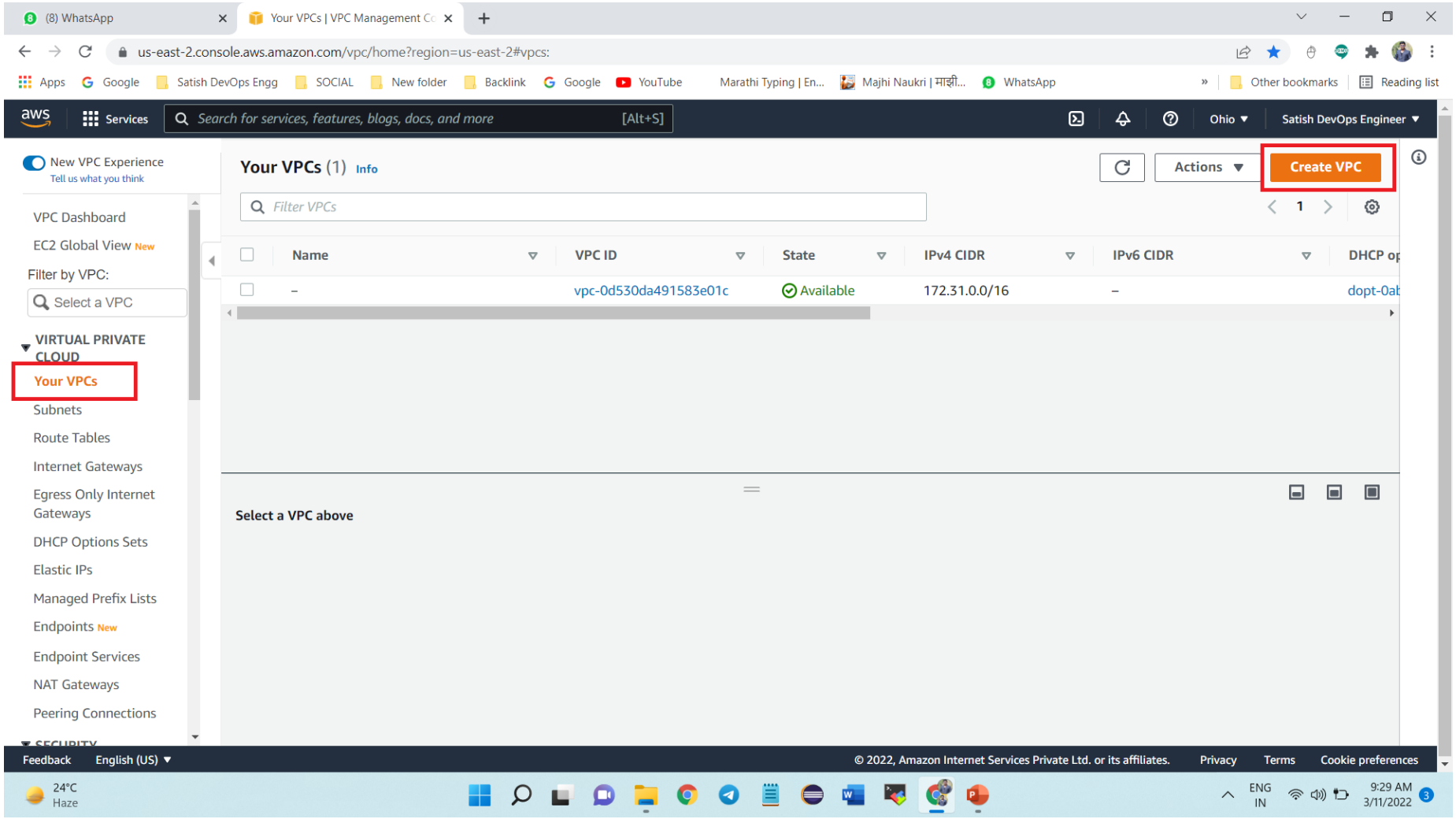
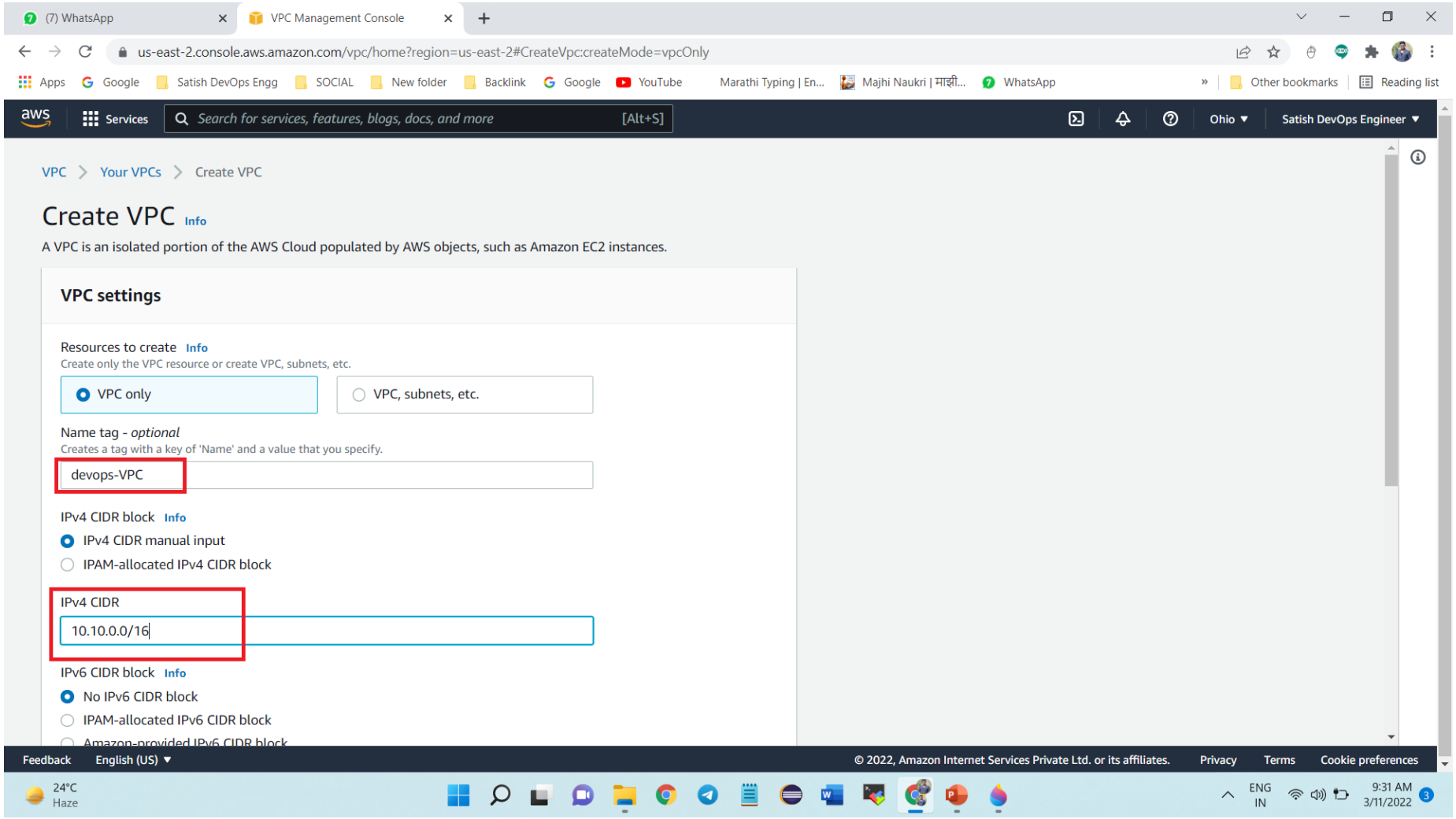
019 Practical

VPC, Subnet, Internet Gateway, Route Table and NAT Gateway

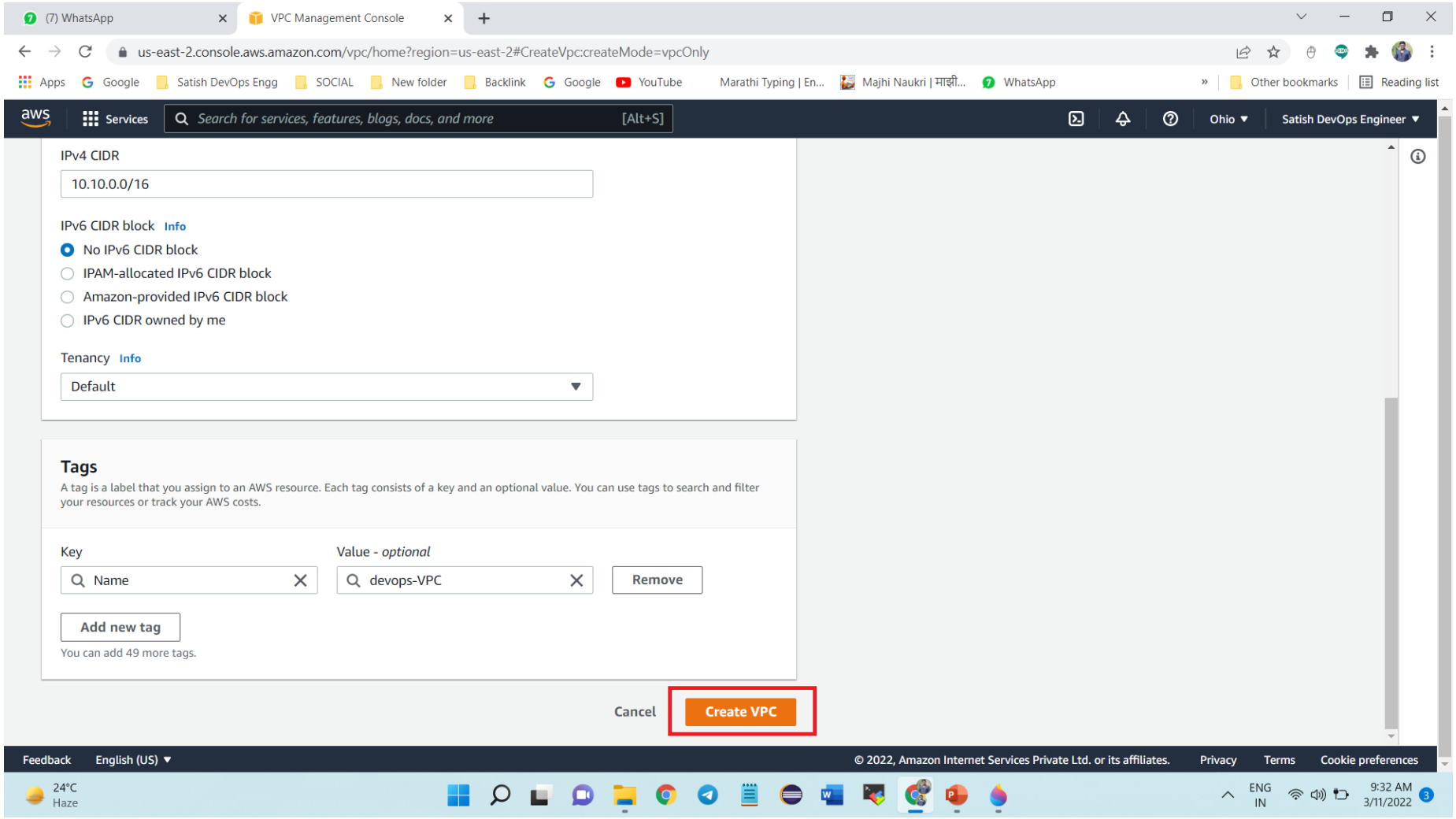
Now we have to create VPC



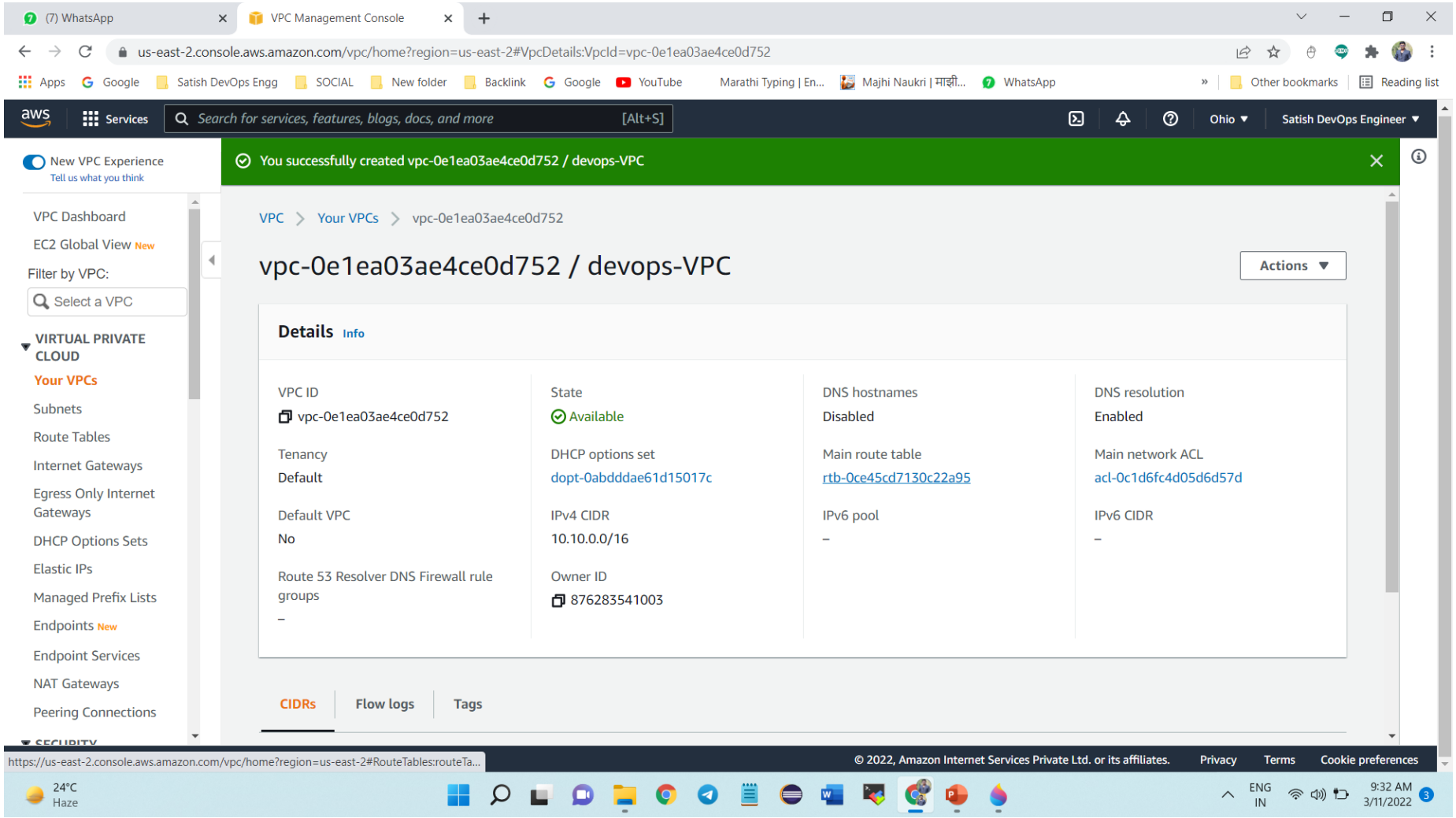
- Go to VPC
- Click on Create VPC



- Give tag
- Enter proper CIDR as per requirement

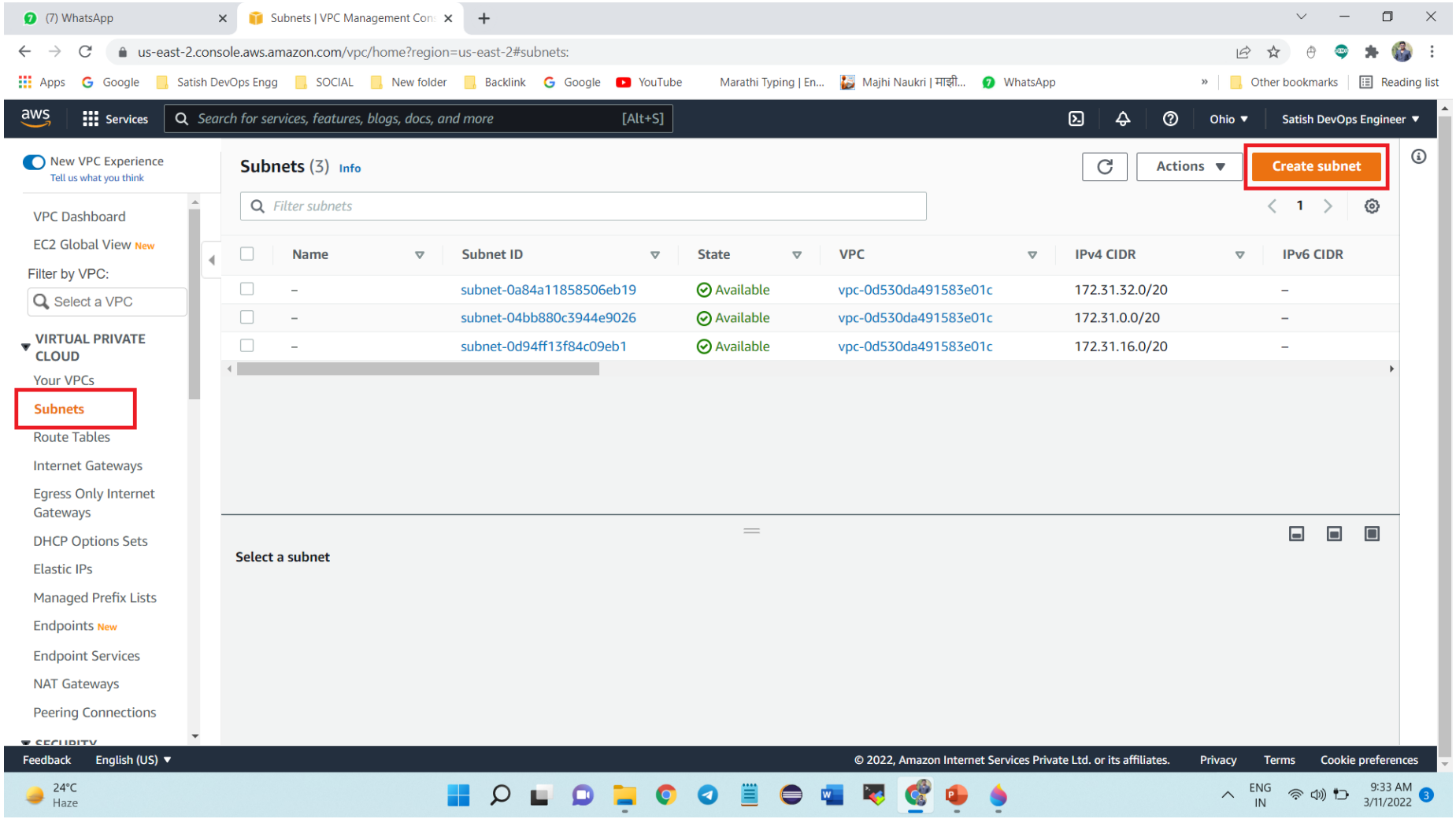


- Click on Create VPC

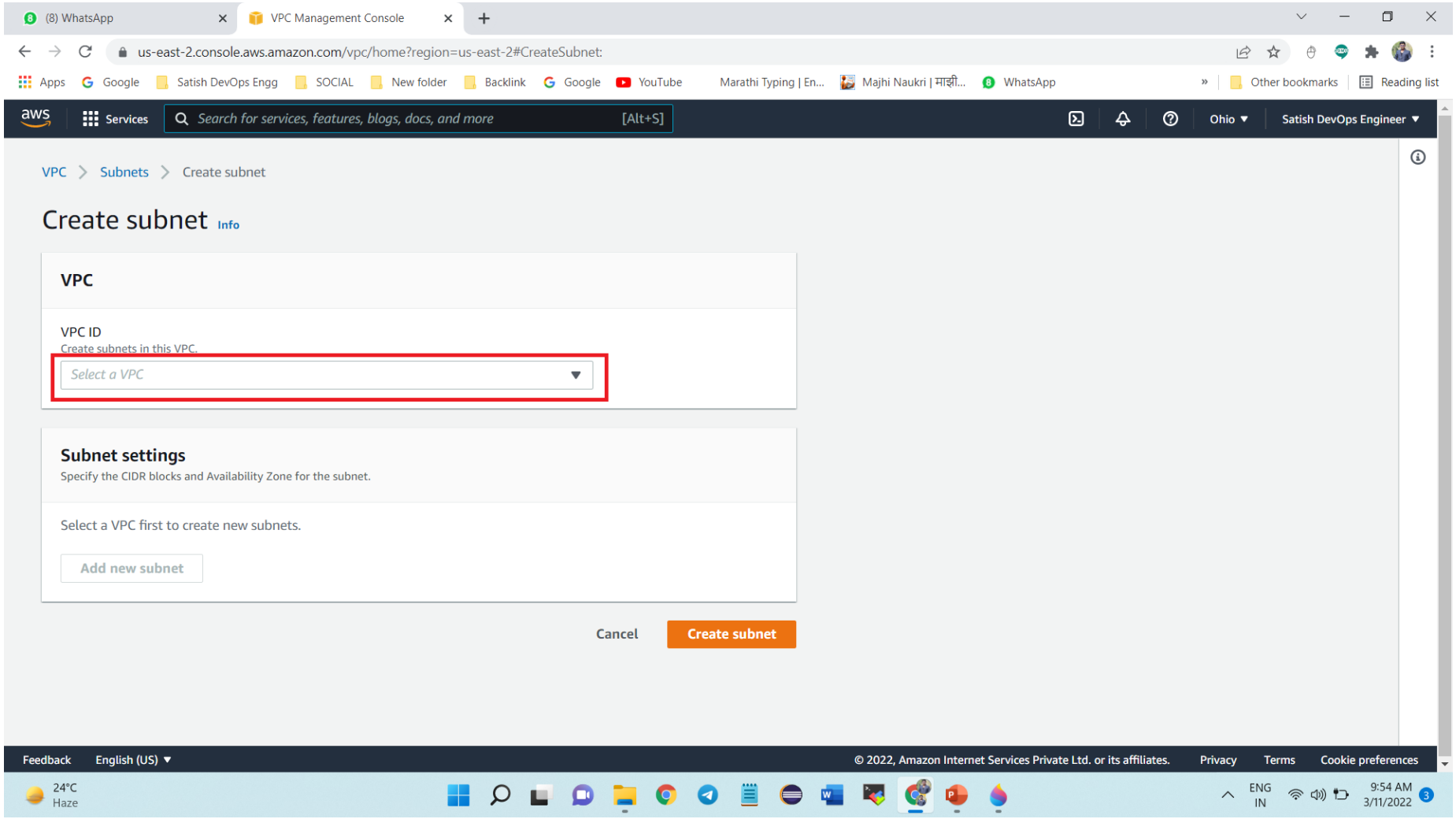


- Our VPC Created Successfully

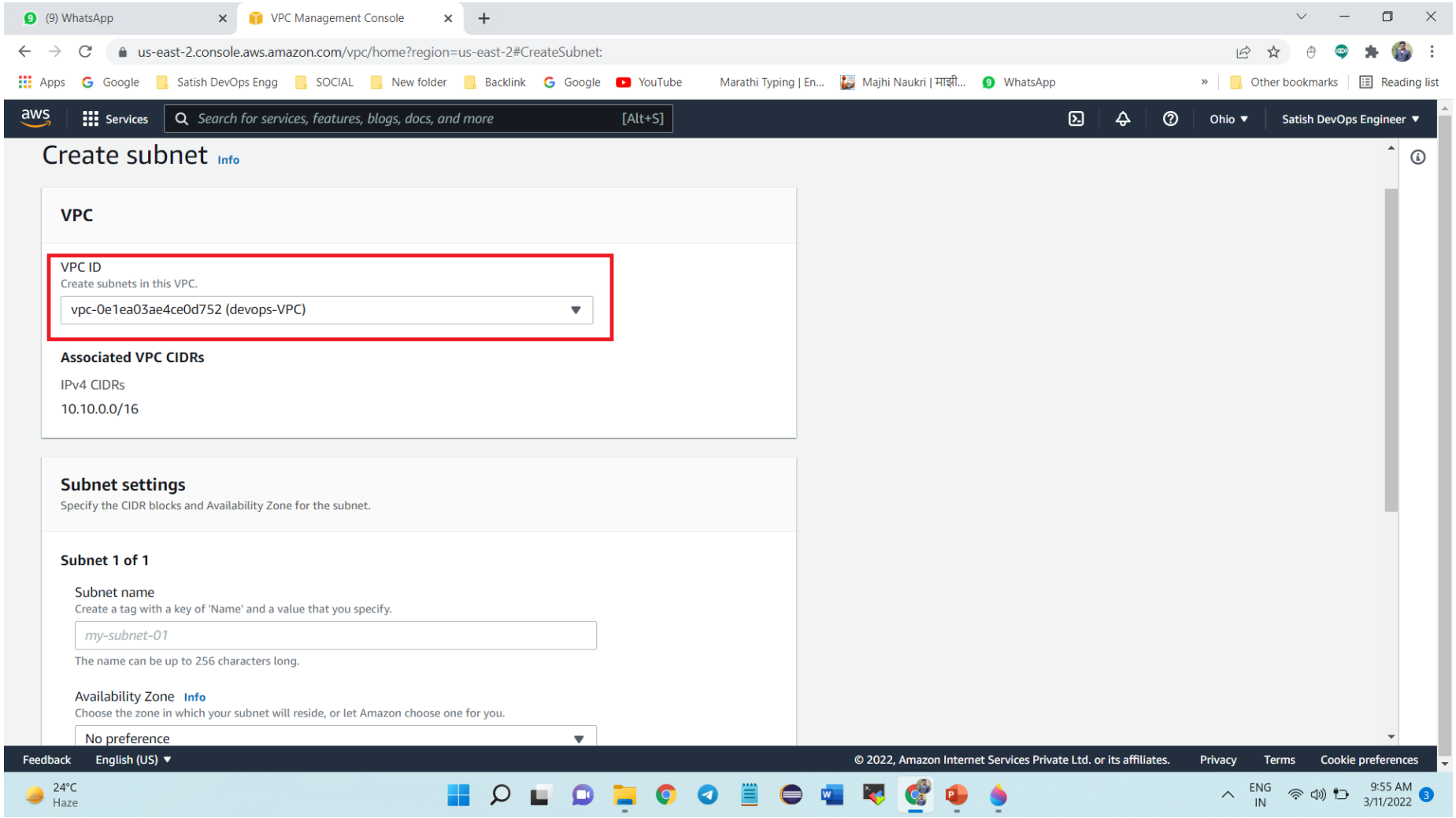
Now we have to create Subnet

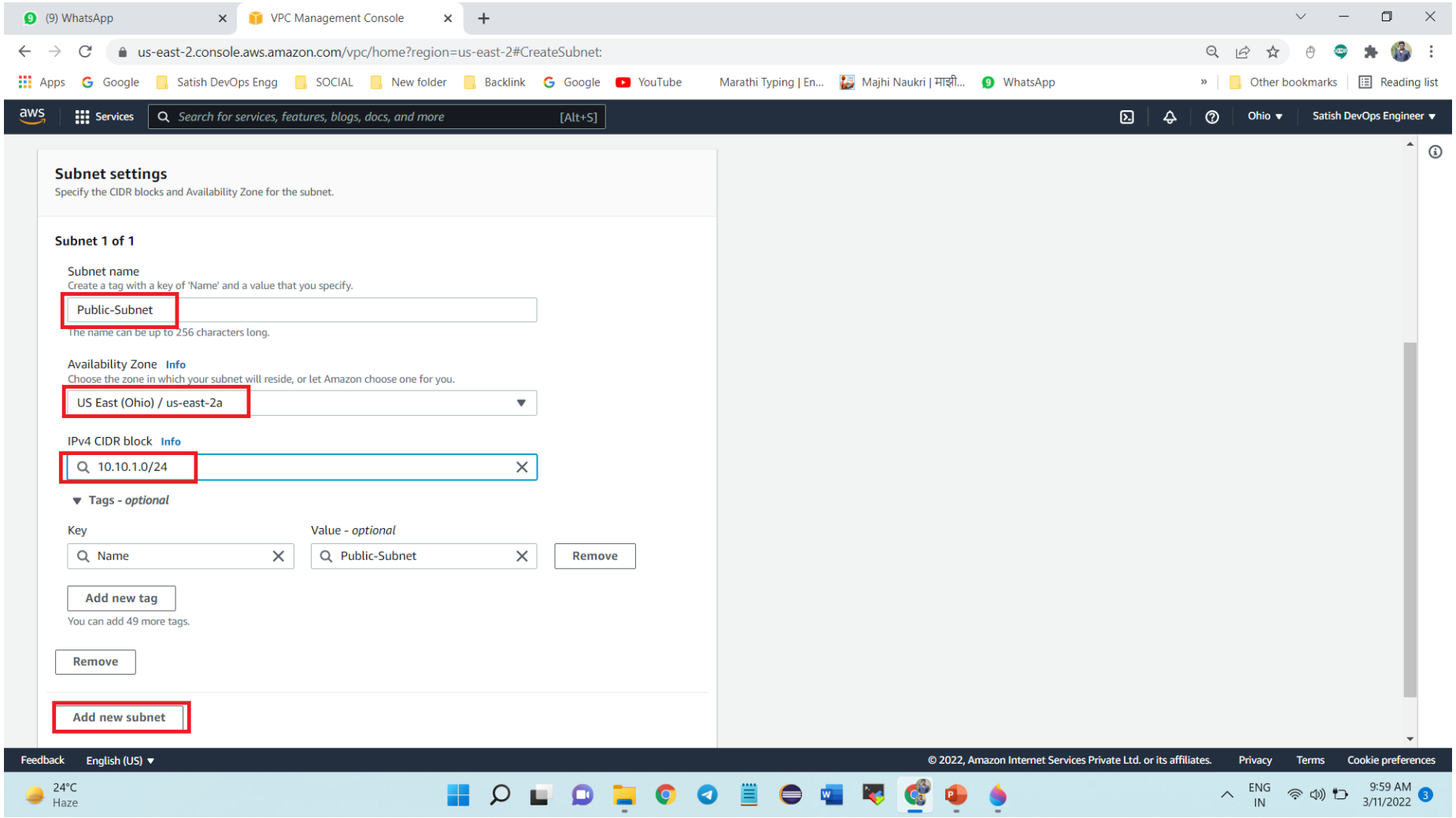


- Go to Subnets
- Click on Create Subnet

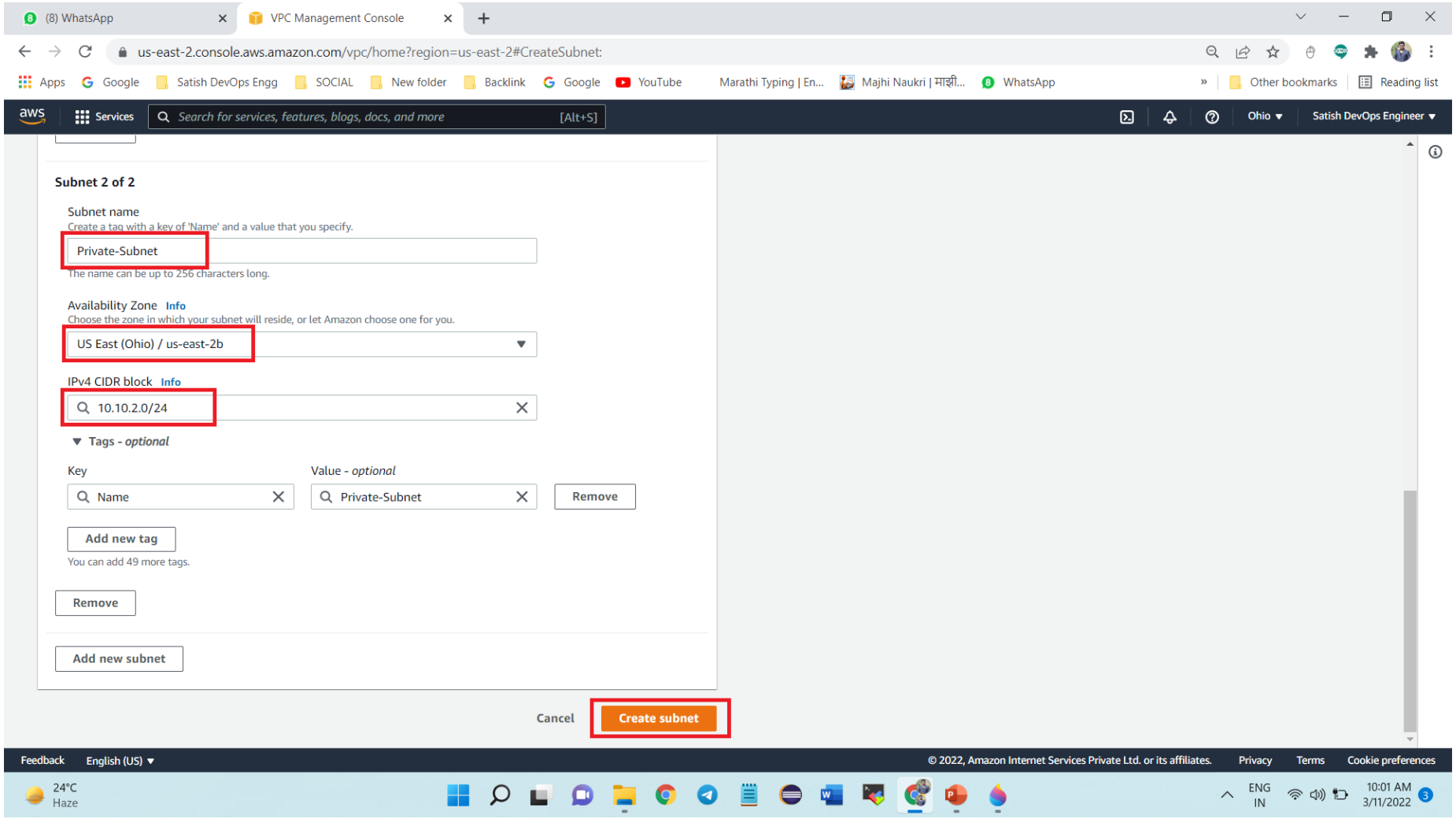


- Select our own created VPC

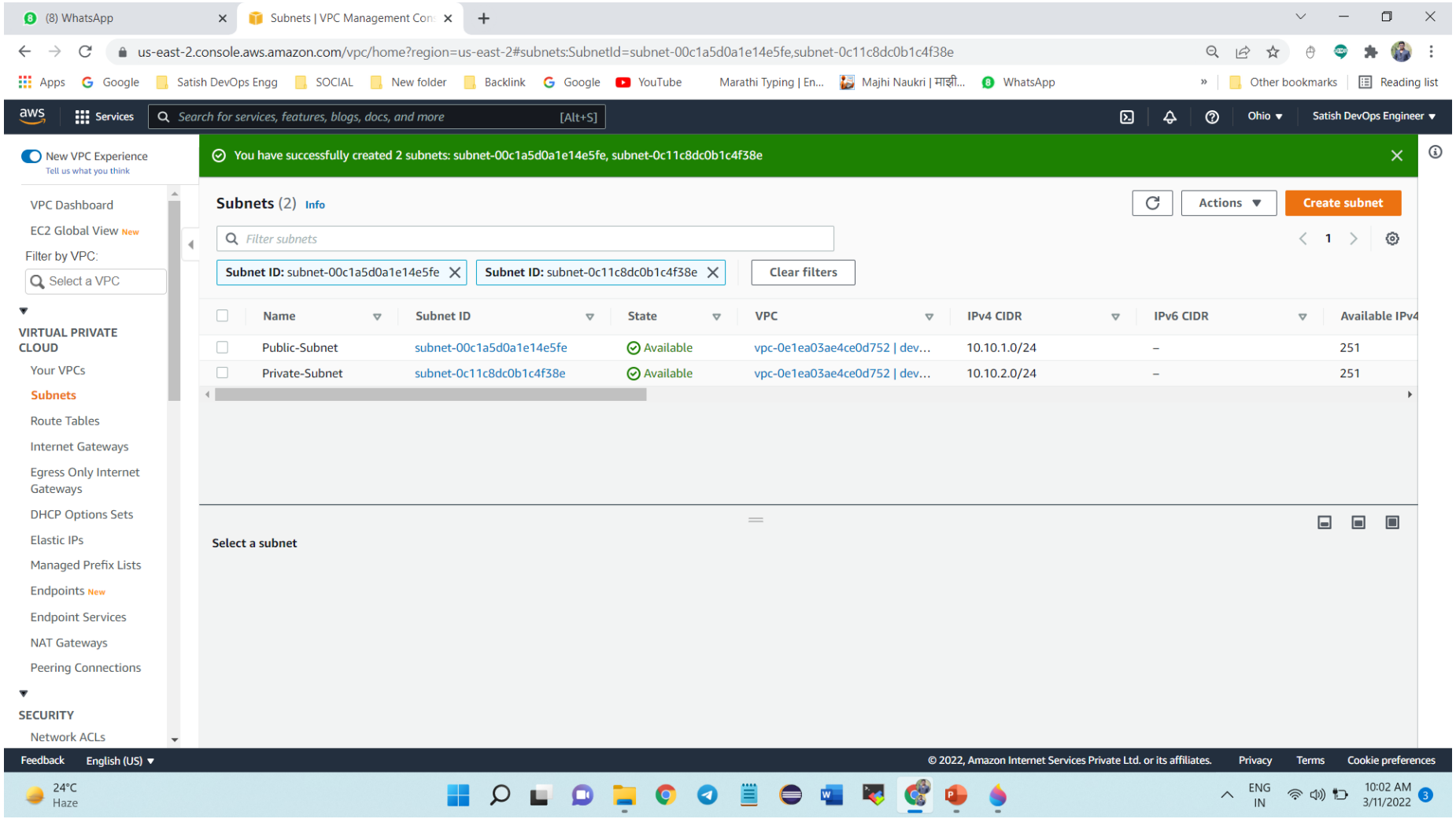

(9) WhatsApp
VPC Management Console
us-east-2.console.aws.amazon.com/vpc/home?region=us-east-2#CreateSubnet:
Apps
Google
Satish DevOps Engg
SOCIAL
New folder
Backlink
Google
YouTube
Marathi Typing | En...
Majhi Naukri | माझी...
WhatsApp
Other bookmarks
Reading list
aws
Services
Search for services, features, blogs, docs, and more
[Alt+S]
Ohio
Satish DevOps Engineer
Create subnet Info
VPC
VPC ID
Create subnets in this VPC.
vpc-0e1ea03ae4ce0d752 (devops-VPC)
Associated VPC CIDRs
IPv4 CIDRs
10.10.0.0/16
Subnet settings
Specify the CIDR blocks and Availability Zone for the subnet.
Subnet 1 of 1
Subnet name
Create a tag with a key of 'Name' and a value that you specify.
my-subnet-01
The name can be up to 256 characters long.
Availability Zone Info
Choose the zone in which your subnet will reside, or let Amazon choose one for you.
No preference
Feedback
English (US)
© 2022, Amazon Internet Services Private Ltd. or its affiliates.
Privacy
Terms
Cookie preferences
24°C
Haze
ENG
IN
9:55 AM
3/11/2022



- **Give Subnet Name as Public**
- **Select AZ**
- **Enter proper CIDR for Subnet**
- **Click on Add New Subnet**

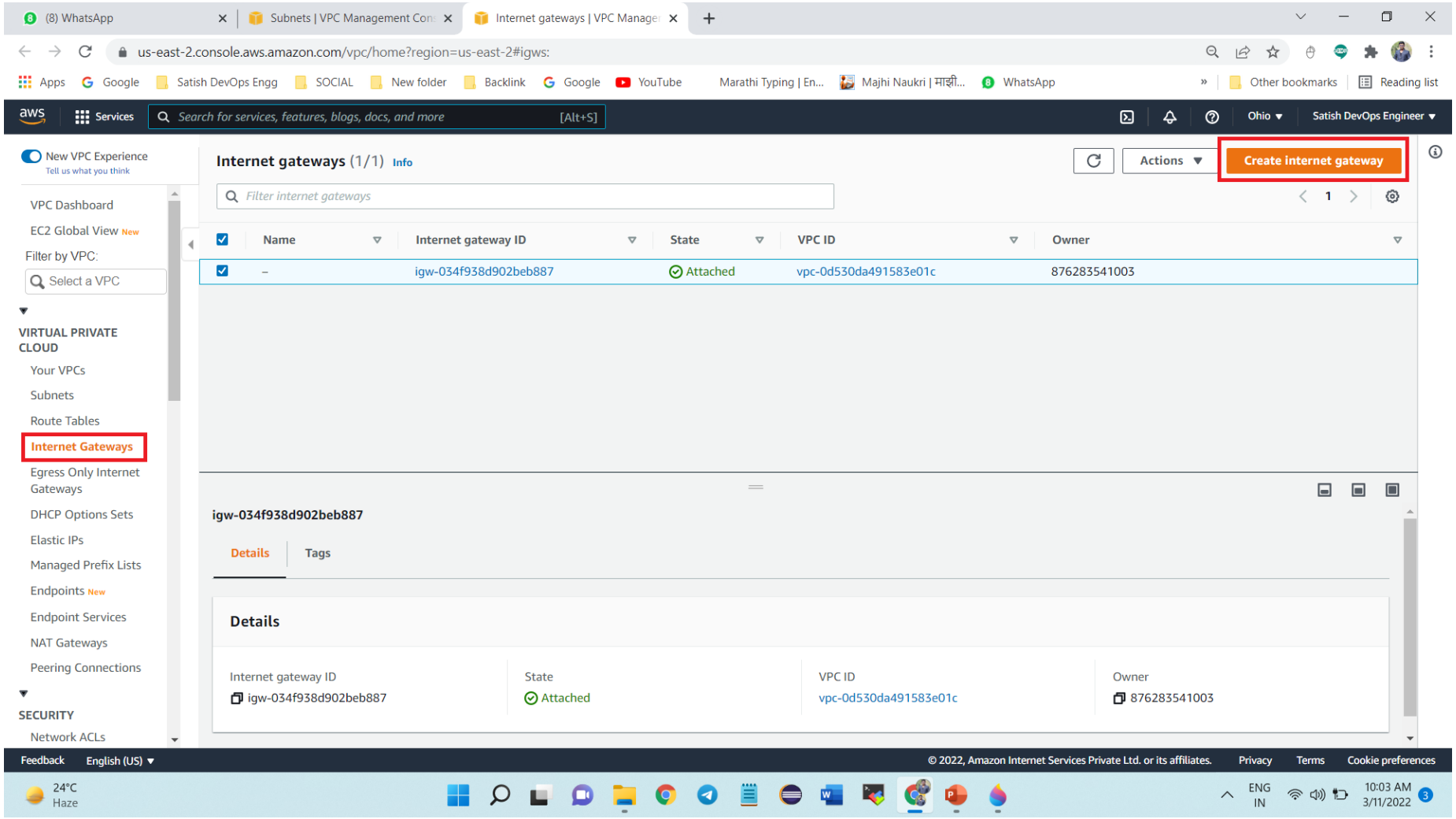


- **Give Subnet Name as Private**
- **Select AZ**
- **Enter proper CIDR for Subnet**

