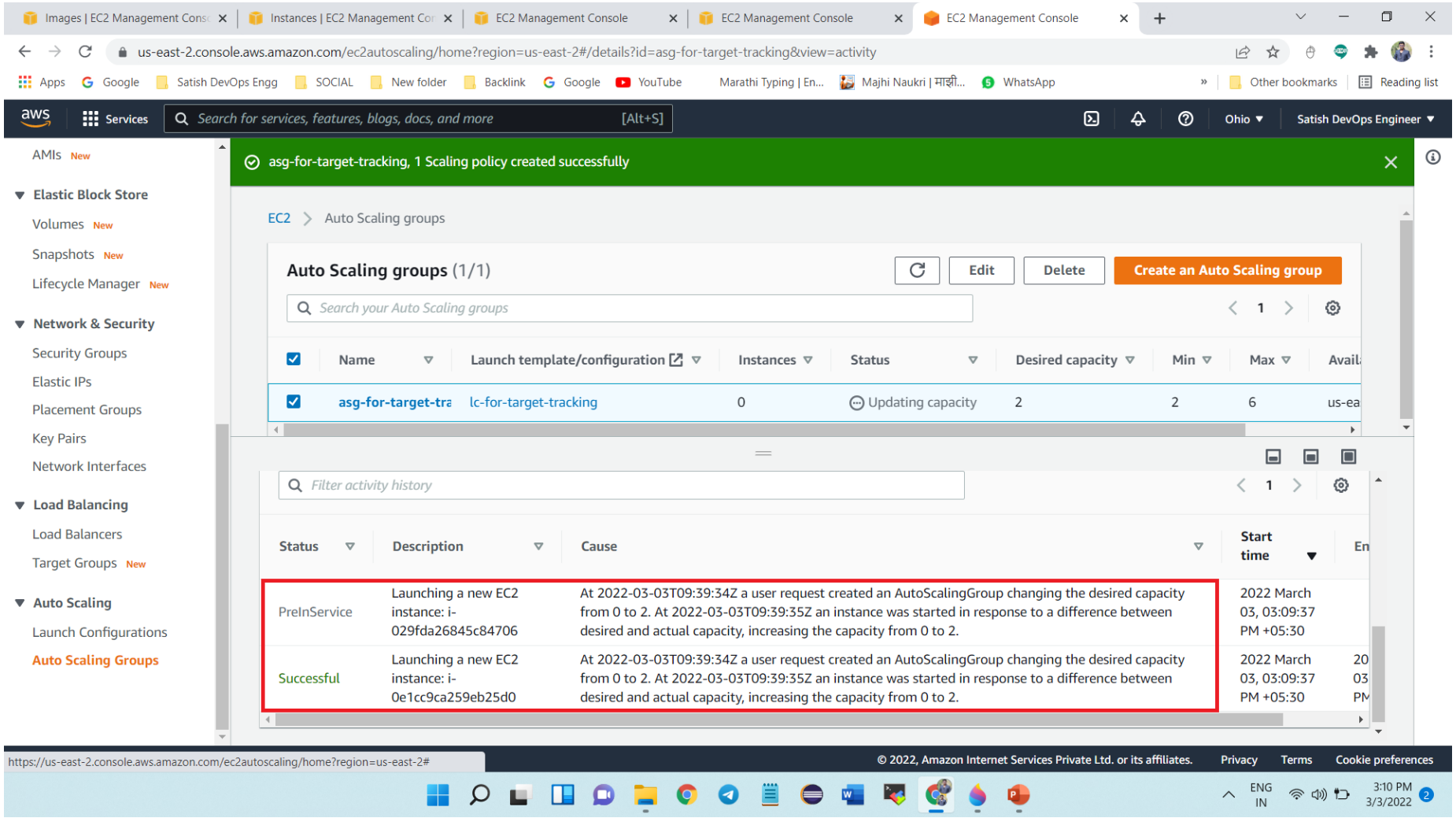
- Review and Scroll Down



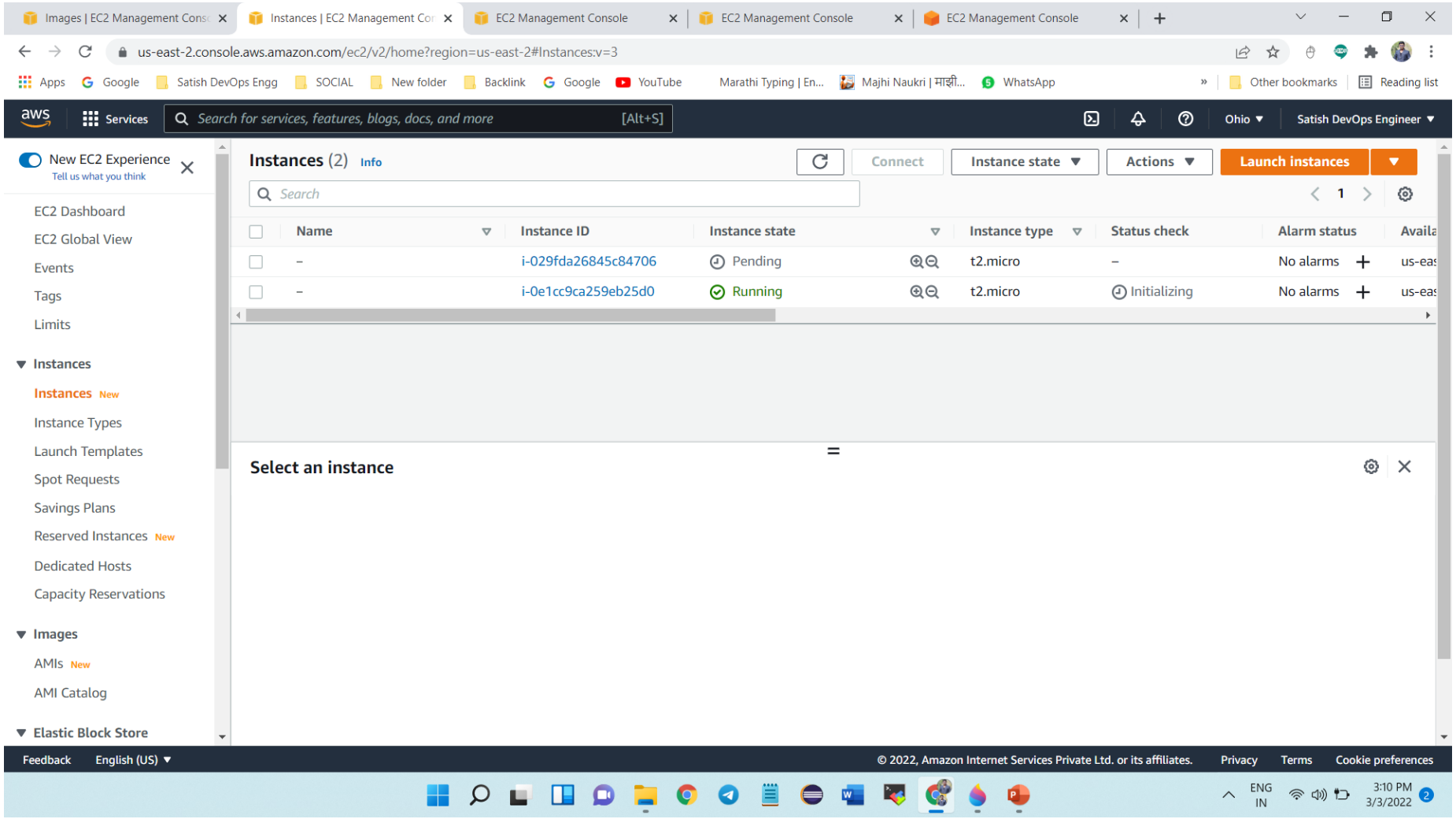
- Click on Create Auto Scaling Group



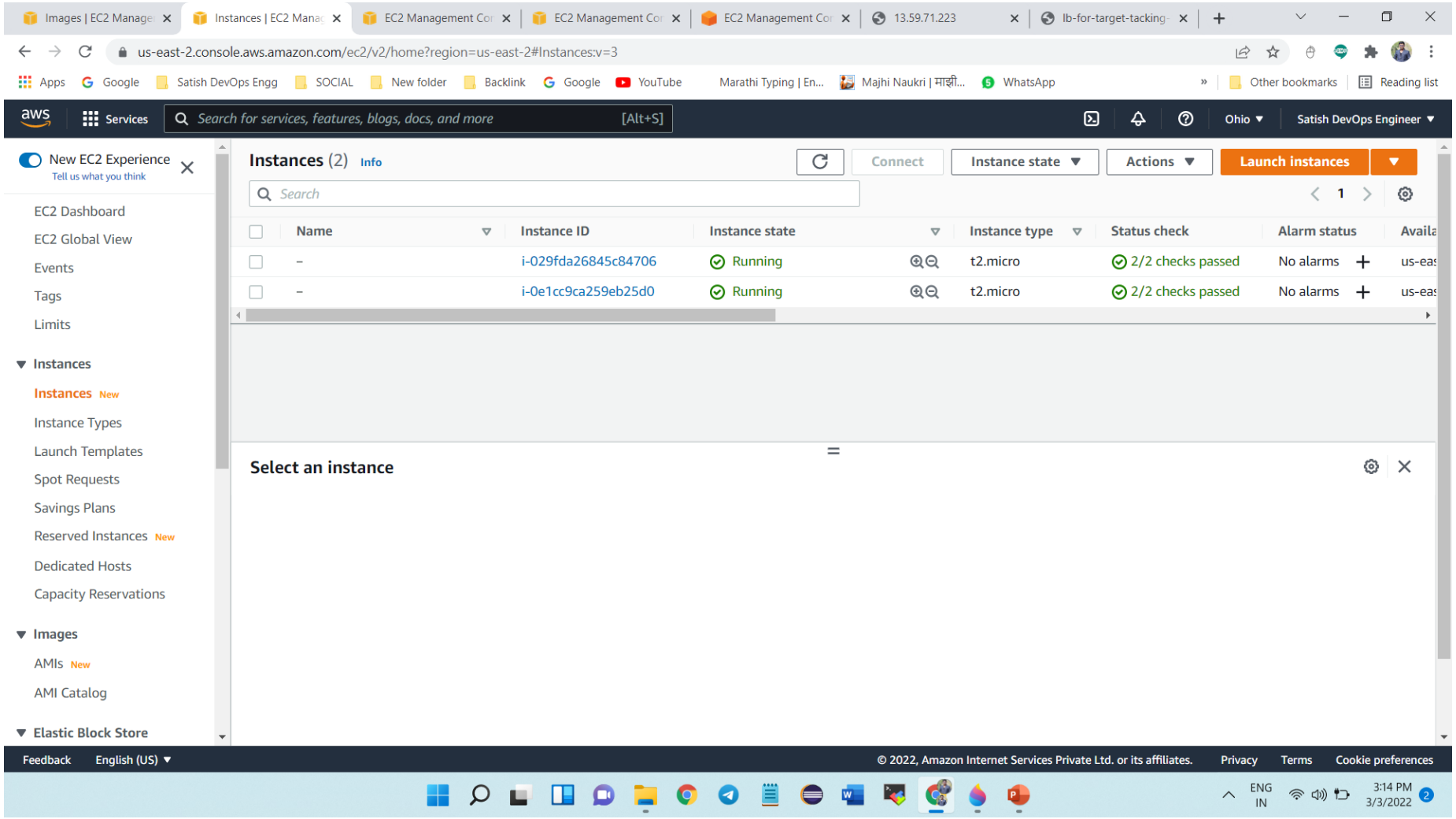
- Here we can see that our ASG is updating Capacity because our desired capacity is 2



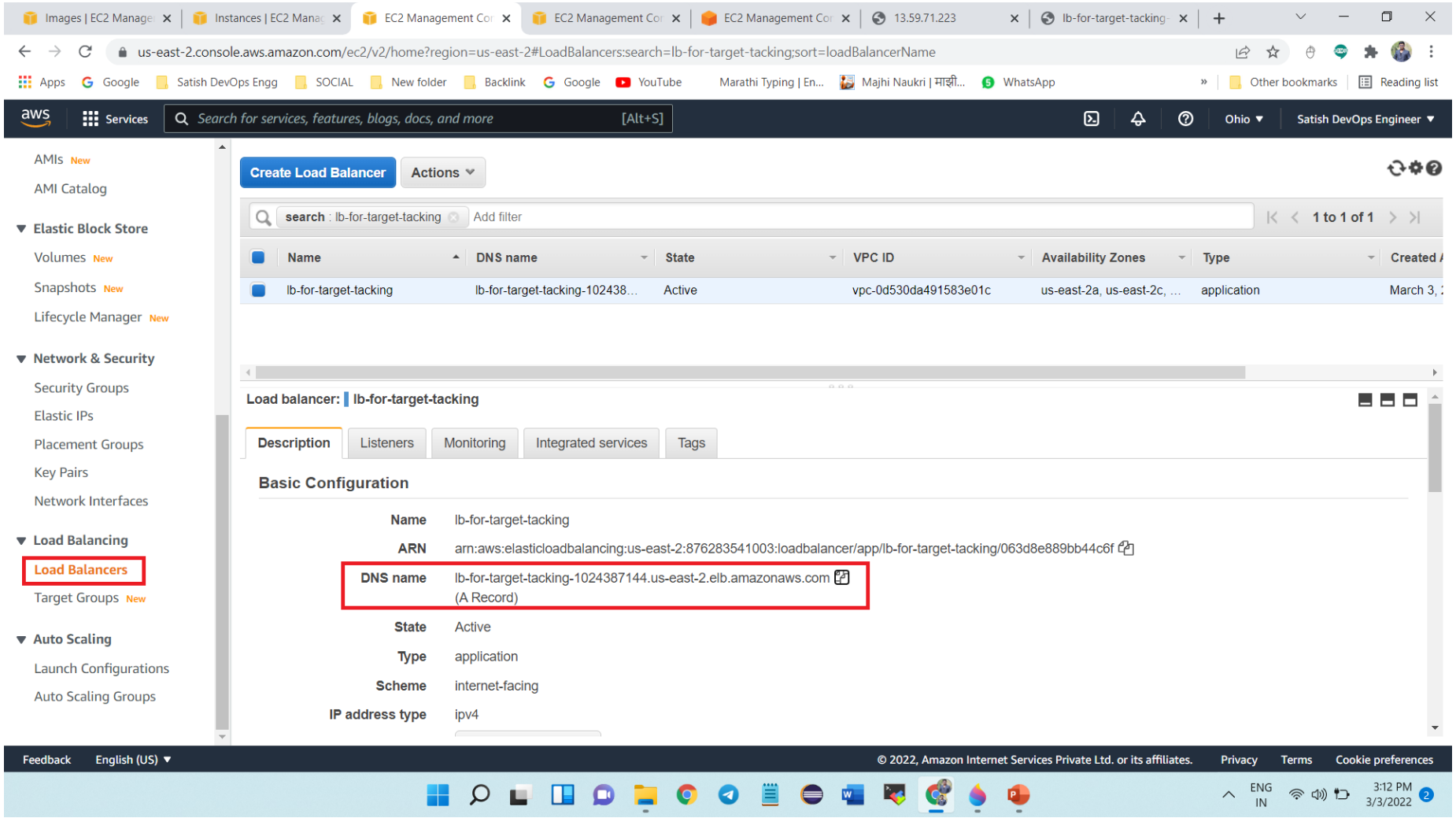
- We can see in Activity also



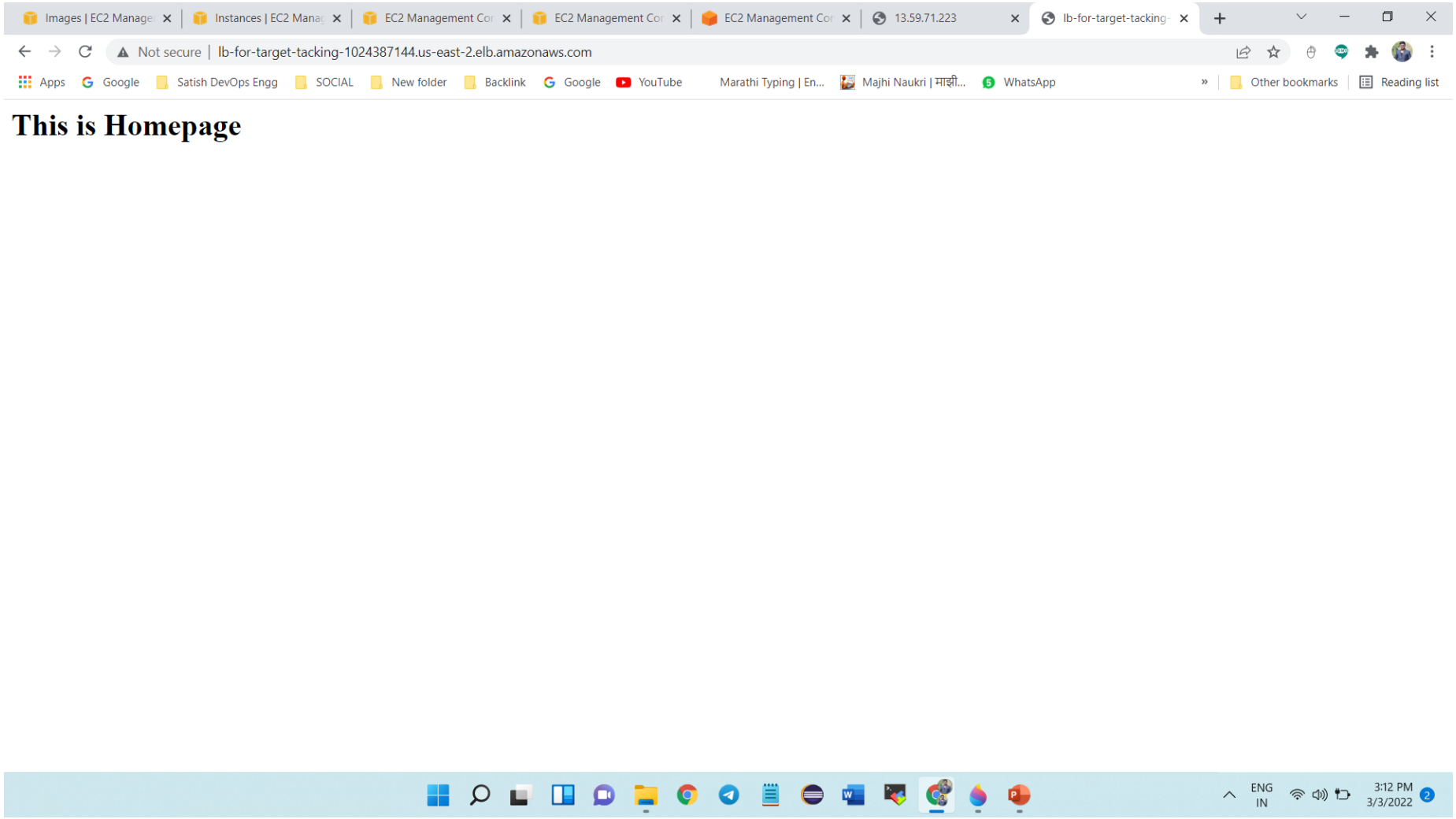
- In instances we can see that 2 instances are launched



