**31.AWS-B30-EFS-AWSBackup**

--- **Note** – you have application deployed on multiple availability zones. That application wants to access same type of data then we will use AWS EFS.

--- aws EFS is shared file system.

**Architecture**

--- I have an application deployed on 3 different zones. AWS EFS is mounted on these servers.

--- **note** – we can use AWS EFS between multiple availability zones.

--- **use cases for AWS EFS.**

1. Centralized logs or Backups or APP data.
2. Kubernetes shared storage
3. Master and slave applications. Eg we can use aws EFS in jenkins master slave architecture.

**Prerequisites for testing NFS**

--- **note** - I will create 3 servers in 3 different availability zones with below user date.

--- **startup script**

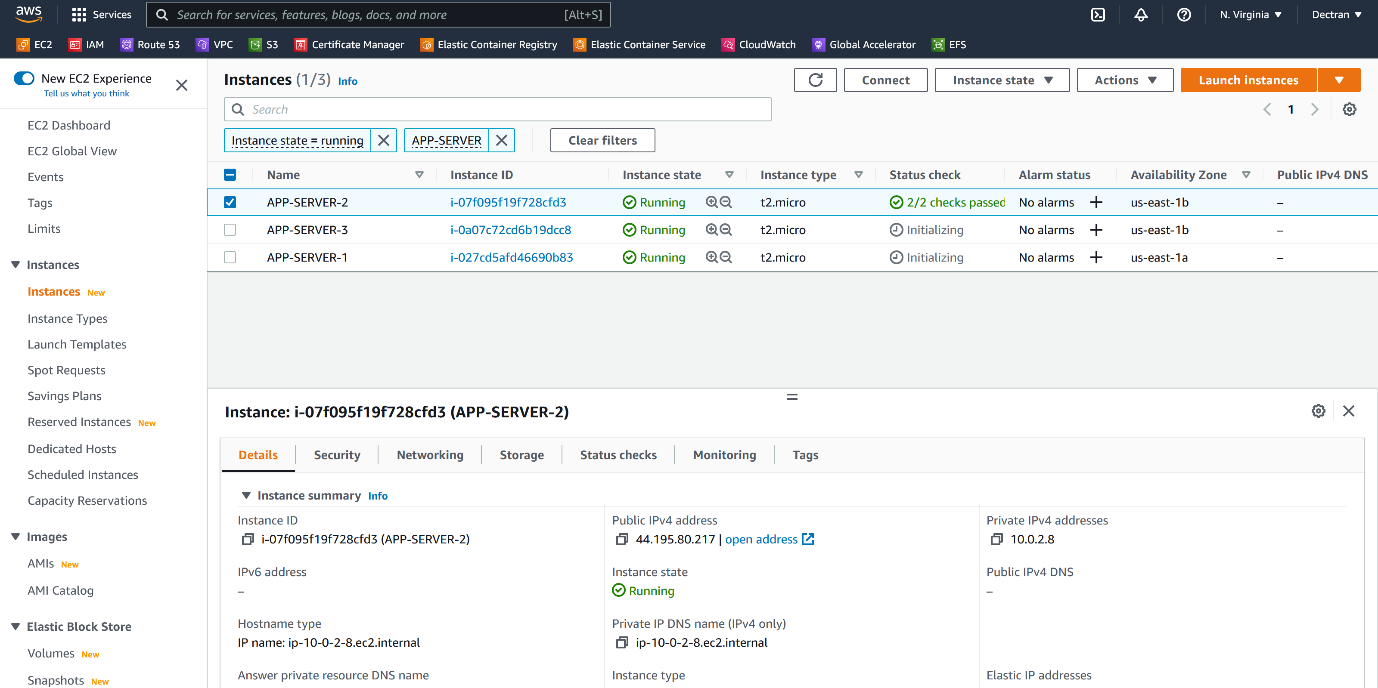
#!/bin/bash

apt update -y

apt install -y openjdk-8-jdk

apt install -y nfs-common

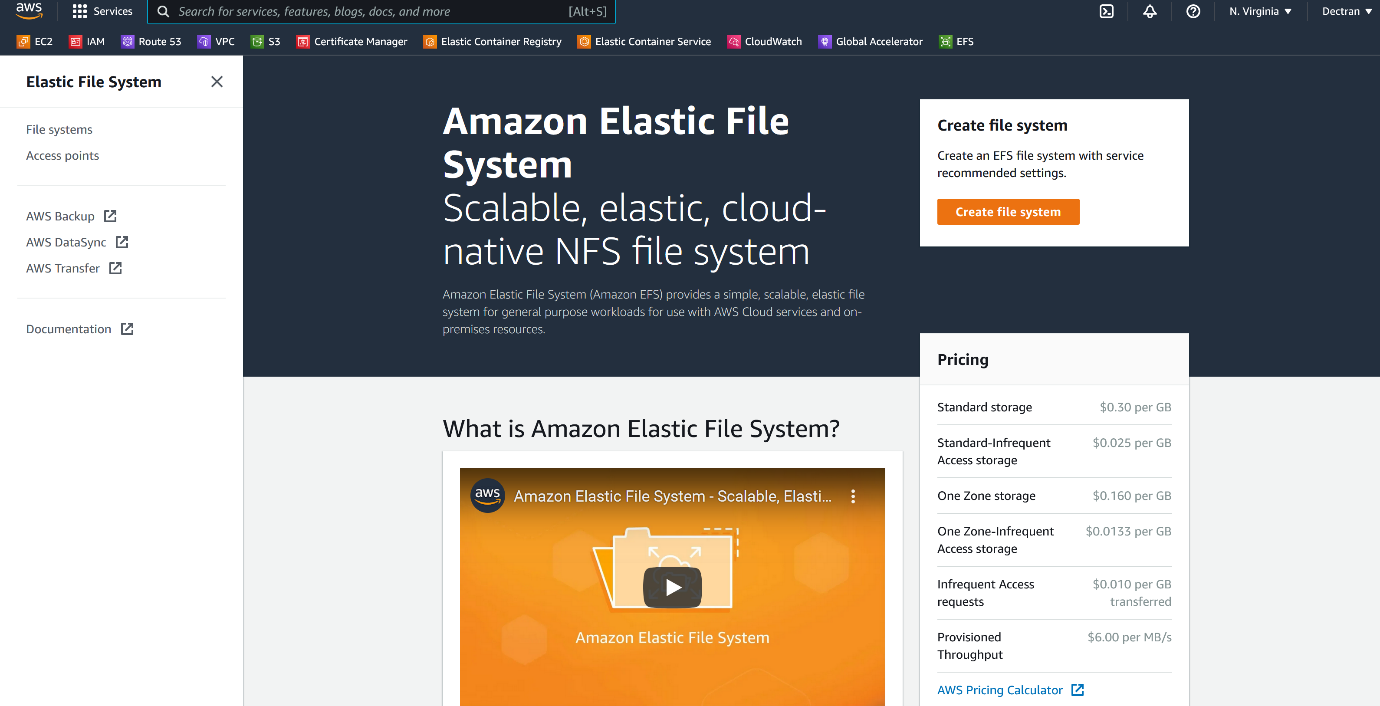
--- **NOTE** – provision the instance with above data.



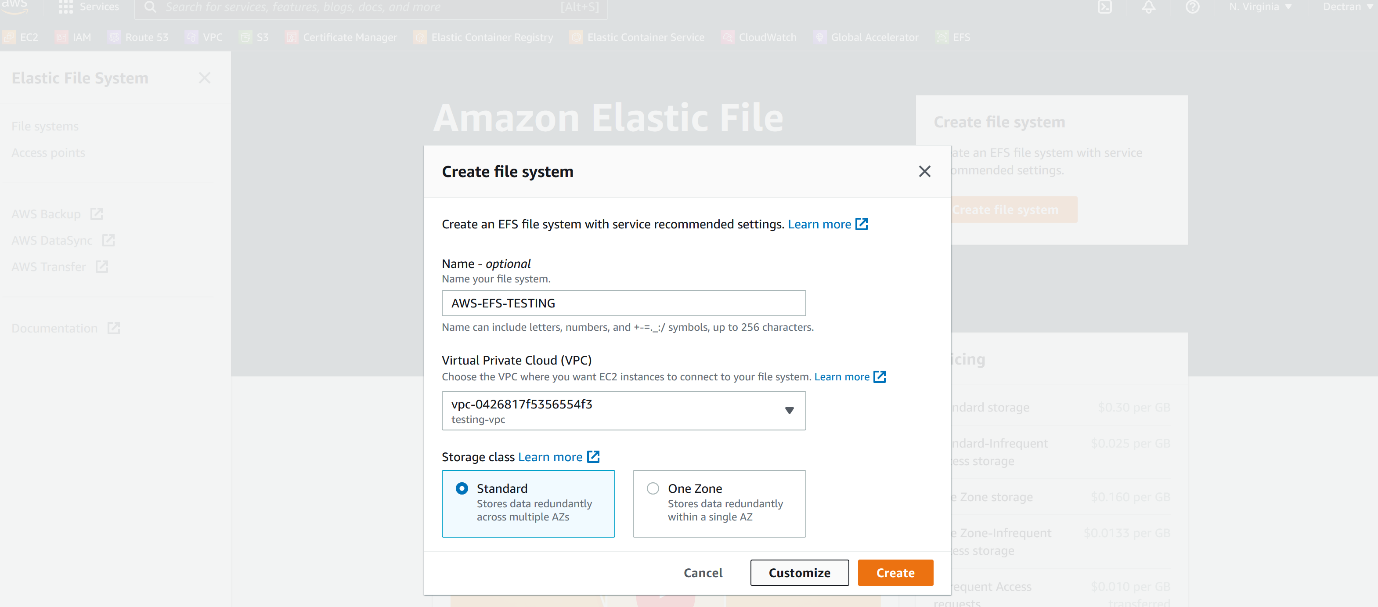
--- **note** – all the 3 servers are provisioned in 3 different zones.

