Kubernetes & VPC

RAJESH JAGARLAMUDI
rajesh@srishsoft.com , 9948867777
SRISH TECHNOLOGIES, KPHB, Beside Arjun Theater

Introduction

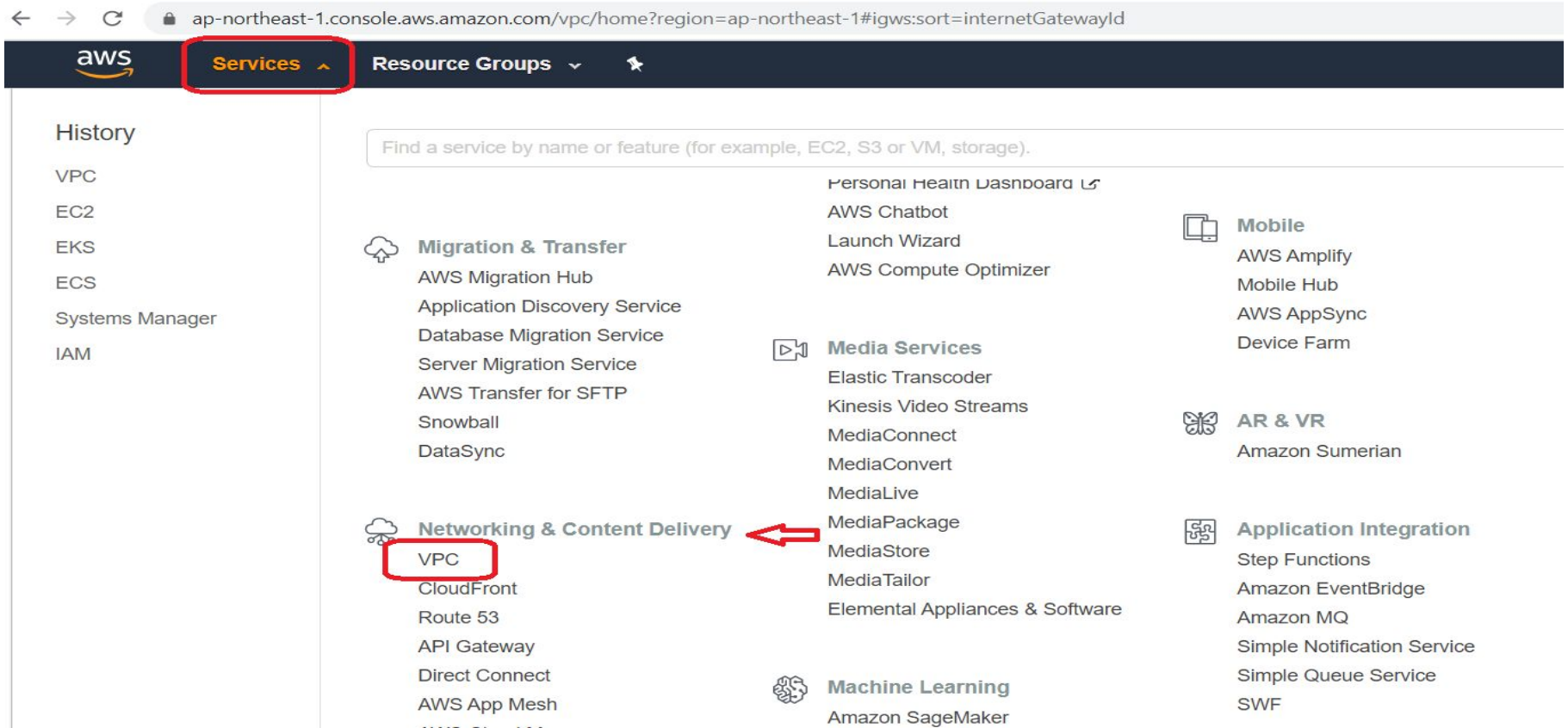
Topics to be discussed

- Basics
- VPC - Subnets - Routing table - Internet Gateway
- Baston - Host configuration (configuring AWS CLI)
- Secure Systems Manager
- Kubernetes installation at Client
- Configuring EKS (Elastic Kubernetes Service)
- Managing Nodes
- Pod’s - Templates - Services with examples
- Service types (ClusterIP, Node IP, Load Balancer)
- Integrating below as Project

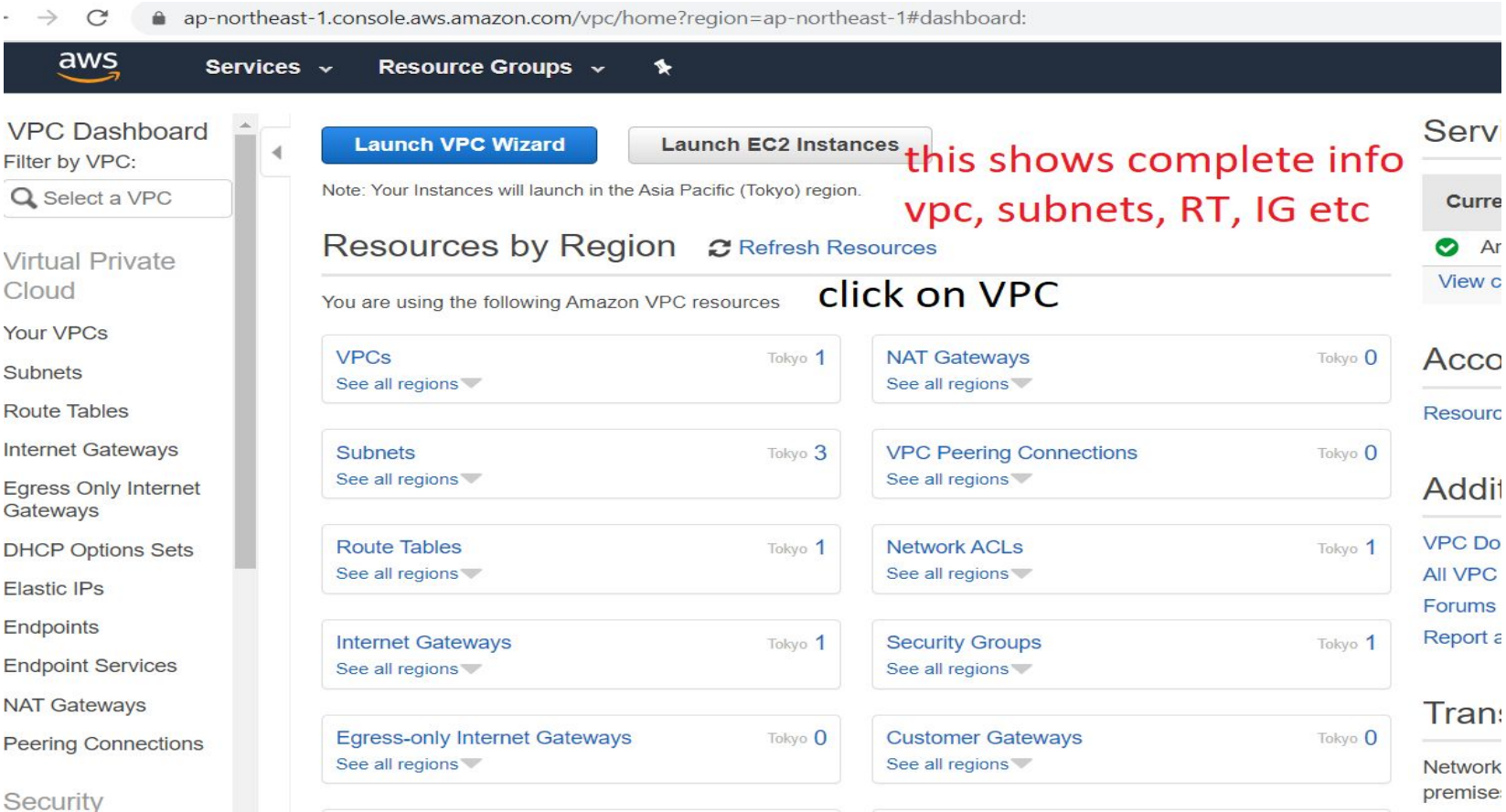
GITLAB - JENKINS - ECR - EKS (DOCKER - CONTAINERS)

VPC - Subnets - Route Tables - Internet Gateway

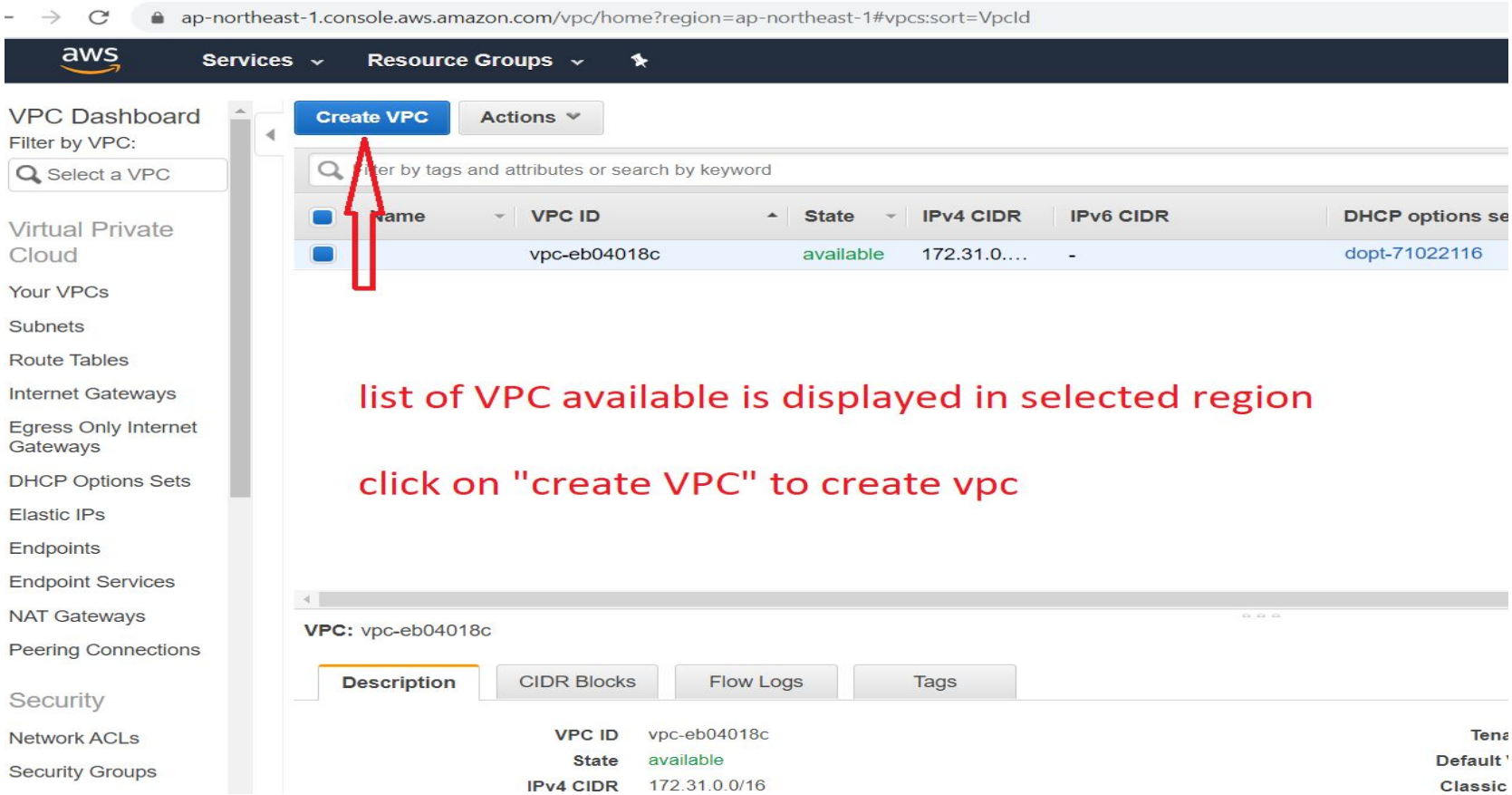
Login to aws console → Services → Networking & Content Delivery → VPC



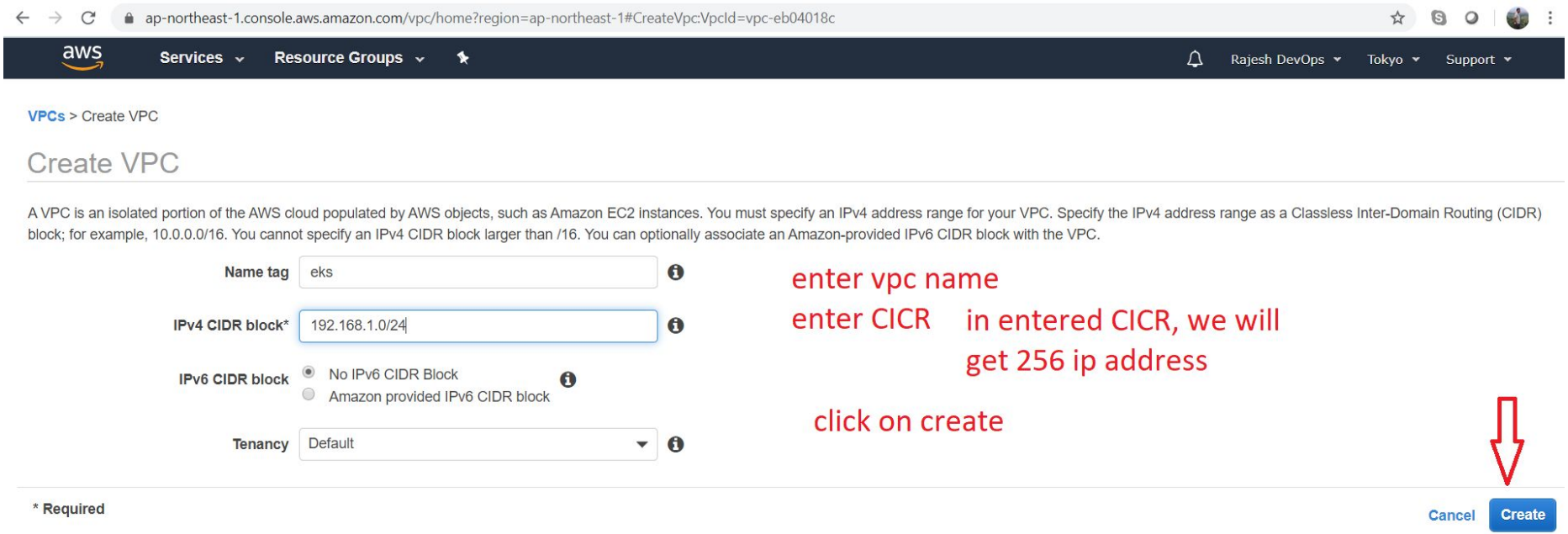
VPC - Subnets - Route Tables - Internet Gateway



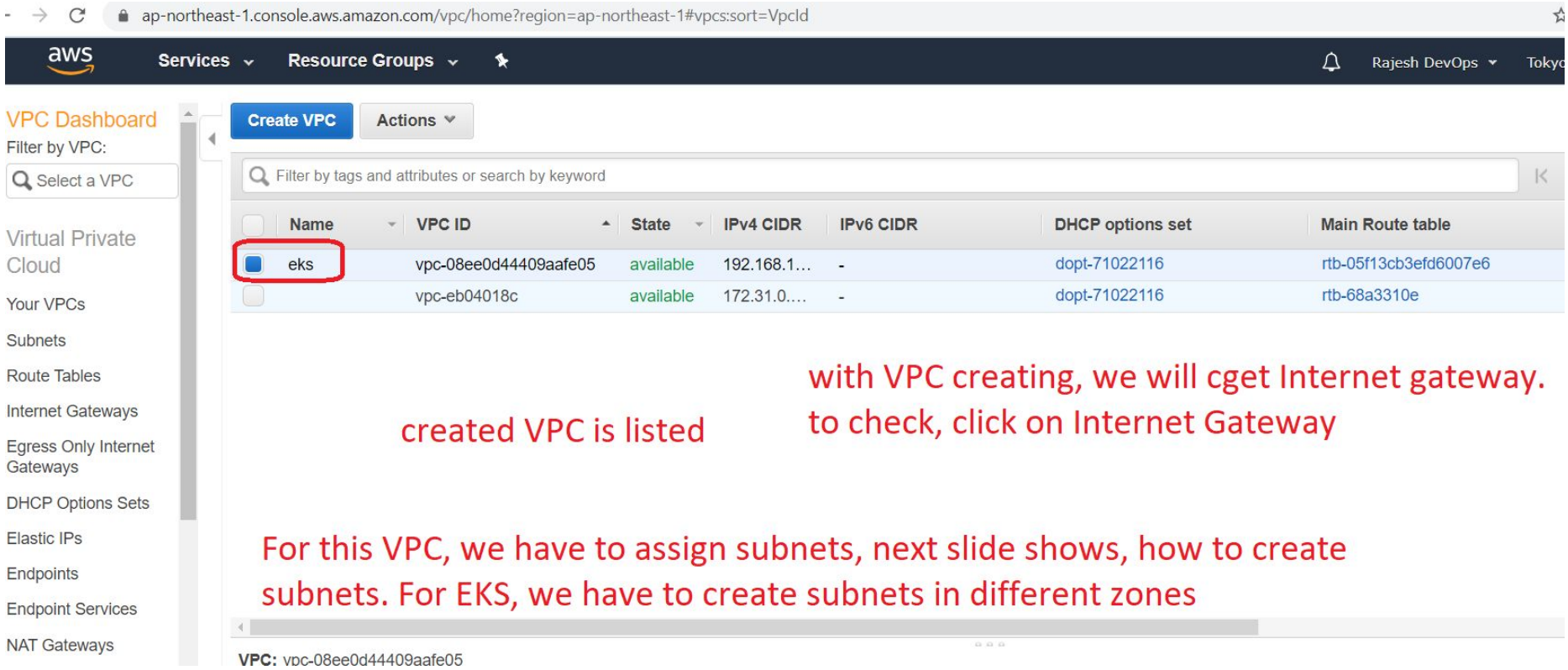
VPC - Subnets - Route Tables - Internet Gateway