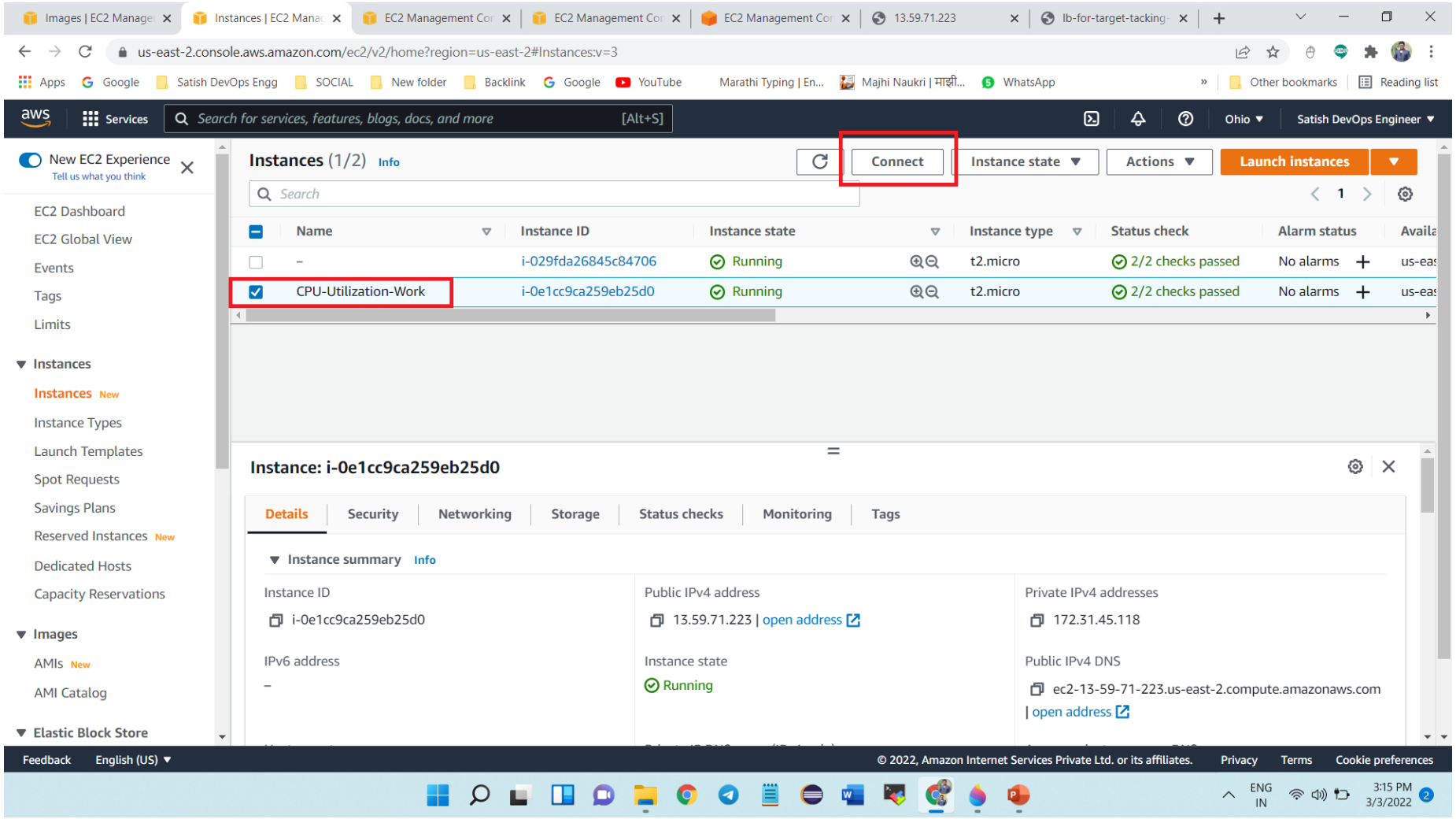
- Also they are running ok



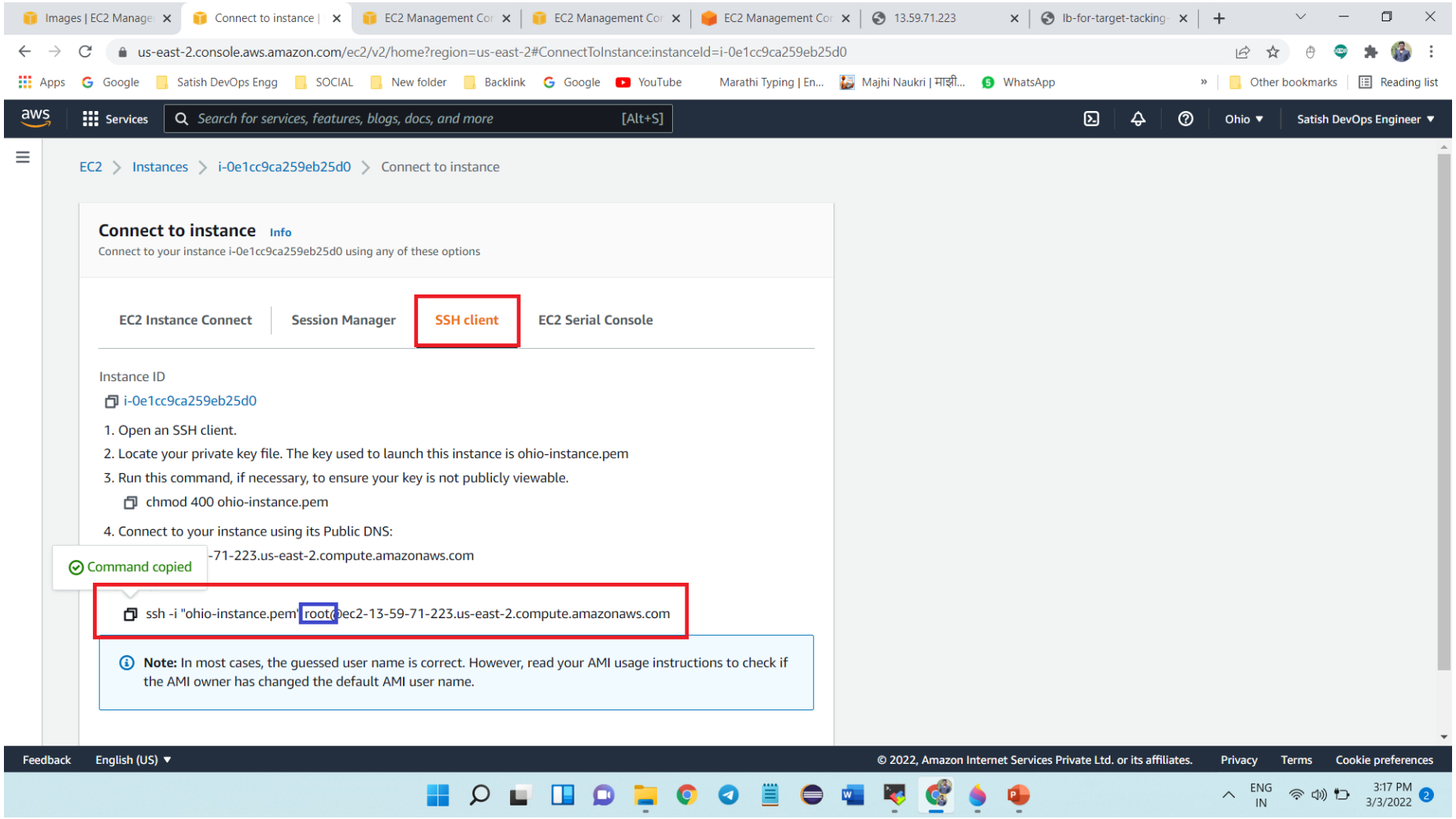
- Now copy DNS of Load Balancer



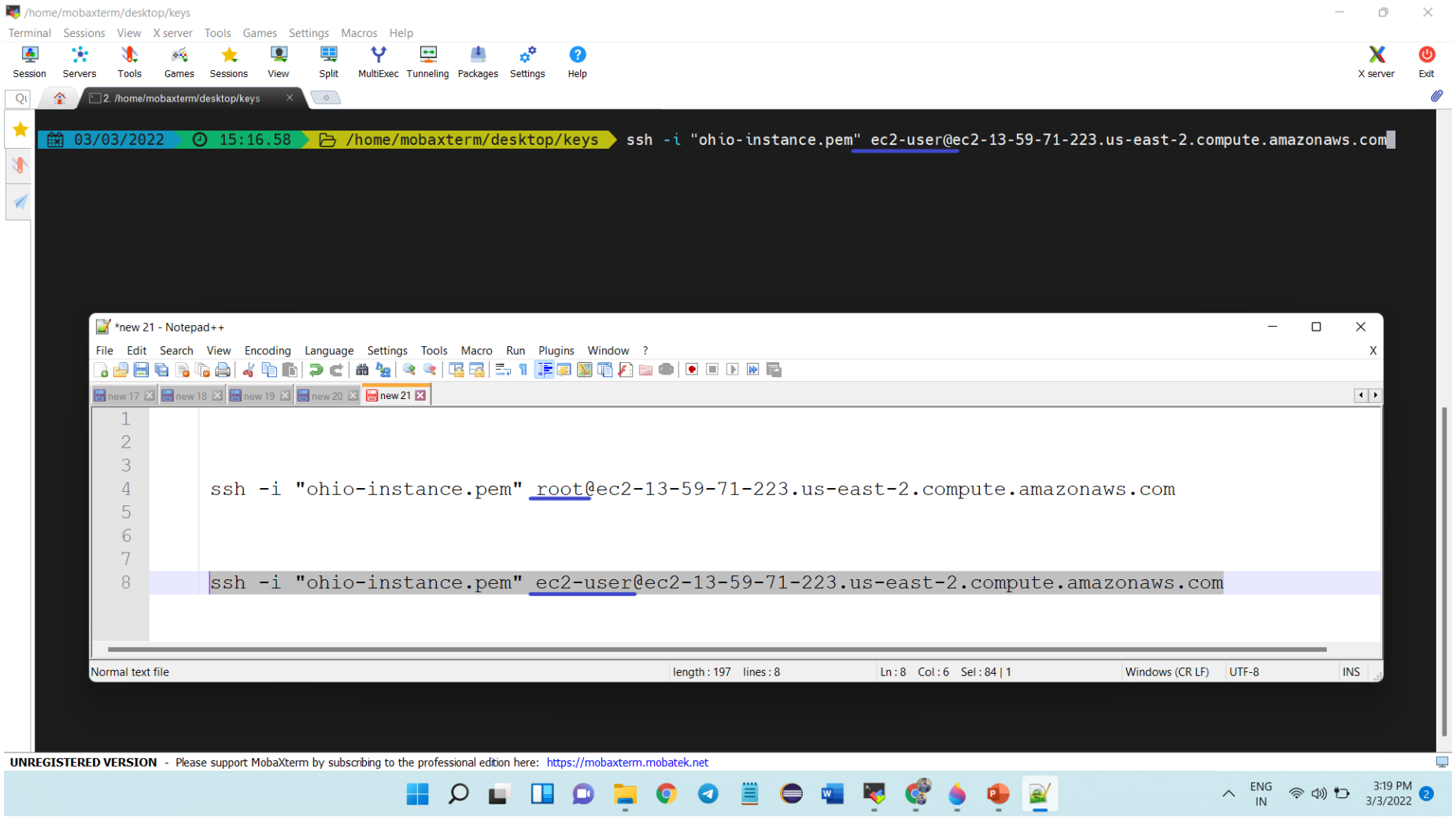
- Check into Browser
- It is working fine



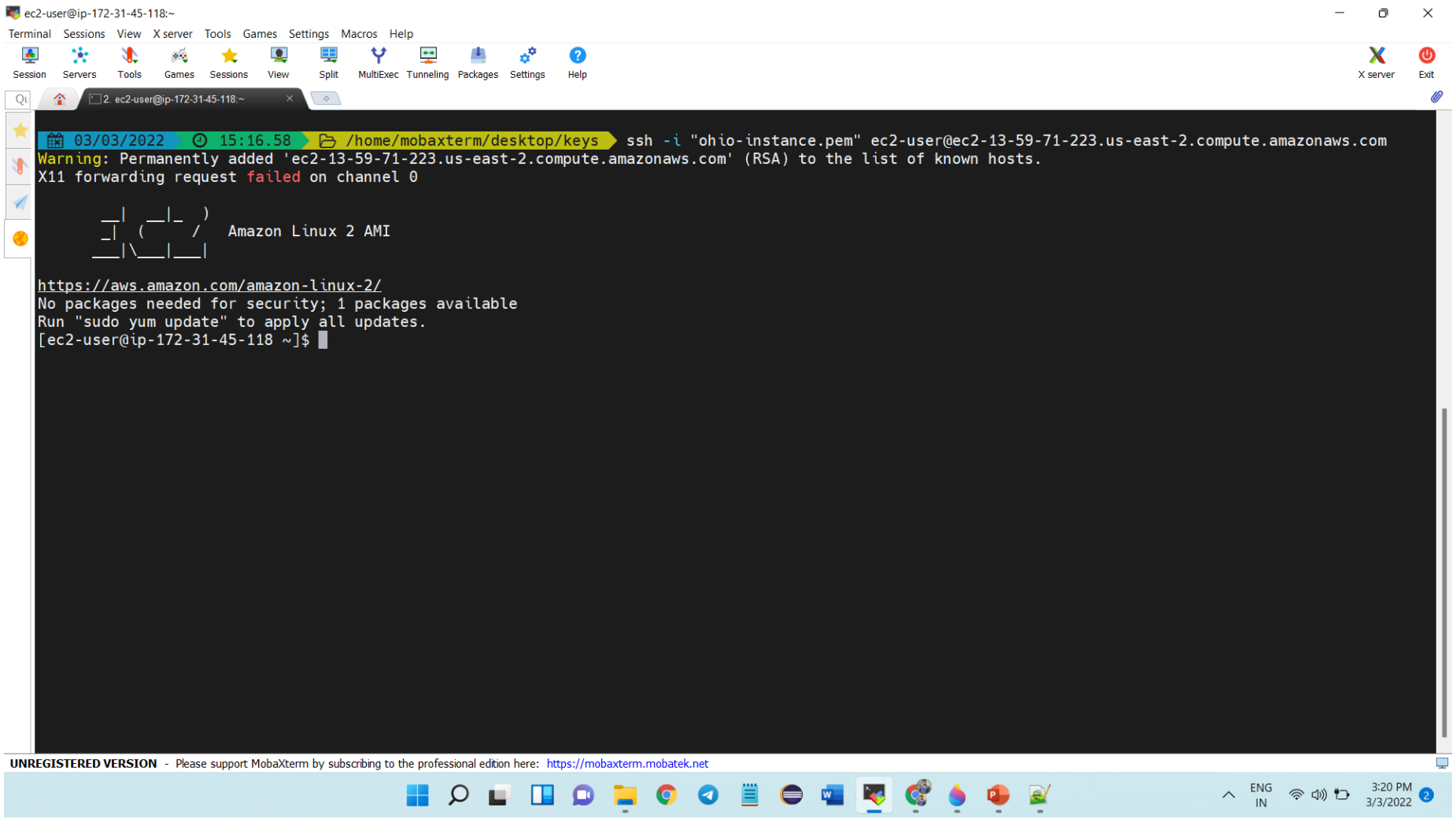
- Now give name to any one instance for recognization
- Then click on connect



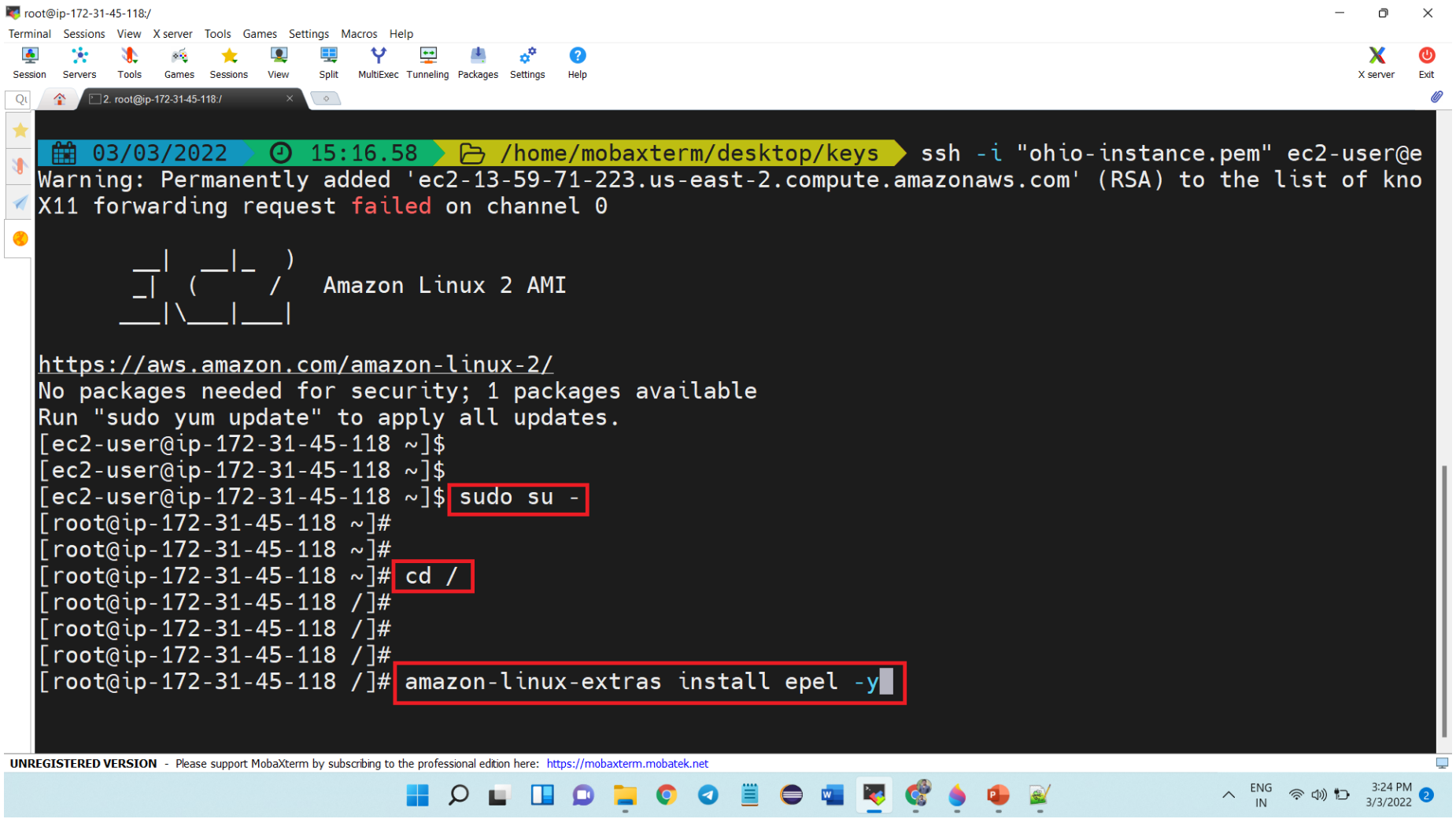
- Click on SSH Client
- Copy command from example
- And at the place of **root** use **ec2-user**


/home/mobaxterm/desktop/keys
Terminal Sessions View X server Tools Games Settings Macros Help
Session Servers Tools Games Sessions View Split MultiExec Tunneling Packages Settings Help
X server Exit
2. /home/mobaxterm/desktop/keys
03/03/2022 15:16.58 /home/mobaxterm/desktop/keys ssh -i "ohio-instance.pem" ec2-user@ec2-13-59-71-223.us-east-2.compute.amazonaws.com
*new 21 - Notepad++
File Edit Search View Encoding Language Settings Tools Macro Run Plugins Window ?
new 17 new 18 new 19 new 20 new 21
1
2
3
4 ssh -i "ohio-instance.pem" root@ec2-13-59-71-223.us-east-2.compute.amazonaws.com
5
6
7
8 ssh -i "ohio-instance.pem" ec2-user@ec2-13-59-71-223.us-east-2.compute.amazonaws.com
Normal text file length : 197 lines : 8 Ln : 8 Col : 6 Sel : 84 | 1 Windows (CR LF) UTF-8 INS
UNREGISTERED VERSION - Please support MobaXterm by subscribing to the professional edition here: https://mobaxterm.mobatek.net
ENG IN
3:19 PM 3/3/2022

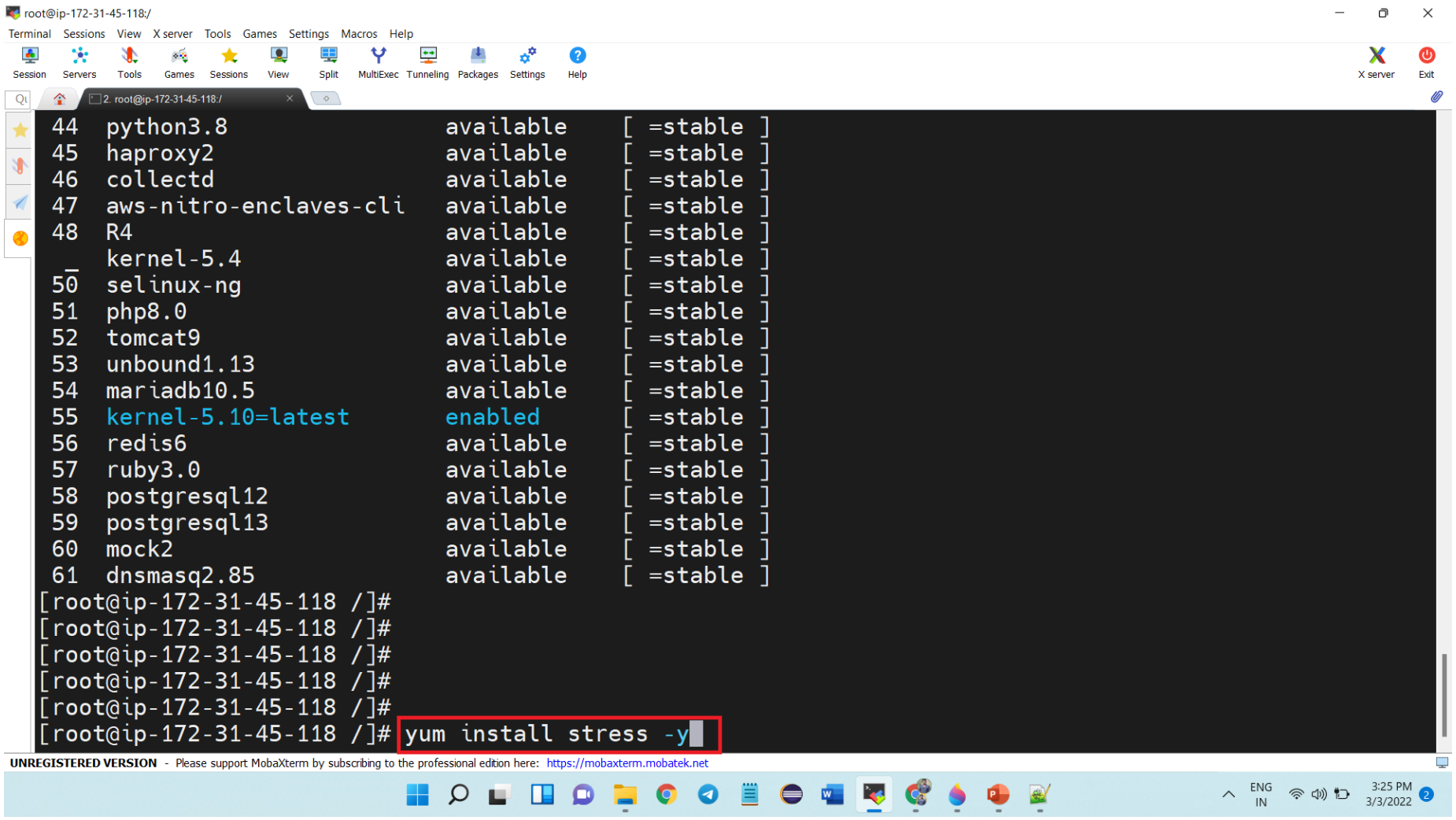
- Like this



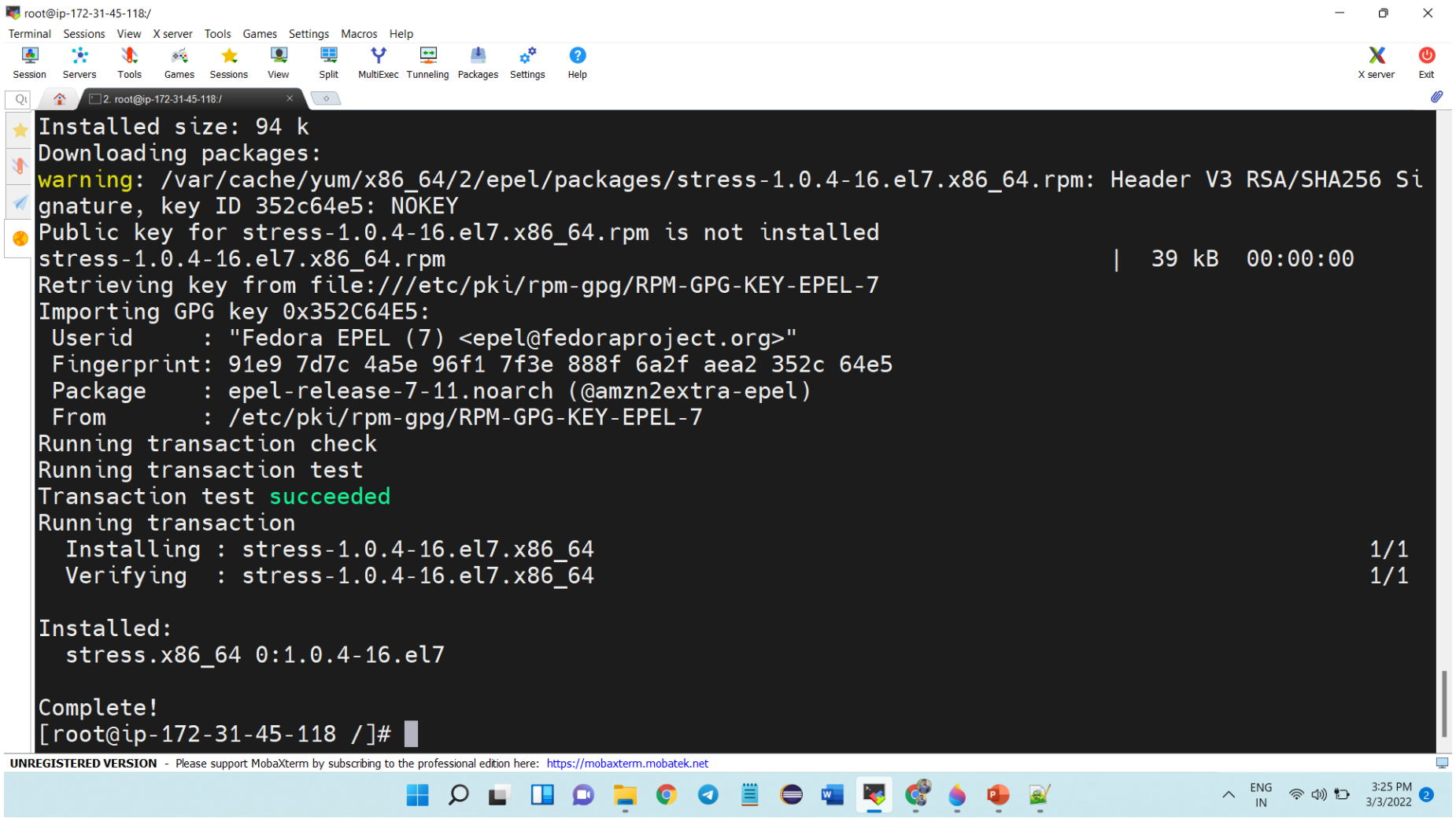
- Here we can see that we are logged into our instance



