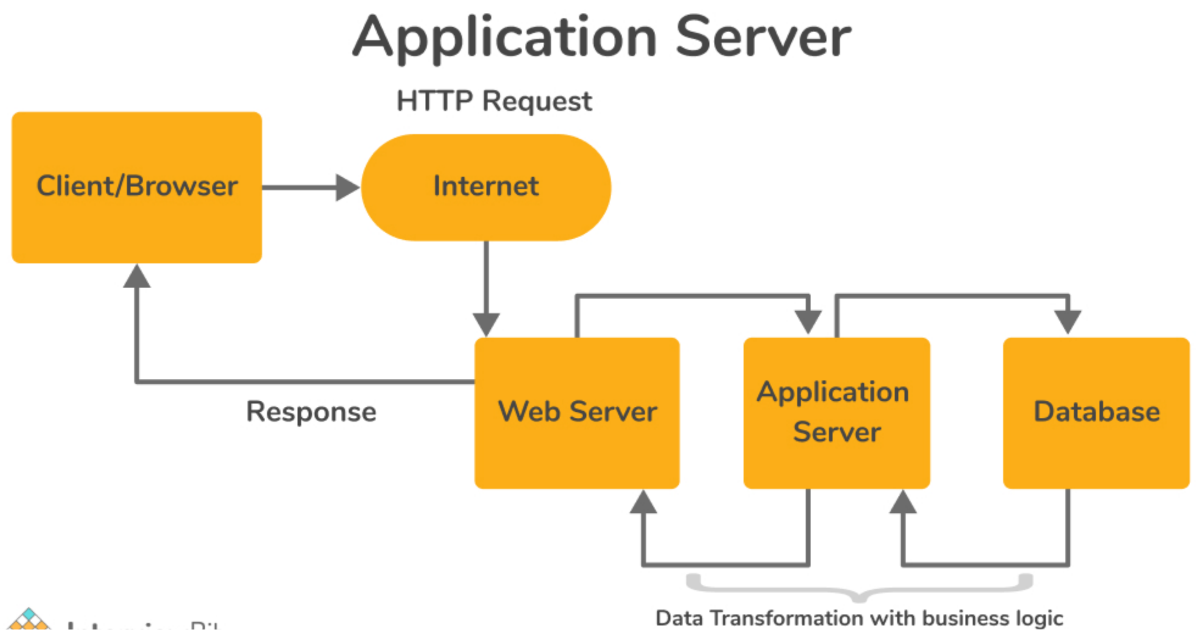
1. What is an Application Server and a Web server?



S.NO	Web Server	Application Server
1.	Web server encompasses web container only.	While application server encompasses Web container as well as EJB container.
2.	Web server is useful or fitted for static content.	Whereas application server is fitted for dynamic content.
3.	Web server consumes or utilizes less resources.	While application server utilize more resources.
4.	Web servers arrange the run environment for web applications.	While application servers arrange the run environment for enterprises applications.
5.	In web servers, multithreading is supported.	While in application server, multithreading is not supported.
6.	Web server's capacity is lower than application server.	While application server's capacity is higher than web server.
7.	In web server, HTML and HTTP protocols are used.	While in this, GUI as well as HTTP and RPC/RMI protocols are used.

8.	Processes that are not resource-intensive are supported.	Processes that are resource-intensive are supported.
9.	Transactions and connection pooling is not supported.	Transactions and connection pooling is supported.
10.	The capacity of fault tolerance is low as compared to application servers.	It has high fault tolerance.
11.	Web Server examples are Apache HTTP Server , Nginx.	Application Servers example are JBoss , Glassfish.
12.	Subset of the application server.	Superset of a web server.
13.	Reduces longer running processes that are resource-intensive.	Reduces web traffic, which is not resource-intensive.

Features of Web Server:

1) Handles HTTP Protocol (static contents)
2) No Server-side Programming.
3) Support web-Based Applications (JSP, Servlets, PHP, HTML, etc.)
4) Not support Database Connection Pooling.
5) Not provide EJB support.