**Create EFS file system**

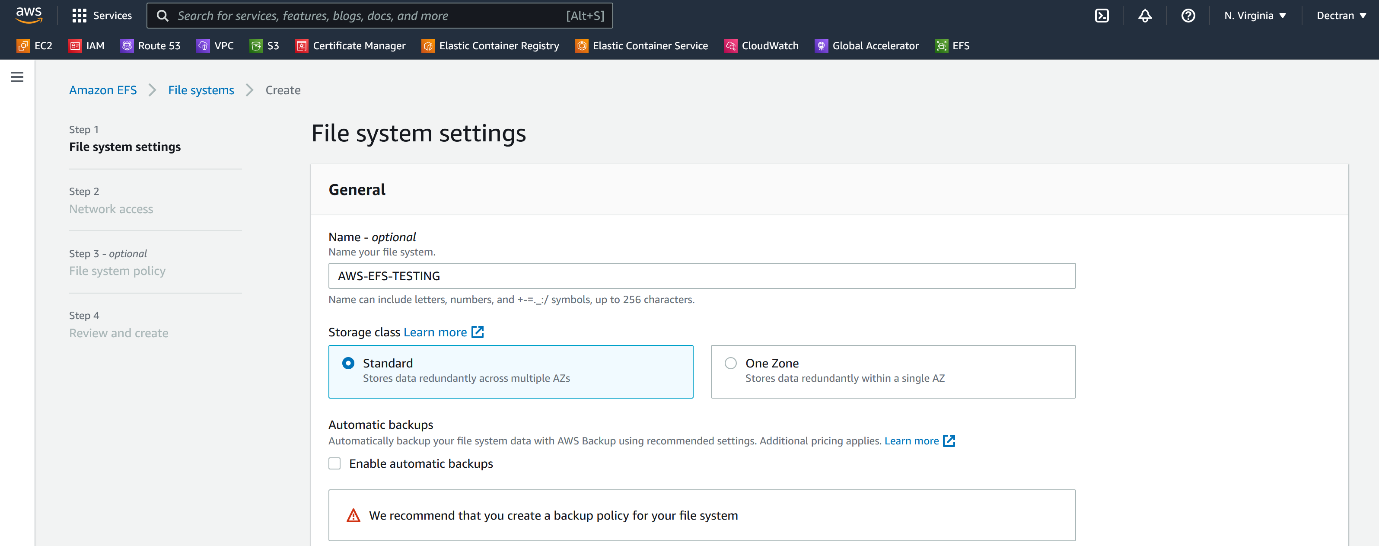
--- go to AWS EFS.

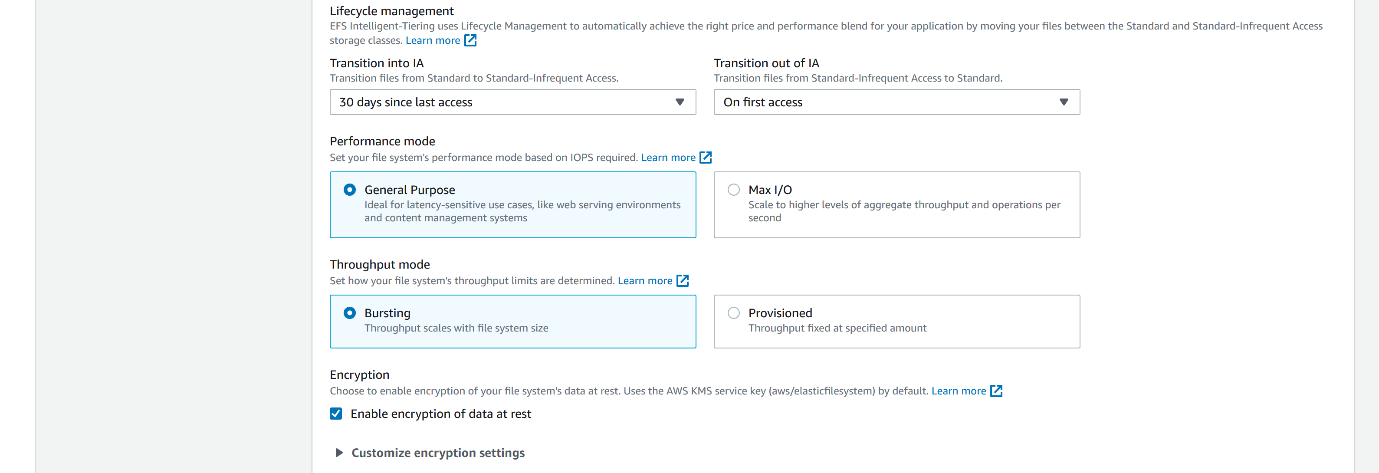


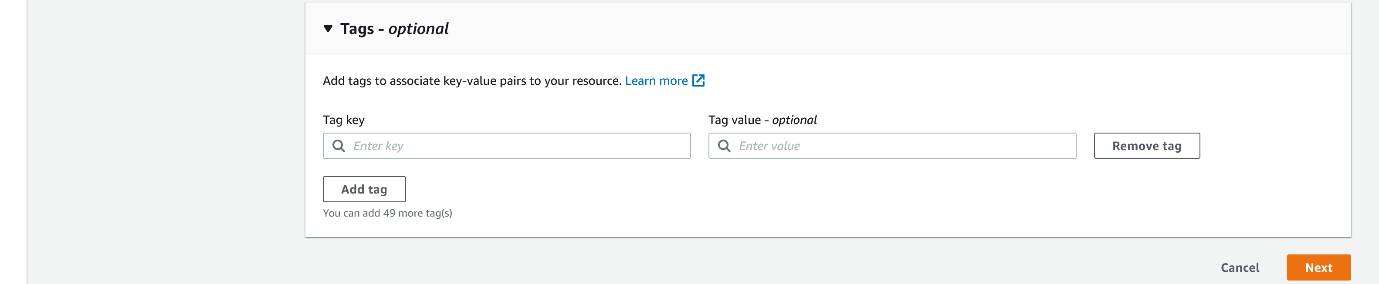
--- click on create file system



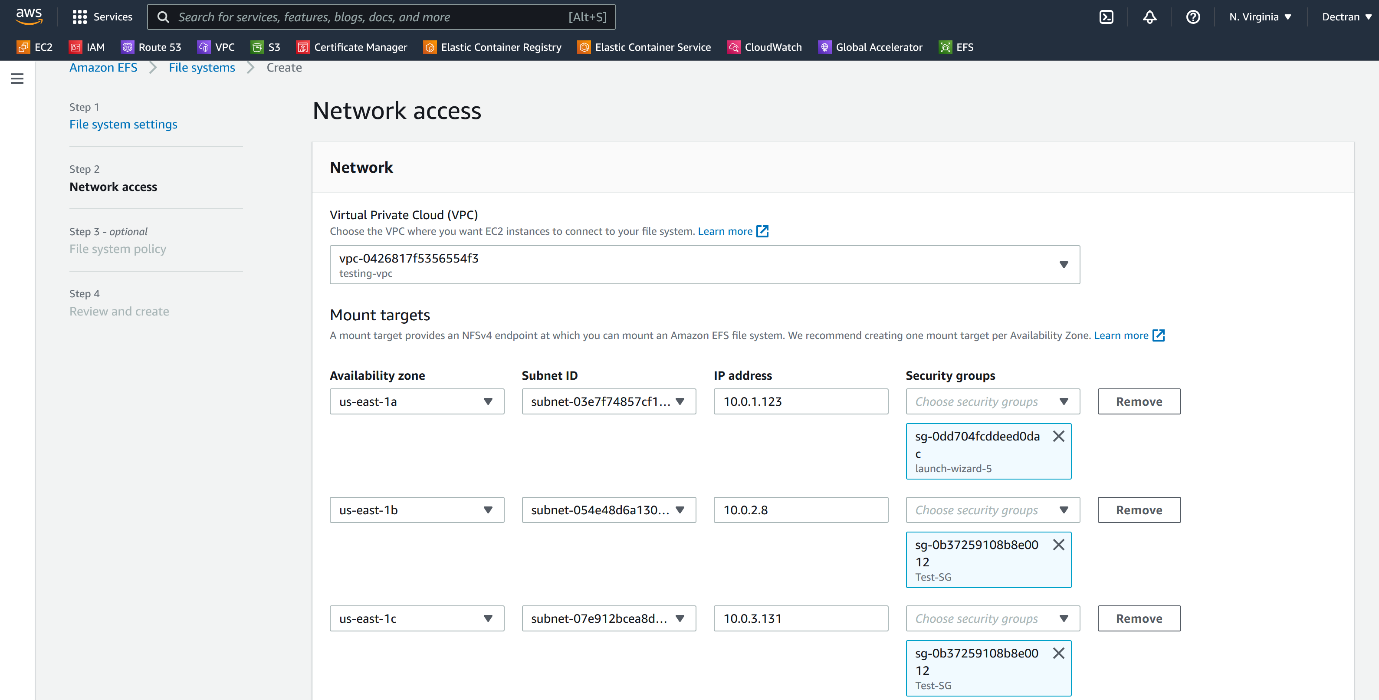
--- click on customize.

--- if you want, please check the automatic backups.



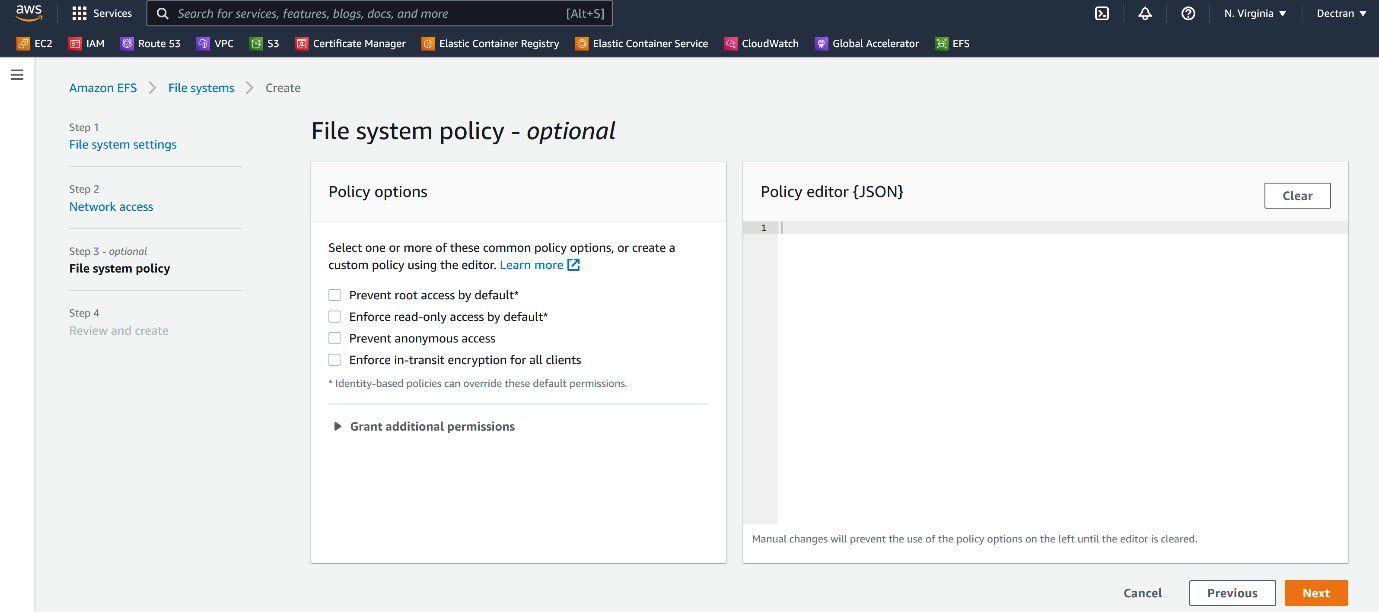


--- click on next.