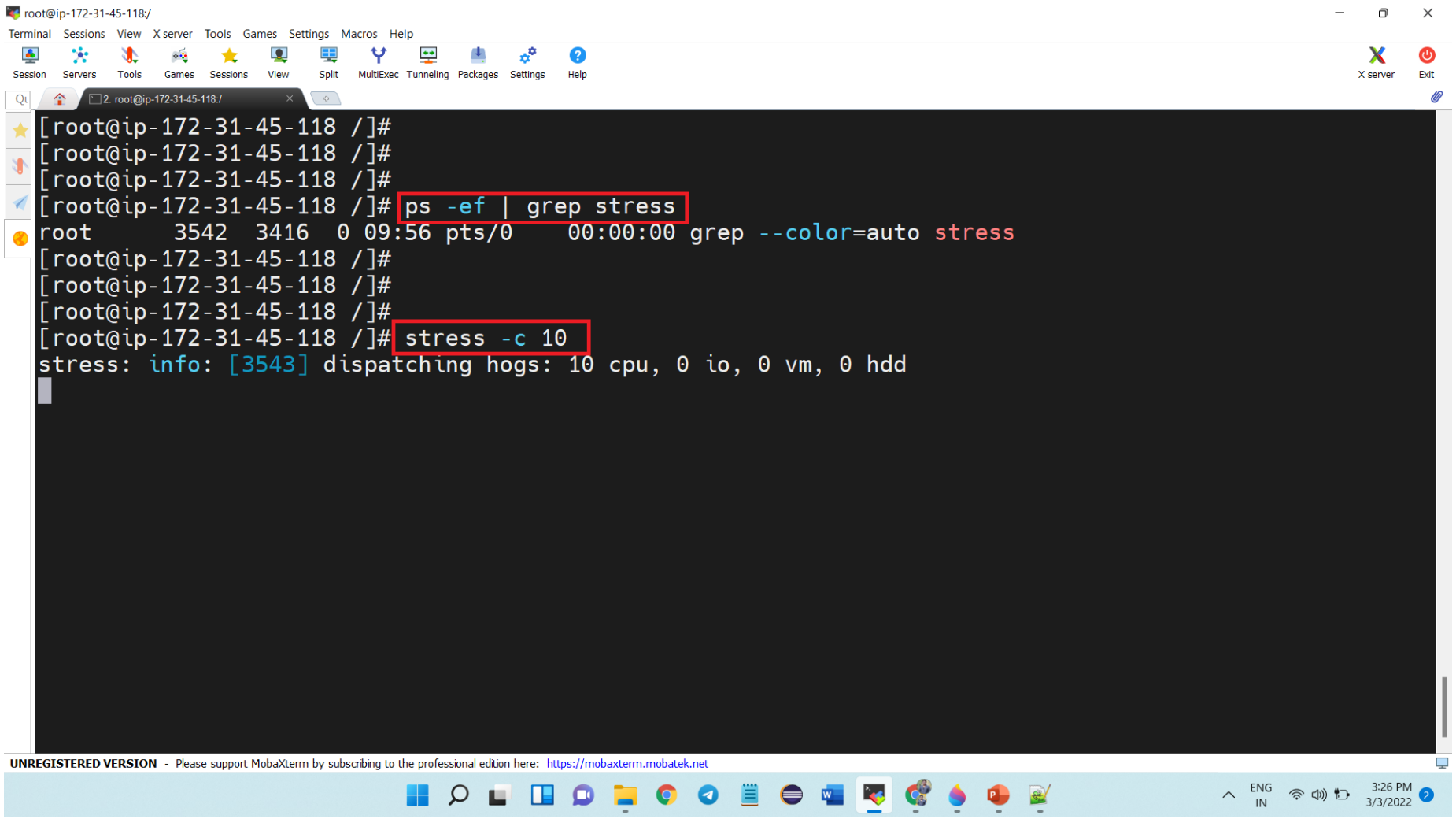
- Login as Root
- Now use **amazon-linux-extras install epel -y**
- To install extra packages into machine



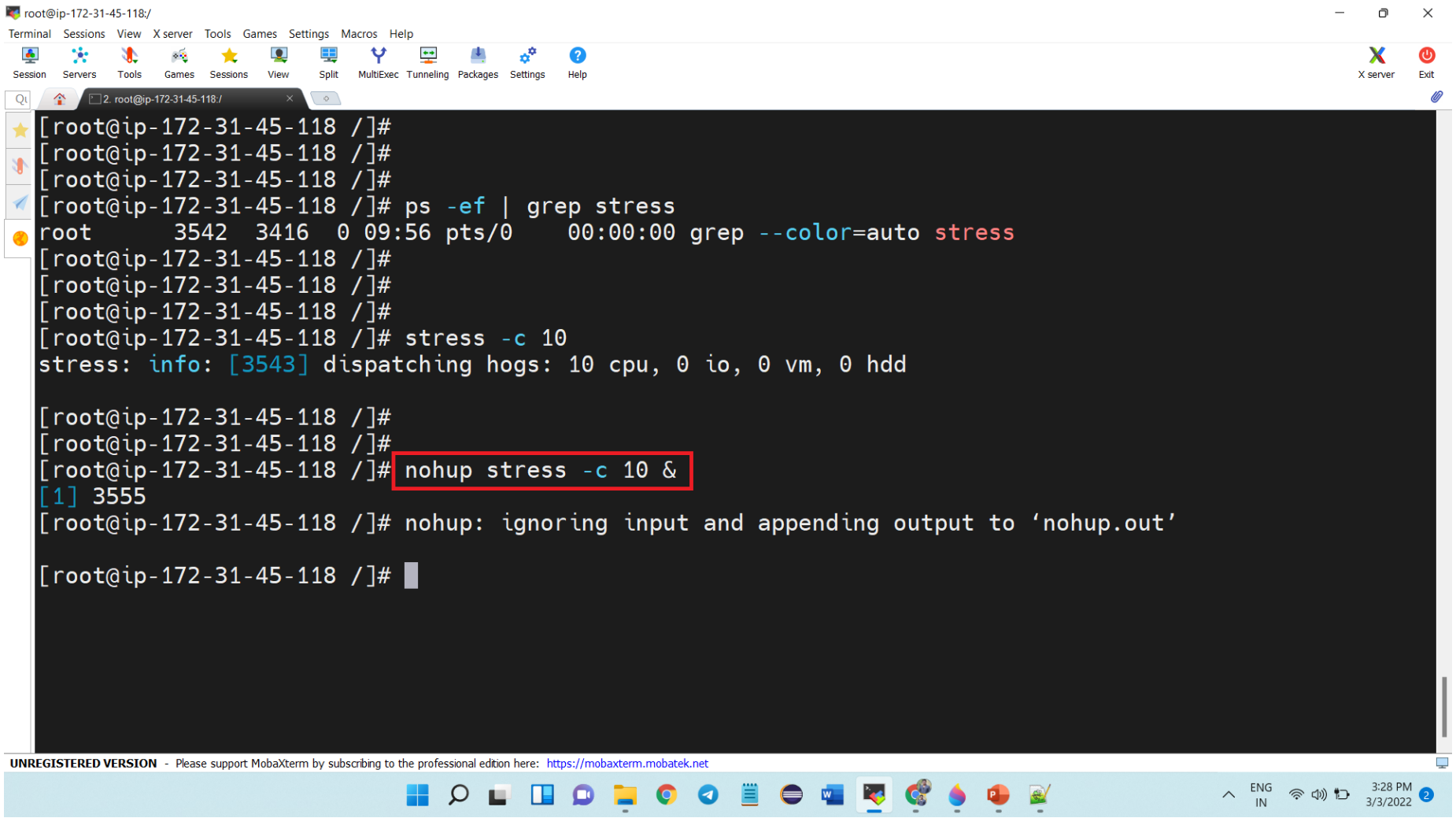
- Use **yum install stress -y**



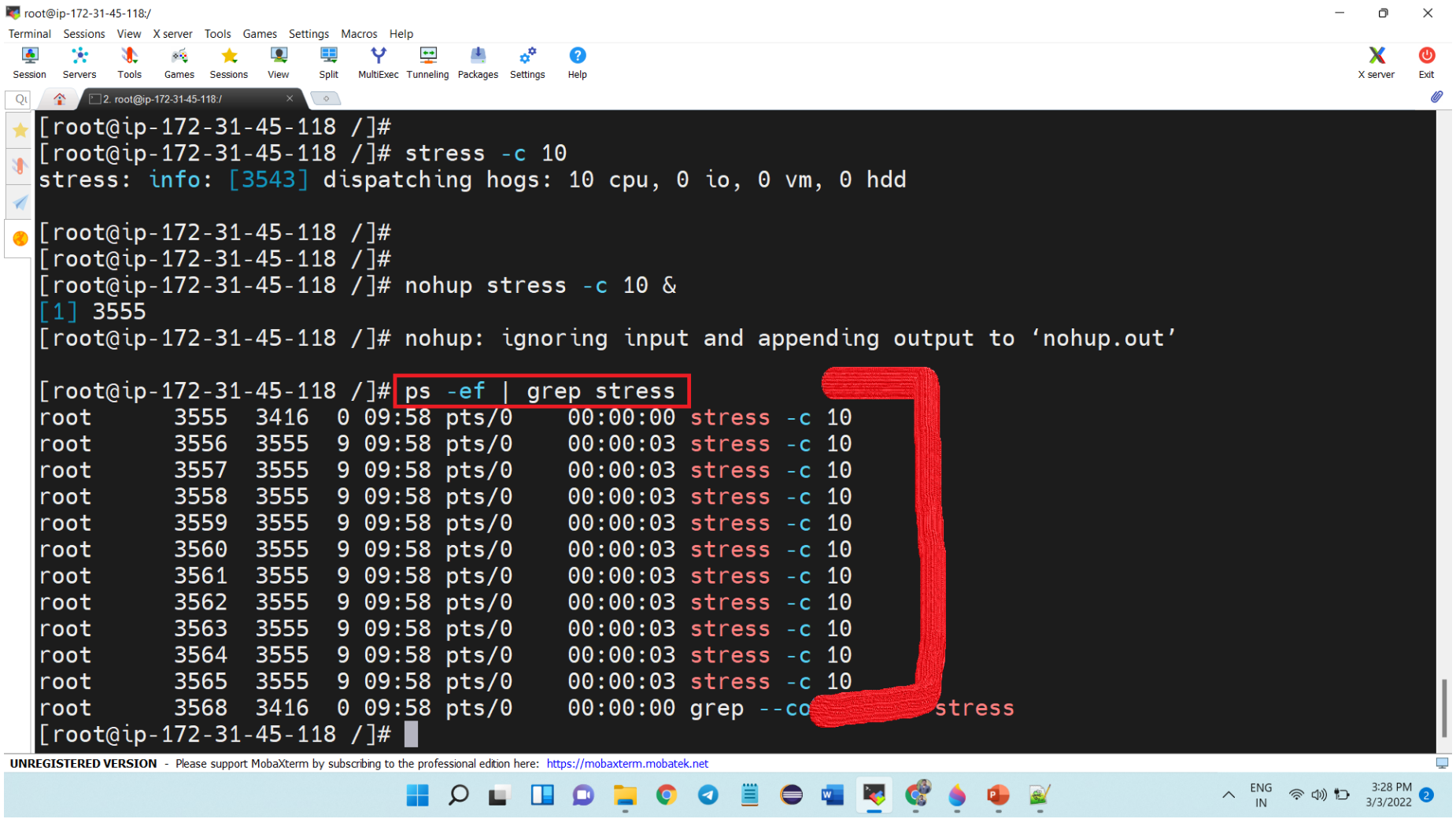
- It is installed successfully



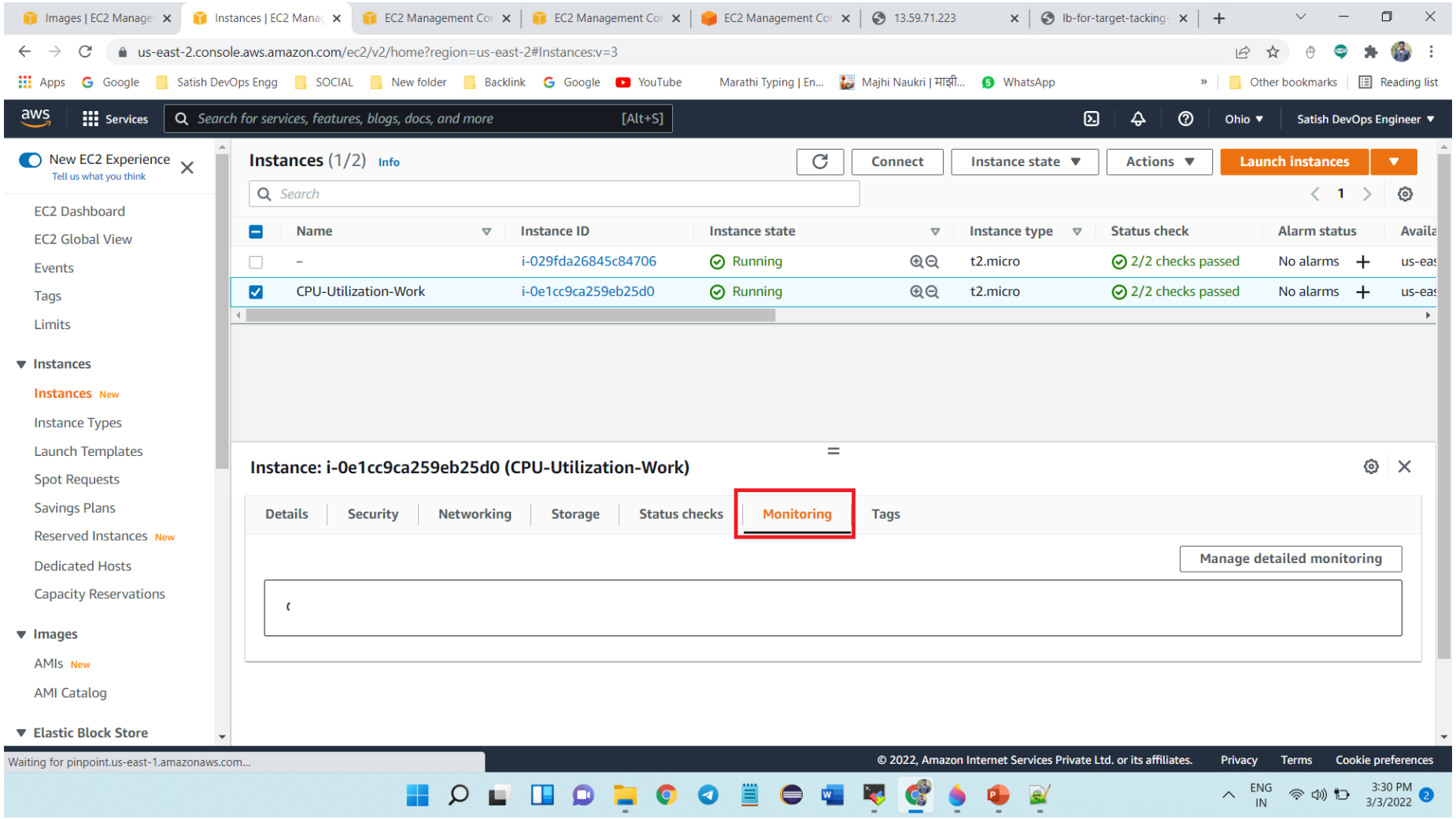
- Use **ps -ef | grep stress** to check ongoing processes of stress
- Now use **stress -c 10**
- But if you pressed **ctrl+c** it will exit from stress



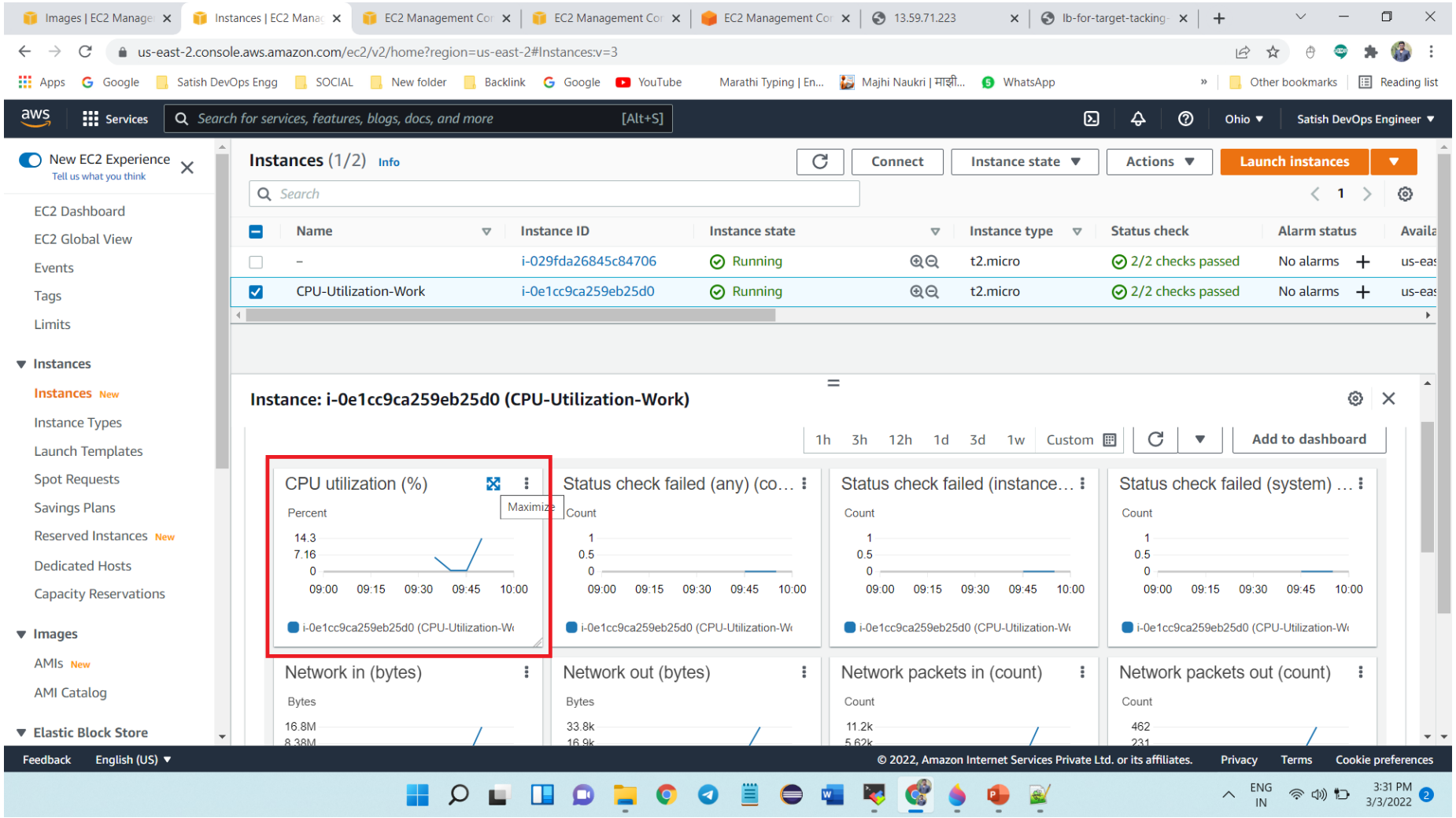
- So we will use stress in backend
- Now use **nohup stress -c 10 &**
- Press **ctrl+c**



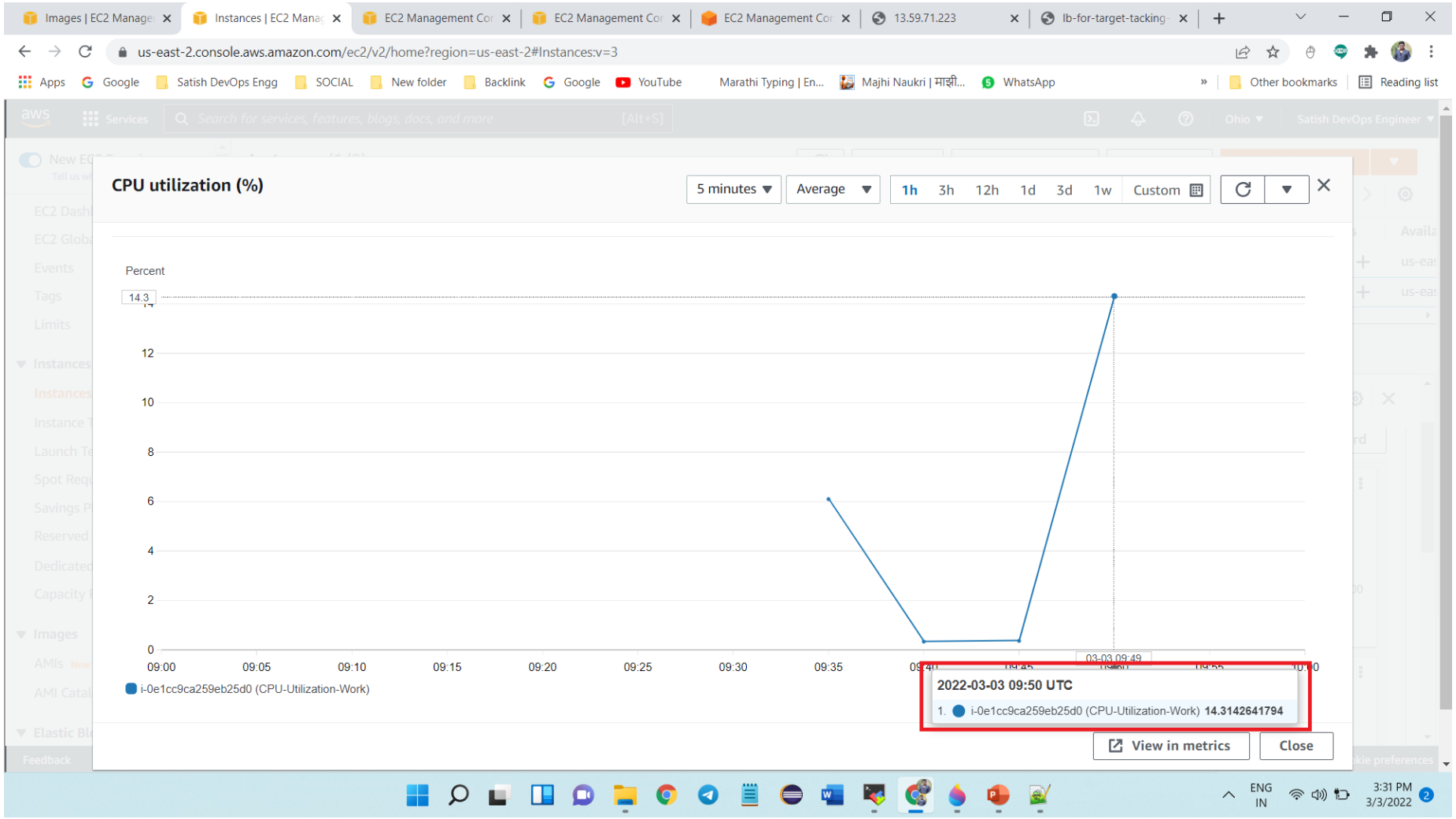
- Now use **ps -ef | grep stress**
- Here 10 processes are ongoing for stress