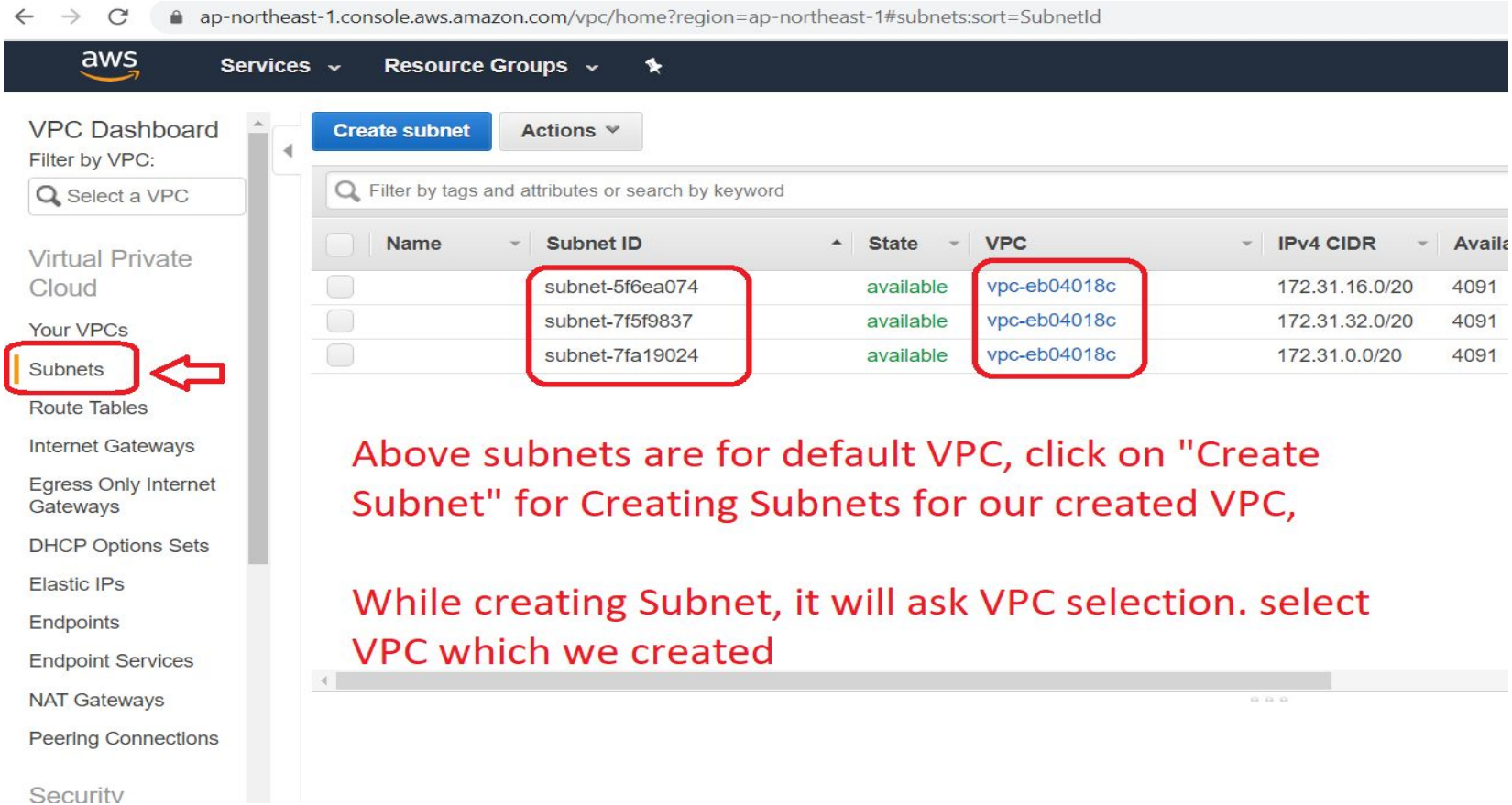
VPC - Subnets - Route Tables - Internet Gateway



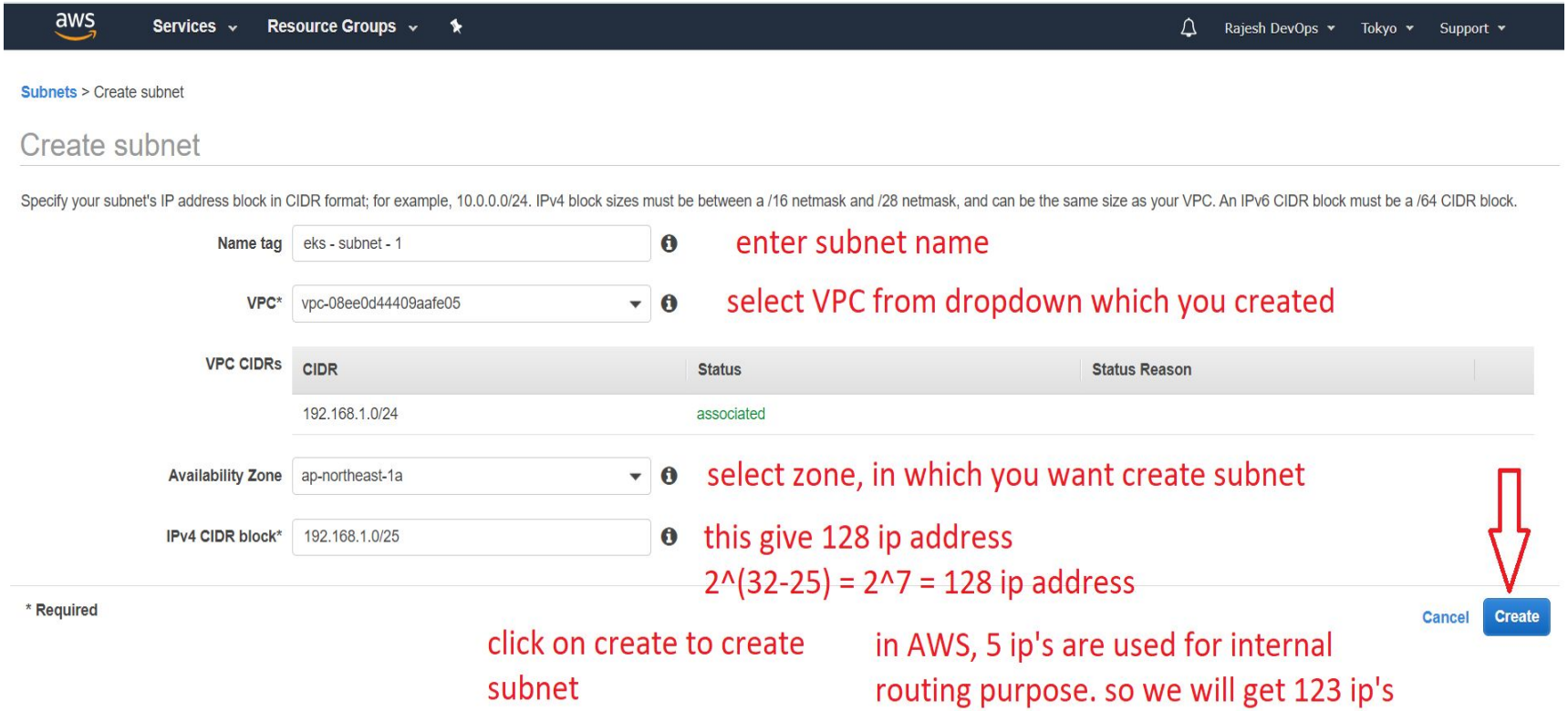
VPC - Subnets - Route Tables - Internet Gateway



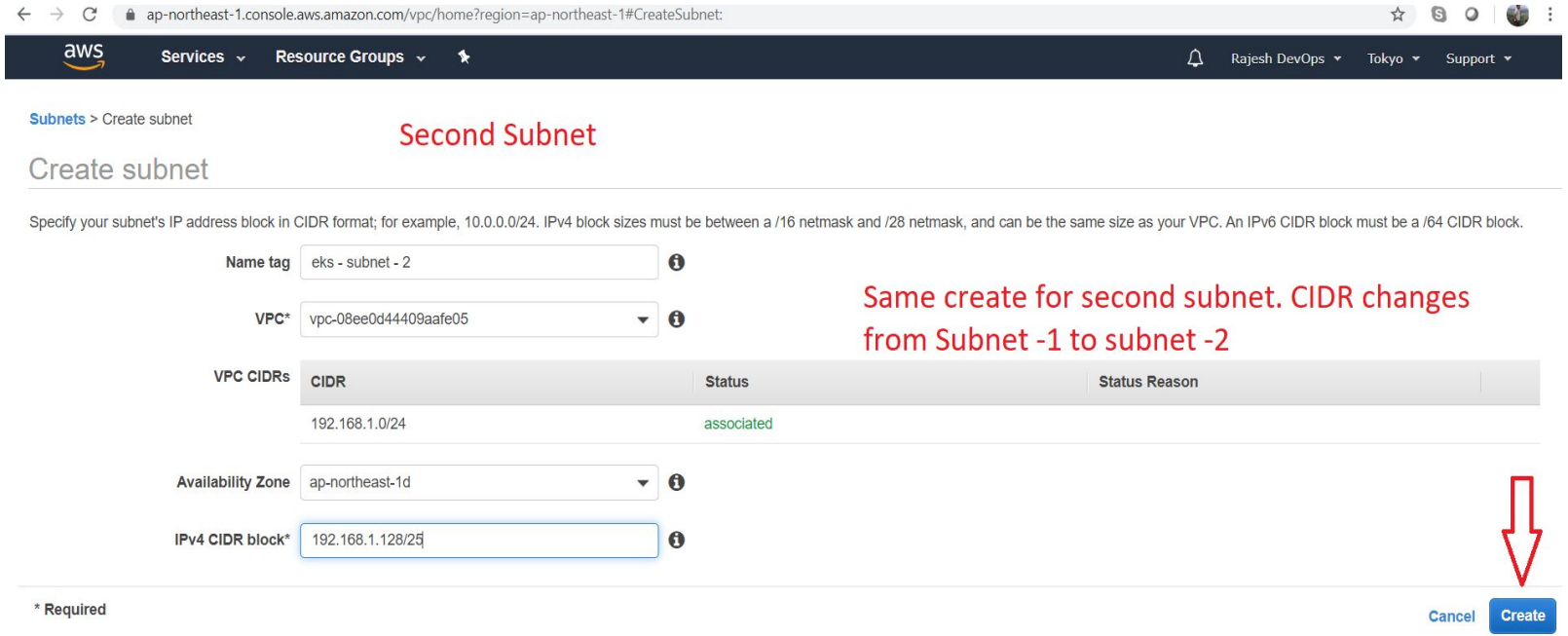
VPC - Subnets - Route Tables - Internet Gateway



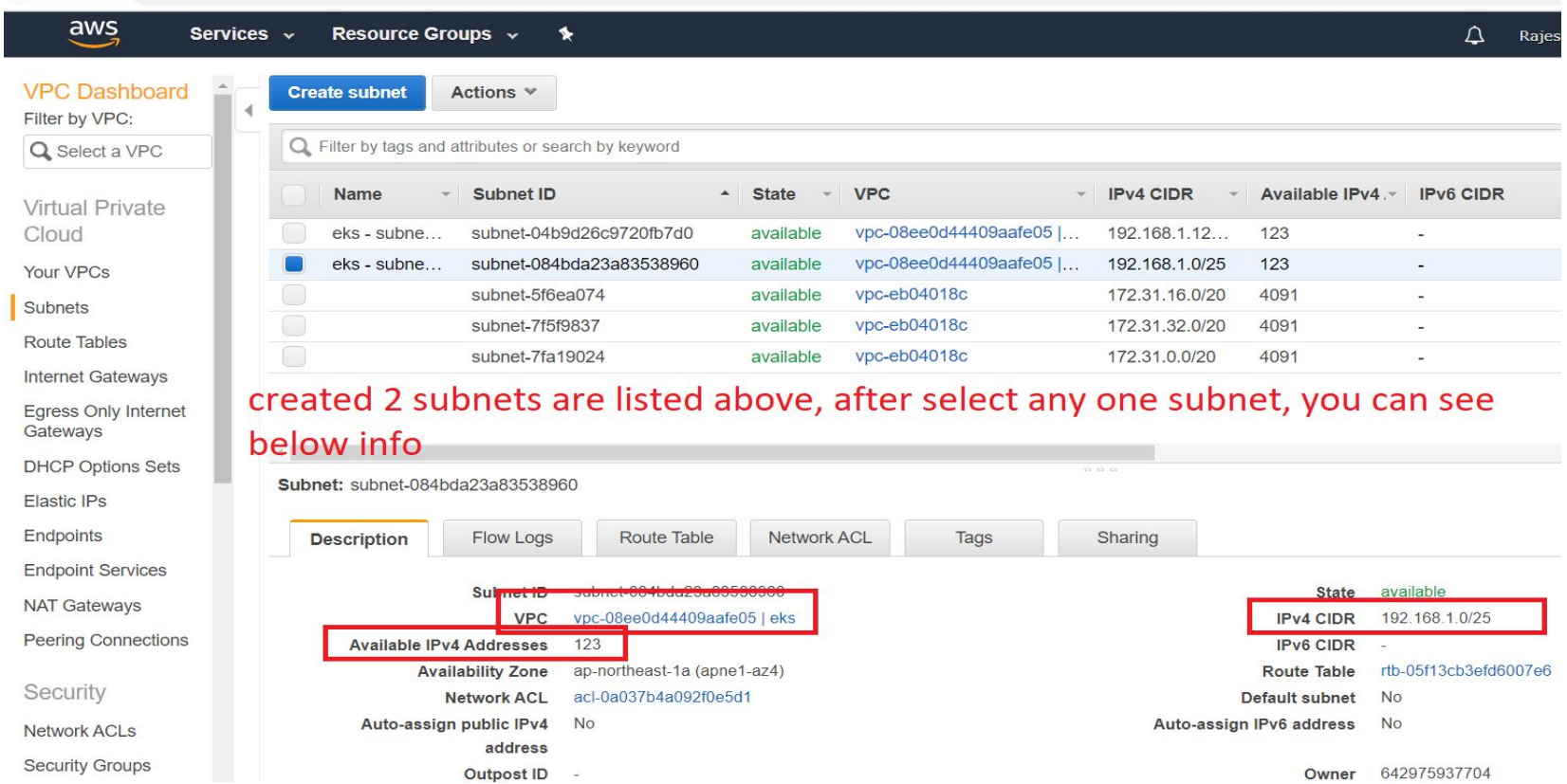
VPC - Subnets - Route Tables - Internet Gateway



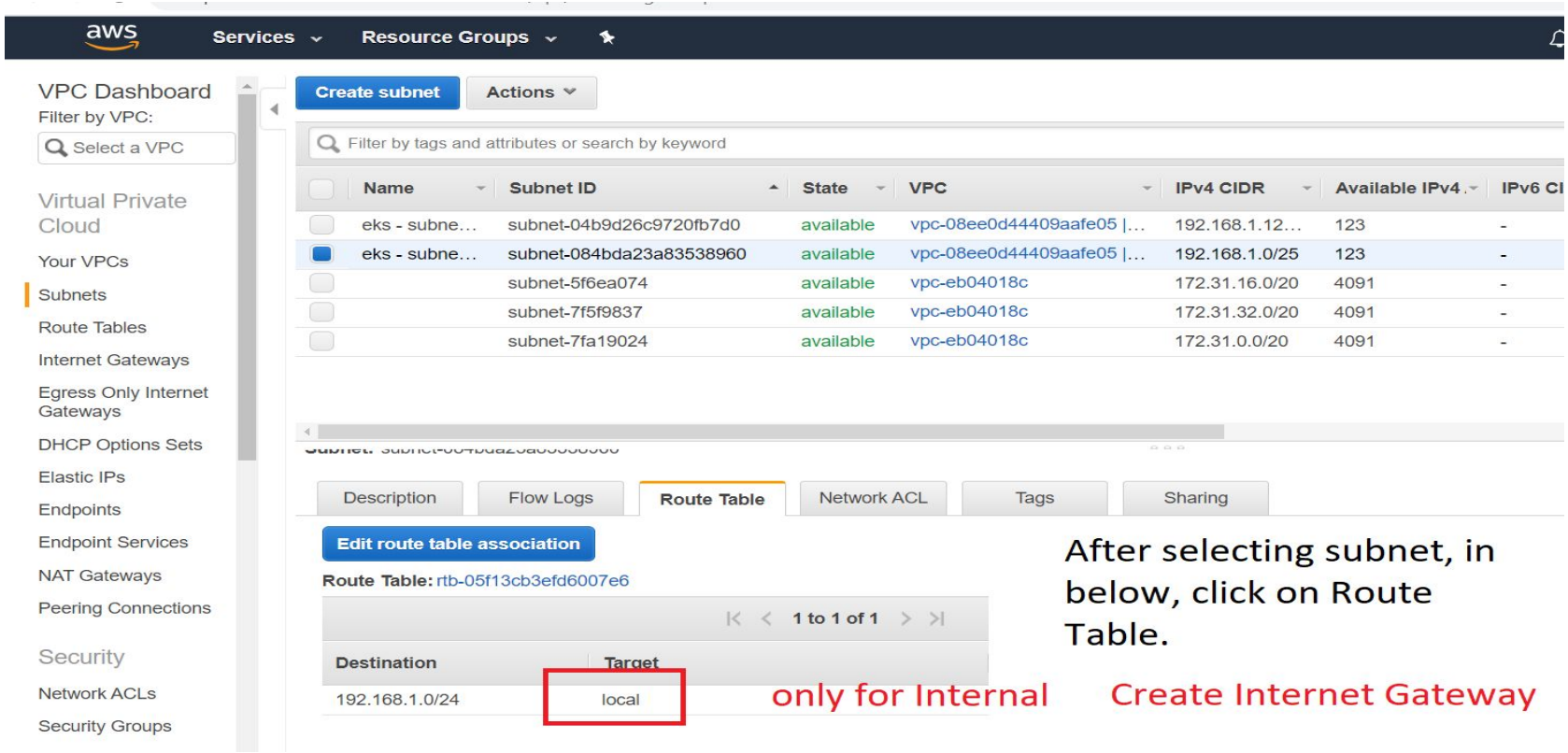
VPC - Subnets - Route Tables - Internet Gateway



