
AWS
re:Invent

STG306

# Security best practices with Amazon EFS

Vasily Pantyukhin

Senior Solutions Architect, Storage

AWS

AWS re:Invent

© 2020, Amazon Web Services, Inc. or its affiliates. All rights reserved.

# Agenda

Amazon Elastic File System (Amazon EFS) overview

Security best practices

Compliance

Summary



AWS re:Invent

© 2020, Amazon Web Services, Inc. or its affiliates. All rights reserved.

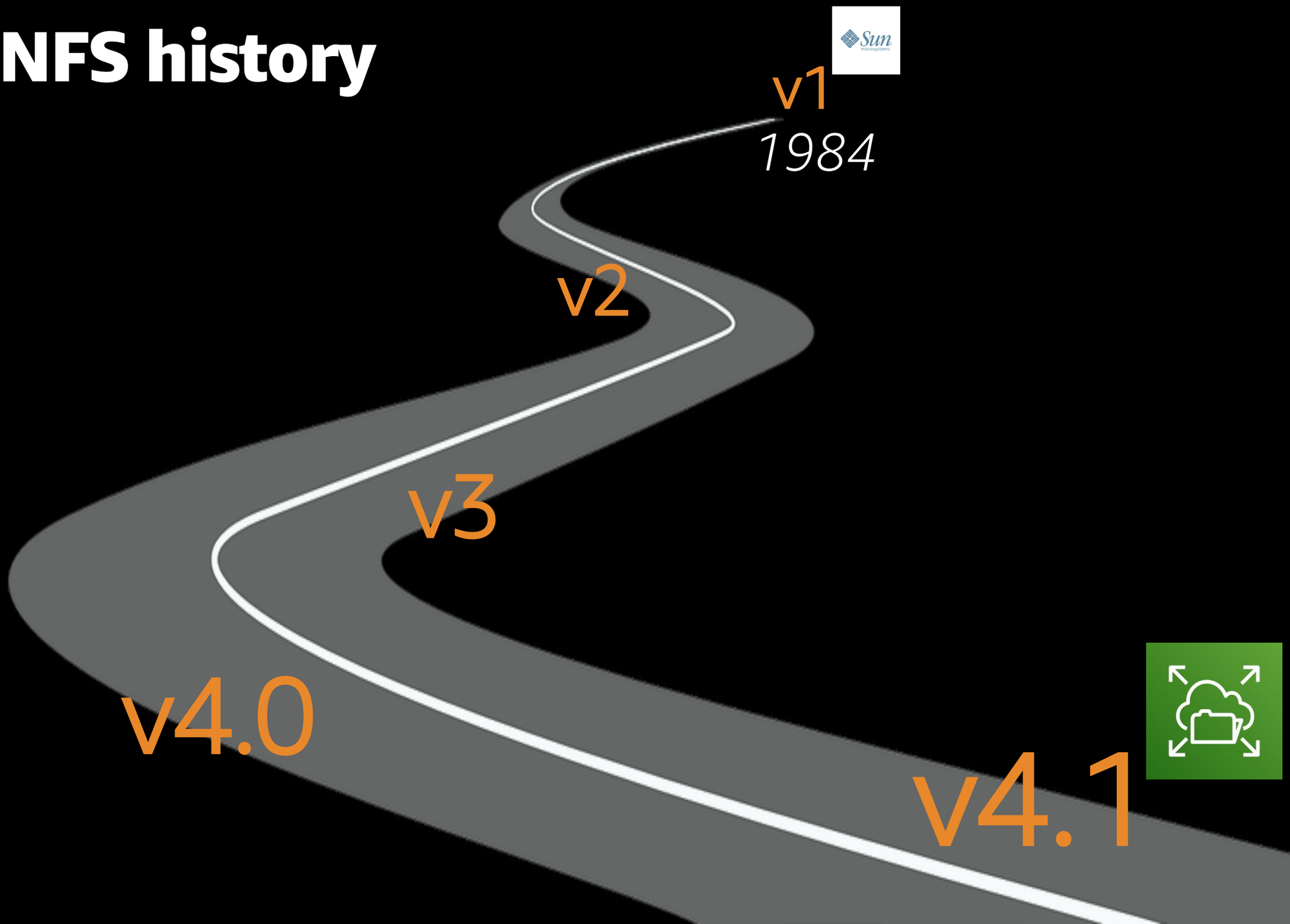
# Amazon EFS overview



AWS re:Invent

© 2020, Amazon Web Services, Inc. or its affiliates. All rights reserved.

# NFS history



# Amazon Elastic File System



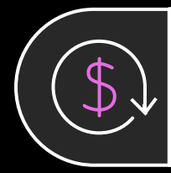
Amazon EFS is a fully managed file system that is . . .



