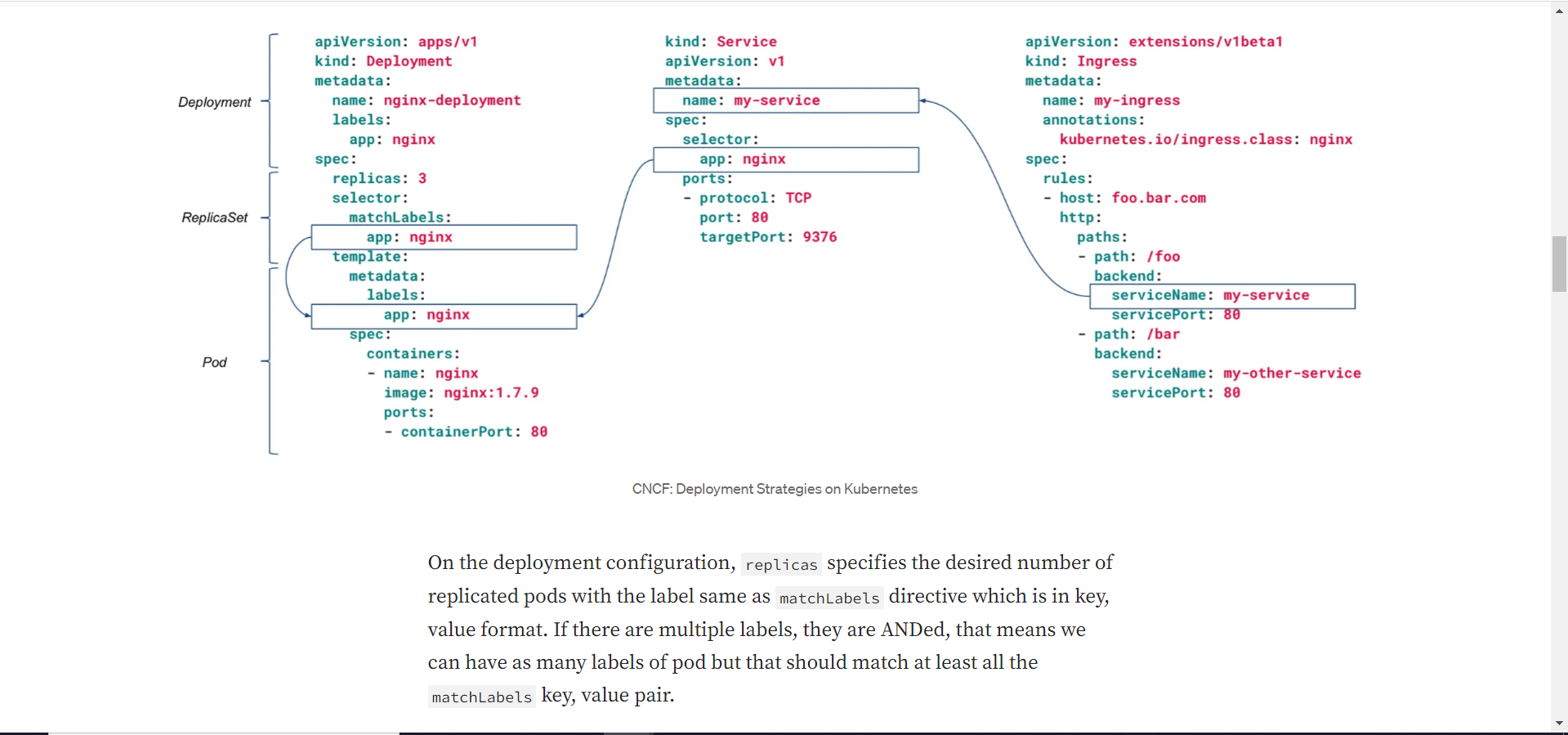
Relationship b/w deployment – service – ingress

https://dwdraju.medium.com/how-deployment-service-ingress-are-related-in-their-manifest-a2e553cf0ffb