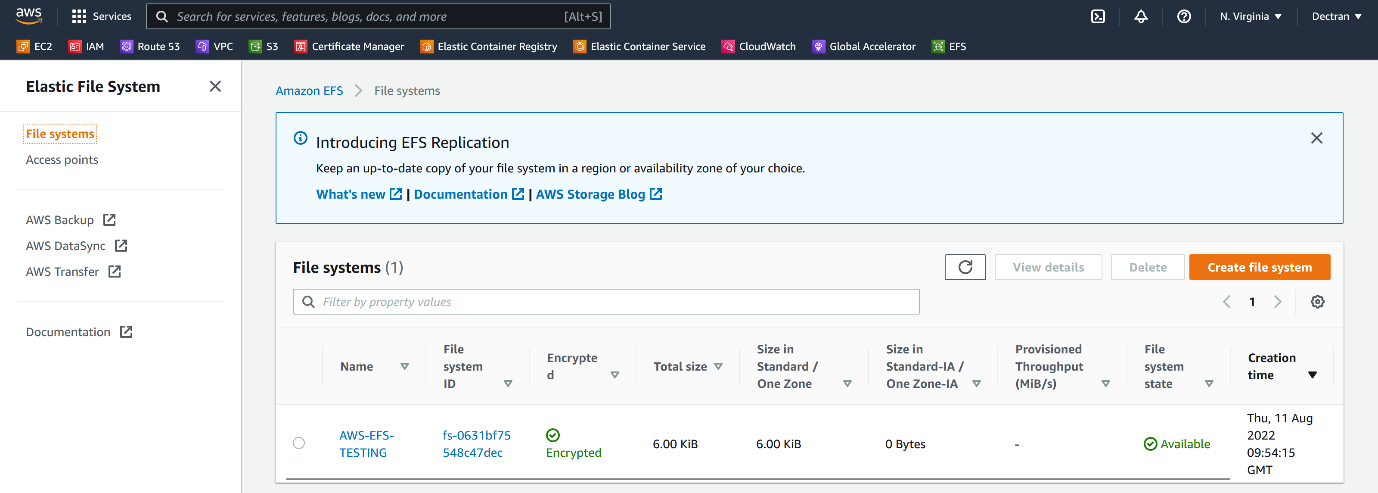


--- **note** – you do not need to give ip addresses of the servers, ip address automatically taken and please select instance security group. Here, I added private ip address of instance.

--- click on next.



--- click on next and click on create.



--- **note** – the EFS file system got created.

**Server side**

--- note - please login to 3 servers and create the below directory in all servers.

**# Create /var/lib/jenkins directory in all 3 servers.**

--- mkdir -p /var/lib/jenkins

# List mounting paths on the servers.

--- df -h



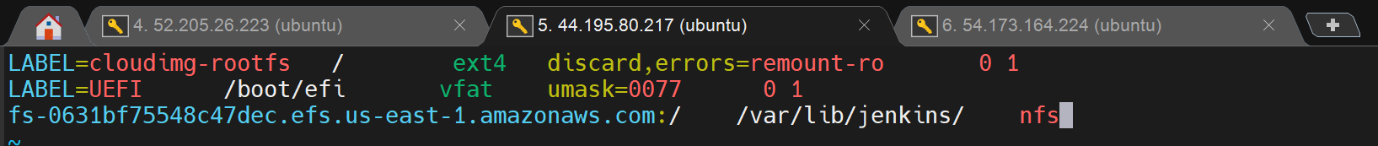
--- note - the efs is not mounted for /var/lib/jenkins directory.

**/etc/fstab**

**# copy this command to /etc/fstab**

--- fs-0631bf75548c47dec**.efs.us-east-1.amazonaws.com:/ /var/lib/jenkins/ nfs**

**--- vi /etc/fstab**



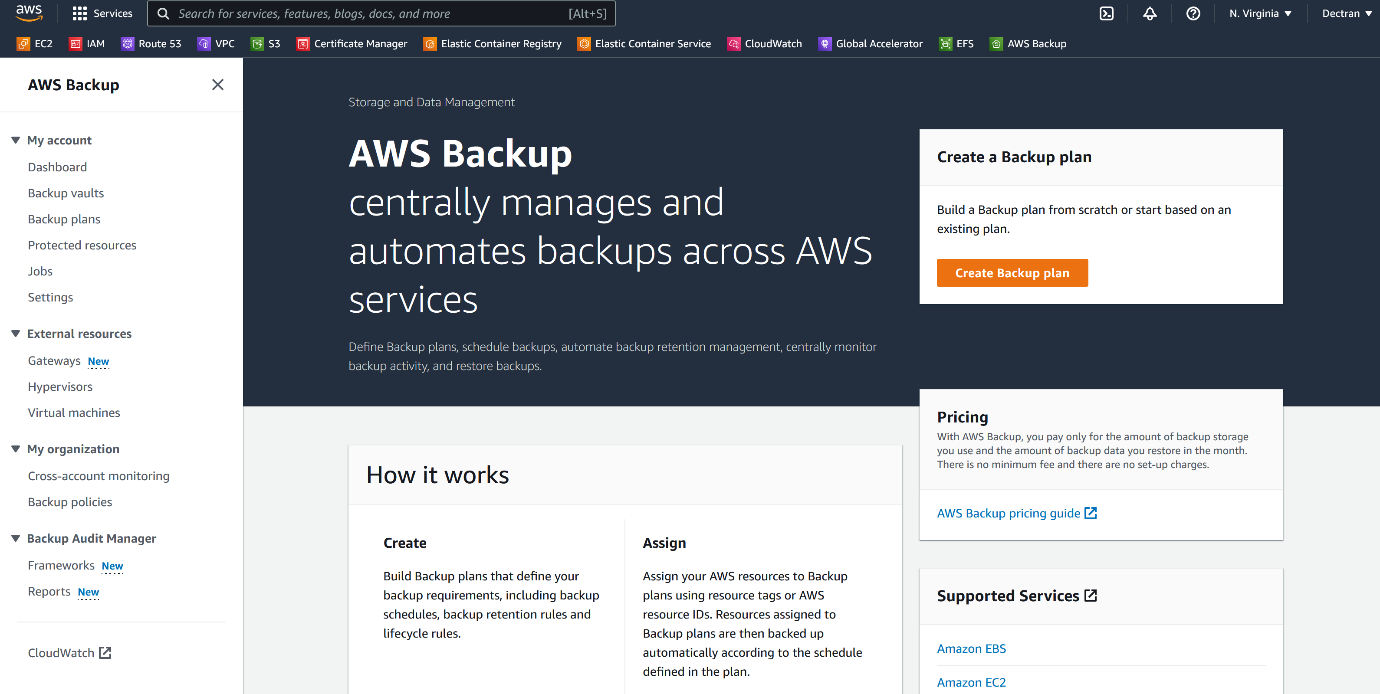
**--- note – copy this command to all 3 servers.**