**Cloud native**



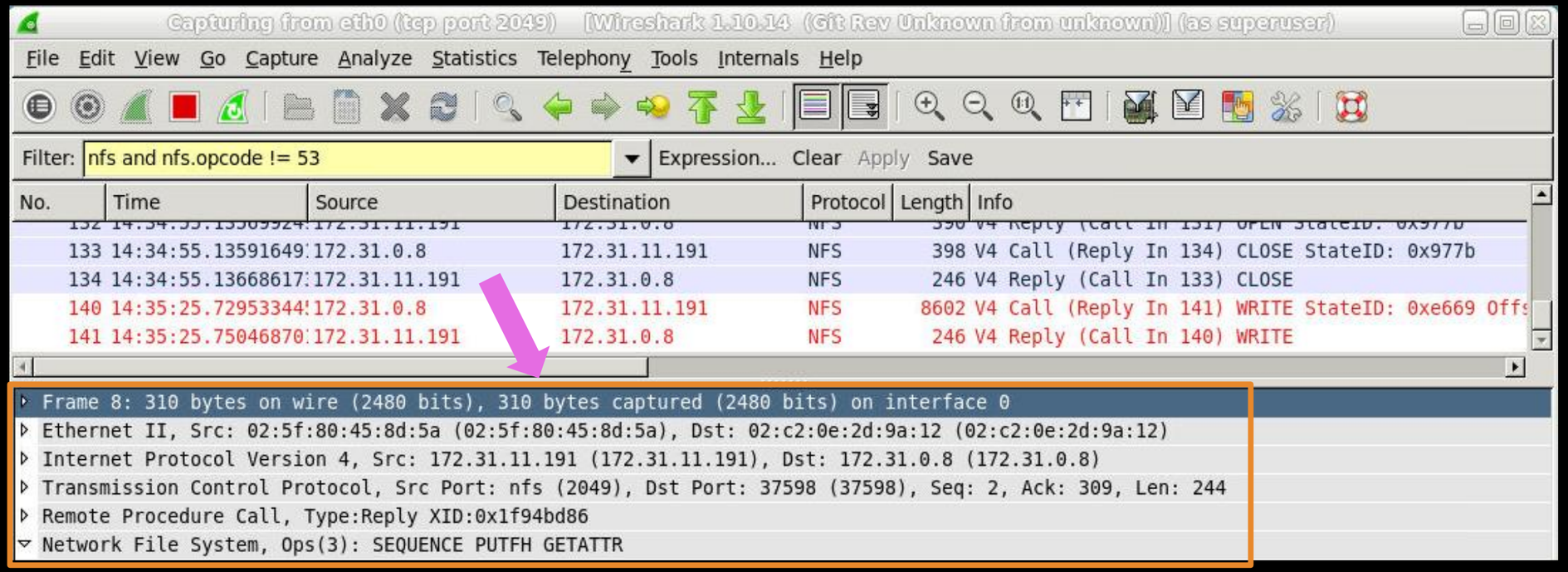
**Highly reliable**



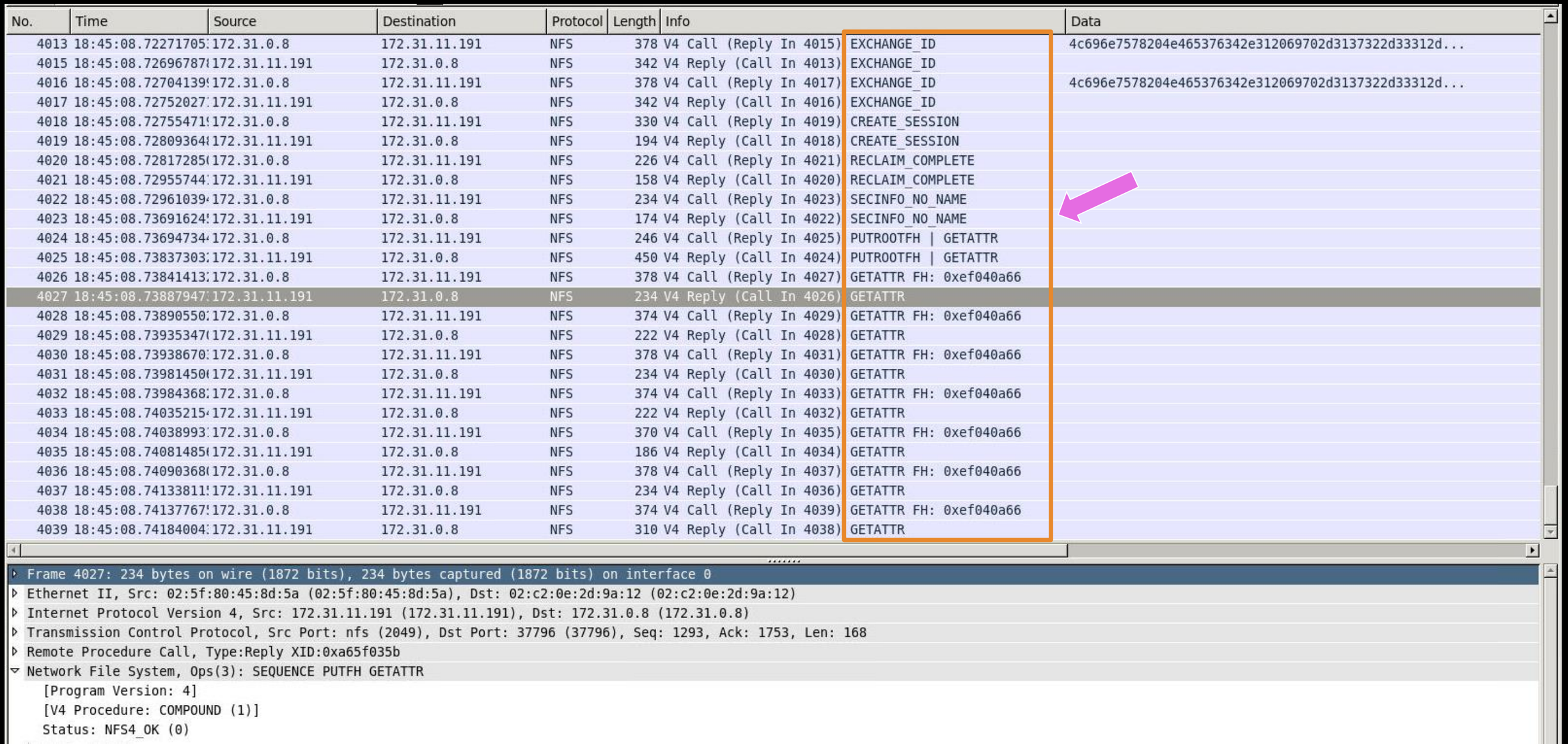
**Cost optimized**

# Protocol stack

- Application — *NFS*
- Presentation — *XDR*
- Session — *RPC*
- Transport
- Network
- Data link
- Physical