AWS ec2 to eks cluster auth via iam role:

AWS AND KUBECTL INSTALLATION:

<https://docs.aws.amazon.com/cli/latest/userguide/getting-started-install.html> -as per version of EKS cluster

<https://docs.aws.amazon.com/eks/latest/userguide/install-kubectl.html> - as per version of EKS cluster

* aws sts assume-role --role-arn arn:aws:iam::858728894252:role/**nextwork-pre-staging-eks-admin** --role-session-name eksauthtest1

**OR**

AWS keys login from console which has access to assume eks-admin role.

take the token and export them as below :

A black background with white text

Description automatically generated

A black screen with text

Description automatically generated

Then run below commands on ec2 machine from where need to access eks:

* aws eks update-kubeconfig --region eu-central-1 --name nextwork-pre-staging-eks --role-arn arn:aws:iam::858728894252:role/nextwork-pre-staging-eks-admin

[root@ip-10-0-140-17 bin]# aws sts assume-role --role-arn arn:aws:iam::858728894252:role/nextwork-pre-staging-eks-admin --role-session-name eksauthtest1

{

"Credentials": {

"AccessKeyId": "ASIA4P4B3B4WHD57SBUT",

"SecretAccessKey": "kdEZK298ol72MtmDivnCDlN8u9JI0a9BLryQDXFI",

"SessionToken": "",

"Expiration": "2024-01-19T12:59:23+00:00"

},

"AssumedRoleUser": {

"AssumedRoleId": "AROA4P4B3B4WCUZUBOLUR:eksauthtest1",

"Arn": "arn:aws:sts::858728894252:assumed-role/nextwork-pre-staging-eks-admin/eksauthtest1"

}

}

[root@ip-10-0-140-17 bin]# export AWS\_SECRET\_ACCESS\_KEY=ASIA4P4B3B4WHD57SBUT

[root@ip-10-0-140-17 bin]# export AWS\_SECRET\_ACCESS\_KEY=kdEZK298ol72MtmDivnCDlN8u9JI0a9BLryQDXFI

[root@ip-10-0-140-17 bin]# export AWS\_SESSION\_TOKEN=IQoJb3JpZ2luX2VjEEQaDGV1LWNlbnRyYWwtMSJHMEUCIBa9YEKqfyqgzLAwdSrH9zZYq+BtKpLhbnY8PNTsExPwAiEAz9zDL0Cci21IZT/WpeRAYvYWh/K5LjmTbgv2BxLwYKoqogII7f//////////ARAAGgw4NTg3Mjg4OTQyNTIiDIFXp6IAdB/FJjCLrCr2Ae0JbFM0Q8+2hTweSnaGrZq2gCNz5CltccOOMkKkldYX9ygwzwsG6oaSeAfEXQJGWipY3lcQ0q2e30FGE4nO[root@ip-10-0-140-17 bin]#

BASTION HOST IAM ROLE :

Bastion host role should be allowed to assume EKS-admin role which happens after we attached below policies to bastion IAM role.