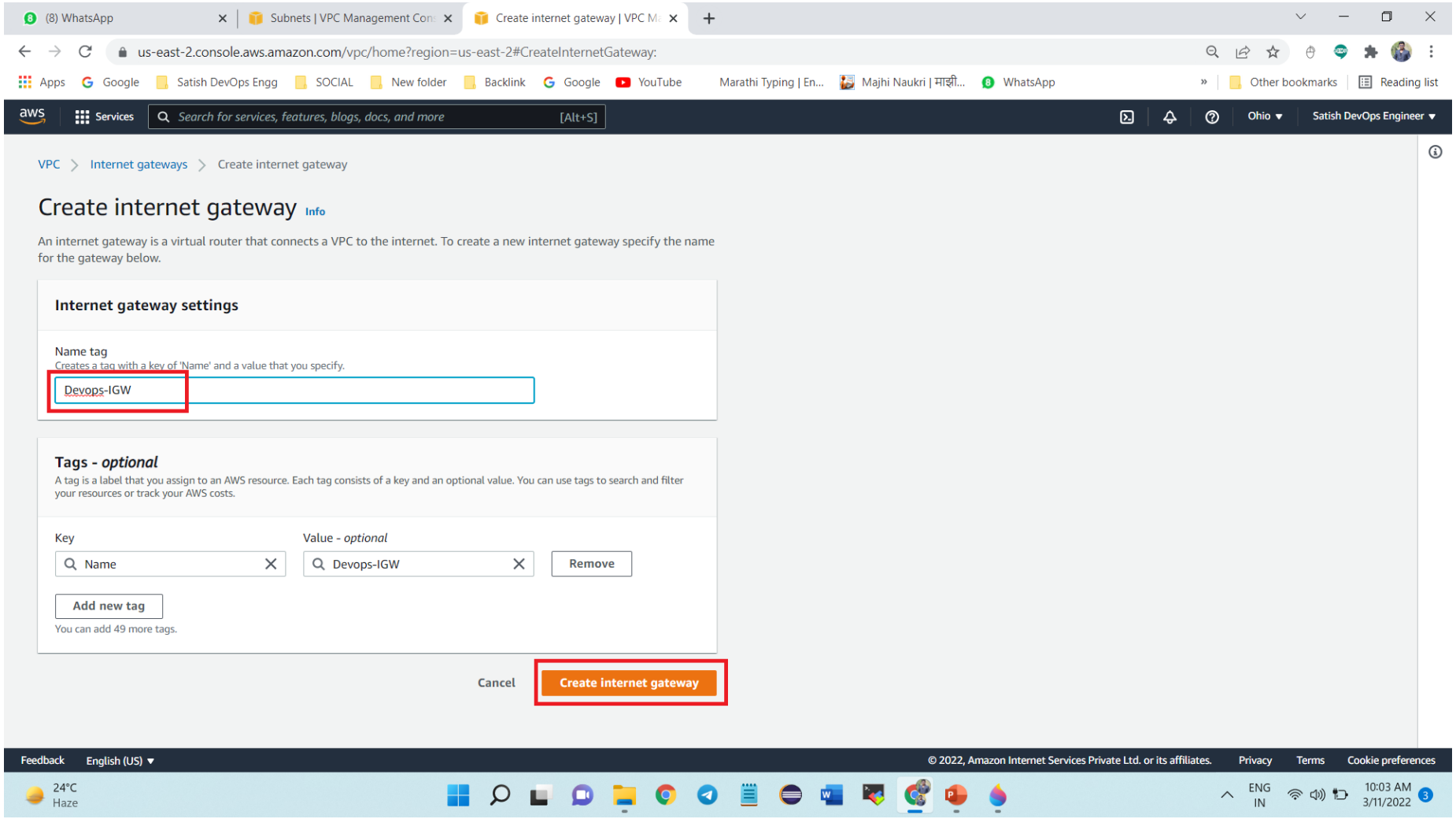


- Here our both subnets created successfully

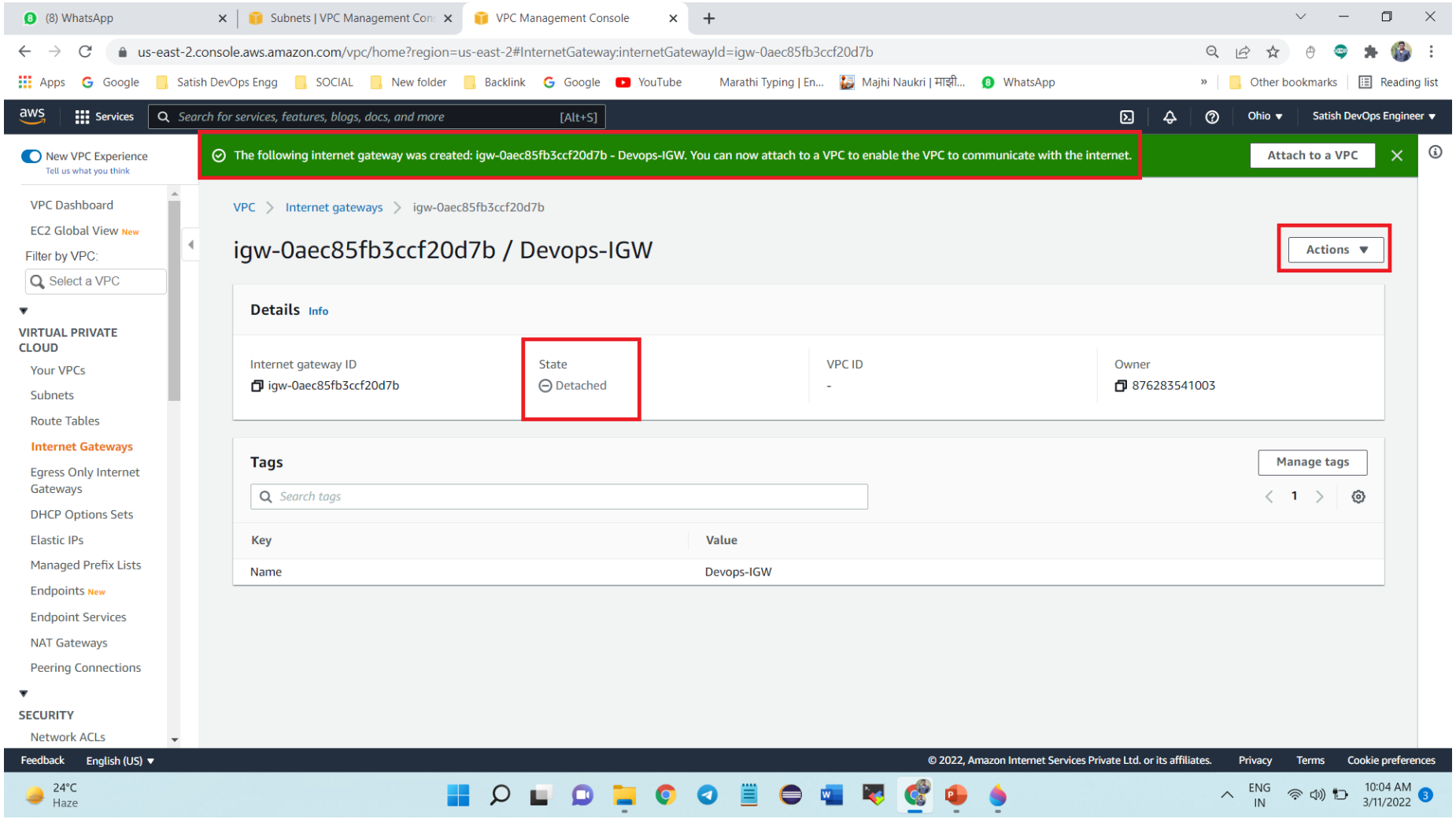
Now we have to create Internet Gateway



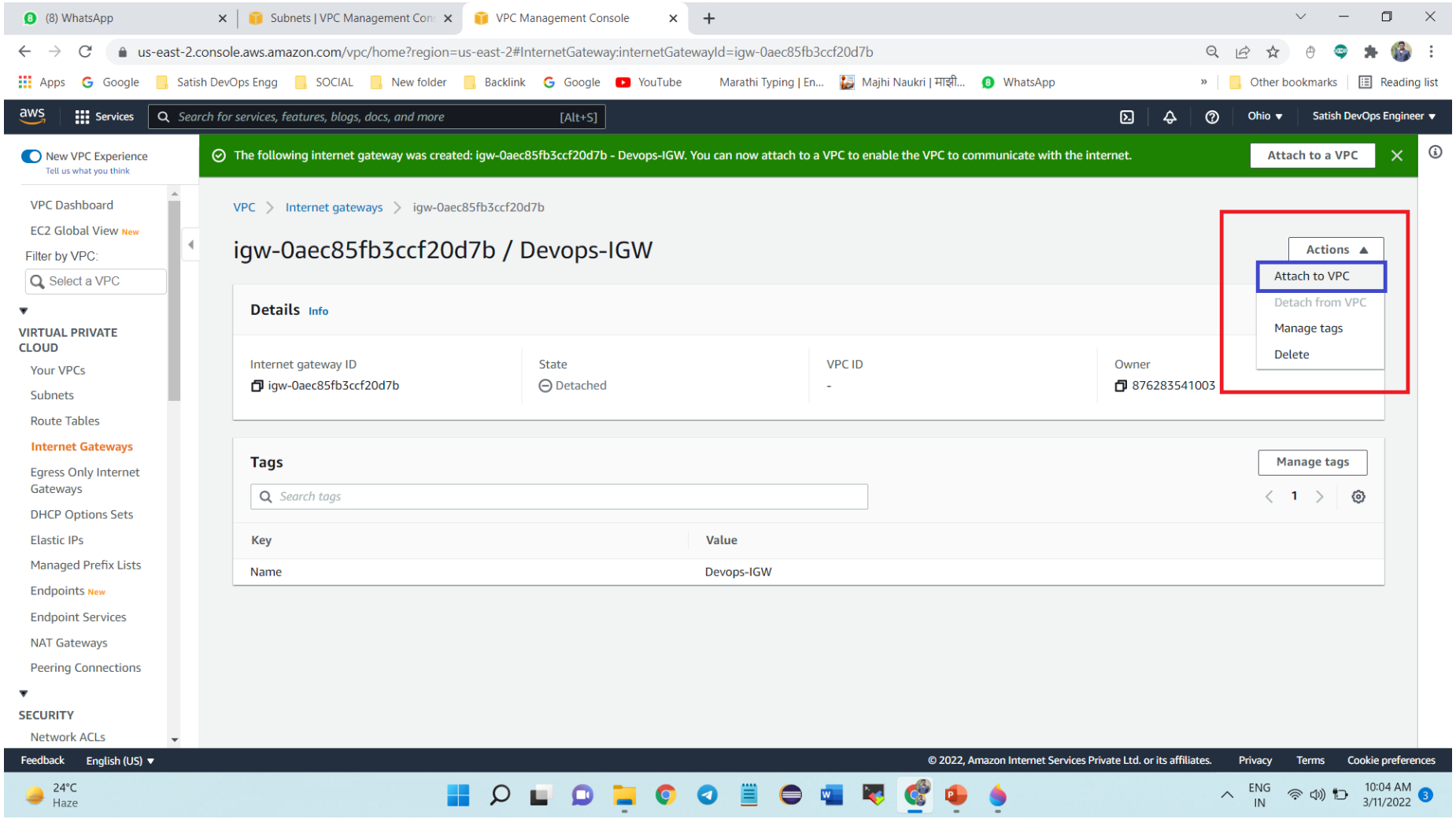
- Go to Internet Gateways
- Click on Create Internet Gateway



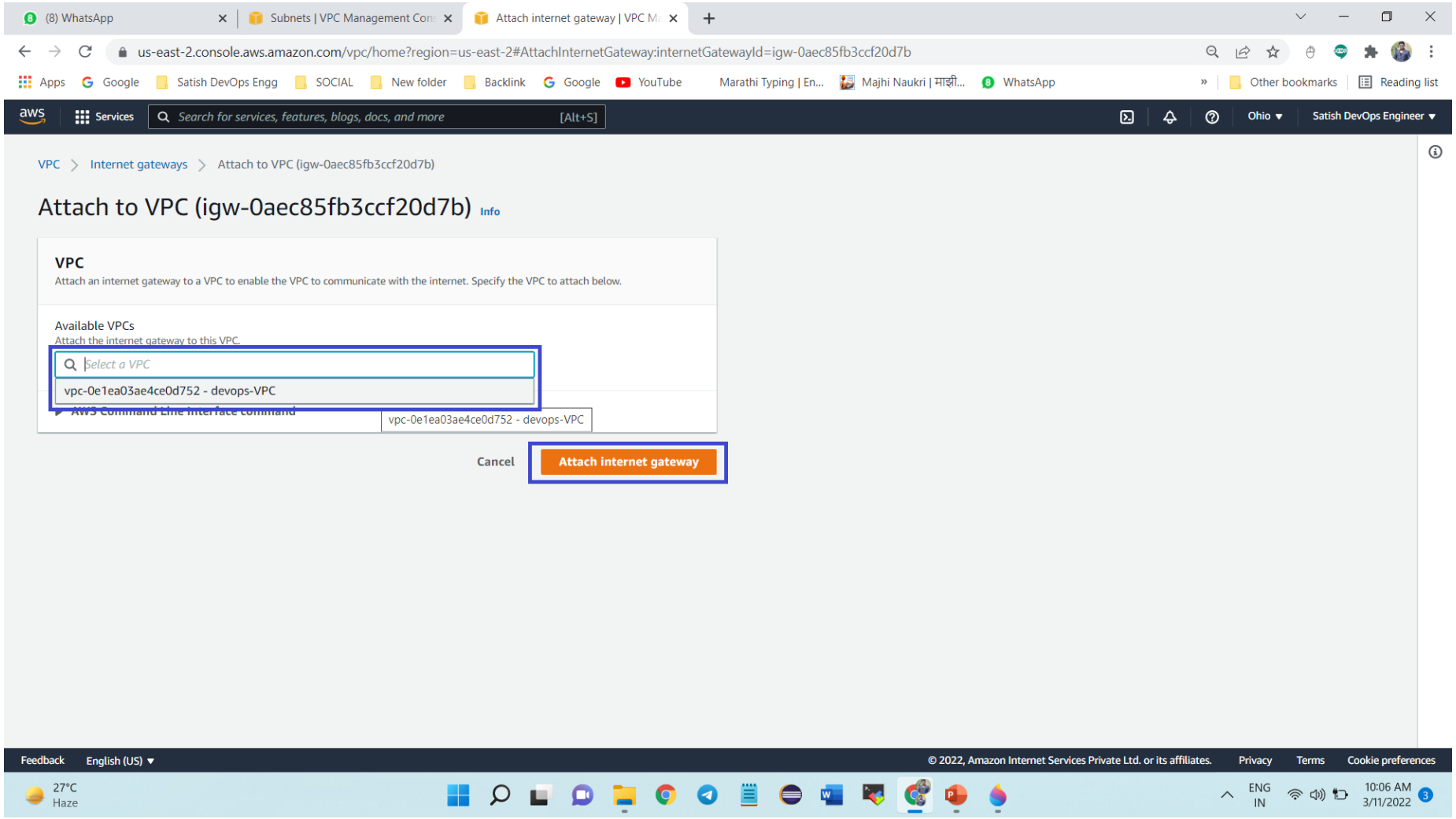
- Give Name Tag
- Click on Create Internet Gateway



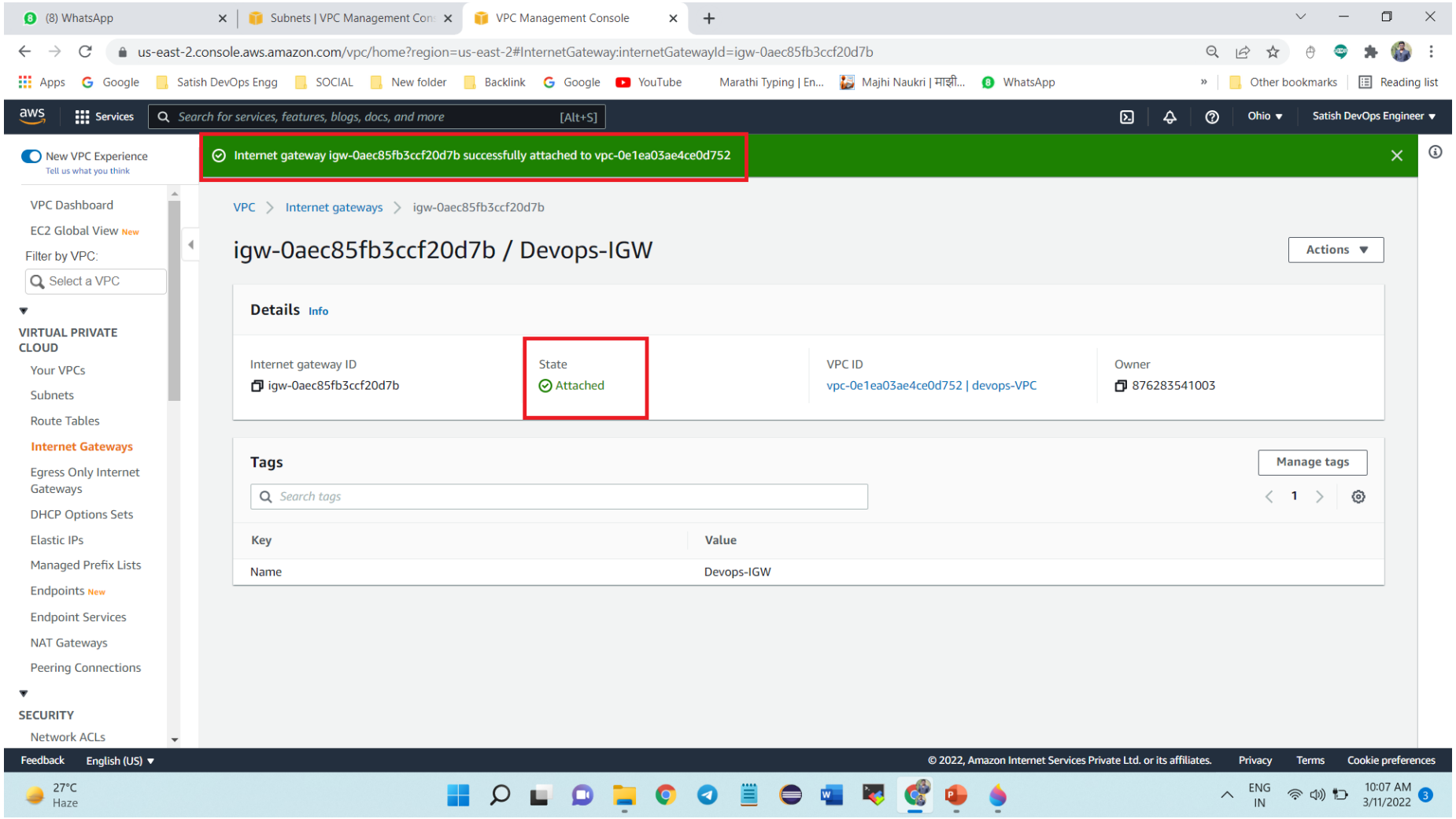
- Our Internet gateway is created successfully
- But it state is Detached
- Click on Actions



- Click on Attach to VPC

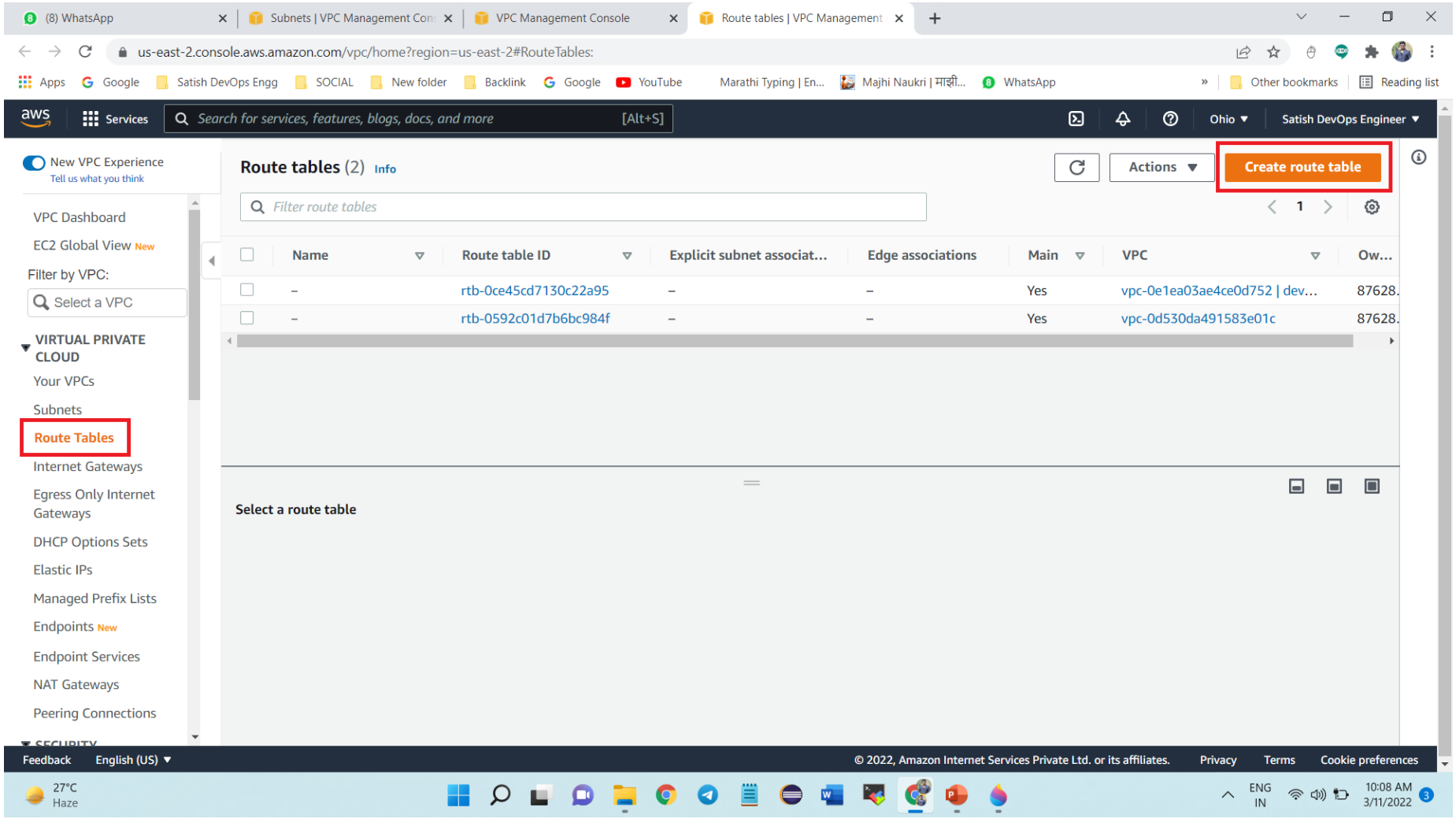


- Select our created VPC
- Click on Attach Internet gateway

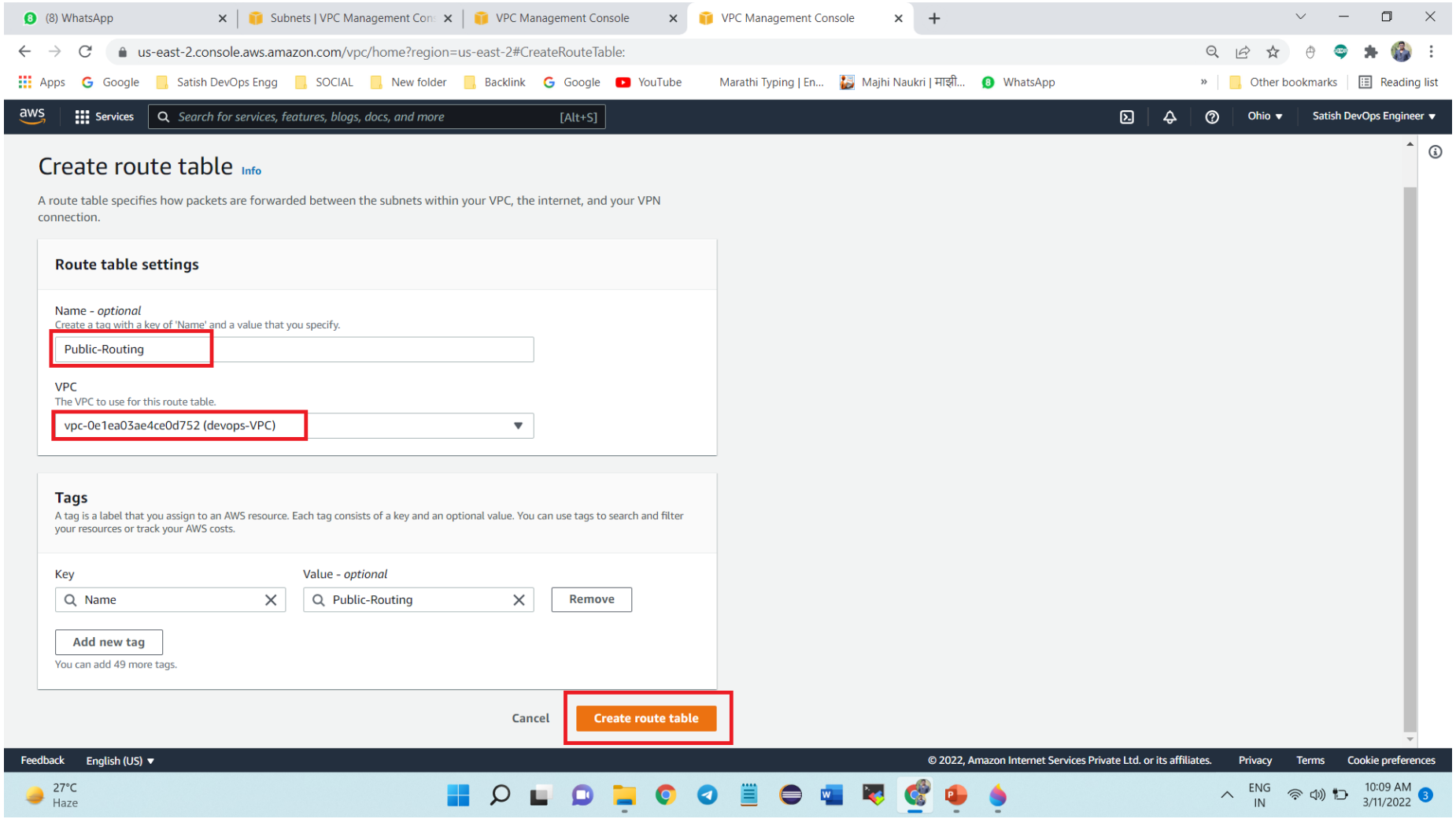


- It is now attached successfully

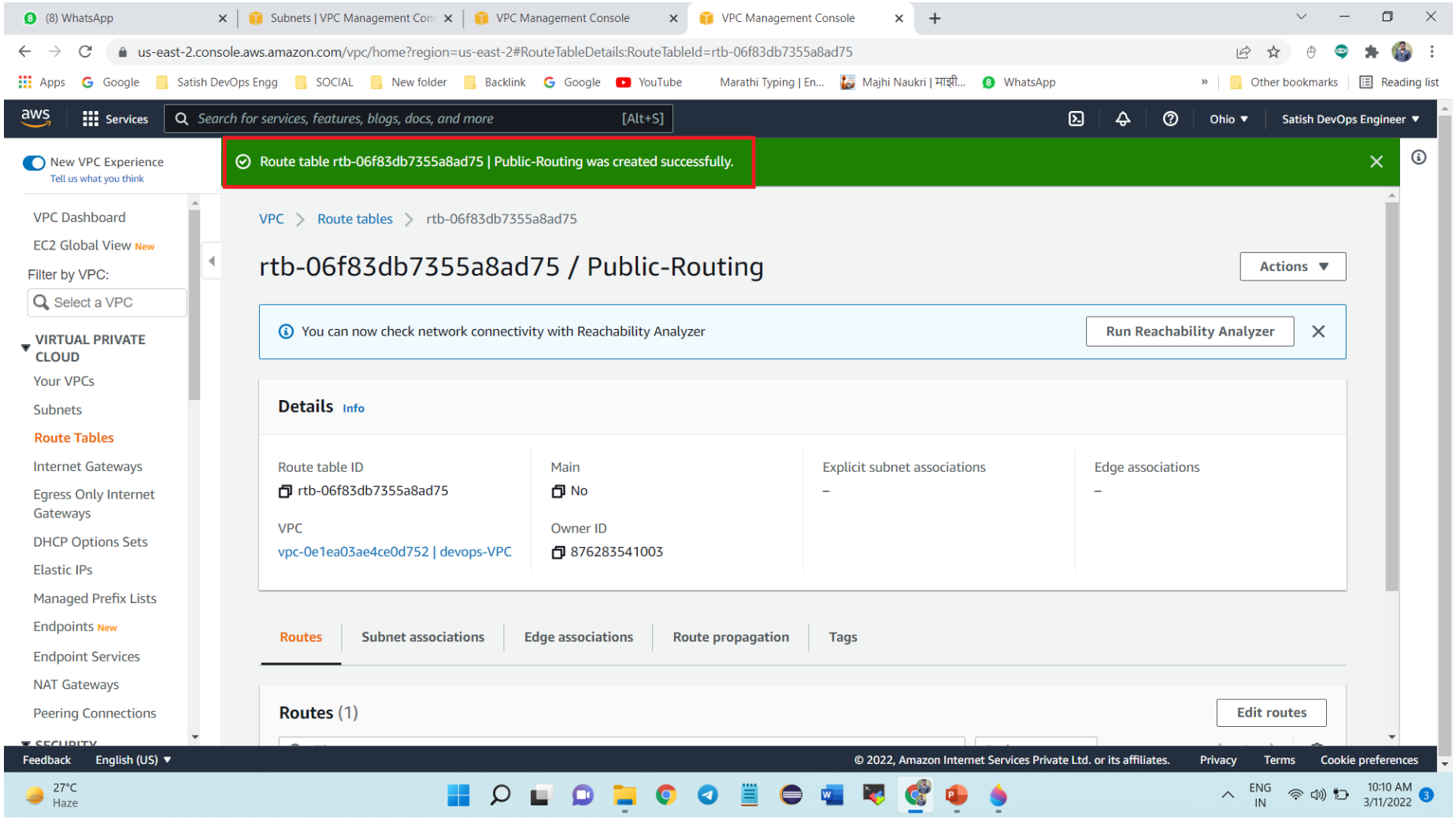
Now we have to create Route Table



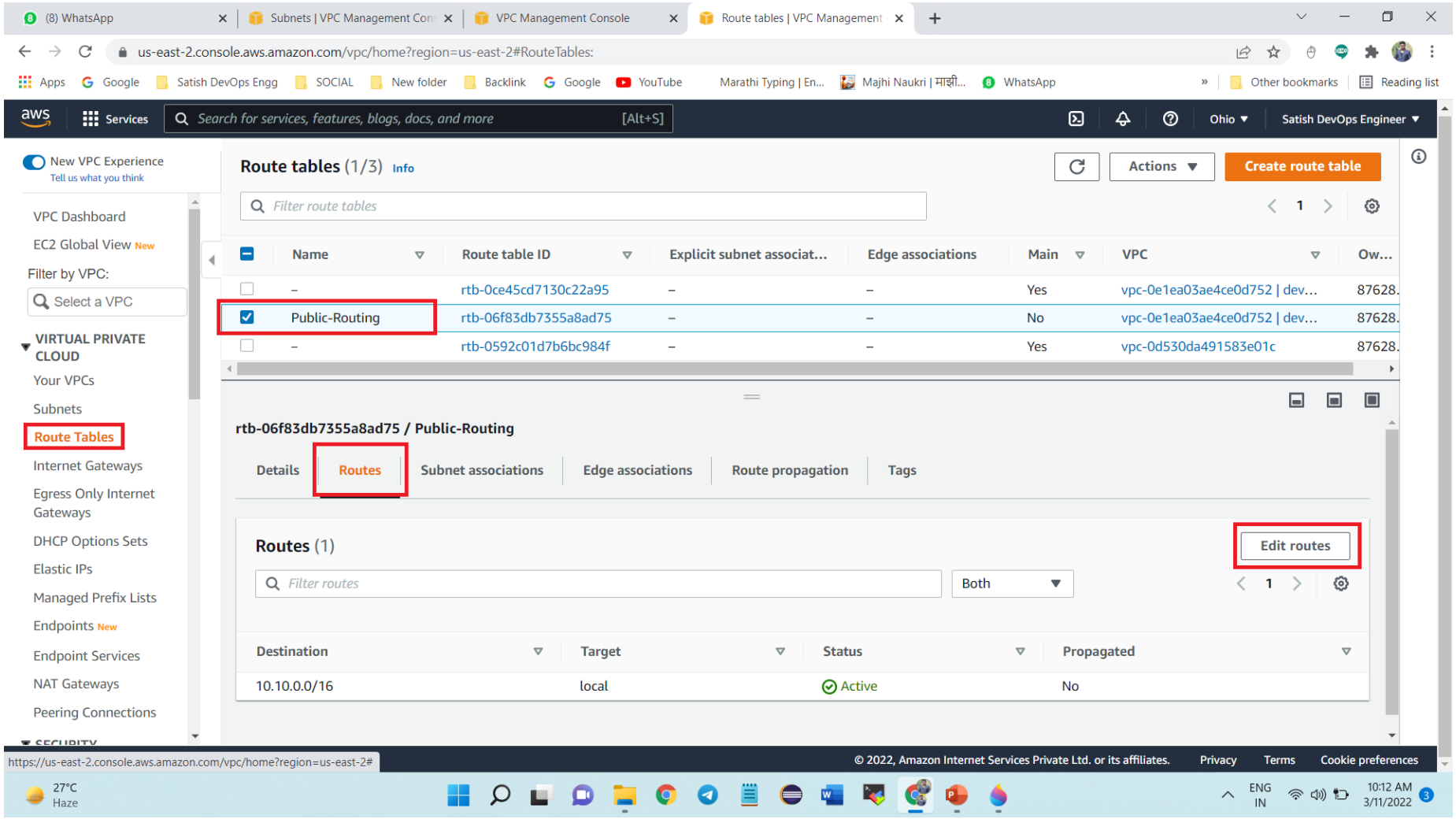
- Go to Route Table
- Click on Create Route Table



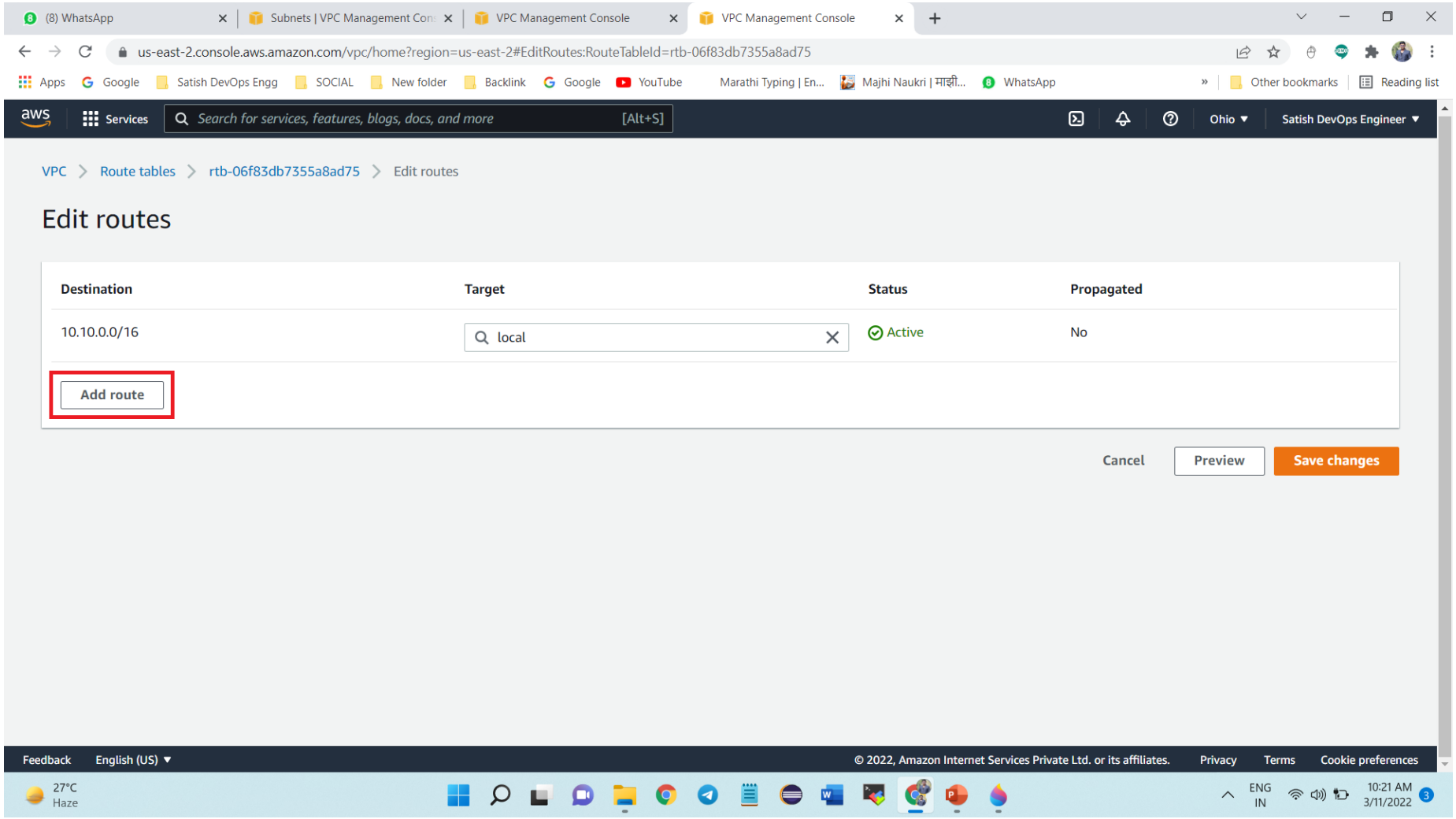
- Give Name as Public because we are using it with Internet gateway
- Select Our VPC
- Click on Create Route table



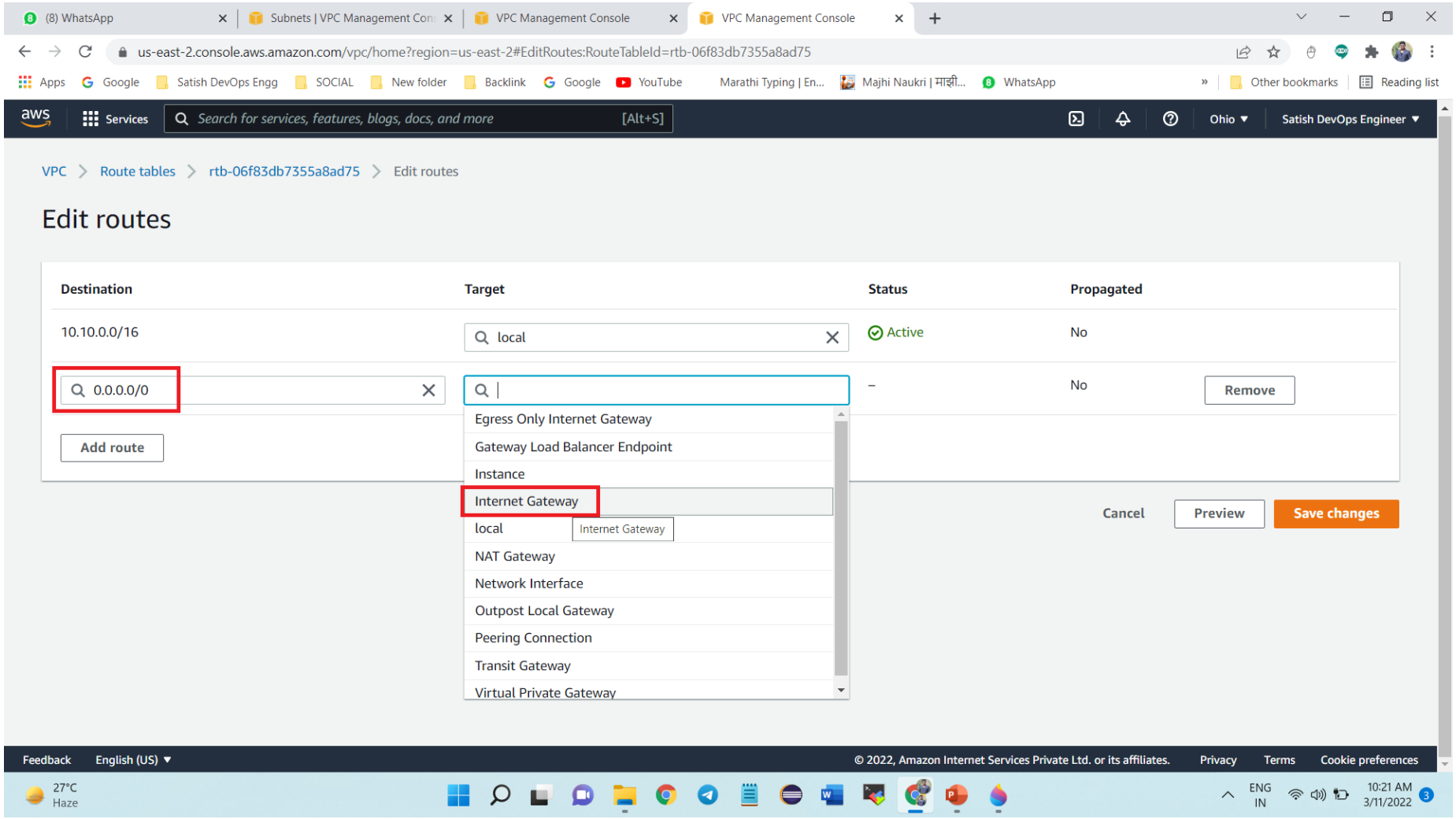
- It is created Successfully



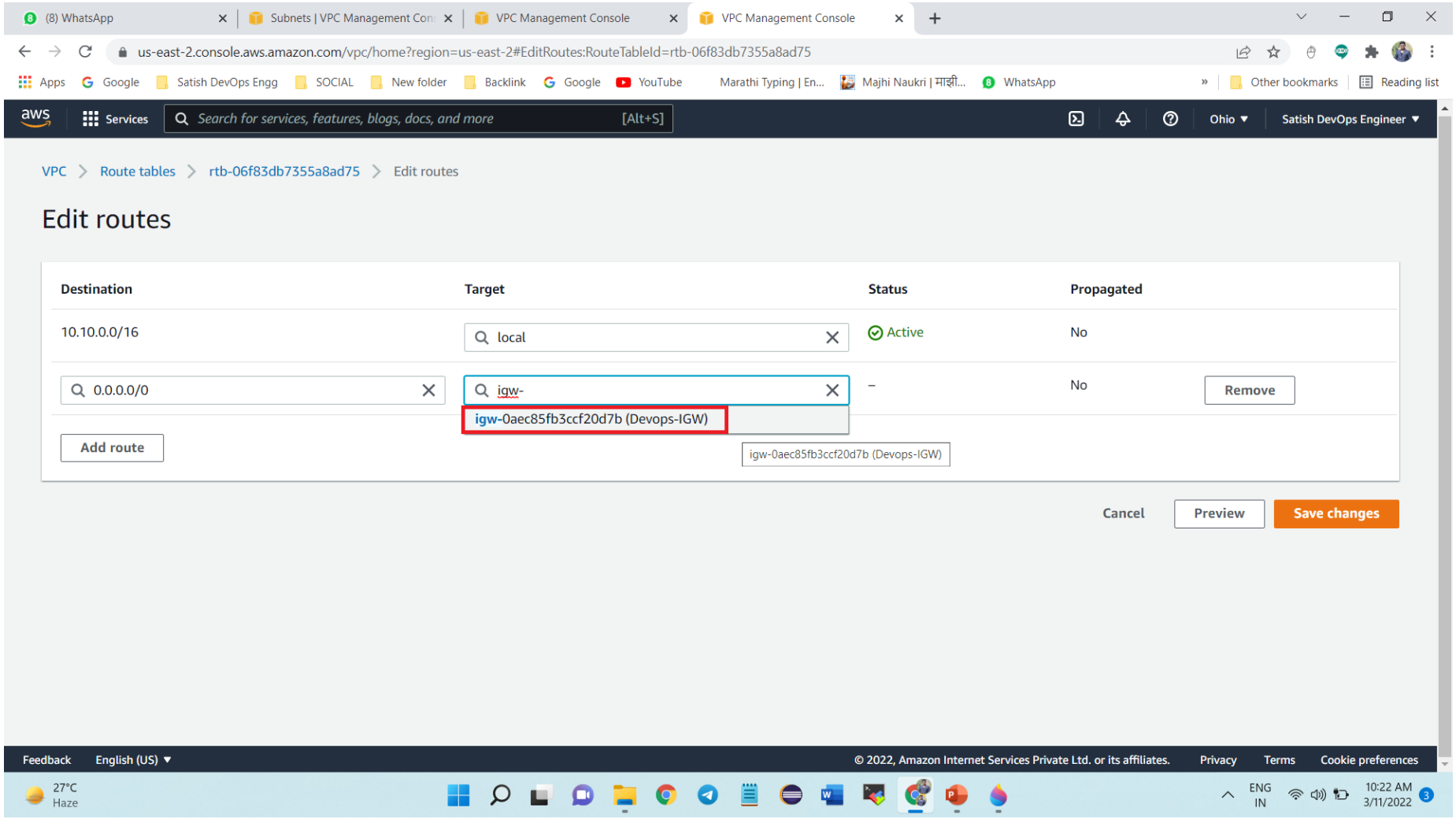
- **Go to Route Tables again**
- **Select our Public Route Table**
- **Click on Routes**
- **Click on Edit Routes**