**# Mount a volume**

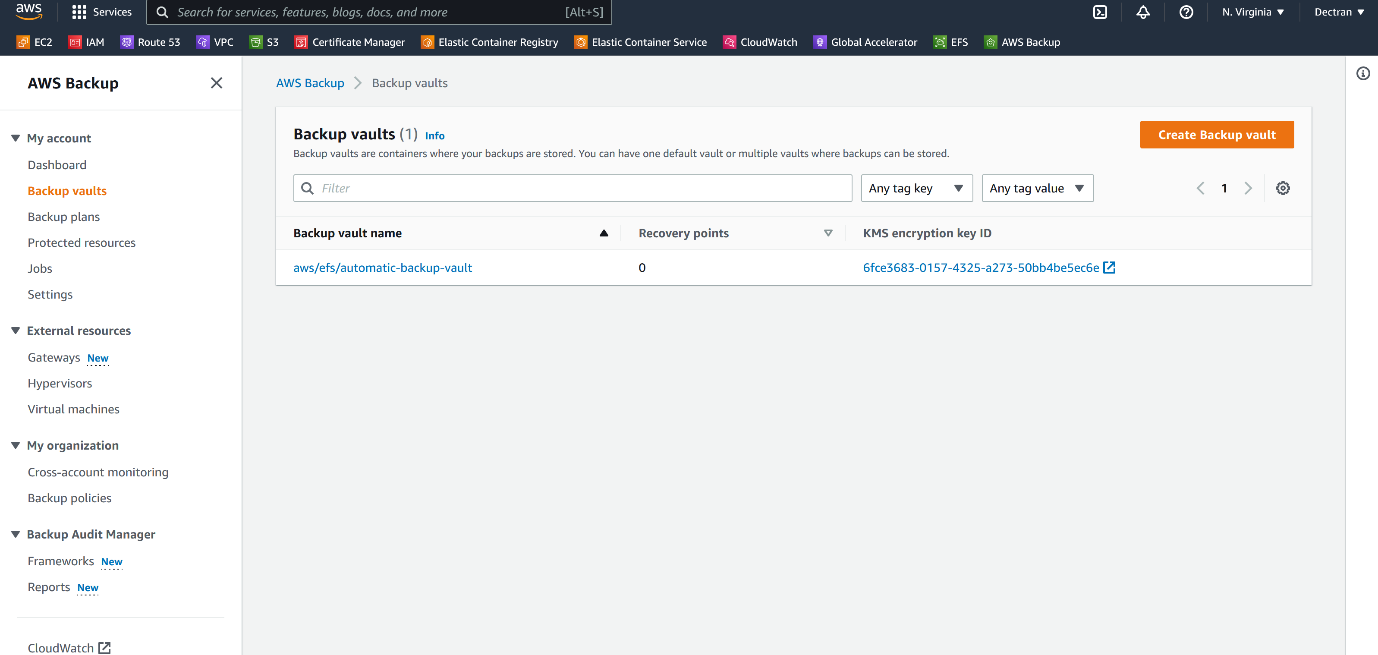
**--- mount -a**

**AWS BAKCUPS**

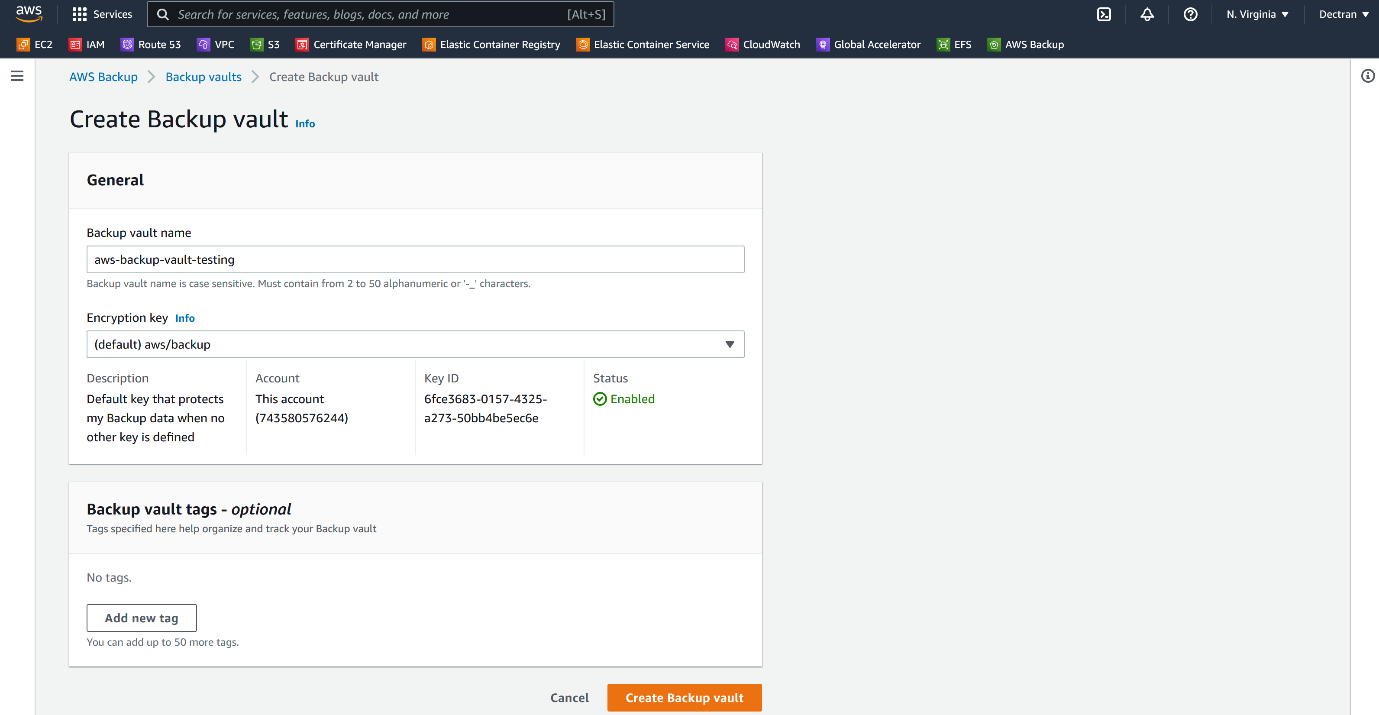
**--- go to aws backups**



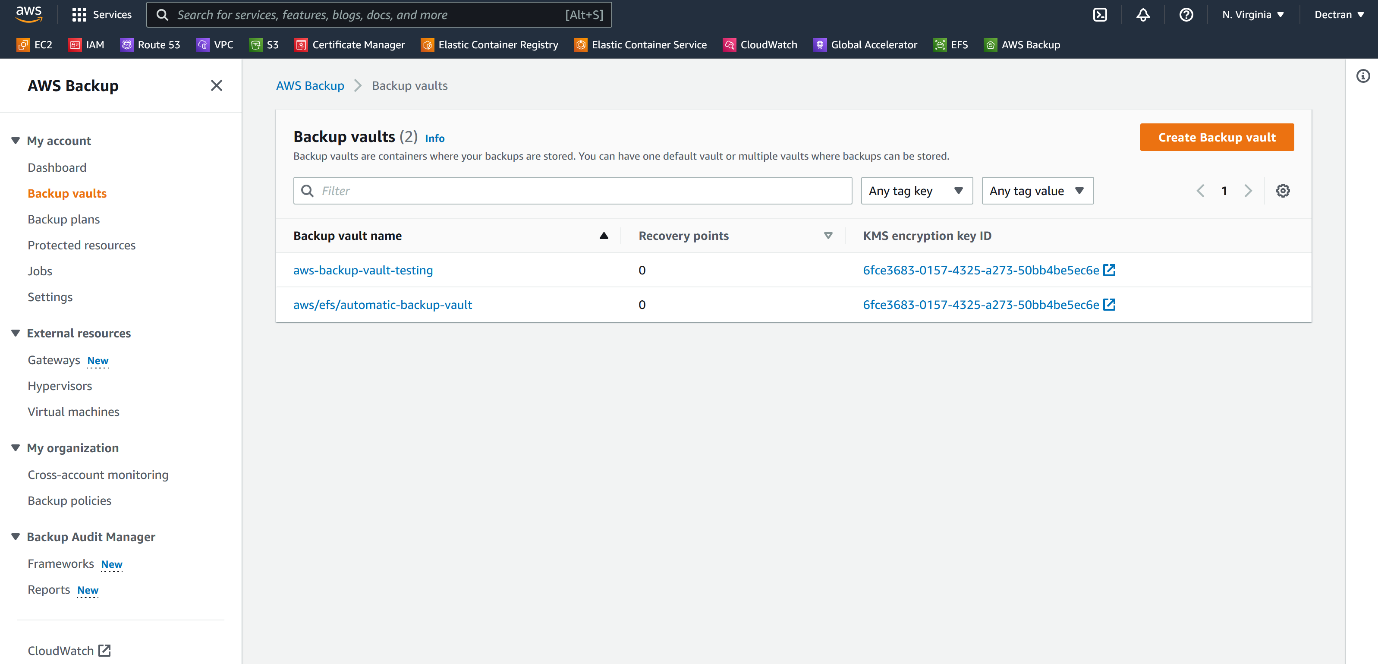
**Create Backup vaults**



**--- click on create backup valut.**

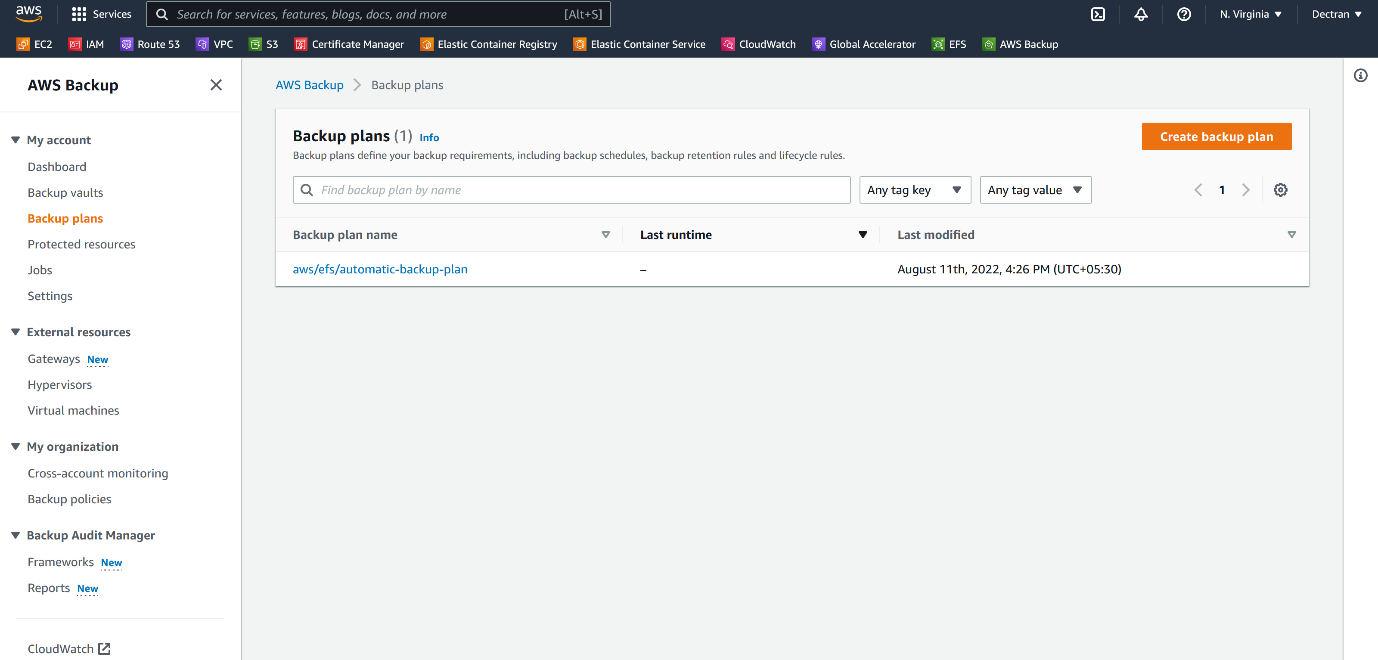


**--- I am using default encryption key.**

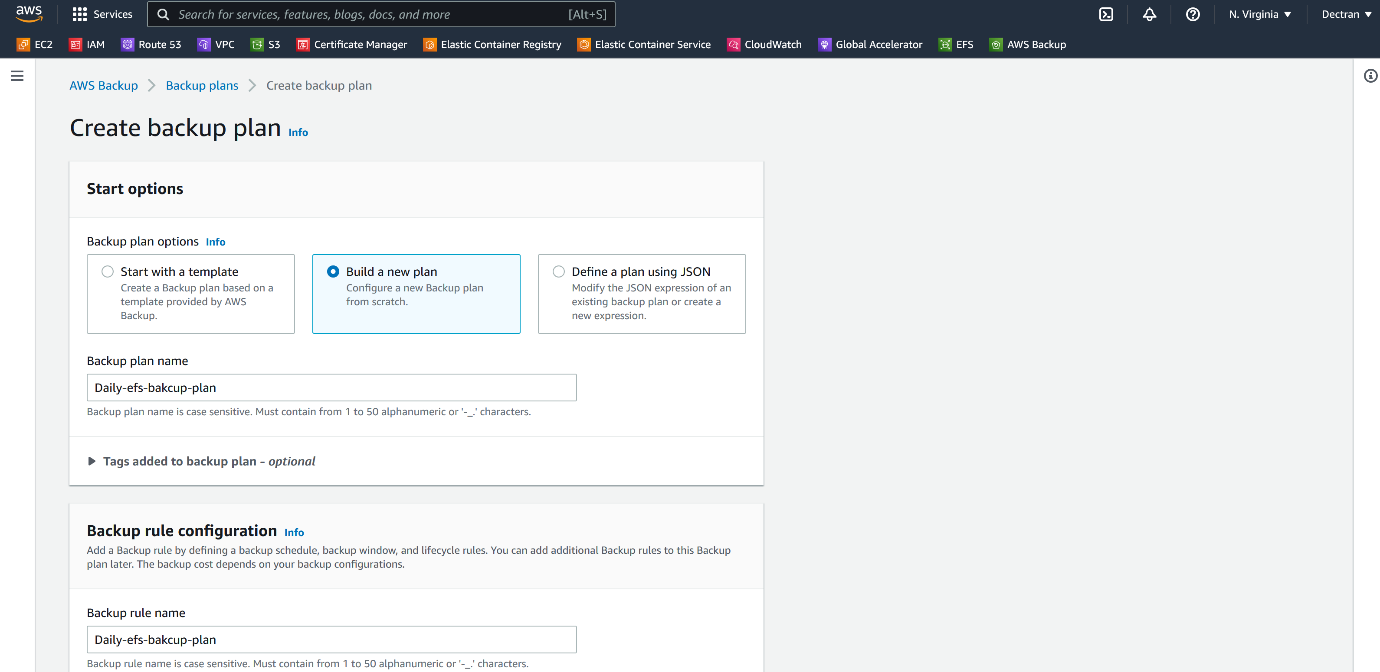


