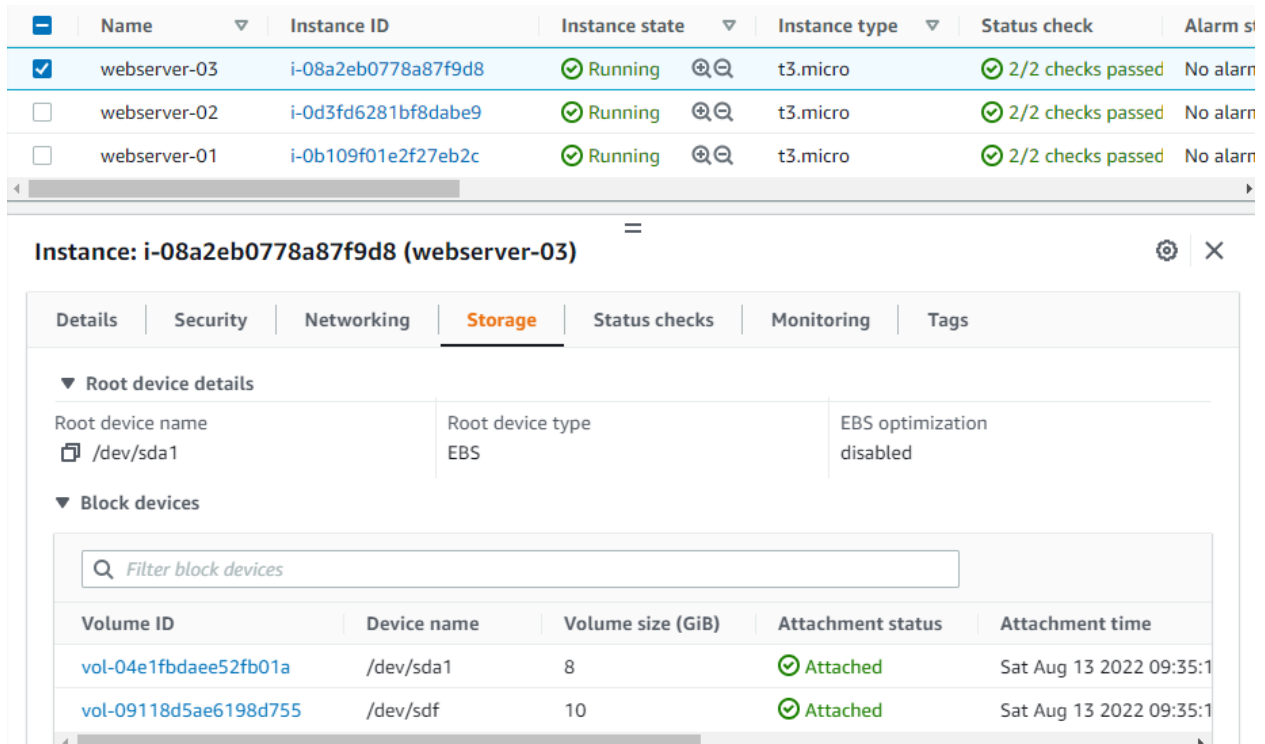


Reduce Storage Cost with EFS

In this step by step we are required to create an EFS volume and attach it to our EC2 instance. We also detach the old volume and delete it.

1. Go into the EC2 instance > Instances running
2. In this lab click on each individual webserver
 - a. You can see that they all have the same configuration
 - b. Yes I have only shown webserver-03 but it is the same for 1 and 2



The screenshot displays the AWS Management Console. At the top, a table lists three EC2 instances: webserver-03, webserver-02, and webserver-01. All are in a 'Running' state with 't3.micro' instance types and '2/2 checks passed' status. Below this, the 'Instance: i-08a2eb0778a87f9d8 (webserver-03)' page is shown, with the 'Storage' tab selected. Under 'Root device details', the root device is '/dev/sda1' (EBS, 8 GiB). Under 'Block devices', a table shows two attached volumes: 'vol-04e1fbd4ee52fb01a' (8 GiB) and 'vol-09118d5ae6198d755' (10 GiB), both attached to '/dev/sda1' and '/dev/sdf' respectively, with an attachment time of 'Sat Aug 13 2022 09:35:11'.