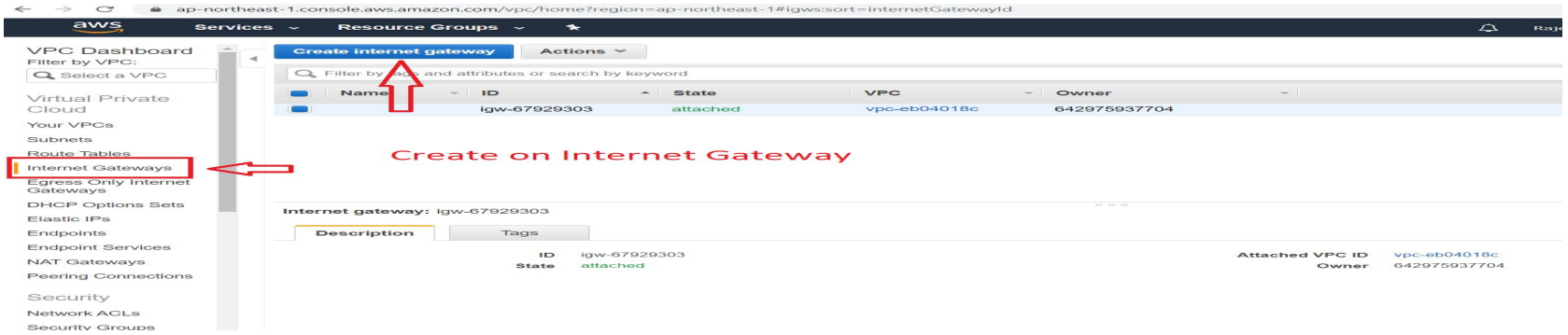
VPC - Subnets - Route Tables - Internet Gateway

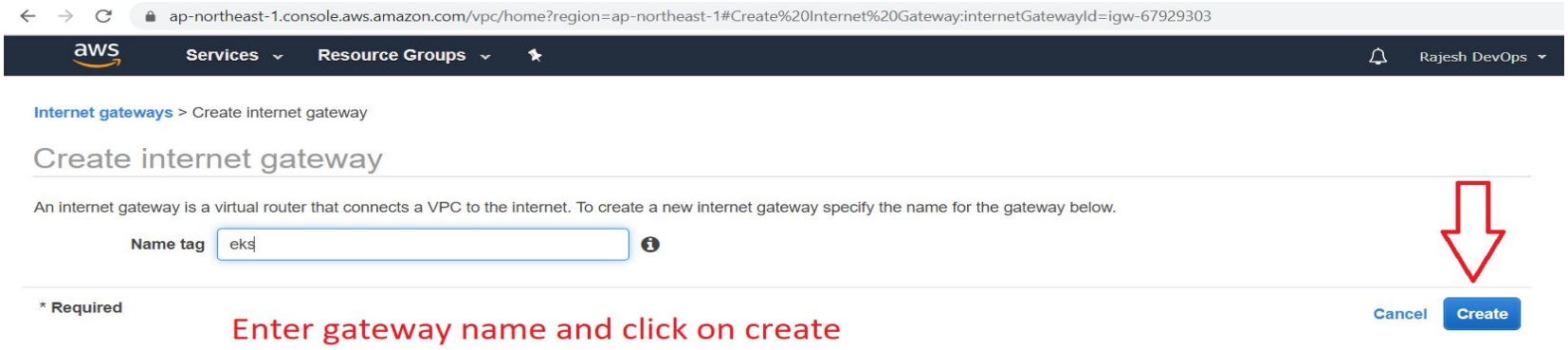


VPC - Subnets - Route Tables - Internet Gateway

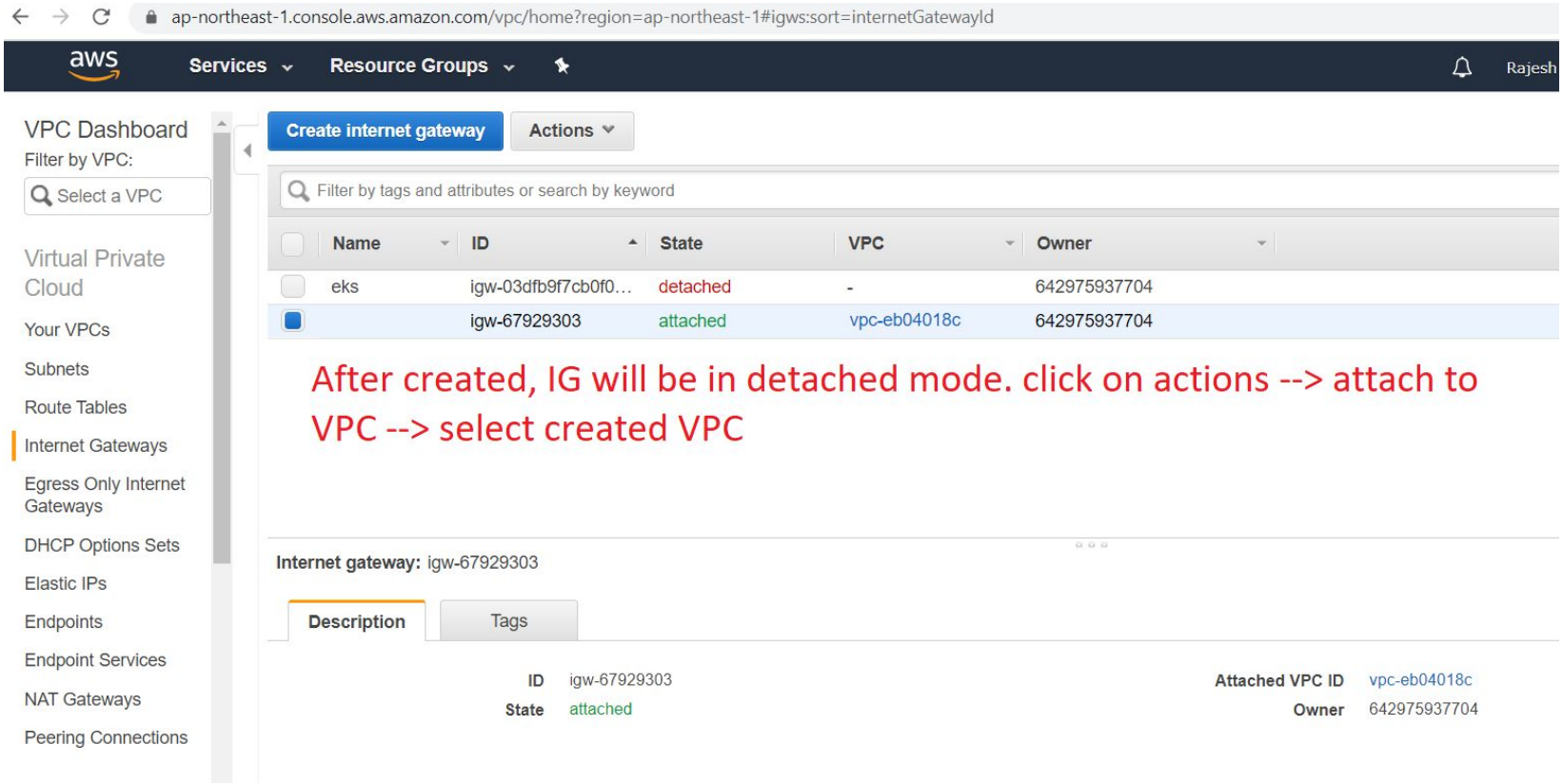


VPC - Subnets - Route Tables - Internet Gateway

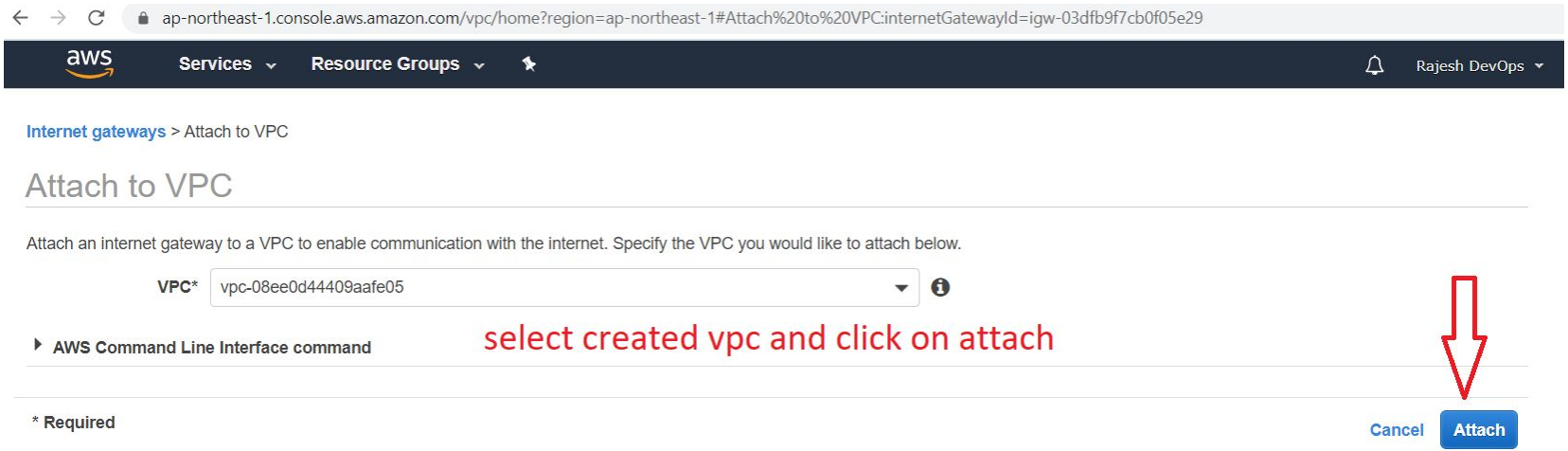




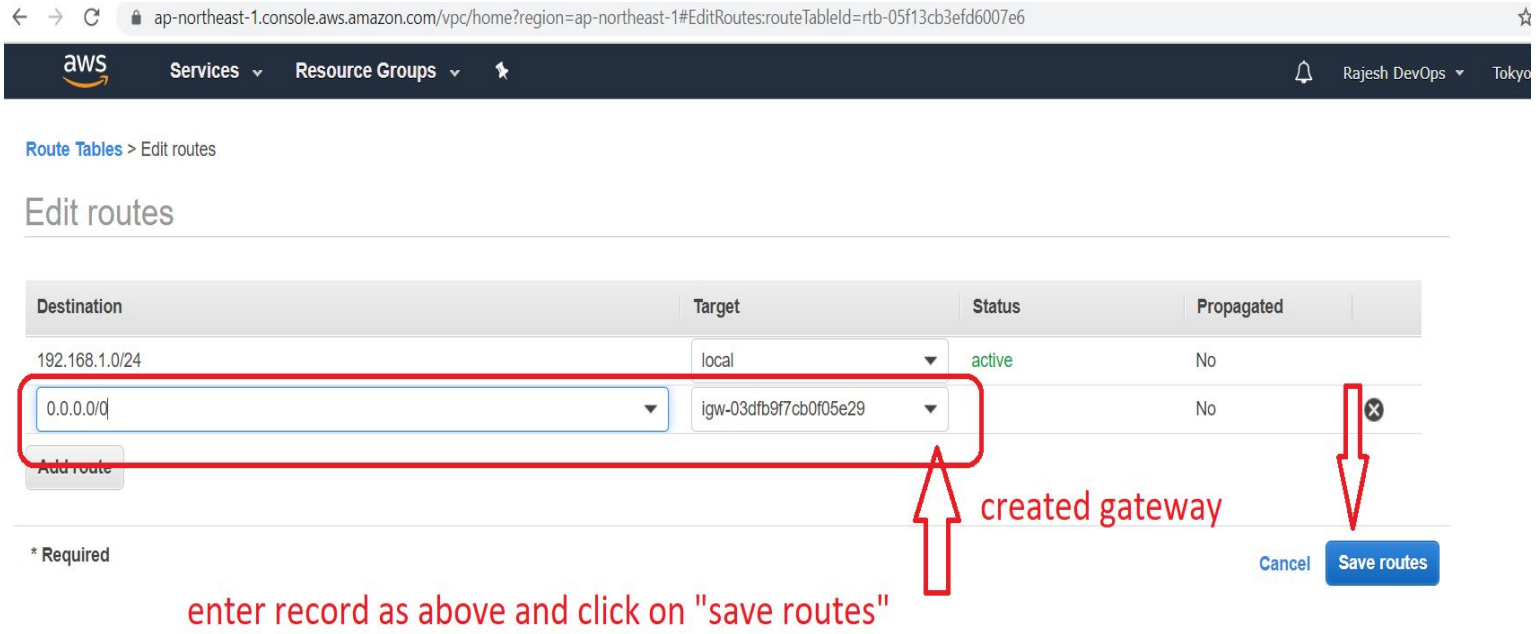
VPC - Subnets - Route Tables - Internet Gateway



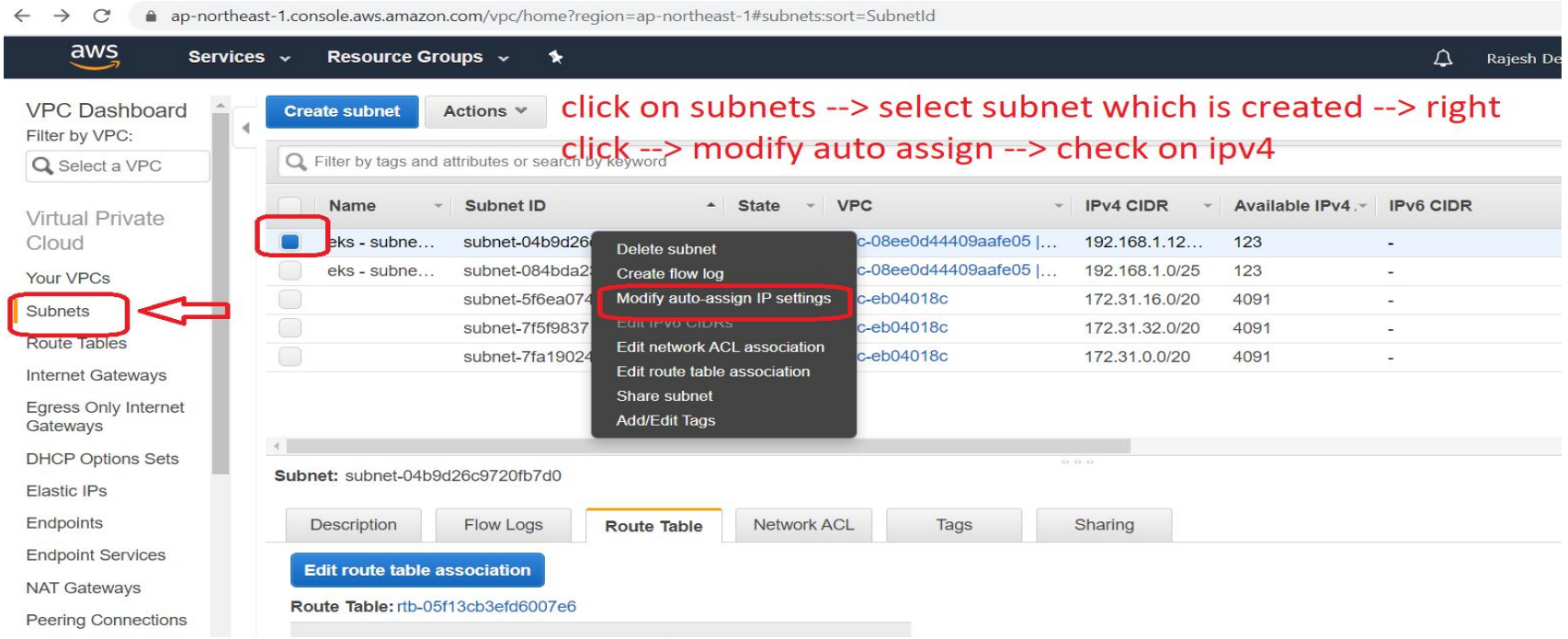
VPC - Subnets - Route Tables - Internet Gateway