Features of Application Server:

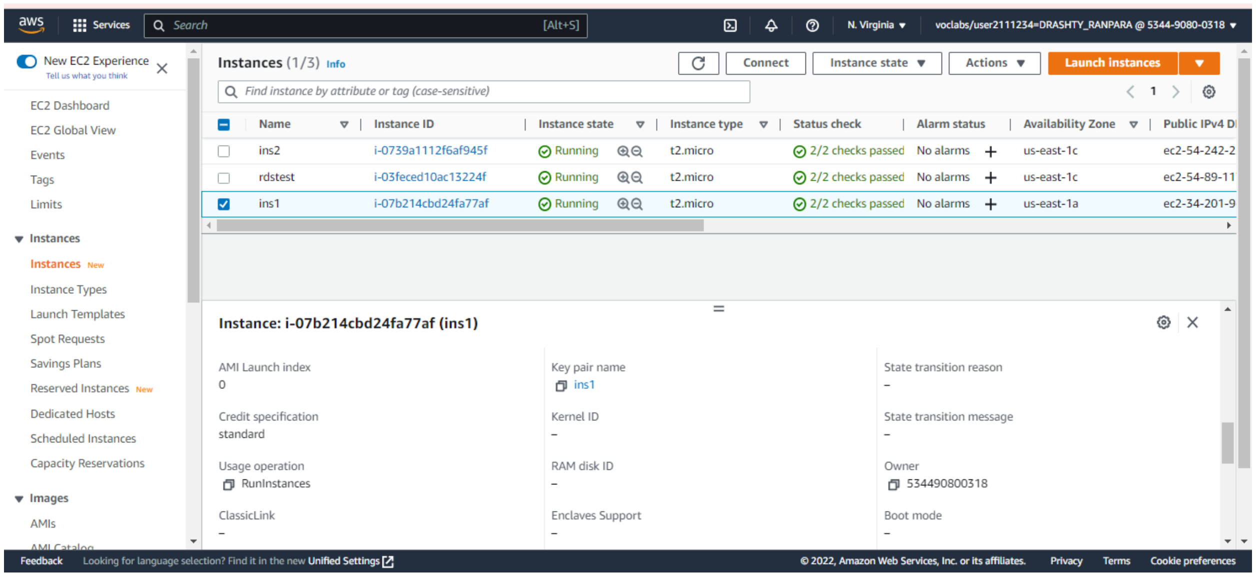
1) Serves dynamic business logic.
2) It helps you to manage backend logic like calculations, database, processing, etc.
3) It helps you to deploy applications, dependency injection, security, etc. database pooling, and EJB.
4) The superior server of Web Server.

2. Create a storage space which can be accessed across availability zone by may instances at a time. Demonstrate it by using 2 instances, extract the data from S3 bucket and try to modify any one file using 1 st instance, while accessing the same file using 2 nd instance the changes should be reflected.

For LINUX

1. Login to AWS Portal. Create 2 instances belonging to the same region and different availability zone. Here I've created an instance in zone - a and b. While creating an instance, give the security groups as SSH, NFS, HTTP, HTTPS by clicking on Edit

Settings under Networks, Keep the AMI as AWS Linux OS with security group as ppk file.