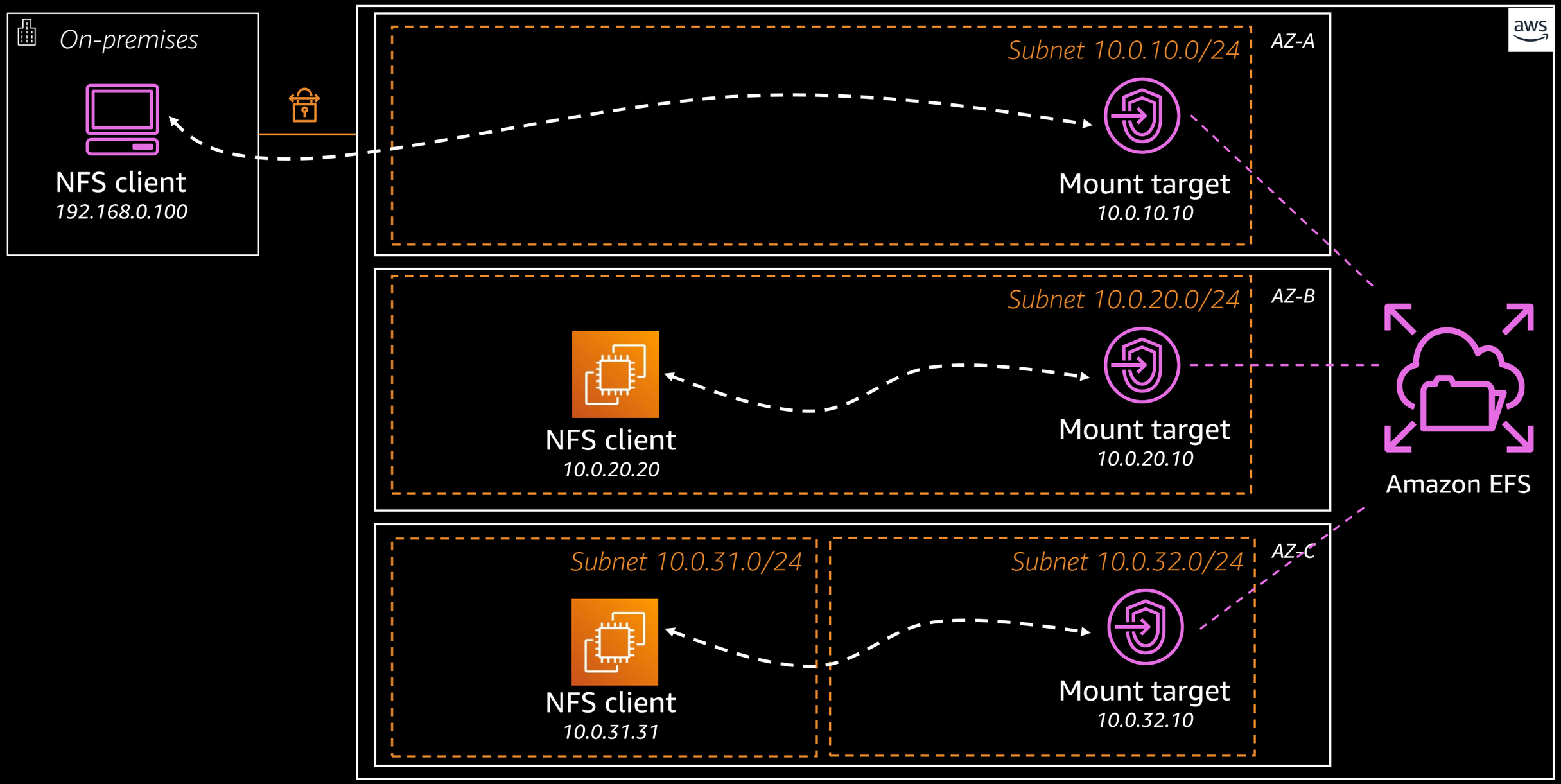
# Example: NFS mount



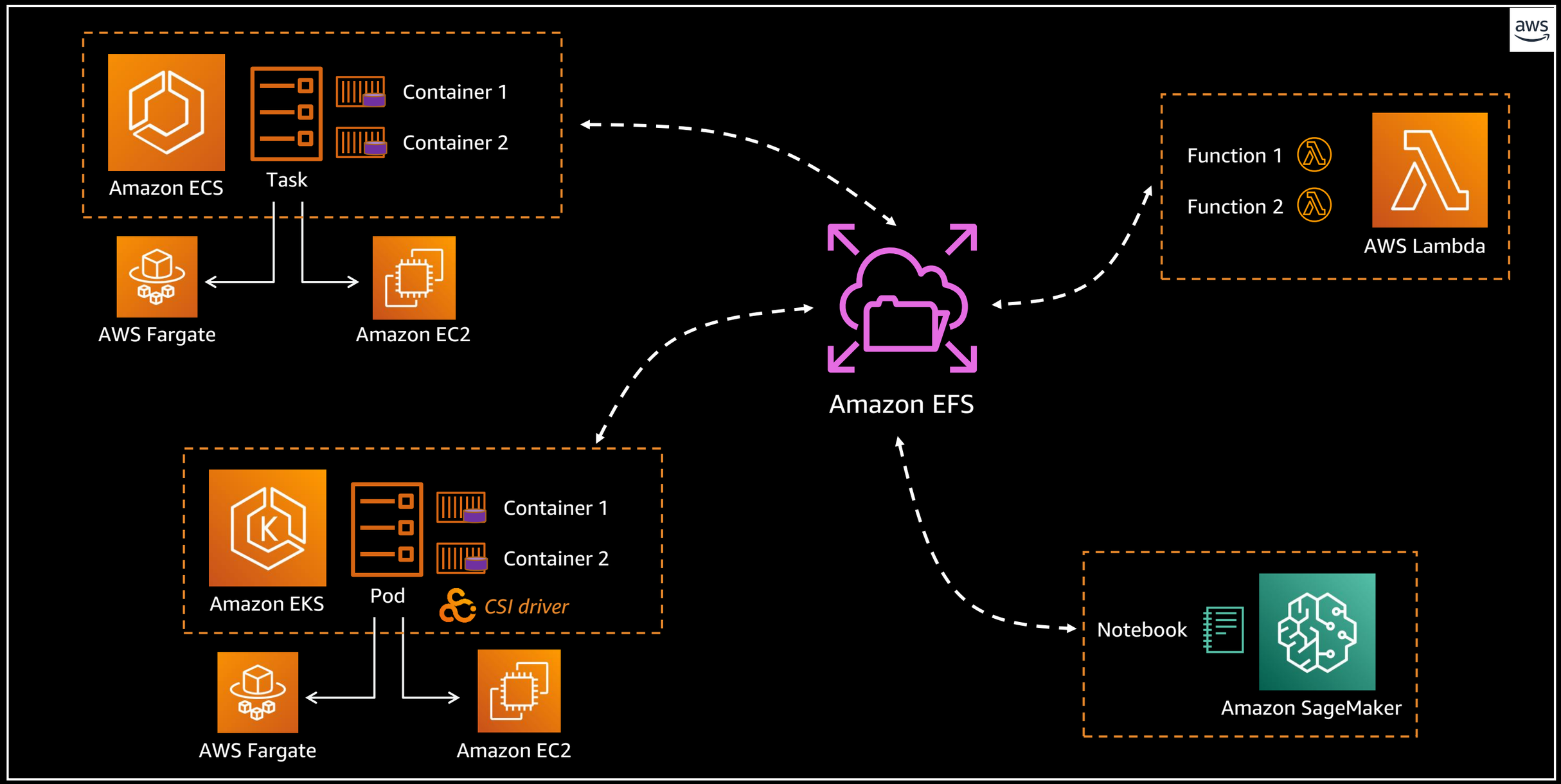
| No. | Time | Source | Destination | Protocol | Length | Info | Data |
|---|---|---|---|---|---|---|---|
| 4013 | 18:45:08.72271705 | 172.31.0.8 | 172.31.11.191 | NFS | 378 | V4 Call (Reply In 4015) EXCHANGE_ID | 4c696e7578204e465376342e312069702d3137322d33312d... |
| 4015 | 18:45:08.72696787 | 172.31.11.191 | 172.31.0.8 | NFS | 342 | V4 Reply (Call In 4013) EXCHANGE_ID | |
| 4016 | 18:45:08.72704139 | 172.31.0.8 | 172.31.11.191 | NFS | 378 | V4 Call (Reply In 4017) EXCHANGE_ID | 4c696e7578204e465376342e312069702d3137322d33312d... |
| 4017 | 18:45:08.72752027 | 172.31.11.191 | 172.31.0.8 | NFS | 342 | V4 Reply (Call In 4016) EXCHANGE_ID | |
| 4018 | 18:45:08.72755471 | 172.31.0.8 | 172.31.11.191 | NFS | 330 | V4 Call (Reply In 4019) CREATE_SESSION | |
| 4019 | 18:45:08.72809364 | 172.31.11.191 | 172.31.0.8 | NFS | 194 | V4 Reply (Call In 4018) CREATE_SESSION | |
| 4020 | 18:45:08.72817285 | 172.31.0.8 | 172.31.11.191 | NFS | 226 | V4 Call (Reply In 4021) RECLAIM_COMPLETE | |
| 4021 | 18:45:08.72955744 | 172.31.11.191 | 172.31.0.8 | NFS | 158 | V4 Reply (Call In 4020) RECLAIM_COMPLETE | |
| 4022 | 18:45:08.72961039 | 172.31.0.8 | 172.31.11.191 | NFS | 234 | V4 Call (Reply In 4023) SECINFO_NO_NAME | |
| 4023 | 18:45:08.73691624 | 172.31.11.191 | 172.31.0.8 | NFS | 174 | V4 Reply (Call In 4022) SECINFO_NO_NAME | |
| 4024 | 18:45:08.73694734 | 172.31.0.8 | 172.31.11.191 | NFS | 246 | V4 Call (Reply In 4025) PUTROOTFH \| GETATTR | |
| 4025 | 18:45:08.73837303 | 172.31.11.191 | 172.31.0.8 | NFS | 450 | V4 Reply (Call In 4024) PUTROOTFH \| GETATTR | |
| 4026 | 18:45:08.73841413 | 172.31.0.8 | 172.31.11.191 | NFS | 378 | V4 Call (Reply In 4027) GETATTR FH: 0xef040a66 | |
| 4027 | 18:45:08.73887947 | 172.31.11.191 | 172.31.0.8 | NFS | 234 | V4 Reply (Call In 4026) GETATTR | |
