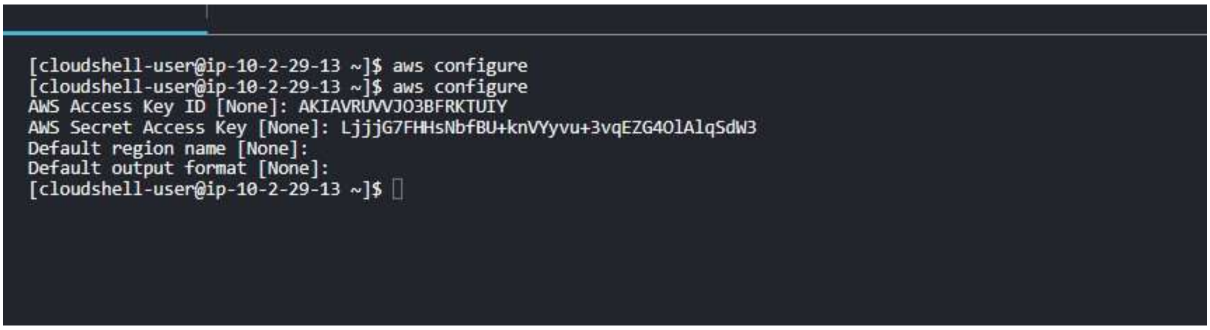
TASK = Pod and services creating using manifest file

NAME = Akshay choudhary

Step 1

Create an EKS cluster & Node group

- Use cloudshell



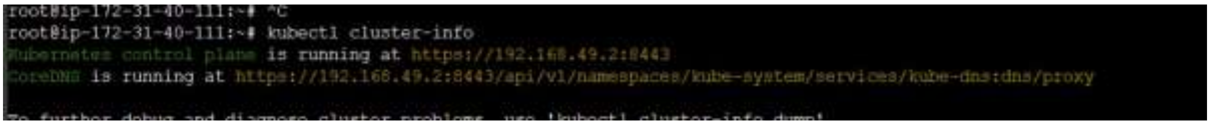
```
[cloudshell-user@ip-10-2-29-13 ~]$ aws configure
[cloudshell-user@ip-10-2-29-13 ~]$ aws configure
AWS Access Key ID [None]: AKIAVRUVVJO3BFRKTUIY
AWS Secret Access Key [None]: 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
Default region name [None]:
Default output format [None]:
[cloudshell-user@ip-10-2-29-13 ~]$ 
```

- Connect the cluster
- Aws eks update-kubeconfig –region us east-2 –name Eks-cluster(clustername)

Step 2

Check nodes& cluster-info

- Kubectl get nodes
- Kubectl cluster-info
-



```
root@ip-172-31-40-111:~# ^C
root@ip-172-31-40-111:~# kubectl cluster-info
Kubernetes control plane is running at https://192.168.49.2:8443
CoreDNS is running at https://192.168.49.2:8443/api/v1/namespaces/kube-system/services/kube-dns:dns/proxy

To further debug and diagnose cluster problems, use 'kubectl cluster-info dump'
```

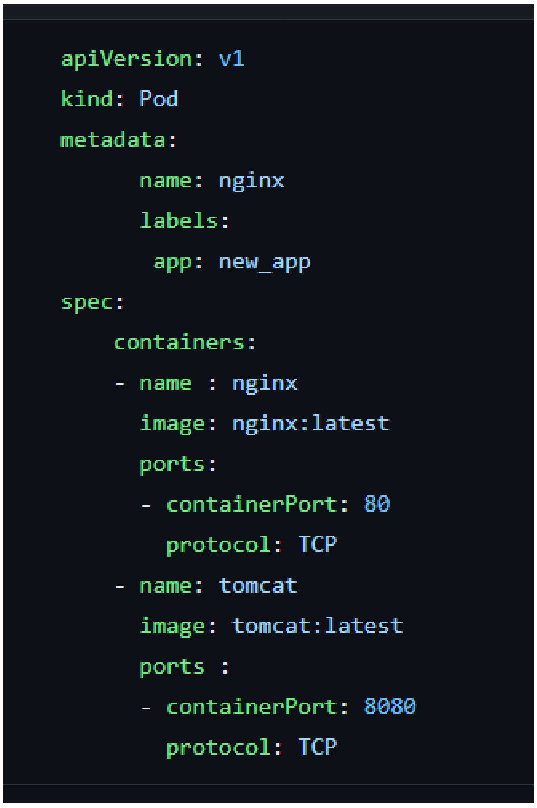
Step 3

Create   manifest file(exec.   Yml.)