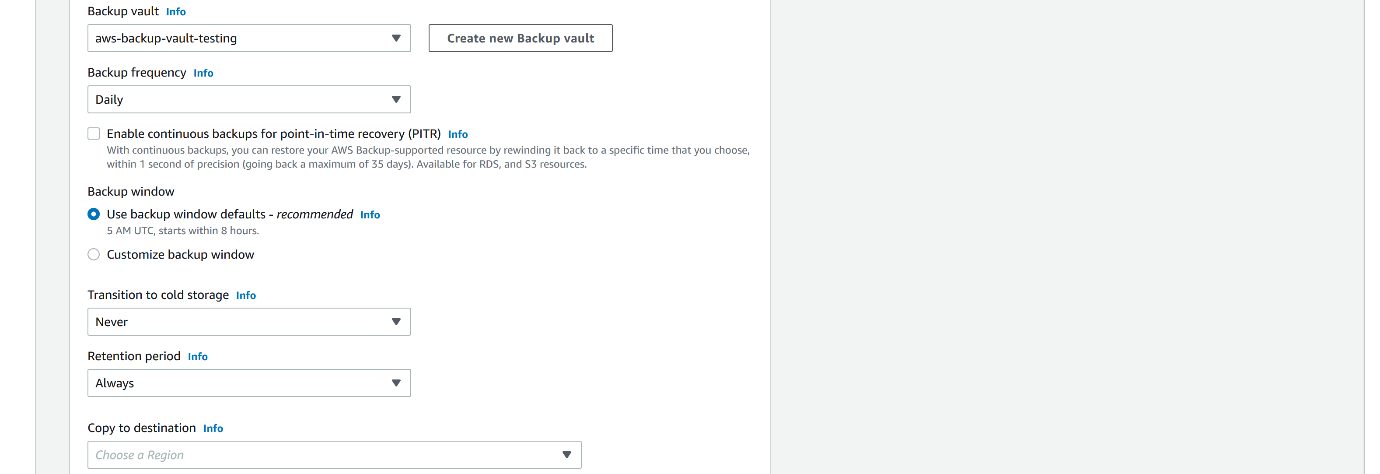
**--- note – our backup vaults got created.**

**Create backup plans**



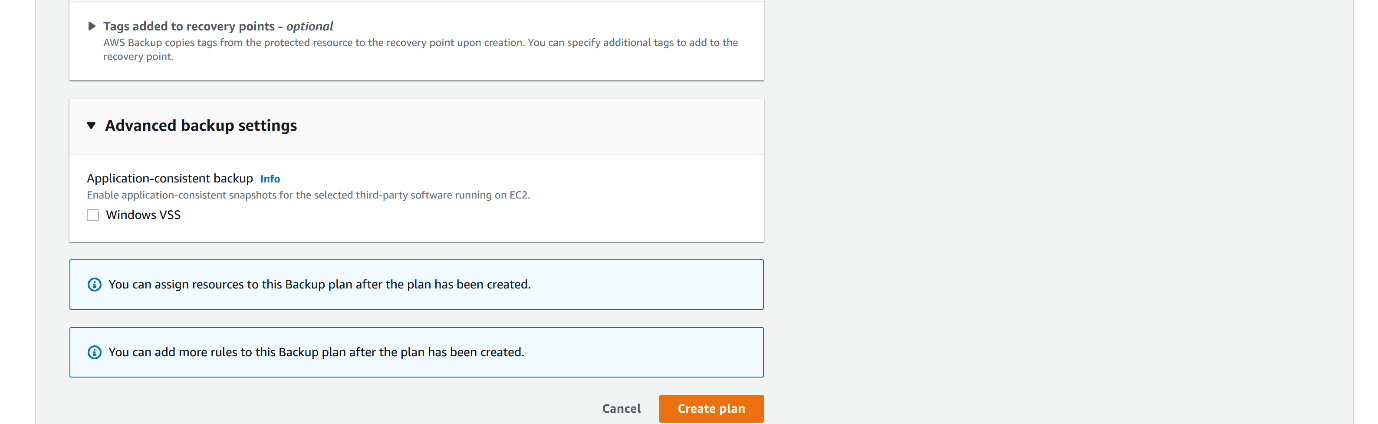
**--- click on create backup plan.**



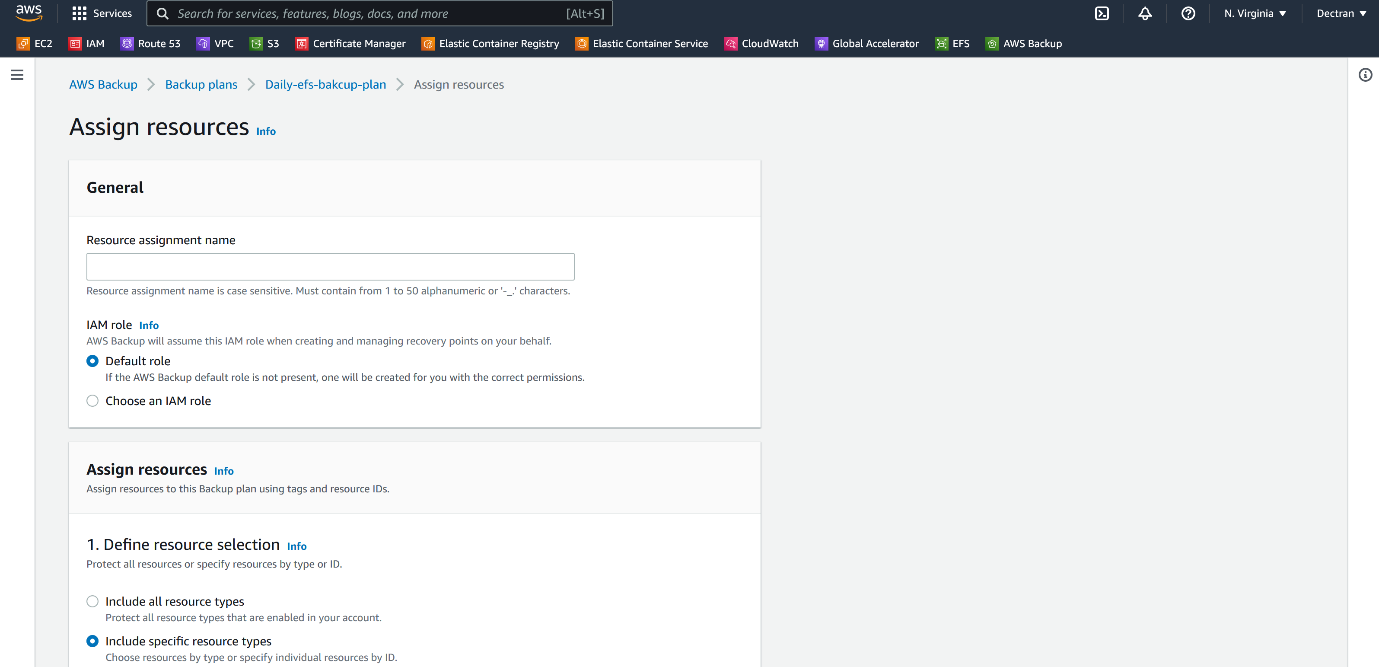


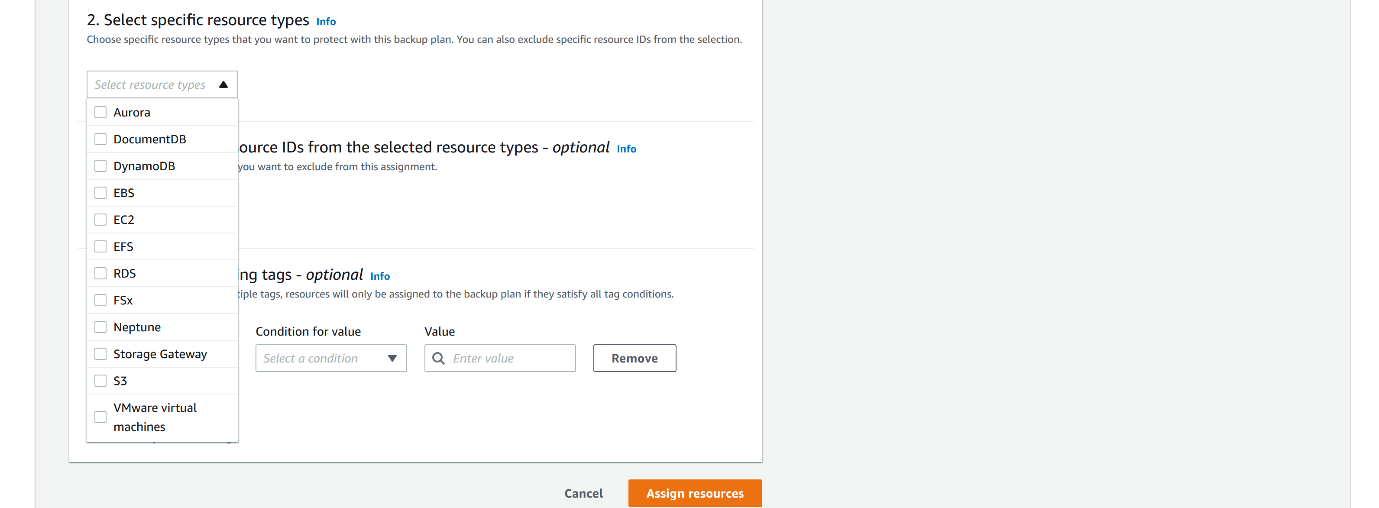
**--- note – you can also choose the destination, you can copy to other regions as well but it is not free.**