| 4028 | 18:45:08.73890550 | 172.31.0.8 | 172.31.11.191 | NFS | 374 | V4 Call (Reply In 4029) GETATTR FH: 0xef040a66 | |
| 4029 | 18:45:08.73935347 | 172.31.11.191 | 172.31.0.8 | NFS | 222 | V4 Reply (Call In 4028) GETATTR | |
| 4030 | 18:45:08.73938670 | 172.31.0.8 | 172.31.11.191 | NFS | 378 | V4 Call (Reply In 4031) GETATTR FH: 0xef040a66 | |
| 4031 | 18:45:08.73981450 | 172.31.11.191 | 172.31.0.8 | NFS | 234 | V4 Reply (Call In 4030) GETATTR | |
| 4032 | 18:45:08.73984368 | 172.31.0.8 | 172.31.11.191 | NFS | 374 | V4 Call (Reply In 4033) GETATTR FH: 0xef040a66 | |
| 4033 | 18:45:08.74035215 | 172.31.11.191 | 172.31.0.8 | NFS | 222 | V4 Reply (Call In 4032) GETATTR | |
| 4034 | 18:45:08.74038993 | 172.31.0.8 | 172.31.11.191 | NFS | 370 | V4 Call (Reply In 4035) GETATTR FH: 0xef040a66 | |
| 4035 | 18:45:08.74081485 | 172.31.11.191 | 172.31.0.8 | NFS | 186 | V4 Reply (Call In 4034) GETATTR | |
| 4036 | 18:45:08.74090368 | 172.31.0.8 | 172.31.11.191 | NFS | 378 | V4 Call (Reply In 4037) GETATTR FH: 0xef040a66 | |
| 4037 | 18:45:08.74133811 | 172.31.11.191 | 172.31.0.8 | NFS | 234 | V4 Reply (Call In 4036) GETATTR | |
| 4038 | 18:45:08.74137767 | 172.31.0.8 | 172.31.11.191 | NFS | 374 | V4 Call (Reply In 4039) GETATTR FH: 0xef040a66 | |
| 4039 | 18:45:08.74184004 | 172.31.11.191 | 172.31.0.8 | NFS | 310 | V4 Reply (Call In 4038) GETATTR | |

```
▷ Frame 4027: 234 bytes on wire (1872 bits), 234 bytes captured (1872 bits) on interface 0
▷ Ethernet II, Src: 02:5f:80:45:8d:5a (02:5f:80:45:8d:5a), Dst: 02:c2:0e:2d:9a:12 (02:c2:0e:2d:9a:12)
▷ Internet Protocol Version 4, Src: 172.31.11.191 (172.31.11.191), Dst: 172.31.0.8 (172.31.0.8)
▷ Transmission Control Protocol, Src Port: nfs (2049), Dst Port: 37796 (37796), Seq: 1293, Ack: 1753, Len: 168
▷ Remote Procedure Call, Type:Reply XID:0xa65f035b
▽ Network File System, Ops(3): SEQUENCE PUTFH GETATTR
    [Program Version: 4]
    [V4 Procedure: COMPOUND (1)]
    Status: NFS4_OK (0)
```