- Pod.yml
- Services.yml

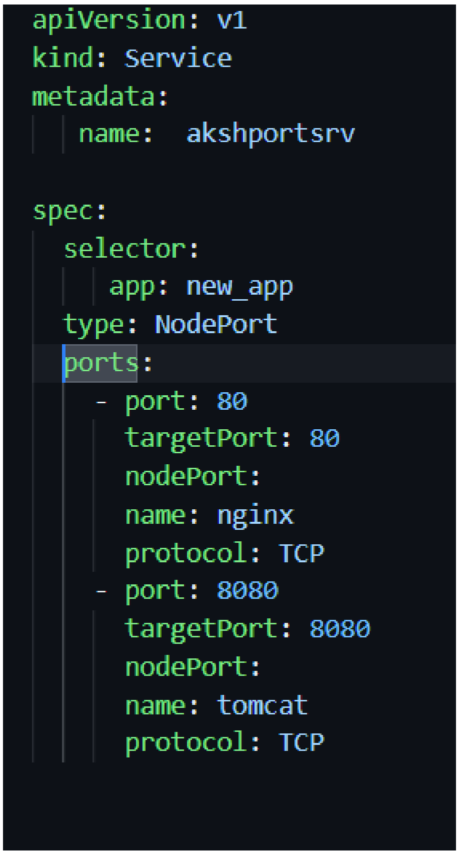
Step 4

Create a pods.yml file



```
apiVersion: v1
kind: Pod
metadata:
      name: nginx
      labels:
       app: new_app
spec:
    containers:
    - name : nginx
      image: nginx:latest
      ports:
      - containerPort: 80
        protocol: TCP
    - name: tomcat
      image: tomcat:latest
      ports :
      - containerPort: 8080
        protocol: TCP
```

Services.yml (file)



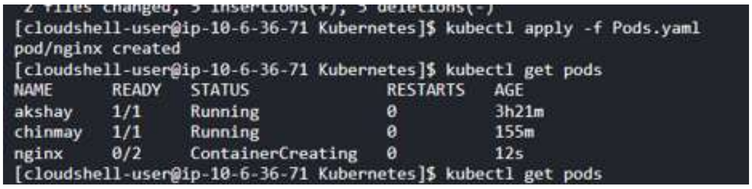
```
apiVersion: v1
kind: Service
metadata:
   name:  akshportsrv

spec:
  selector:
     app: new_app
  type: NodePort
  ports:
    - port: 80
      targetPort: 80
      nodePort:
      name: nginx
      protocol: TCP
    - port: 8080
      targetPort: 8080
      nodePort:
      name: tomcat
      protocol: TCP
```

Step 5

- This is pos.ym & services.yml file in create in my repo
- Git clone https://github.com/Akshay-7693032755/Kubernetes.git

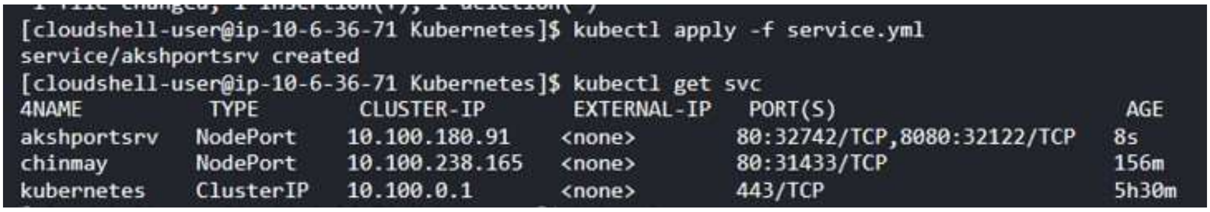
Step 6

- Using manifest file
- Pods.yml
- Kubectl apply -f posd.yml
- Kubectl get pods



```
 2 files changed, 5 insertions(+), 5 deletions(-)
[cloudshell-user@ip-10-6-36-71 Kubernetes]$ kubectl apply -f Pods.yaml
pod/nginx created
[cloudshell-user@ip-10-6-36-71 Kubernetes]$ kubectl get pods
NAME      READY   STATUS              RESTARTS   AGE
akshay    1/1     Running             0          3h21m
chinmay   1/1     Running             0          155m
nginx     0/2     ContainerCreating   0          12s
[cloudshell-user@ip-10-6-36-71 Kubernetes]$ kubectl get pods
```