**--- Retention period – how many days you want to keep the backups, please mention your retention period here.**



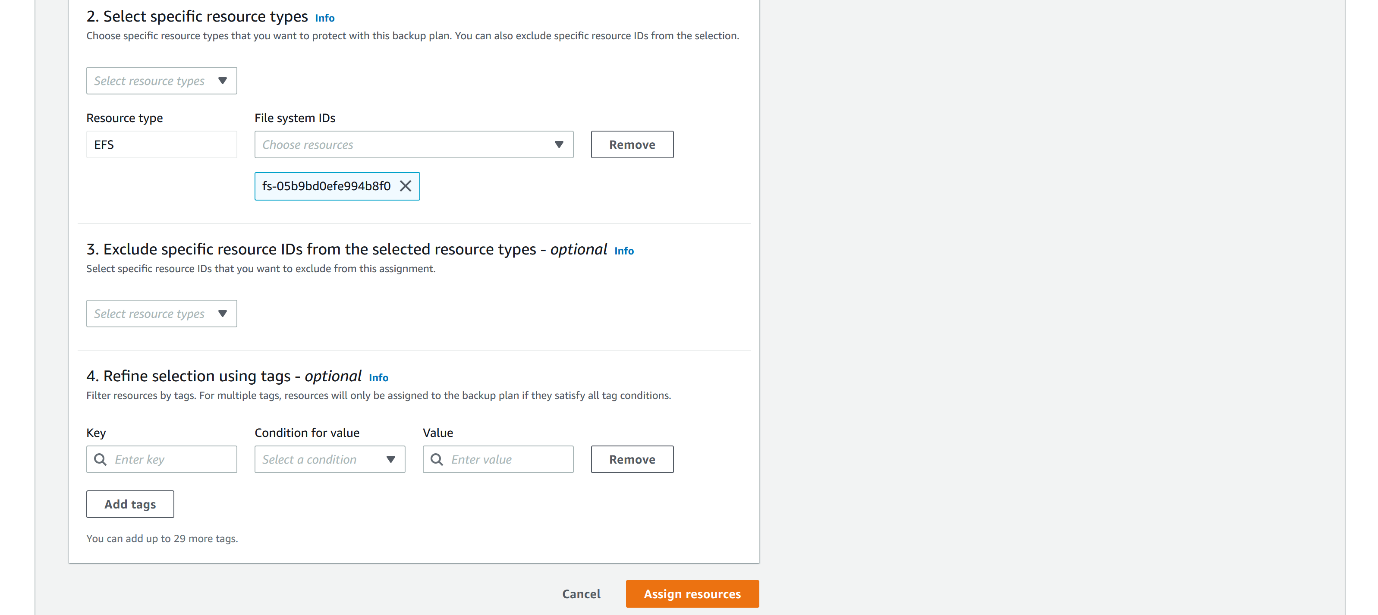
**--- click on create plan.**





**--- note – if you check include specific resource type then we can select specific resource to take backup.**

**--- if I want to take the efs the I can select EFS from here.**

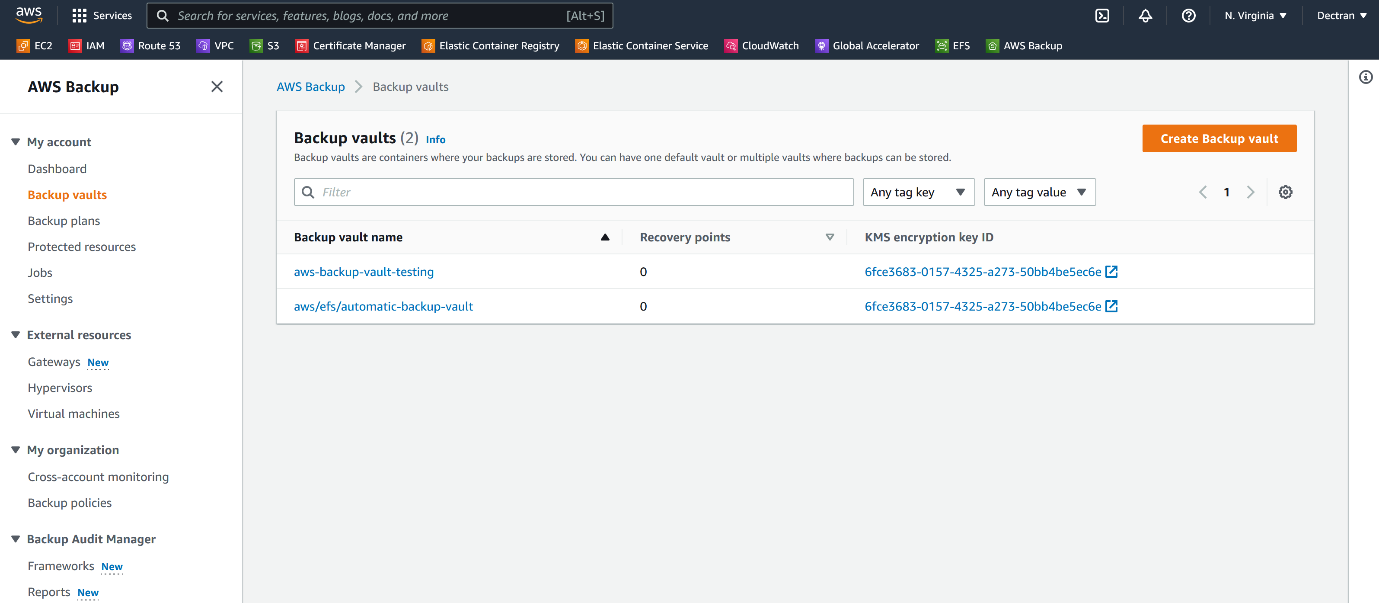


**--- note - now I can take the backup of that EFS daily.**

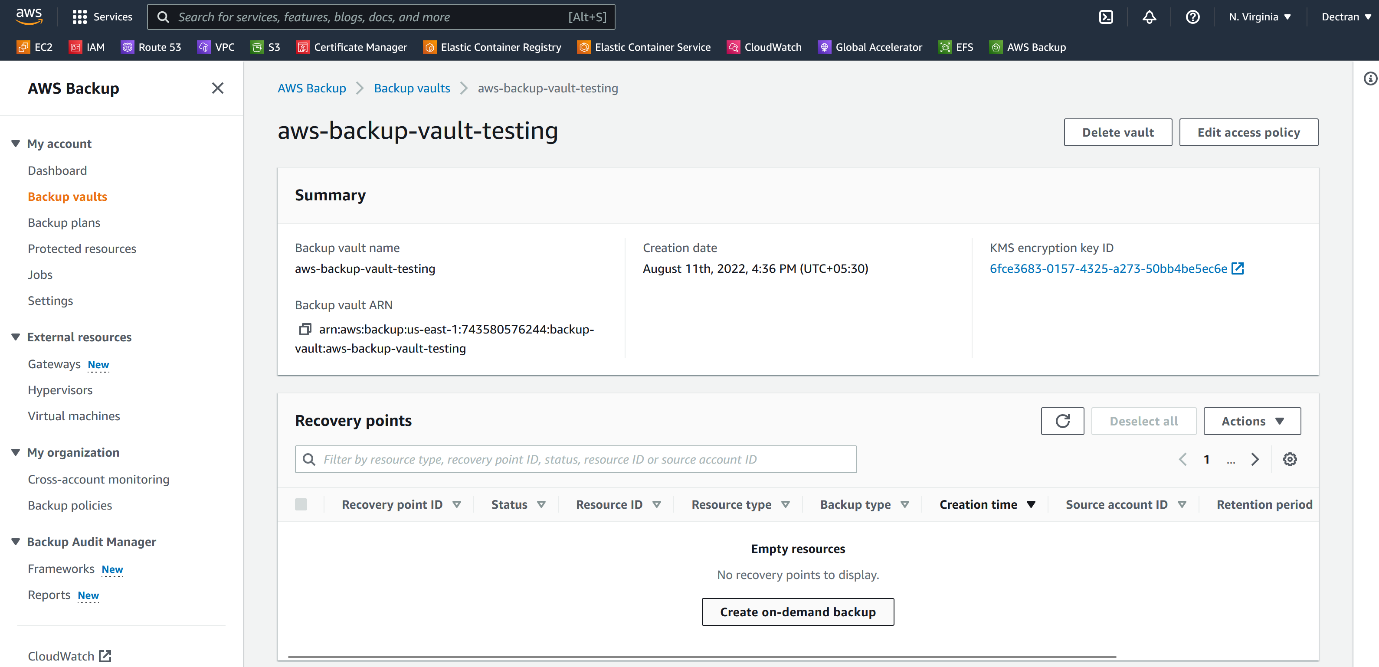
**--- important – we can also take backup of ec2 instances as well based on tags.**