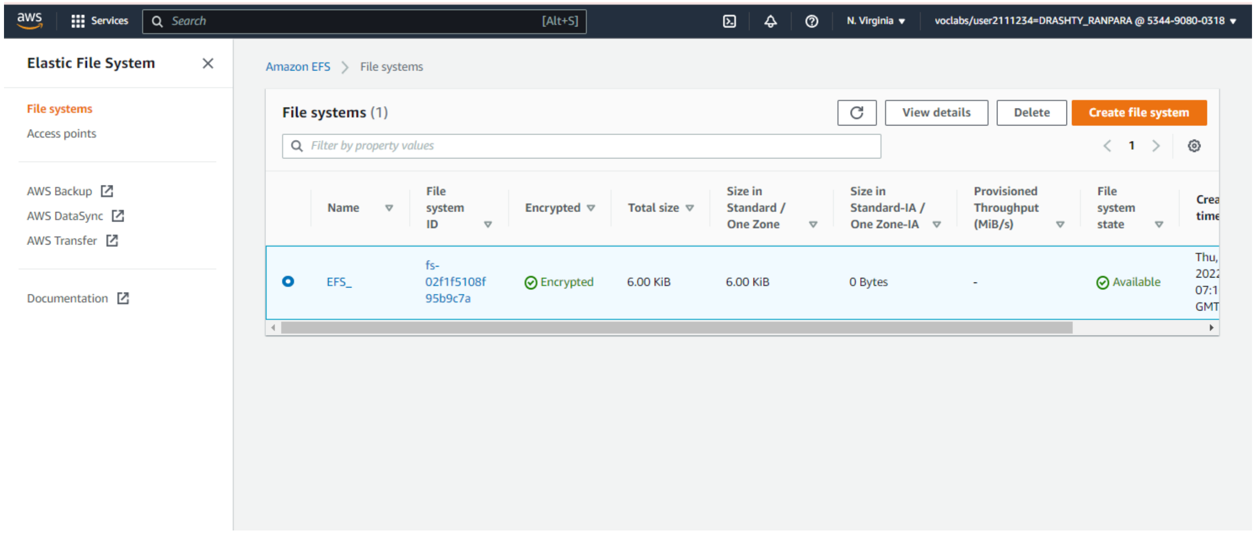
Instances (1/3) Info

	Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public IPv4 D
☐	ins2	i-0739a1112f6af945f	Running	t2.micro	2/2 checks passed	No alarms	us-east-1c	ec2-54-242-2
☐	rdstest	i-03feced10ac13224f	Running	t2.micro	2/2 checks passed	No alarms	us-east-1c	ec2-54-89-11
✓	ins1	i-07b214cbd24fa77af	Running	t2.micro	2/2 checks passed	No alarms	us-east-1a	ec2-34-201-9

Instance: i-07b214cbd24fa77af (ins1)

AMI Launch index 0	Key pair name ins1	State transition reason –
Credit specification standard	Kernel ID –	State transition message –
Usage operation RunInstances	RAM disk ID –	Owner 534490800318
ClassicLink –	Enclaves Support –	Boot mode –

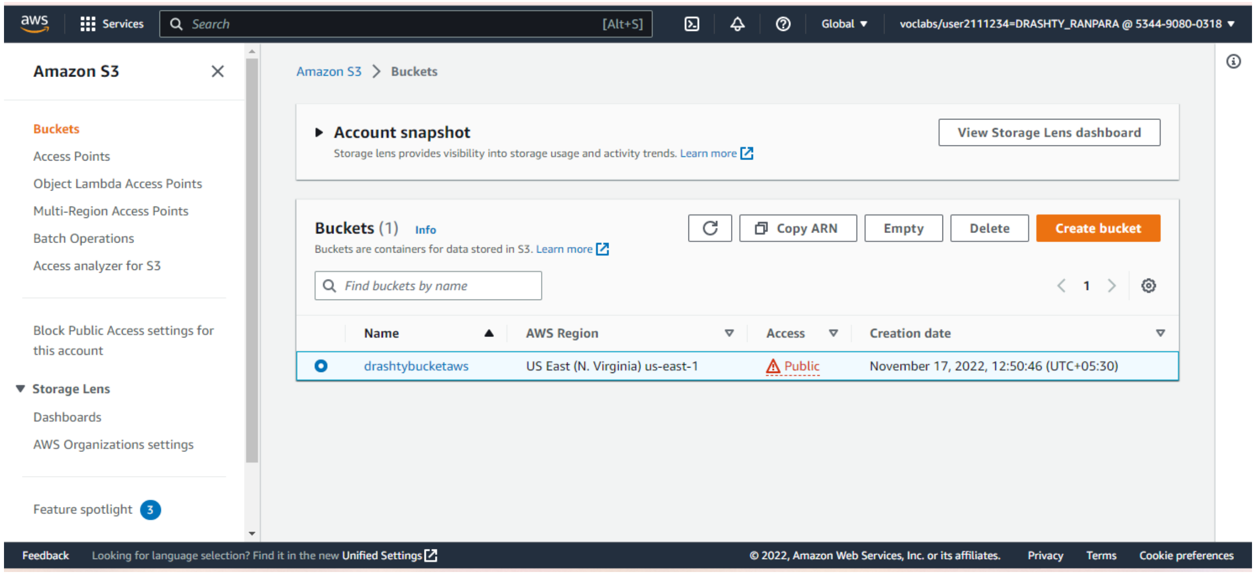
2. Create EFS by navigating to EFS Service and click on Create File System by providing its name and click on create.