VPC - Subnets - Route Tables - Internet Gateway



VPC - Subnets - Route Tables - Internet Gateway

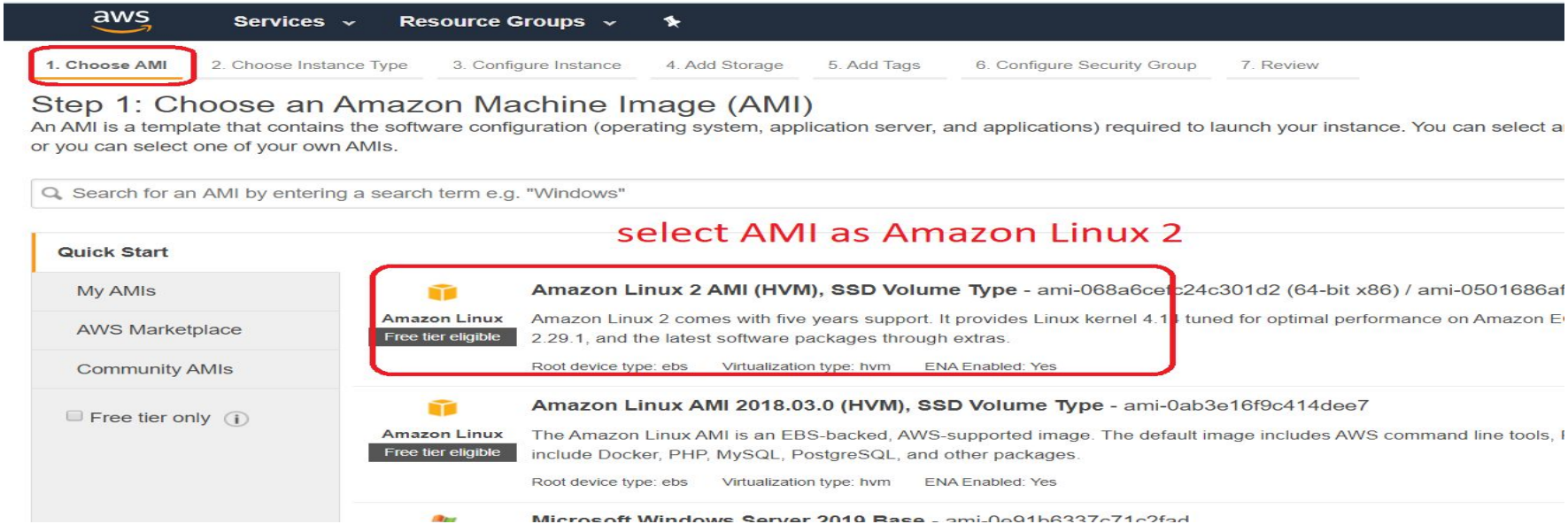


--- VPC CREATION COMPLETED ---

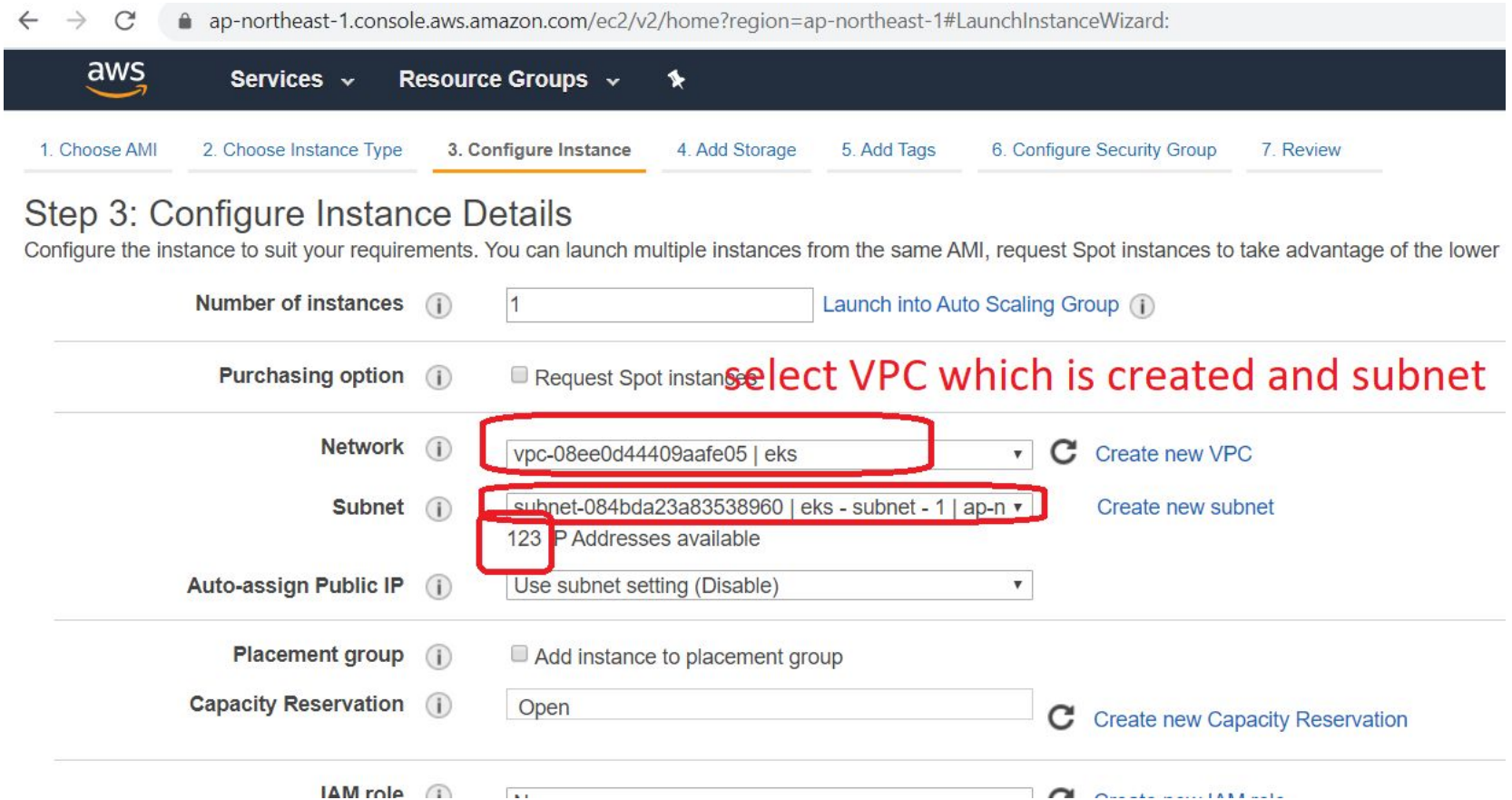
Baston - Host Launching and Configuration

Server or instance which acts as medium between user/client and EKS is known as Baston host

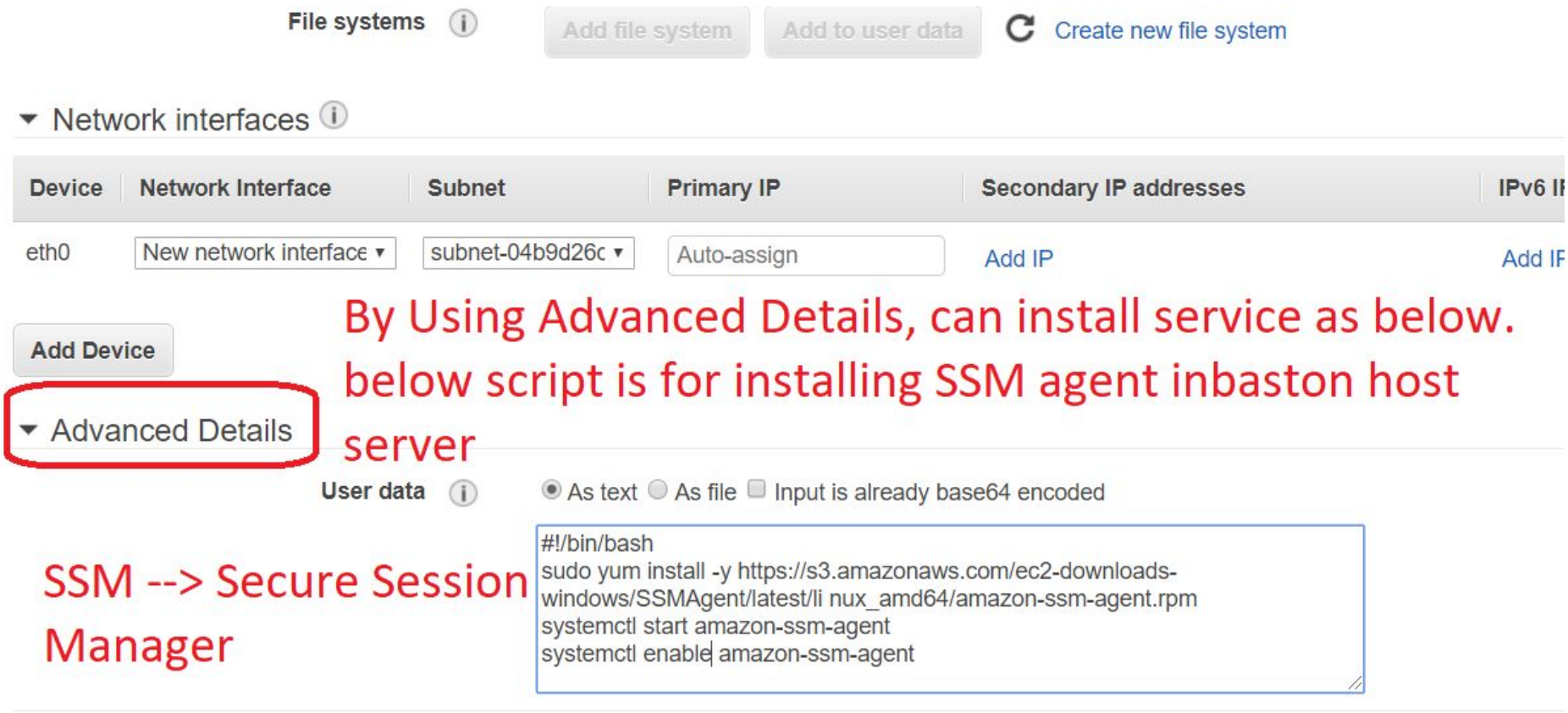
Launch EC2 Instance and select VPC which we created while launching.



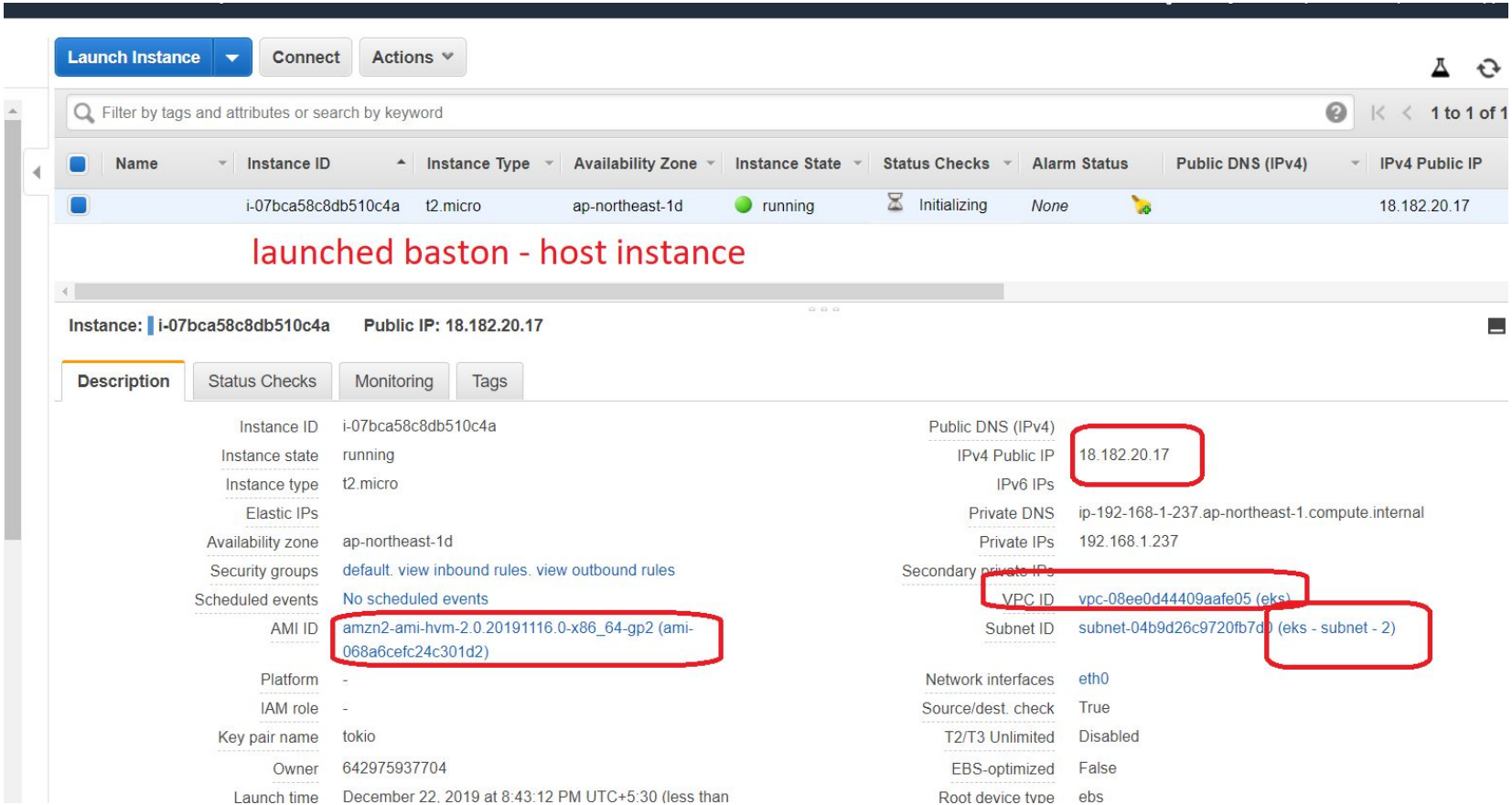
Baston - Host Launching and Configuration



Baston - Host Launching and Configuration



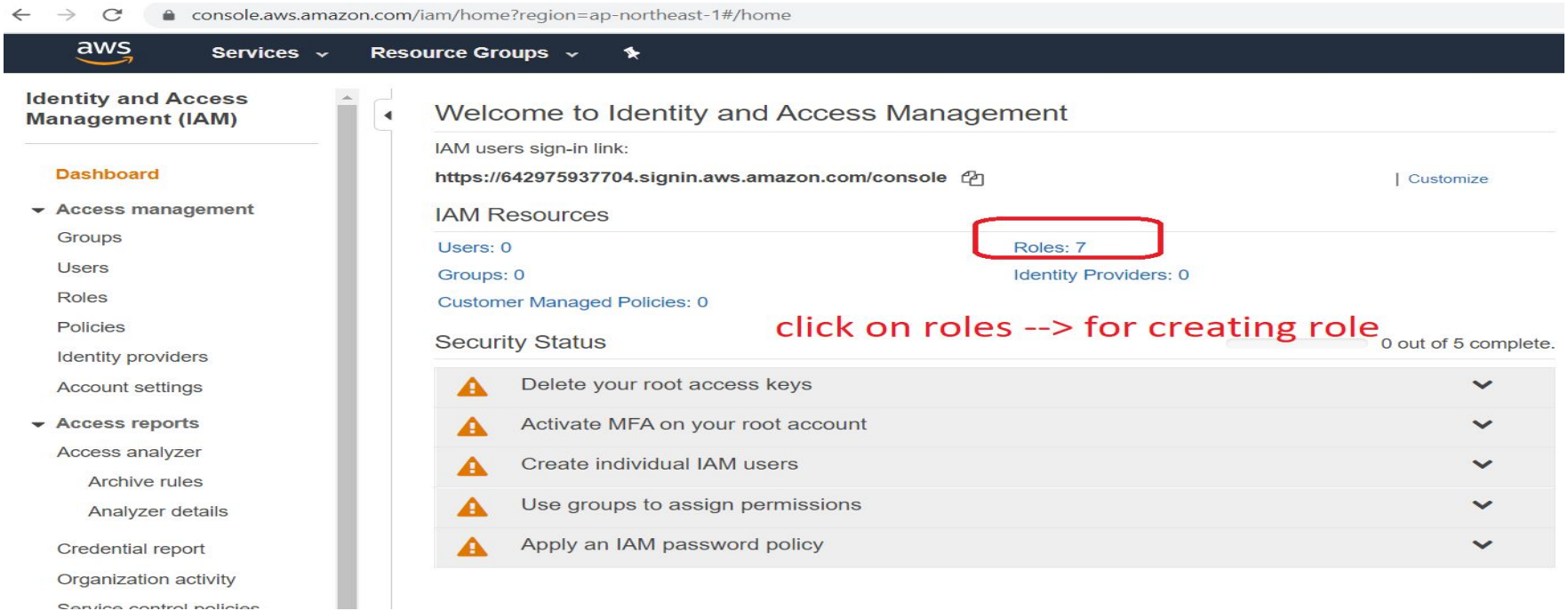
Baston - Host Launching and Configuration