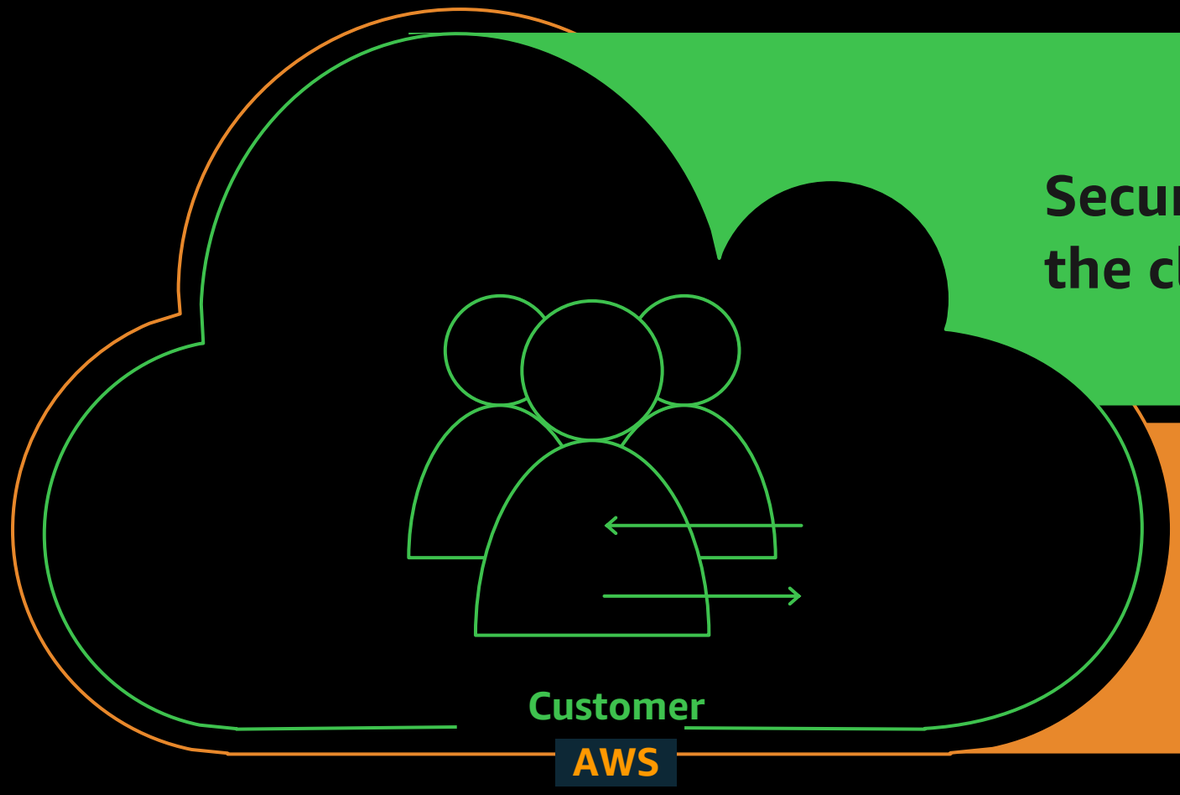
# Amazon EC2 and on-premises



# Cloud native



# Shared responsibility model




**Security *in* the cloud**

Customer responsibility is determined by the AWS Cloud services that a customer selects

**Security *of* the cloud**

AWS is responsible for protecting the infrastructure that runs all of the services offered in the AWS Cloud

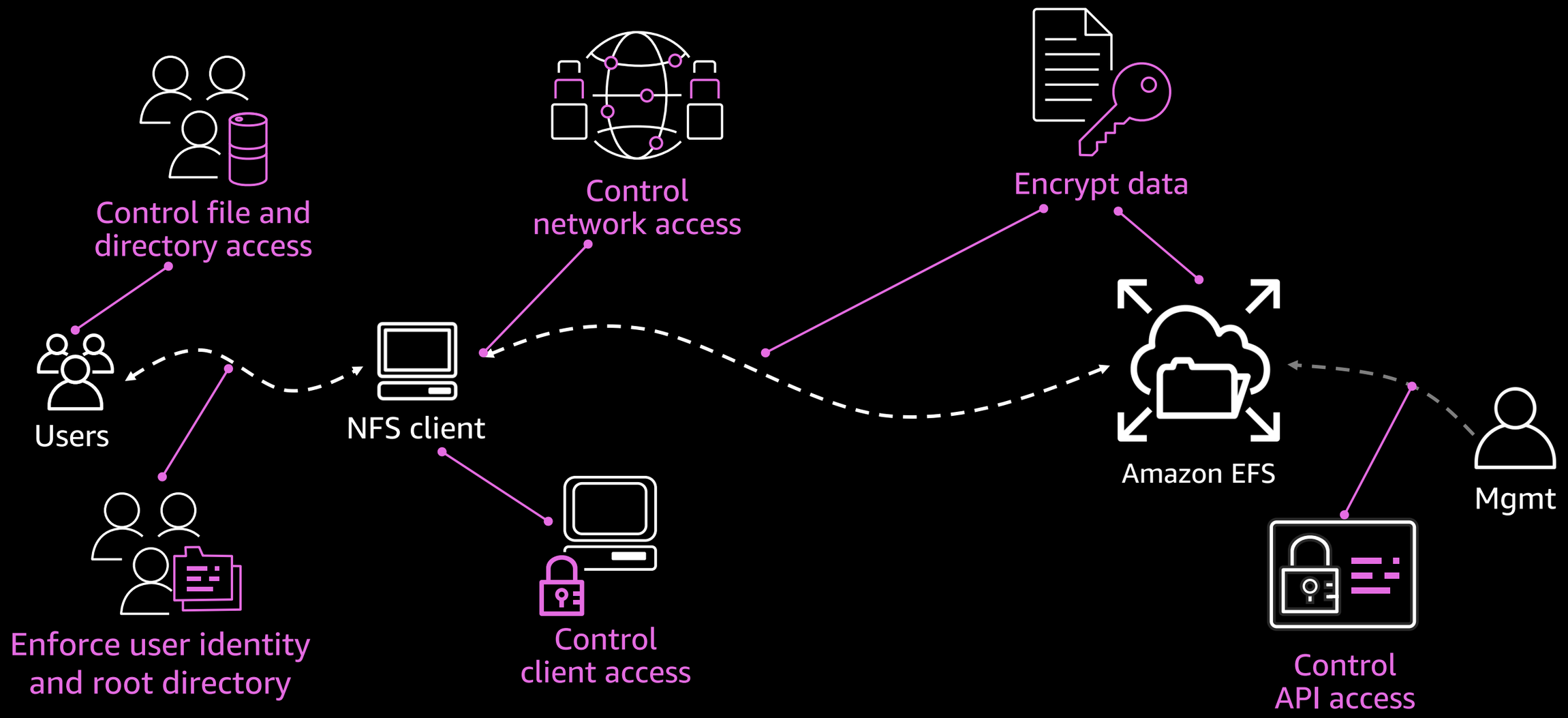
We help secure companies' services in the cloud