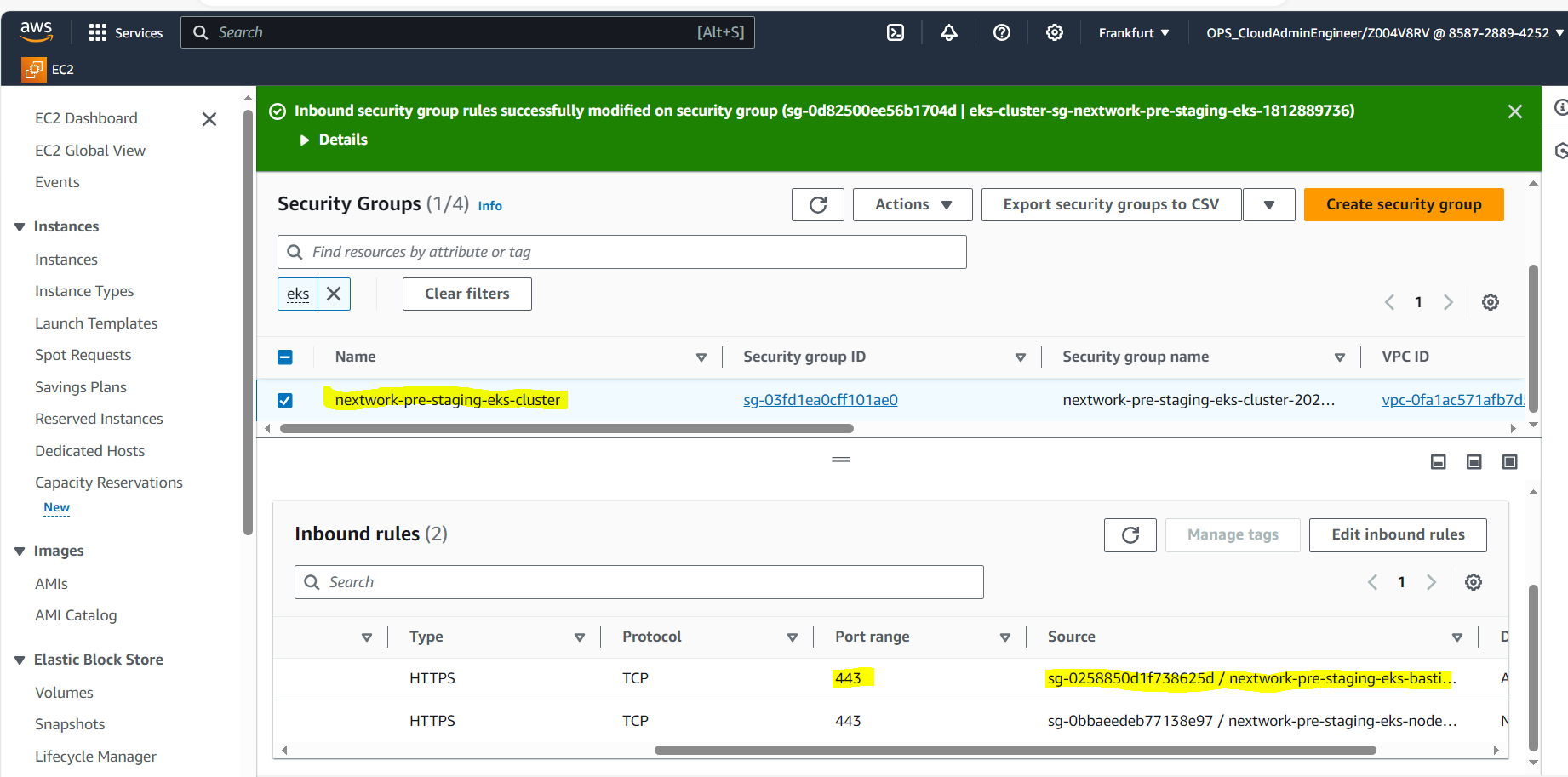
* aws sts assume-role --role-arn **arn:aws:iam::858728894252:role/nextwork-pre-staging-eks-admin** --role-session-name eksauthtest1

Below attached policies **on bastion iam role** assumes eks role **arn:aws:iam::858728894252:role/nextwork-pre-staging-eks-admin else it will not work.**

A screenshot of a computer

Description automatically generated

Note Bastion host should be allowed to have traffic on EKS (i.e sg on eks should be allowed ingress from bastion host on ports which need to be accessed by bastion on eks cluster eg: 443 only required)



Test Deployment :