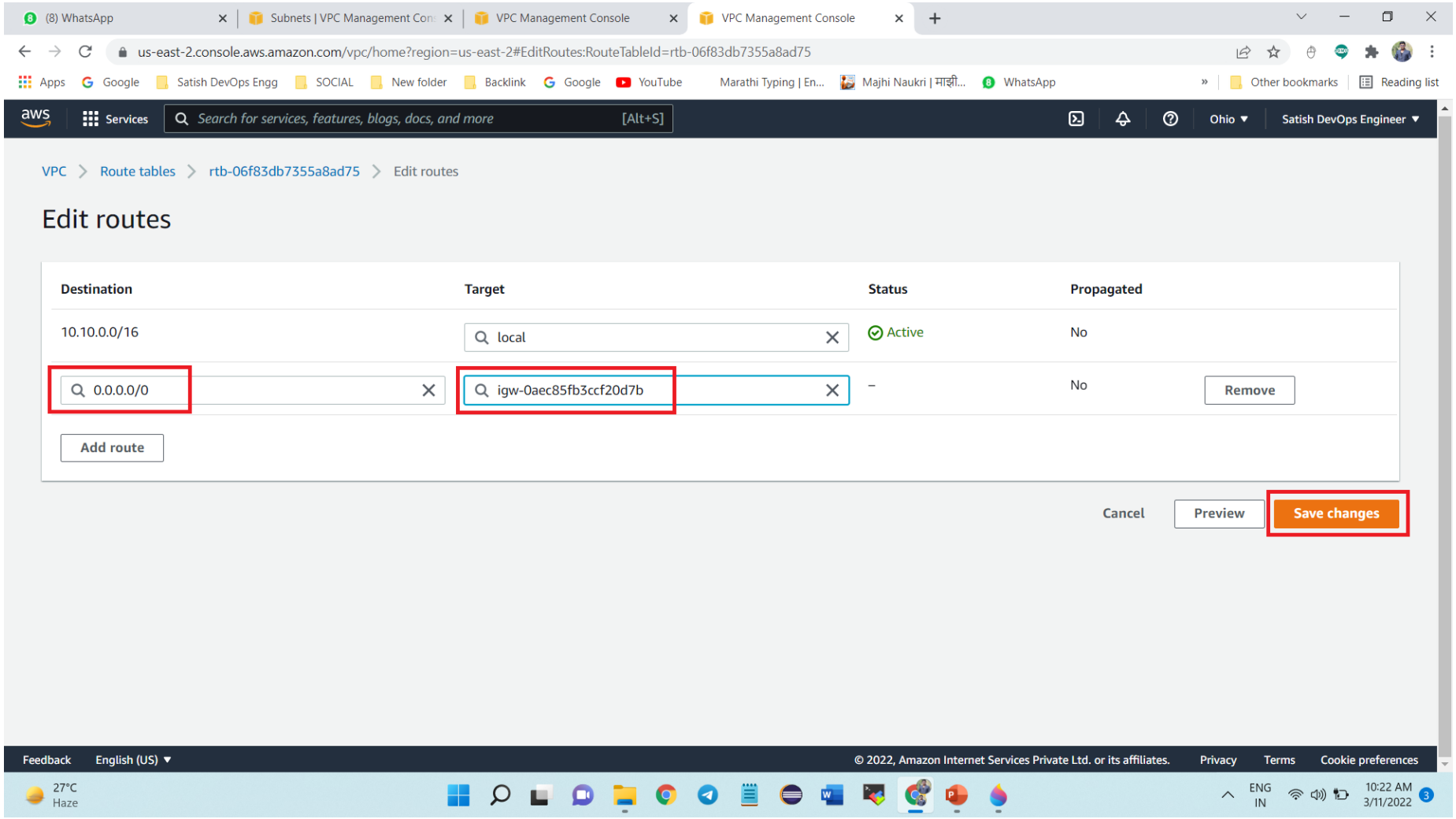
- Click on Add Route



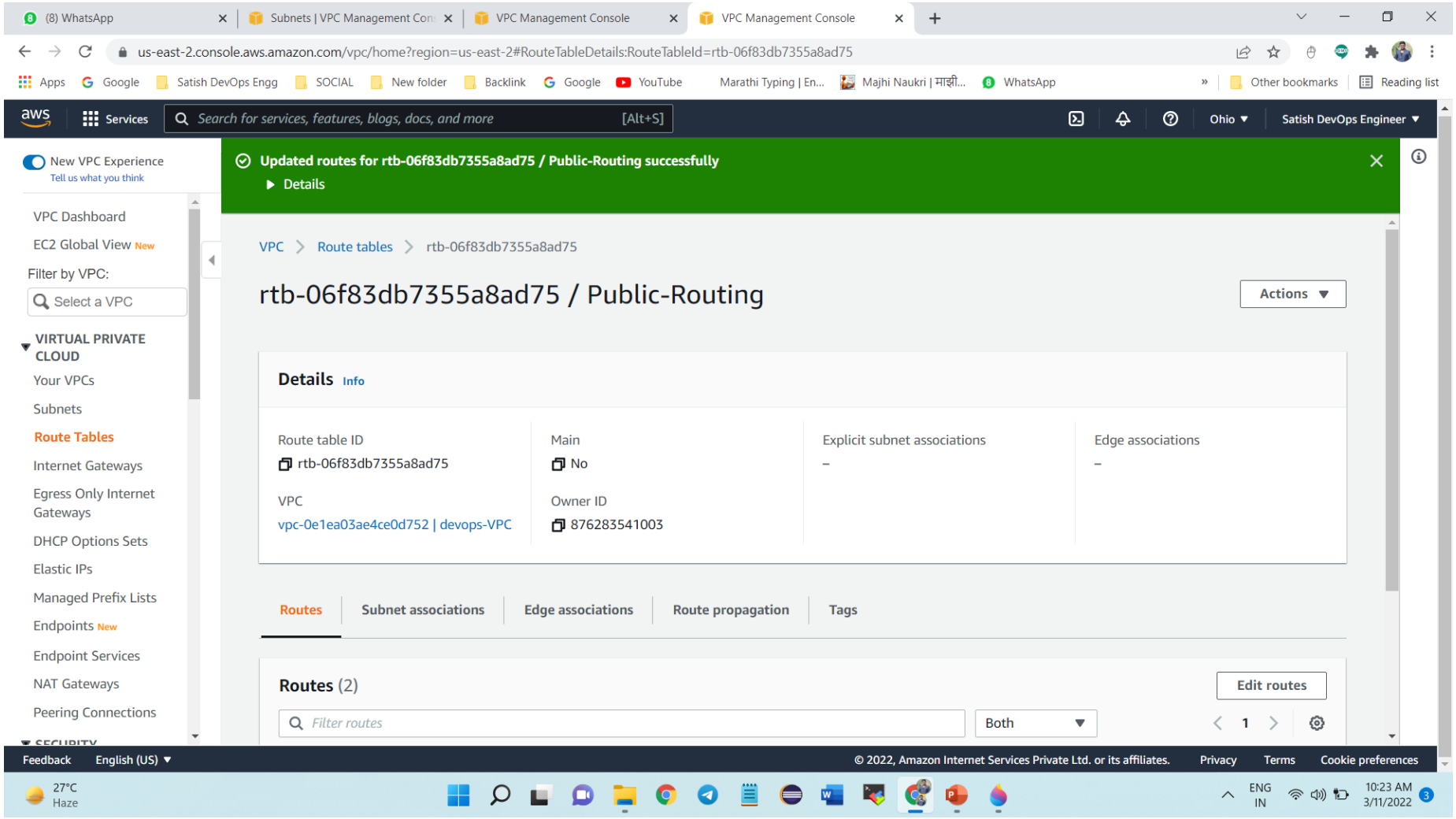
- Enter 0.0.0.0/0 in Destination
- Search internet gateway in Target



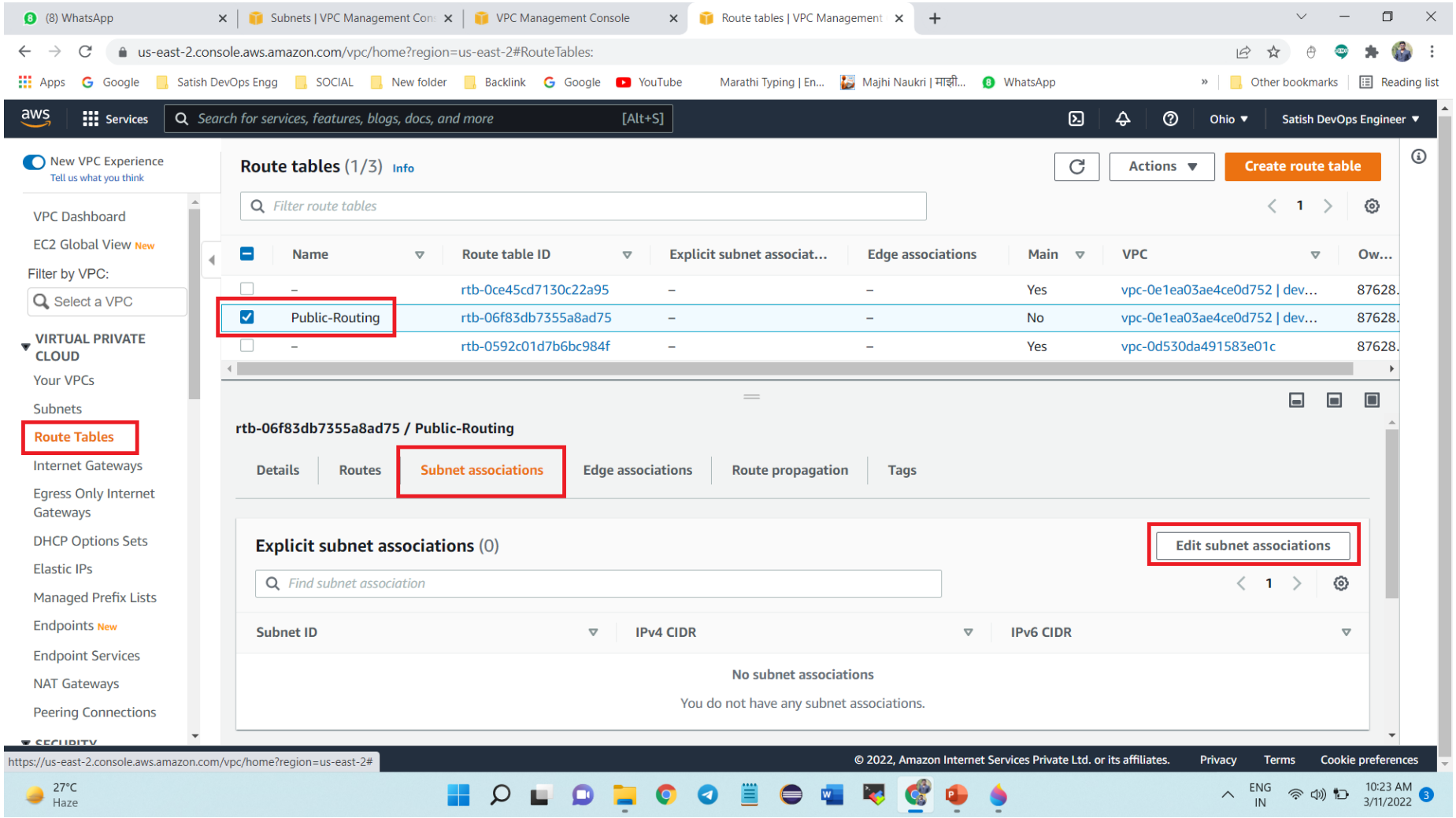
- Select your Internet gateway



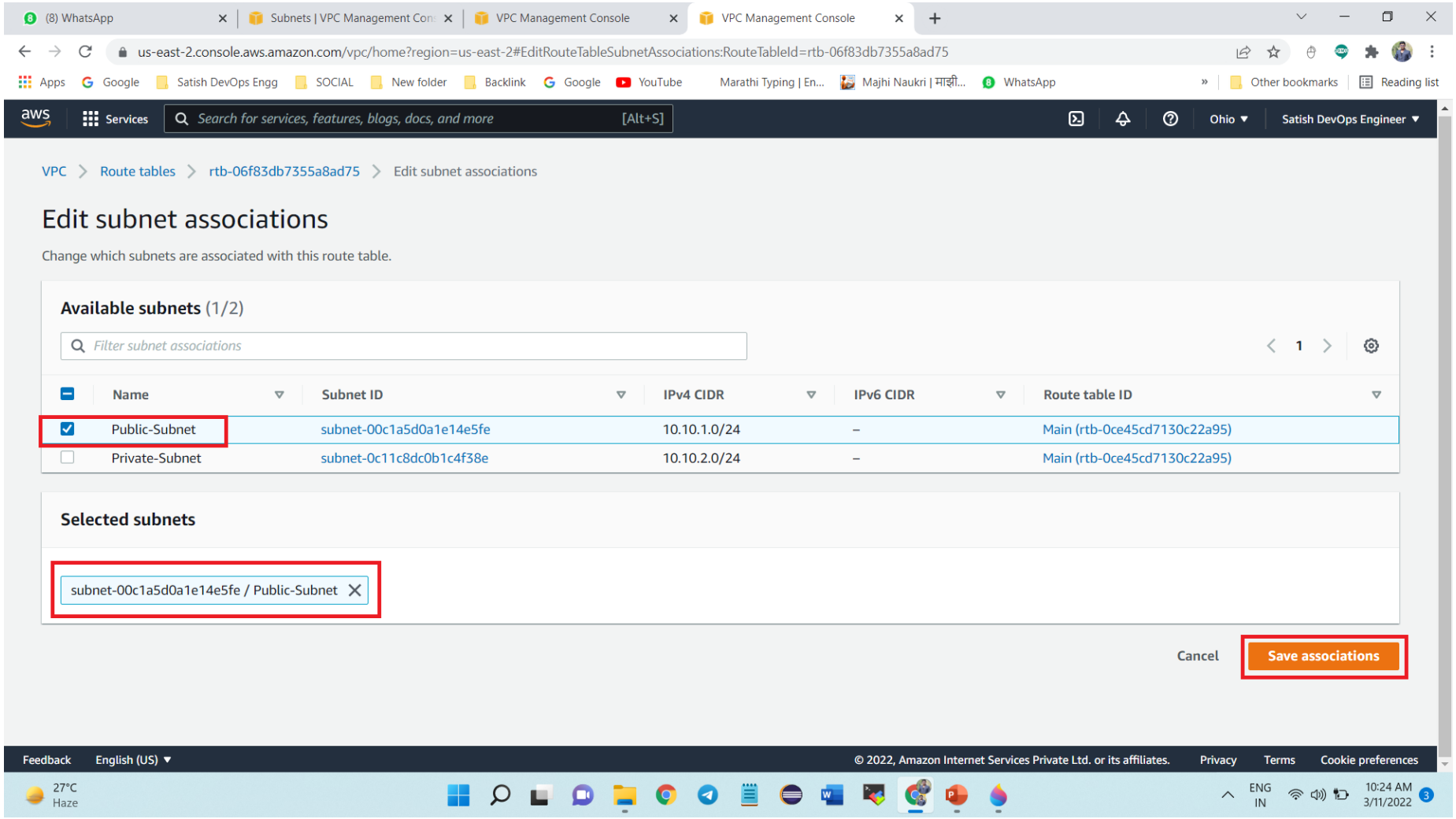
- Click on Save Changes



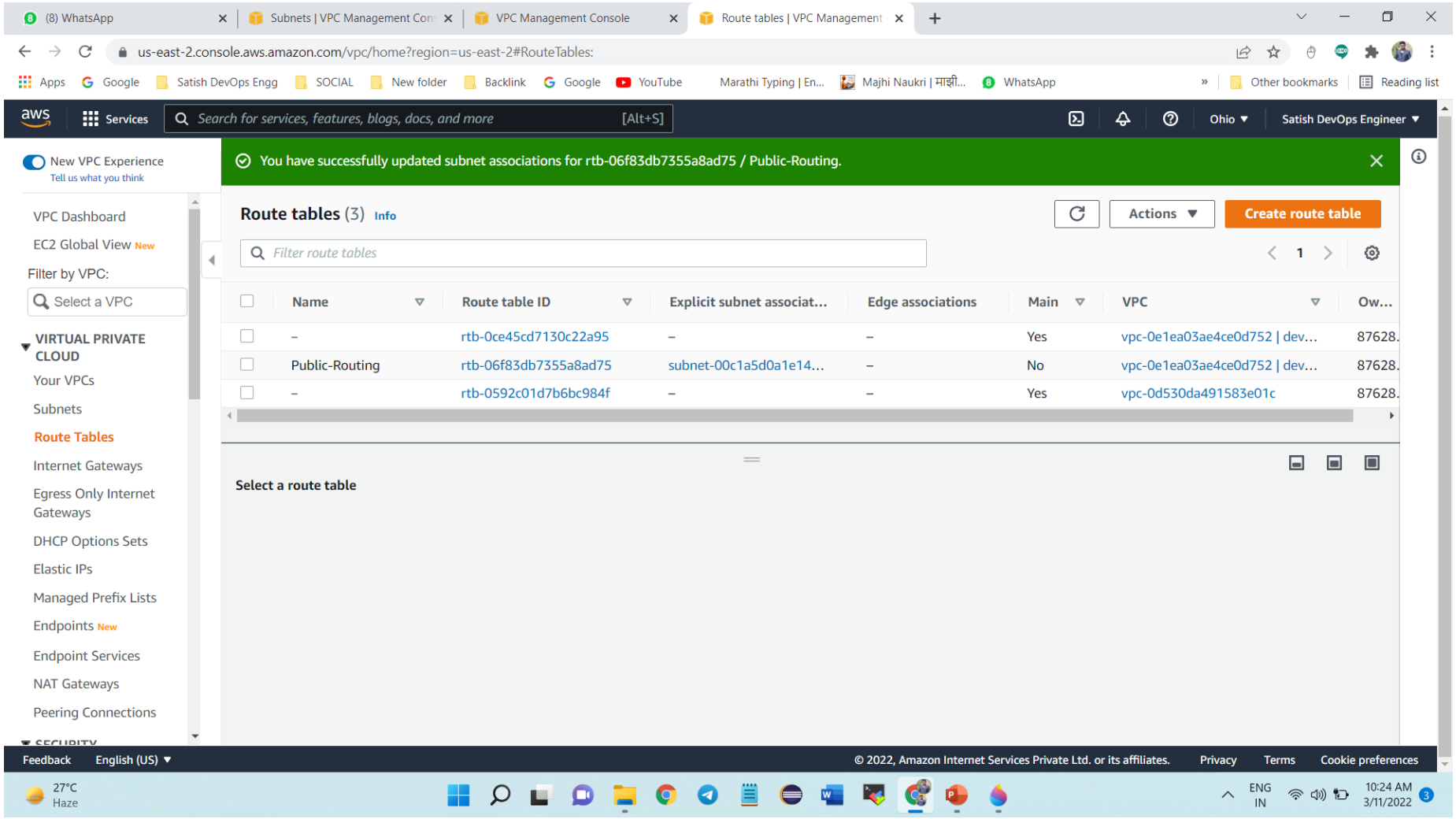
- We have added entry of IGW in Route Table successfully.



- **Go to Route Tables again**
- **Select our Public Route Table**
- **Click on Subnet Associations**
- **Click on Edit Subnet Associations**

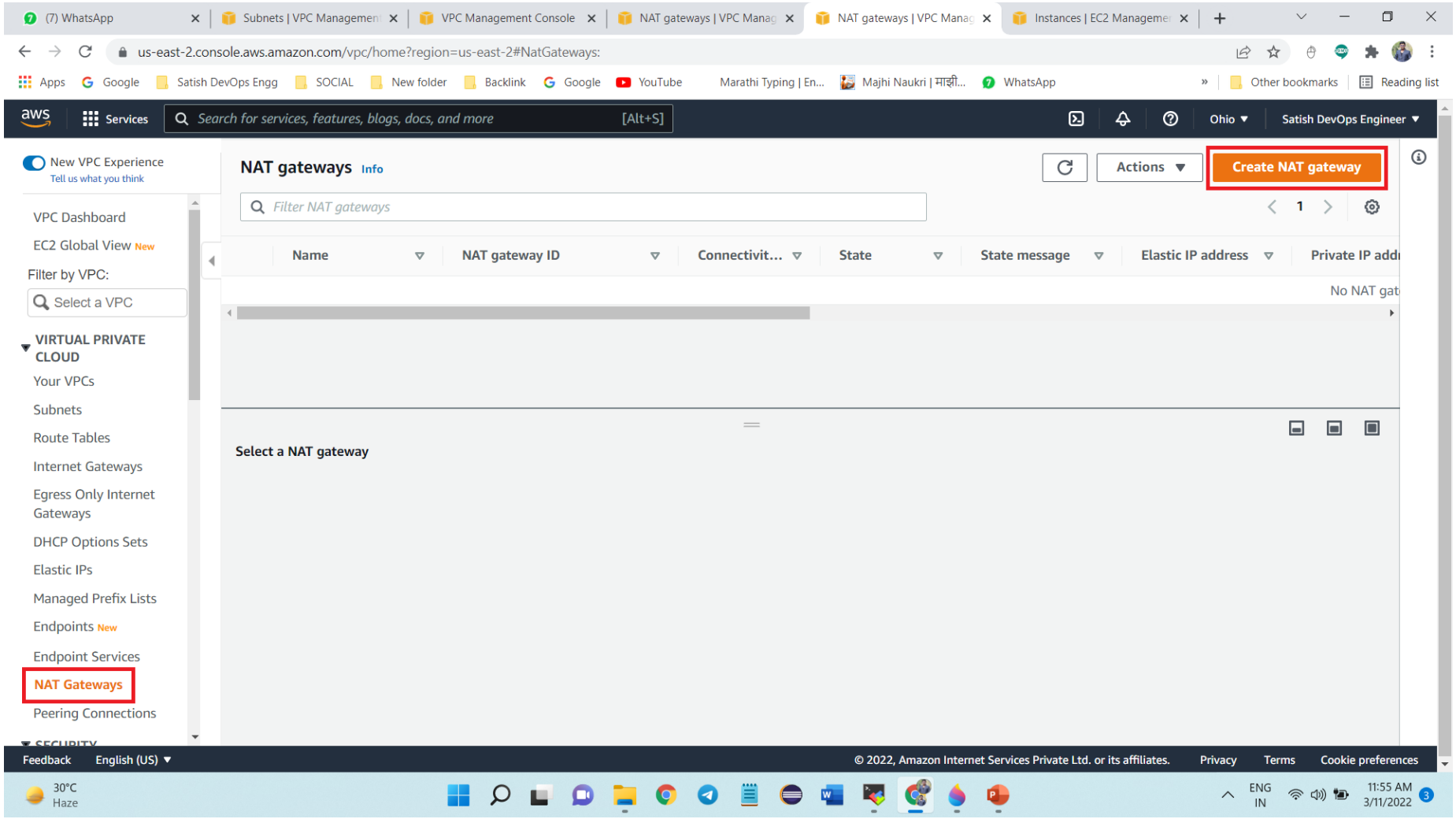


- Select your Public Subnet
- Click on Save Associations

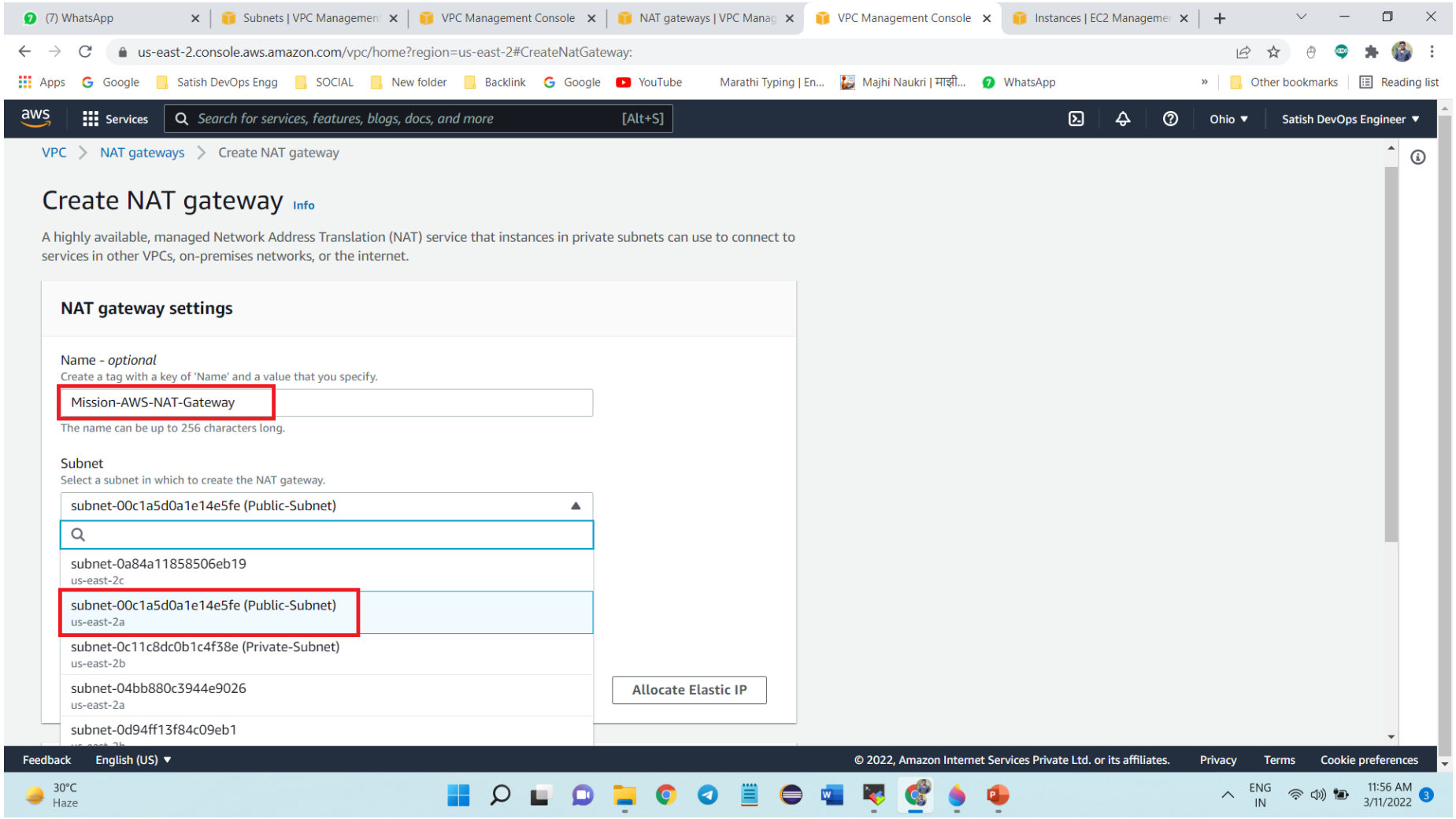


- We have associated subnet successfully.

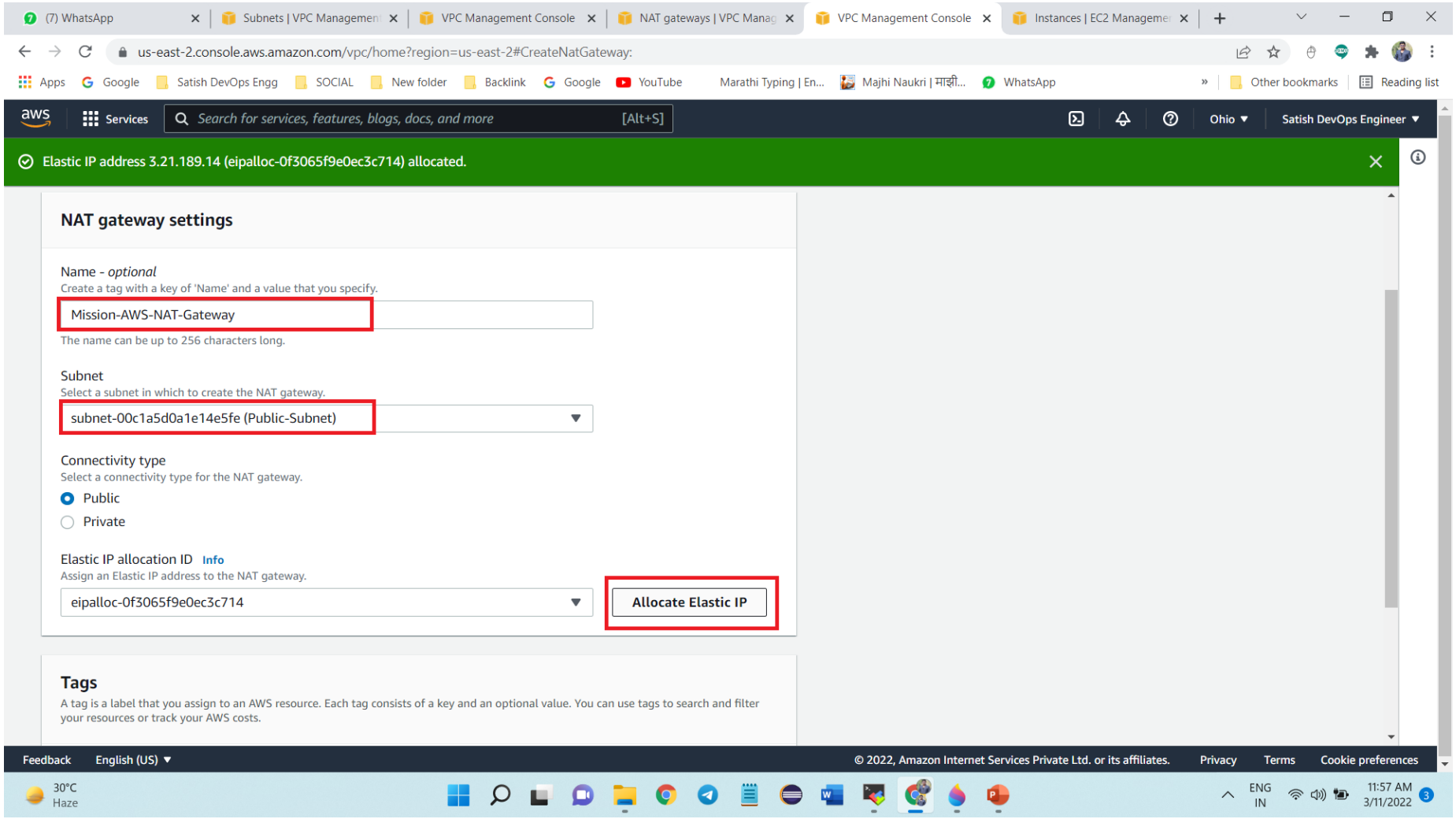
Now we have to create NAT Gateway



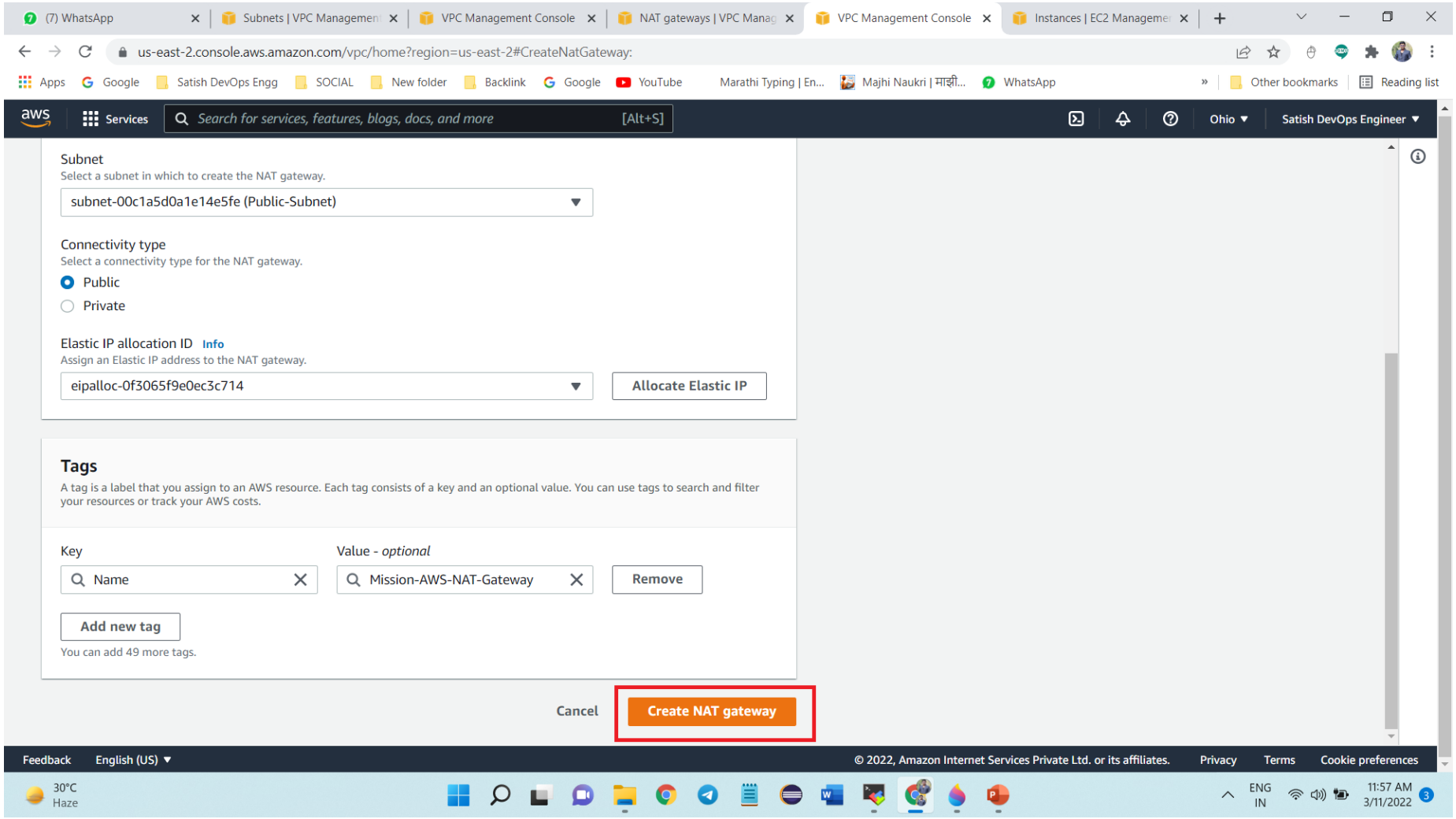
- Go to NAT Gateways
- Click on Create NAT Gateway