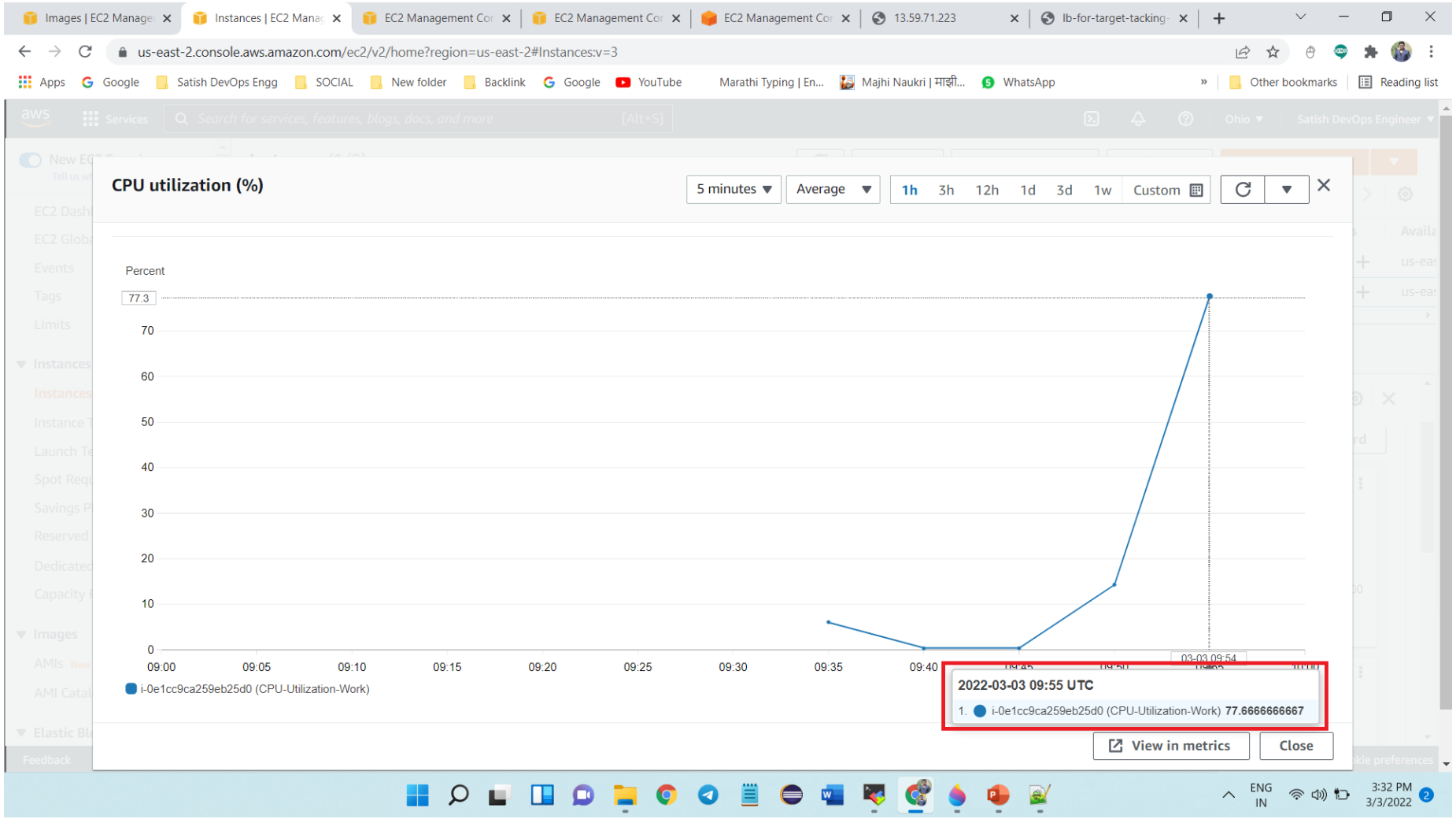
- Now go to instances
- Select the instance which you have connected then click on monitoring



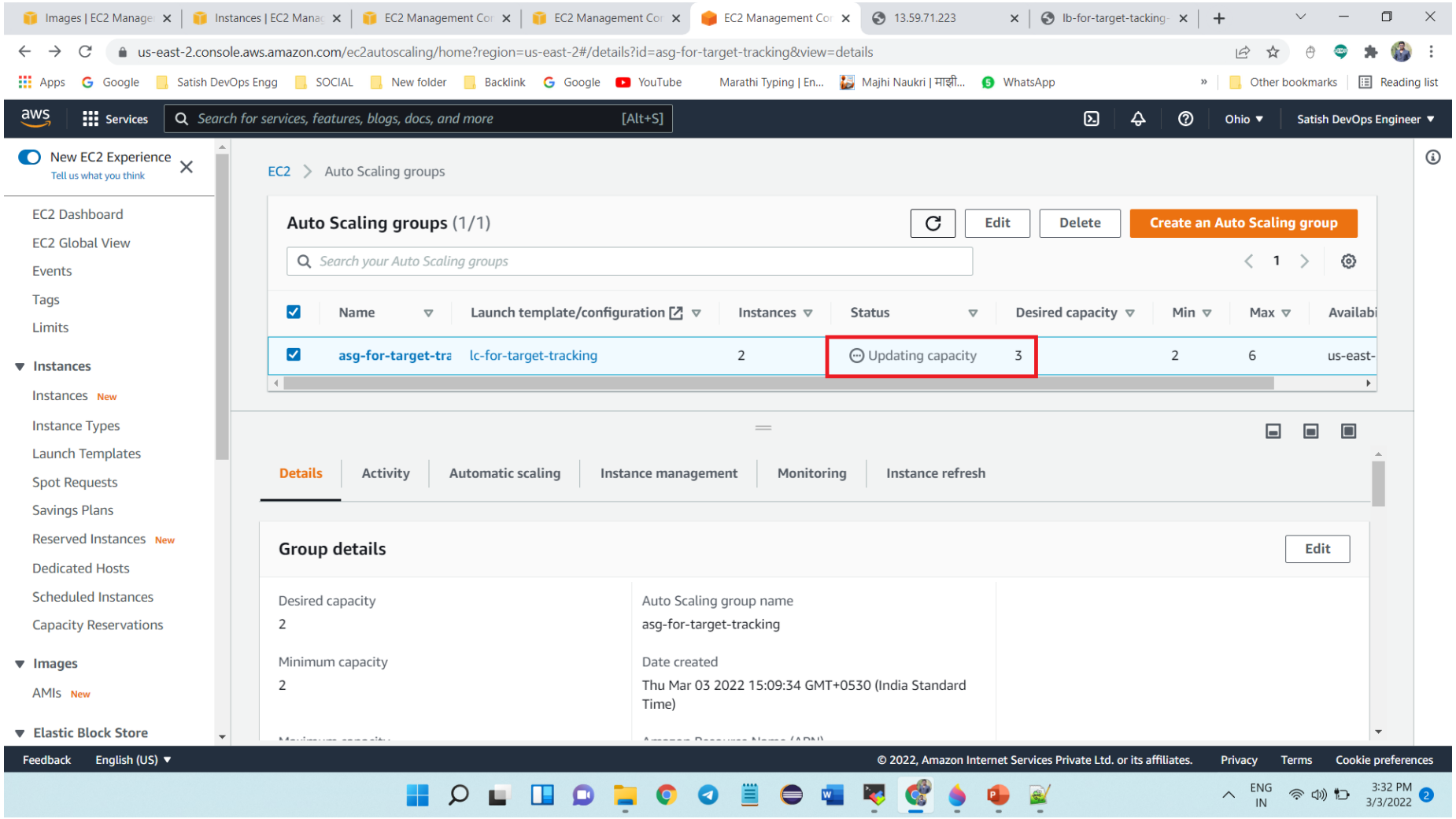
- Maximize the **CPU Utilization**



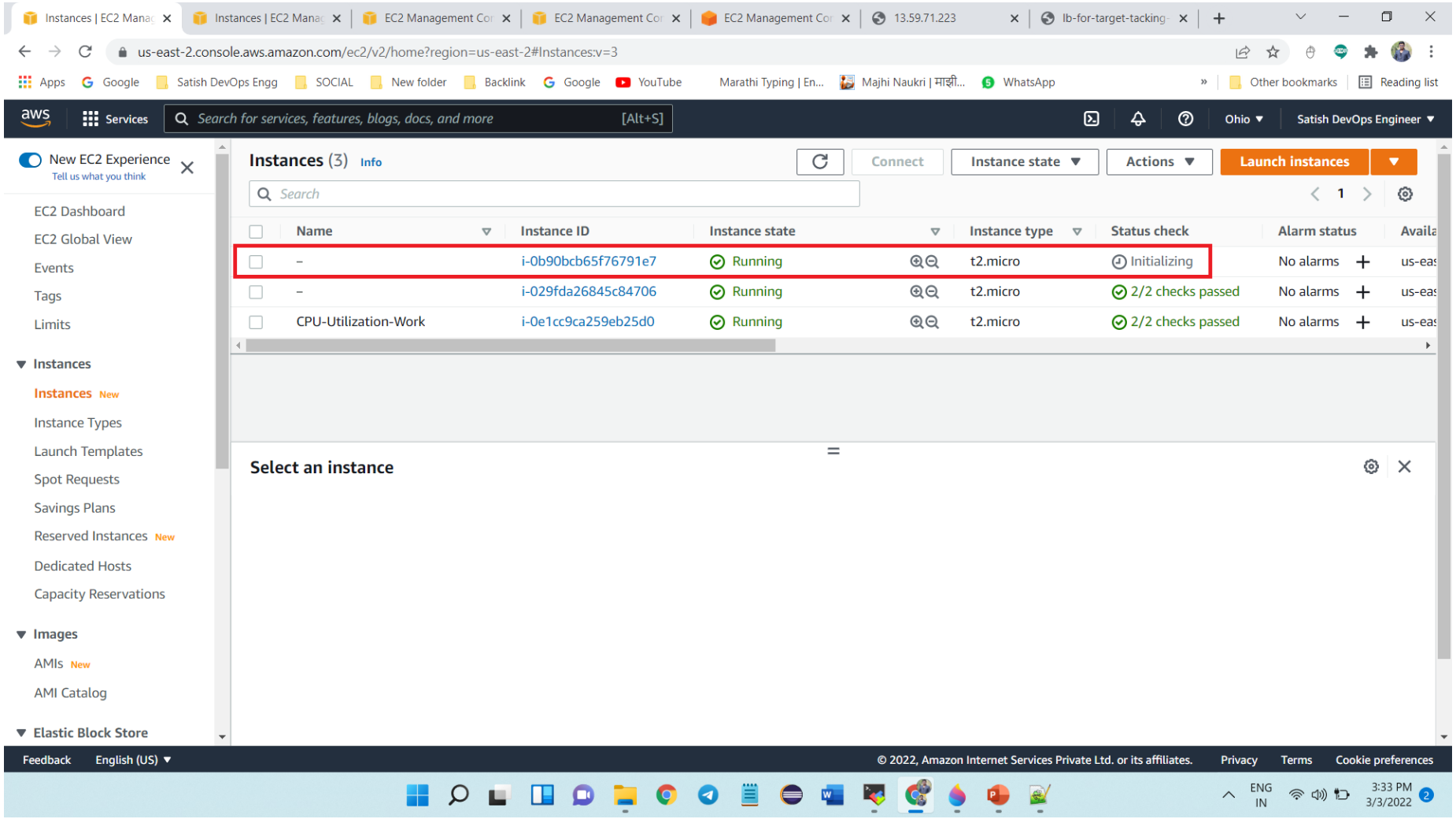
- Here we can see that our cpu utilization is increasing



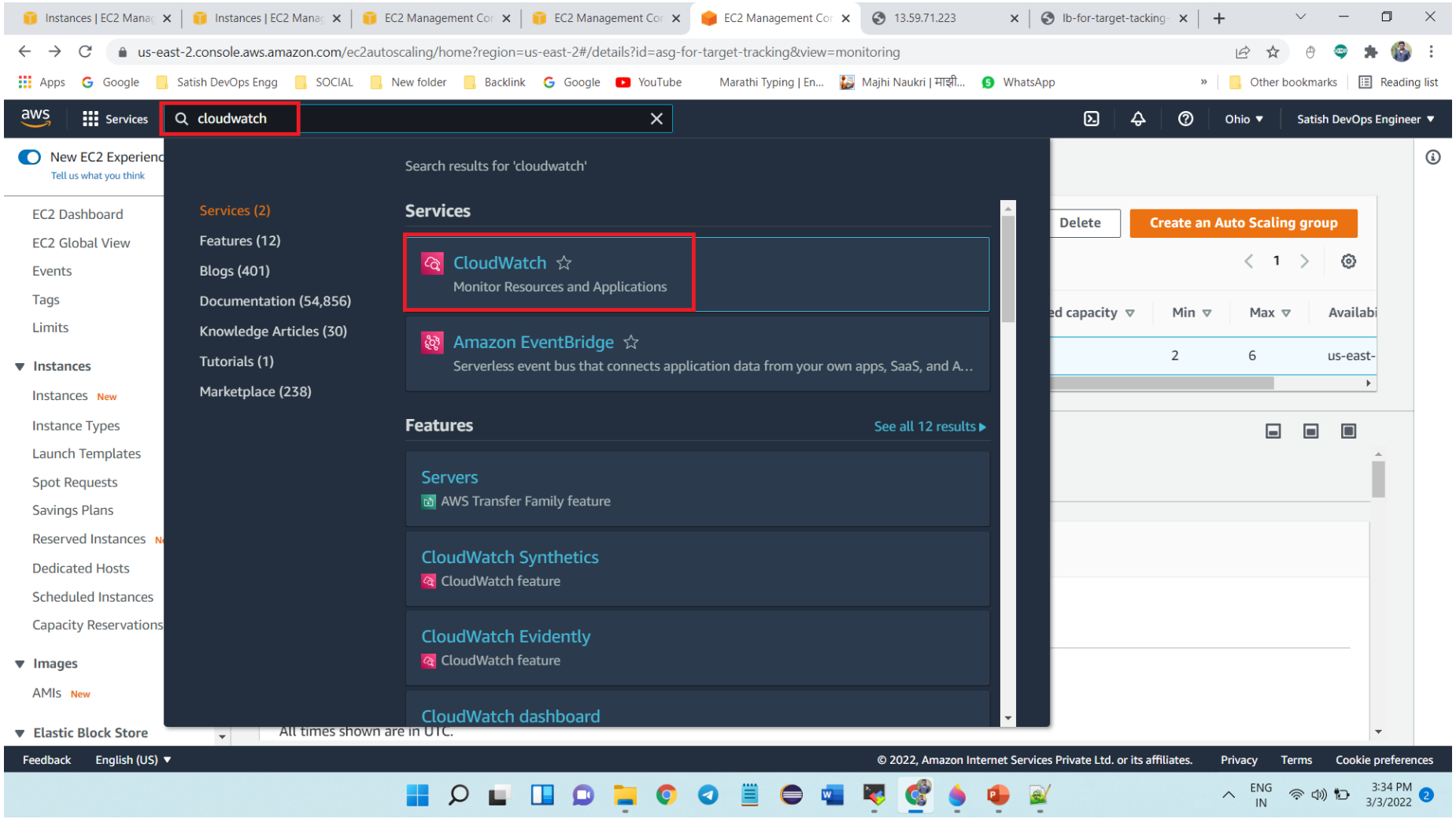
- It is now above 60%



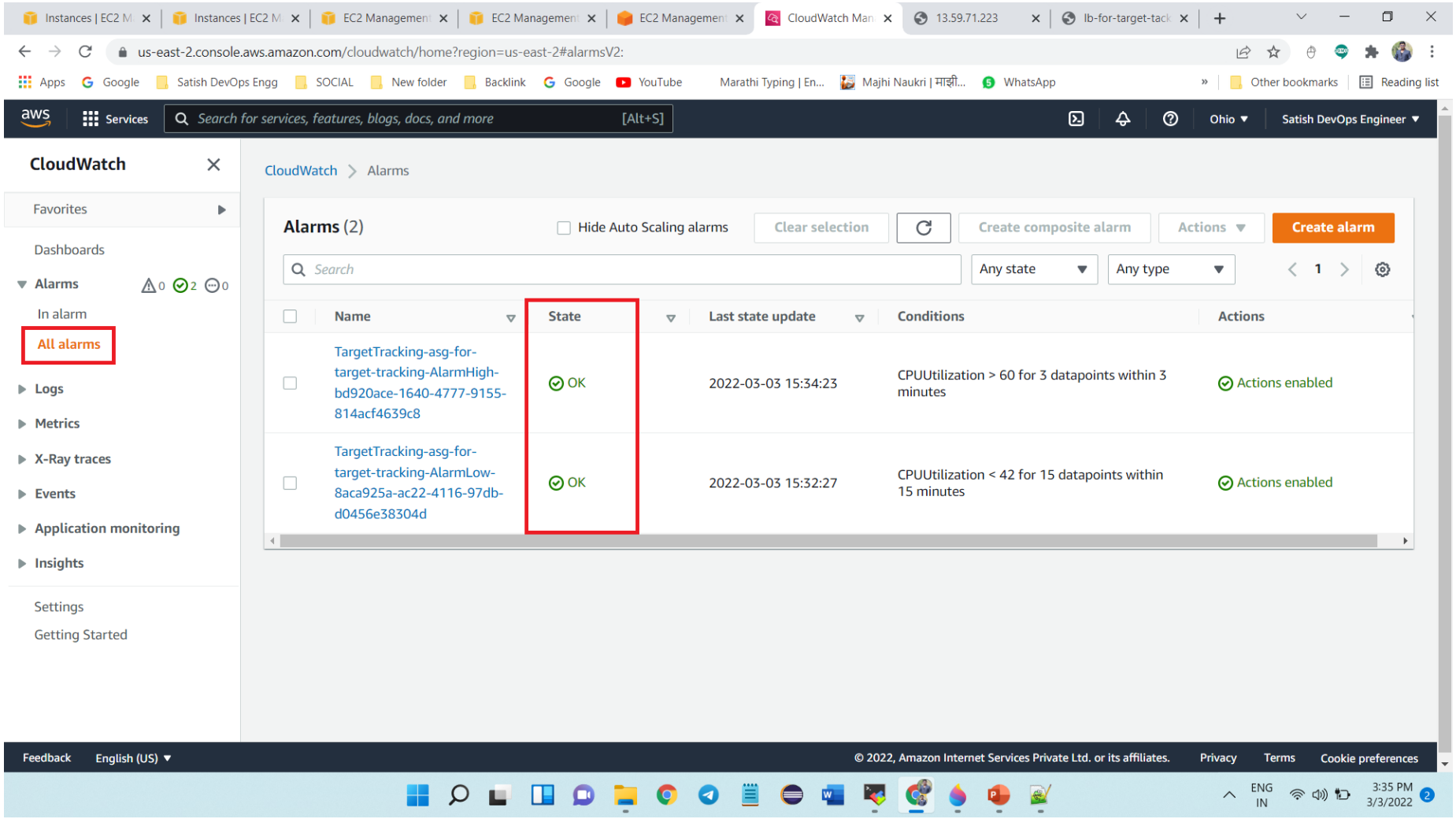
- Here we can see that as the load increases it will scale out the instances