Name	Instance ID	Instance state	Instance type	Status check	Alarm s
webserver-03	i-08a2eb0778a87f9d8	Running	t3.micro	2/2 checks passed	No alarm
webserver-02	i-0d3fd6281bf8dabe9	Running	t3.micro	2/2 checks passed	No alarm
webserver-01	i-0b109f01e2f27eb2c	Running	t3.micro	2/2 checks passed	No alarm

Instance: i-08a2eb0778a87f9d8 (webserver-03)

Details | Security | Networking | **Storage** | Status checks | Monitoring | Tags

▼ Root device details

Root device name	Root device type	EBS optimization
/dev/sda1	EBS	disabled

▼ Block devices

Filter block devices

Volume ID	Device name	Volume size (GiB)	Attachment status	Attachment time
vol-04e1fbd4ee52fb01a	/dev/sda1	8	Attached	Sat Aug 13 2022 09:35:11
vol-09118d5ae6198d755	/dev/sdf	10	Attached	Sat Aug 13 2022 09:35:11

3. So, we are going to reduce the 10 GB storage disk on each server and replace it with a singled shared EFS volume
4. Go into EFS into a new tab > Create file system
5. Name the file system
6. Keep the VPC by default
7. And change the storage class to "One Zone"
 - a. This is because all of the resources we'll be using will be in the us-east-1a AZ

Create file system

Create an EFS file system with service recommended settings. [Learn more](#)

Name - optional

Name your file system.

SharedWeb-01

Name can include letters, numbers, and +-=_:/ symbols, up to 256 characters.

Virtual Private Cloud (VPC)

Choose the VPC where you want EC2 instances to connect to your file system. [Learn more](#)

vpc-015c3c39d8e43343a

Storage class [Learn more](#)

☐ Standard

Stores data redundantly across multiple AZs

☒ One Zone

Stores data redundantly within a single AZ

Availability Zone

Choose the Availability Zone where you want to create your file system

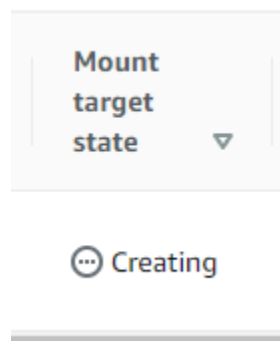
us-east-1a

Cancel

Customize

Create

8. And click create
9. Click on the file system to view it
10. Network Tab
11. You can see that the mount target state shows "Creating"



12. Once it becomes available click "Manage" beside the refresh button
13. We need to remove the security group that is there by default so click the X
14. And then we will select our EC2 group that is listed
15. Go back to our EC2 tab and on the left side pane, go into the security groups under "Network & Security"
16. Select the one that doesn't have the default security name

	Name	Security group ID	Security group name
<input checked="" type="checkbox"/>	-	sg-005c5d878f945491a	cfst-3358-005baa9e8...
<input type="checkbox"/>	-	sg-06492ebd2f14b7821	default