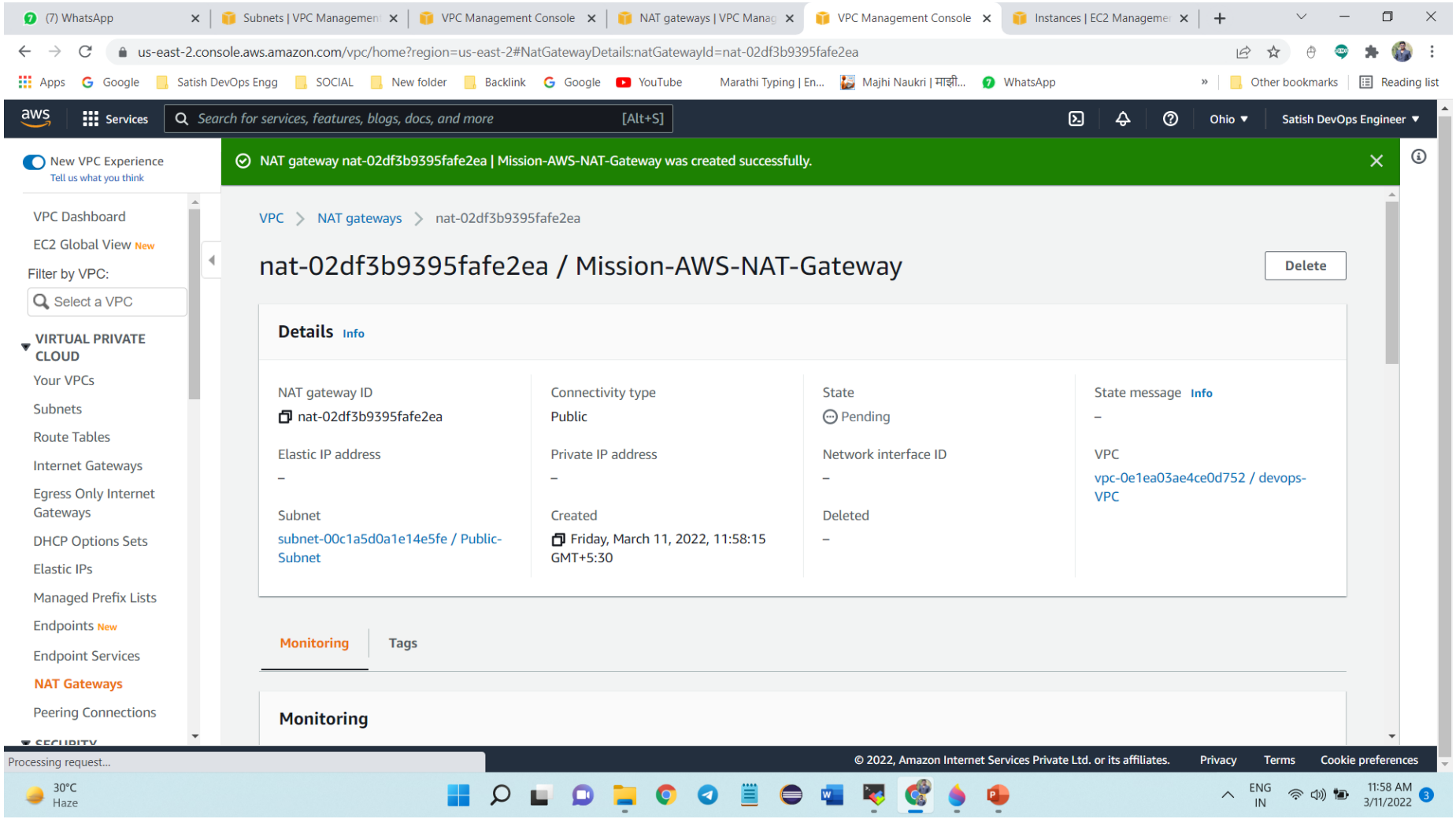
- Give Name
- Select Your Public Subnet



- Click on Allocate Elastic IP

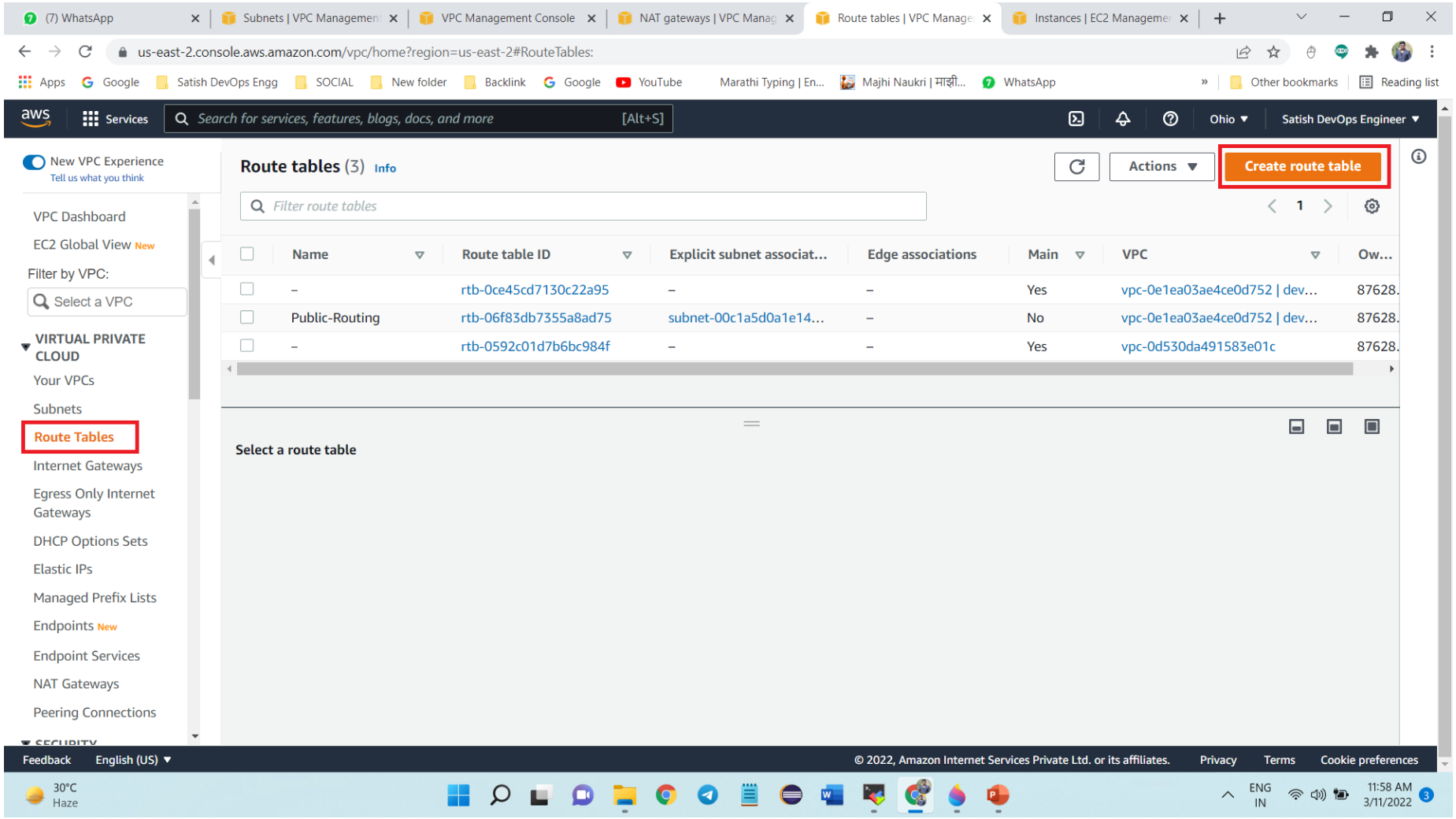


- Click on create NAT Gateway

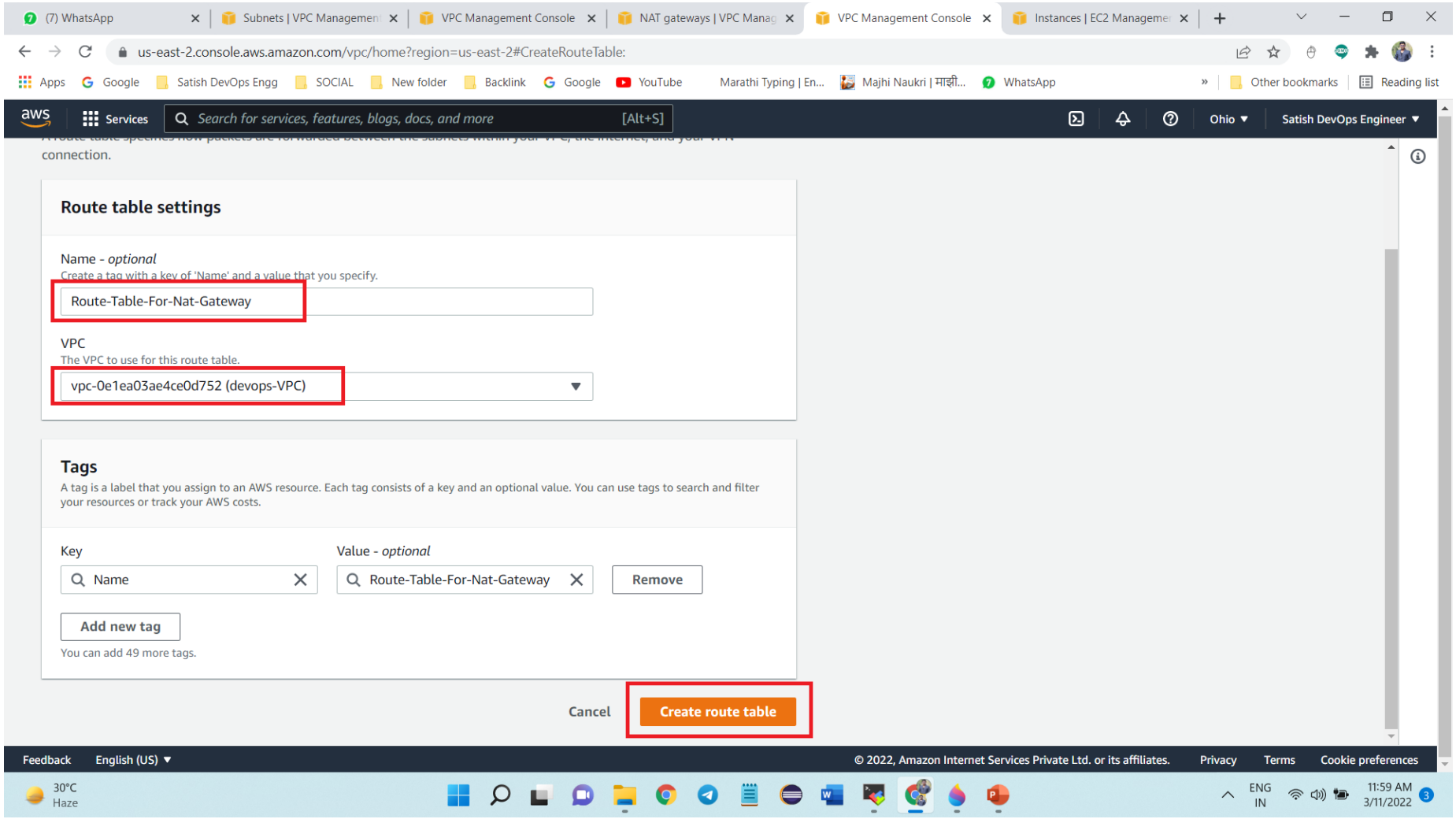


- We have created NAT Gateway Successfully.

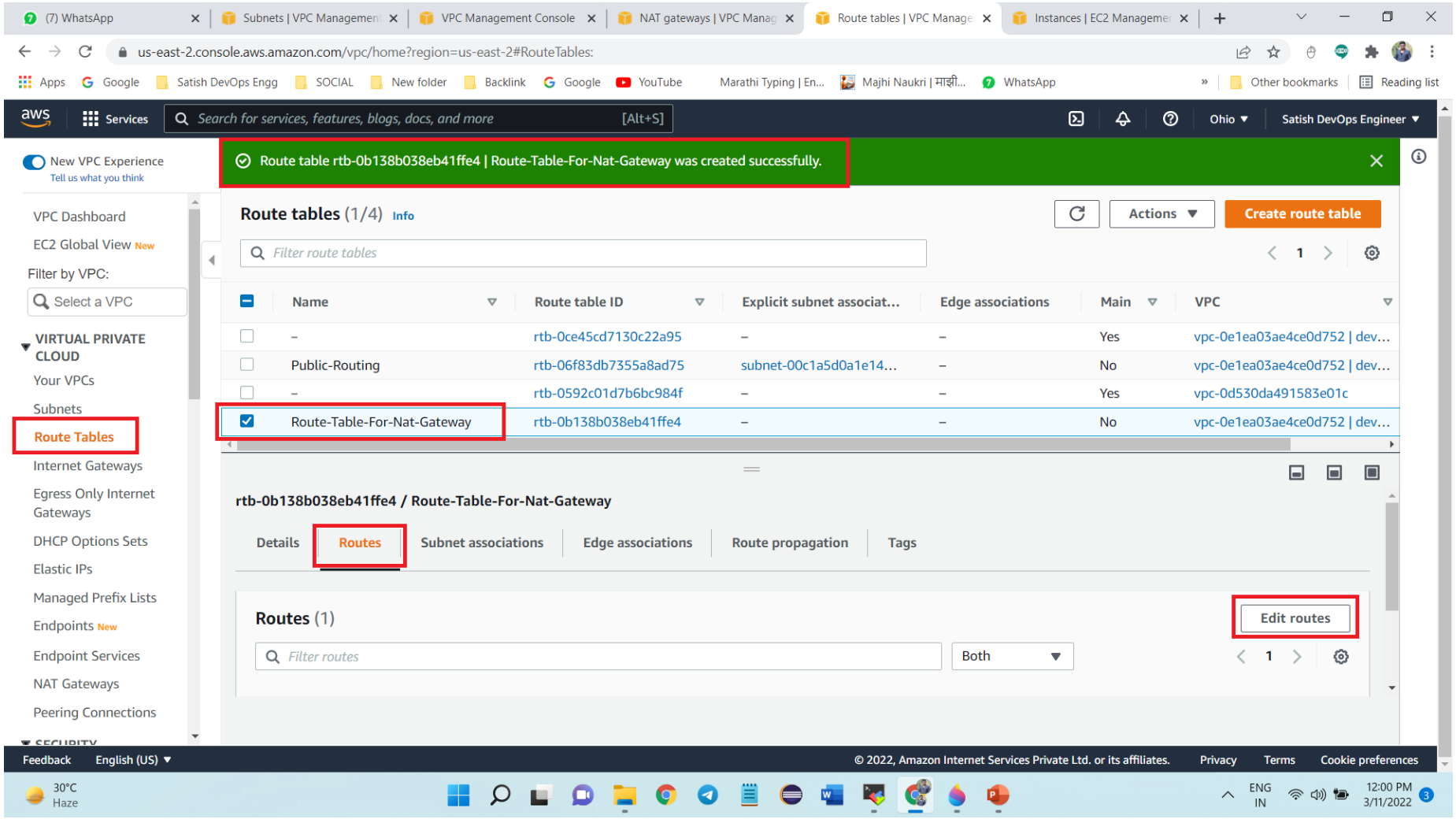
Now we have to create Route Table for NAT Gateway



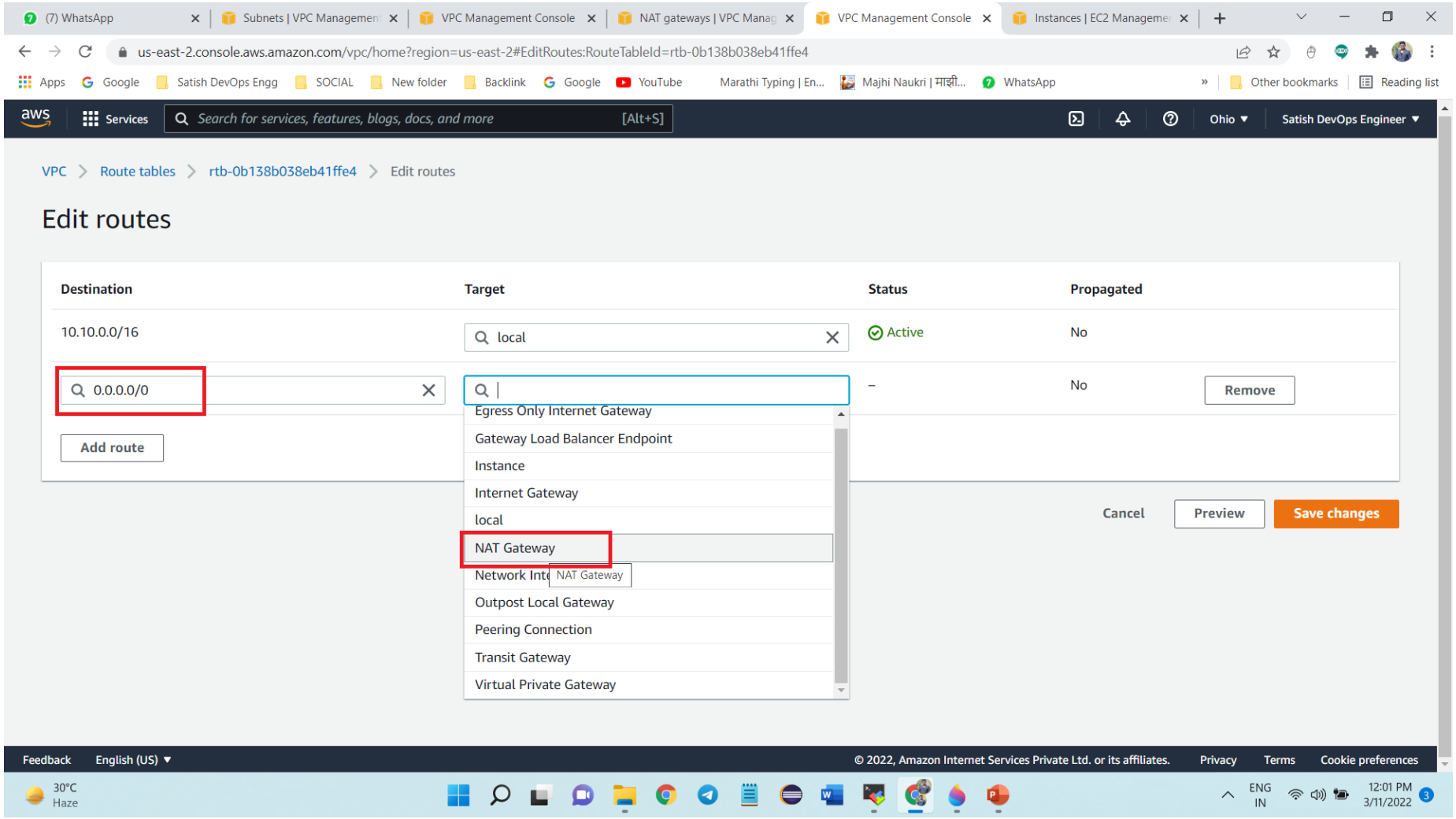
- Go to Route Table
- Click on Create Route Table



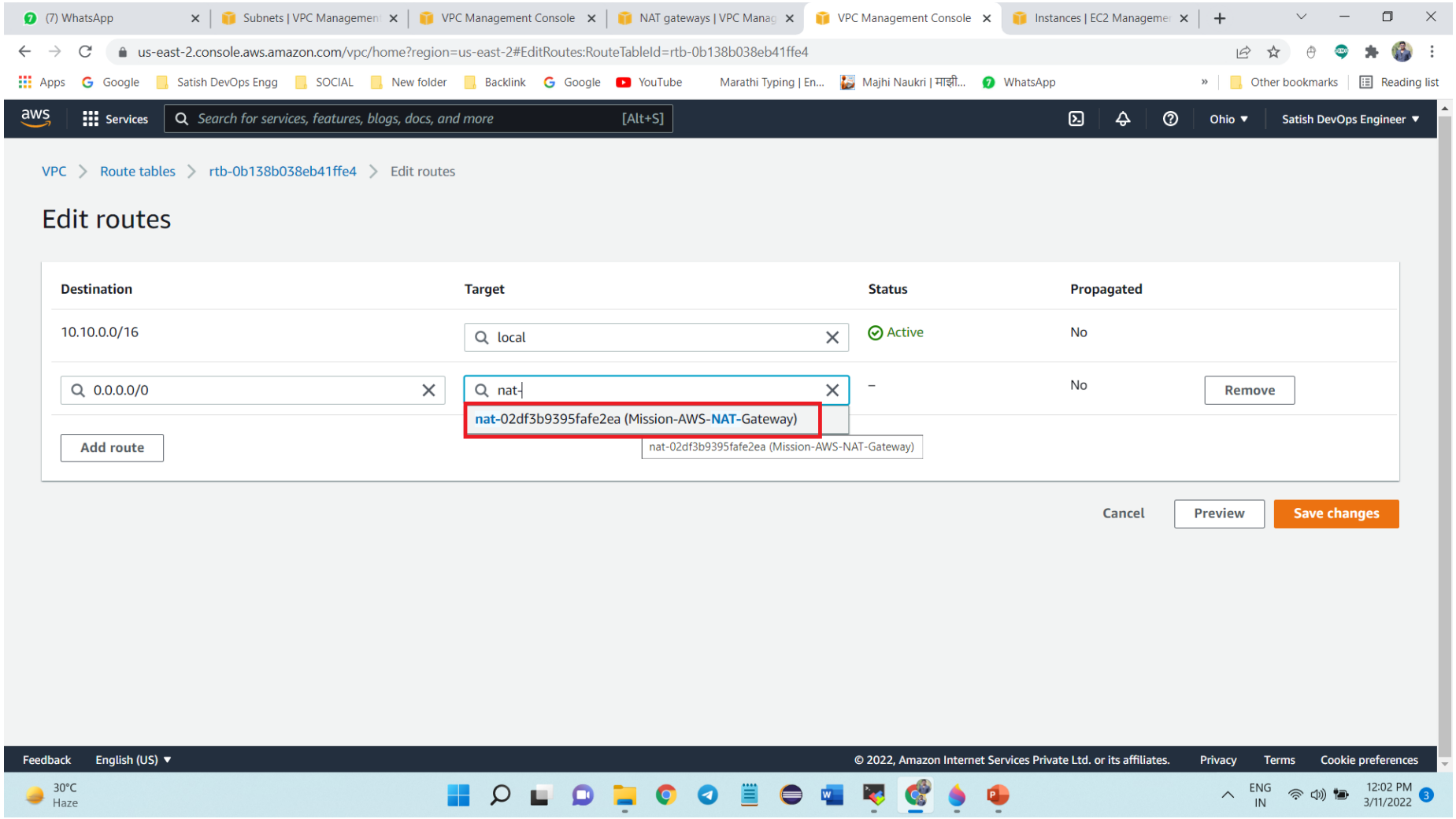
- **Give name properly because we are creating this route table for NAT Gateway**
- **Select our created VPC**
- **Click on Create Route Table**



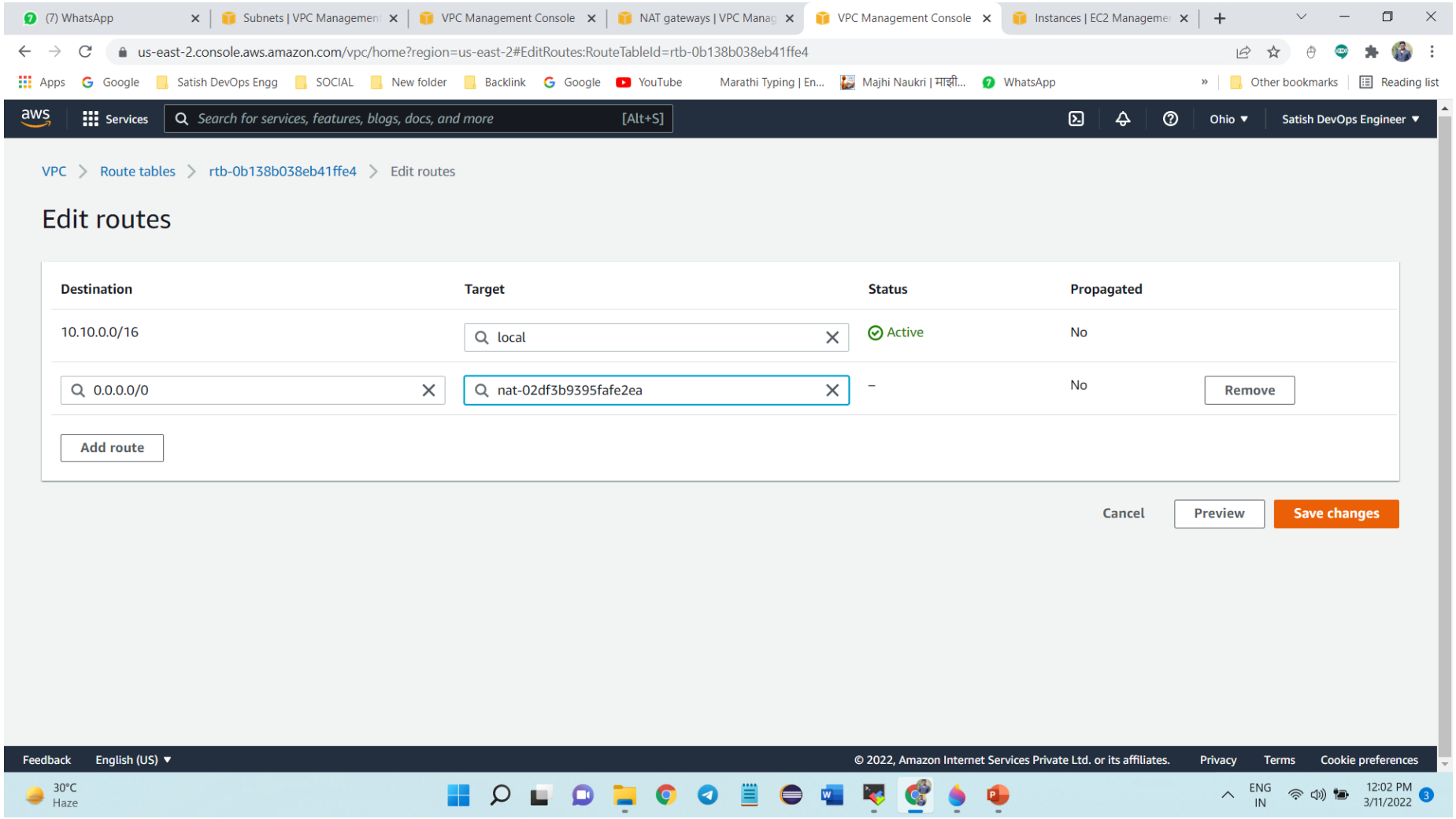
- **Go to Route Tables again**
- **Select our NAT Gateway Route Table**
- **Click on Routes**
- **Click on Edit Routes**



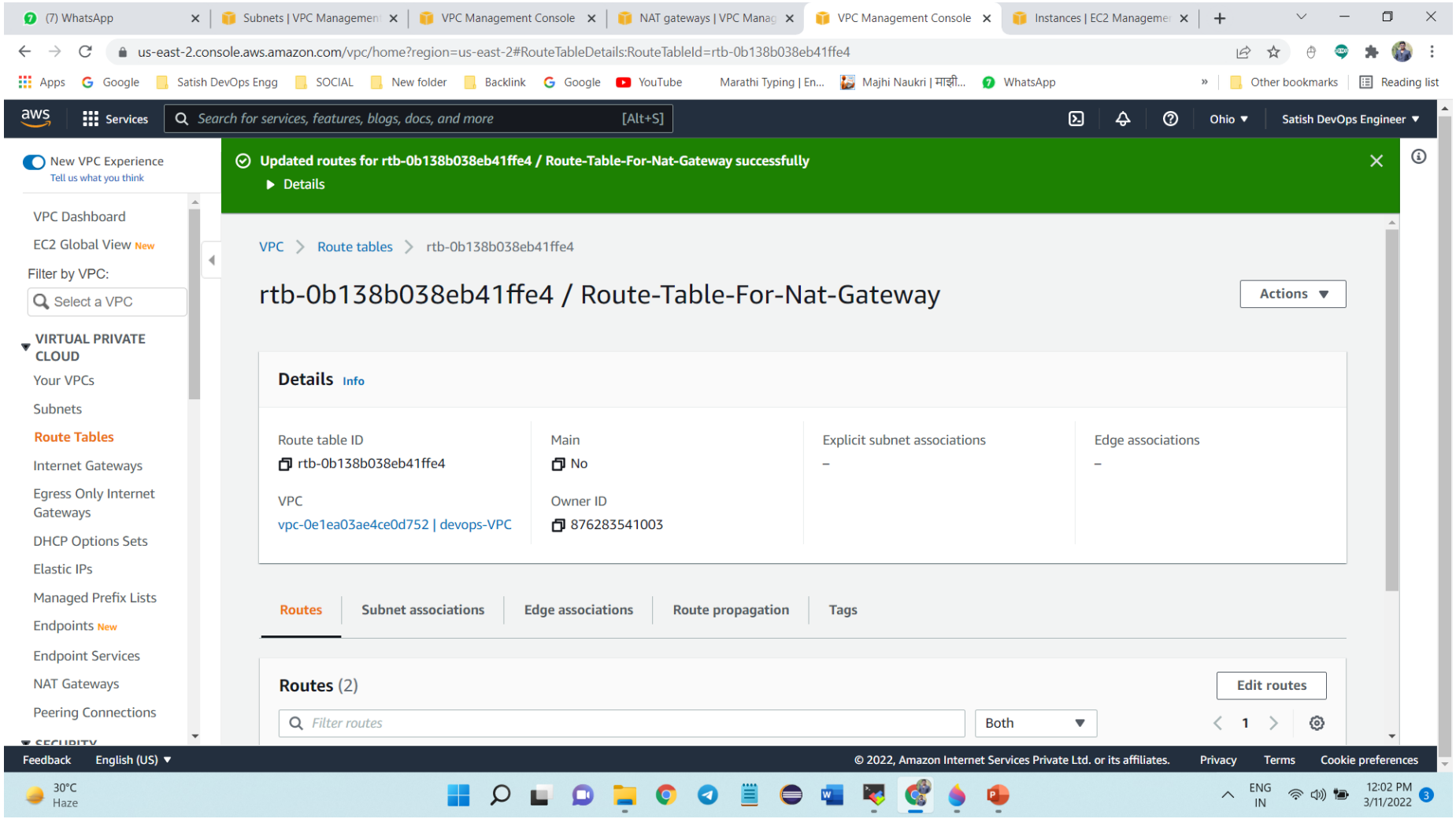
- Enter 0.0.0.0/0 in Destination
- Search NAT Gateway in Target



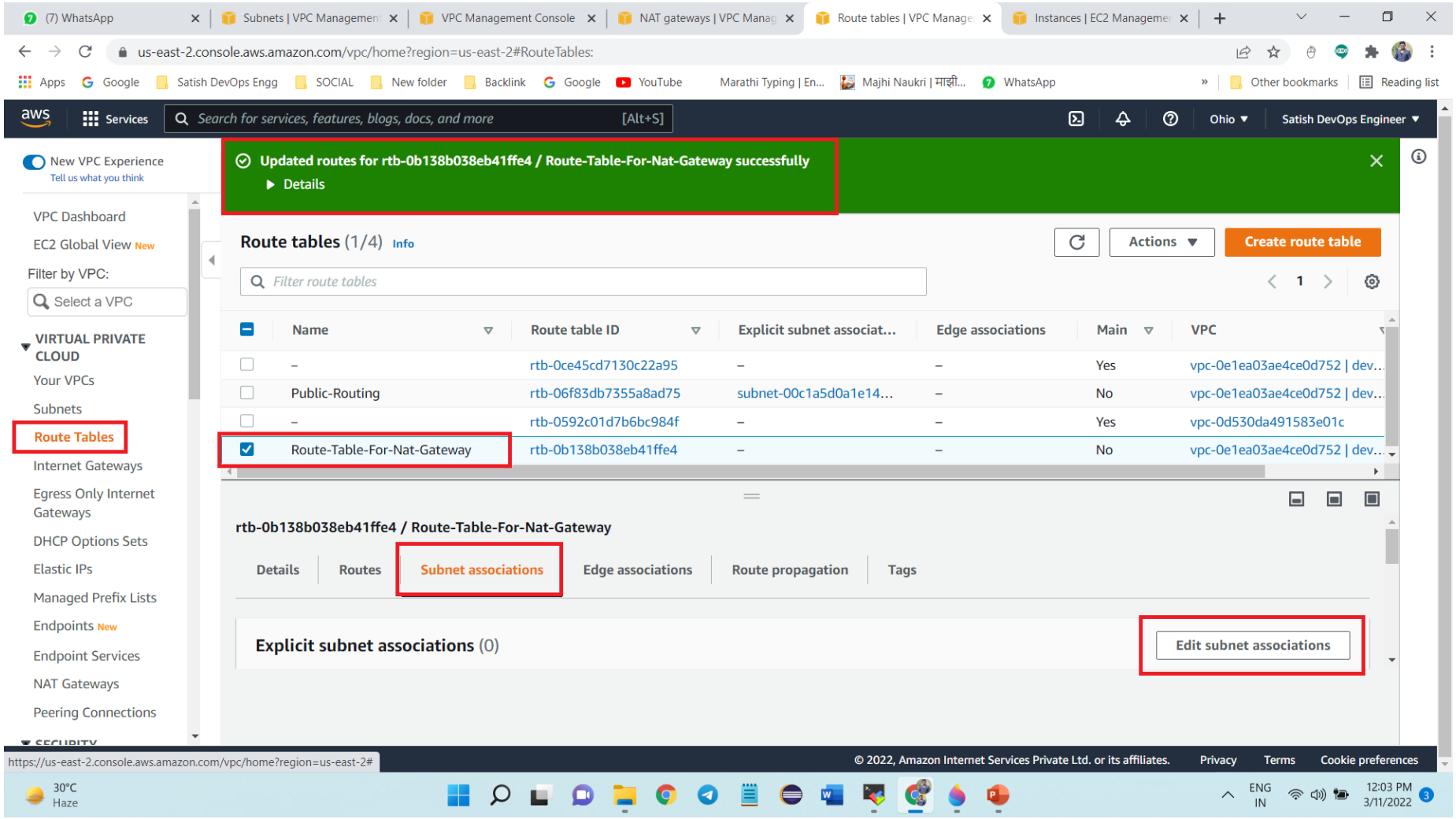
- Select your NAT Gateway



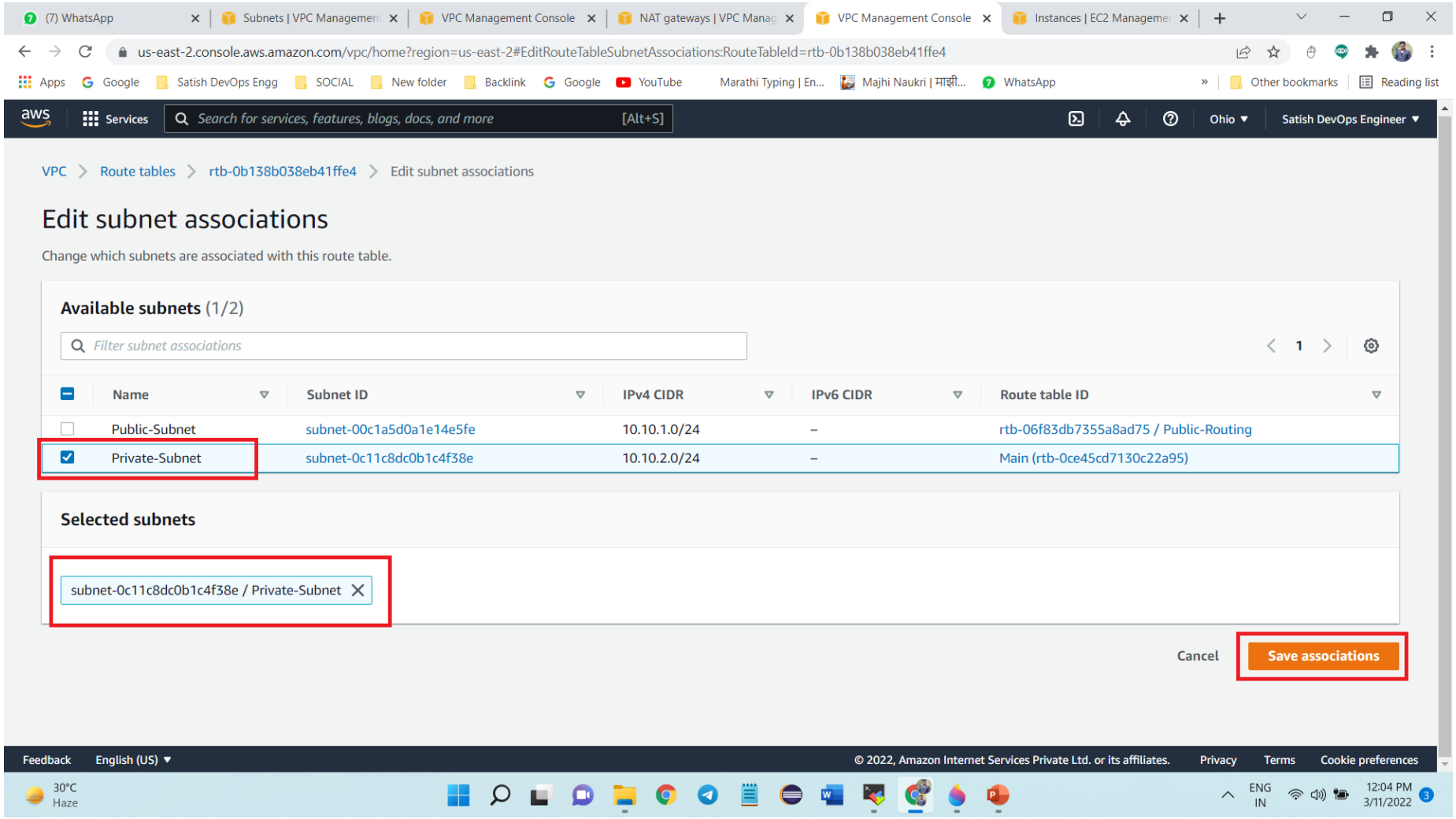
- Click on Save Changes



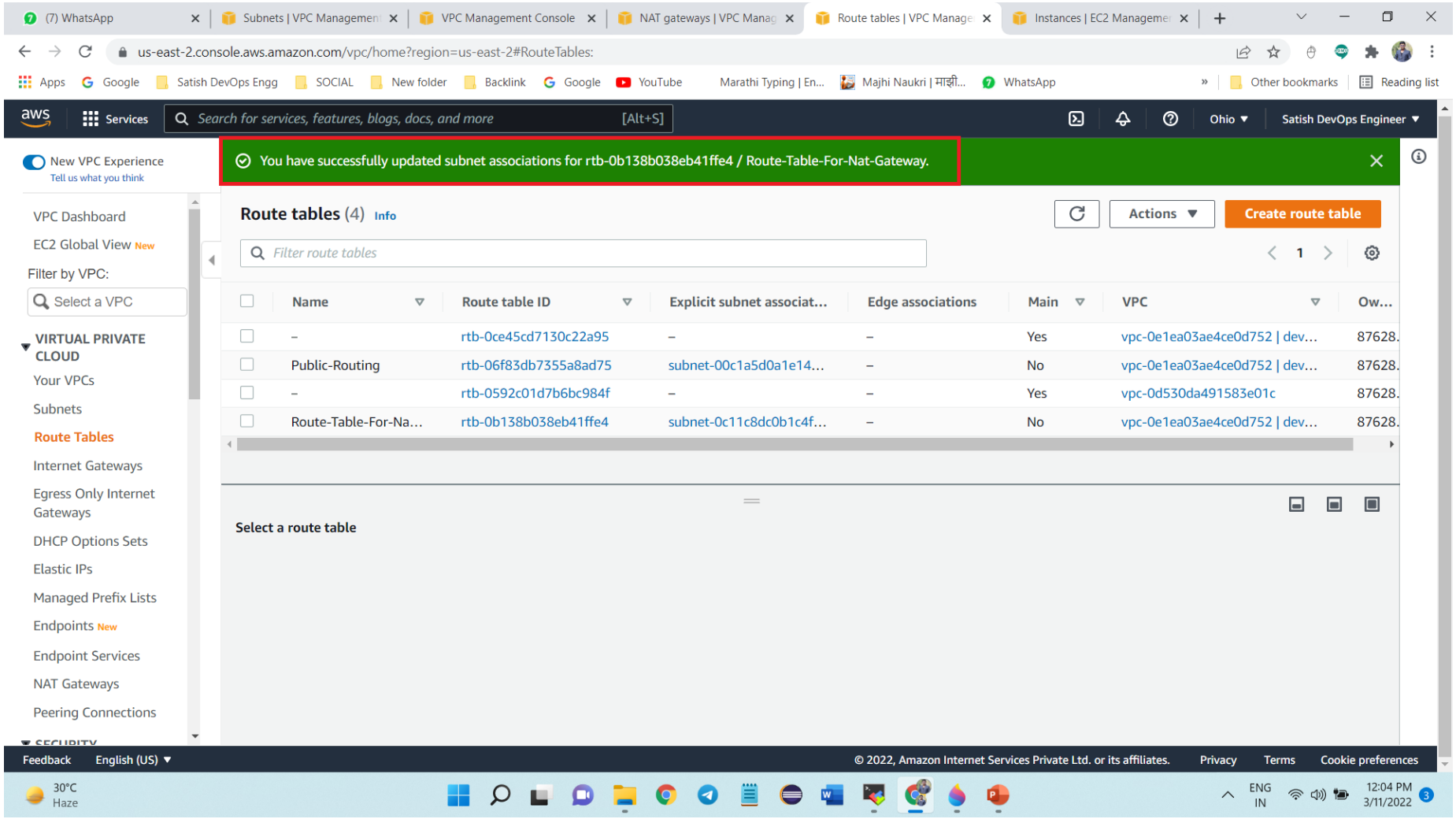
- We have updated route table for NAT Gateway Successfully.



- **Go to Route Tables again**
- **Select our NAT Gateway Route Table**
- **Click on Subnet Associations**
- **Click on Edit Subnet Associations**



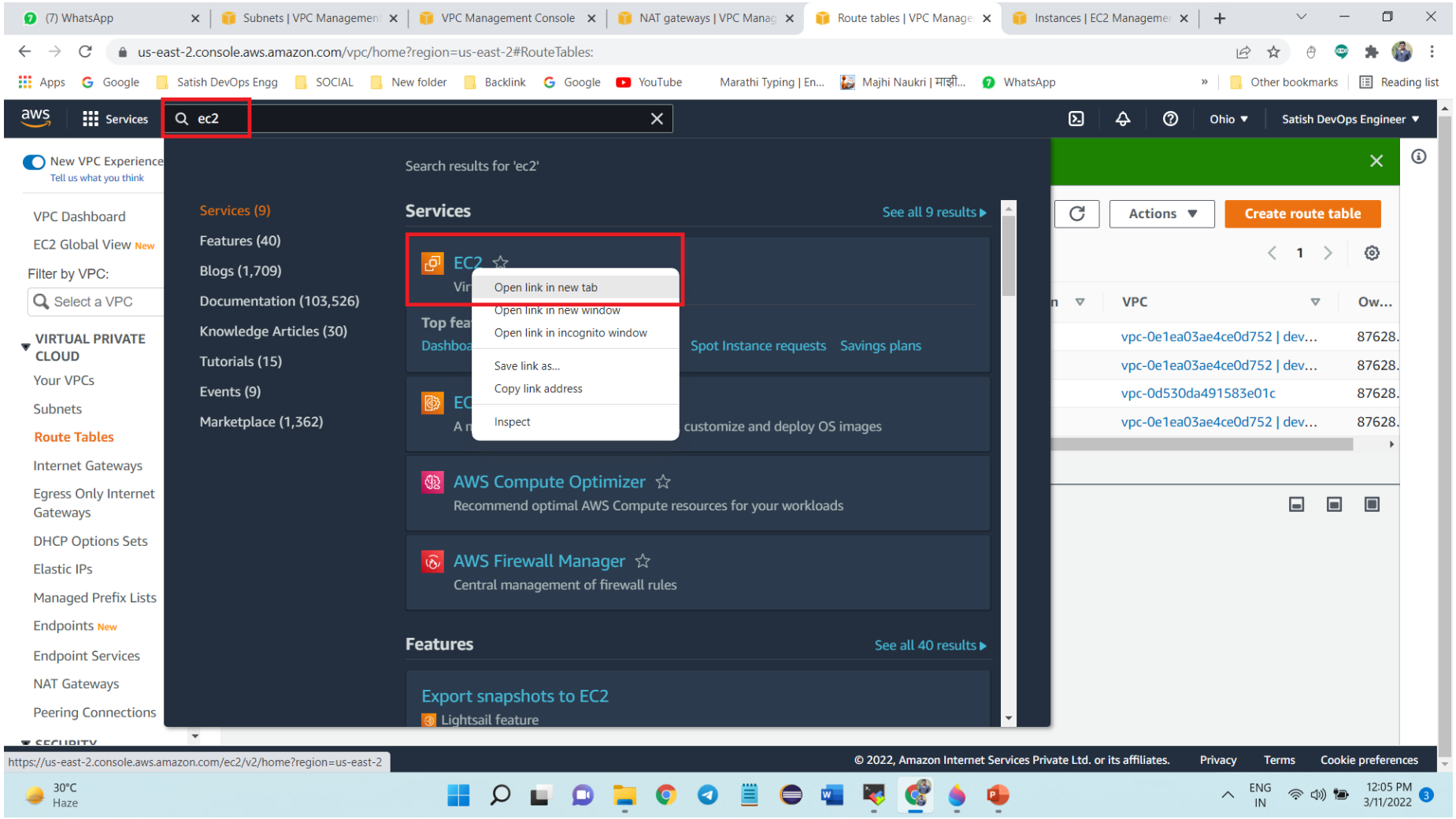
- Select Private Subnet
- Click on Save Associations



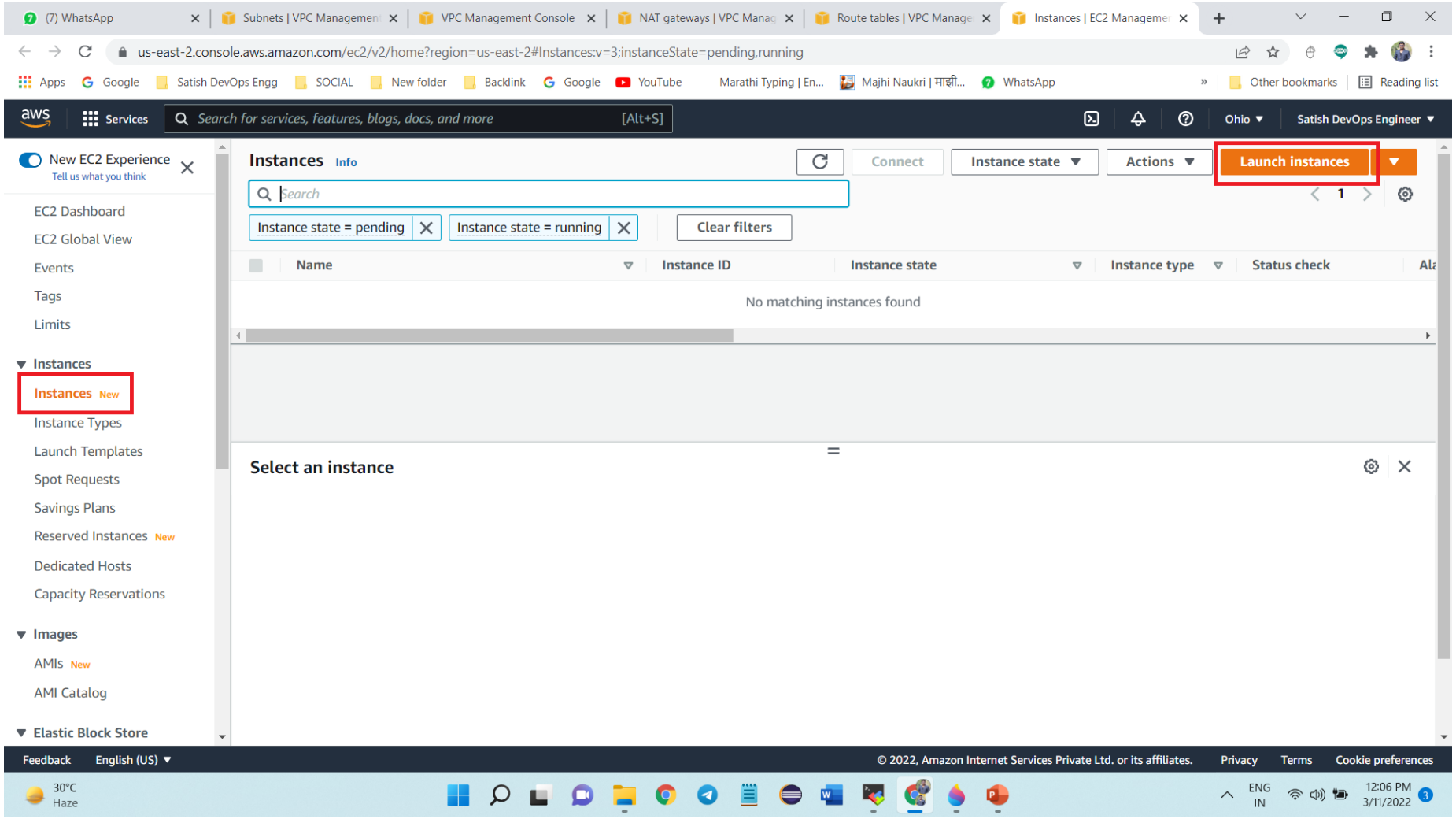
- Our private subnet is associated with NAT Gateway Successfully.