# Security best practices

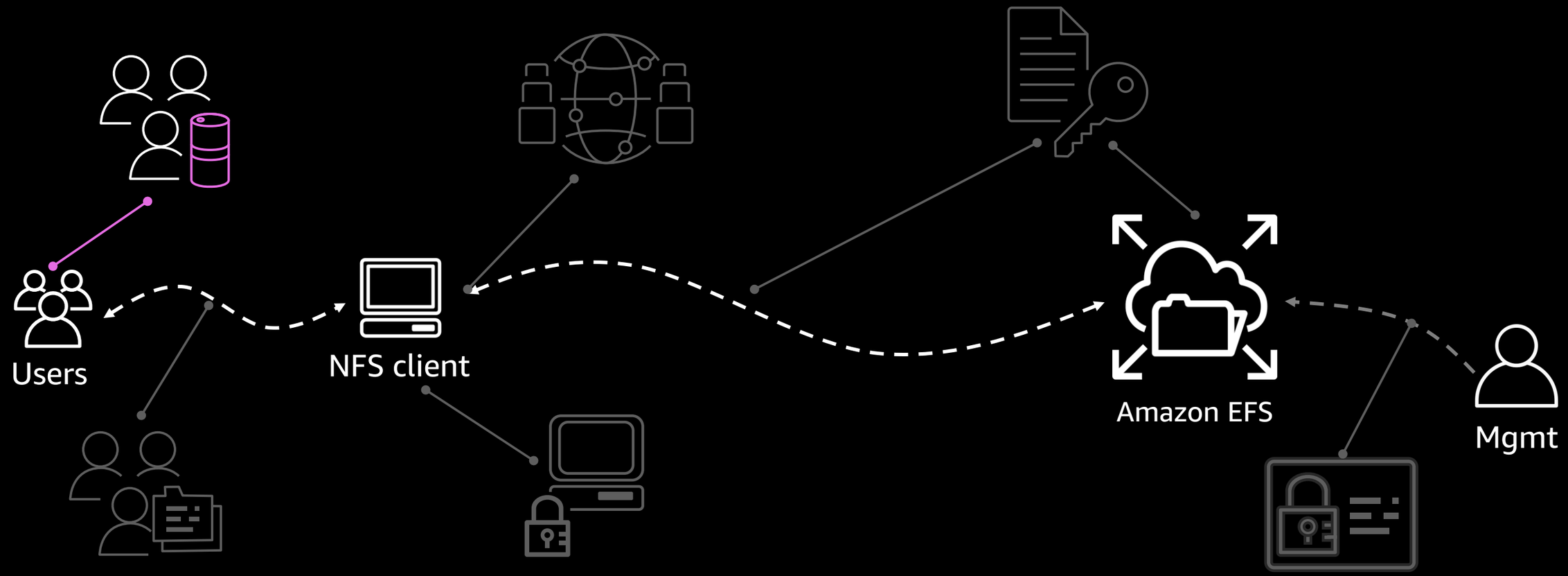


AWS re:Invent

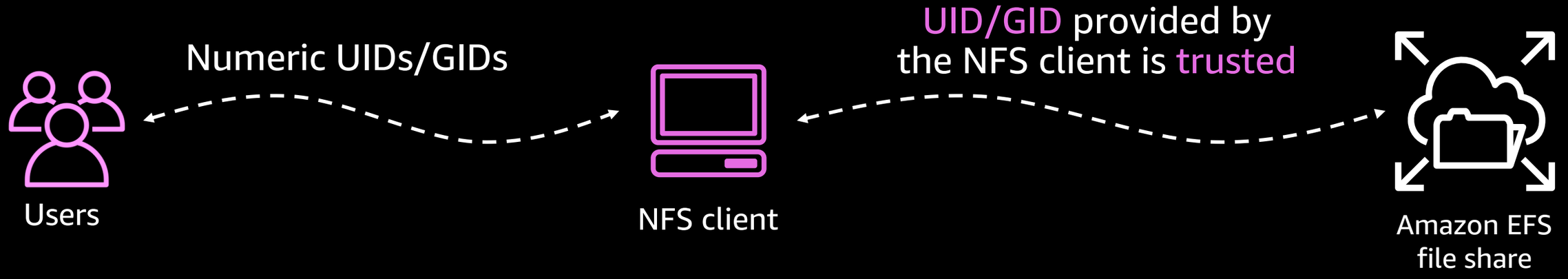
© 2020, Amazon Web Services, Inc. or its affiliates. All rights reserved.


Control file and directory access
Control network access
Encrypt data
Users
NFS client
Amazon EFS
Mgmt
Enforce user identity and root directory
Control client access
Control API access

# Control file and directory access



# Restricting file/directory access with POSIX




```
$ ls -ld /efs
drwxrwxr-x 2 root friends 6144 Oct  4 09:32 /efs

$ id
uid=5001(bob) gid=5000(friends)

$ echo "Hello Bob" > /efs/bobs-file
```

Bob has R/W permissions as
the member of "friends" group

```
$ id
uid=5002(alice) gid=5000(friends)

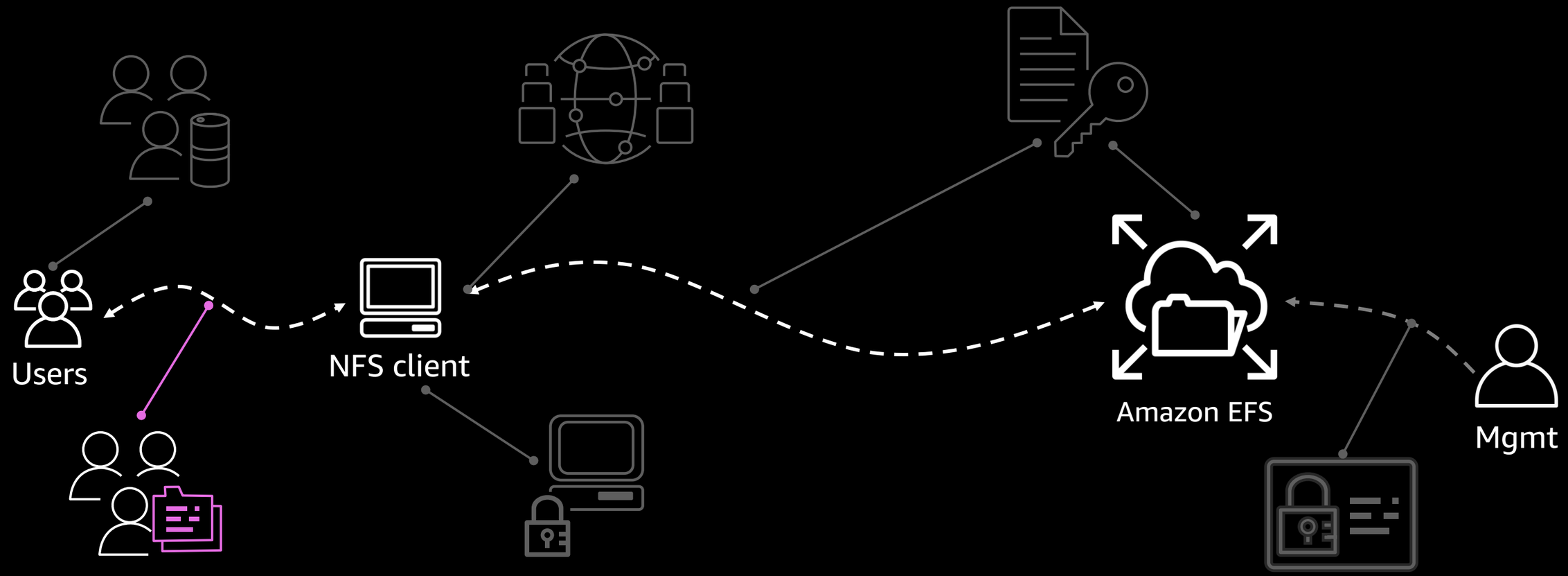
$ ls -l /efs/bobs-file
-rw-r--r-- 1 bob friends 10 Oct  4 09:36 bobs-file

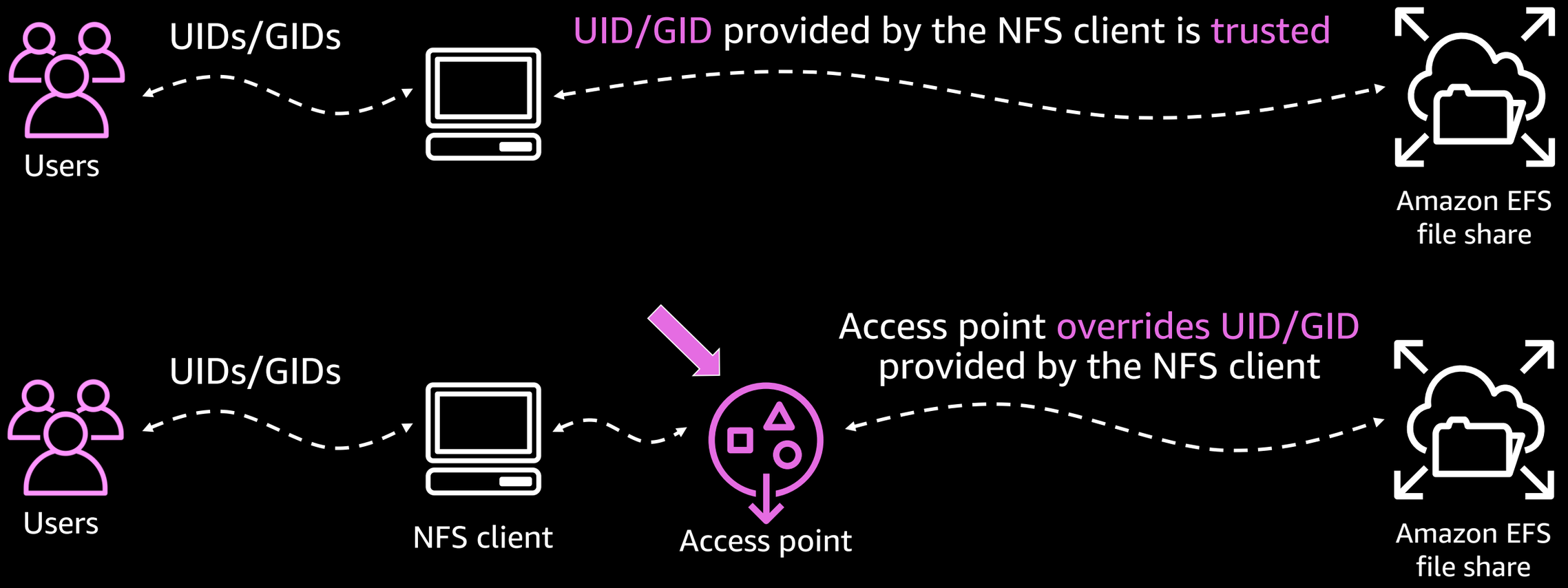
$ cat /efs/bobs-file
Hello Bob

$ echo "Hello Alice" > /efs/bobs-file
bash: /efs/bobs-file: Permission denied
```