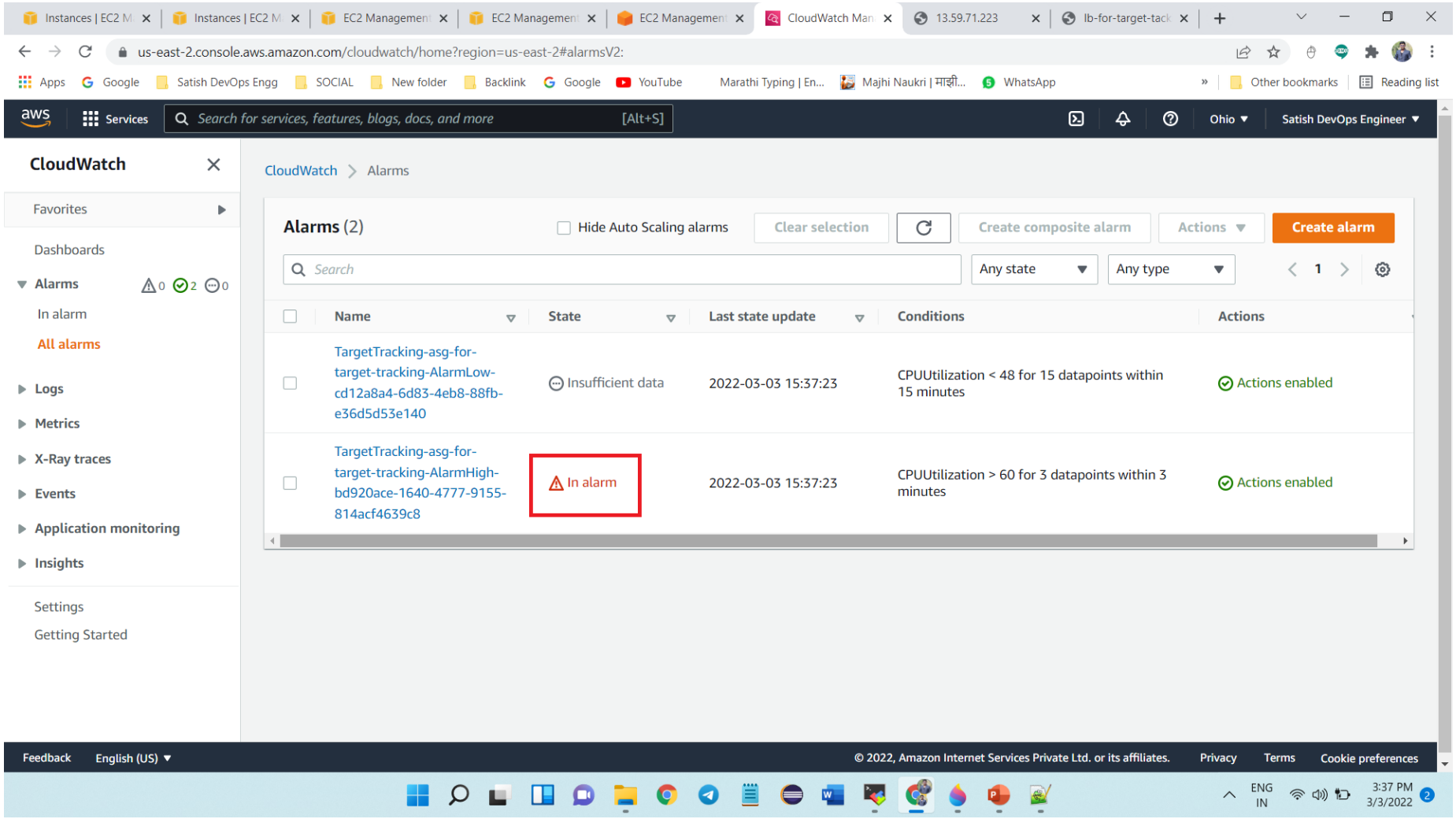
- One new instance is launched



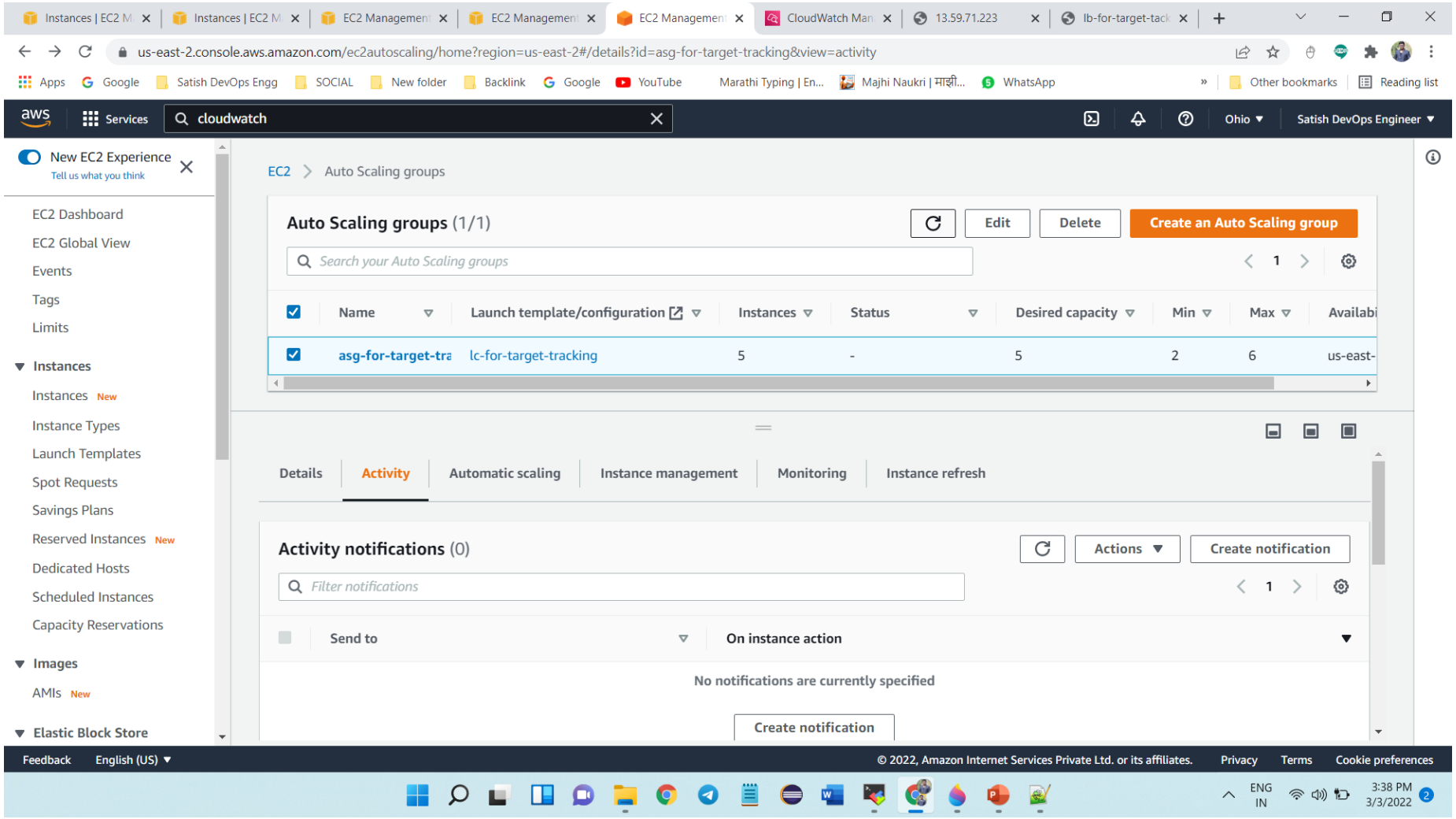
- Now in main search
- Search for CloudWatch
- Click on CloudWatch



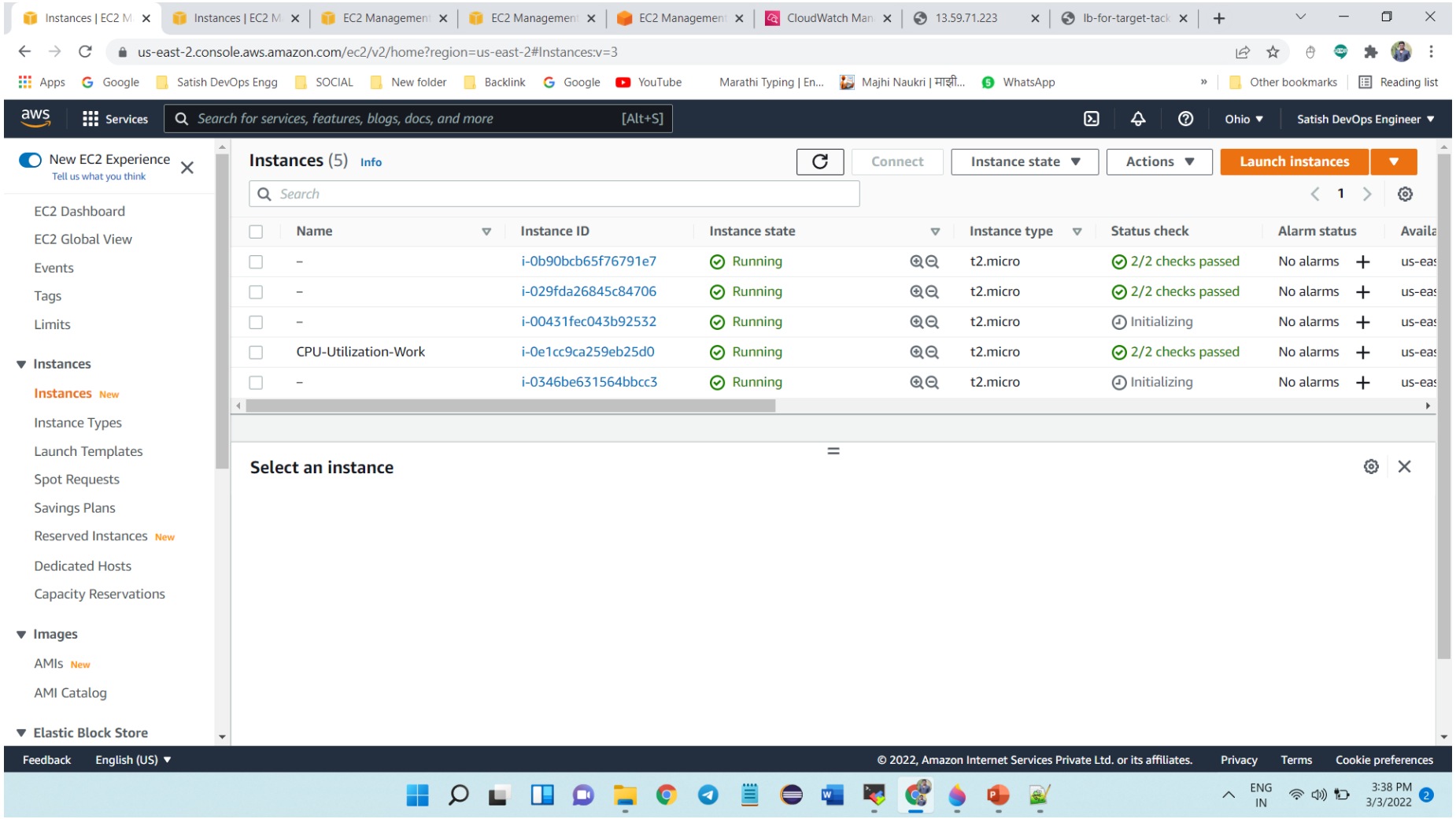
- Here we can see that our alarm state is OK



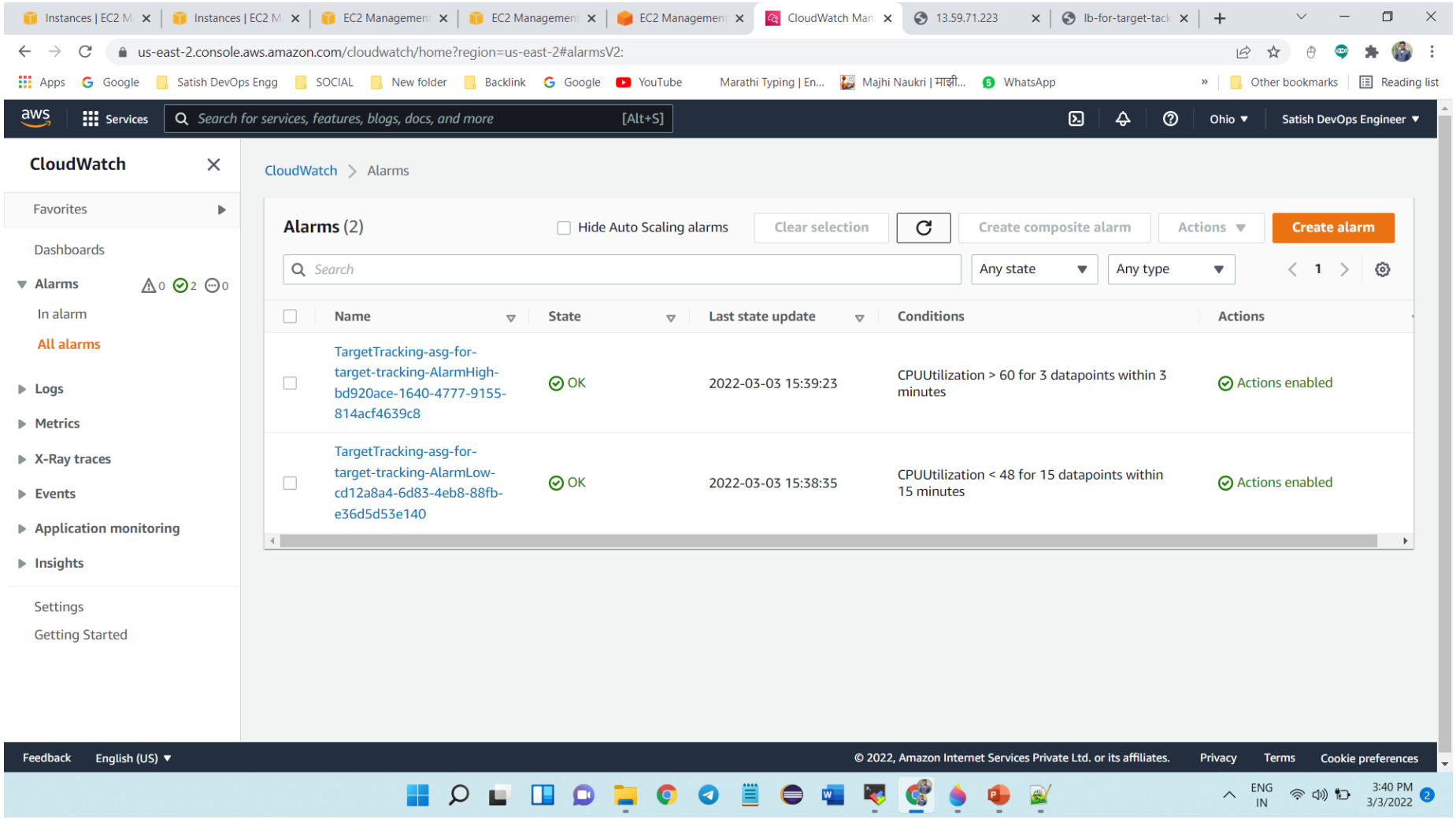
- Now it is changed to In Alarm state



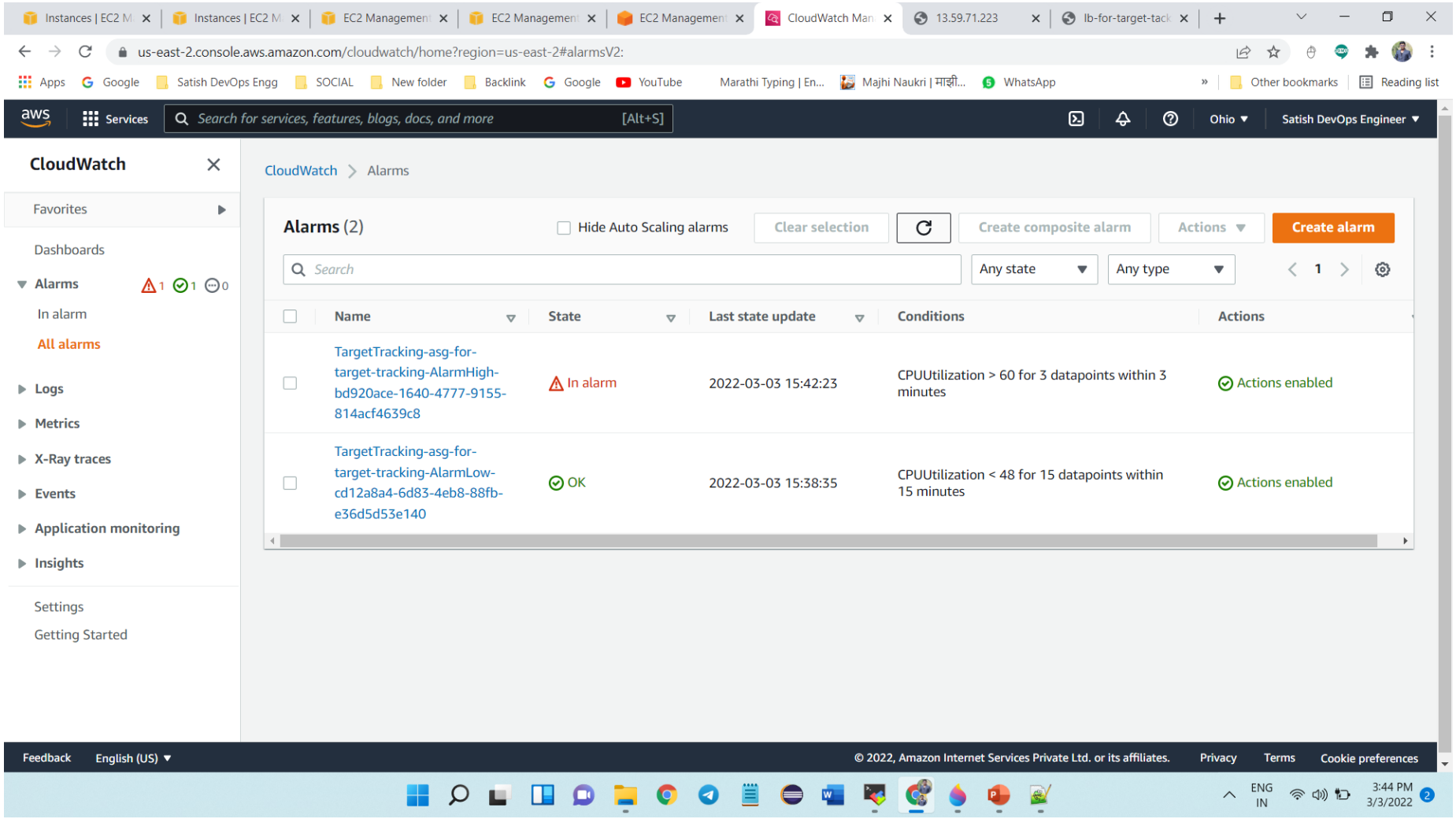
- It has again the desired capacity from 3 to 5



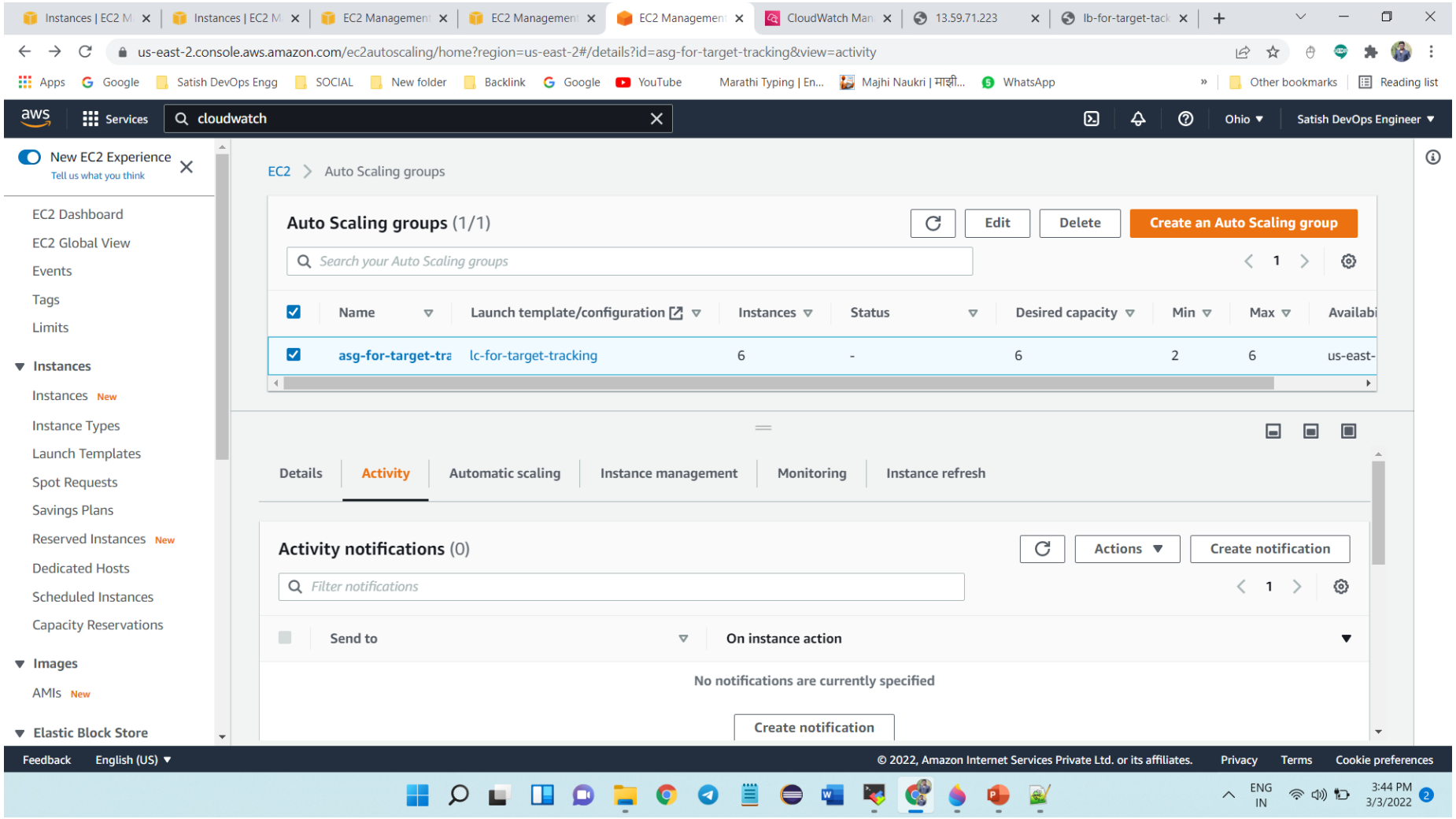
- Again 2 New instances are launched



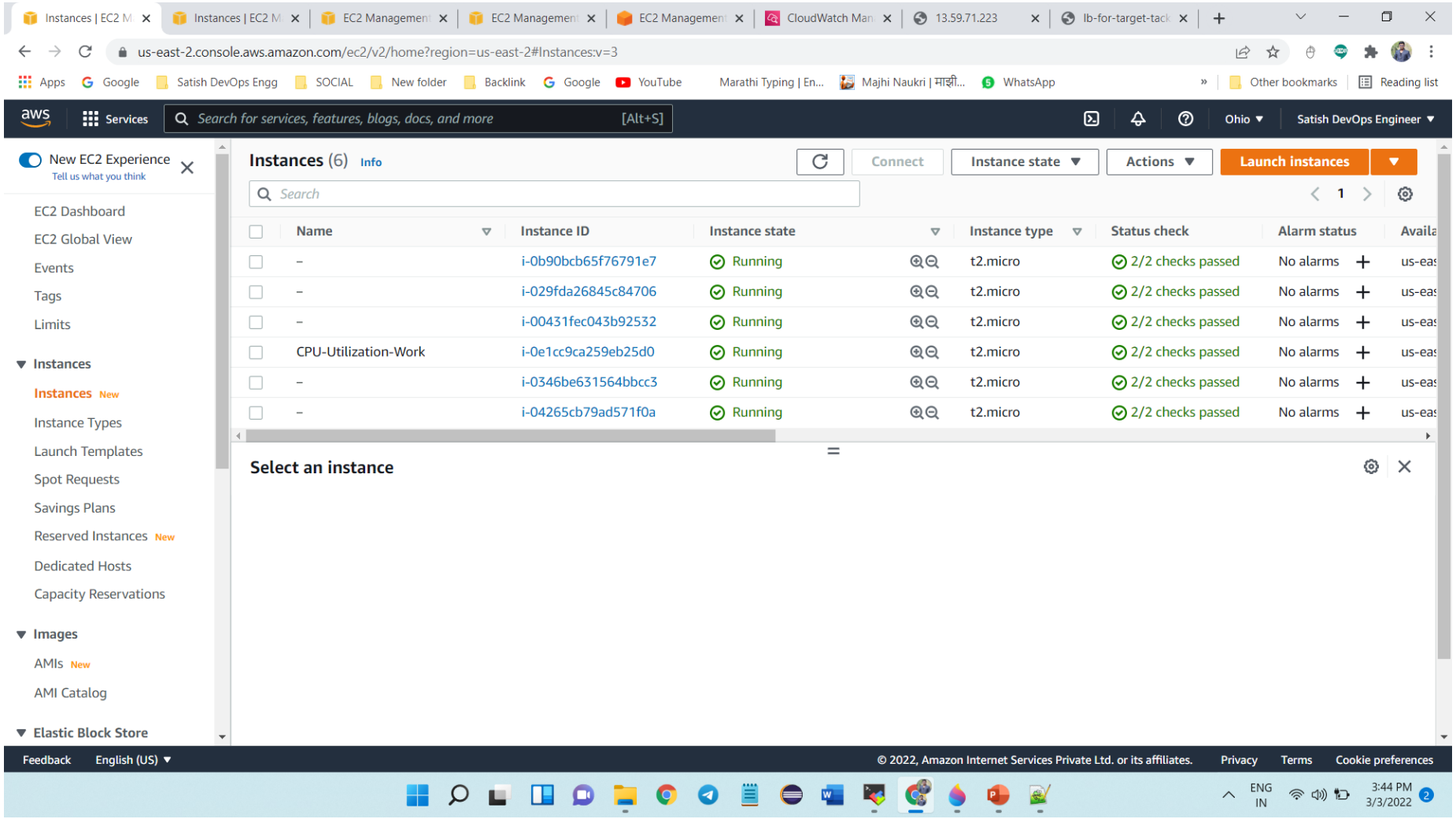
- Here we can see that our alarm state is OK