Now we have to launch EC2 Instances with our created **VPC** and **Subnets**

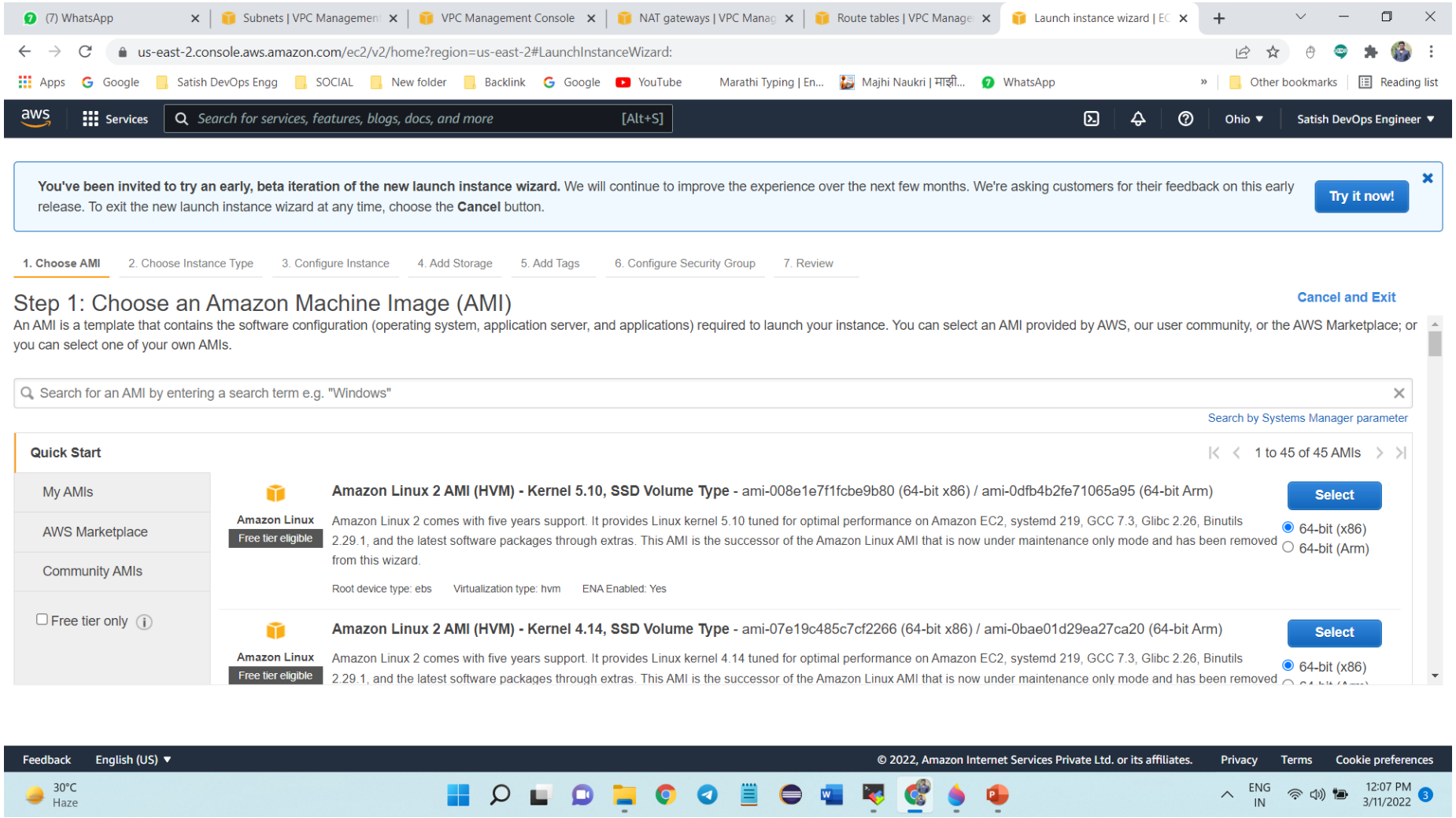
1. **With Public IP and in Public Subnet**
2. **Without Public IP and in Public Subnet**
3. **Without Public IP and in Private Subnet**



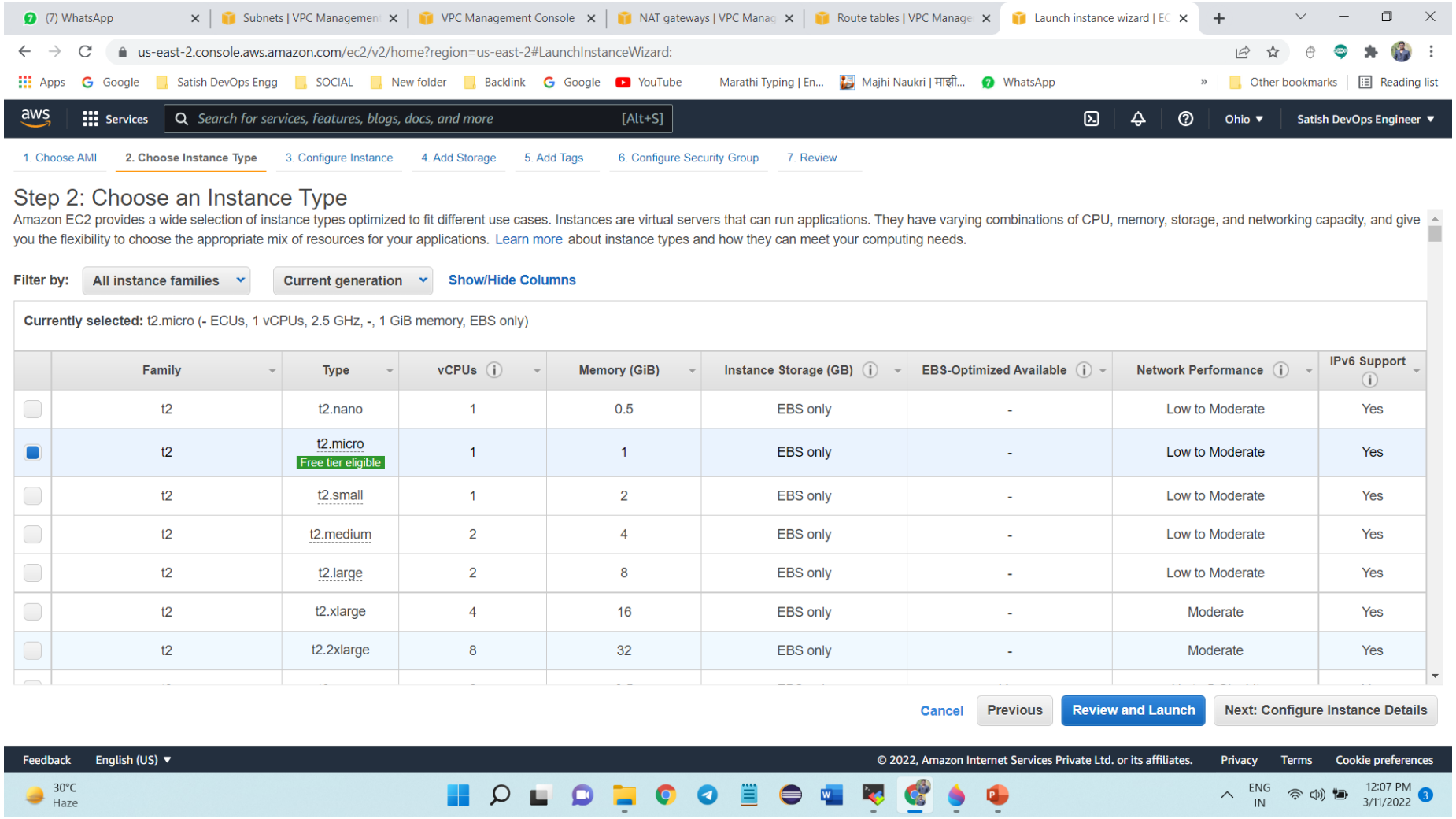
- Search EC2 in AWS Search Bar
- Open in New Tab



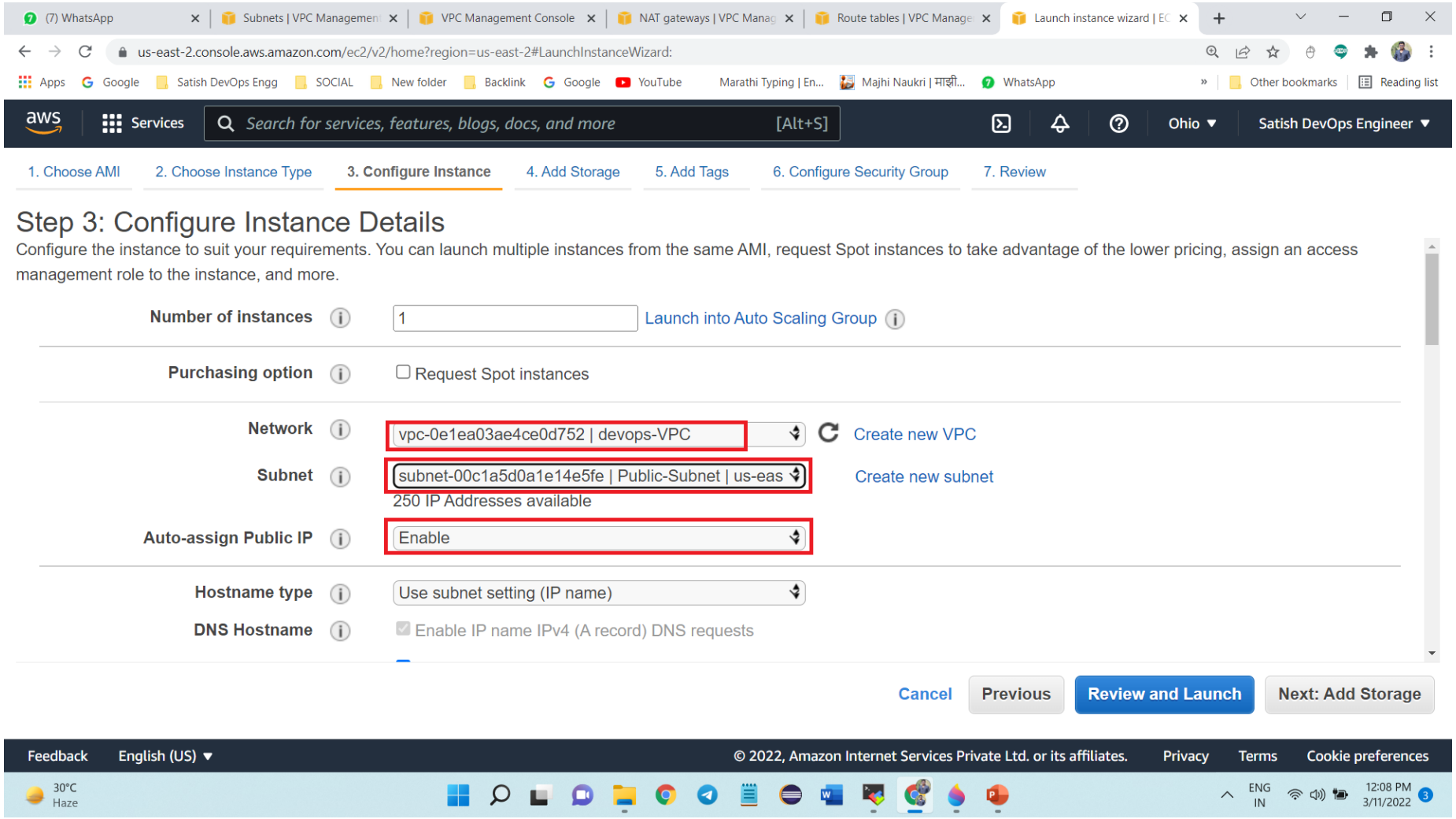
- Go to Instances
- Click on Launch Instances


(7) WhatsApp
Subnets | VPC Management
VPC Management Console
NAT gateways | VPC Manag
Route tables | VPC Manage
Launch instance wizard | EC
us-east-2.console.aws.amazon.com/ec2/v2/home?region=us-east-2#LaunchInstanceWizard:
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
You've been invited to try an early, beta iteration of the new launch instance wizard. We will continue to improve the experience over the next few months. We're asking customers for their feedback on this early release. To exit the new launch instance wizard at any time, choose the Cancel button.
Try it now!
1. Choose AMI
2. Choose Instance Type
3. Configure Instance
4. Add Storage
5. Add Tags
6. Configure Security Group
7. Review
Step 1: Choose an Amazon Machine Image (AMI)
Cancel and Exit
An AMI is a template that contains the software configuration (operating system, application server, and applications) required to launch your instance. You can select an AMI provided by AWS, our user community, or the AWS Marketplace; or you can select one of your own AMIs.
Search for an AMI by entering a search term e.g. "Windows"
Search by Systems Manager parameter
Quick Start
1 to 45 of 45 AMIs
My AMIs
AWS Marketplace
Community AMIs
Free tier only
Amazon Linux
Free tier eligible
Amazon Linux 2 AMI (HVM) - Kernel 5.10, SSD Volume Type - ami-008e1e7f1fcbe9b80 (64-bit x86) / ami-0dfb4b2fe71065a95 (64-bit Arm)
Amazon Linux 2 comes with five years support. It provides Linux kernel 5.10 tuned for optimal performance on Amazon EC2, systemd 219, GCC 7.3, Glibc 2.26, Binutils 2.29.1, and the latest software packages through extras. This AMI is the successor of the Amazon Linux AMI that is now under maintenance only mode and has been removed from this wizard.
Root device type: ebs
Virtualization type: hvm
ENA Enabled: Yes
Select
64-bit (x86)
64-bit (Arm)
Amazon Linux
Free tier eligible
Amazon Linux 2 AMI (HVM) - Kernel 4.14, SSD Volume Type - ami-07e19c485c7cf2266 (64-bit x86) / ami-0bae01d29ea27ca20 (64-bit Arm)
Amazon Linux 2 comes with five years support. It provides Linux kernel 4.14 tuned for optimal performance on Amazon EC2, systemd 219, GCC 7.3, Glibc 2.26, Binutils 2.29.1, and the latest software packages through extras. This AMI is the successor of the Amazon Linux AMI that is now under maintenance only mode and has been removed
Select
64-bit (x86)
Feedback
English (US)
© 2022, Amazon Internet Services Private Ltd. or its affiliates.
Privacy
Terms
Cookie preferences
30°C
Haze
ENG
IN
12:07 PM
3/11/2022

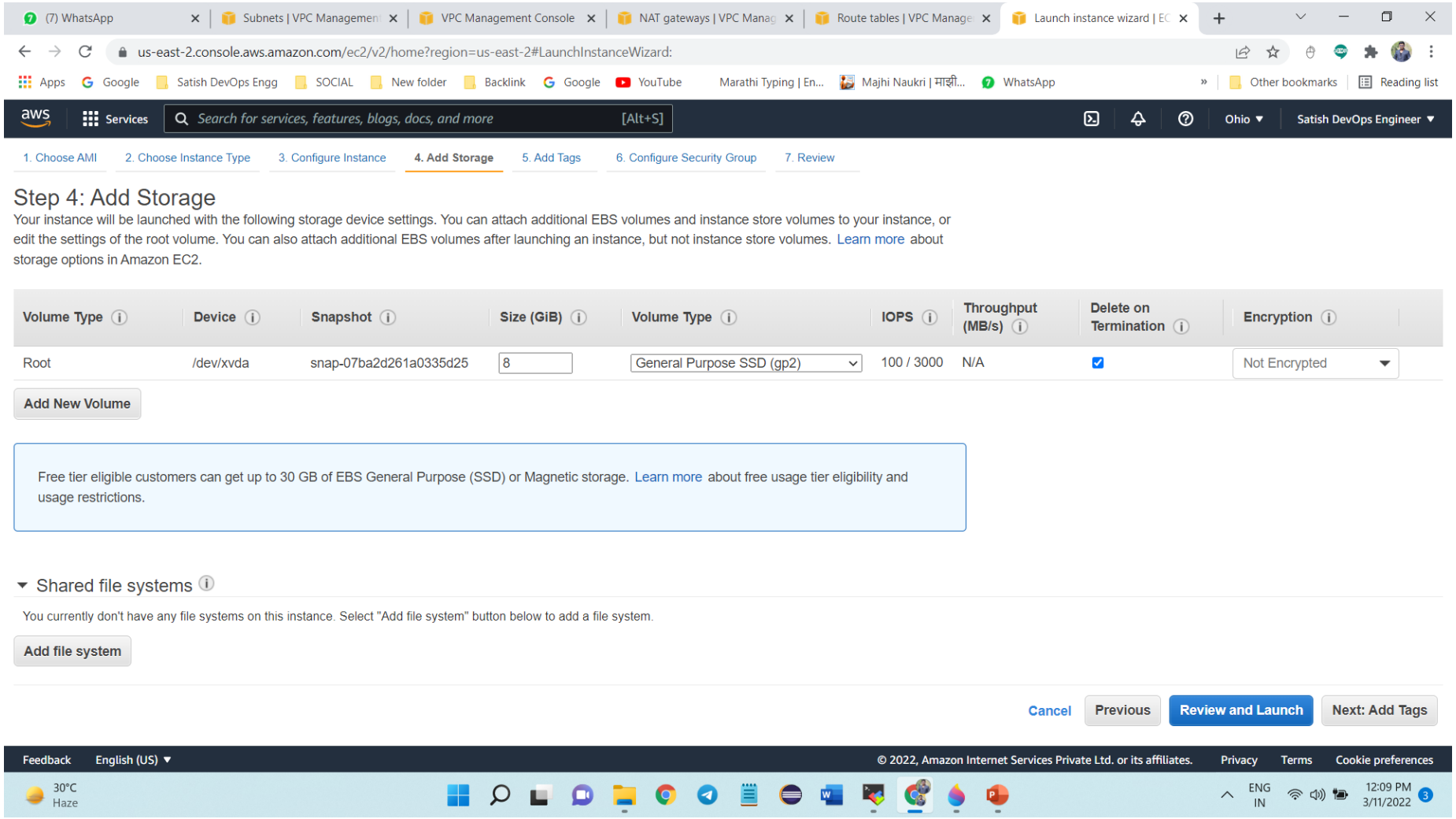
- Select AMI



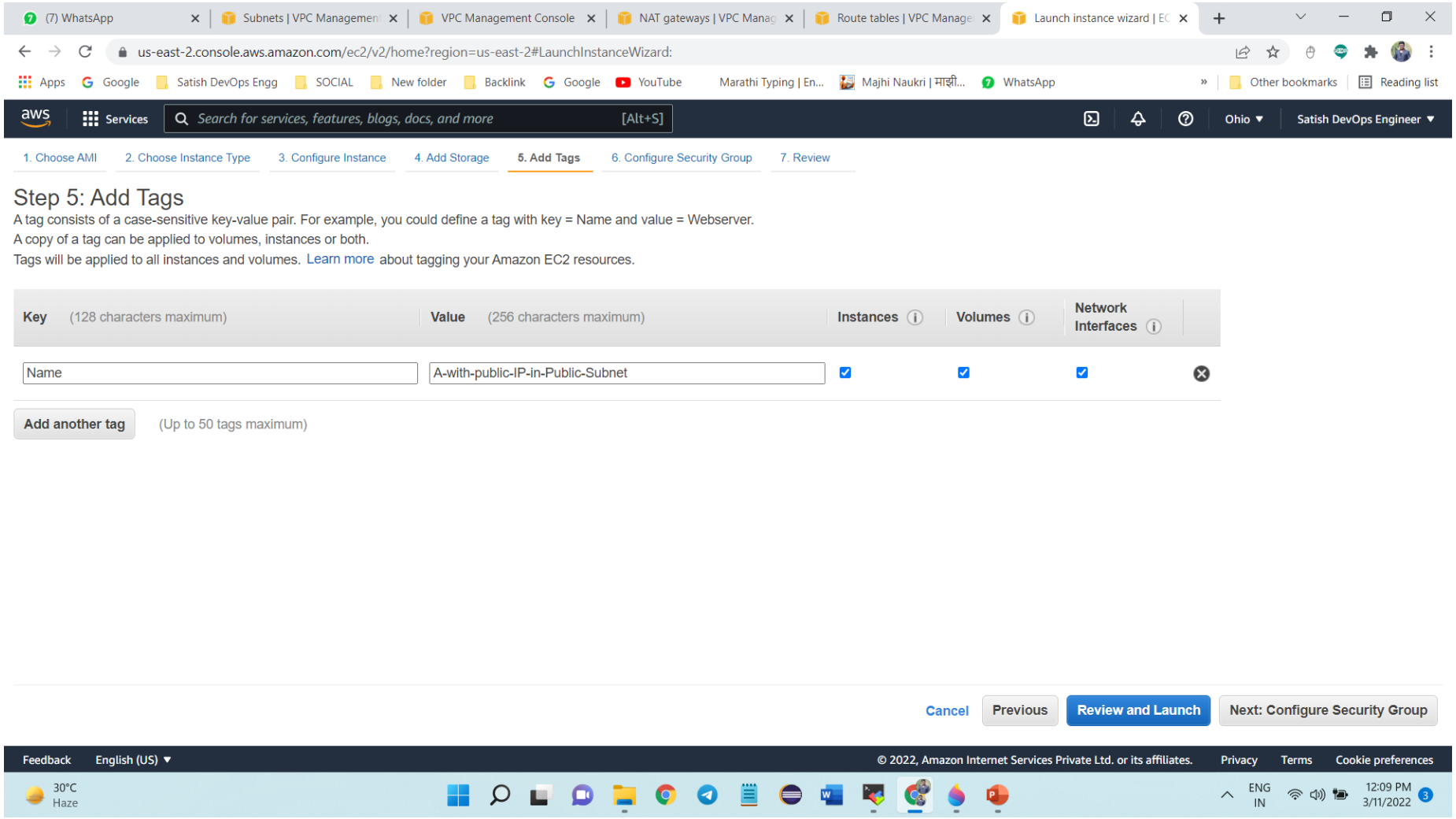
- Select Instance Type



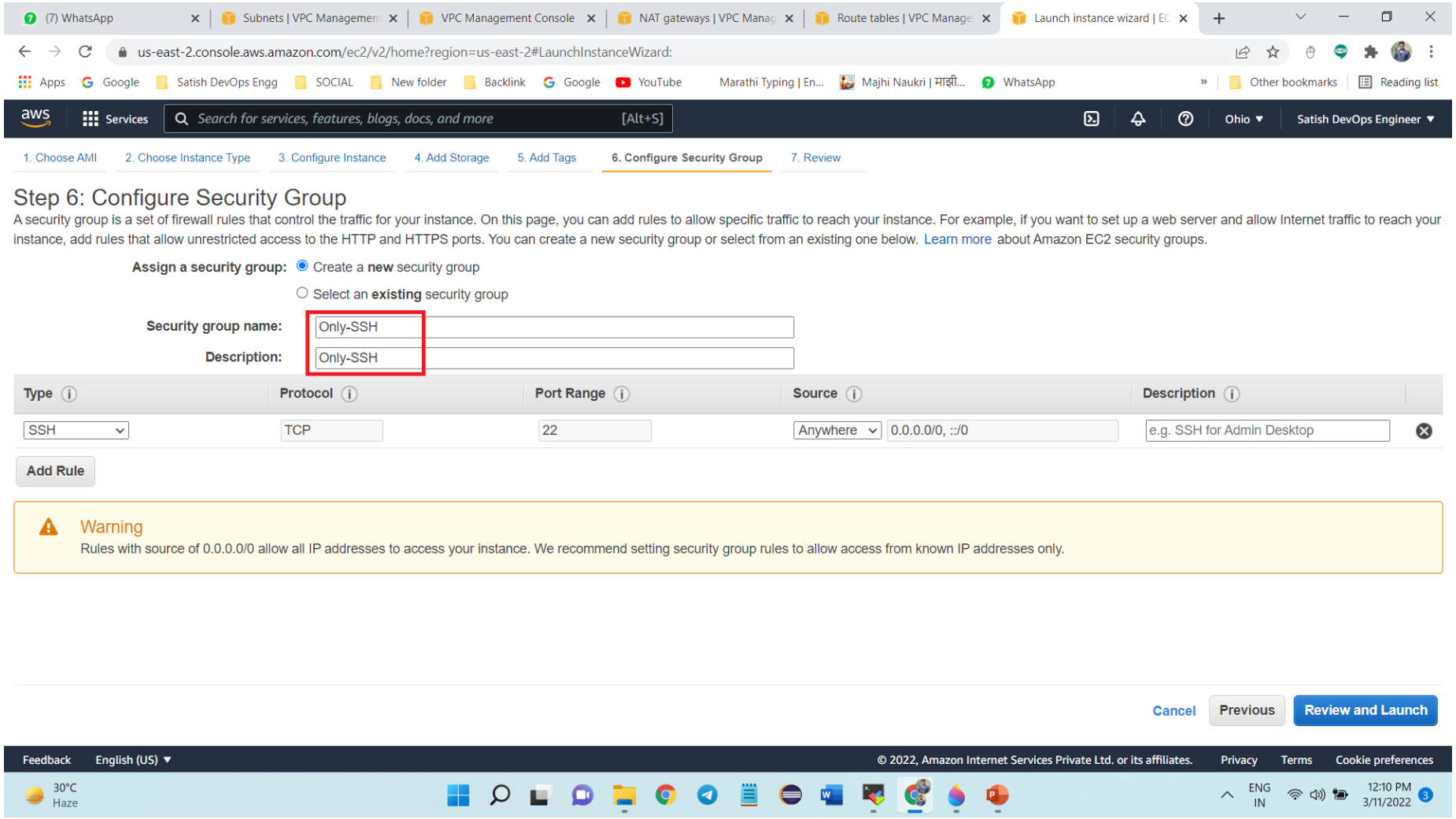
- **Select Our VPC**
- **Select Public Subnet**
- **Enable Public IP**
- **Click on Add Storage**



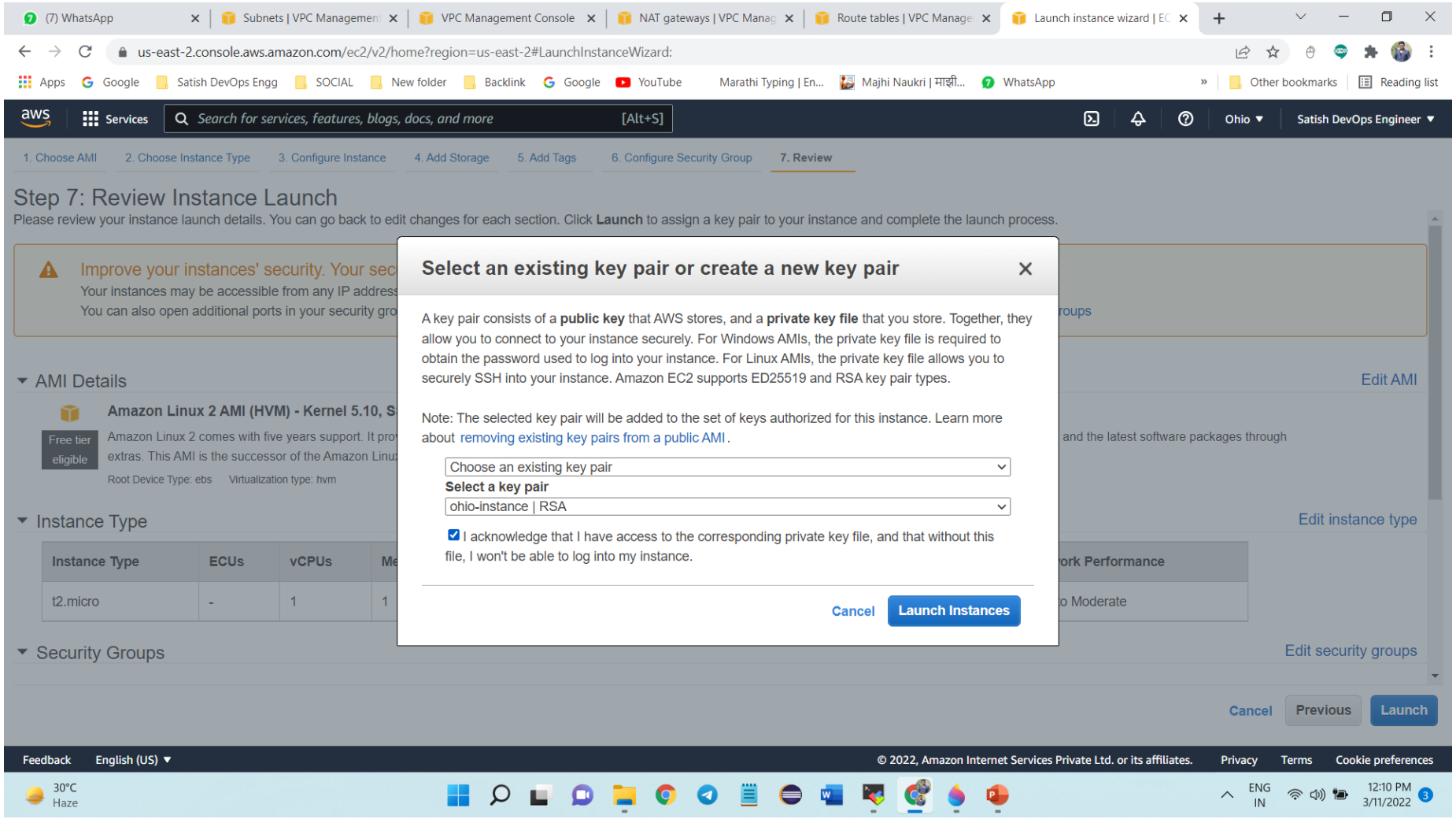
- Click on Next



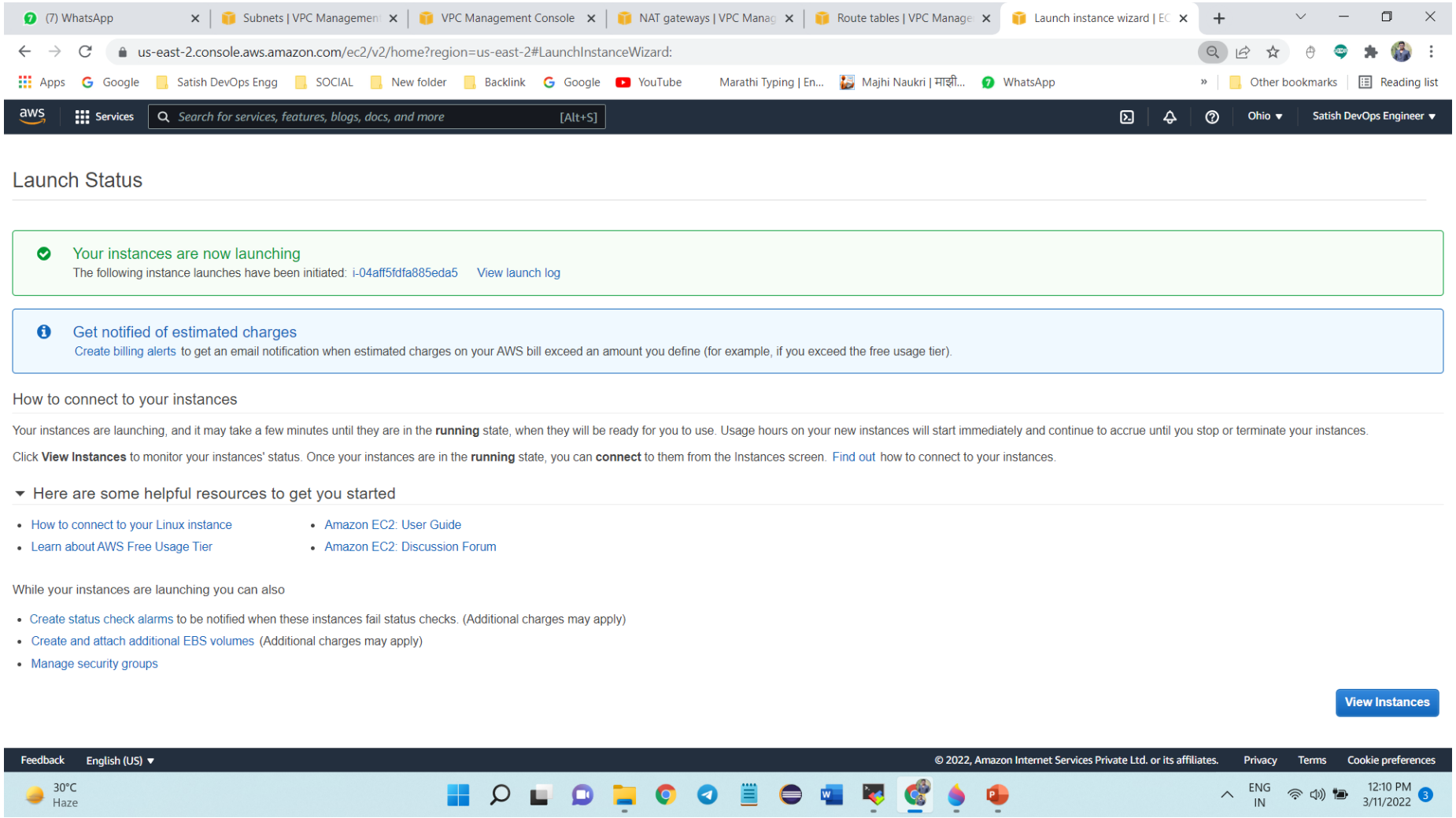
- Give Proper tags



- Create security group with SSH 22 as in inbound rule
- Click on Review and Launch

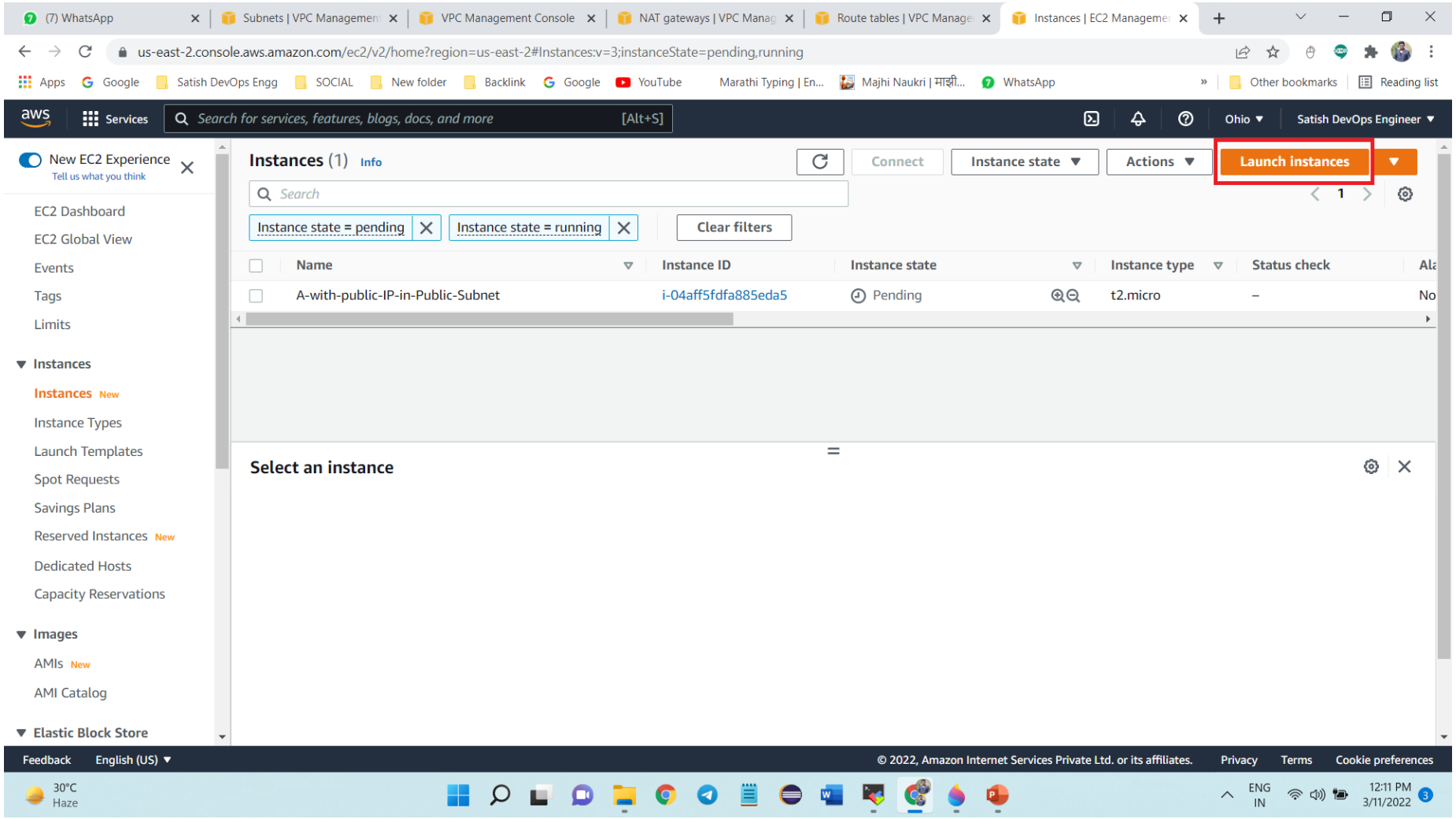


- Select Key-Pair
- Click on Launch Instances

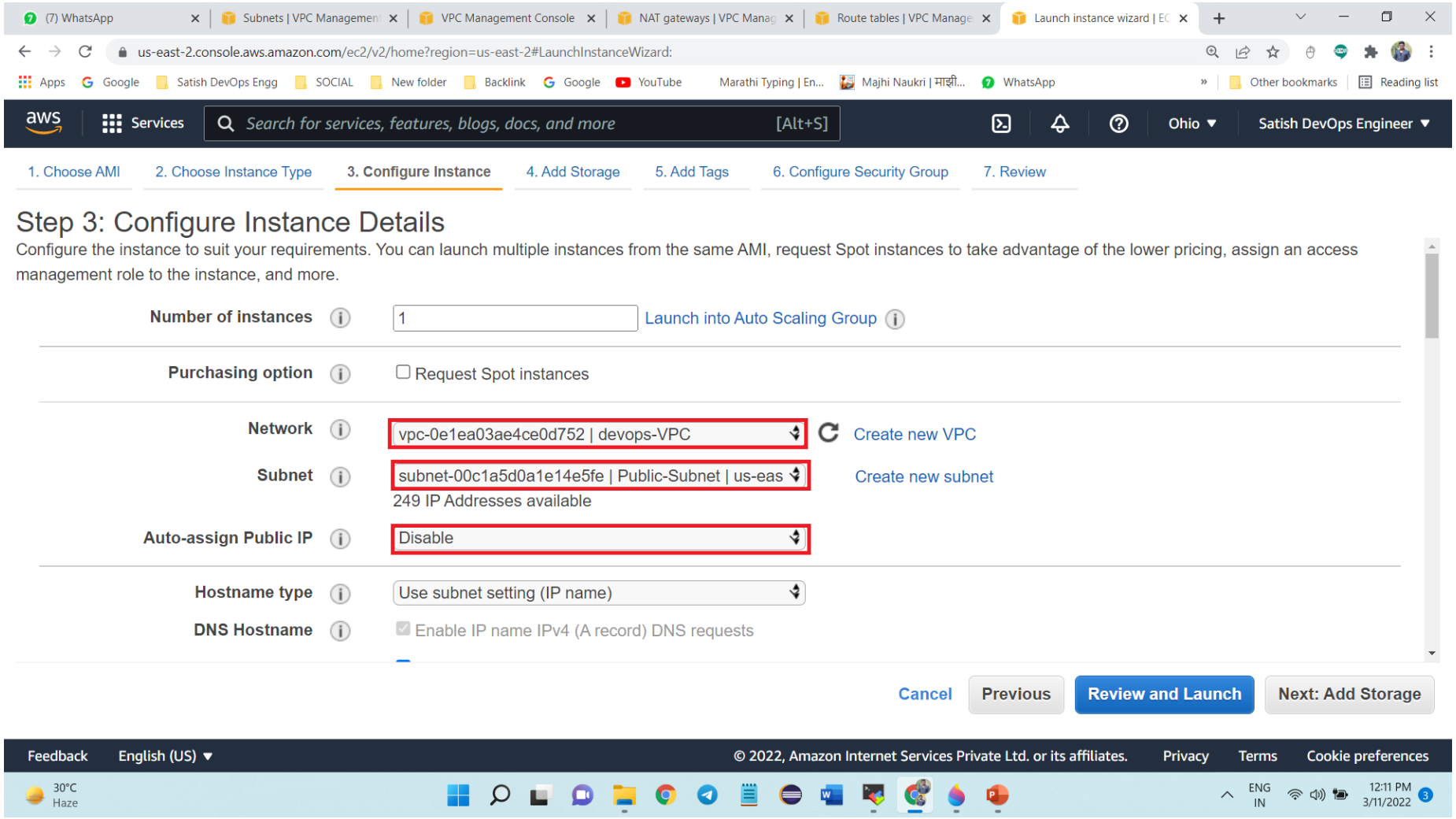


- Click on view instnaces

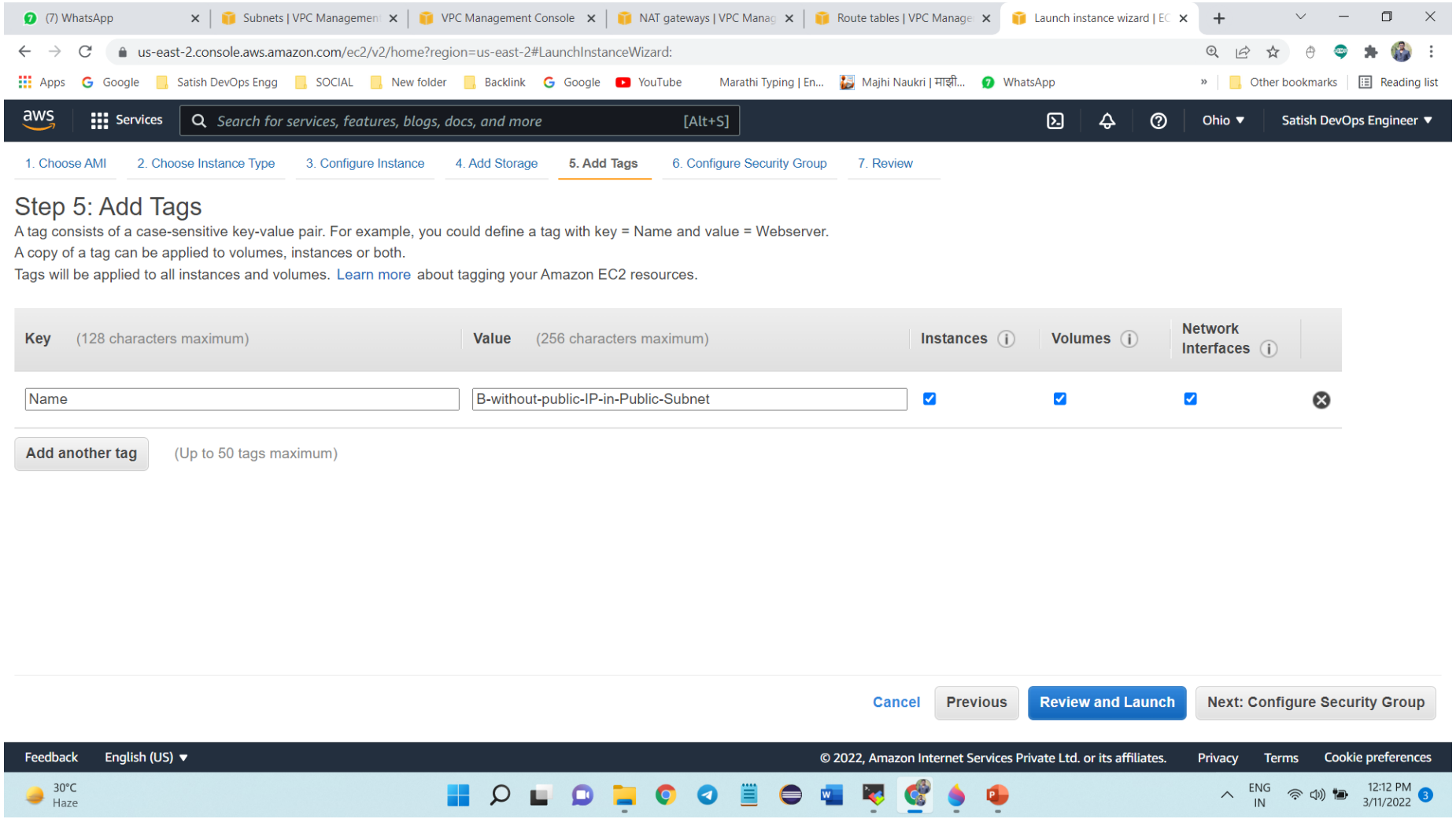
Now we will launch one more instance on Public Subnet but disable Public IP