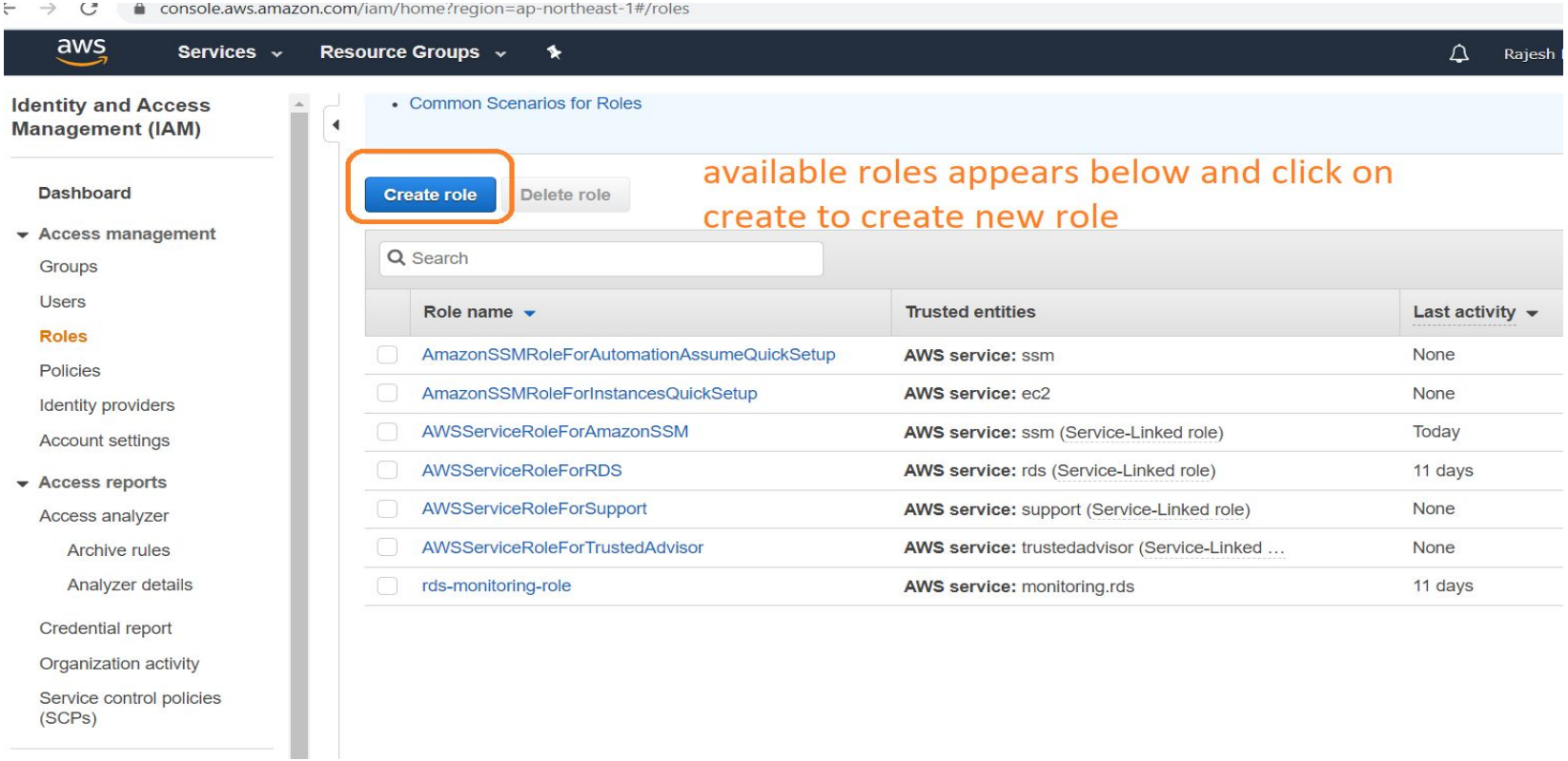
IAM - Identity & Access Management

Create a role (IAM) for SSM login and attach to Baston - host EC2 instance

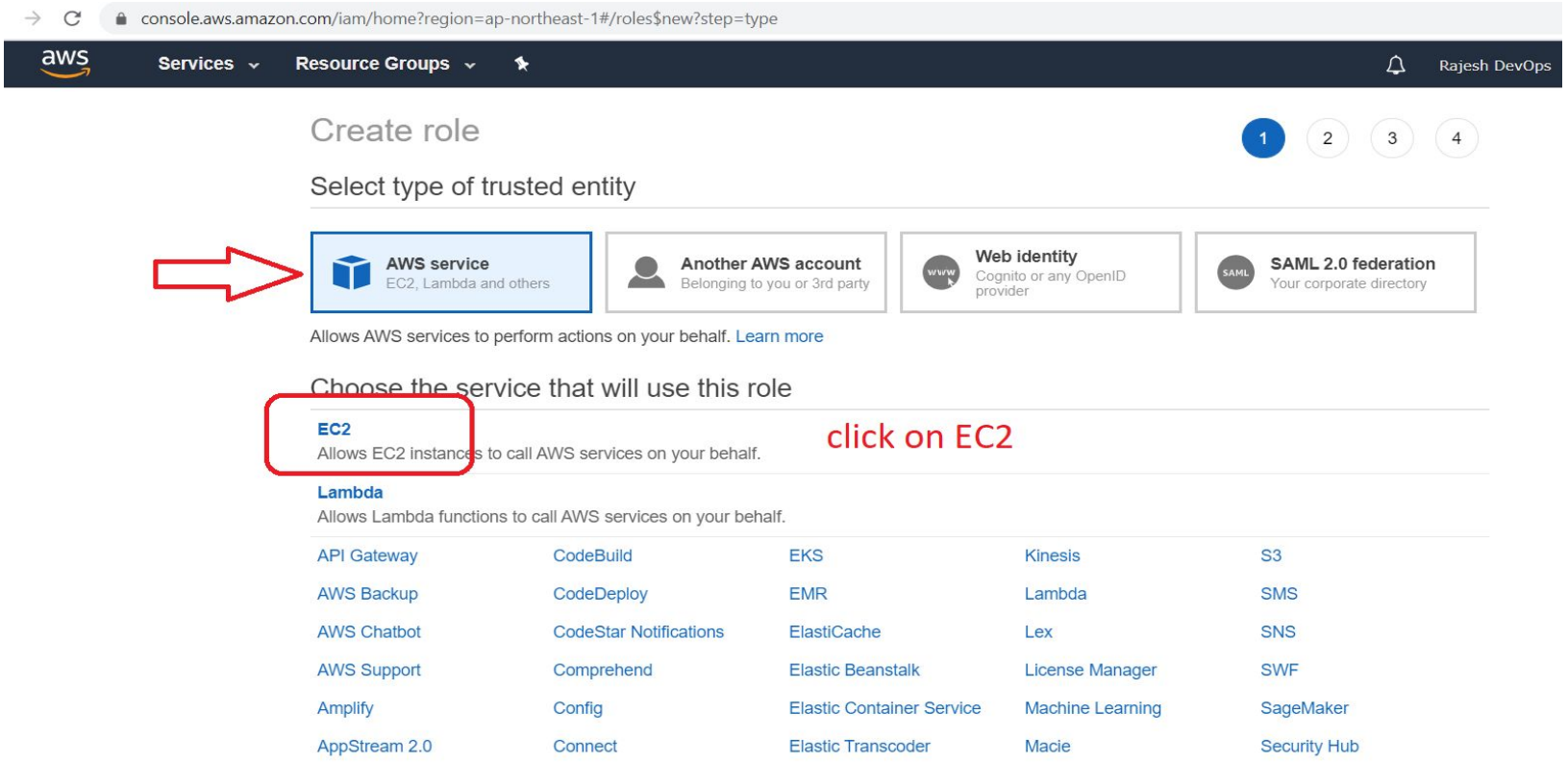
AWS Login → Services → Security, Identity & Compliences → IAM (click), below page appears