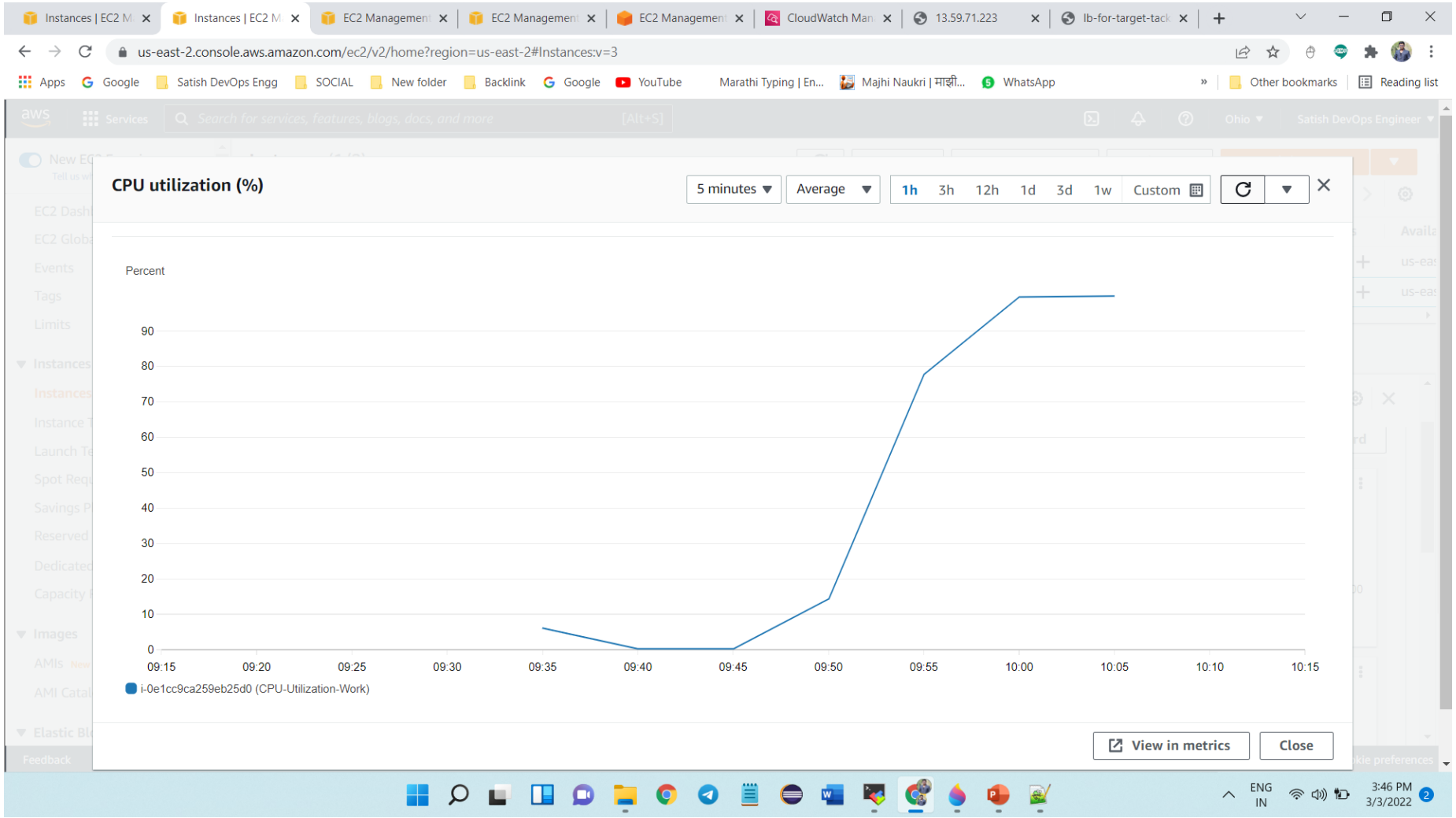
- Now it is changed to In Alarm state

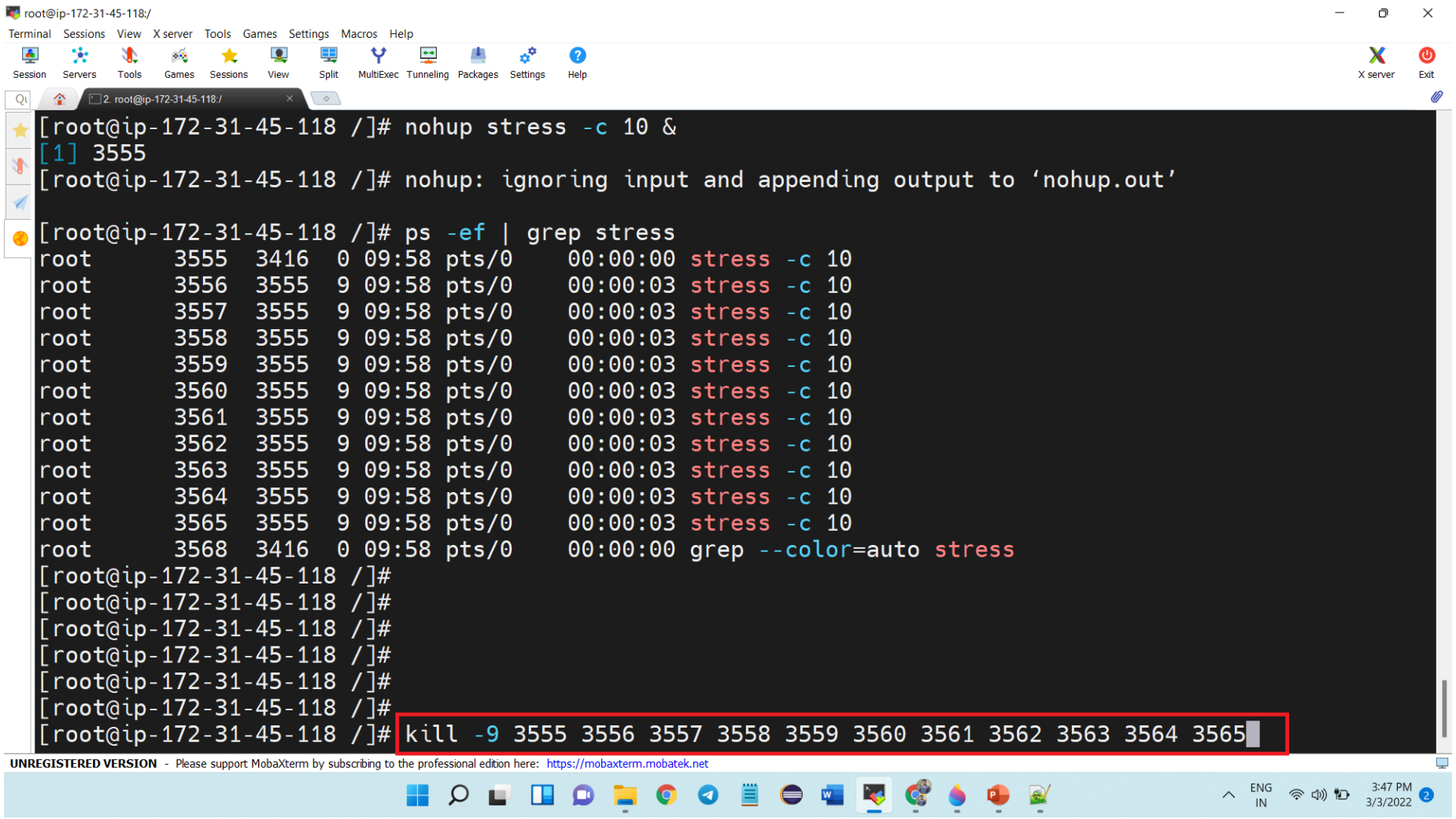


- It has again the desired capacity from 5 to 6

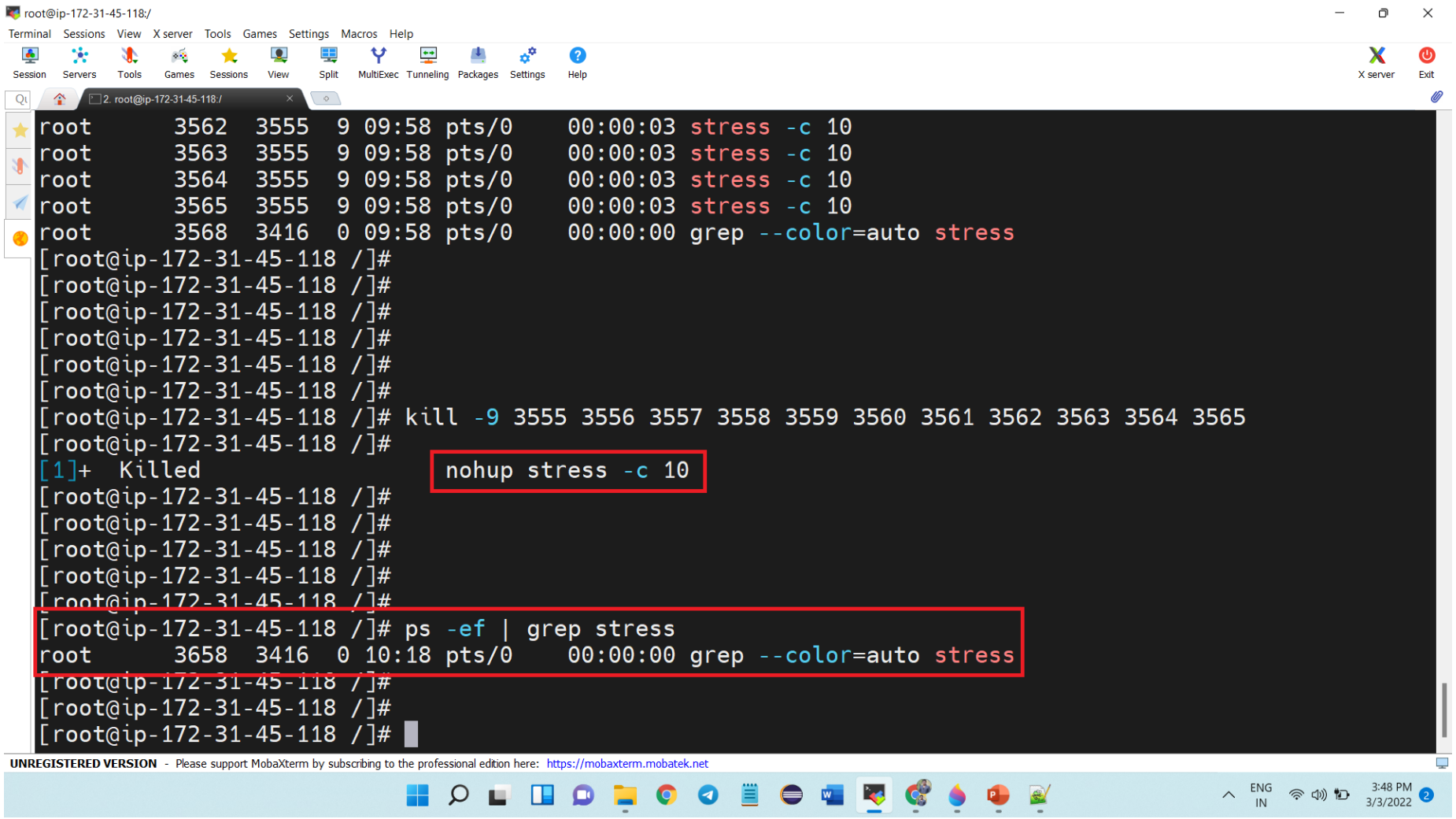


- Now we are reached to maximum capacity

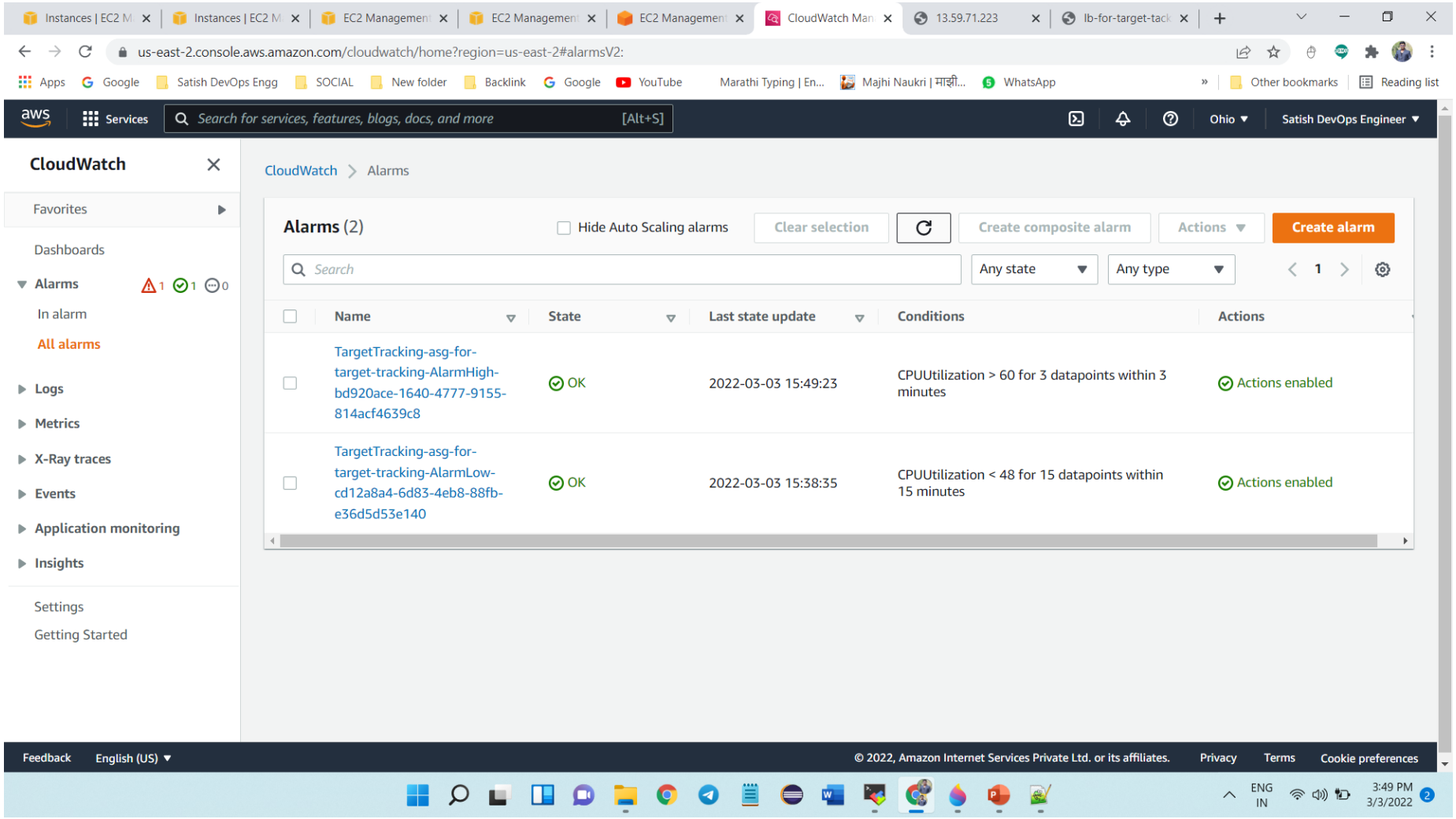

CPU utilization (%)
5 minutes
Average
1h
3h
12h
1d
3d
1w
Custom
Percent
90
80
70
60
50
40
30
20
10
0
09:15
09:20
09:25
09:30
09:35
09:40
09:45
09:50
09:55
10:00
10:05
10:10
10:15
i-0e1cc9ca259eb25d0 (CPU-Utilization-Work)
View in metrics
Close



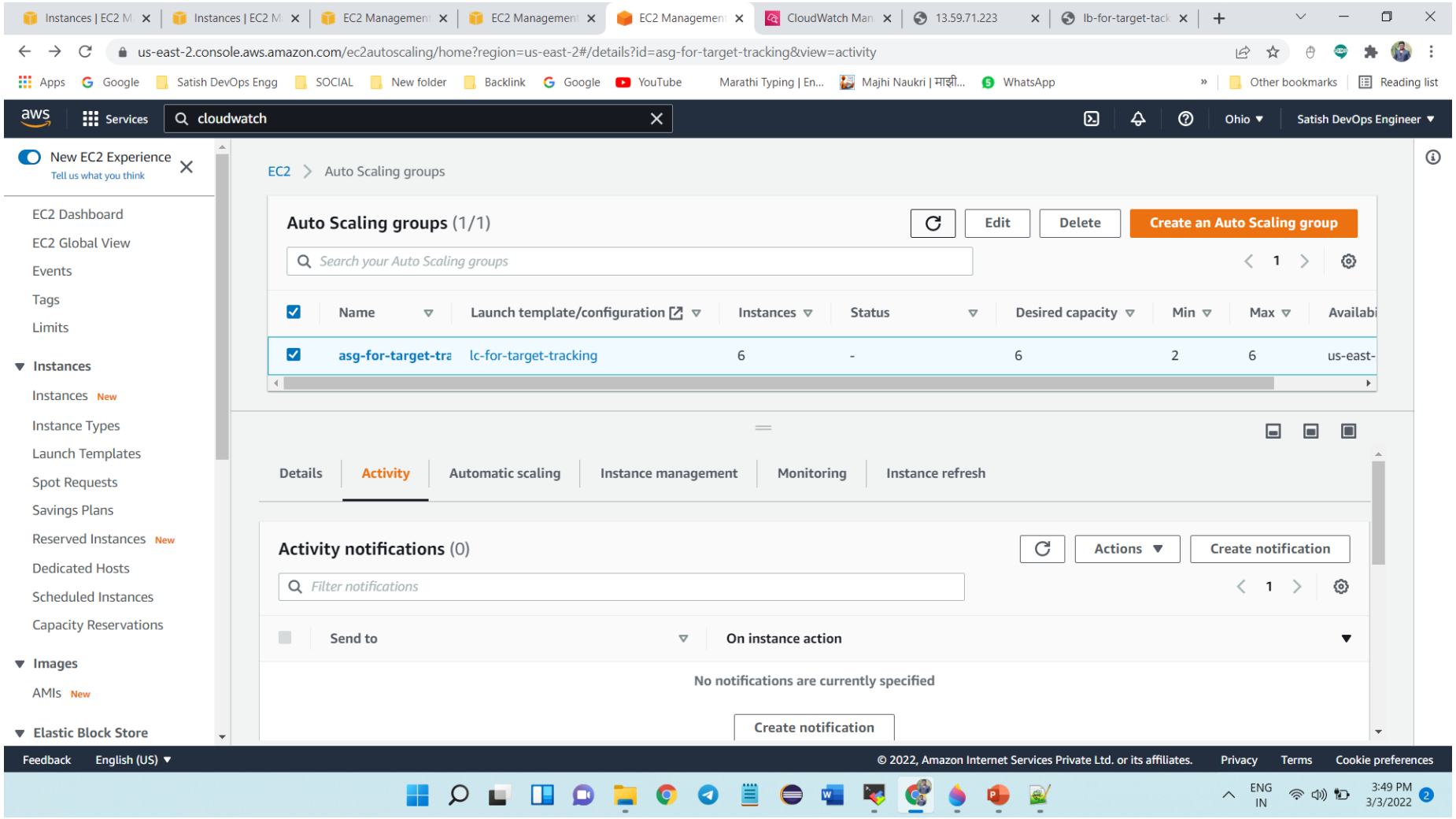
- Now go to MobaXTerm
- Then kill the processed