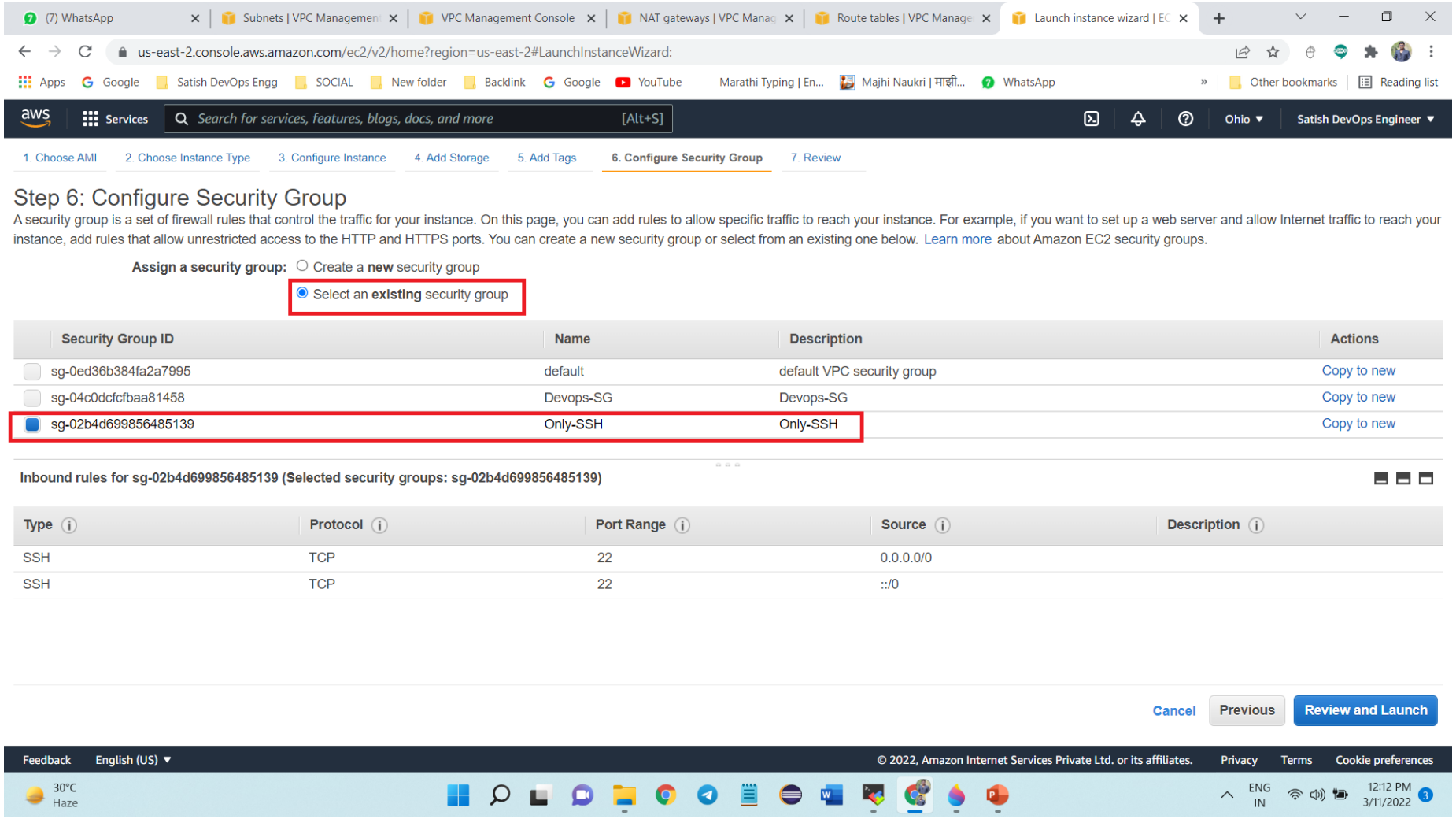
- Click on Launch Instances



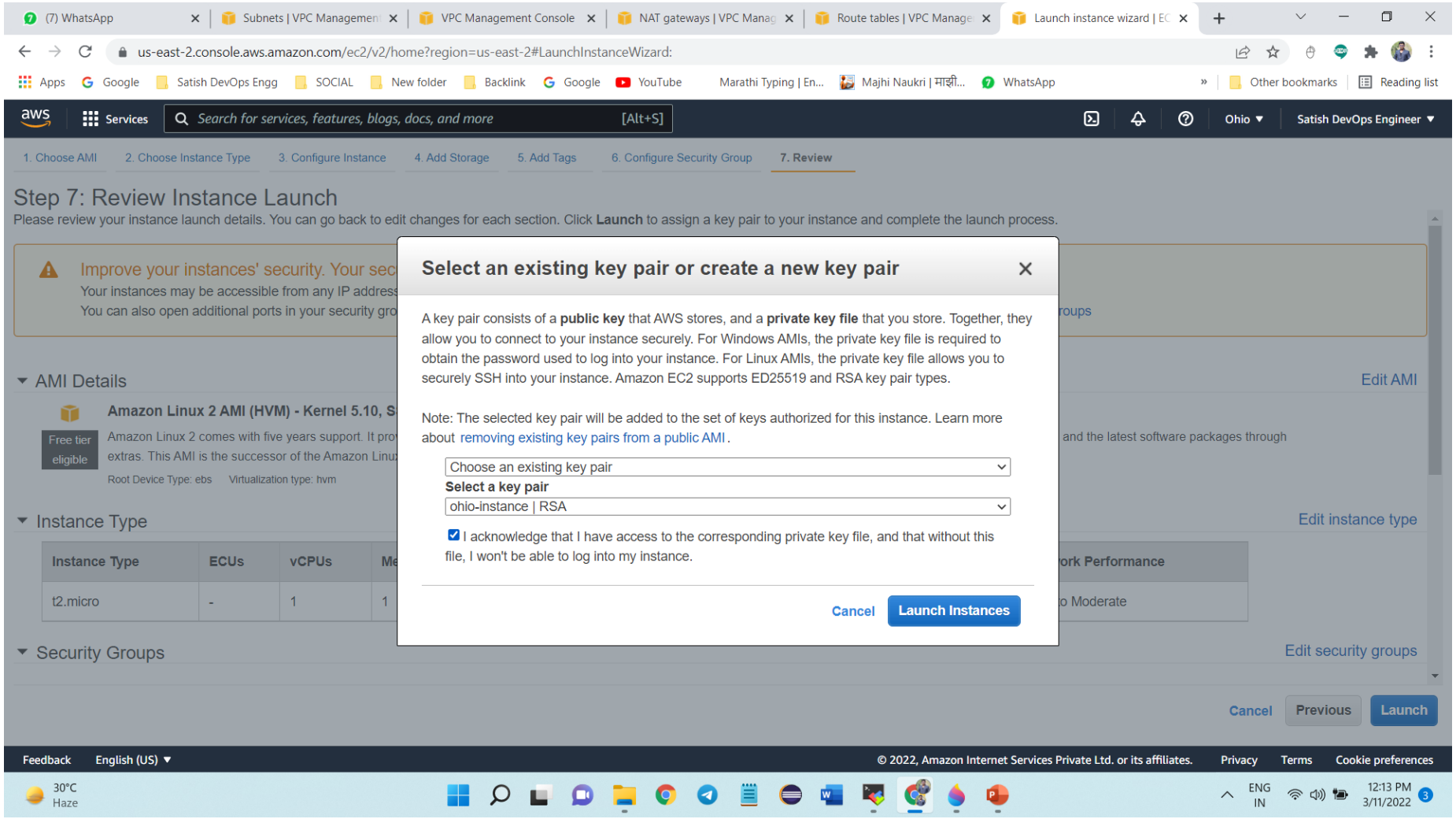
- **Select our VPC**
- **Select Public Subnet**
- **Disable Public IP**
- **Click Next**



- Give Proper Tags

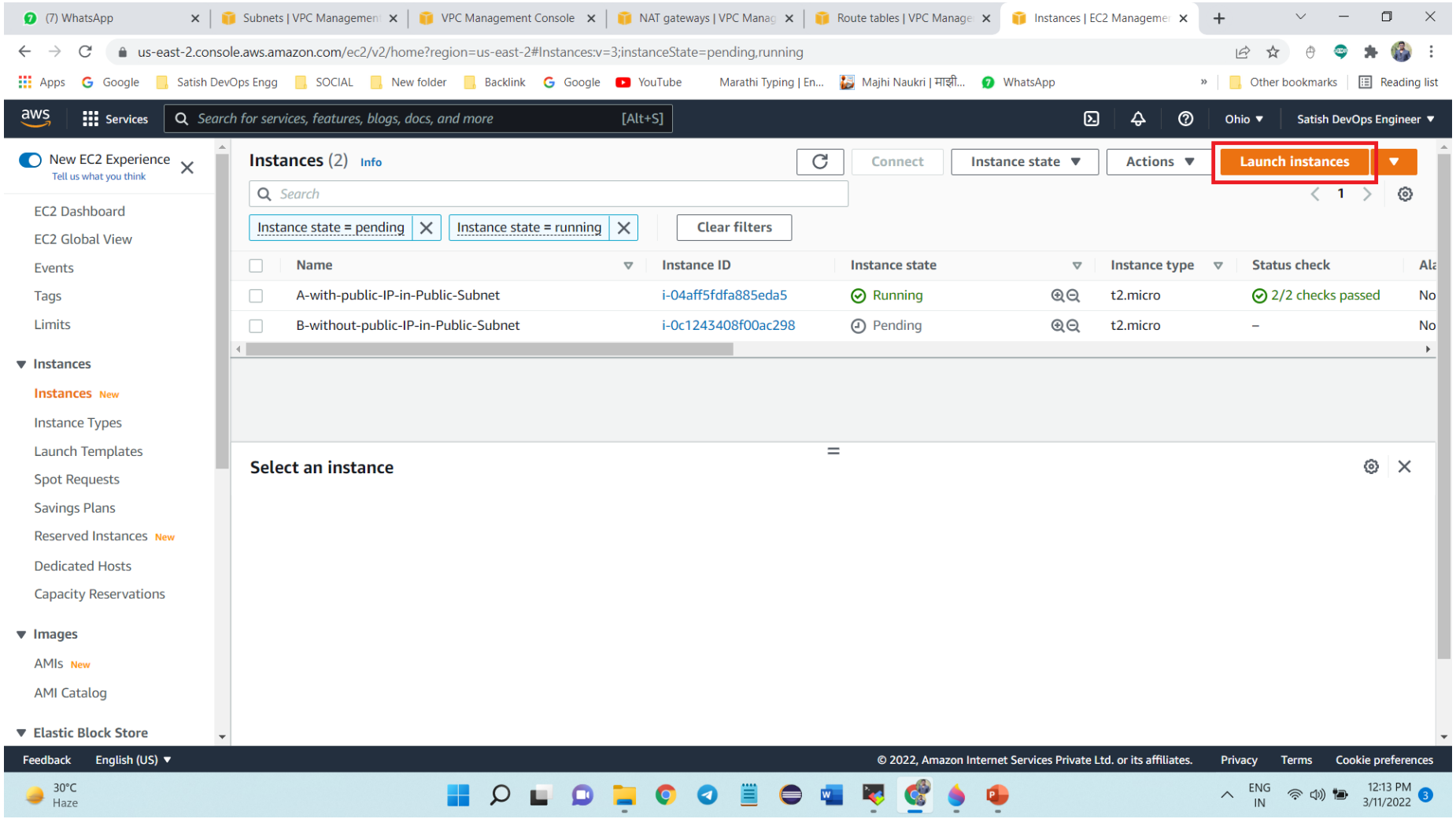


- Select our existing Security Group

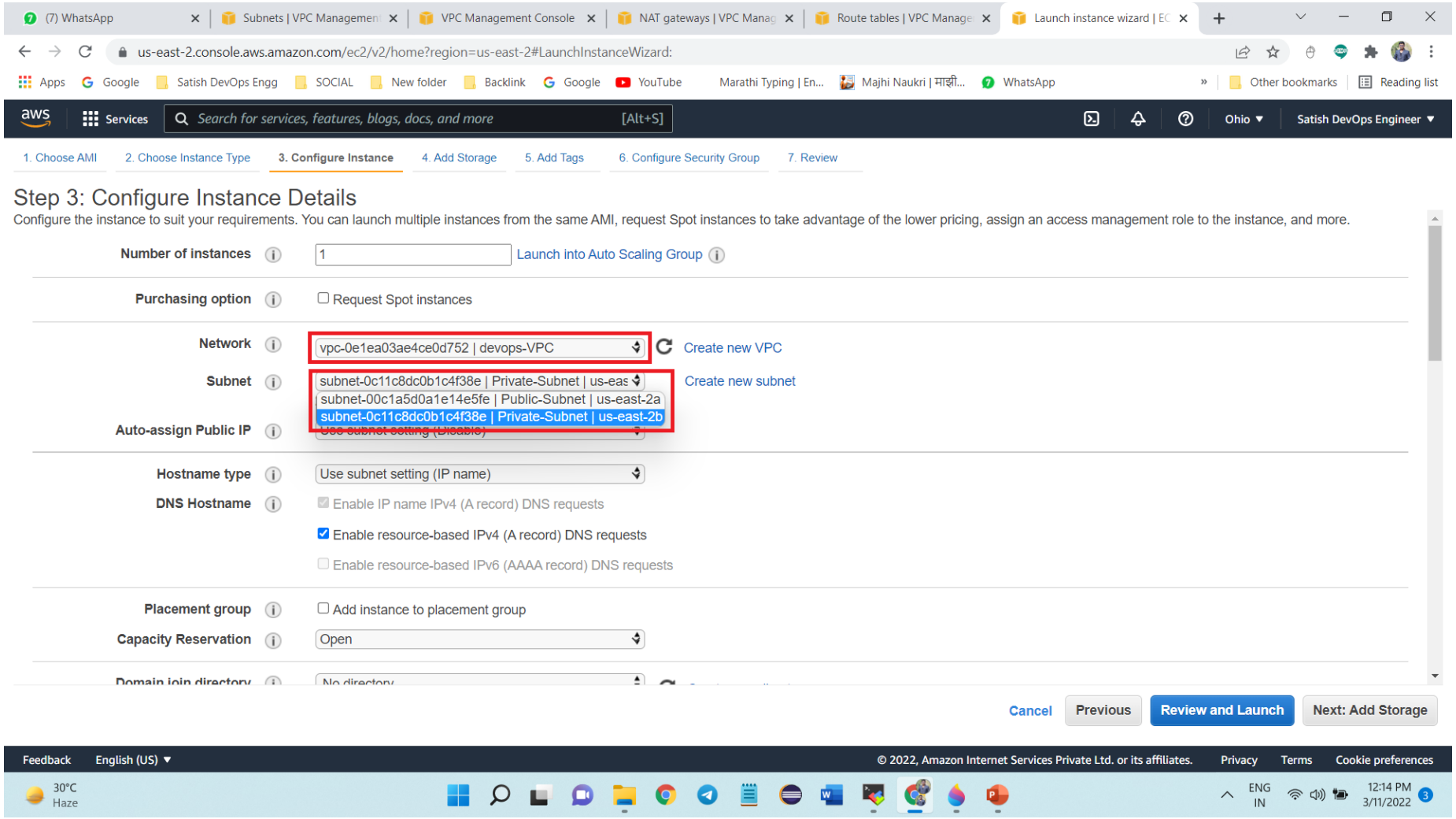


- Select Key-Pair
- Launch Instance

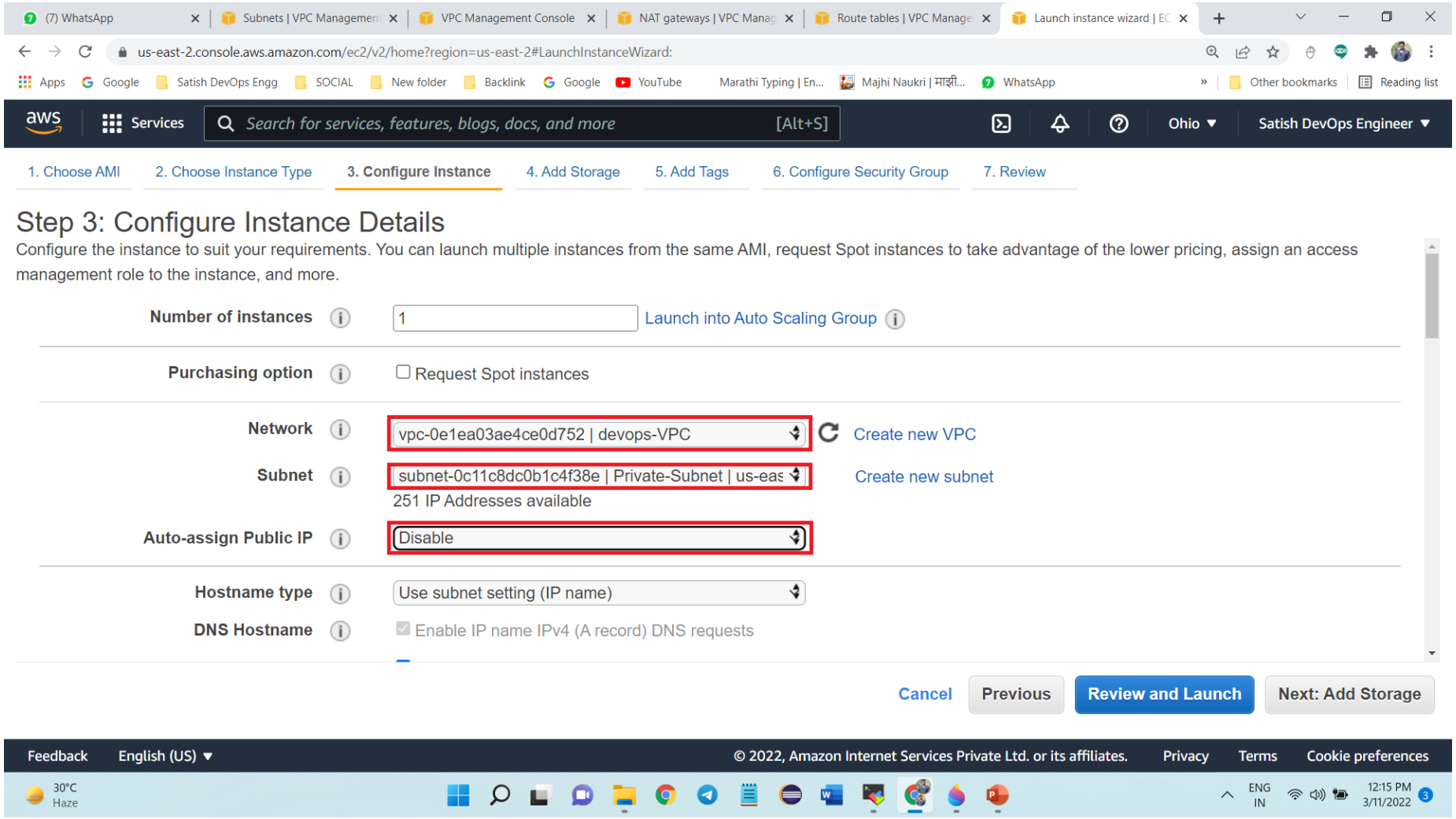
Now we will launch one more instance on Private Subnet and disable Public IP



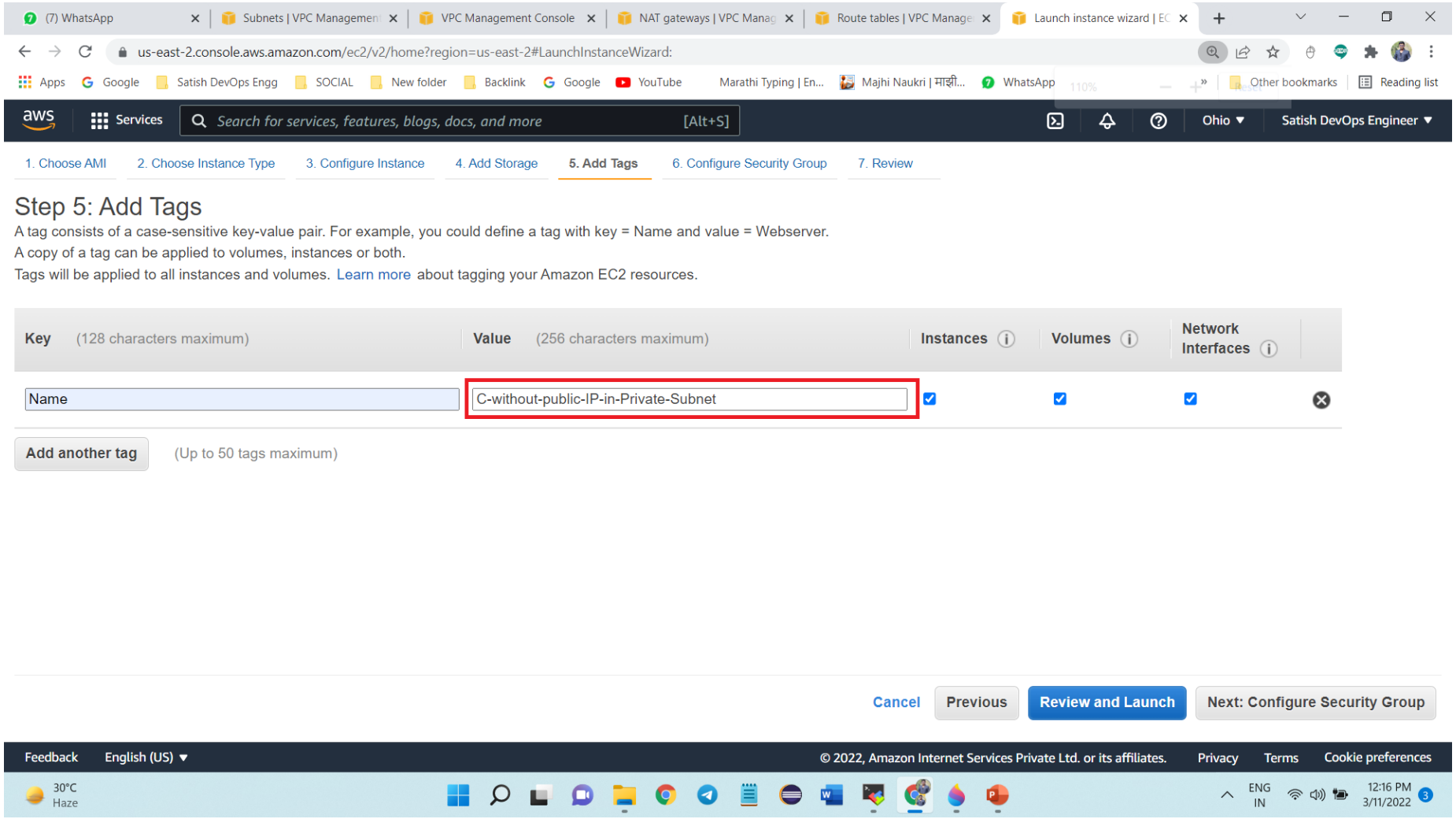
- Click on Launch Instnaces



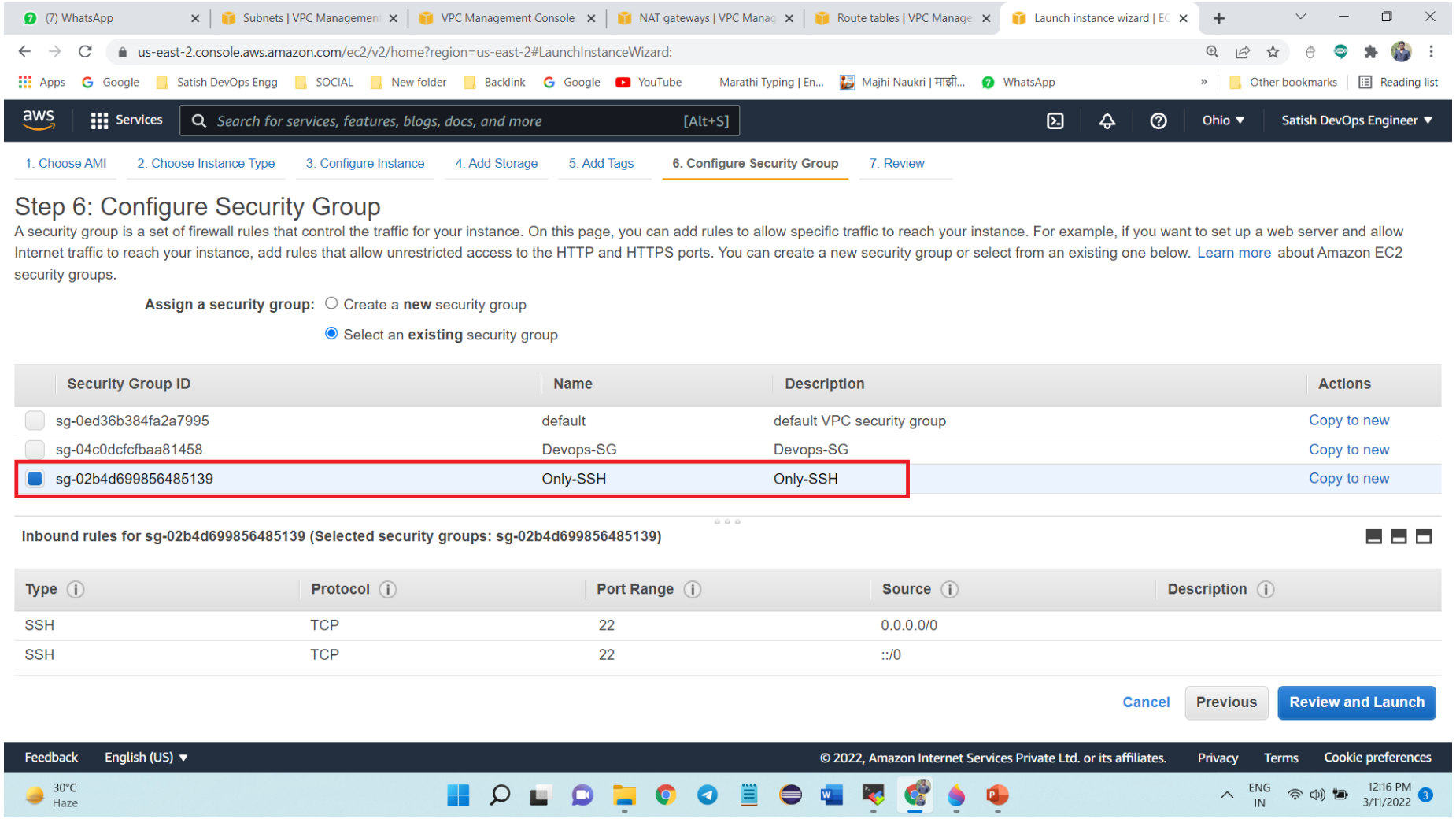
- Select our VPC
- Select Our Private Subnet



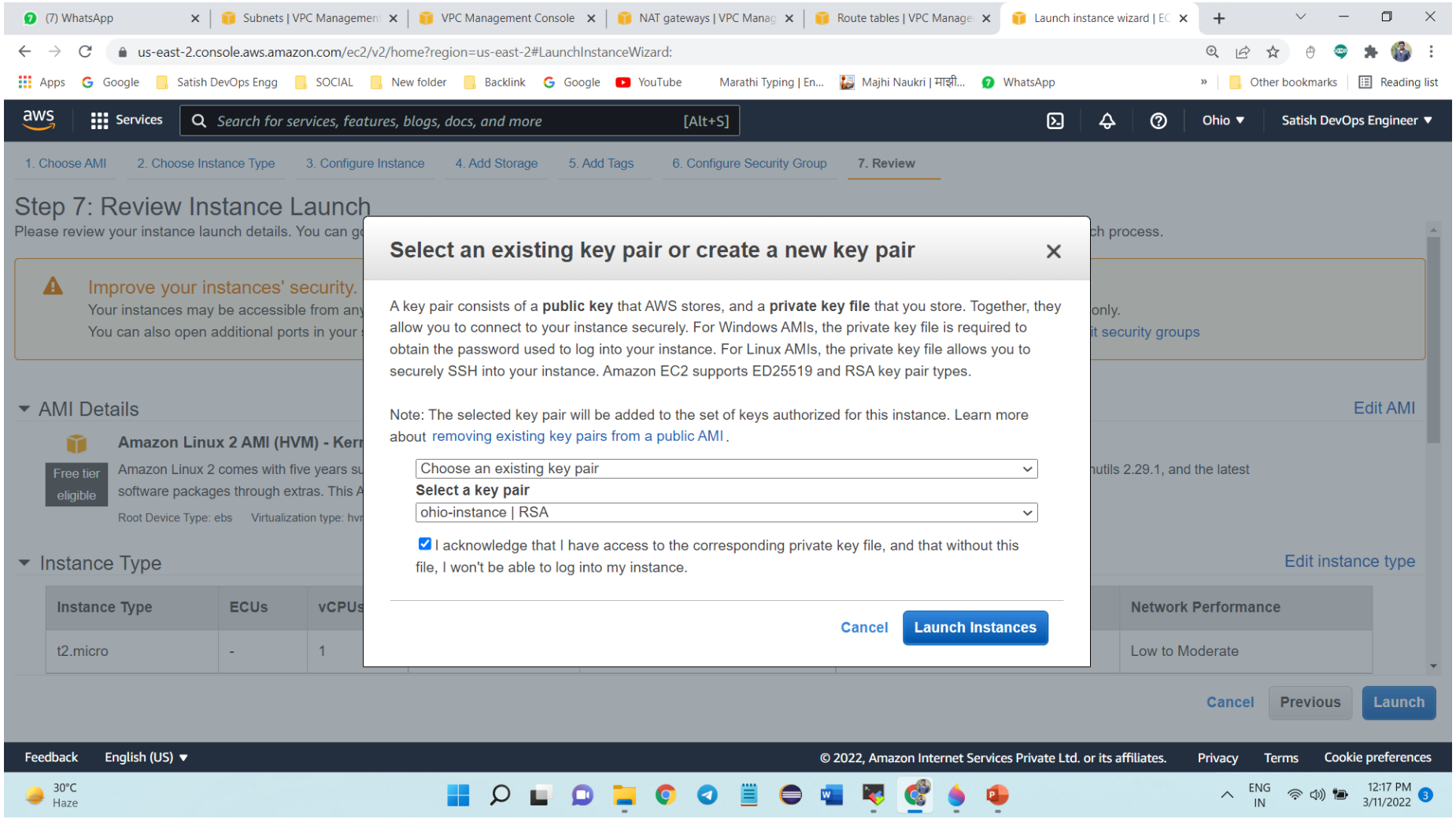
- Disable Public IP
- Click next



- Give Proper Tags



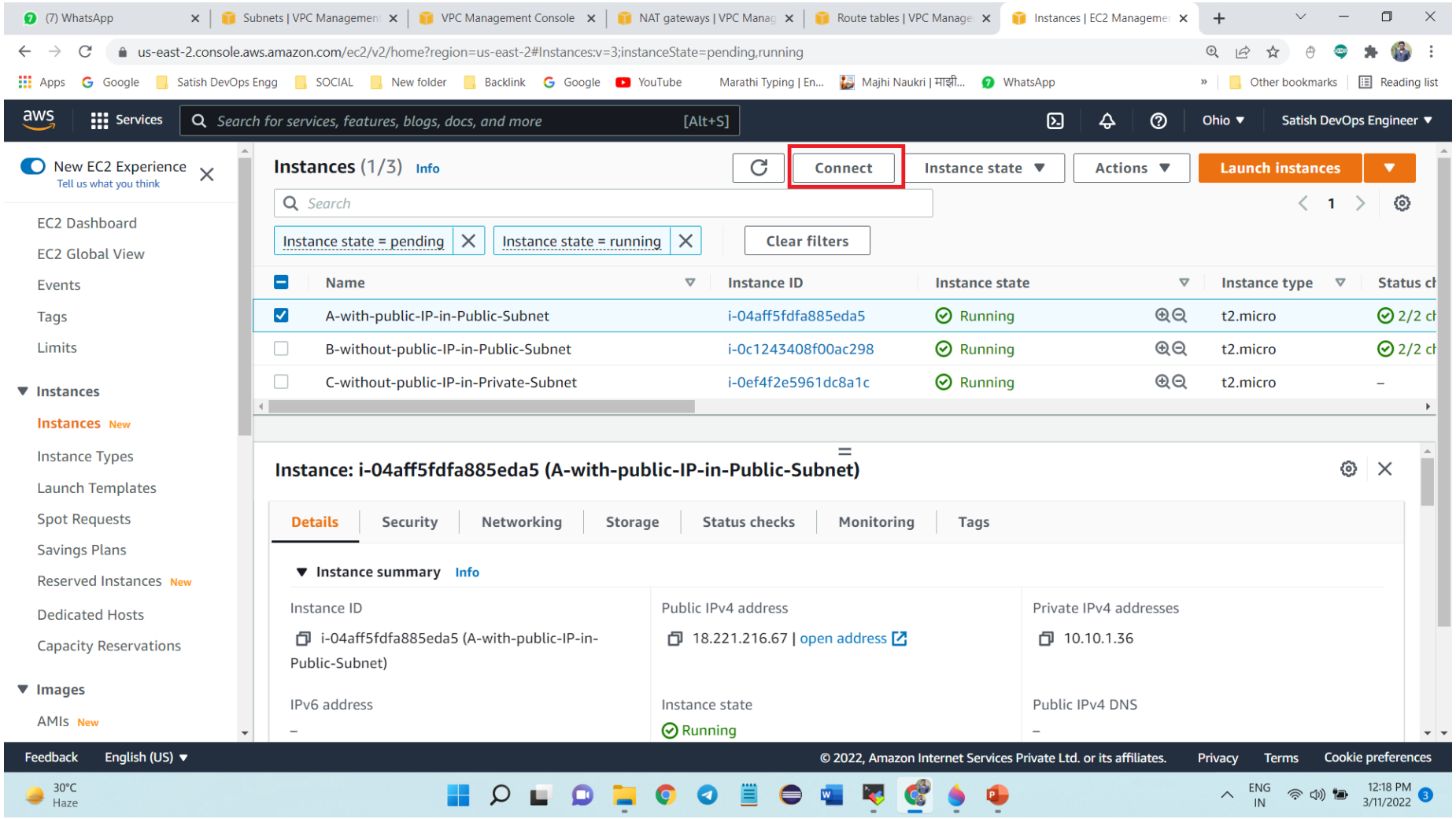
- Select Existing Security Group



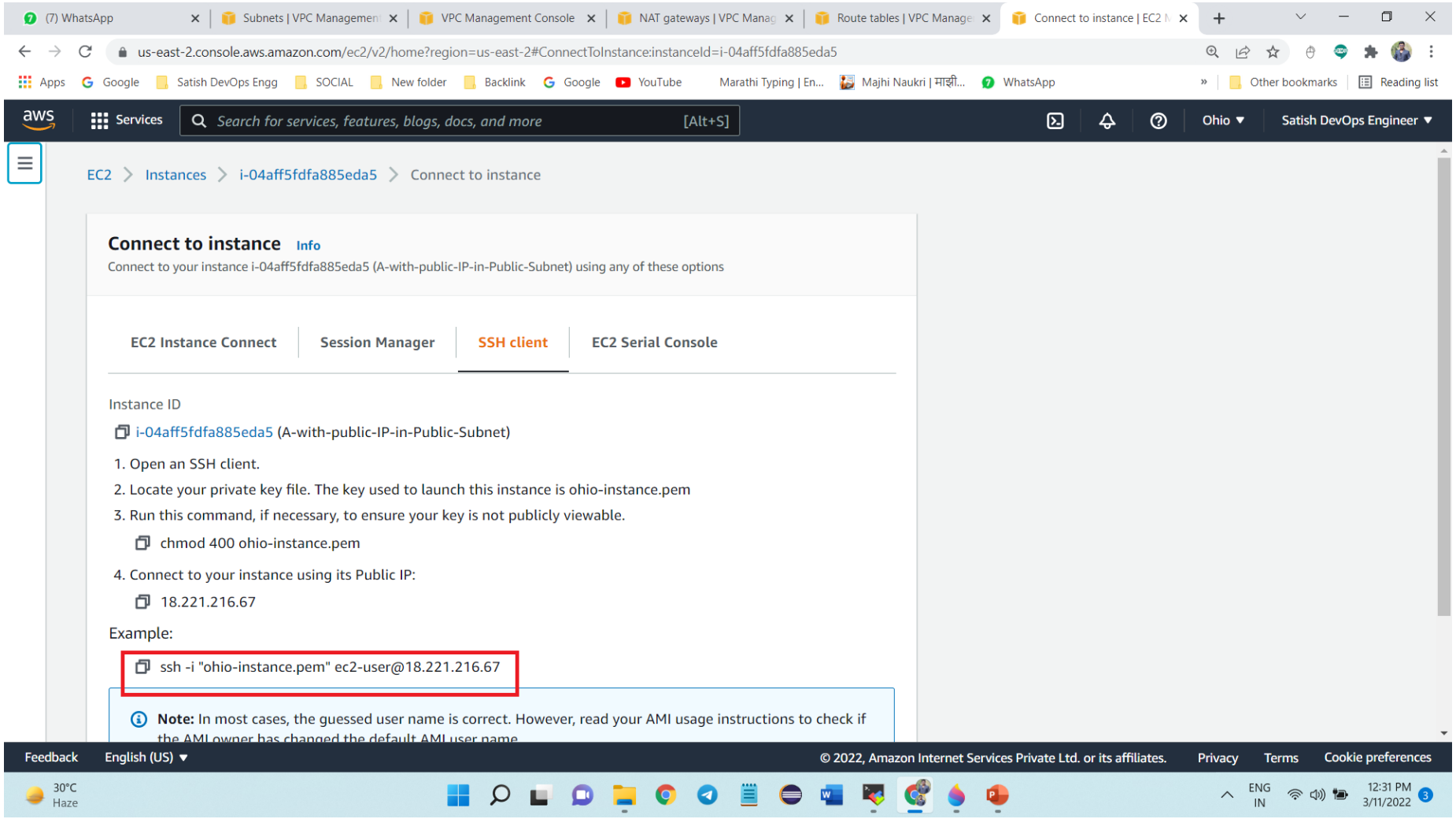
- Select Key-Pair
- Launch Instance

- **Here we have created all three instances as per requirement.**

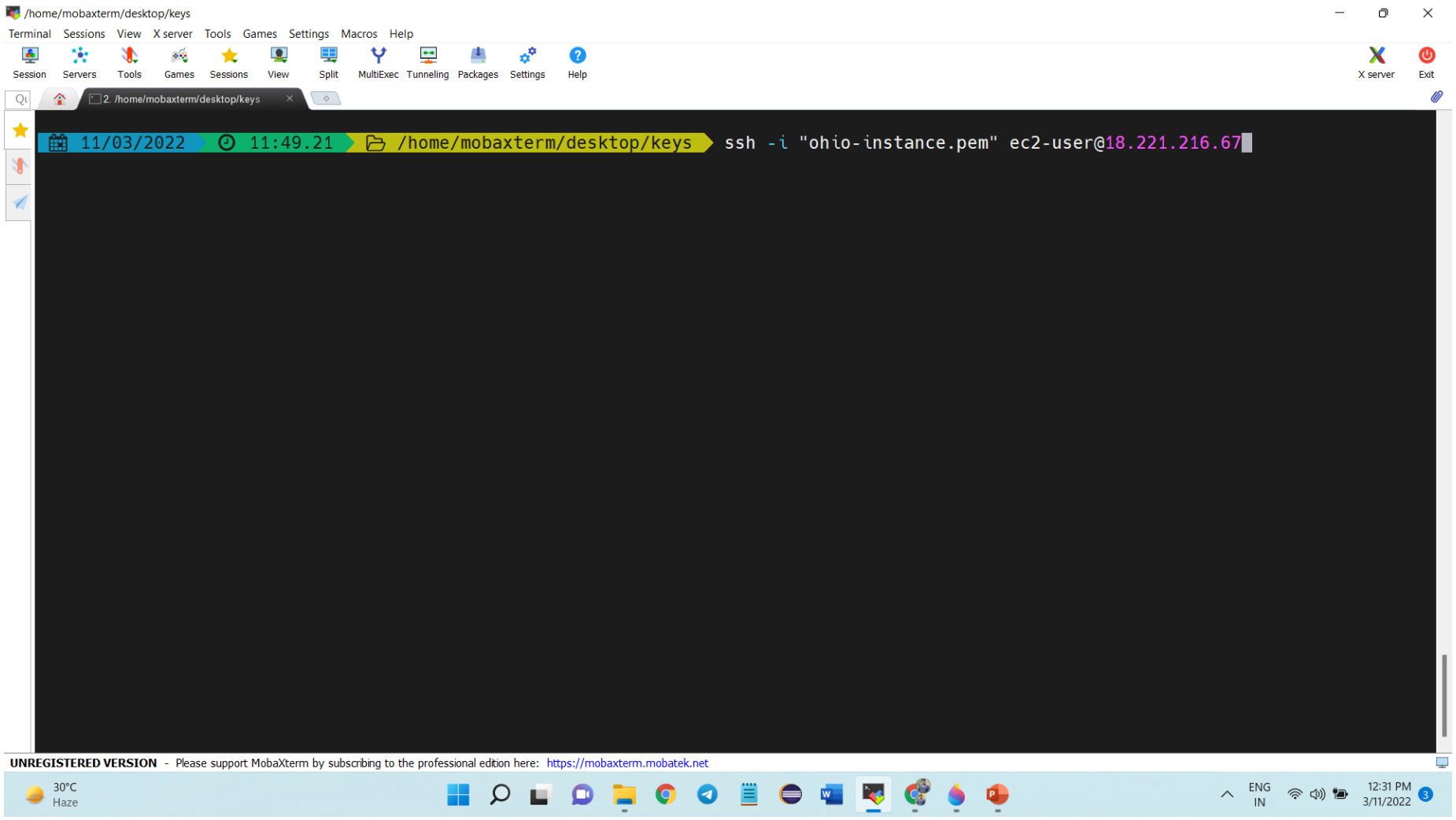
- **Now connect to instance which has Public IP and present in Public Subnet**