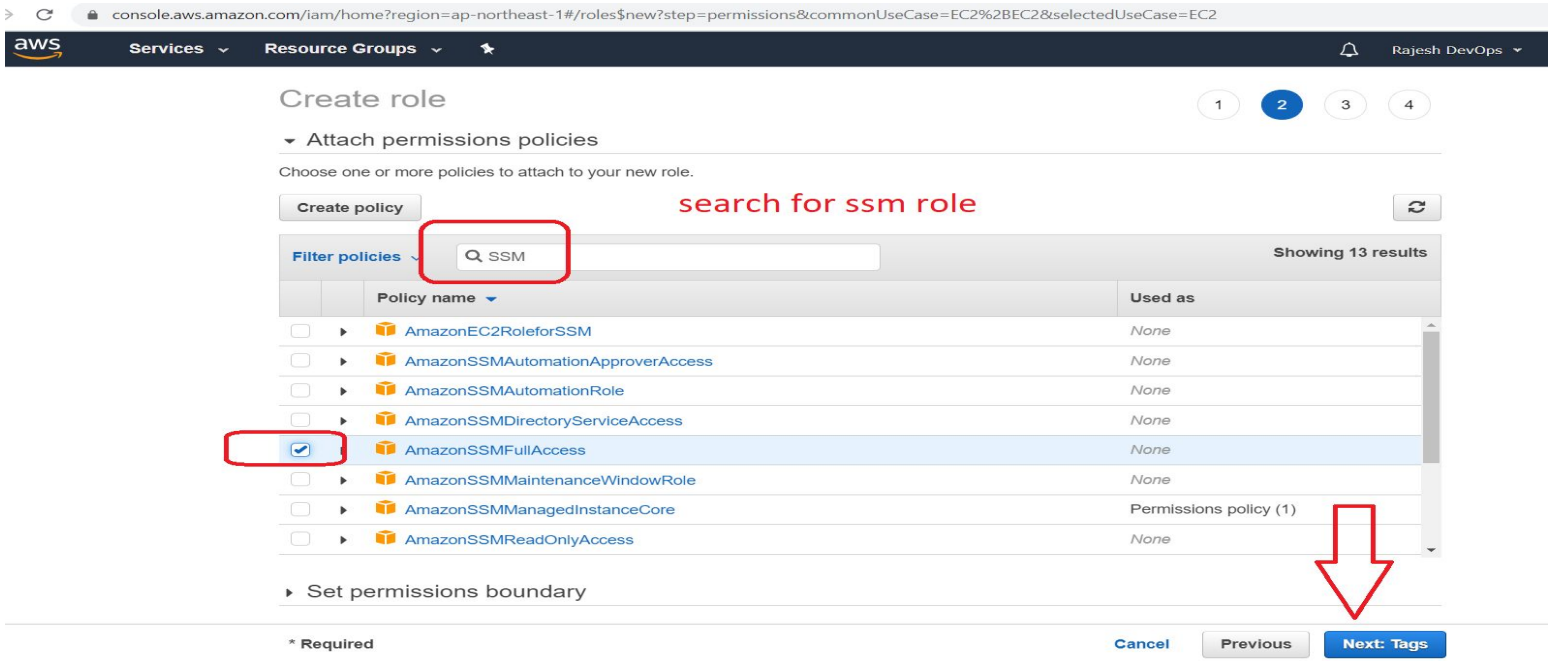
IAM - Identity & Access Management



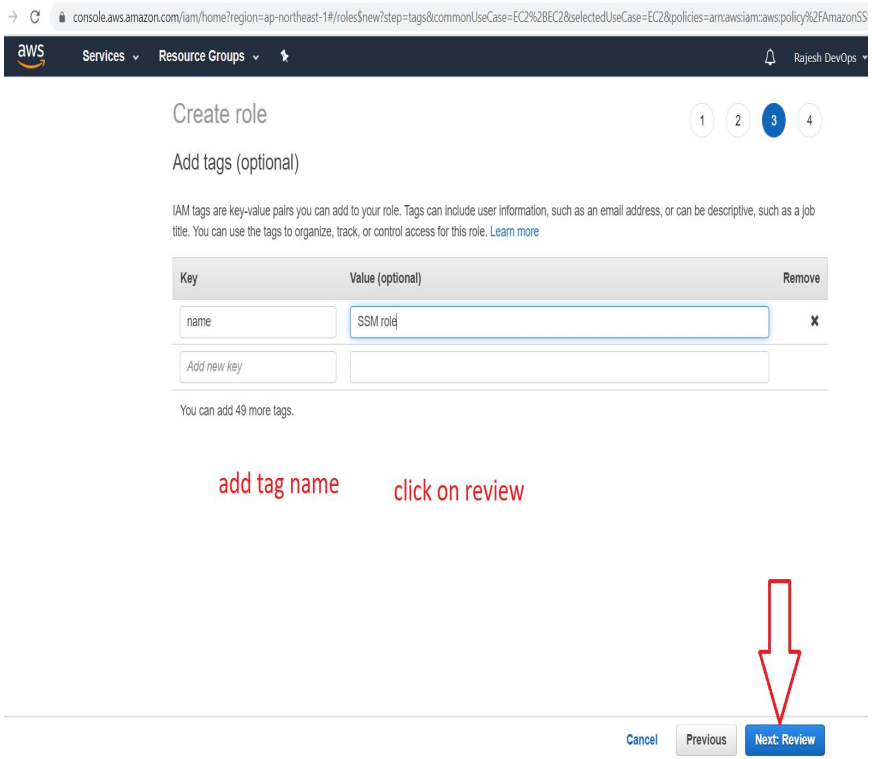
IAM - Identity & Access Management

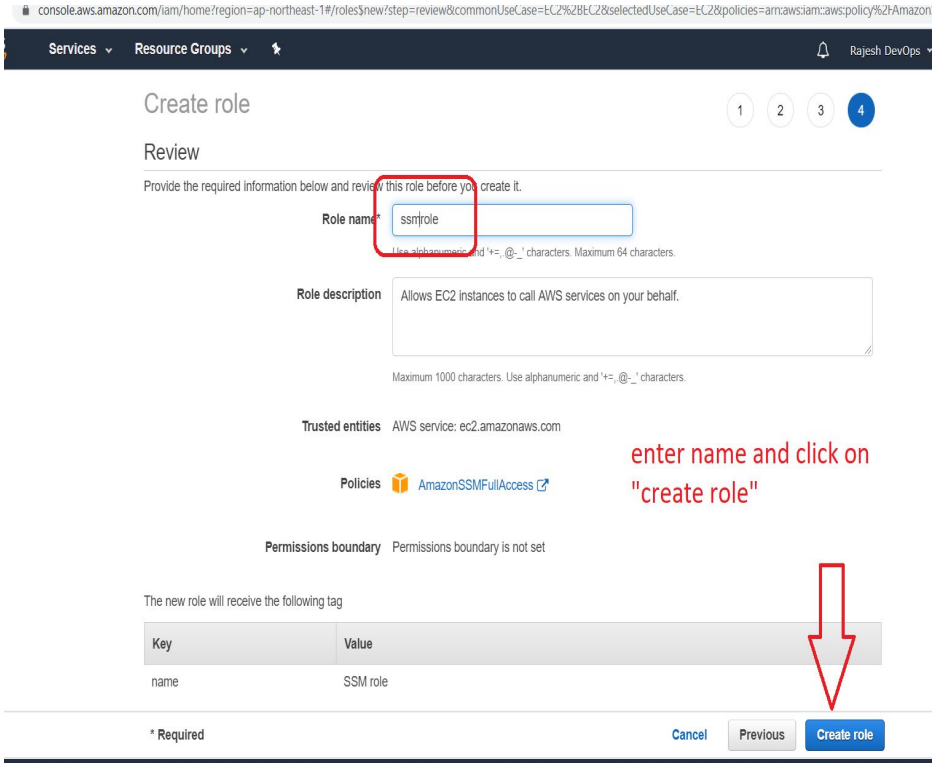


IAM - Identity & Access Management

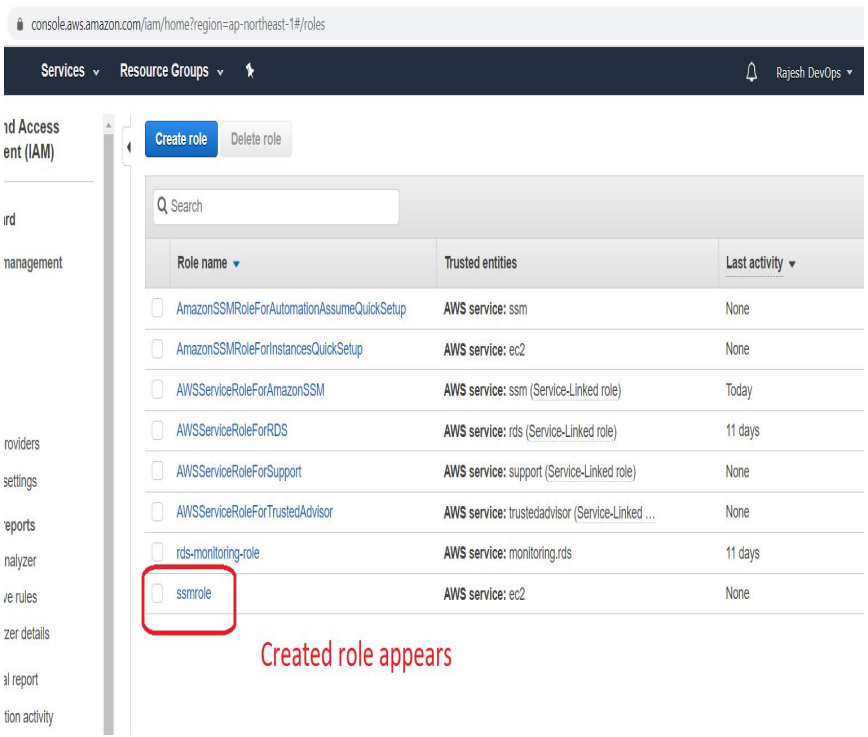


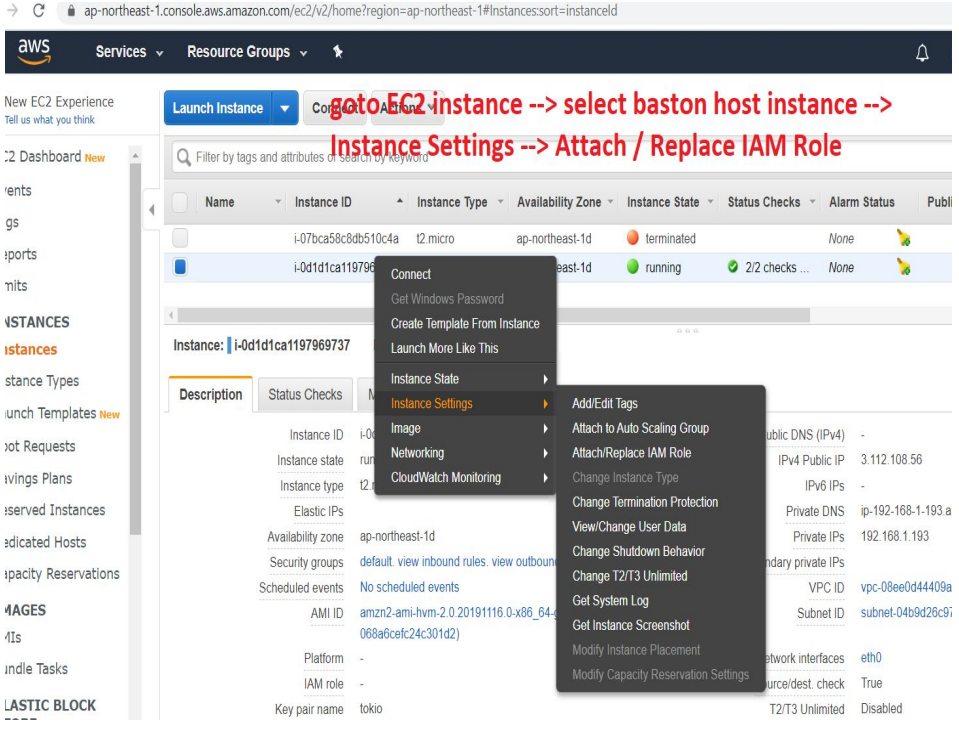
IAM - Identity & Access Management



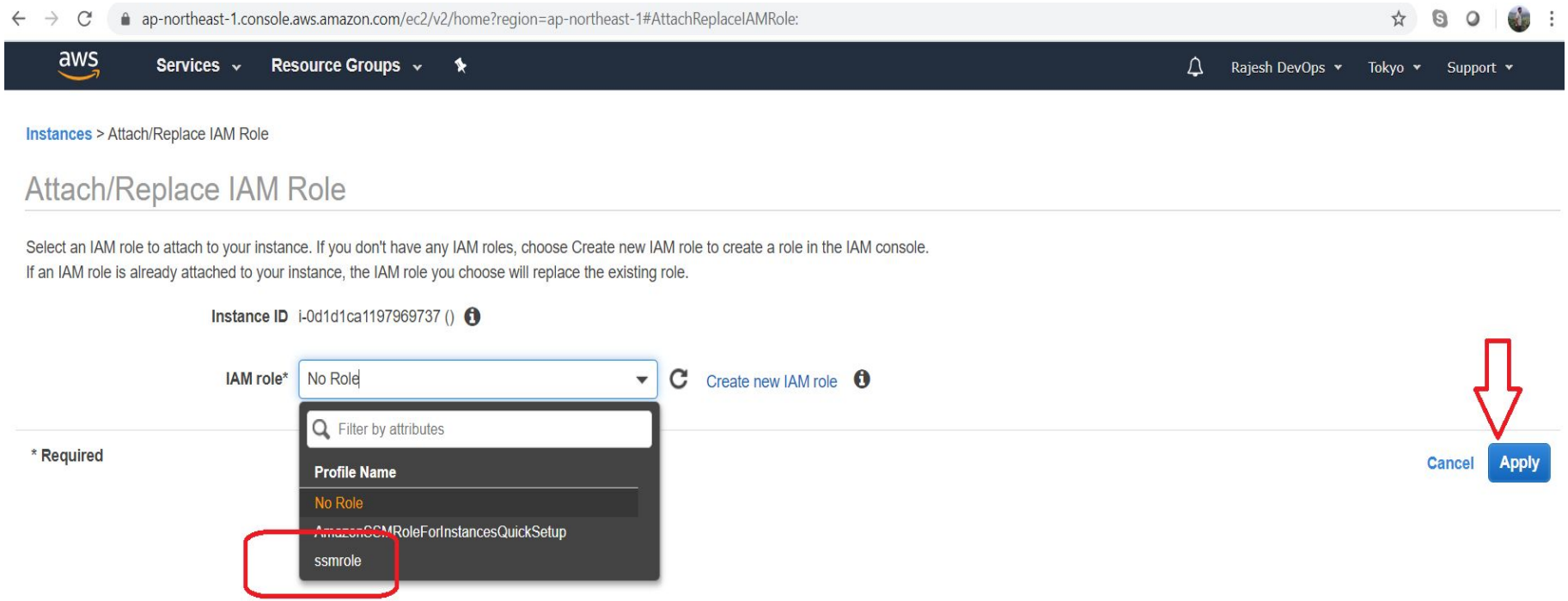


Attaching Created IAM role (SSM) to Baston instance

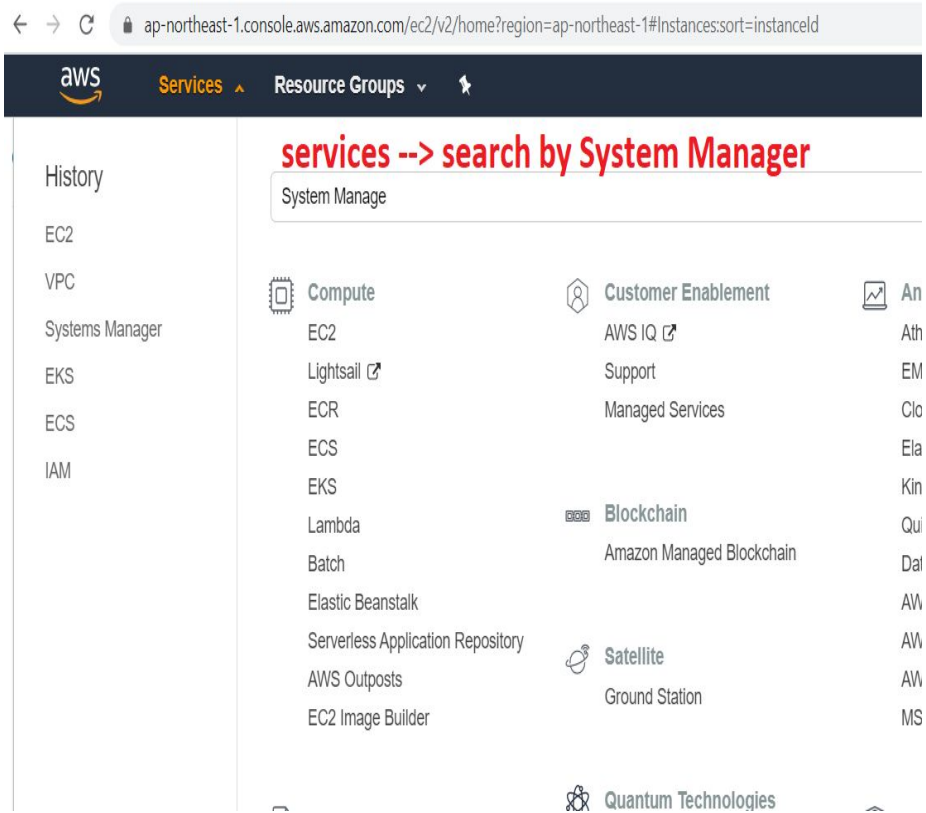


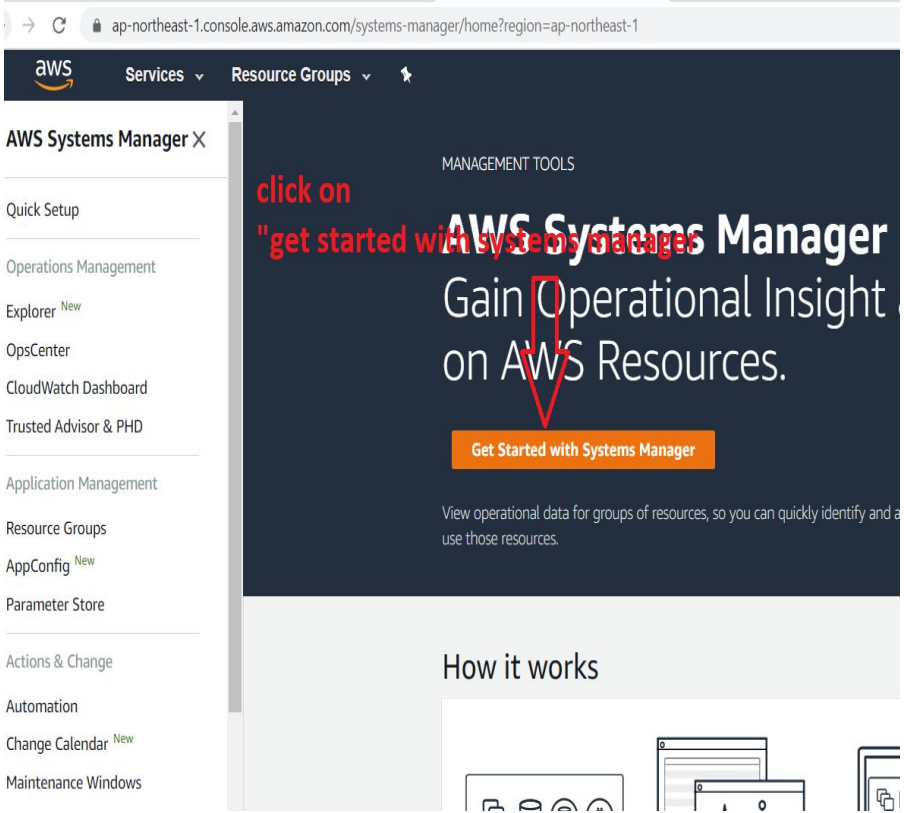


Attaching Created IAM role (SSM) to Baston instance

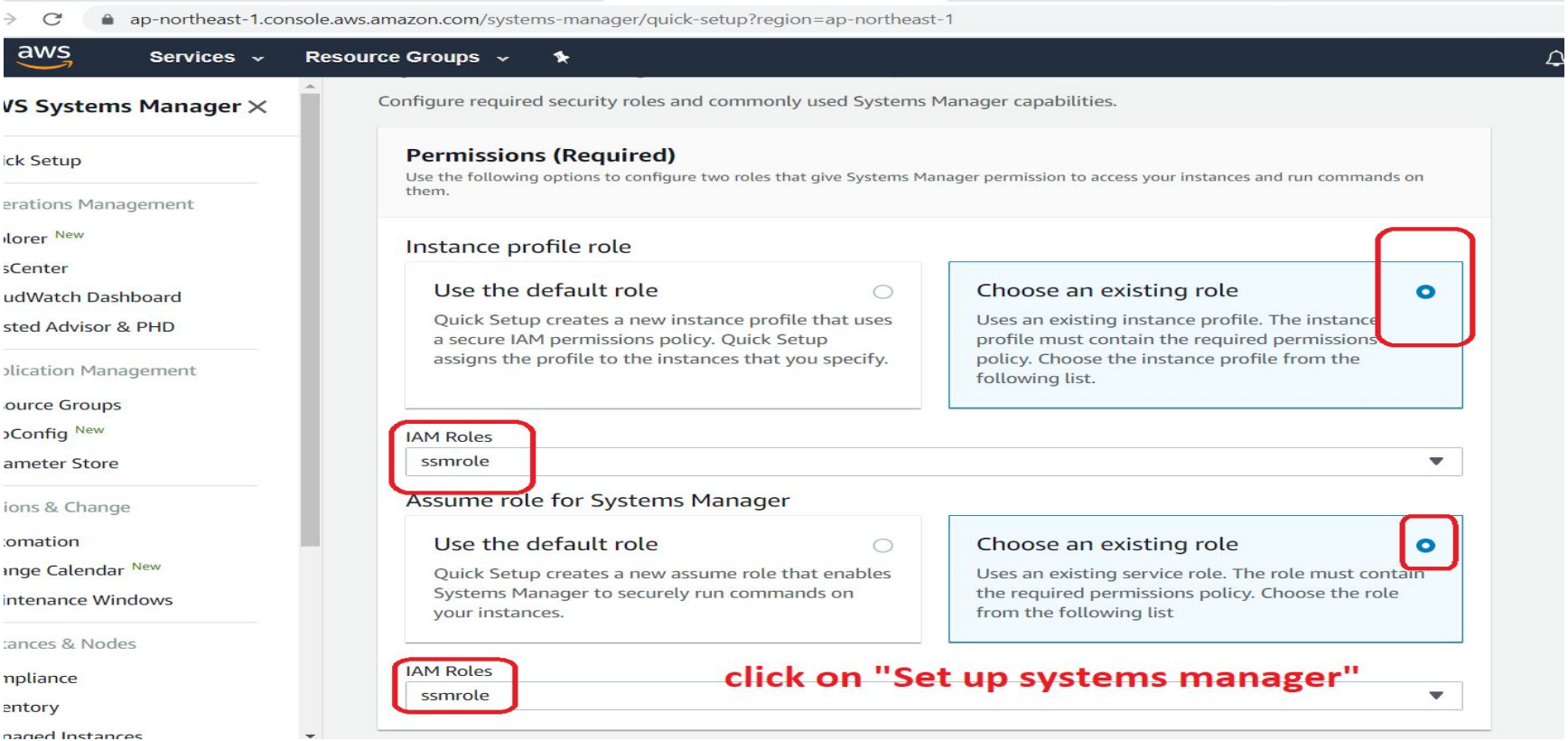


Configuring System Manager

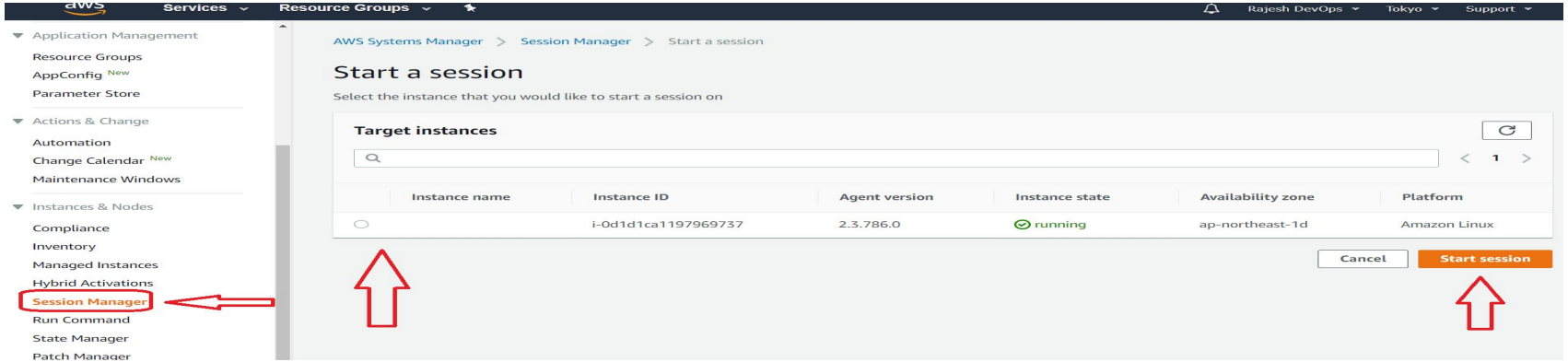




Configuring System Manager

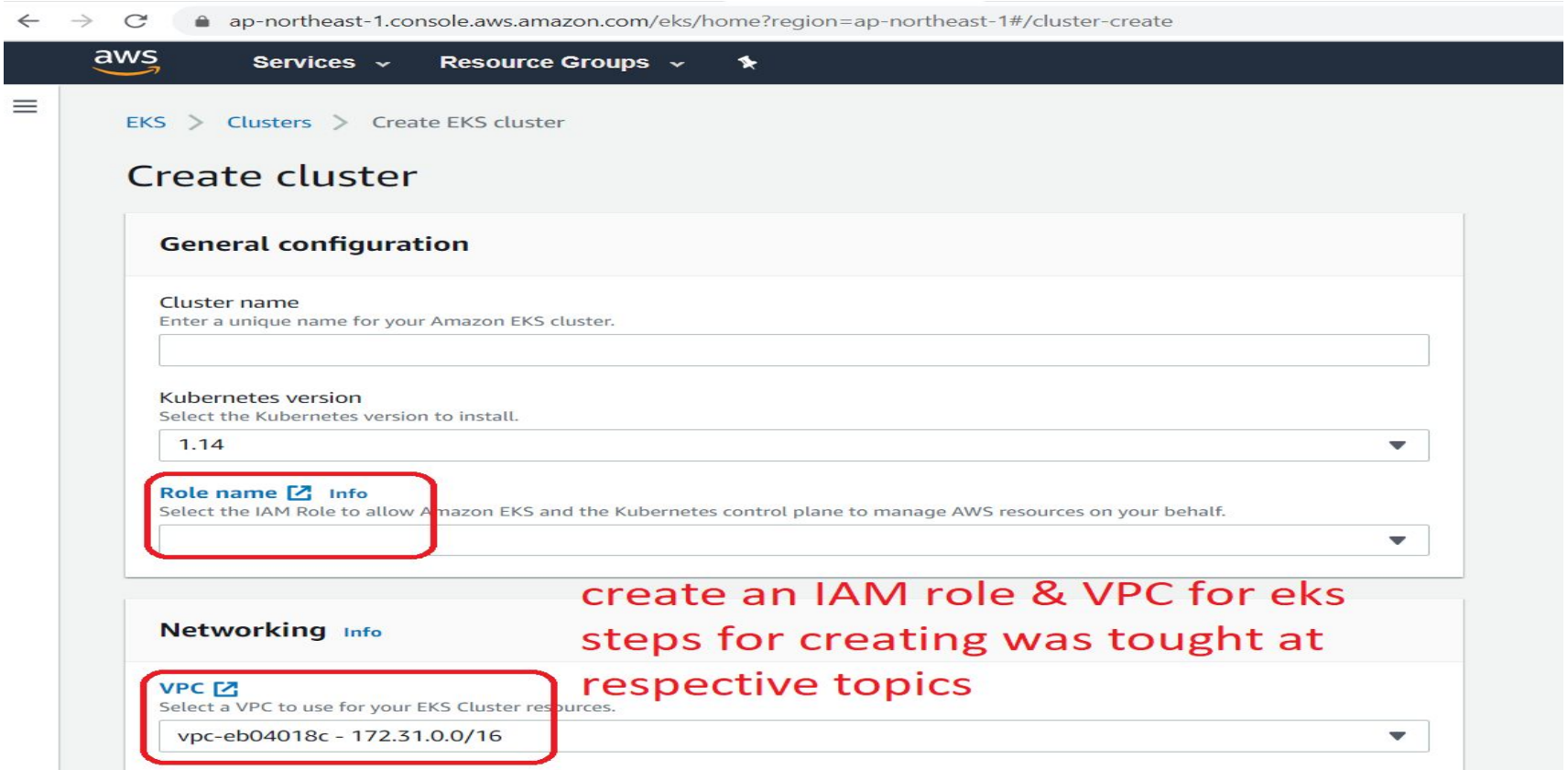


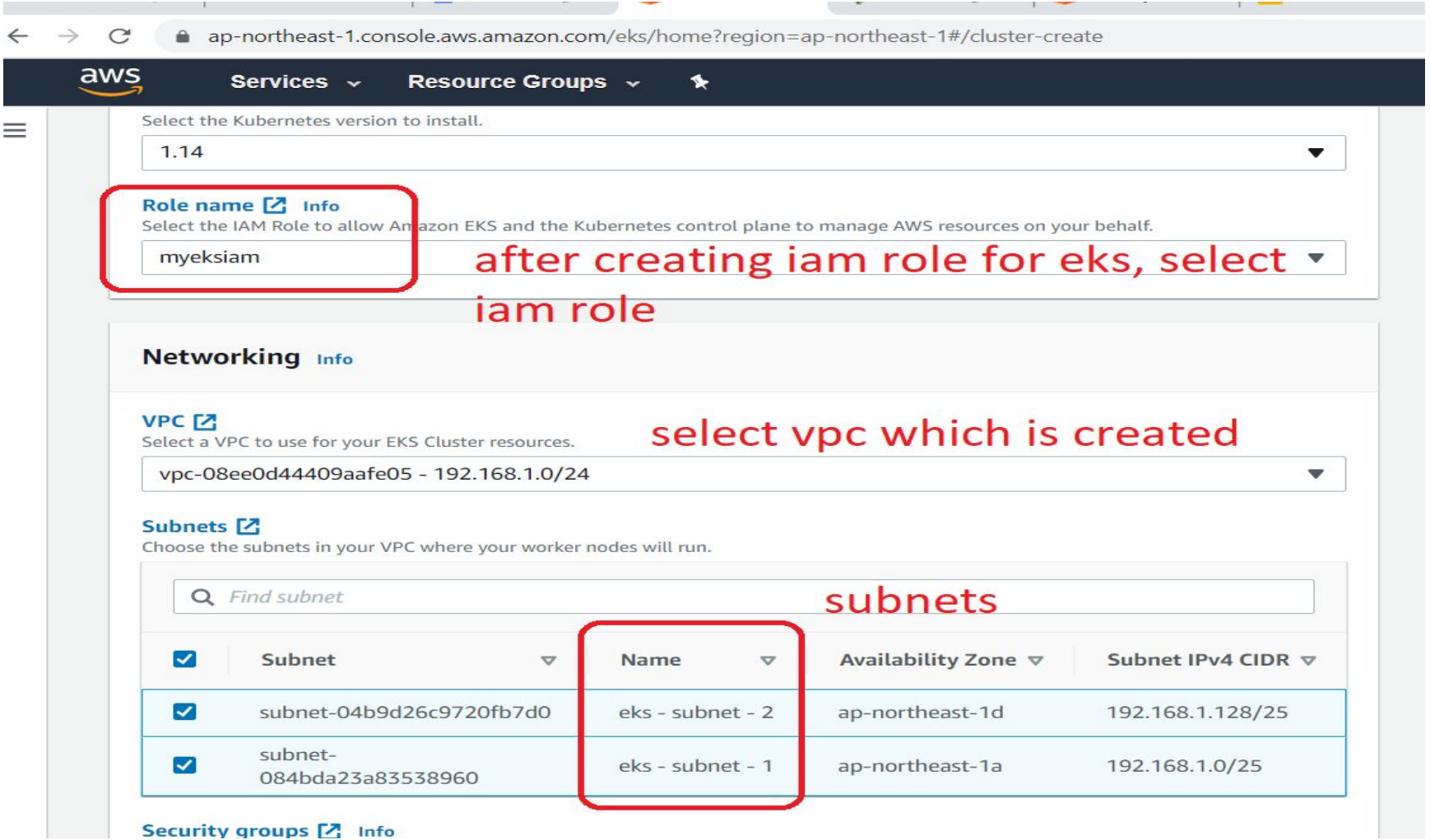
Configuring System Manager

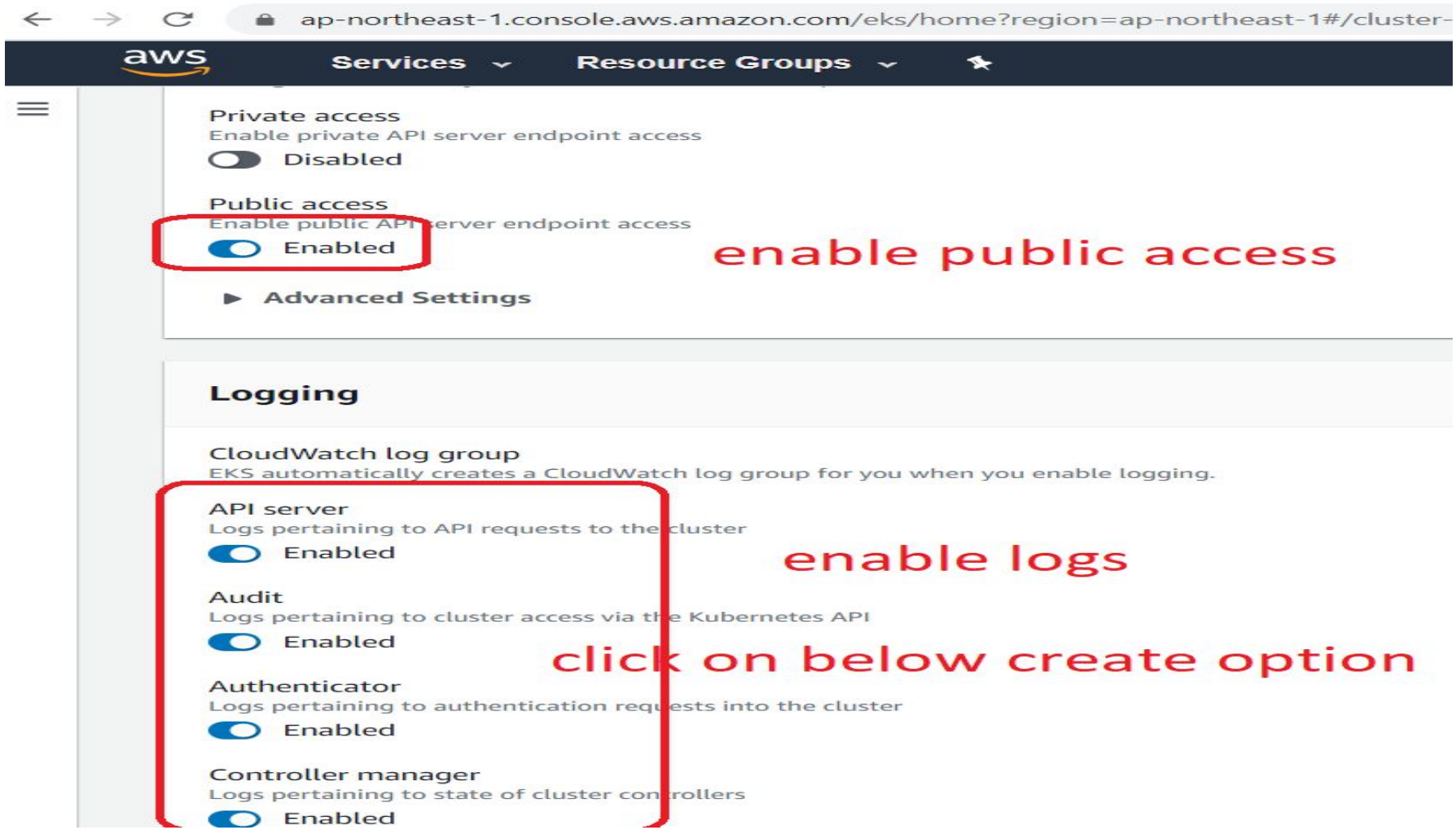


AWS - EKS

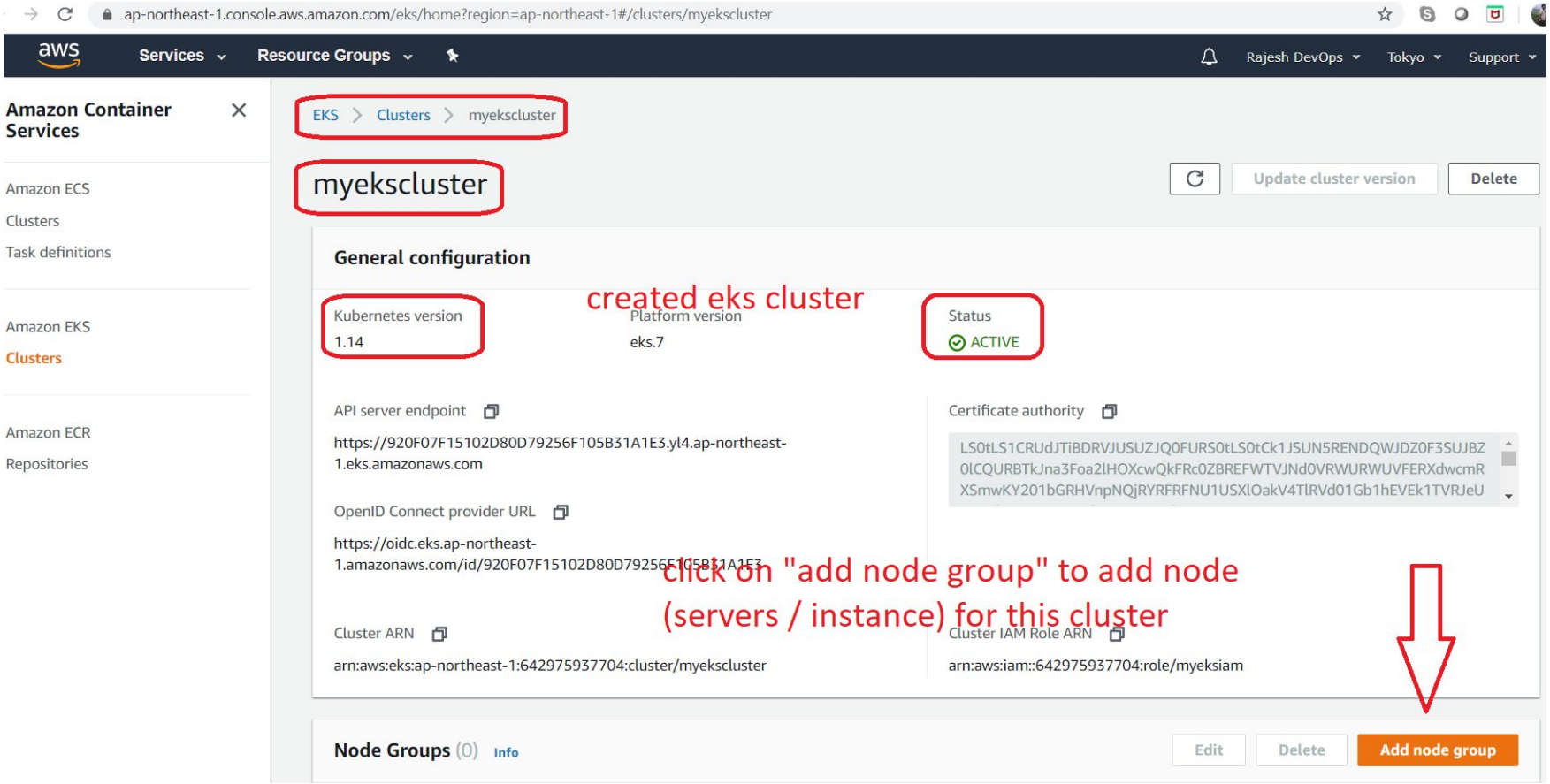
AWS → Services → Compute → EKS → clusters (left menu) → create cluster




ap-northeast-1.console.aws.amazon.com/eks/home?region=ap-northeast-1#/cluster-create
aws
Services
Resource Groups
Select the Kubernetes version to install.
1.14
Role name Info
Select the IAM Role to allow Amazon EKS and the Kubernetes control plane to manage AWS resources on your behalf.
myeksiam
after creating iam role for eks, select iam role
Networking Info
VPC
Select a VPC to use for your EKS Cluster resources.
select vpc which is created
vpc-08ee0d44409aafe05 - 192.168.1.0/24
Subnets
Choose the subnets in your VPC where your worker nodes will run.