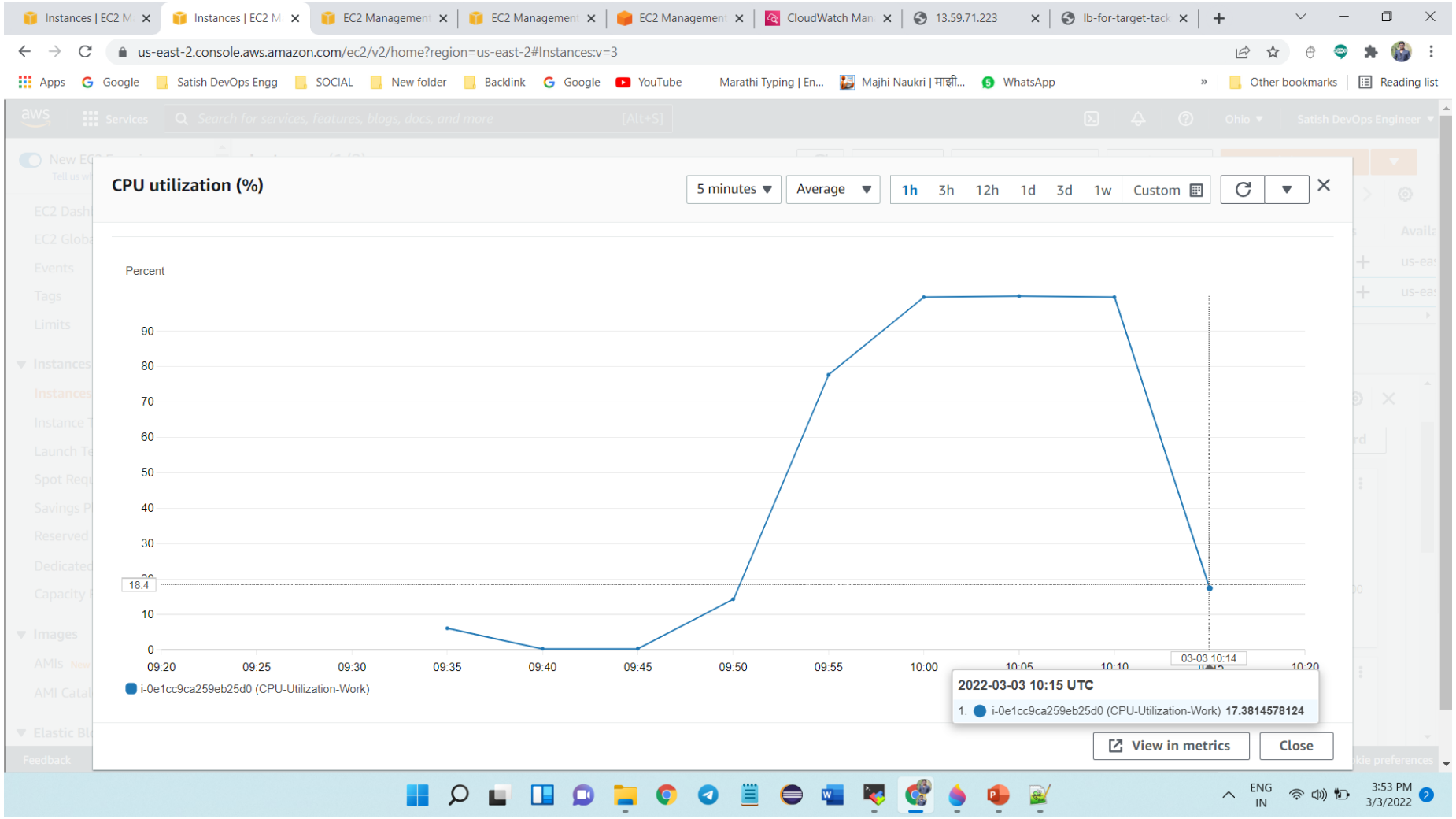
- Now there is no any ongoing process



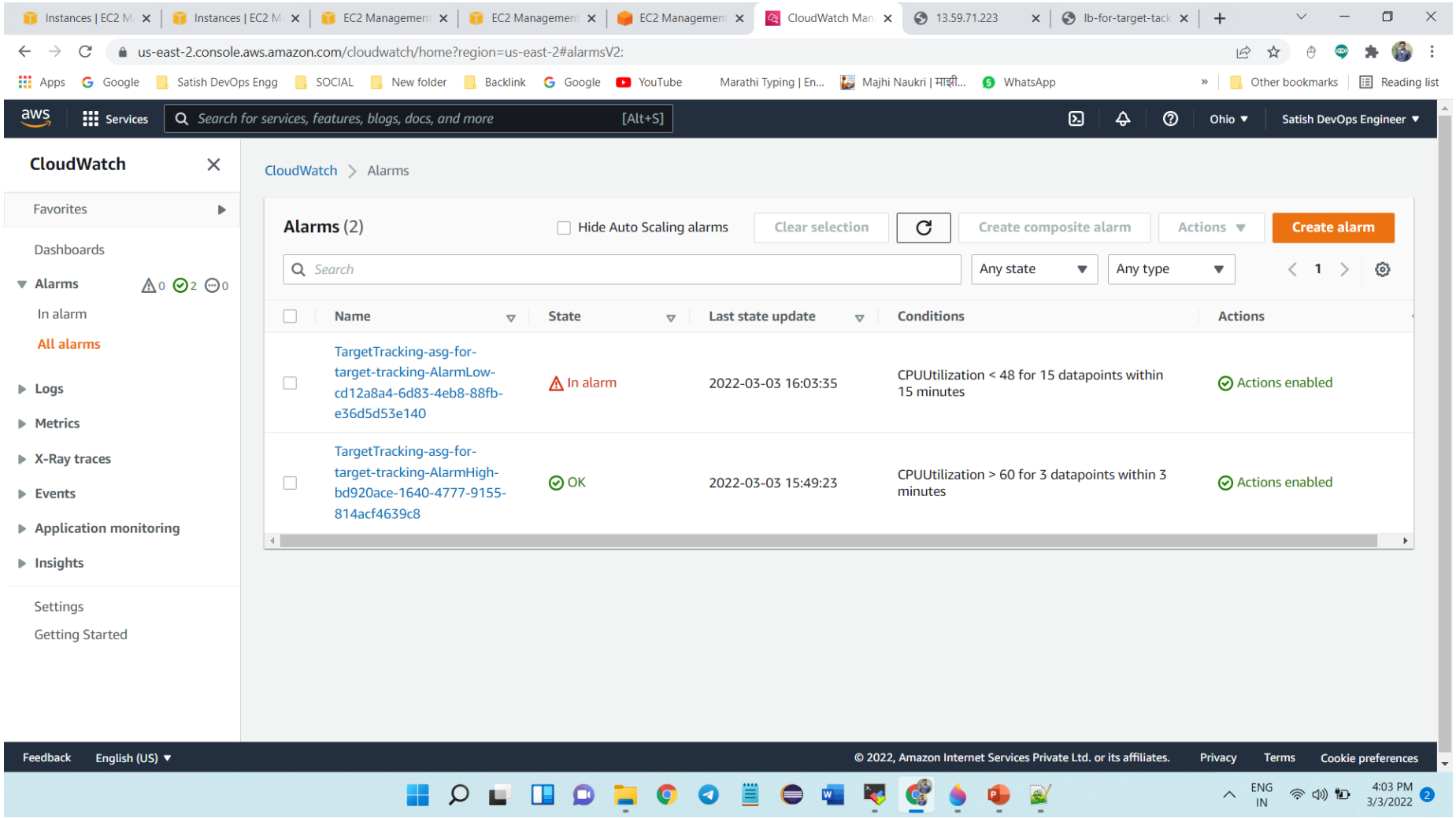
- Here we can see that our alarm state is OK



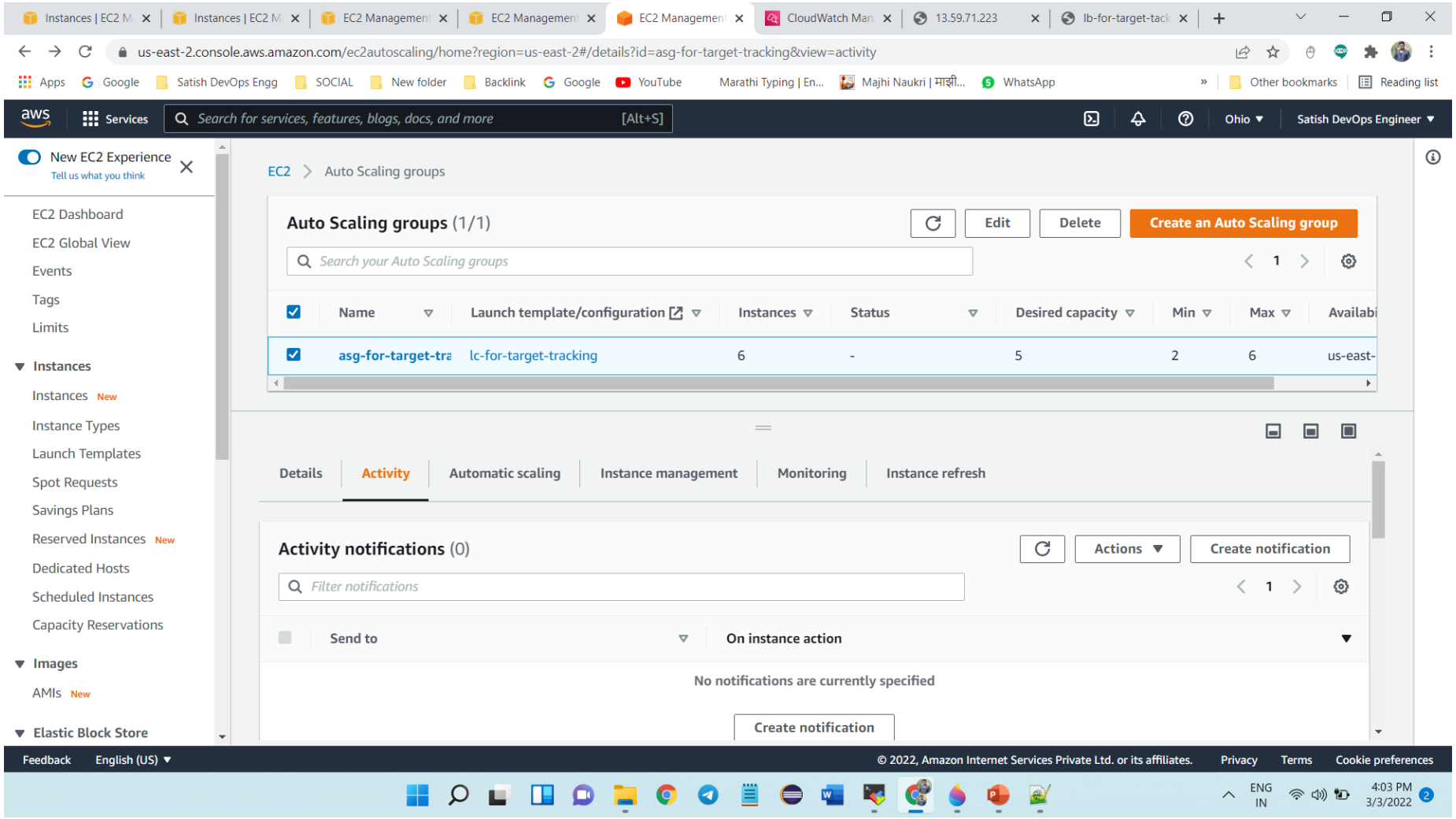
- Wait for Some Time



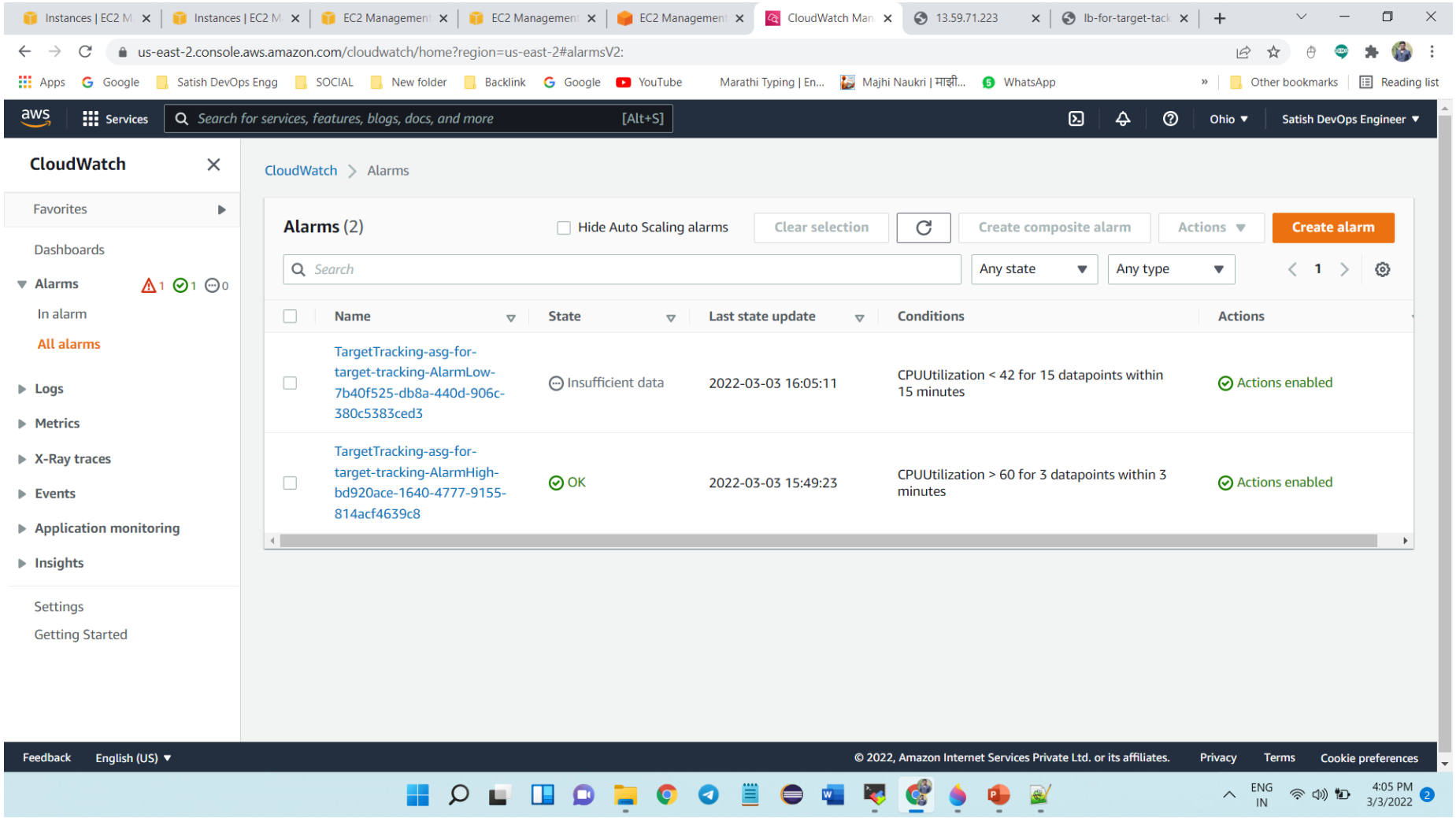
- Now our CPU utilization is decreased



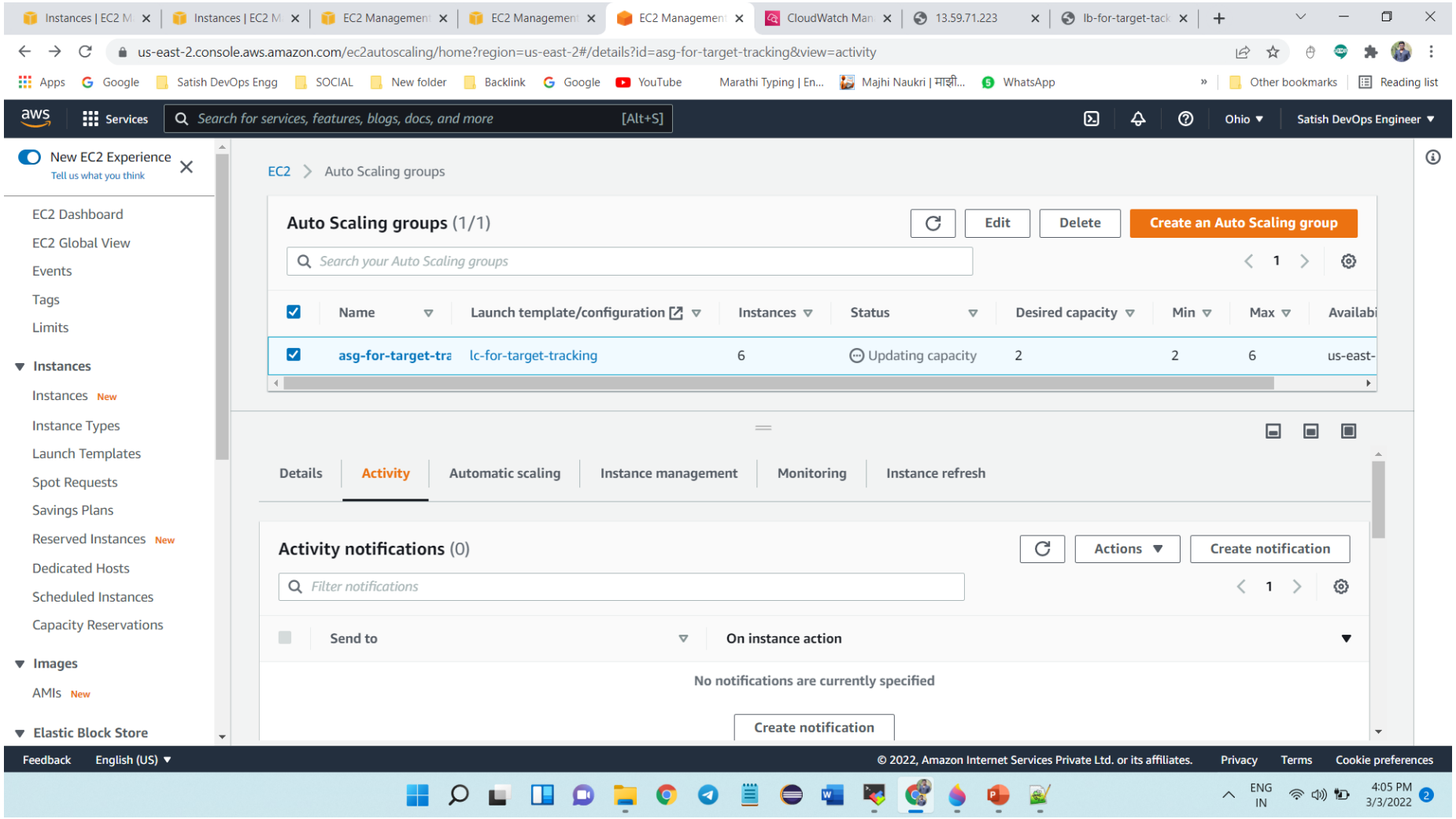
- Now it is changed to In Alarm state



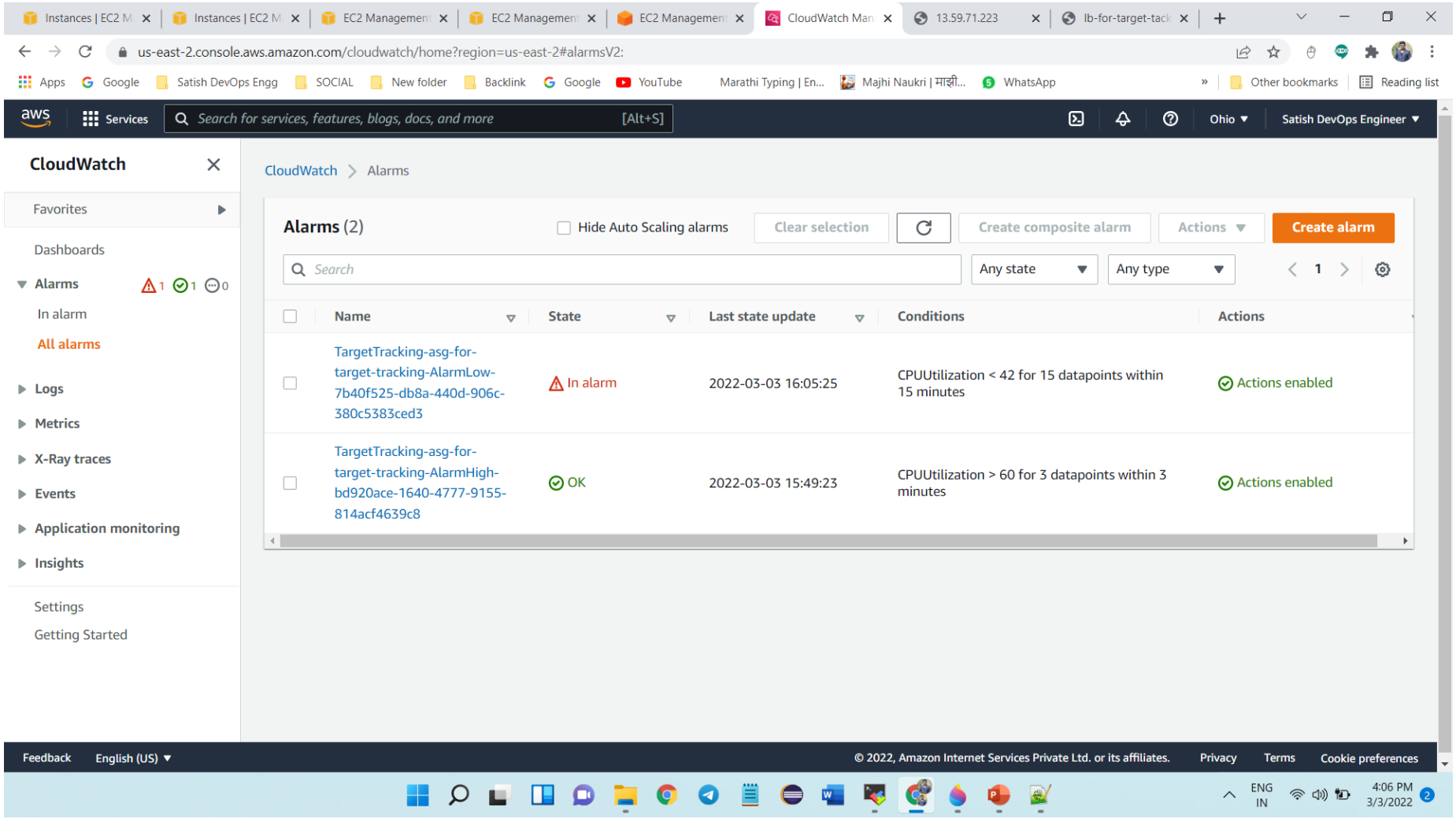
- It has again the desired capacity from 6 to 5