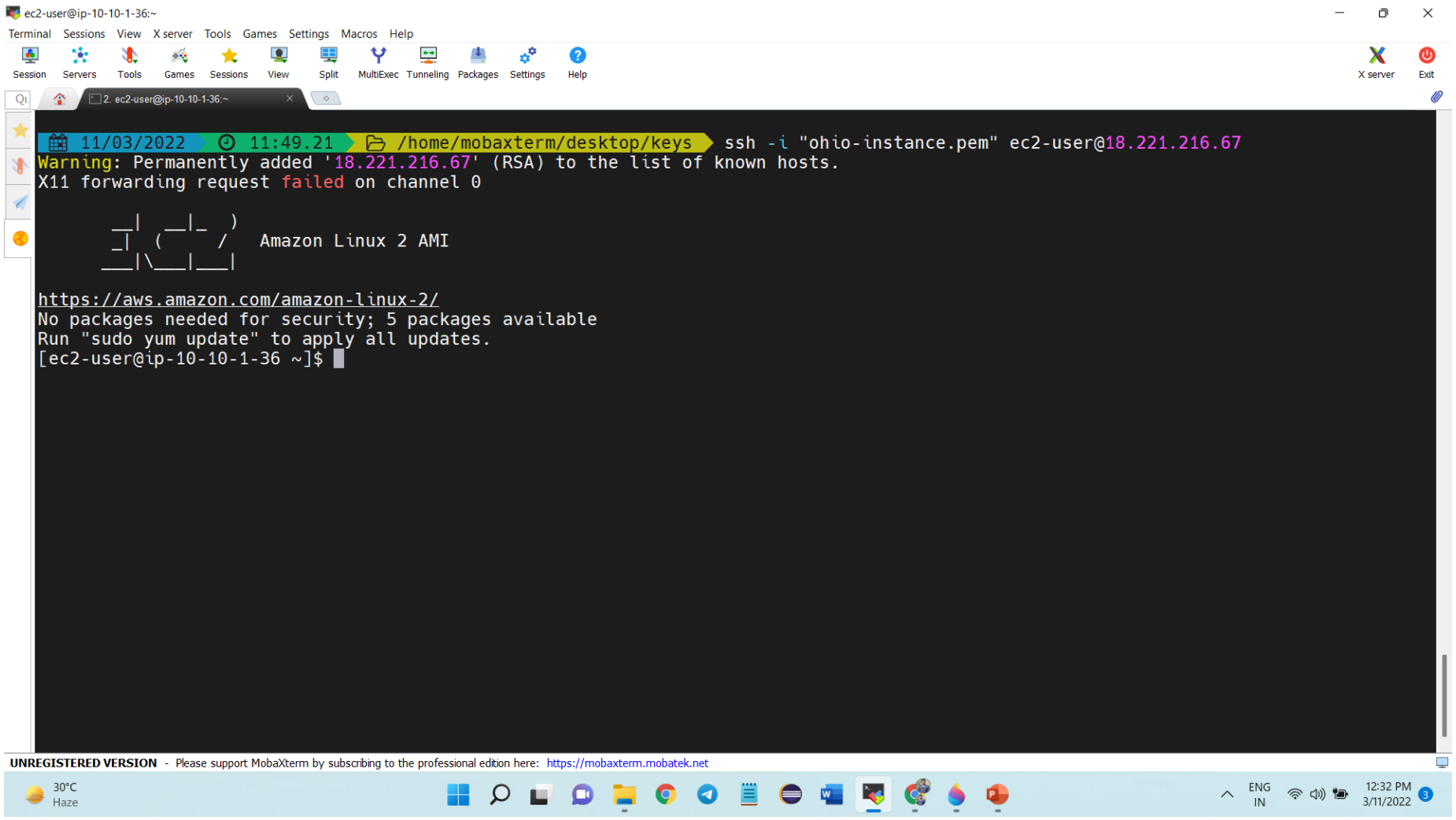
- Select the instance
- Click on Connect



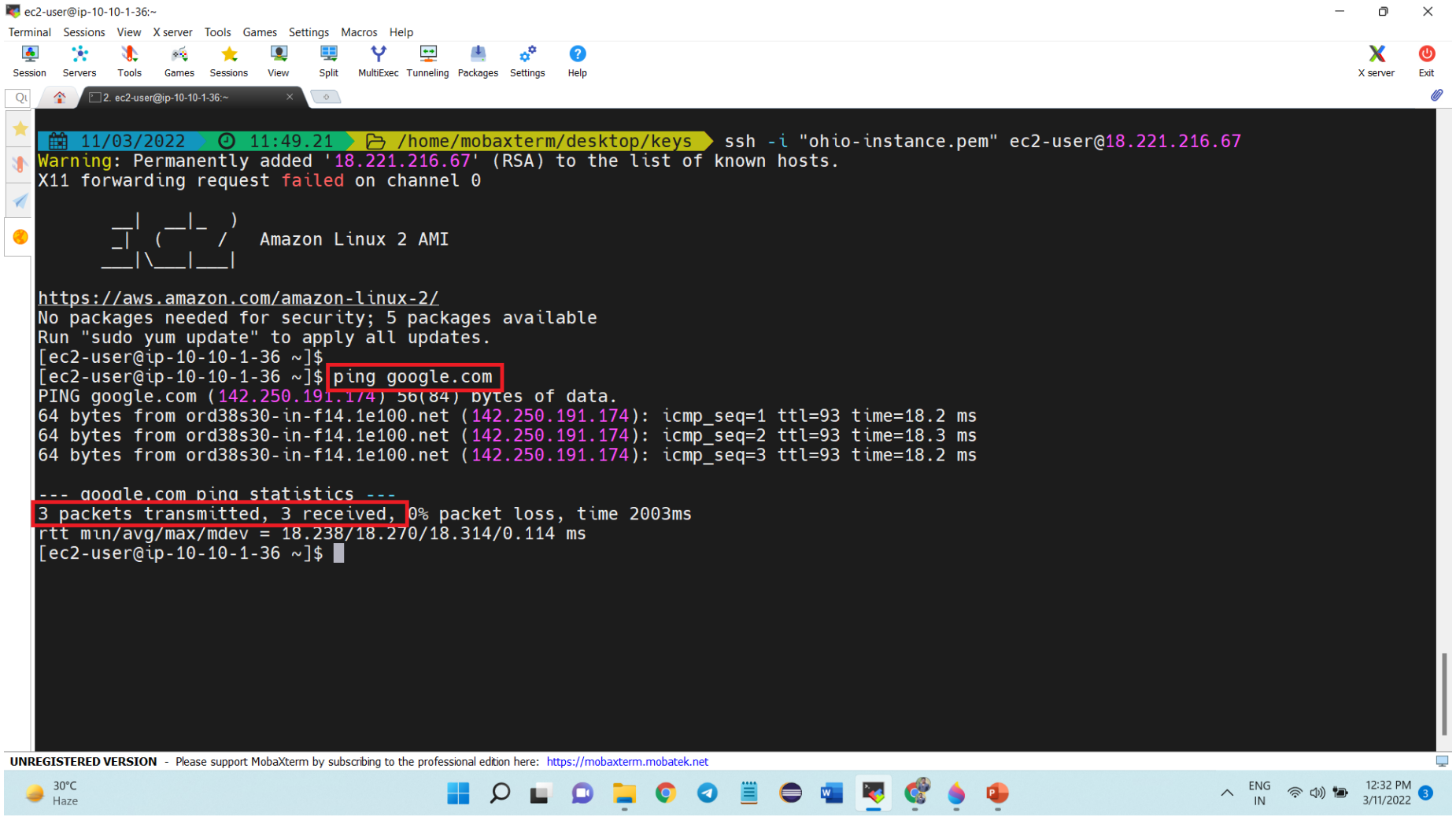
- Copy the Last Command



- Open MobaxTerm
- Paste Command and Press Enter

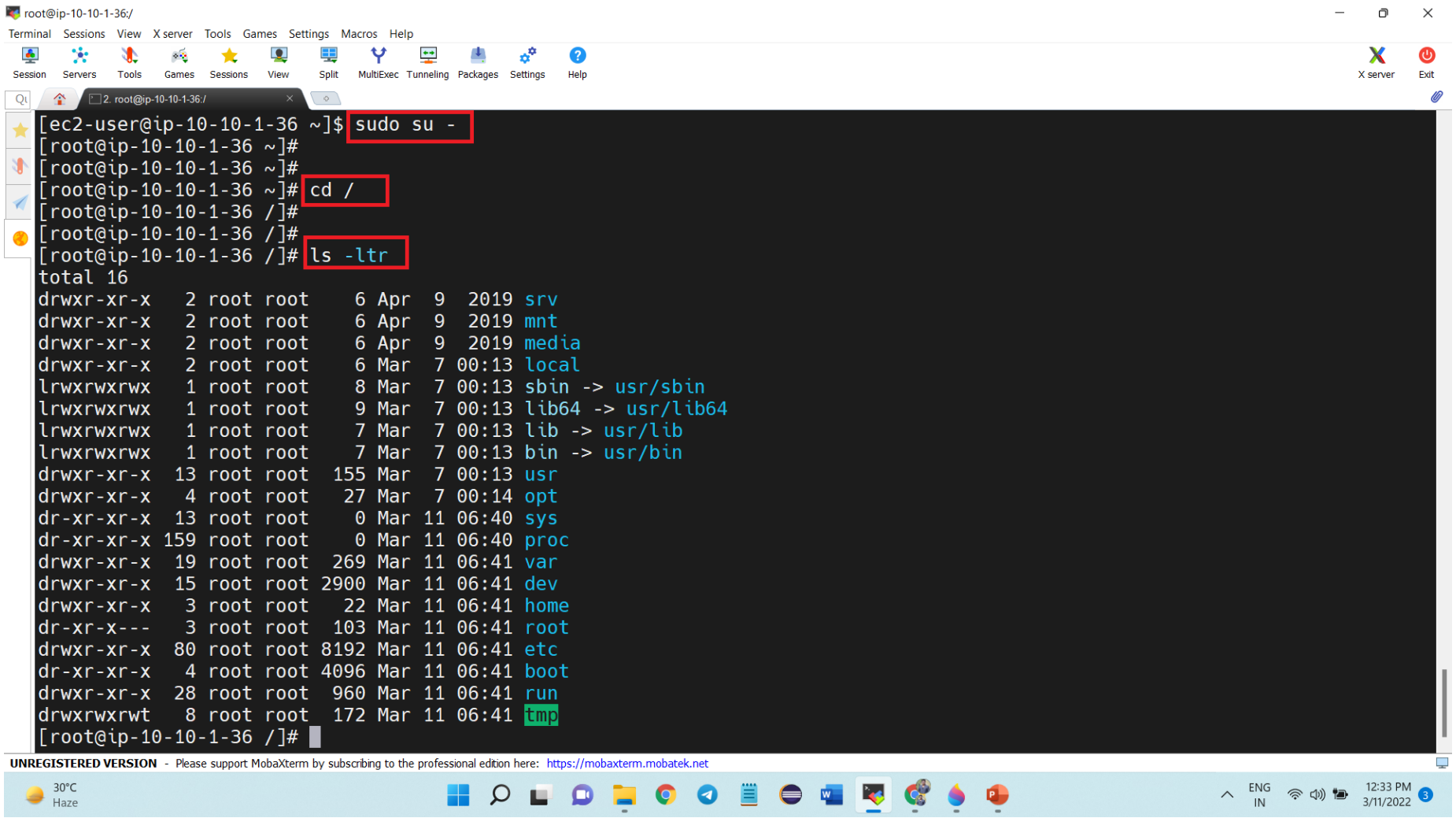


- Here we have connected to our instance.

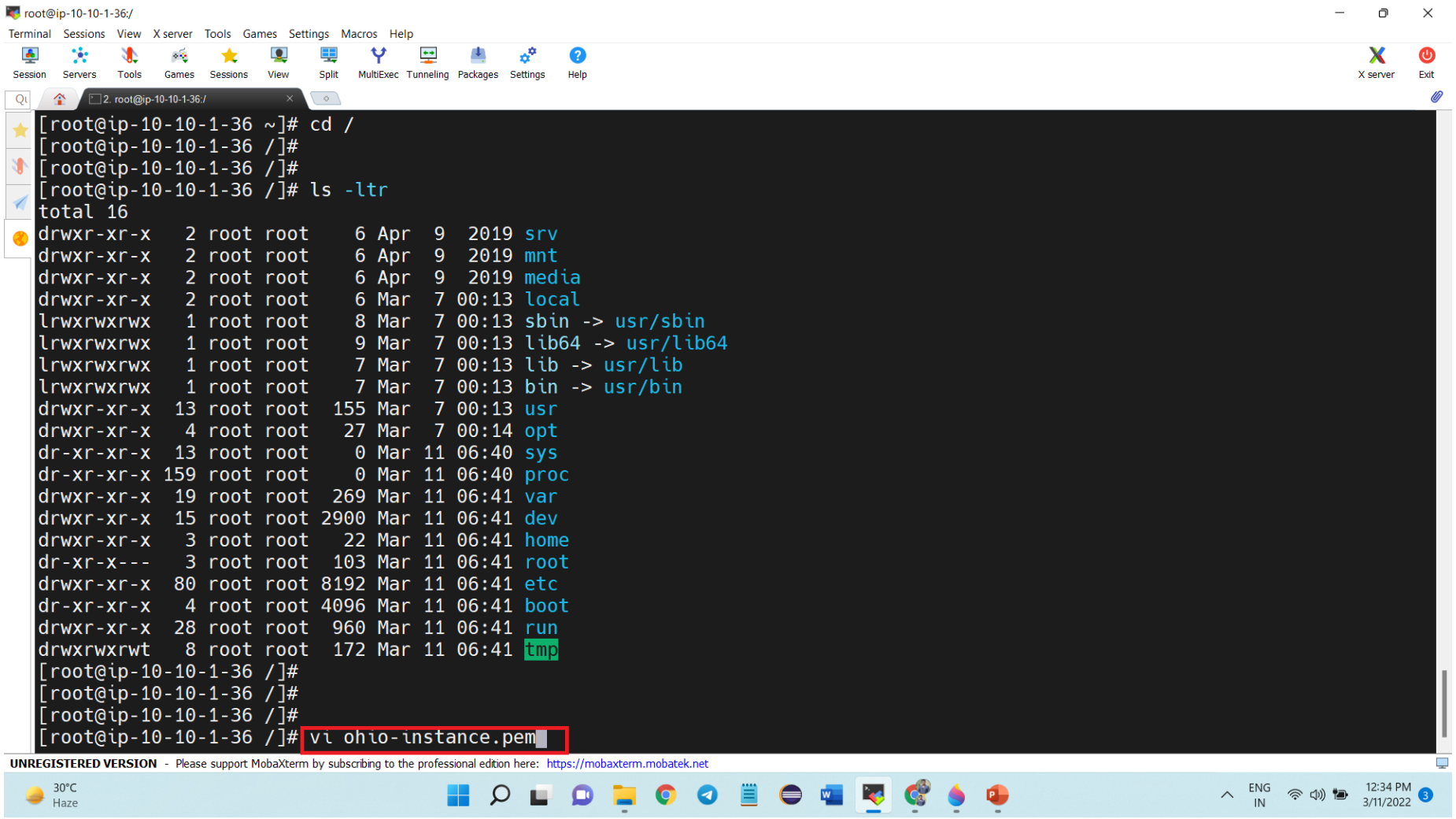


- Use **ping google.com**
- We can able to use ping command because this instance has Public IP and present in Public Subnet.

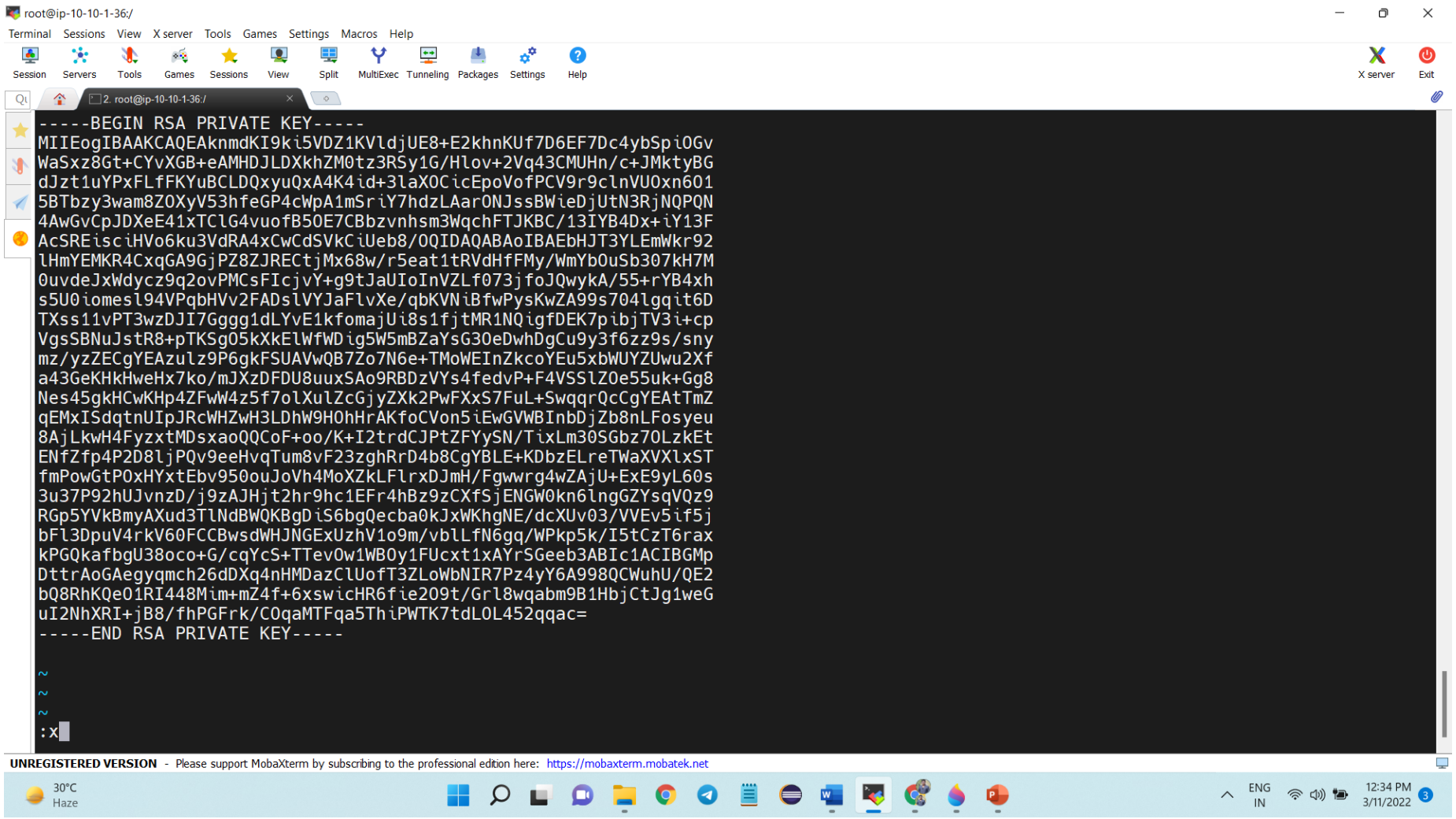
- **Now we will go inside of our second Instance**
- **Which is present in Public Subnet but Disabled Public IP**



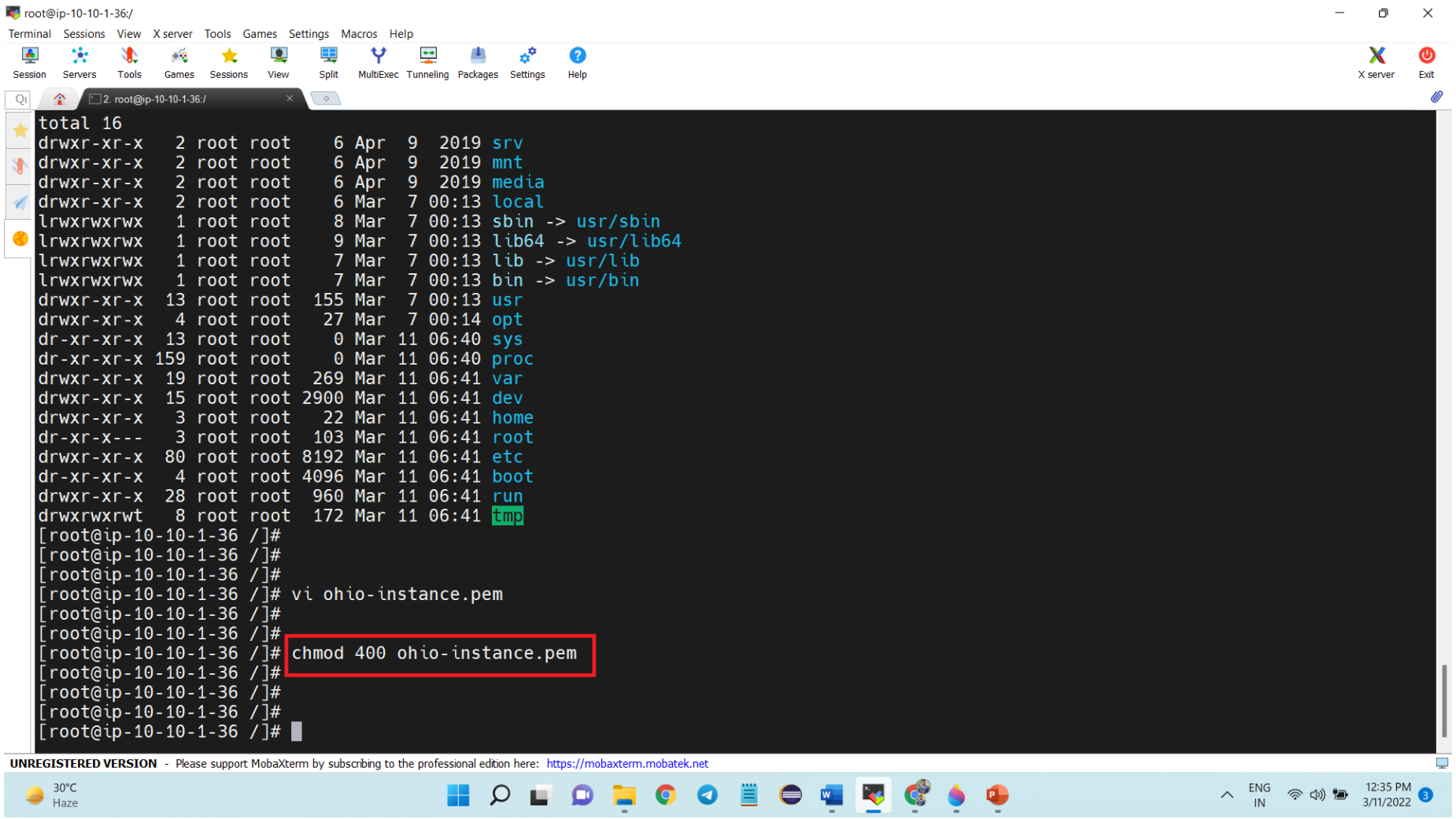
- Use **sudo su –**
- Use **cd /**
- Use **ls -ltr**



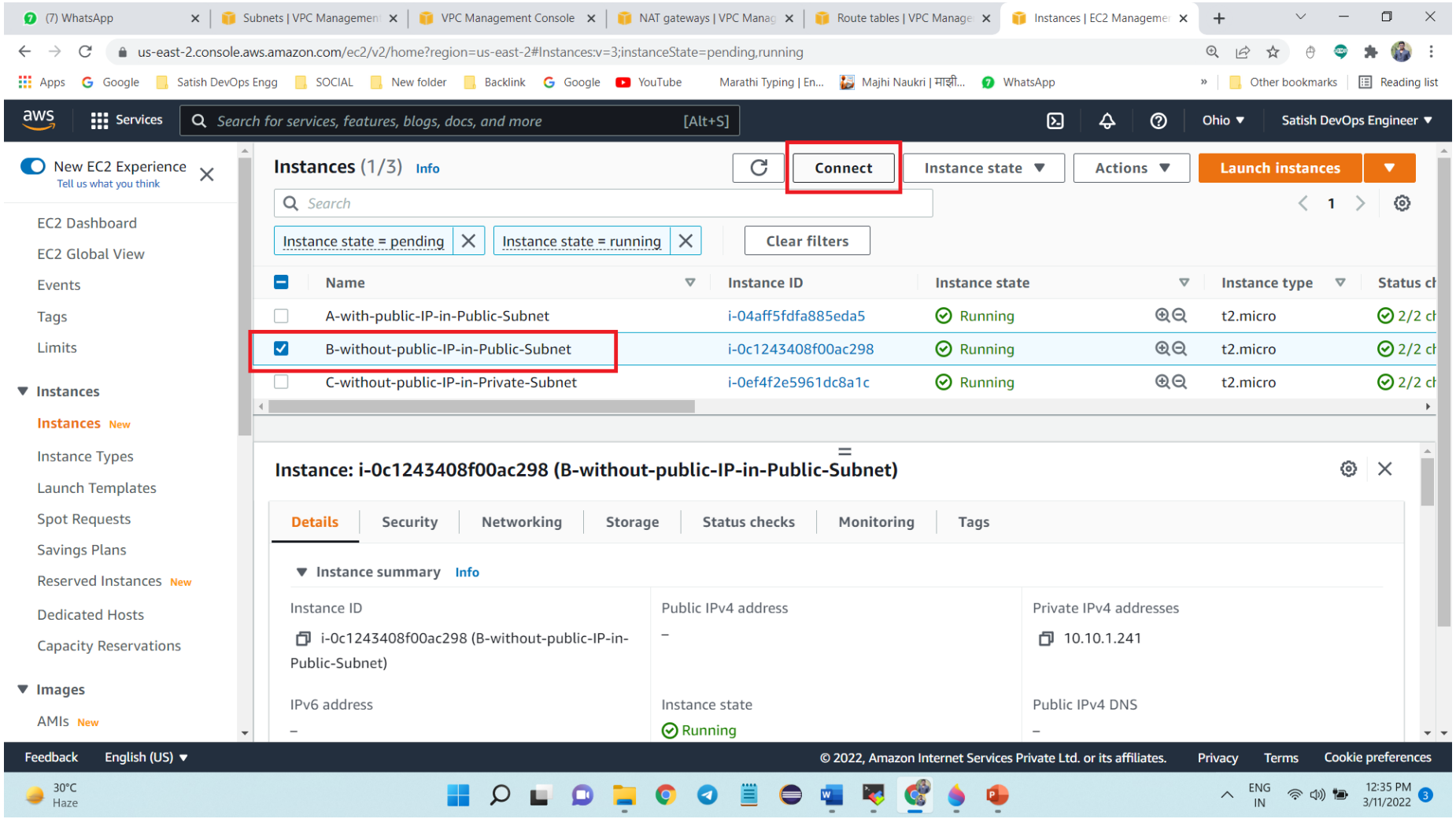
- Use **vi ohio-instance.pem**



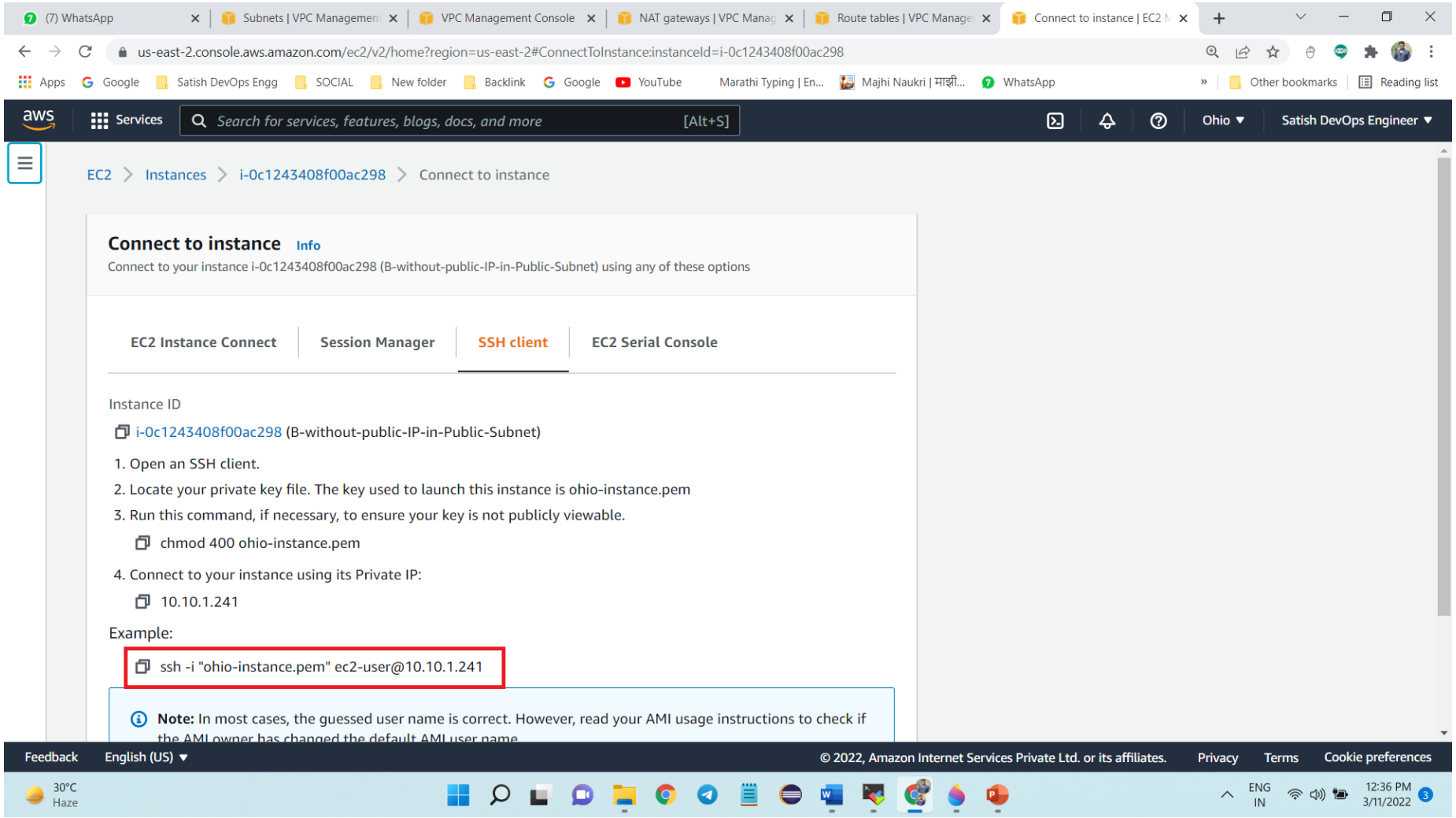
- Paste your data which is in Key Pair



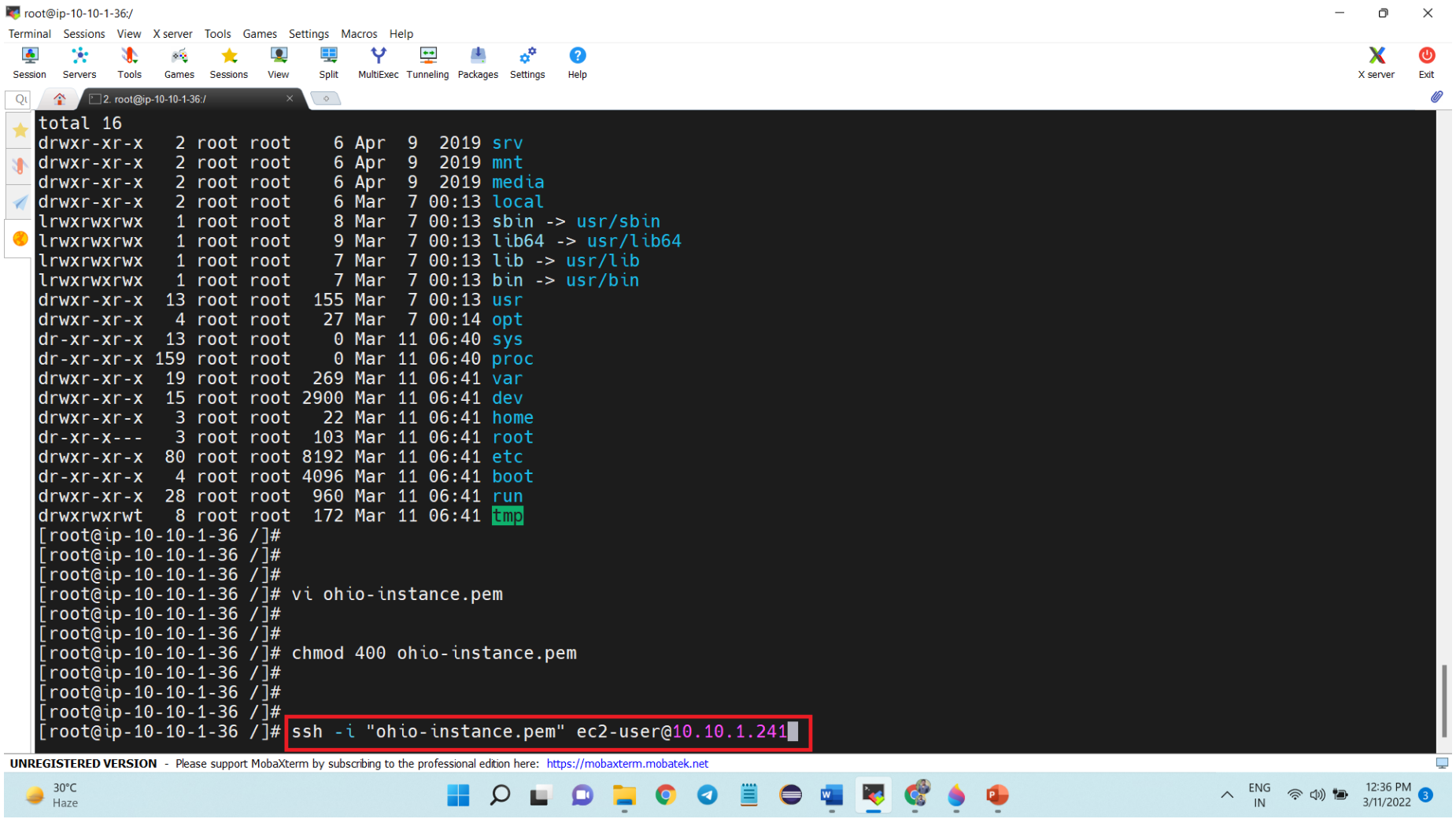
- Use **chmod 400 ohio-instance.pem**



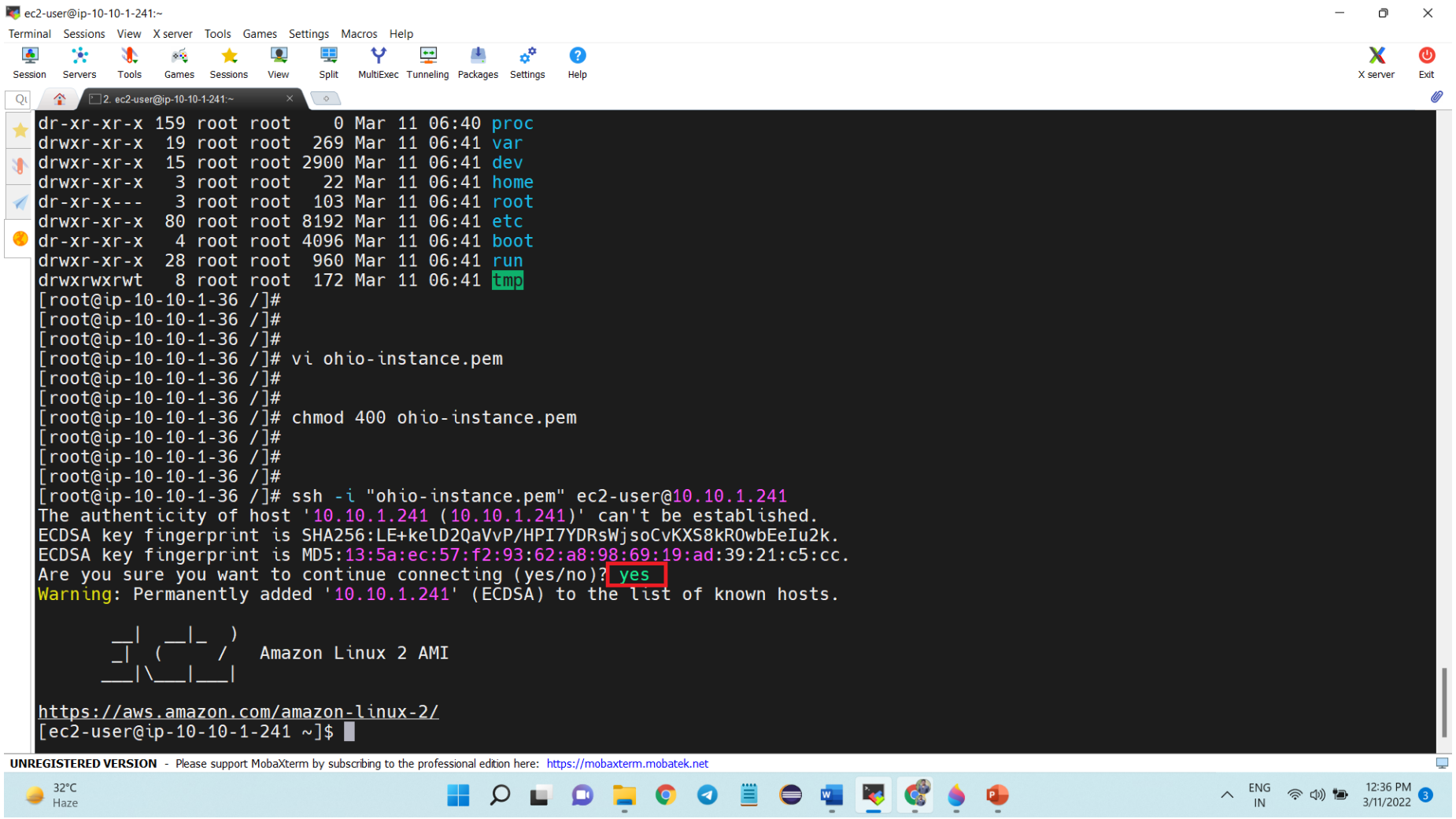
- Select your 2nd instance
- Click on connect