File systems (1)

Name	File system ID	Encrypted	Total size	Size in Standard / One Zone	Size in Standard-IA / One Zone-IA	Provisioned Throughput (MiB/s)	File system state	Crea time
EFS_	fs-02f1f5108f95b9c7a	Encrypted	6.00 KiB	6.00 KiB	0 Bytes	-	Available	Thu, 2022 07:1 GMT

3. Under Amazon S3 navigate to Buckets, click on Create bucket and provide bucket name, region, object ownership as “ACLs disabled”. Keep the Block public access ticked and click on create bucket.



Storage lens provides visibility into storage usage and activity trends. Learn more

Buckets (1) Info

Buckets are containers for data stored in S3. Learn more

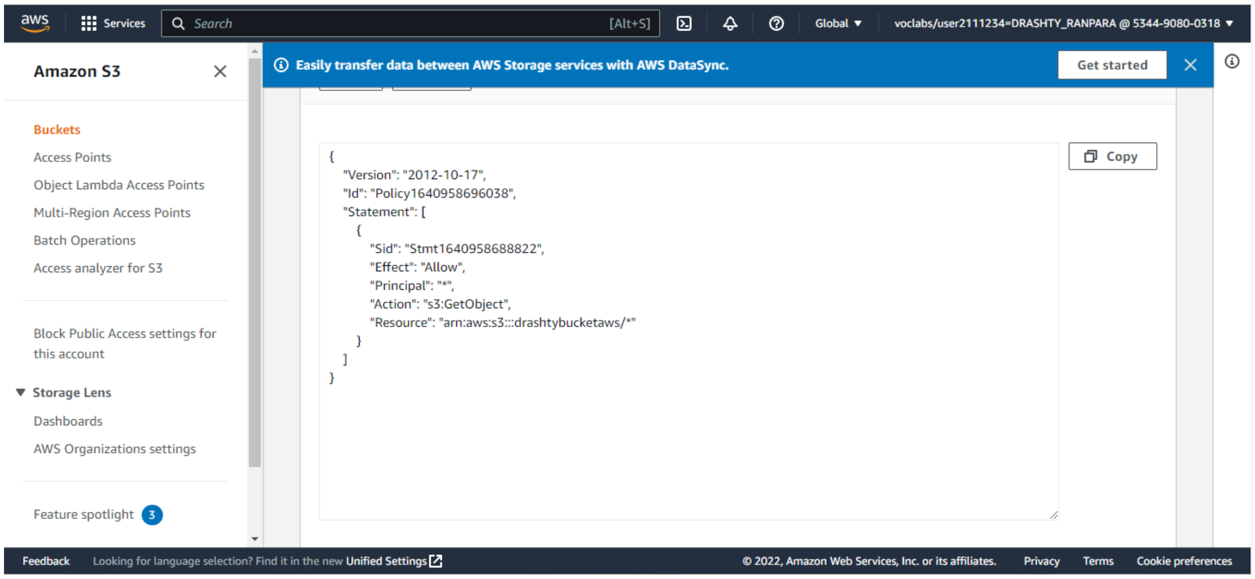
Name	AWS Region	Access	Creation date
drashtybucketaws	US East (N. Virginia) us-east-1	Public	November 17, 2022, 12:50:46 (UTC+05:30)

Looking for language selection? Find it in the new Unified Settings

© 2022, Amazon Web Services, Inc. or its affiliates.