Find subnet
subnets
Subnet
Name
Availability Zone
Subnet IPv4 CIDR
subnet-04b9d26c9720fb7d0
eks - subnet - 2
ap-northeast-1d
192.168.1.128/25
subnet-084bda23a83538960
eks - subnet - 1
ap-northeast-1a
192.168.1.0/25
Security groups Info


ap-northeast-1.console.aws.amazon.com/eks/home?region=ap-northeast-1#/cluster-
aws
Services
Resource Groups
Private access
Enable private API server endpoint access
Disabled
Public access
Enable public API server endpoint access
Enabled
enable public access
Advanced Settings
Logging
CloudWatch log group
EKS automatically creates a CloudWatch log group for you when you enable logging.
API server
Logs pertaining to API requests to the cluster
Enabled
enable logs
Audit
Logs pertaining to cluster access via the Kubernetes API
Enabled
click on below create option
Authenticator
Logs pertaining to authentication requests into the cluster
Enabled
Controller manager
Logs pertaining to state of cluster controllers
Enabled

Created EKS cluster



IAM Role for Adding Nodes to cluster

Creating the Amazon EKS Worker Node IAM Role

If you created your worker nodes by following the steps in the Getting Started with the AWS Management Console or Getting Started with eksctl topics, then the worker node role account already exists and you don't need to manually create it. You can use the following procedure to create the Amazon EKS worker node role if you do not already have one for your account.

this slide for creating IAM role for adding nodes

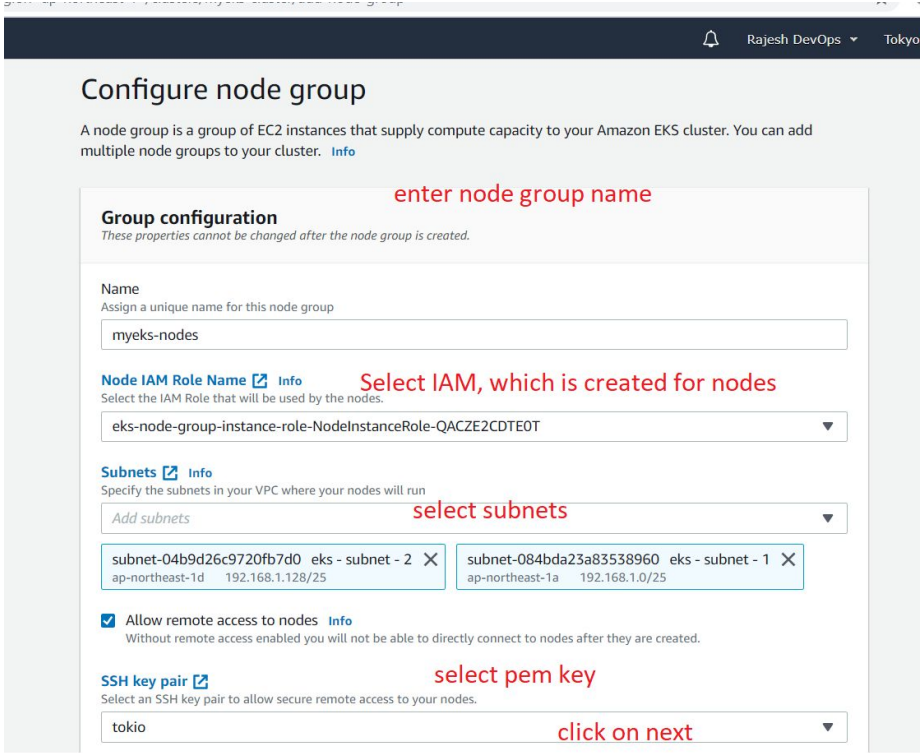
To create your Amazon EKS worker node IAM role