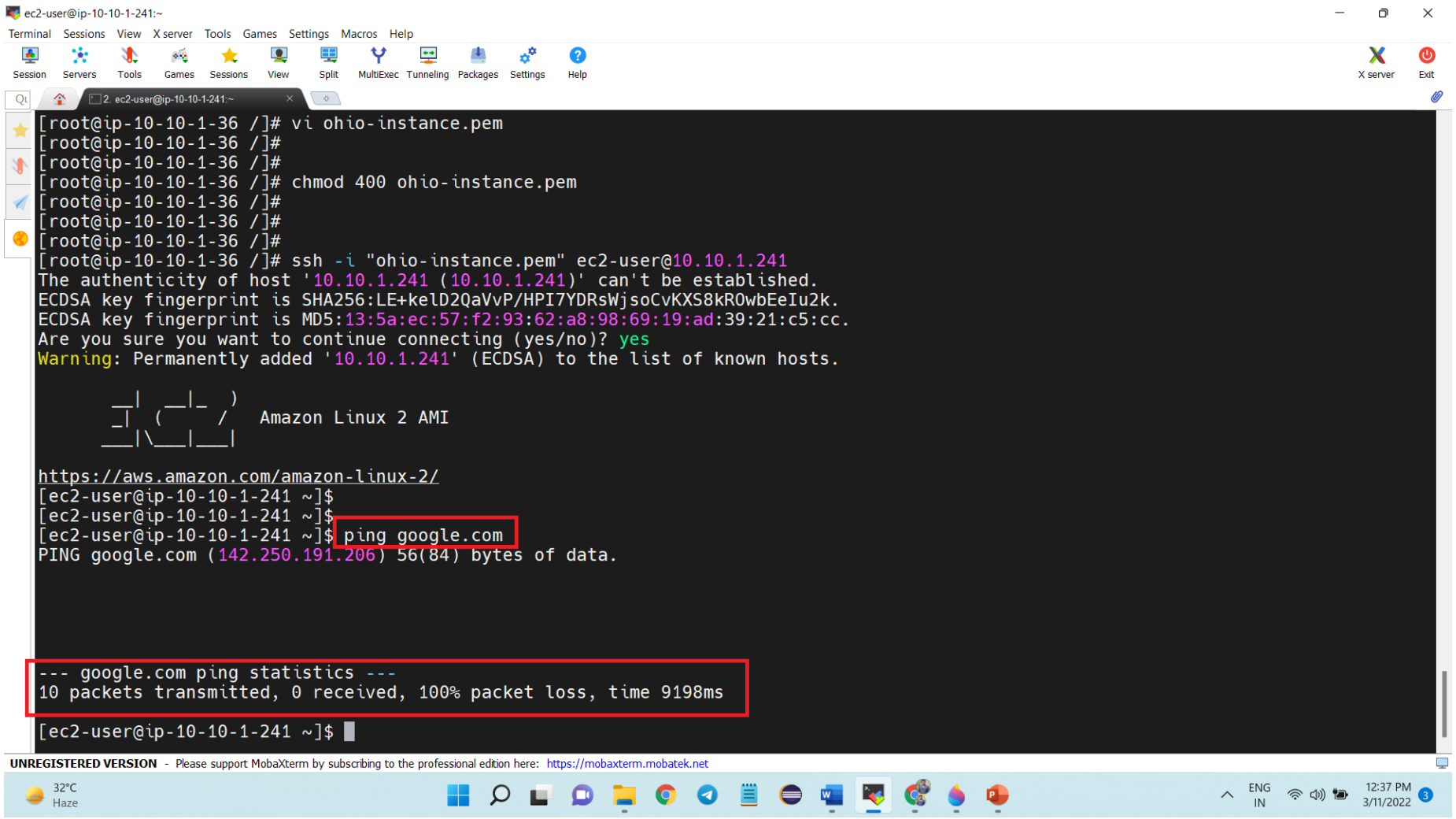
- Copy last command



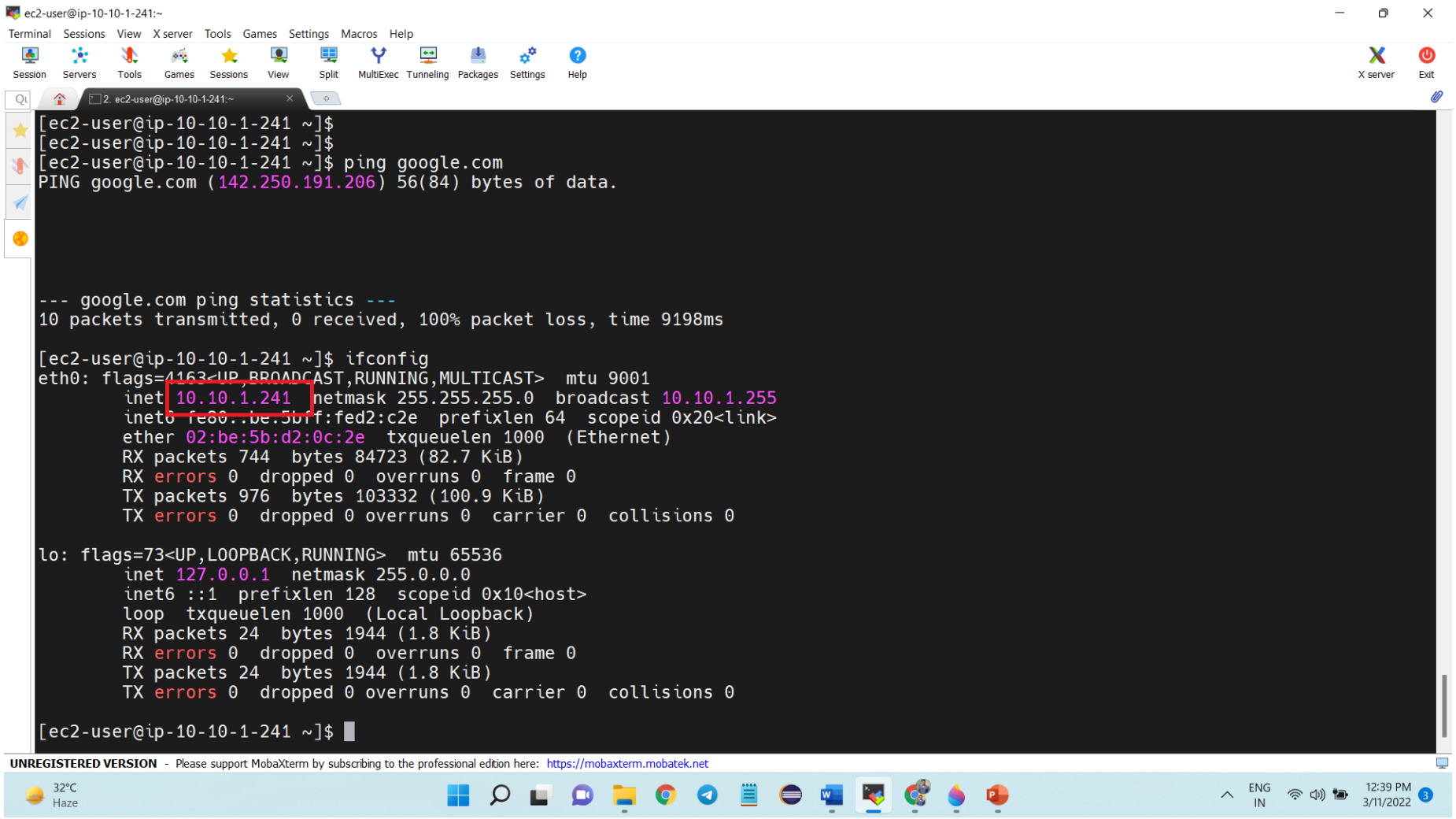
- Paste in MobaxTerm
- Press Enter



- Enter yes
- Press Enter
- Now we are connected to 2nd instance

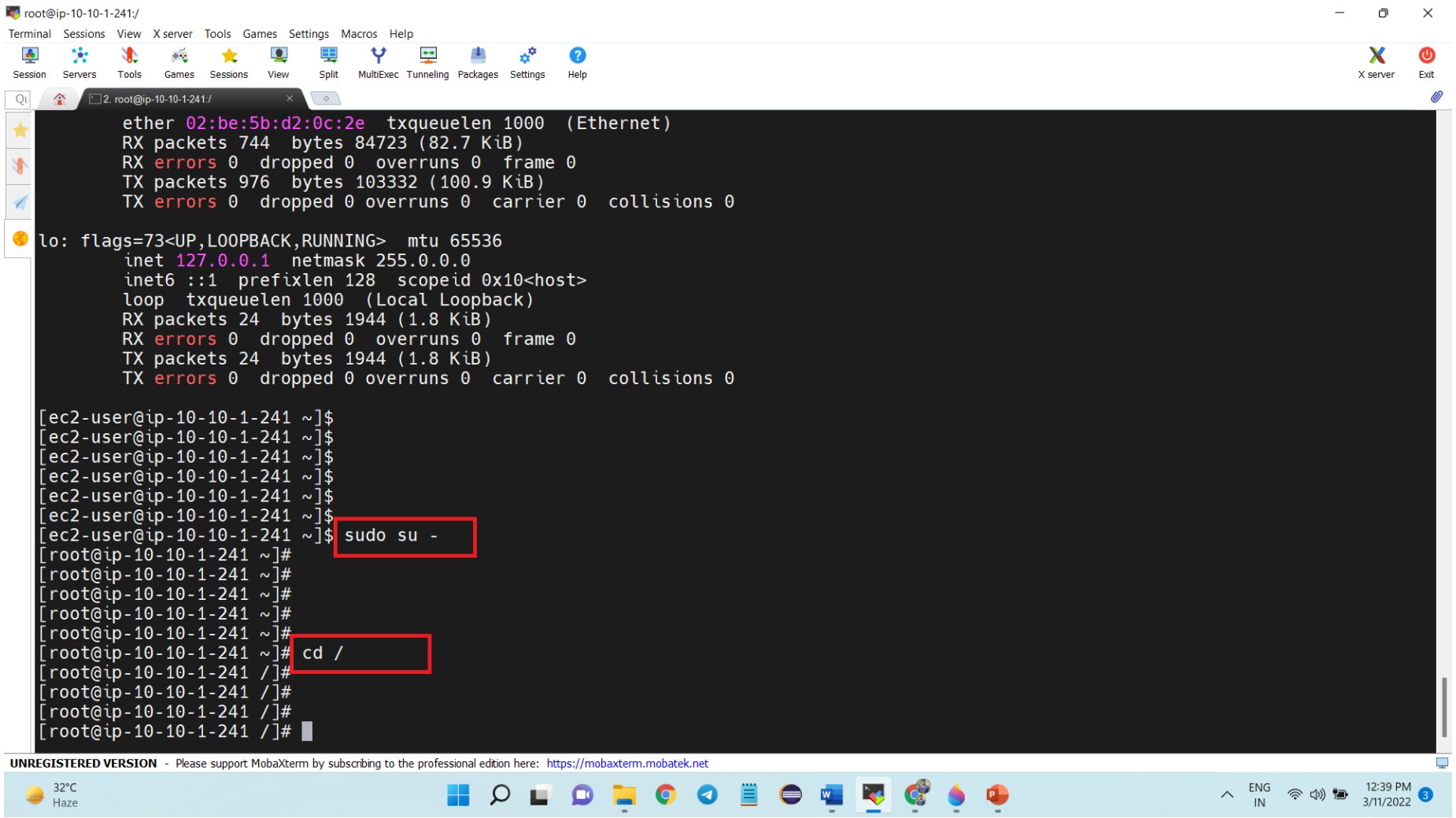


- Use **ping google.com**
- **Here our ping command is not working properly**
- **Because it doesn't have Public IP and also it doesn't present in Public Subnet.**

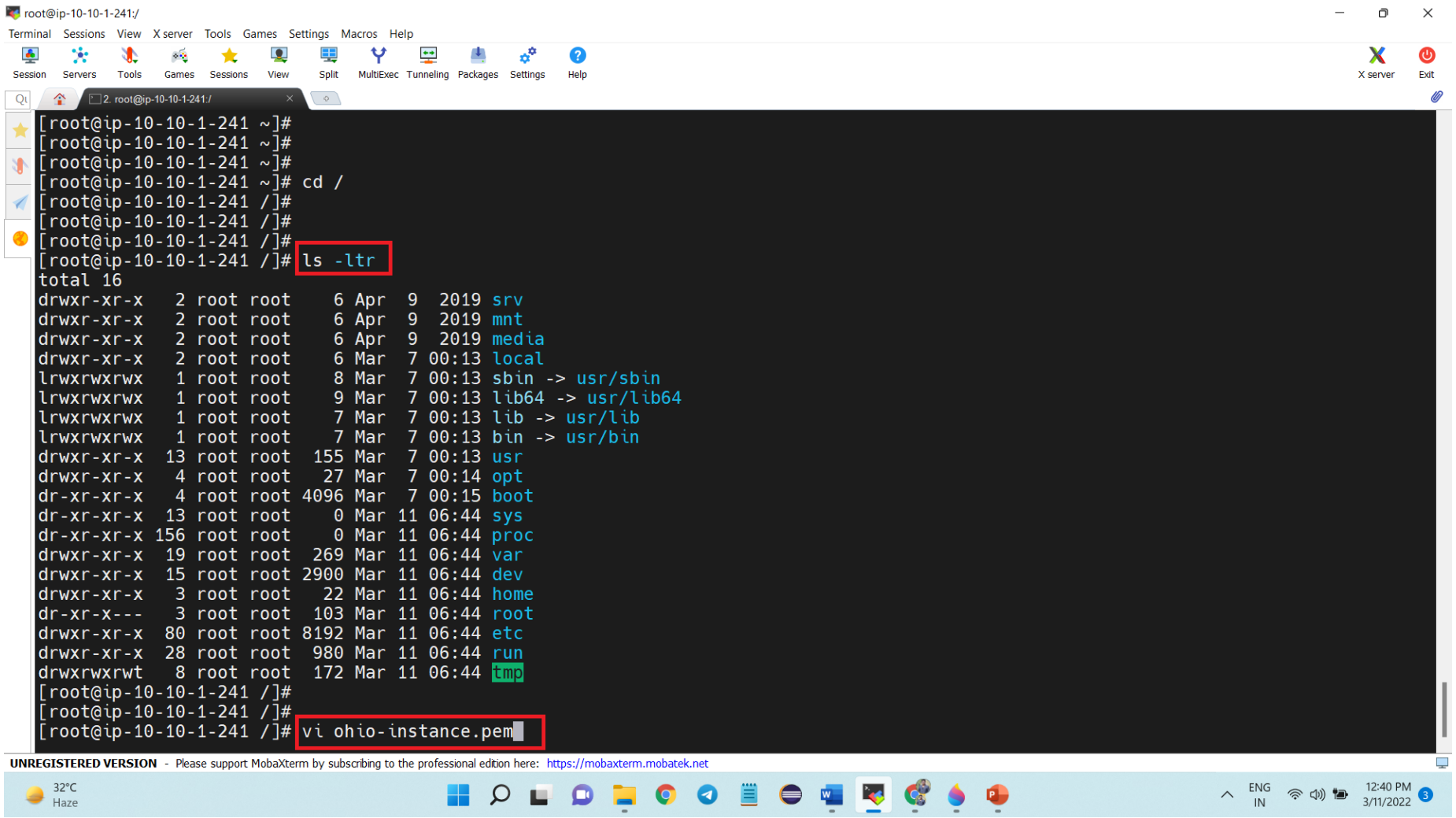


- We can cross check our Private IP as well

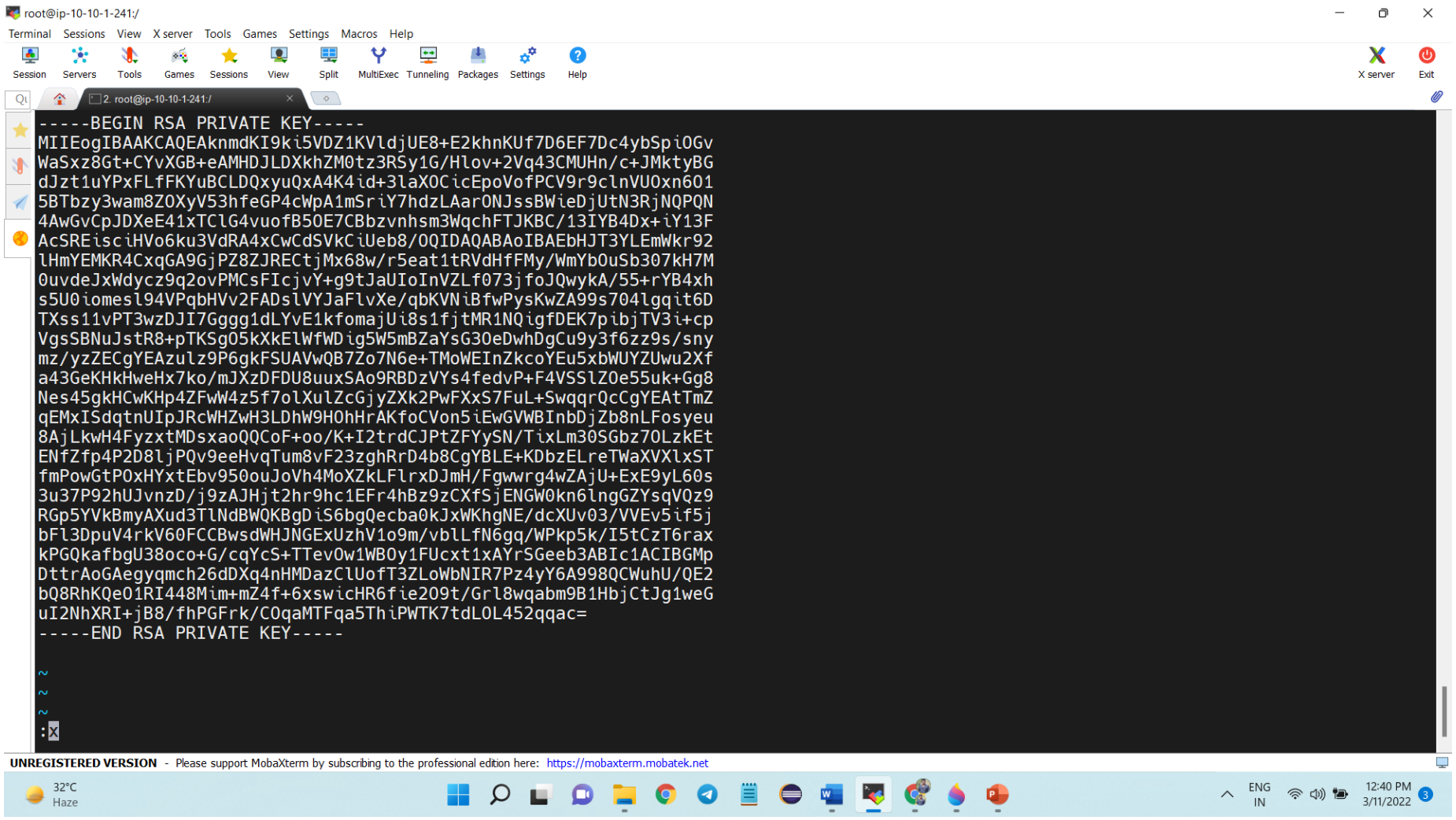
- **Now we will go inside of our third Instance**
- **Which is present in Private Subnet but Disabled Public IP**



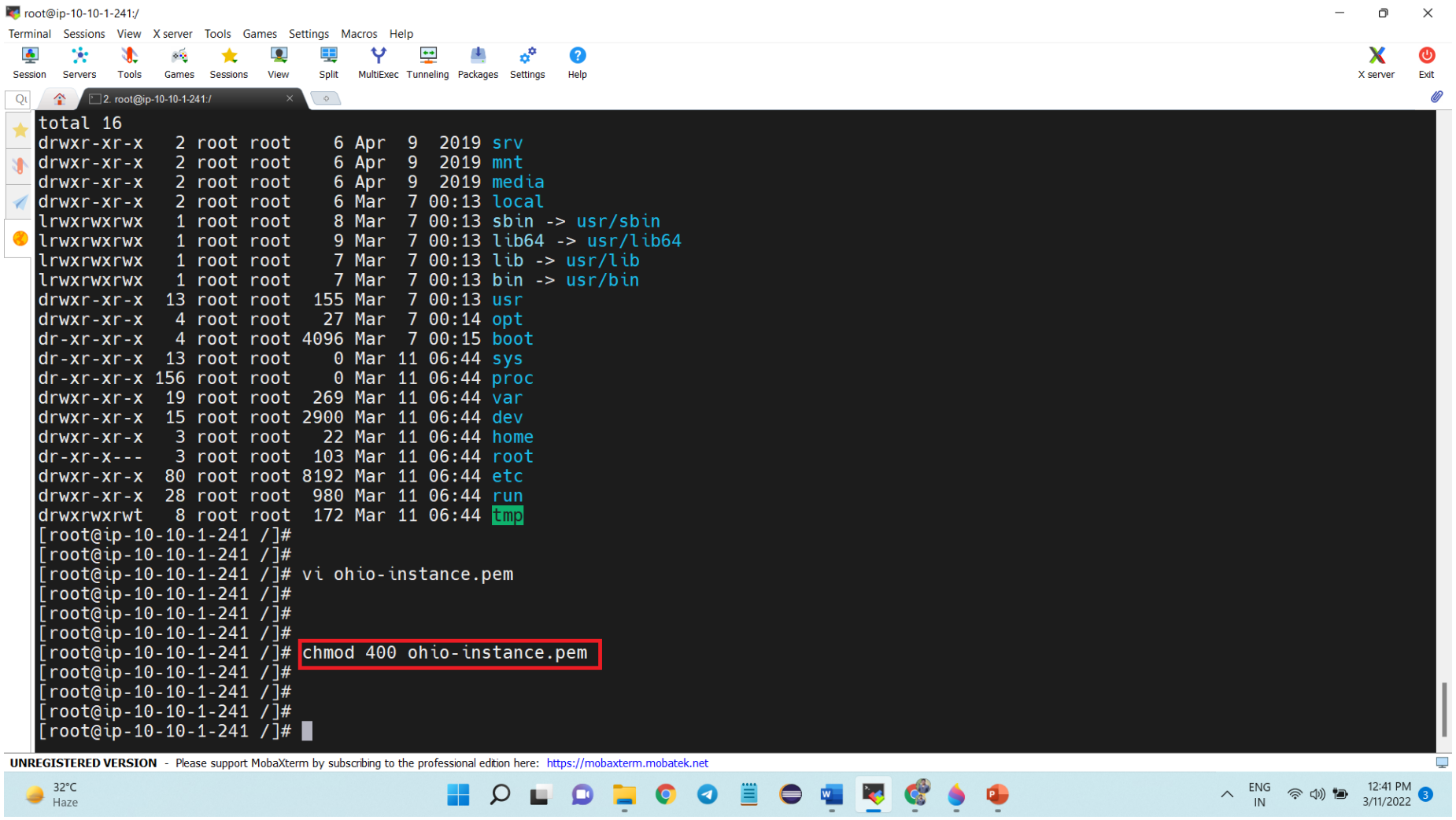
- Use **sudo su -**
- Use **cd /**



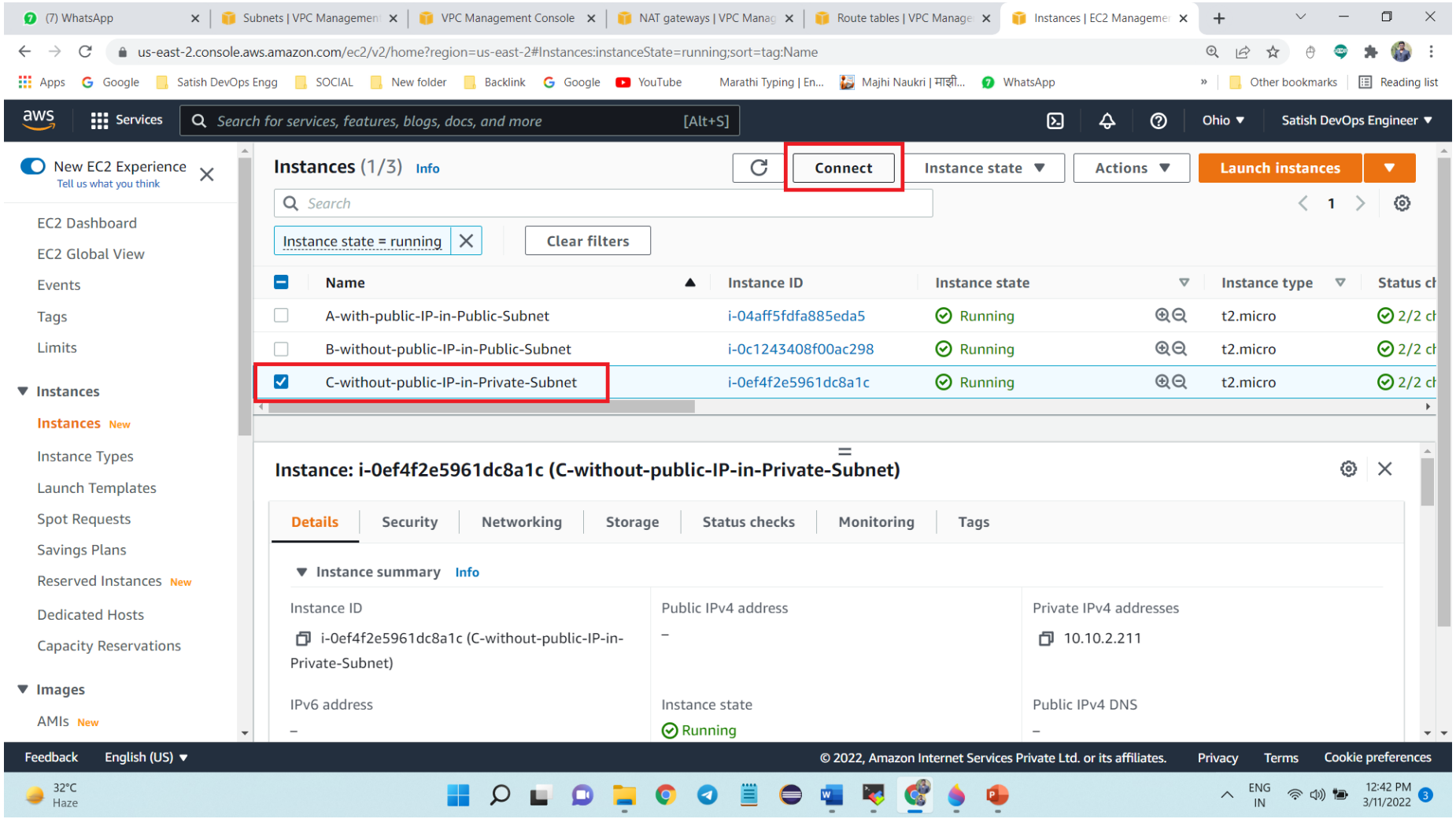
- Use **vi ohio-instance.pem**



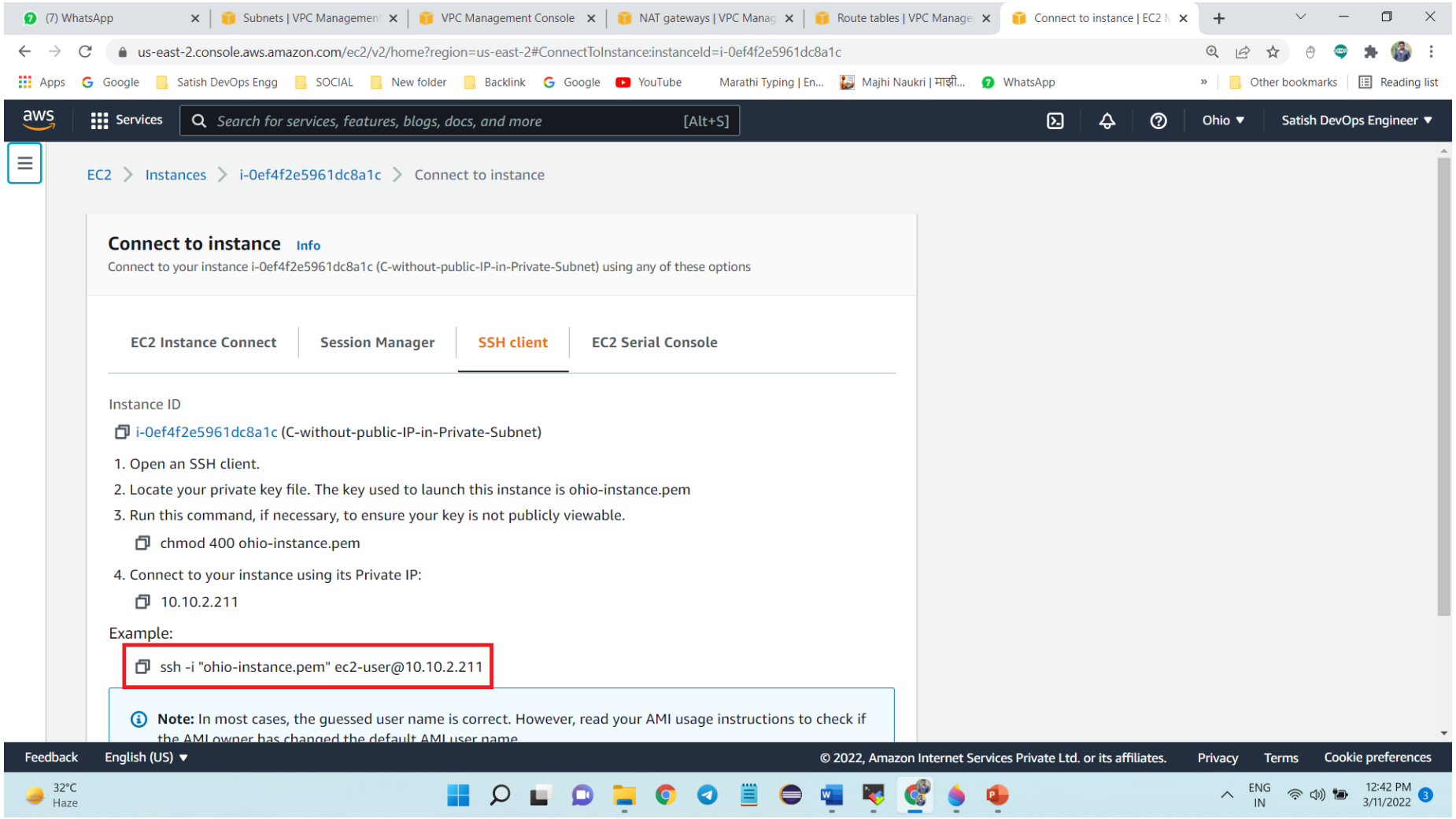
- Paste your key-pair details and save file



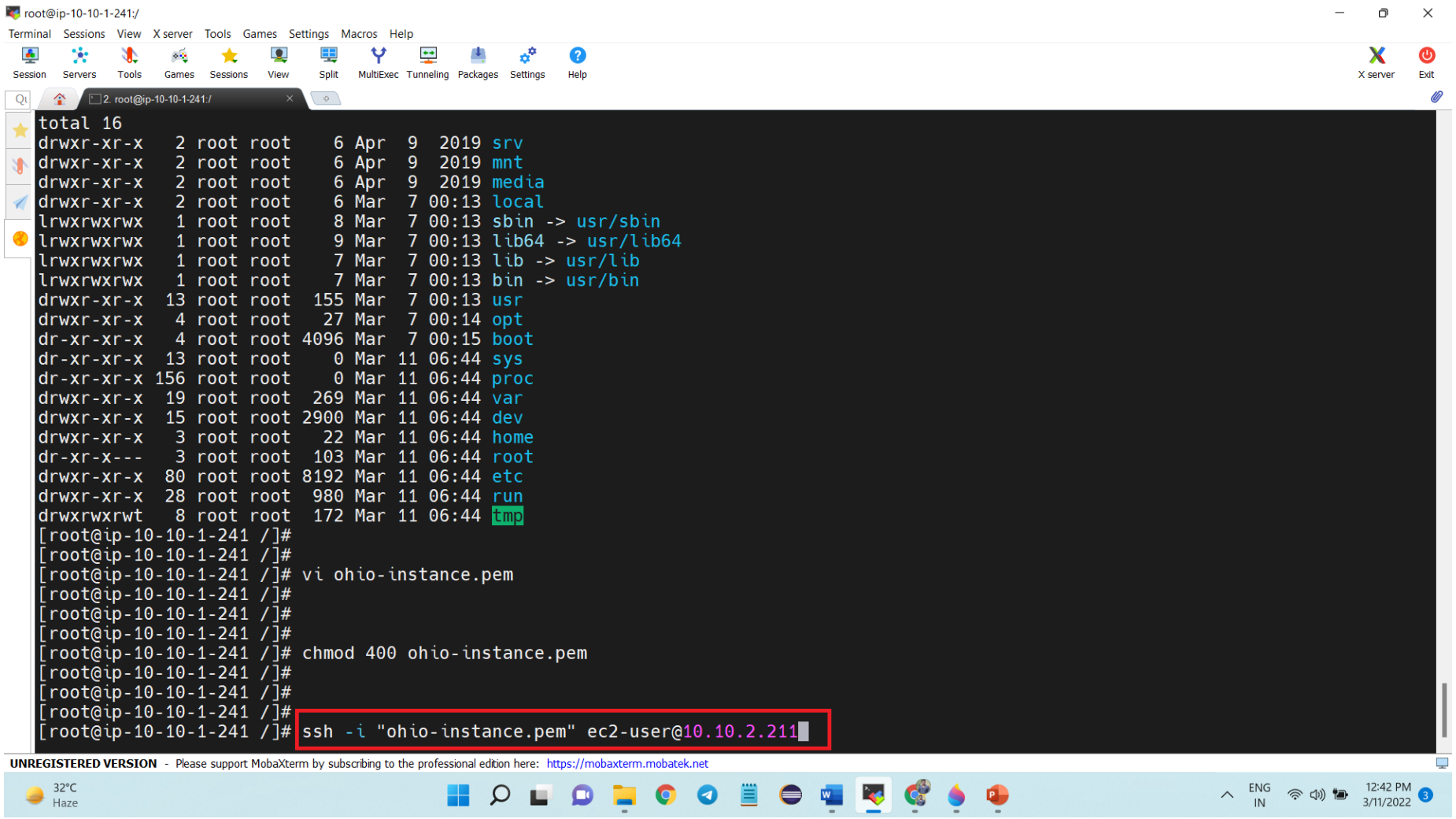
- Use **chmod 400 ohio-instance.pem**



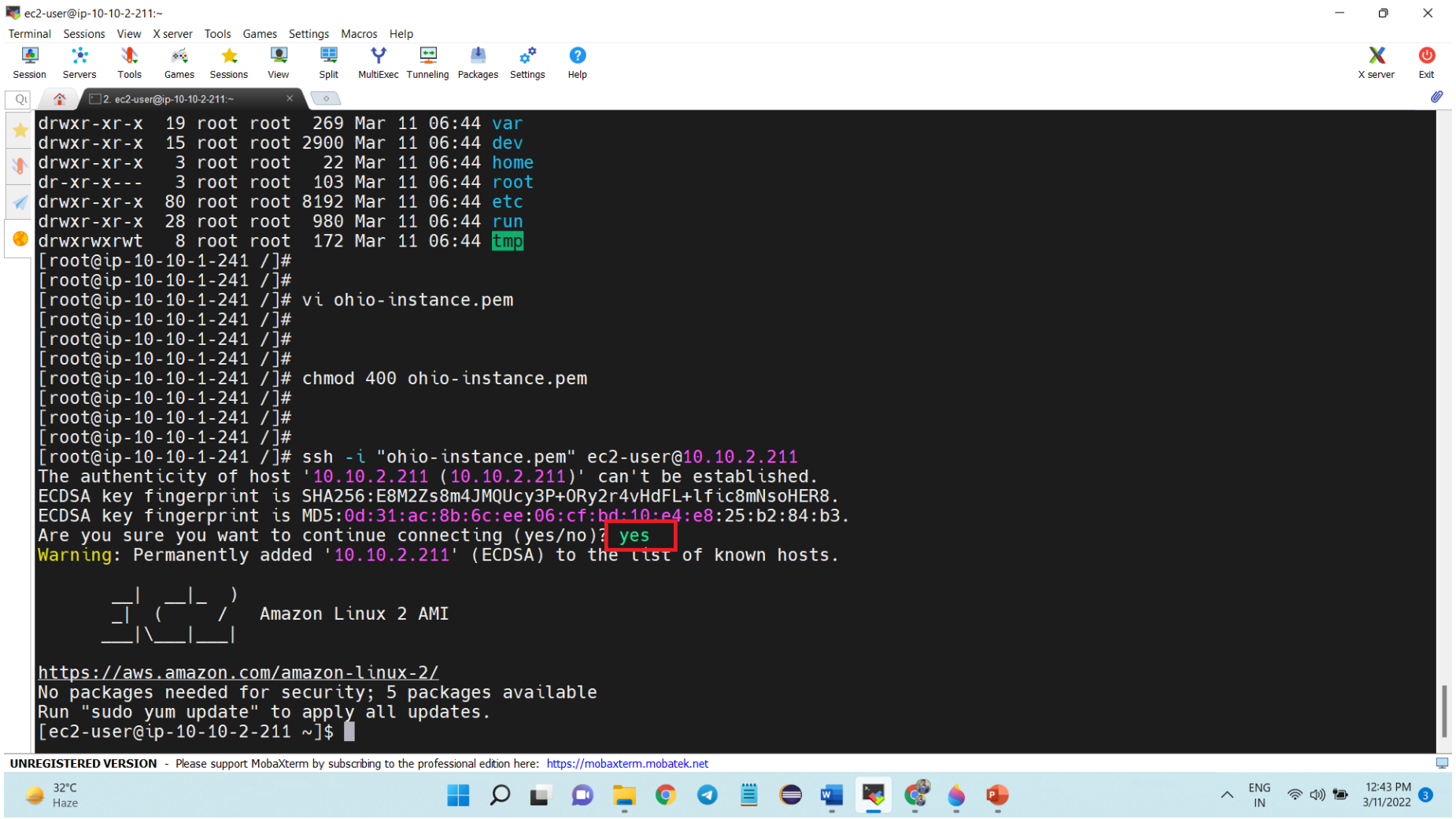
- Select 3rd instance
- Click on Connect



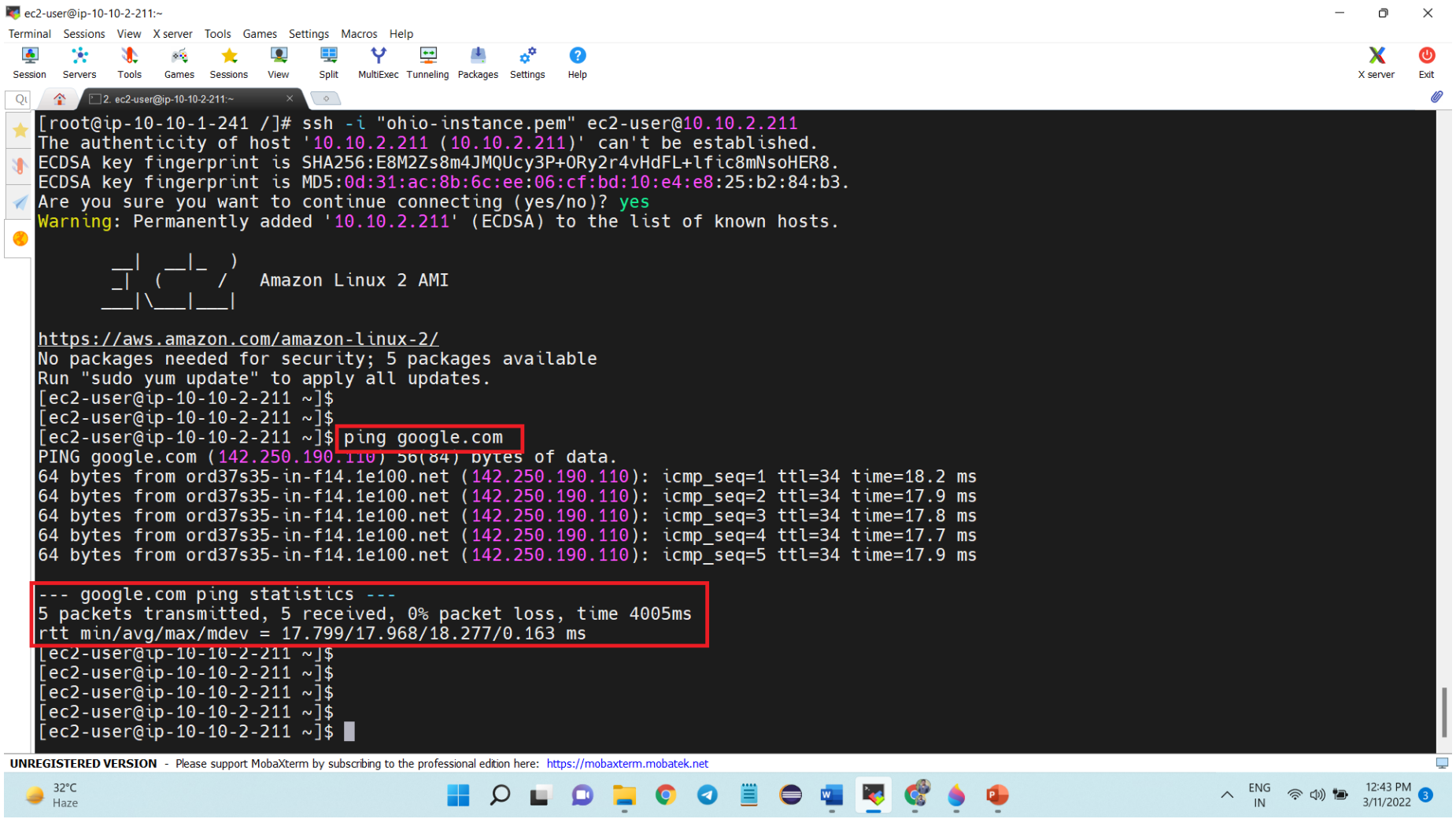
- Copy last command



- Paste in MobaxTerm and Press Enter



- Enter yes and press enter
- Here you can see that you are connected to 3rd instance.



- Use **ping google.com**
- **Here we can see that our ping command is running fine because this instance is in private subnet and this private subnet is connected to NAT Gateway.**

- So here we have seen that working of **Internet gateway** in Public Subnet.
- Also we have seen that working of **NAT gateway** in Private Subnet.
- All the working is totally depend upon Route Table entry and Subnet Associations with respect to particular **Gateway**.
- Also while creating VPC and Subnet enter **CIDR** correctly.

Thanks for Doing Practical with Us