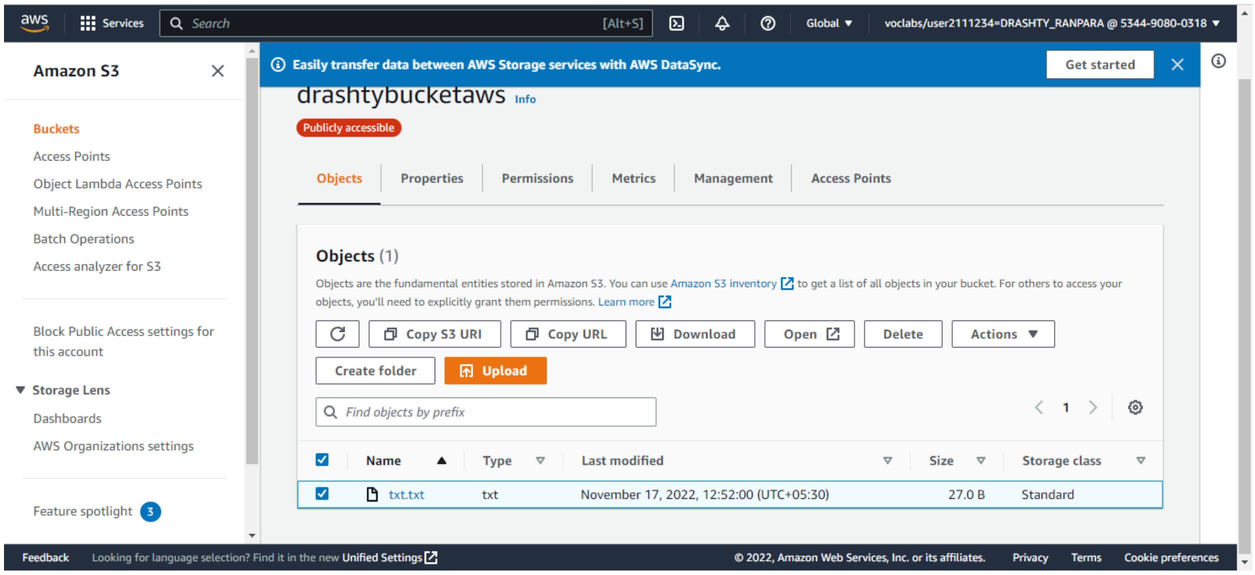
4. Click on the bucket that is created. Then go to Objects → Upload - a text file for testing. Furthermore, Navigate to Permissions → Untick Block all services, under Bucket Policy → {
   a. "Version": "2012-10-17",
   b. "Id": "Policy1640958696038",
   c. "Statement": [
   d. {
   e. "Sid": "Stmt1640958688822",
   f. "Effect": "Allow",
   g. "Principal": "*",
   h. "Action": "s3:GetObject",
   i. "Resource": "arn:aws:s3:::drashtybucketaws/*"
   j. }
   k.]
   l. }

   And then save it. Navigate to the txt file and press copy url to get the file link.



aws
Services
Search
[Alt+S]
Global
voclabs/user2111234=DRASHTY_RANPARA @ 5344-9080-0318
Amazon S3
Buckets
Access Points
Object Lambda Access Points
Multi-Region Access Points
Batch Operations
Access analyzer for S3
Block Public Access settings for this account
Storage Lens
Dashboards
AWS Organizations settings
Feature spotlight 3
Easily transfer data between AWS Storage services with AWS DataSync.
Get started
Copy
Feedback
Looking for language selection? Find it in the new Unified Settings
© 2022, Amazon Web Services, Inc. or its affiliates.
Privacy
Terms
Cookie preferences

```
{
    "Version": "2012-10-17",
    "Id": "Policy1640958696038",
    "Statement": [
      {
        "Sid": "Stmt1640958688822",
        "Effect": "Allow",
        "Principal": "*",
        "Action": "s3:GetObject",
        "Resource": "arn:aws:s3:::drashtybucketaws/*"
      }
    ]
}
```