**Create on demand backup for ec2**

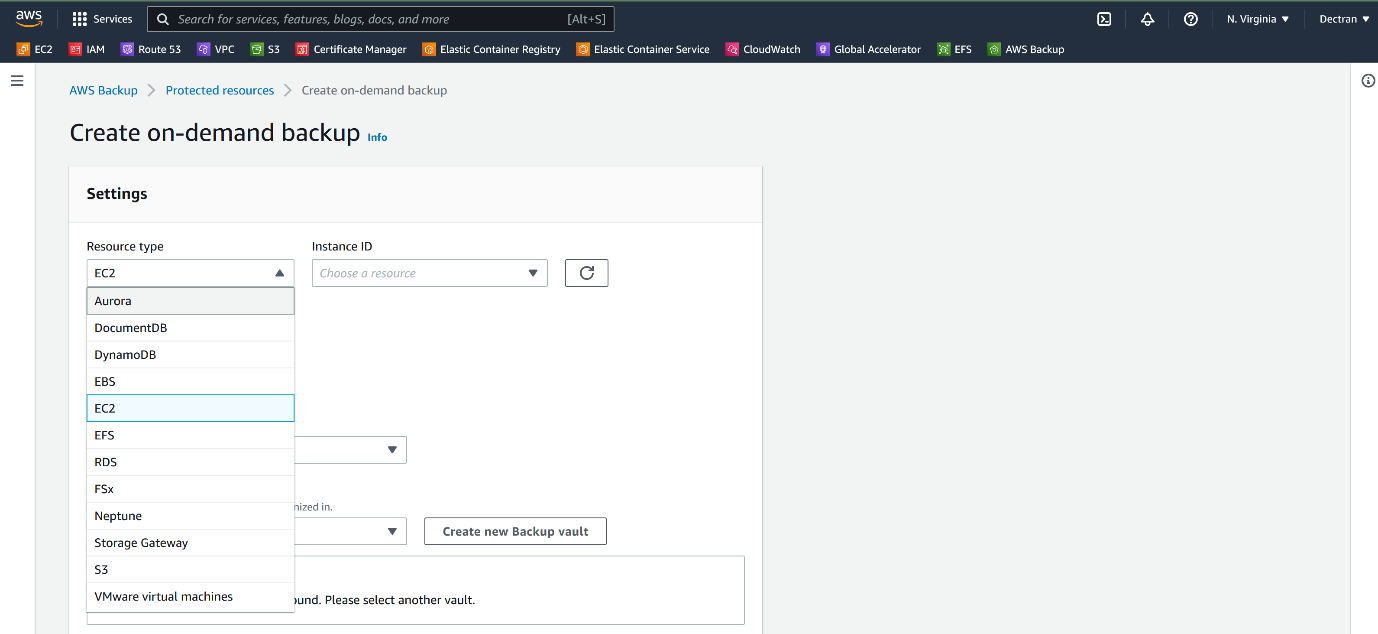
**--- go to backup vault**



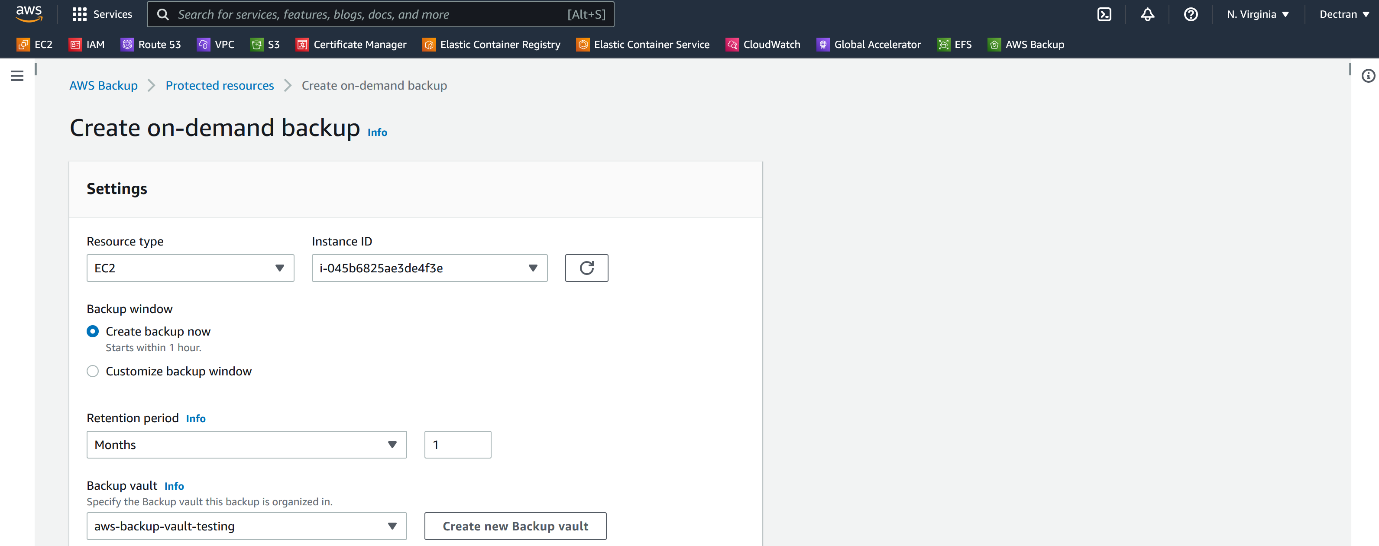
**--- click on aws-backup-vault-testing**

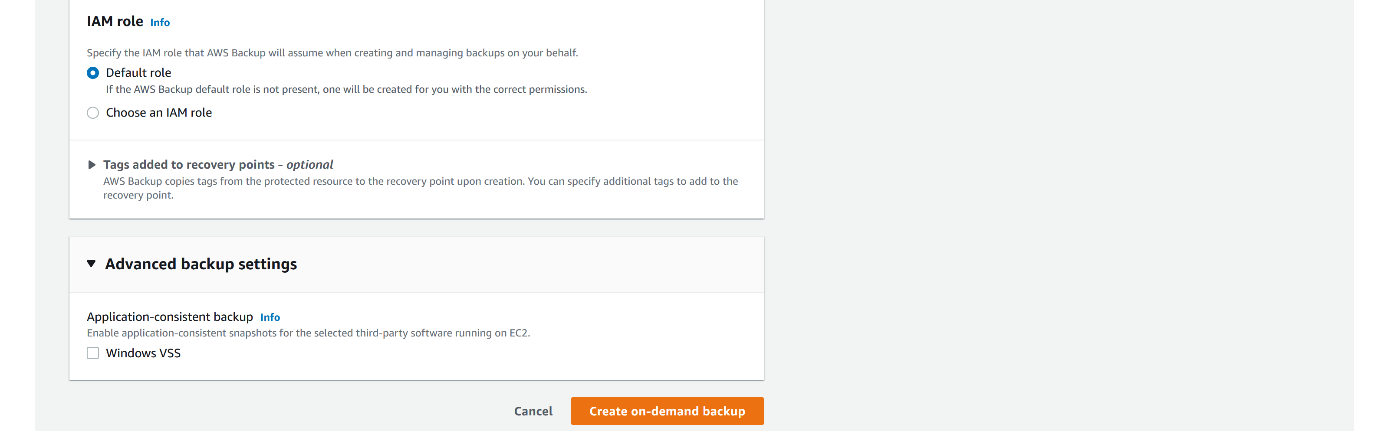


**--- click on create on-demand backup.**

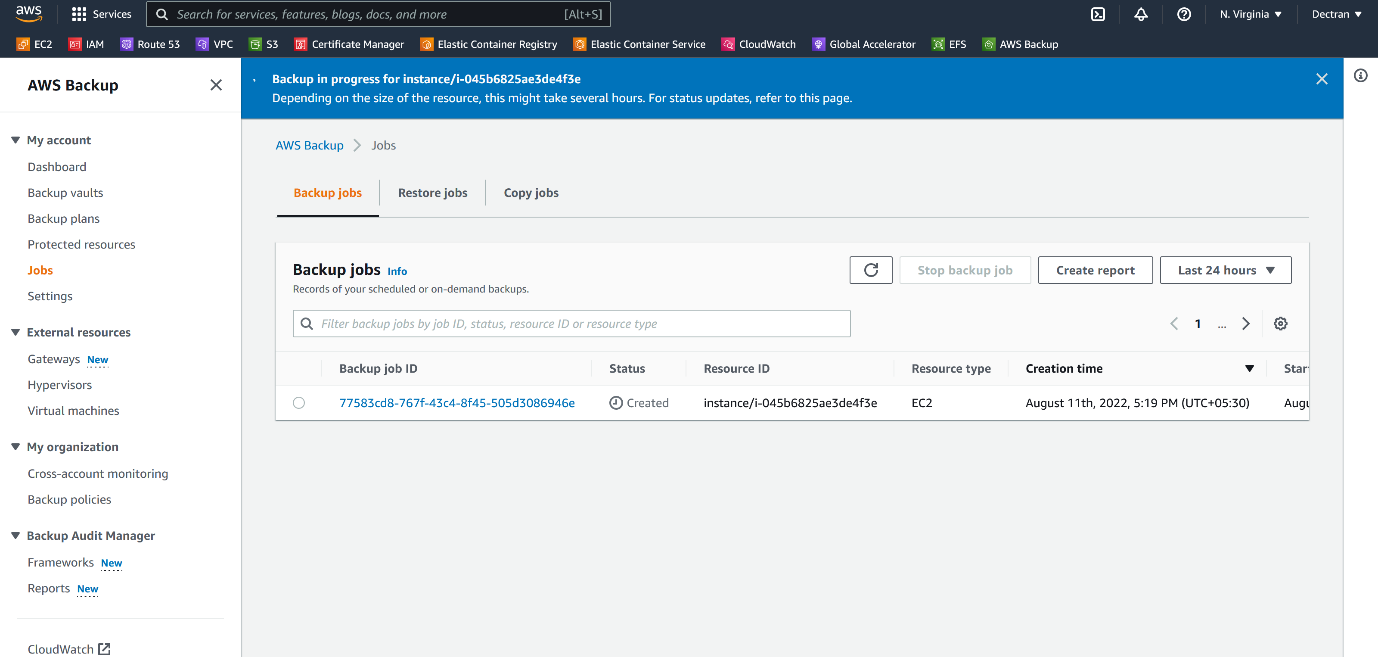


**--- note – we can take on demand backup for these aws resources. I will take backup for prod instance. So, the prod instance should have a tag because aws backup work on the backup. I have already given env=prod to that prod instance.**