17. Once selected go to the inbound rules tab below
18. Click edit inbound rules
19. Add rule
20. For under Type, search up NFS and select it
21. For the search column type in or select 0.0.0.0/0
 - a. This means everywhere
22. Save rules
23. Go back to the dashboard > Instances running
24. We are going into webserver-01
25. Click connect > Connect under EC2 instance connect going into the terminal session
26. First thing in the console, we are going to take a look at what we have in the server
27. First command would be `lsblk`
 - a. Which shows block devices listed
28. As you can see we have our 10 GB volume mounted on /data
29. So in order to see what is in there type in `ls /data`
30. We have file 1-10

```
ubuntu@ip-10-0-0-100:~$ lsblk
NAME        MAJ:MIN RM  SIZE RO TYPE MOUNTPOINT
loop1         7:1      0   25M  1 loop /snap/amazon-ssm-agent/4046
loop2         7:2      0  114M  1 loop /snap/core/13425
loop4         7:4      0  55.5M  1 loop /snap/core18/2284
loop5         7:5      0  25.1M  1 loop /snap/amazon-ssm-agent/5656
loop6         7:6      0  103M  1 loop /snap/lxd/23367
loop7         7:7      0  61.9M  1 loop /snap/core20/1328
loop9         7:9      0  76.2M  1 loop /snap/lxd/22340
loop10        7:10     0  55.6M  1 loop /snap/core18/2538
loop11        7:11     0   62M  1 loop /snap/core20/1587
nvme0n1       259:0     0    8G   0 disk 
└─nvme0n1p1   259:1     0    8G   0 part /
nvme1n1       259:2     0   10G   0 disk /data
ubuntu@ip-10-0-0-100:~$ ls data
ls: cannot access 'data': No such file or directory
ubuntu@ip-10-0-0-100:~$ ls /data
file.01 file.02 file.03 file.04 file.05 file.06 file.07 file.08 file.09 file.10
ubuntu@ip-10-0-0-100:~$
```

31. Lets create a mount point or directory to attach our EFS volume
32. To do this, type in `sudo mkdir /efs`
33. You won't see any changes, but we did create a mount point. So lets head back over EFS in our other tab
34. Once there click the attach button
35. From here select the "Mount via IP"
 - a. Down below you will see a command listed for this so we can run on our server.
 - b. But before we do this, we have to make a small change
36. Copy that command line and go back into the terminal tab
37. Paste it in there but before you press enter we need to add the "/" before efs since it is missing

```
ubuntu@ip-10-0-0-100:~$ sudo mkdir /efs
ubuntu@ip-10-0-0-100:~$
```