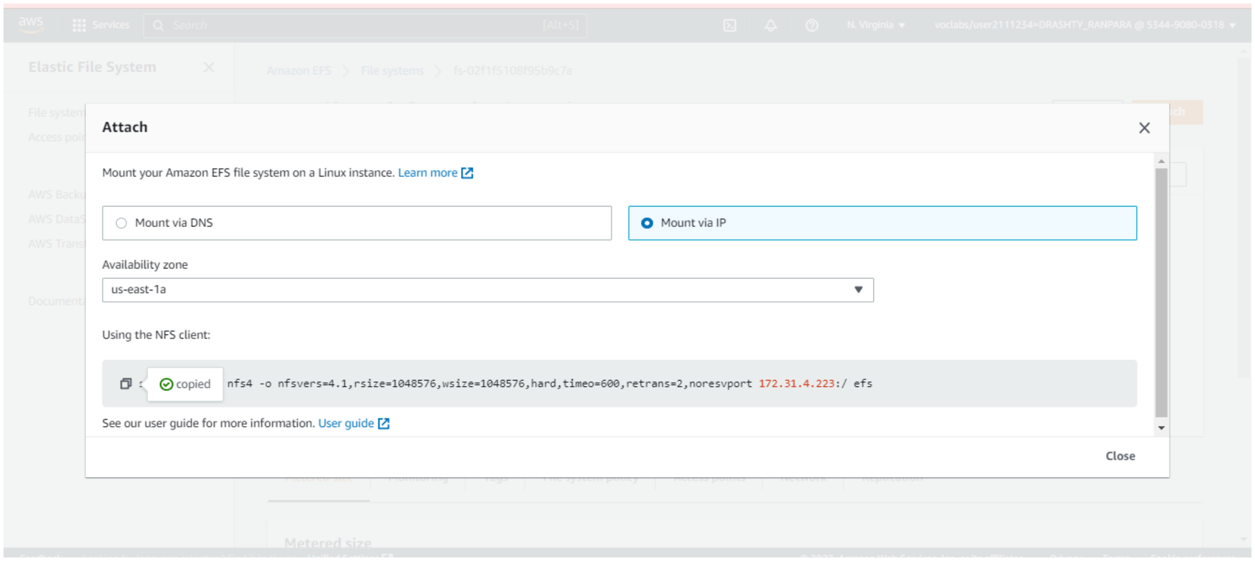


Objects (1)

Objects are the fundamental entities stored in Amazon S3. You can use Amazon S3 inventory to get a list of all objects in your bucket. For others to access your objects, you'll need to explicitly grant them permissions. Learn more

	Name	Type	Last modified	Size	Storage class
✓	txt.txt	txt	November 17, 2022, 12:52:00 (UTC+05:30)	27.0 B	Standard

5. Now, navigate to EFS and click on the newly created efs. Next, click on Attach to mount the EFS file to the instances available in zone a and b respectively. Click on radio button Mount via IP and copy the link available below.



Attach

Mount your Amazon EFS file system on a Linux instance. Learn more

Mount via DNS [✓ Mount via IP]