- Services.yml
- Kubectl apply -f services.yml
- Kubectl get services

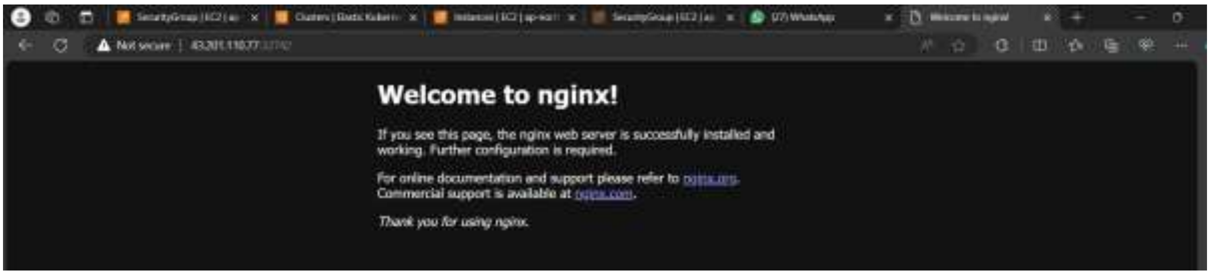


```
 1 file changed, 1 insertion(+), 1 deletion(-)
[cloudshell-user@ip-10-6-36-71 Kubernetes]$ kubectl apply -f service.yml
service/akshportsrv created
[cloudshell-user@ip-10-6-36-71 Kubernetes]$ kubectl get svc
4NAME         TYPE        CLUSTER-IP       EXTERNAL-IP   PORT(S)                         AGE
akshportsrv   NodePort    10.100.180.91    <none>        80:32742/TCP,8080:32122/TCP     8s
chinmay       NodePort    10.100.238.165   <none>        80:31433/TCP                    156m
kubernetes    ClusterIP   10.100.0.1       <none>        443/TCP                         5h30m
```

Step 7

- Instance IP copy & past in browser
- Out put (first   nginx (2)   tomcat

NGINX PAGE HOST



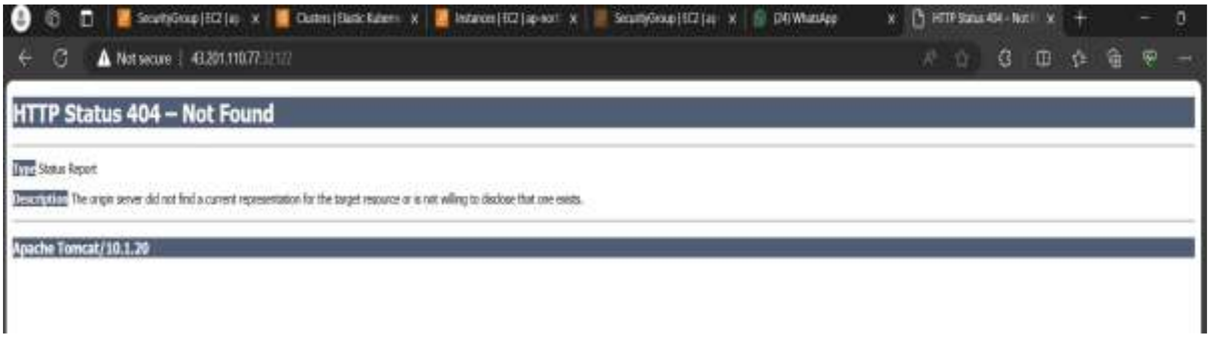
Welcome to nginx!

If you see this page, the nginx web server is successfully installed and working. Further configuration is required.

For online documentation and support please refer to nginx.org. Commercial support is available at nginx.com.

Thank you for using nginx.

TOMCAT PAGE HOST



HTTP Status 404 – Not Found

Type Status Report

Description The origin server did not find a current representation for the target resource or is not willing to disclose that one exists.

Apache Tomcat/10.1.20