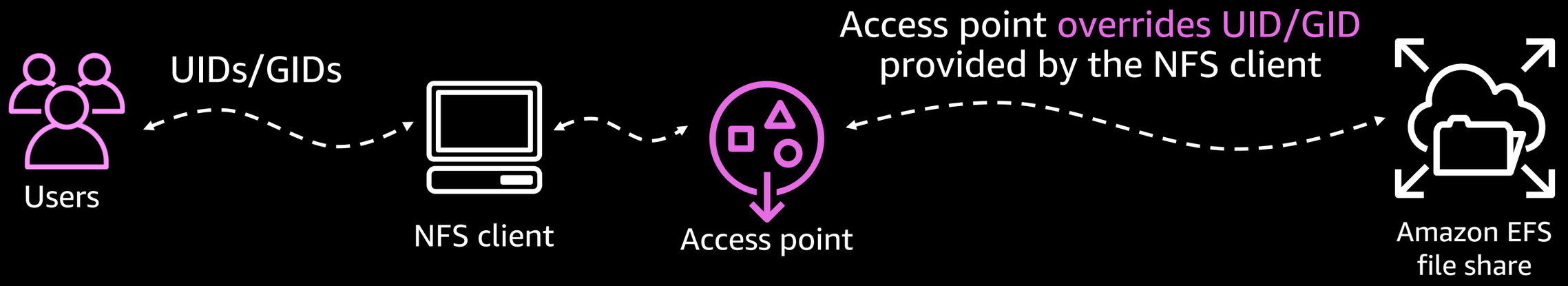
Alice has R/o permissions as
the member of "friends" group

# Enforce user identity and root directory




UIDs/GIDs
UID/GID provided by the NFS client is trusted
Users
Amazon EFS
file share
UIDs/GIDs
Access point overrides UID/GID
provided by the NFS client
Users
NFS client
Access point
Amazon EFS
file share

# Enforce user identity




Amazon EFS > Access points > fsap-03497eb29c1d115b

## Alice AP (fsap-03497eb29c1d115b9)

Delete | Attach

### Details

File system ID
fs-8e5ed6f6

Root directory path
/

State
Available

**POSIX user**

User ID
5002
User "alice"

Group ID
5000
Group "friends"