38. Above you can see what I mean by adding in the "/"

39. We don't have anything in the efs. But you can type in the command to see what is in there by type `ls /efs`
40. Typing in `lsblk` to see if it is in our block devices, which it isn't there just yet
41. So lets view the mounts by typing in `mount`

```
ubuntu@ip-10-0-0-100:~$ mount
/dev/nvme0n1p1 on / type ext4 (rw,relatime,discard)
devtmpfs on /dev type devtmpfs (rw,relatime,size=476564k,nr_inodes=119141,mode=755,inode64)
sysfs on /sys type sysfs (rw,nosuid,nodev,noexec,relatime)
proc on /proc type proc (rw,nosuid,nodev,noexec,relatime)
securityfs on /sys/kernel/security type securityfs (rw,nosuid,nodev,noexec,relatime)
tmpfs on /dev/shm type tmpfs (rw,nosuid,nodev,inode64)
devpts on /dev/pts type devpts (rw,nosuid,noexec,relatime,gid=5,mode=620,ptmxmode=000)
tmpfs on /run type tmpfs (rw,nosuid,nodev,size=96636k,mode=755,inode64)
tmpfs on /run/lock type tmpfs (rw,nosuid,nodev,noexec,relatime,size=5120k,inode64)
tmpfs on /sys/fs/cgroup type tmpfs (ro,nosuid,nodev,noexec,mode=755,inode64)
cgroup2 on /sys/fs/cgroup/unified type cgroup2 (rw,nosuid,nodev,noexec,relatime,nsdelegate)
cgroup on /sys/fs/cgroup/systemd type cgroup (rw,nosuid,nodev,noexec,relatime,xattr,name=systemd)
pstore on /sys/fs/pstore type pstore (rw,nosuid,nodev,noexec,relatime)
none on /sys/fs/bpf type bpf (rw,nosuid,nodev,noexec,relatime,mode=700)
cgroup on /sys/fs/cgroup/pids type cgroup (rw,nosuid,nodev,noexec,relatime,pids)
cgroup on /sys/fs/cgroup/cpuset type cgroup (rw,nosuid,nodev,noexec,relatime,cpuset)
cgroup on /sys/fs/cgroup/blkio type cgroup (rw,nosuid,nodev,noexec,relatime,blkio)
cgroup on /sys/fs/cgroup/hugetlb type cgroup (rw,nosuid,nodev,noexec,relatime,hugetlb)
cgroup on /sys/fs/cgroup/perf_event type cgroup (rw,nosuid,nodev,noexec,relatime,perf_event)
cgroup on /sys/fs/cgroup/devices type cgroup (rw,nosuid,nodev,noexec,relatime,devices)
cgroup on /sys/fs/cgroup/net_cls,net_prio type cgroup (rw,nosuid,nodev,noexec,relatime,net_cls,net_prio)
cgroup on /sys/fs/cgroup/rdma type cgroup (rw,nosuid,nodev,noexec,relatime,rdma)
cgroup on /sys/fs/cgroup/cpu,cpuacct type cgroup (rw,nosuid,nodev,noexec,relatime,cpu,cpuacct)
cgroup on /sys/fs/cgroup/memory type cgroup (rw,nosuid,nodev,noexec,relatime,memory)
cgroup on /sys/fs/cgroup/freezer type cgroup (rw,nosuid,nodev,noexec,relatime,freezer)
systemd-1 on /proc/sys/fs/binfmt_misc type autofs (rw,relatime,fd=28,pgrp=1,timeout=0,minproto=5,maxproto=5,direct,pip
hugetlbfs on /dev/hugepages type hugetlbfs (rw,relatime,pagesize=2M)
mqueue on /dev/mqueue type mqueue (rw,nosuid,nodev,noexec,relatime)
debugfs on /sys/kernel/debug type debugfs (rw,nosuid,nodev,noexec,relatime)
tracefs on /sys/kernel/tracing type tracefs (rw,nosuid,nodev,noexec,relatime)
fusectl on /sys/fs/fuse/connections type fusectl (rw,nosuid,nodev,noexec,relatime)
configfs on /sys/kernel/config type configfs (rw,nosuid,nodev,noexec,relatime)
/var/lib/napd/snap/amazon-ssm-agent_4046.snap on /snap/amazon-ssm-agent/4046 type squashfs (ro,nodev,relatime,x-gdu.
/var/lib/napd/snap/core18_2284.snap on /snap/core18/2284 type squashfs (ro,nodev,relatime,x-gdu.hide)
/var/lib/napd/snap/core20_1328.snap on /snap/core20/1328 type squashfs (ro,nodev,relatime,x-gdu.hide)
/var/lib/napd/snap/lxd_22340.snap on /snap/lxd/22340 type squashfs (ro,nodev,relatime,x-gdu.hide)
tmpfs on /run/napd/ns type tmpfs (rw,nosuid,nodev,size=96636k,mode=755,inode64)
/dev/nvme1n1 on /data type xfs (rw,relatime,attr2,inode64,logbufs=8,logbsize=32k,sunit=8,swidth=8,noquota)
/var/lib/napd/snap/core18_2538.snap on /snap/core18/2538 type squashfs (ro,nodev,relatime,x-gdu.hide)
/var/lib/napd/snap/core20_1587.snap on /snap/core20/1587 type squashfs (ro,nodev,relatime,x-gdu.hide)
/var/lib/napd/snap/core_13425.snap on /snap/core/13425 type squashfs (ro,nodev,relatime,x-gdu.hide)
/var/lib/napd/snap/amazon-ssm-agent_5656.snap on /snap/amazon-ssm-agent/5656 type squashfs (ro,nodev,relatime,x-gdu.
/var/lib/napd/snap/lxd_23367.snap on /snap/lxd/23367 type squashfs (ro,nodev,relatime,x-gdu.hide)
nsfs on /run/napd/ns/lxd.mnt type nsfs (rw)
tmpfs on /run/user/1000 type tmpfs (rw,nosuid,nodev,relatime,size=96636k,mode=700,uid=1000,gid=1000,inode64)
10.0.0.63:/ on /efs type nfs4 (rw,relatime,vers=4.1,rsize=1048576,wsize=1048576,namlen=255,hard,noresvport,proto=tcp,t
ans=2,sec=sys,clientaddr=10.0.0.100,local_lock=none,addr=10.0.0.63)
ubuntu@ip-10-0-0-100:~$
```