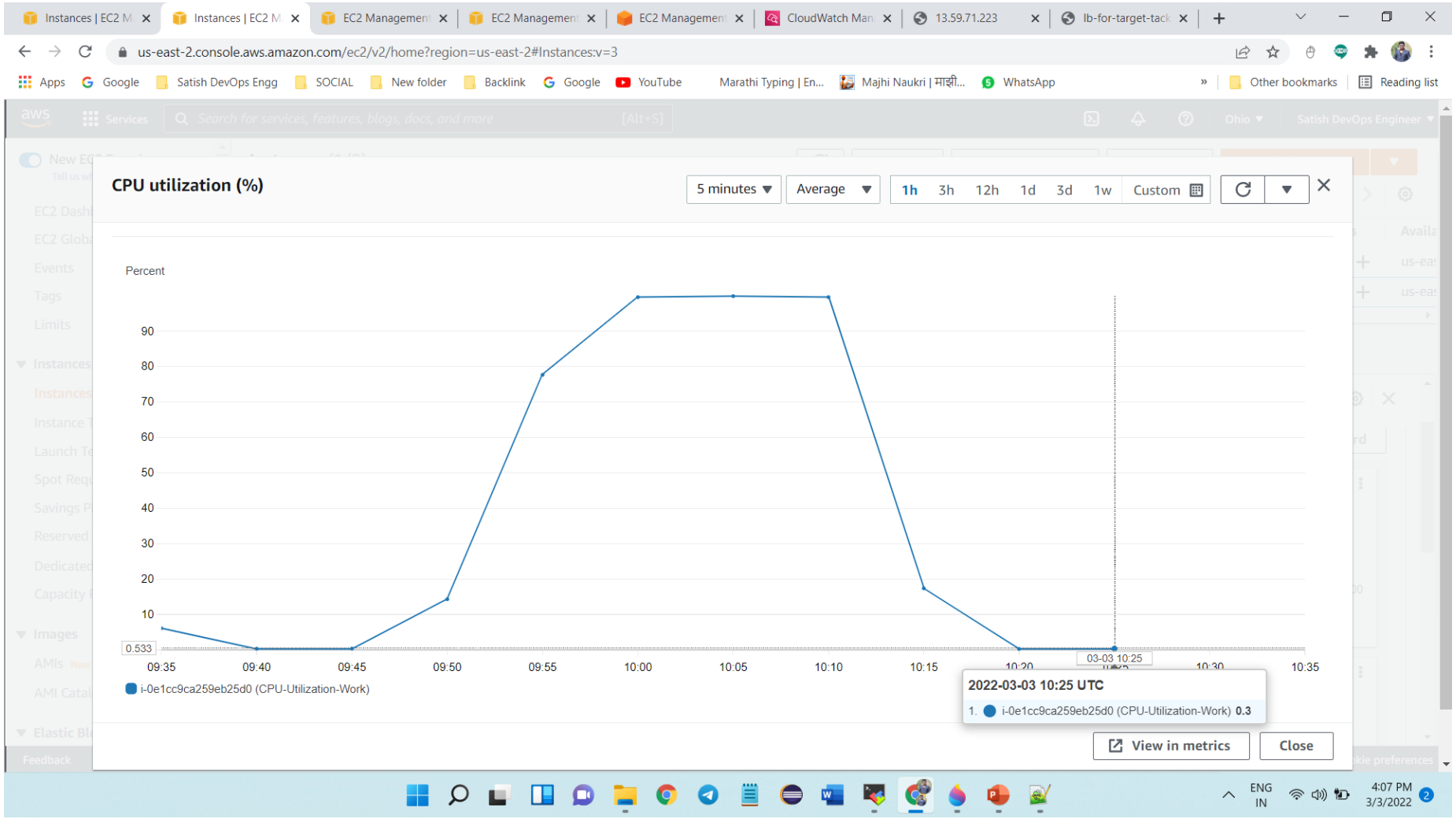
- Please wait for some time



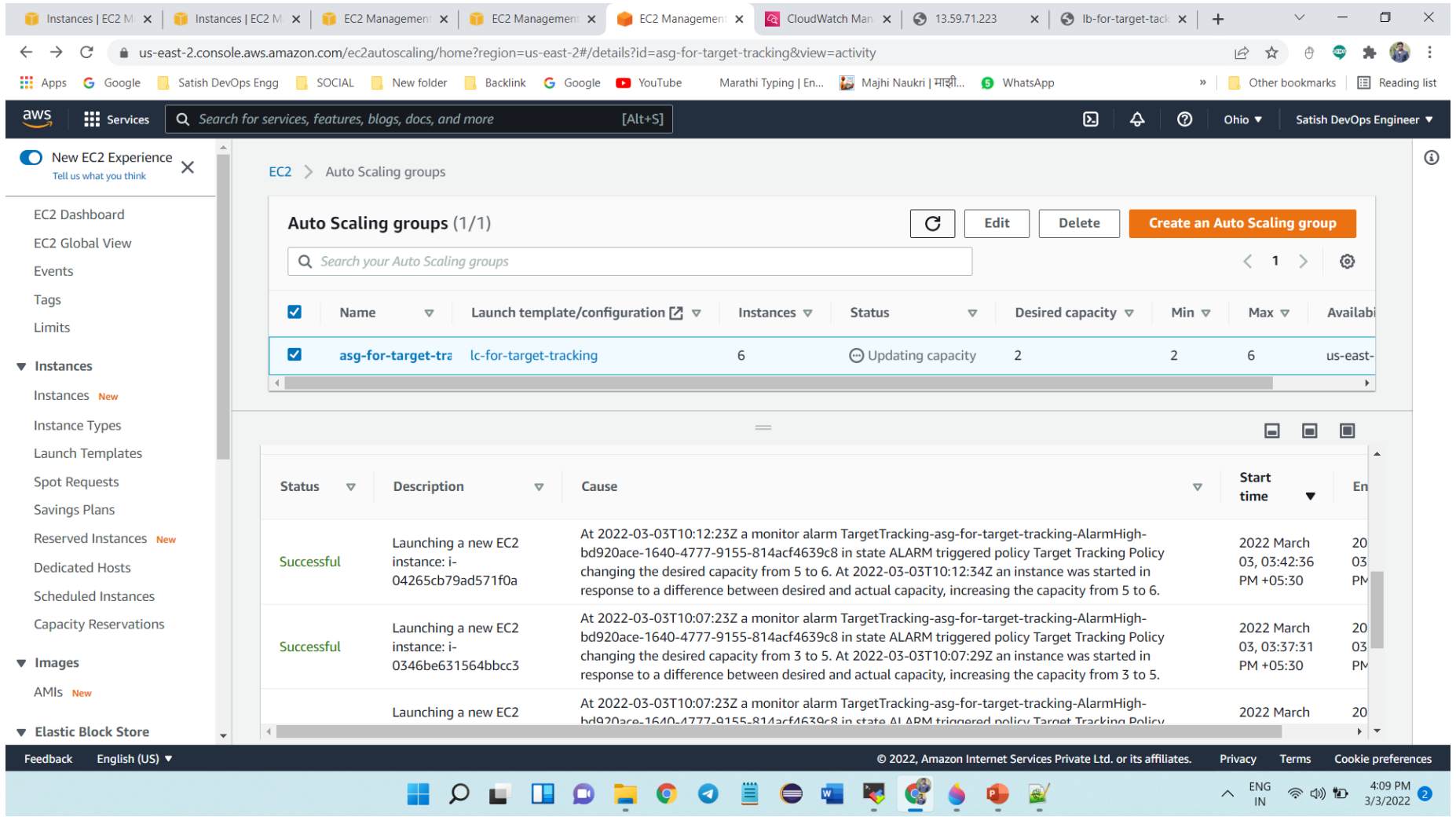
- Here we can see that it is updating capacity
- Now desired capacity is becomes 2

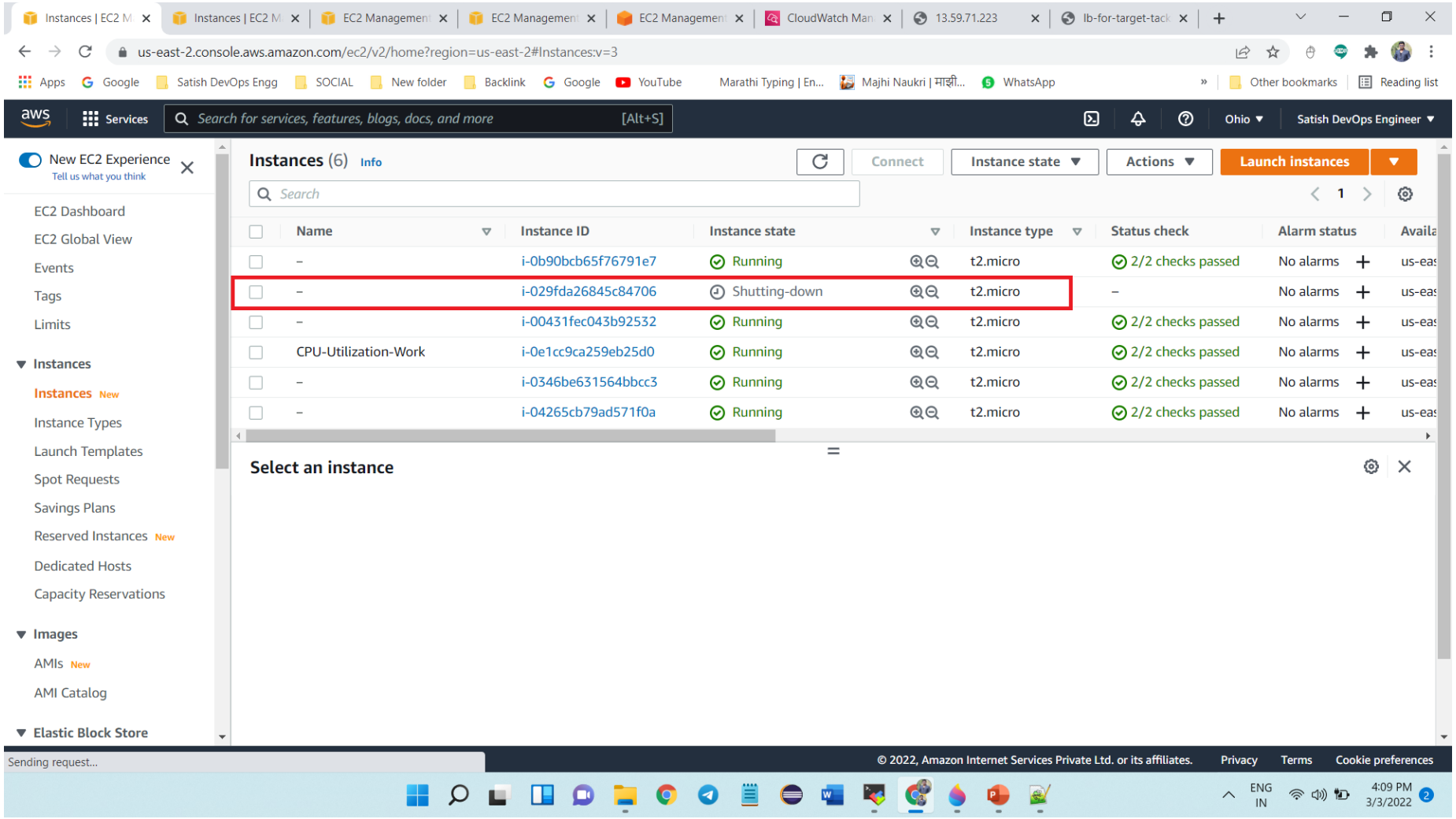


- Wait for some time again

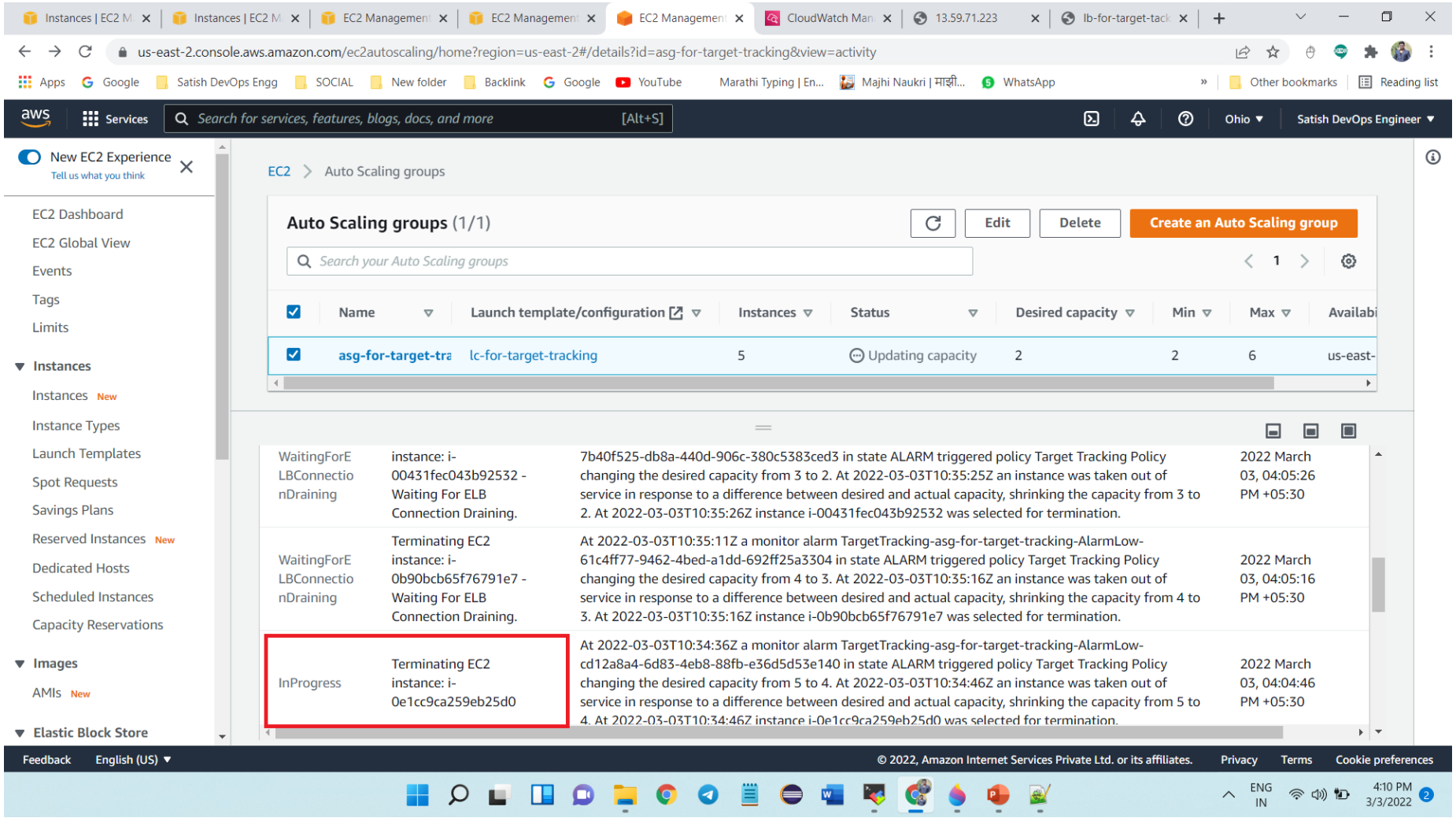


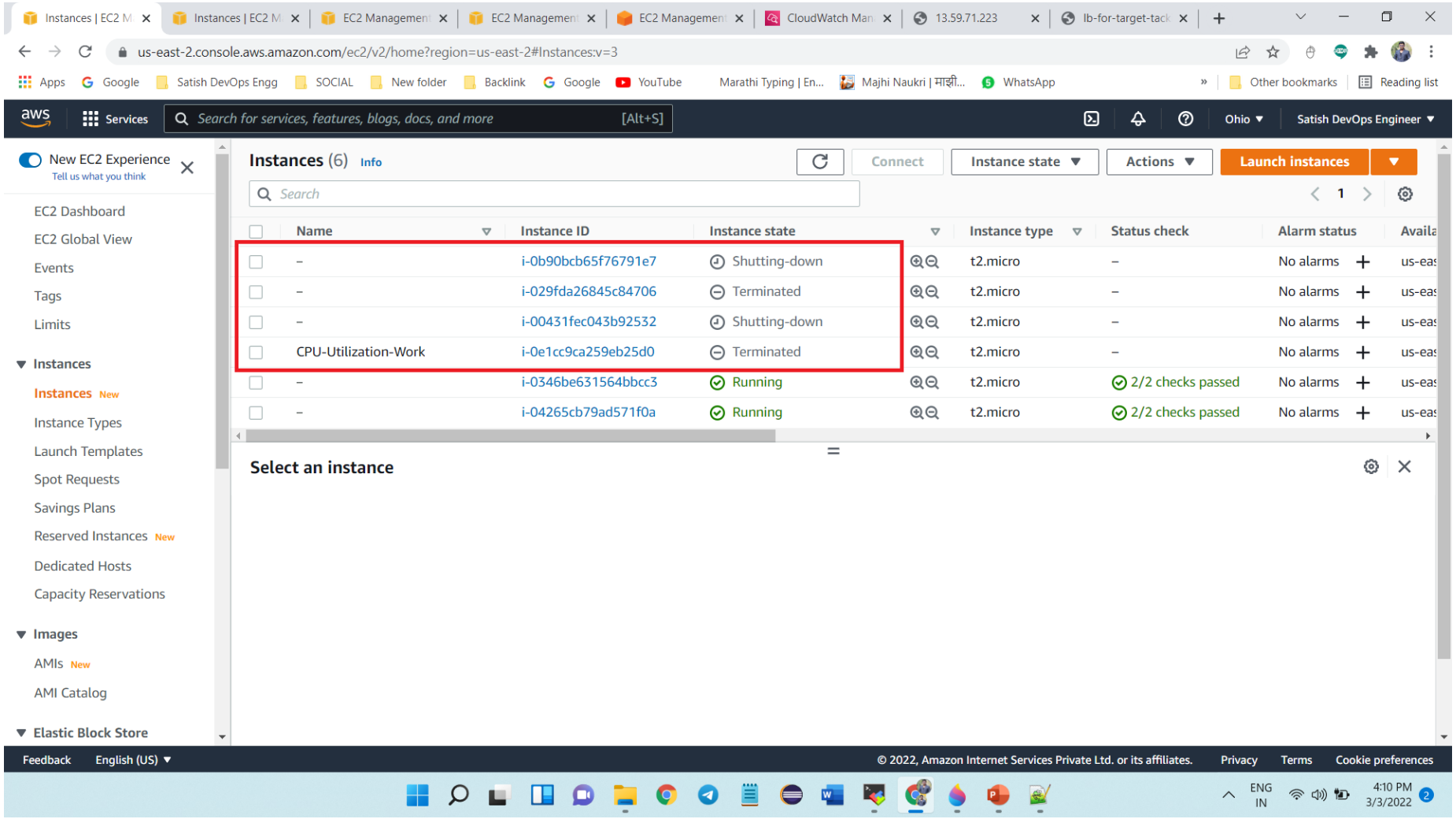
- CPU utilization Becomes 0


Instances | EC2 M
Instances | EC2 M
EC2 Management
EC2 Management
EC2 Management
CloudWatch Man
13.59.71.223
lb-for-target-tack
us-east-2.console.aws.amazon.com/ec2autoscaling/home?region=us-east-2#/details?id=asg-for-target-tracking&view=activity
Apps
Google
Satish DevOps Engg
SOCIAL
New folder
Backlink
Google
YouTube
Marathi Typing | En...
Majhi Naukri | माझी...
WhatsApp
Other bookmarks
Reading list
aws
Services
Search for services, features, blogs, docs, and more
[Alt+S]
Ohio
Satish DevOps Engineer
New EC2 Experience
Tell us what you think
EC2 Dashboard
EC2 Global View
Events
Tags
Limits
Instances
Instances New
Instance Types
Launch Templates
Spot Requests
Savings Plans
Reserved Instances New
Dedicated Hosts
Scheduled Instances
Capacity Reservations
Images
AMIs New
Elastic Block Store
EC2 > Auto Scaling groups
Auto Scaling groups (1/1)
Edit
Delete
Create an Auto Scaling group
Search your Auto Scaling groups
1
Name
Launch template/configuration
Instances
Status
Desired capacity
Min
Max
Availabi
asg-for-target-tra
lc-for-target-tracking
6
Updating capacity
2
2
6
us-east-
Status
Description
Cause
Start time
En
Successful
Launching a new EC2 instance: i-04265cb79ad571f0a
At 2022-03-03T10:12:23Z a monitor alarm TargetTracking-asg-for-target-tracking-AlarmHigh-bd920ace-1640-4777-9155-814acf4639c8 in state ALARM triggered policy Target Tracking Policy changing the desired capacity from 5 to 6. At 2022-03-03T10:12:34Z an instance was started in response to a difference between desired and actual capacity, increasing the capacity from 5 to 6.
2022 March 03, 03:42:36 PM +05:30
Successful
Launching a new EC2 instance: i-0346be631564bbcc3
At 2022-03-03T10:07:23Z a monitor alarm TargetTracking-asg-for-target-tracking-AlarmHigh-bd920ace-1640-4777-9155-814acf4639c8 in state ALARM triggered policy Target Tracking Policy changing the desired capacity from 3 to 5. At 2022-03-03T10:07:29Z an instance was started in response to a difference between desired and actual capacity, increasing the capacity from 3 to 5.
2022 March 03, 03:37:31 PM +05:30
Launching a new EC2
At 2022-03-03T10:07:23Z a monitor alarm TargetTracking-asg-for-target-tracking-AlarmHigh-
2022 March
Feedback
English (US)
© 2022, Amazon Internet Services Private Ltd. or its affiliates.
Privacy
Terms
Cookie preferences
ENG IN
4:09 PM 3/3/2022



- Now it is terminating 1 random instance automatically


Instances | EC2 M
Instances | EC2 M
EC2 Management
EC2 Management
EC2 Management
CloudWatch Man
13.59.71.223
lb-for-target-tack
us-east-2.console.aws.amazon.com/ec2autoscaling/home?region=us-east-2#/details?id=asg-for-target-tracking&view=activity
Apps
Google
Satish DevOps Engg
SOCIAL
New folder
Backlink
Google
YouTube
Marathi Typing | En...
Majhi Naukri | माझी...
WhatsApp
Other bookmarks
Reading list
aws
Services
Search for services, features, blogs, docs, and more
[Alt+S]
Ohio
Satish DevOps Engineer
New EC2 Experience
Tell us what you think
EC2 Dashboard
EC2 Global View
Events
Tags
Limits
Instances
Instances New
Instance Types
Launch Templates
Spot Requests
Savings Plans
Reserved Instances New
Dedicated Hosts
Scheduled Instances
Capacity Reservations
Images
AMIs New
Elastic Block Store
EC2 > Auto Scaling groups
Auto Scaling groups (1/1)
Edit
Delete
Create an Auto Scaling group
Search your Auto Scaling groups
1
Name
Launch template/configuration
Instances
Status
Desired capacity
Min
Max
Availabi
asg-for-target-tra
lc-for-target-tracking
5
Updating capacity
2
2
6
us-east-
WaitingForELBConnectionDraining
instance: i-00431fec043b92532 - Waiting For ELB Connection Draining.
7b40f525-db8a-440d-906c-380c5383ced3 in state ALARM triggered policy Target Tracking Policy changing the desired capacity from 3 to 2. At 2022-03-03T10:35:25Z an instance was taken out of service in response to a difference between desired and actual capacity, shrinking the capacity from 3 to 2. At 2022-03-03T10:35:26Z instance i-00431fec043b92532 was selected for termination.
2022 March 03, 04:05:26 PM +05:30
WaitingForELBConnectionDraining
Terminating EC2 instance: i-0b90bcb65f76791e7 - Waiting For ELB Connection Draining.
At 2022-03-03T10:35:11Z a monitor alarm TargetTracking-asg-for-target-tracking-AlarmLow-61c4ff77-9462-4bed-a1dd-692ff25a3304 in state ALARM triggered policy Target Tracking Policy changing the desired capacity from 4 to 3. At 2022-03-03T10:35:16Z an instance was taken out of service in response to a difference between desired and actual capacity, shrinking the capacity from 4 to 3. At 2022-03-03T10:35:16Z instance i-0b90bcb65f76791e7 was selected for termination.
2022 March 03, 04:05:16 PM +05:30
InProgress
Terminating EC2 instance: i-0e1cc9ca259eb25d0
At 2022-03-03T10:34:36Z a monitor alarm TargetTracking-asg-for-target-tracking-AlarmLow-cd12a8a4-6d83-4eb8-88fb-e36d5d53e140 in state ALARM triggered policy Target Tracking Policy changing the desired capacity from 5 to 4. At 2022-03-03T10:34:46Z an instance was taken out of service in response to a difference between desired and actual capacity, shrinking the capacity from 5 to 4. At 2022-03-03T10:34:46Z instance i-0e1cc9ca259eb25d0 was selected for termination.
2022 March 03, 04:04:46 PM +05:30
Feedback
English (US)
© 2022, Amazon Internet Services Private Ltd. or its affiliates.
Privacy
Terms
Cookie preferences
ENG IN
4:10 PM 3/3/2022



- Now it has terminated total 4 random instances automatically

In this way we have seen that how Target tracking Policy works