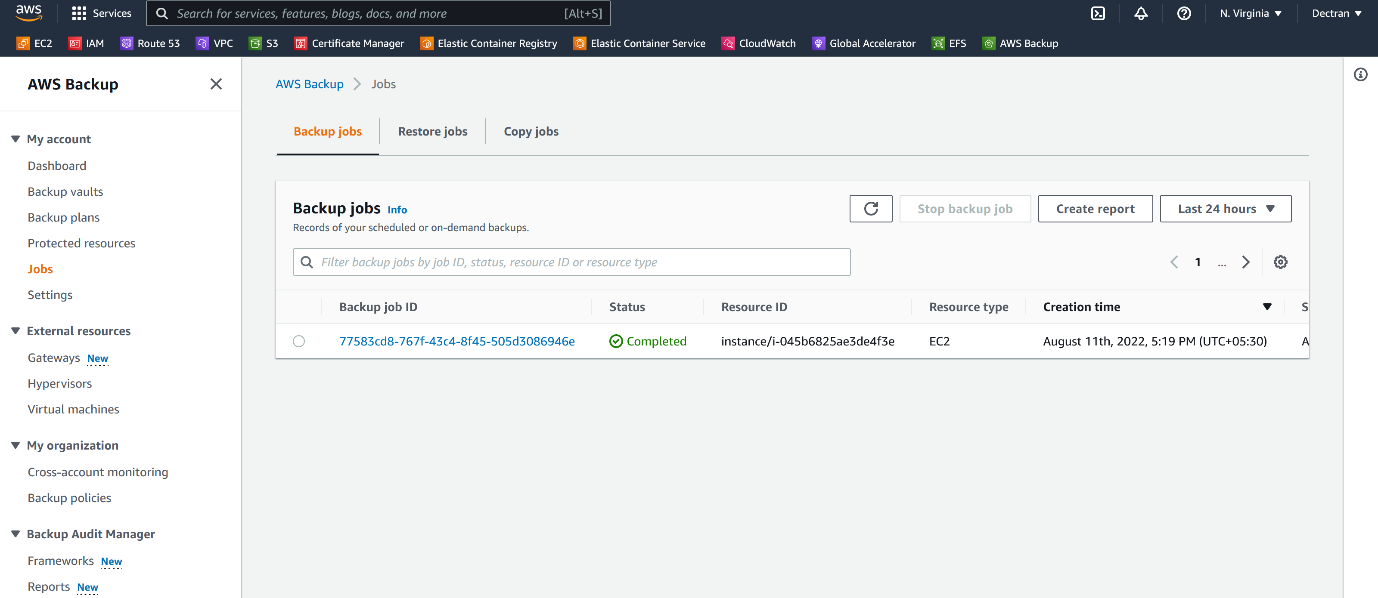




**--- note – I have given retention period of 30 days, so, aws backups will keep 30 days backups of that instance.**



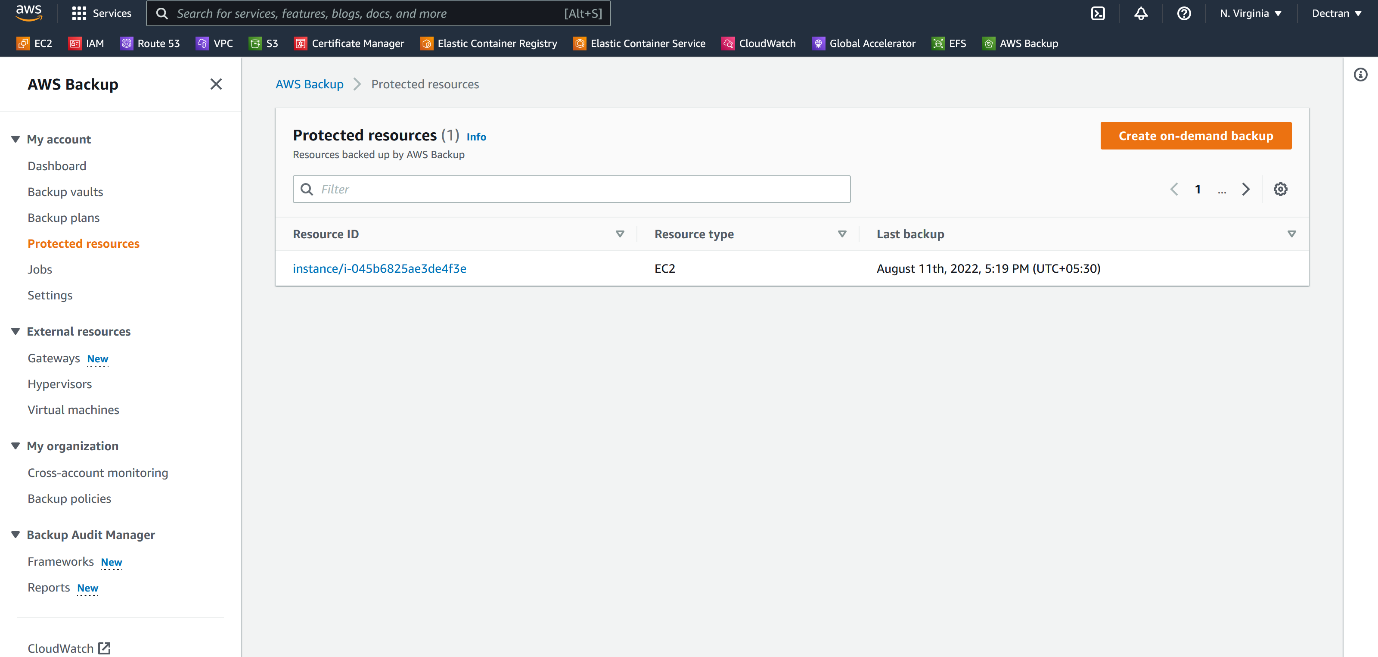
**--- you can see that it is taking backup of that instance.**



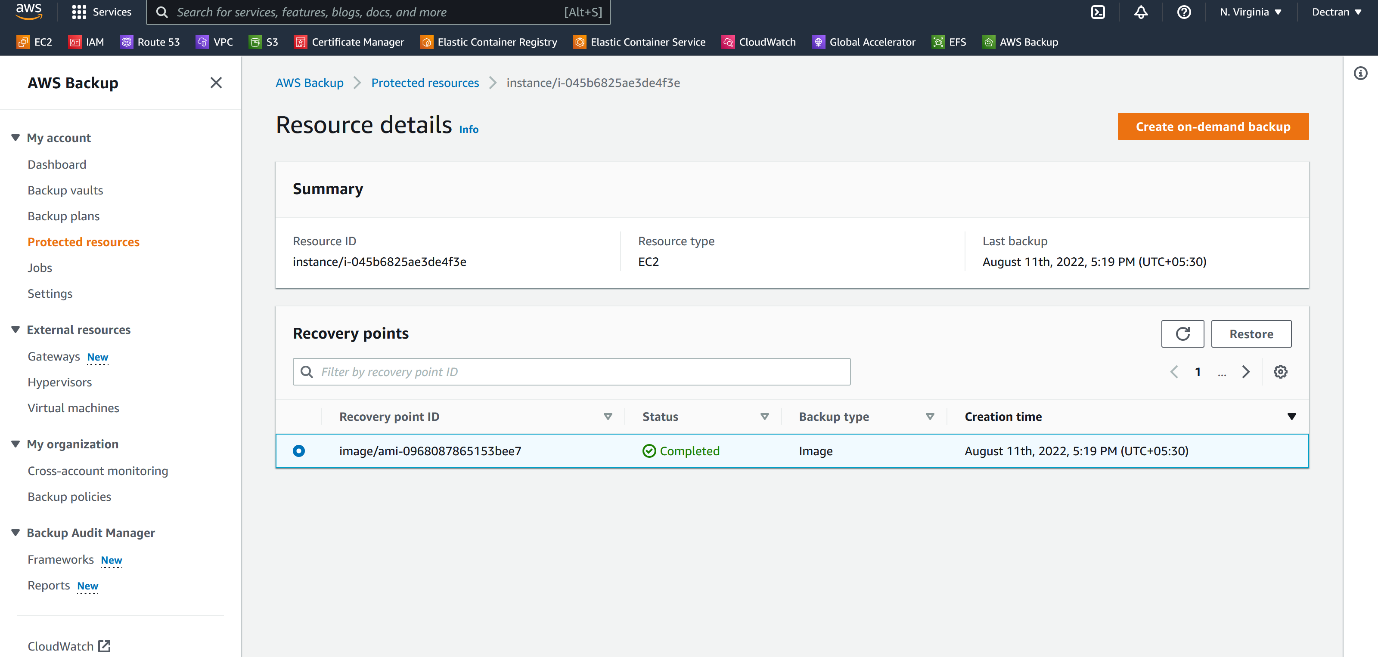
**--- note – the backup is completed.**

**Restore instance from aws backup**

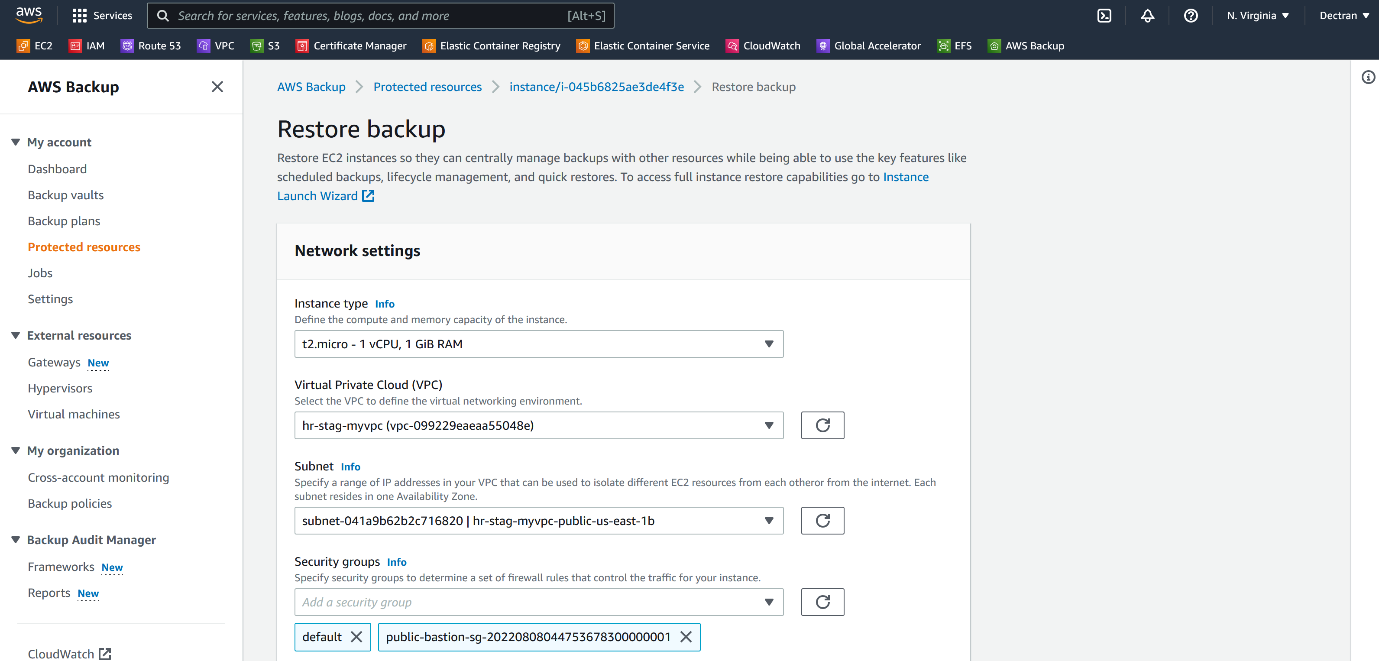
**--- go to aws backups.**

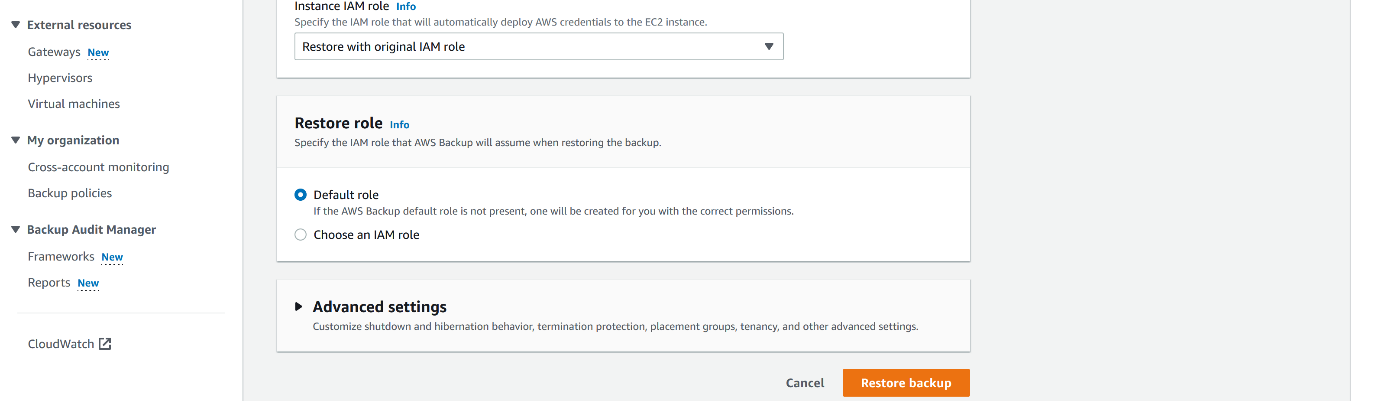


**--- click on protected resources and click on the resource ID.**



**--- NOW, select the image and click on restore.**





**--- now click on restore backup.**