Availability zone: us-east-1a

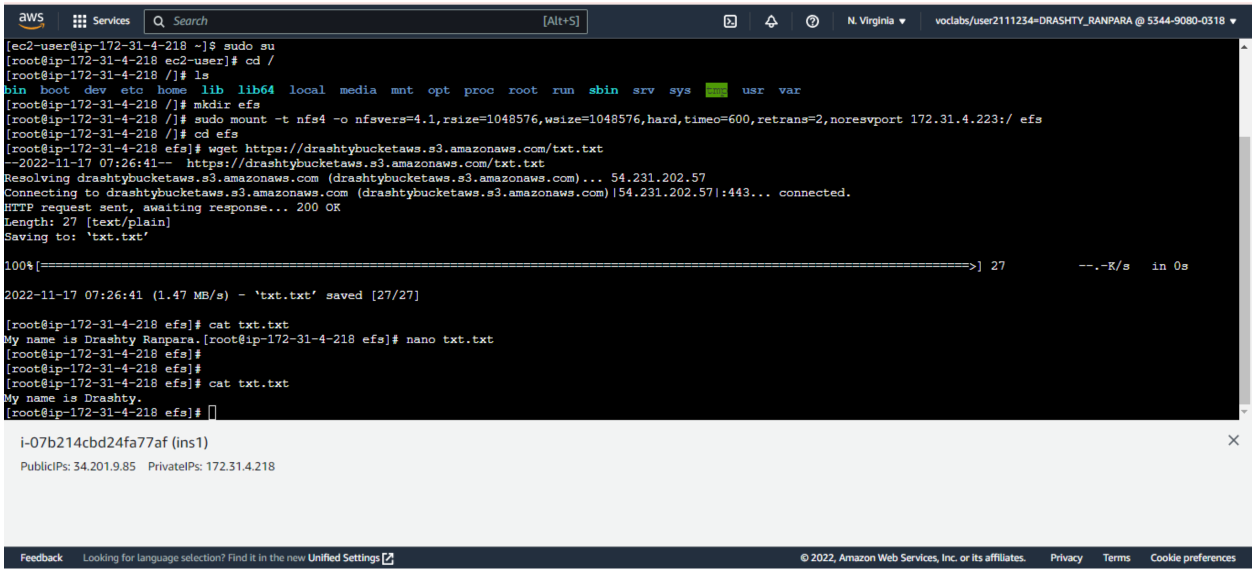
Using the NFS client:

```
nfs4 -o nfsvers=4.1,rsize=1048576,wsize=1048576,hard,timeo=600,retrans=2,noresvport 172.31.4.223:/ efs
```

See our user guide for more information. User guide

6. Now, navigate to the EC2 instance and select one of the instances. In this case I'm selecting ins1 and clicking on Connect → EC2 Instance Connect → Connect. This will navigate you to the console wherein you can work with the commands as follows:

# sudo su	→ navigate to root dir
# cd /	→ go to next dir
# ls	→ check files in that dir
# mkdir efs	→ create an efs folder
# sudo mount…	→ your copied mount code of EFS for instance in zone a
# cd efs/	→ change dir to efs
# wget https://bucket..	→ paste the copied S3 bucket url here
# ls	→ again check the files and you'll find the file uploaded in S3.