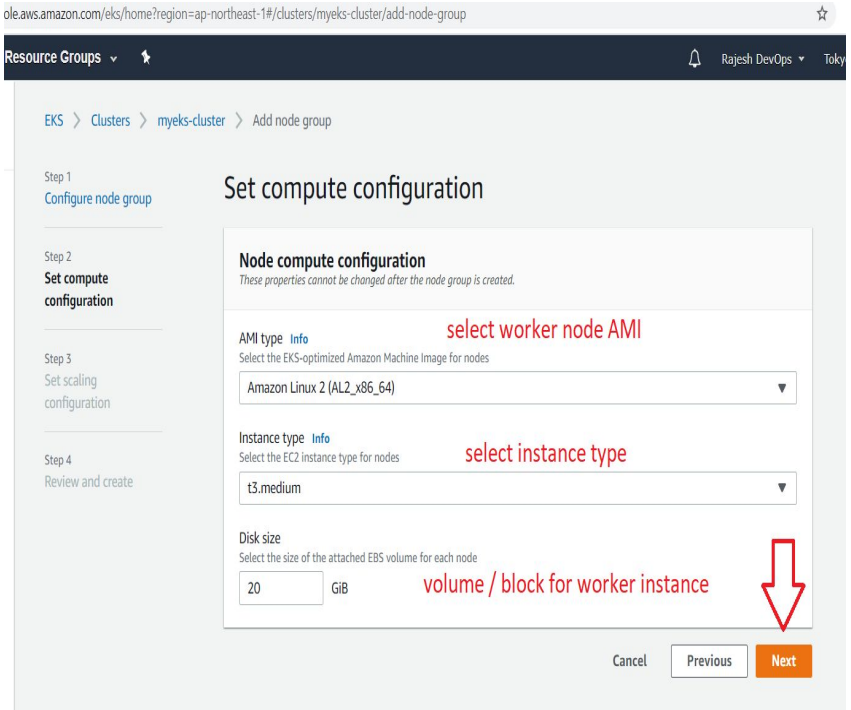
1. Open the AWS CloudFormation console at https://console.aws.amazon.com/cloudformation.
2. Choose **Create stack**.
3. For **Choose a template**, select **Specify an Amazon S3 template URL**.
4. Paste the following URL into the text area and choose **Next**:

```
https://amazon-eks.s3-us-west-2.amazonaws.com/cloudformation/2019-11-15/amazon-eks-nodegroup-role.yaml
```

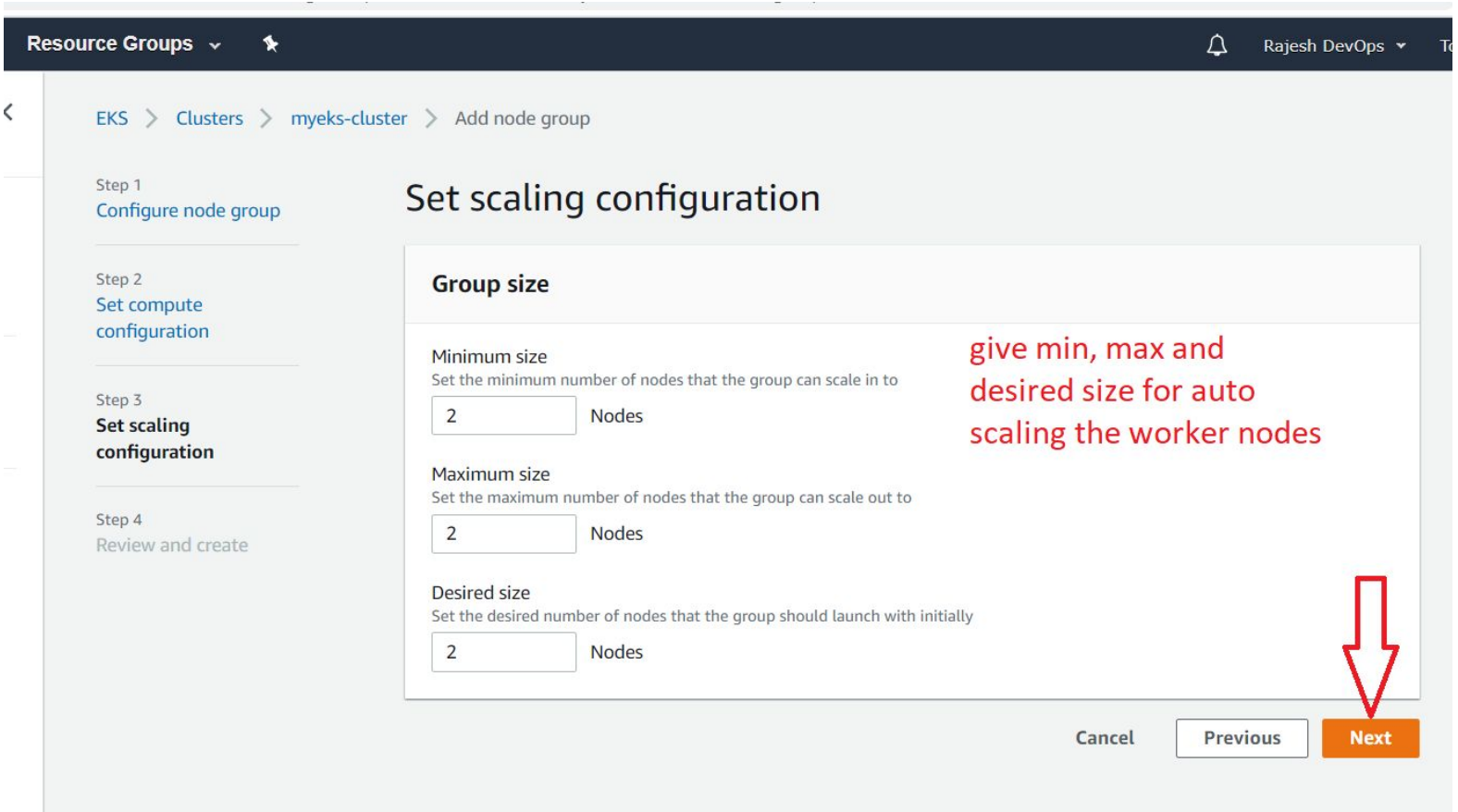
5. On the **Specify Details** page, fill out the parameters accordingly, and then choose **Next**.
   - **Stack name**: Choose a stack name for your AWS CloudFormation stack. For example, you can call it **eks-node-group-instance-role**.
6. (Optional) On the **Options** page, you can choose to tag your stack resources. Choose **Next**.
7. On the **Review** page, check the box in the **Capabilities** section and choose **Create stack**.
8. When your stack is created, select it in the console and choose **Outputs**.
9. Record the **NodeInstanceRole** value for the IAM role that was created. You need this when you create your node group.

Adding Nodes to Cluster

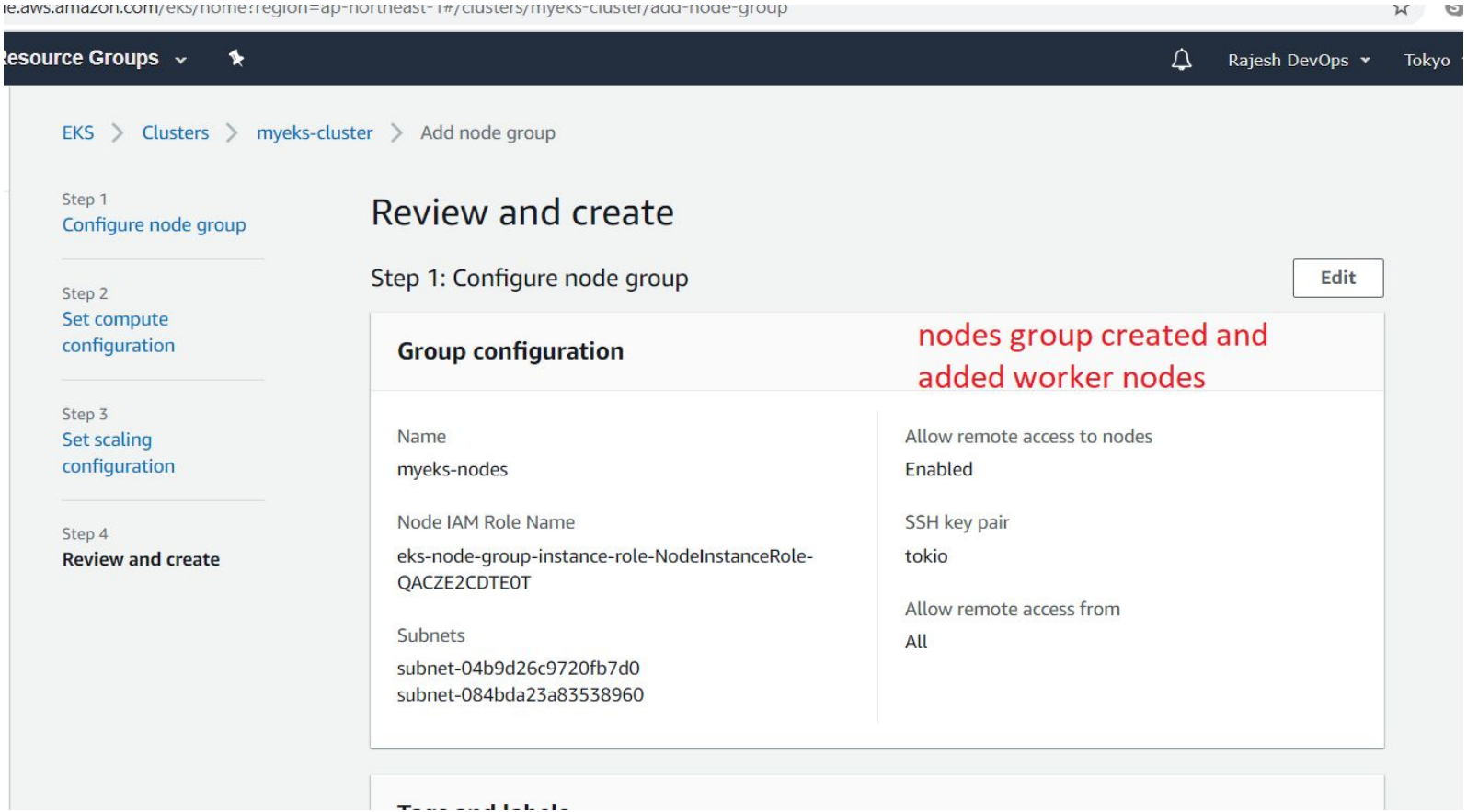




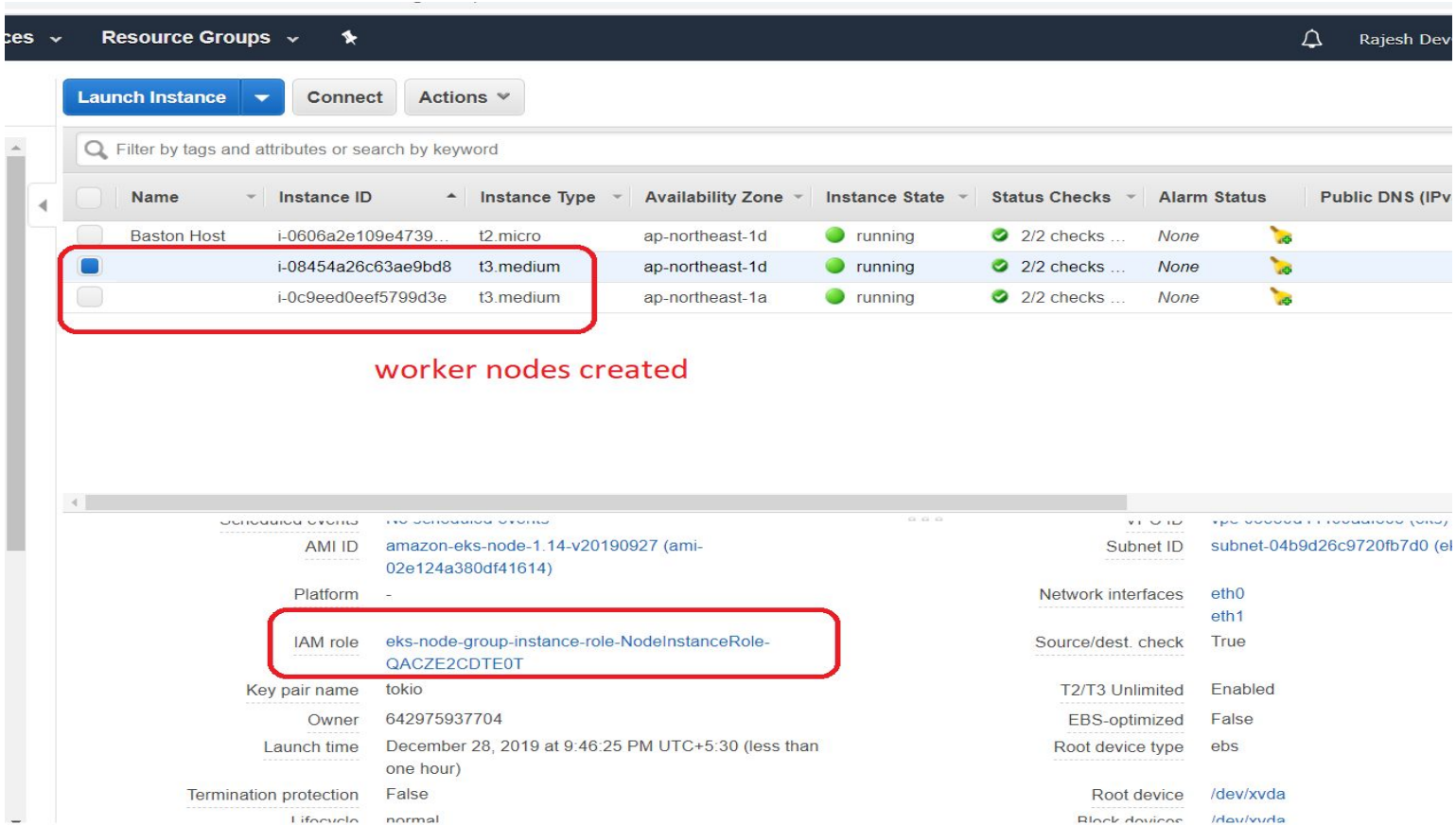
Adding Nodes to Cluster



Adding Nodes to Cluster



For Cross Checking Created Worker Nodes



Configuring AWS CLI on Client

Check for python if installed or not

```
[root@ip-192-168-1-212 /]# python --version
Python 2.7.16
[root@ip-192-168-1-212 /]#
```

Install pip tool, which is from python

```
curl https://bootstrap.pypa.io/get-pip.py -o get-pip.py
python get-pip.py
```

For confirmation → #pip

Configure awscli by using below command / snip

#pip install awscli

#aws configure

Access key id :

Access Secret key :

Region :

For Cross Checking, execute below command.

#aws s3 ls

Display if any s3 buckets available.

Install Kubernetes client in client (baston - host)

Enabling Repo

```
cat <<EOF > /etc/yum.repos.d/kubernetes.repo
[kubernetes]
name=Kubernetes
baseurl=https://packages.cloud.google.com/yum/repos/kubernetes-el7-x86_64
enabled=1
gpgcheck=1
repo_gpgcheck=1
gpgkey=https://packages.cloud.google.com/yum/doc/yum-key.gpg
https://packages.cloud.google.com/yum/doc/rpm-package-key.gpg
EOF
```

Copy & paste above snip in client for enabling repo

Install Kubernetes client in client (baston - host)

Installing Kubectl

```
yum-config-manager --enable kubernetes
yum install kubectl
curl -o kubectl
https://amazon-eks.s3-us-west-2.amazonaws.com/1.14.6/2019-08-22/bin/linux/amd64/kubectl
chmod +x ./kubectl
mkdir -p $HOME/bin && cp ./kubectl $HOME/bin/kubectl && export PATH=$HOME/bin:$PATH
kubectl version --short --client
```

Install AWS IAM Authenticator

```
 curl -o aws-iam-authenticator
https://amazon-eks.s3-us-west-2.amazonaws.com/1.14.6/2019-08-22/bin/linux/amd64/aws-iam-authent
icator

chmod +x ./kubectl
chmod +x ./aws-iam-authenticator
chmod  777 /root/bin/aws-iam-authenticator
```

To check Where AWS Credentials are stored and which keys are stored

→ #cat /root/.aws/credentails

To configure EKS cluster in node,

→ #aws eks update-kubeconfig --name <your eks name>

```
[root@ip-192-168-1-218 /]# aws eks update-kubeconfig --name myeks-cluster
Added new context arn:aws:eks:ap-northeast-1:642975937704:cluster/myeks-cluster to /root/.kube/config
Warning: aws-iam-authenticator is not installed properly or is not in your path.
Refer to the AWS Documentation to download it at https://docs.aws.amazon.com/eks/latest/userguide/configure-kubectl.html
[root@ip-192-168-1-218 /]#
```

EKS Cluster details & configuration is stored in blow dir

→ cat /root/.kube/config

To list all worker nodes available / attached to cluster

→ #kubectl get nodes

```
[root@ip-192-168-1-218 /]#
[root@ip-192-168-1-218 /]# kubectl get nodes
NAME                                              STATUS   ROLES    AGE   VERSION
ip-192-168-1-113.ap-northeast-1.compute.internal  Ready    <none>   53m   v1.14.7-eks-1861c5
ip-192-168-1-241.ap-northeast-1.compute.internal  Ready    <none>   54m   v1.14.7-eks-1861c5
[root@ip-192-168-1-218 /]#
```

to list all pods available in cluster

→ #kubectl get pods → for listing user defined

→ #kubectl get pods --all-namespaces → for listing all

```
[root@ip-192-168-1-218 /]#
[root@ip-192-168-1-218 /]# kubectl get pods --all-namespaces
NAMESPACE     NAME                              READY   STATUS    RESTARTS   AGE
kube-system   aws-node-8gkdv                    1/1     Running   0          57m
kube-system   aws-node-dc8tq                    1/1     Running   0          57m
kube-system   coredns-58986cd576-2775q          1/1     Running   0          65m
kube-system   coredns-58986cd576-cvlws          1/1     Running   0          65m
kube-system   kube-proxy-5492b                  1/1     Running   0          57m
kube-system   kube-proxy-m4fnq                  1/1     Running   0          57m
[root@ip-192-168-1-218 /]#
```

Userful Commands

1. Kubectl get nodes
2. Kubectl get pods
3. Kubectl describe deployment <deployment file name>
4. Kubectl describe pods <pod name>
5. Kubectl apply -f <deployment file>
6. Kubectl apply -f <service file name>
7. Kubectl get deployments
8. Kubectl get svc
9. Kubectl describe pods <pod name>
10. Kubectl delete deployment <deployment name> (if you delete deployment, pods will be deleted)
11. Kubectl delete svc <service name>
12. Kubectl edit deployment <deployment name>
13. Kubectl get nodes --watch