---  
apiVersion: apps/v1  
kind: Deployment  
metadata:  
  name: echoserver  
  namespace: default  
spec:  
  selector:  
    matchLabels:  
      app: echoserver  
  replicas: 1  
  template:  
    metadata:  
      labels:  
        app: echoserver  
    spec:  
      containers:  
      - image: k8s.gcr.io/e2e-test-images/echoserver:2.5  
        name: echoserver  
        ports:  
        - containerPort: 8080  
---  
apiVersion: v1  
kind: Service  
metadata:  
  name: echoserver  
  namespace: default  
spec:  
  ports:  
  - port: 8080  
    protocol: TCP  
  type: NodePort  
  selector:  
    app: echoserver  
A screen shot of a computer

Description automatically generated

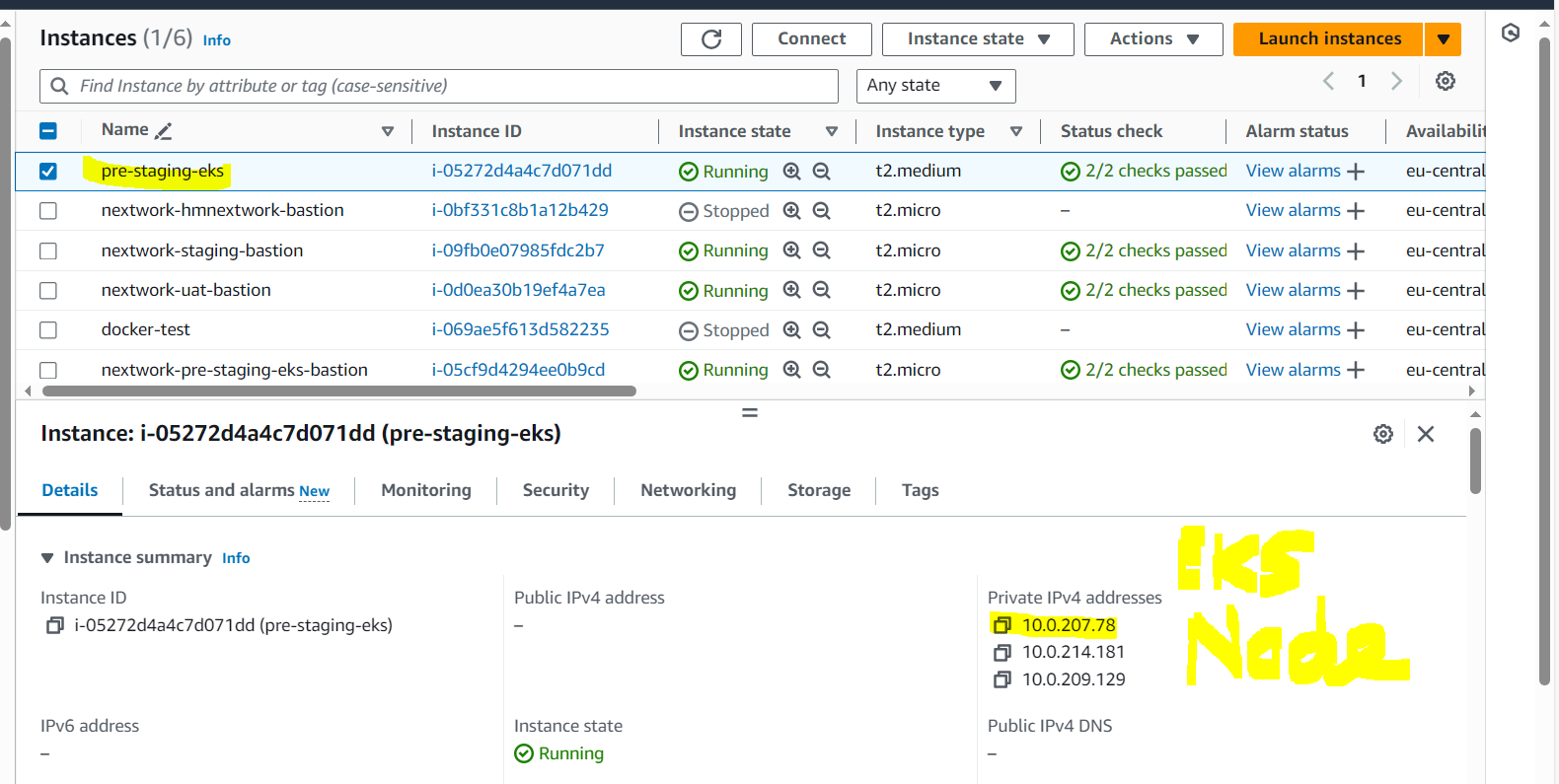
Go to NODE ec2 instance and get its private ip and application will be available on nodeport defined from the Ec2 machine from where we are accessing eks cluster.