42. In the above screenshot you can see on the bottom is our NFS share is mounted on /efs type nfs4
43. We can also view this information with `df -h`
 - a. Df shows the amount of free disk space on each mounted disk
44. And again you can see our NFS share is mounted on /efs

```
ubuntu@ip-10-0-0-100:~$ df -h
Filesystem      Size  Used Avail Use% Mounted on
/dev/root        7.7G  3.2G  4.5G  42% /
devtmpfs         466M    0  466M   0% /dev
tmpfs            472M    0  472M   0% /dev/shm
tmpfs            95M   828K   94M   1% /run
tmpfs            5.0M    0   5.0M   0% /run/lock
tmpfs            472M    0  472M   0% /sys/fs/cgroup
/dev/loop1       25M   25M    0 100% /snap/amazon-ssm-agent/4046
/dev/loop4       56M   56M    0 100% /snap/core18/2284
/dev/loop7       62M   62M    0 100% /snap/core20/1328
/dev/loop9       77M   77M    0 100% /snap/lxd/22340
/dev/nvme1n1     10G  105M   9.9G   2% /data
/dev/loop10      56M   56M    0 100% /snap/core18/2538
/dev/loop11      62M   62M    0 100% /snap/core20/1587
/dev/loop2       114M  114M    0 100% /snap/core/13425
/dev/loop5        26M   26M    0 100% /snap/amazon-ssm-agent/5656
/dev/loop6       104M  104M    0 100% /snap/lxd/23367
tmpfs            95M    0   95M   0% /run/user/1000
10.0.0.63:/      8.0E    0  8.0E   0% /efs
ubuntu@ip-10-0-0-100:~$
```

45. So of course after typing in the command `lsblk` and seeing that 10 GB is still stored there. We would have to move our files from `/data` to `efs`

46. To do this type in `sudo rsync -rav /data/* /efs`

- a. R= recursive
- b. A= retain permissions
- c. V= verbose
- d. `/data/*` = everything in it to push it to `efs`

```
10.0.0.11:/      8.0E    0  8.0E   0% /efs
ubuntu@ip-10-0-0-100:~$ sudo rsync -rav /data/* /efs
sending incremental file list
file.01
file.02
file.03
file.04
file.05
file.06
file.07
file.08
file.09
file.10

sent 46,659 bytes  received 206 bytes  31,243.33 bytes/sec
total size is 46,080  speedup is 0.98
ubuntu@ip-10-0-0-100:~$
```

47. To verify if they moved over type in `ls /efs`

48. There you can see the files 1-10 have been moved into our EFS directory

49. So now we need to remove our data and remove the 10 GB volume all together

50. Type in `sudo umount /data`

51. Next would be to edit our `fstab` so that the system doesn't try to mount to this file system on reboot

52. To complete that task, type in `sudo nano /efs/fstab`

53. You will see the screen somewhat clear, but we need to get rid of the UUID line

- a. UUID = universally unique identifier of a partition
- b. Press `ctrl + K` to delete the line all at once

54. So now we need our file system ID so we head back to EFS

55. The attach window may still be there if not simply click attach and make sure the “Mount via IP” is still selected highlight and copy the IP address listed there
- The IP address listed below is our file system ID

Using the NFS client:

```
sudo mount -t nfs4 -o nfsvers=4.1,rsize=1048576,wsiz=1048576,hard,timeo=600,retrans=2,noresvport 10.0.0.11:/ efs
```

See our user guide for more information. [User guide](#)