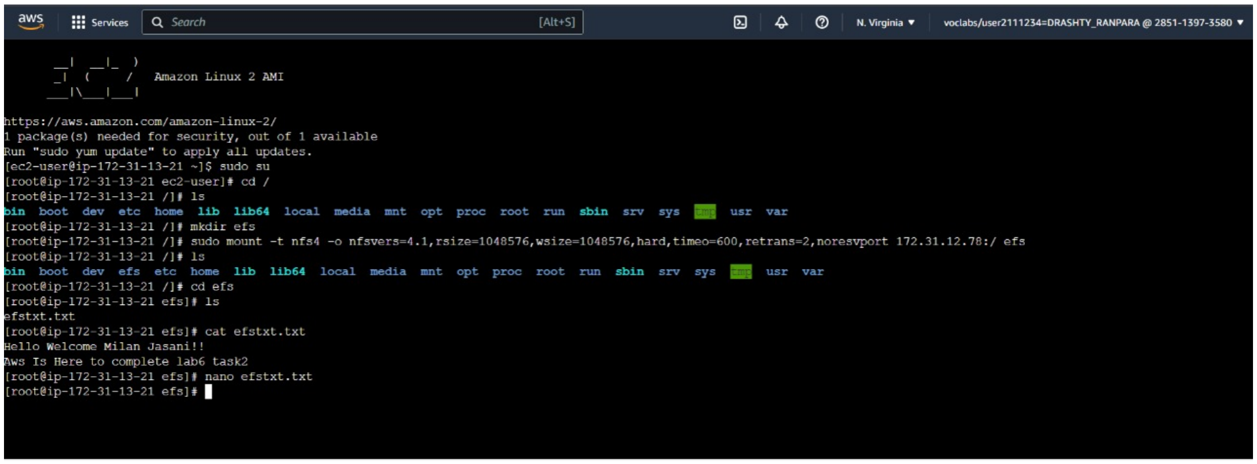
```
[ec2-user@ip-172-31-4-218 ~]$ sudo su
[root@ip-172-31-4-218 ec2-user]# cd /
[root@ip-172-31-4-218 /]# ls
bin  boot  dev  etc  home  lib  lib64  local  media  mnt  opt  proc  root  run  sbin  srv  sys  tmp  usr  var
[root@ip-172-31-4-218 /]# mkdir efs
[root@ip-172-31-4-218 /]# sudo mount -t nfs4 -o nfsvers=4.1,rsize=1048576,wsize=1048576,hard,timeo=600,retrans=2,noresvport 172.31.4.223:/ efs
[root@ip-172-31-4-218 /]# cd efs
[root@ip-172-31-4-218 efs]# wget https://drashtybucketaws.s3.amazonaws.com/txt.txt
--2022-11-17 07:26:41--  https://drashtybucketaws.s3.amazonaws.com/txt.txt
Resolving drashtybucketaws.s3.amazonaws.com (drashtybucketaws.s3.amazonaws.com)... 54.231.202.57
Connecting to drashtybucketaws.s3.amazonaws.com (drashtybucketaws.s3.amazonaws.com)|54.231.202.57|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 27 [text/plain]
Saving to: 'txt.txt'

100%[=====================================================================================================>] 27          --.-K/s   in 0s

2022-11-17 07:26:41 (1.47 MB/s) - 'txt.txt' saved [27/27]

[root@ip-172-31-4-218 efs]# cat txt.txt
My name is Drashty Ranpara.[root@ip-172-31-4-218 efs]# nano txt.txt
[root@ip-172-31-4-218 efs]#
[root@ip-172-31-4-218 efs]#
[root@ip-172-31-4-218 efs]# cat txt.txt
My name is Drashty.
[root@ip-172-31-4-218 efs]# 
```

i-07b214cbd24fa77af (ins1)

PublicIPs: 34.201.9.85 PrivateIPs: 172.31.4.218

Services
Search [Alt+S]
N. Virginia
voclabs/user2111234=DRASHTY_RANPARA @ 5344-9080-0318
Feedback
Looking for language selection? Find it in the new Unified Settings
© 2022, Amazon Web Services, Inc. or its affiliates.
Privacy
Terms
Cookie preferences

7. Now similarly mount EFS for instance2 changing its availability zone to **b** and copy url and execute it in the console.



```
       __|  __|_  )
       _|  (     /   Amazon Linux 2 AMI
      ___|\___|___|

https://aws.amazon.com/amazon-linux-2/
1 package(s) needed for security, out of 1 available
Run "sudo yum update" to apply all updates.
[ec2-user@ip-172-31-13-21 ~]$ sudo su
[root@ip-172-31-13-21 ec2-user]# cd /
[root@ip-172-31-13-21 /]# ls
bin  boot  dev  etc  home  lib  lib64  local  media  mnt  opt  proc  root  run  sbin  srv  sys  tmp  usr  var
[root@ip-172-31-13-21 /]# mkdir efs
[root@ip-172-31-13-21 /]# sudo mount -t nfs4 -o nfsvers=4.1,rsize=1048576,wsize=1048576,hard,timeo=600,retrans=2,noresvport 172.31.12.78:/ efs
[root@ip-172-31-13-21 /]# ls
bin  boot  dev  efs  etc  home  lib  lib64  local  media  mnt  opt  proc  root  run  sbin  srv  sys  tmp  usr  var
[root@ip-172-31-13-21 /]# cd efs
[root@ip-172-31-13-21 efs]# ls
efstxt.txt
[root@ip-172-31-13-21 efs]# cat efstxt.txt
Hello Welcome Milan Jasani!!
Aws Is Here to complete lab6 task2
[root@ip-172-31-13-21 efs]# nano efstxt.txt
[root@ip-172-31-13-21 efs]# █
```

8. Hurray! The 2 instances from two different availability zones can now access the common file kept in the S3 using EFS under the same region.🙂