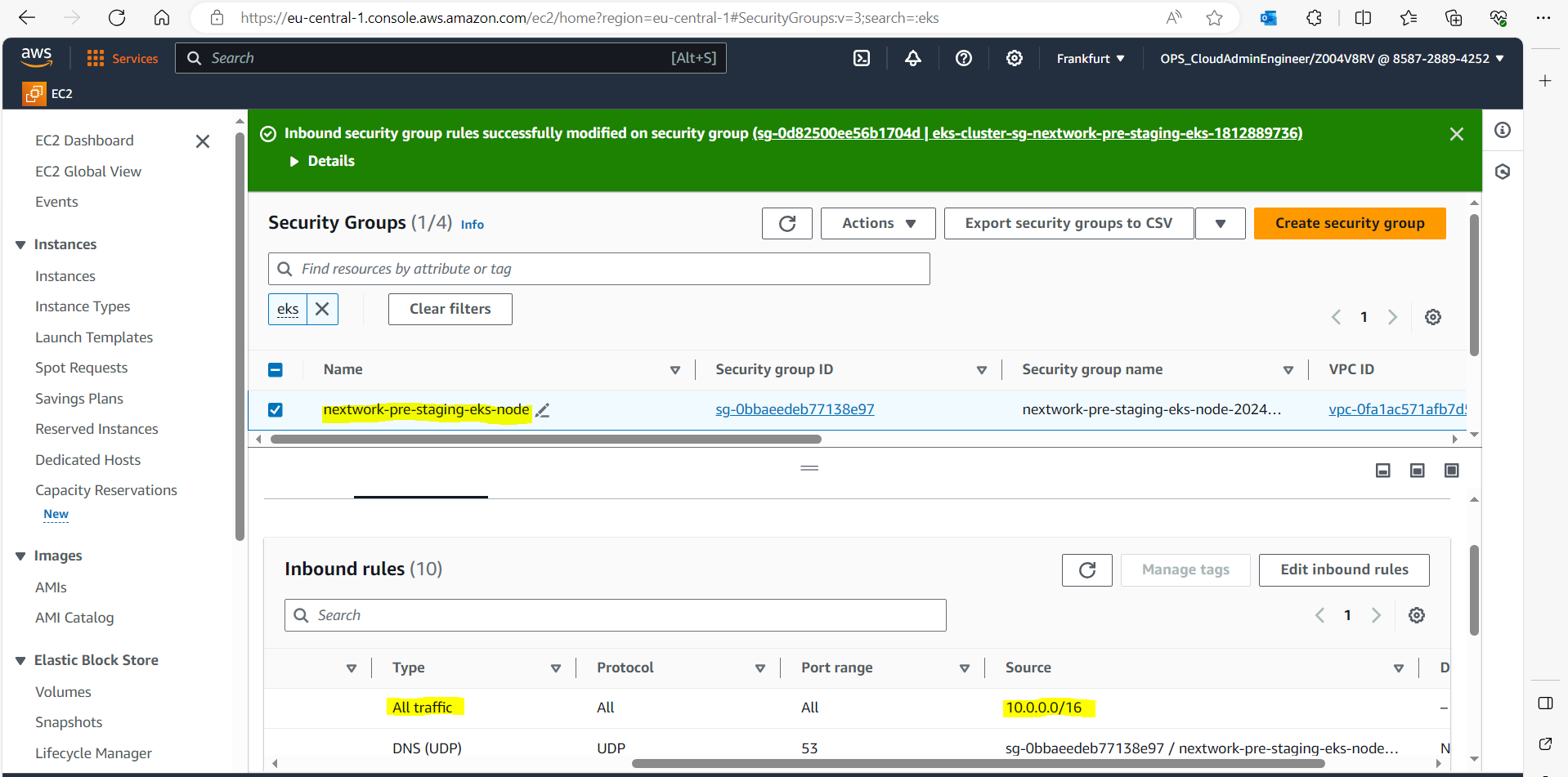
From bastion host below commands will behave as :