56. Once you copy that, go back into the terminal and paste it in the file

57. Once there tab over and type /data

58. Tab again with 1 space and type in nfs4

```
10.0.0.11:/ /data nfs4
```

59. The next part to this is to add in our options. So we head back to our EFS tab

60. The options are going to start right after the -o and end right before the IP address

Using the NFS client:

```
sudo mount -t nfs4 -o nfsvers=4.1,rsize=1048576,wsiz=1048576,hard,timeo=600,retrans=2,noresvport 10.0.0.11:/ efs
```

See our user guide for more information. [User guide](#)

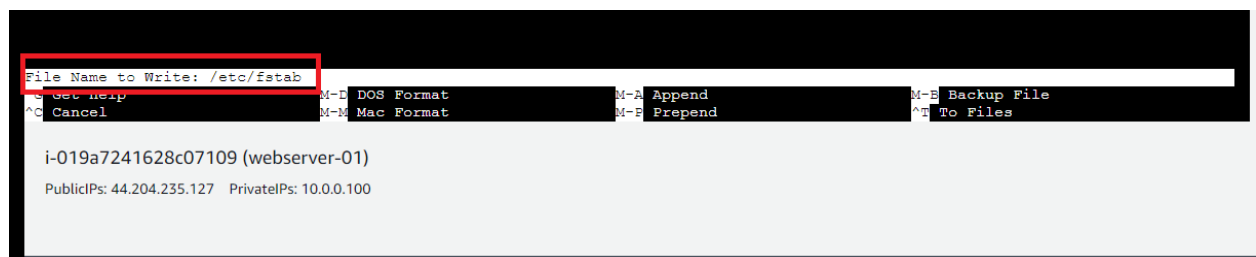
61. Once copied go back into the terminal and press the spacebar and type 0 0 to finish off this mount point line

62. Now lets exit Nano with Ctrl + X

63. Press Y for saving changes

64. And make sure the file is entered if not, enter it

65.



66. Once entered press Y again

67. Next we would verify if this worked by typing `sudo umount /efs`

68. Type in `df -h`

69. Here you can see our /data nor /efs is mounted

70. Type in `sudo mount -a`

71. Type in `df -h`

72. We can see it is on the /data and by typing in `ls /data` we can see files 1-10

73. Go back to EC2 and delete the volume
74. On first tab click on the EC2 tab > Volumes
75. Here we see 6 volumes

Volumes (6)						
<input type="text" value="Search"/>						
<input type="checkbox"/>	Name	Volume ID	Type	Size	IOPS	
<input type="checkbox"/>	-	vol-0d60cfbdc246e3c5	gp2	10 GiB	100	
<input type="checkbox"/>	-	vol-07382c5619b808e19	gp2	10 GiB	100	
<input type="checkbox"/>	-	vol-023e2c64852bfc48f	gp2	10 GiB	100	
<input type="checkbox"/>	-	vol-0f655341e33fbae35	standard	8 GiB	-	
<input type="checkbox"/>	-	vol-04848cf4d1940f833	standard	8 GiB	-	
<input type="checkbox"/>	-	vol-0636916549f8eb79e	standard	8 GiB	-	

76. We know the volume is a 10 GB volume
77. But to be sure you are removing the correct one scroll to the right and expand the attached instances section to see the correct server

Attached Instances		Vol
+	i-0b3e37b78155b55b3 (webserver-03): /dev/sdf (atta...	✓
+	i-017c8121887936dba (webserver-02): /dev/sdf (atta...	✓
+	i-019a7241628c07109 (webserver-01): /dev/sdf (atta...	✓
+	i-017c8121887936dba (webserver-02): /dev/sda1 (att...	✓
+	i-0b3e37b78155b55b3 (webserver-03): /dev/sda1 (at...	✓
+	i-019a7241628c07109 (webserver-01): /dev/sda1 (att...	✓

78. Click in an empty space to select it rather than scroll back over and select it. This will cause room for error
79. Actions > Detach Volume > Detach
80. If you scroll to the volume state you can see it is listed as available
81. Select the volume again > Actions > Delete Volume > Delete
82. Now it is using the EFS share

That is now completed, eventually you would want to do this for the others but it's a repetitive process.