wget 10.0.207.78:31074 – will work

wget 10.0.207.78:8080 – not work

wget 172.20.43.174:31074 and wget 172.20.43.174:8080 from bastion will not work as its eks local ips not node ips they will work locally on going to nodes itself.



Node sg rule should be added for bastion host : (NEED TO UPDATE IN TERRAFORM CODE)  


With this port(443) open only for ec2(bastion host) on eks sg we can run below commands on bastion ec2 host

After deploying UI service we can check as below(deployment ports as 443 in service and deployment yaml) :  
Running below command from Bastion host

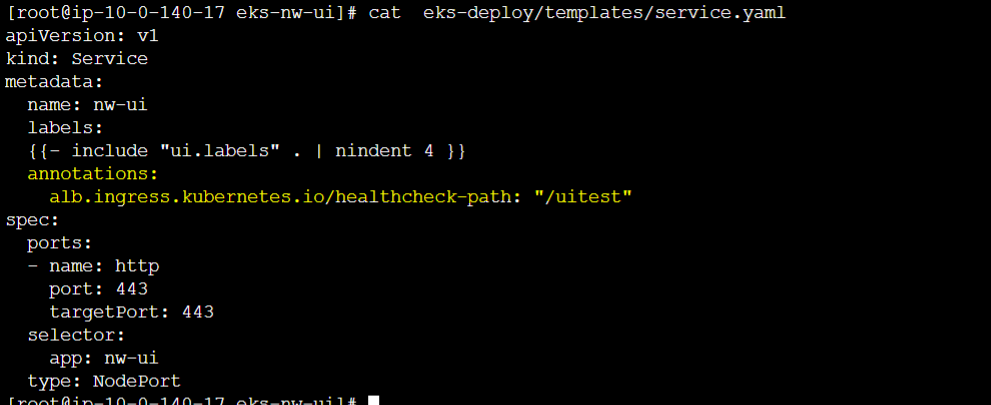
wget https://10.0.218.146:31205 --no-check-certificate( no check is like allowing self signed on browser)

EKS **node ip** - 10.0.218.146 and 31205 **( node port** )

A screenshot of a computer program

Description automatically generated

Health check for multiple microservices is set at service level rather than on Ingress level(for single service it can be on service level)



[root@ip-10-0-140-17 eks-nw-ui]# cat eks-deploy/templates/service.yaml

apiVersion: v1

kind: Service

metadata:

name: nw-ui

labels:

{{- include "ui.labels" . | nindent 4 }}

annotations:

alb.ingress.kubernetes.io/healthcheck-path: "/uitest"

spec:

ports:

- name: http

port: 443

targetPort: 443

selector:

app: nw-ui

type: NodePort

===TLS generate jks as per the document and copy in docker container from repo====

<https://docs.aws.amazon.com/documentdb/latest/developerguide/connect_programmatically.html#connect_programmatically-tls_enabled>

<https://github.com/brianfrankcooper/YCSB/issues/1314>