Secondary group IDs
-

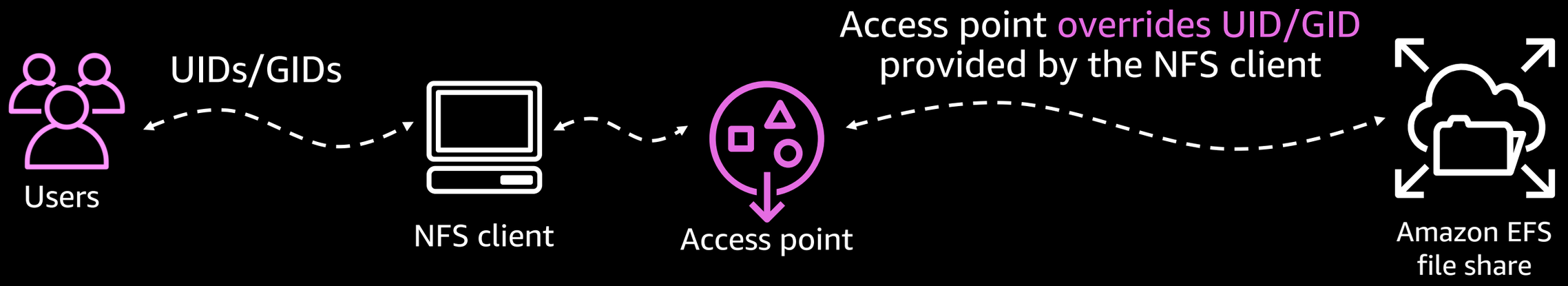
**Root directory creation permissions**

Owner user ID
-

Owner group ID
-

Permissions
-

# Enforce user identity




```
$ mount -t efs -o tls,accesspoint=fsap-03497eb29c1d115b9 fs-8e5ed6f6:/ /efs-accesspoint/

$ ls -ld /efs-accesspoint/
drwxrwxr-x 2 root friends 6144 Oct  4 09:32 /efs-accesspoint/

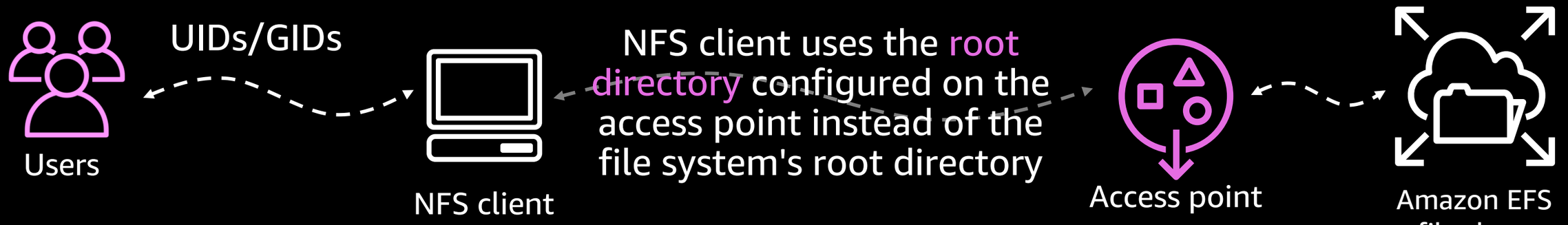
$ id
uid=5001(bob) gid=5000(friends)
$ echo "Hello Alice" > /efs-accesspoint/alices-file

$ ls -l /efs-accesspoint/alices-file
-rw-r--r-- 1 alice friends 12 Oct  4 11:58 /efs-accesspoint/alices-file
```

Bob has R/W permissions as the member of ”friends” group

Access point overrides Bob’s permission by Alice’s permissions

# Enforce root directory




Amazon EFS > Access points > fsap-01c8e21ac04ab544

## Alice directory AP (fsap-01c8e21ac04ab544b)

Delete | Attach

### Details

File system ID
fs-8e5ed6f6

Root directory path
/alices-directory

State
Available

**POSIX user**

User ID
5002
User "alice"

Group ID
5000
Group "friends"

Secondary group IDs
-

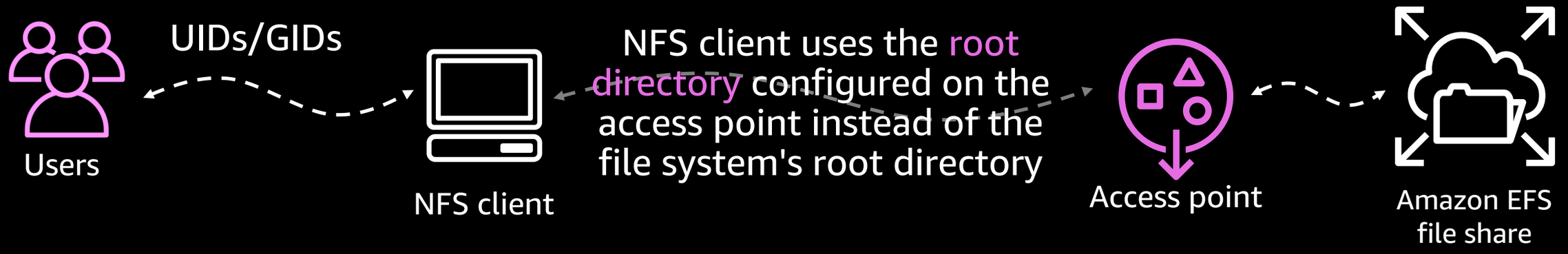
**Root directory creation permissions**

Owner user ID
5002

Owner group ID
5000

Permissions
775

# Applications see only the data they need




```
$ mount -t efs -o tls,accesspoint=fsap-01c8e21ac04ab544b fs-8e5ed6f6:/ /efs-accesspoint/

$ ls -ld /efs/alices-directory/
drwxrwxr-x 2 alice friends 6144 Oct  4 13:00 /efs/alices-directory/

$ id
uid=5001(bob) gid=5000(friends)
$ echo "Hello Alice" > /efs-accesspoint/alices-file

$ ls -l /efs-accesspoint/alices-file
-rw-r--r-- 1 alice friends 12 Oct  4 13:04 /efs-accesspoint/alices-file

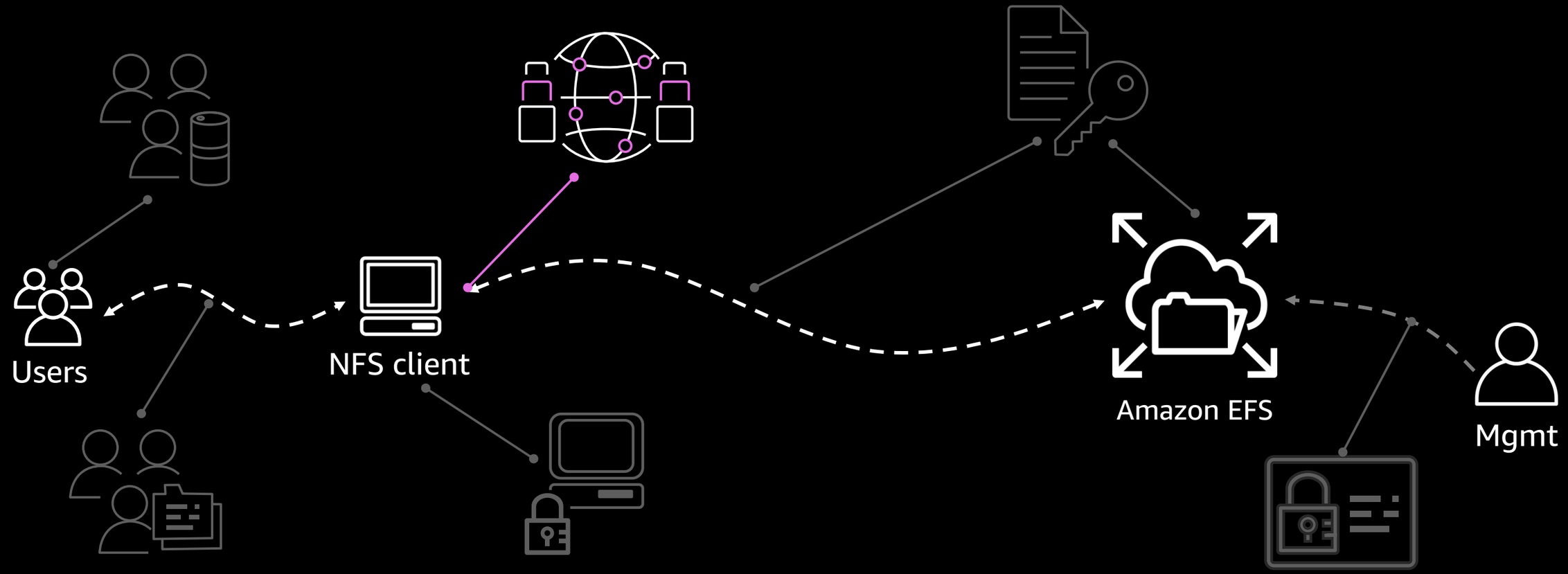
$ ls -l /efs/alices-directory/alices-file
-rw-r--r-- 1 alice friends 12 Oct  4 13:04 /efs/alices-directory/alices-file
```

New root directory for access point is created

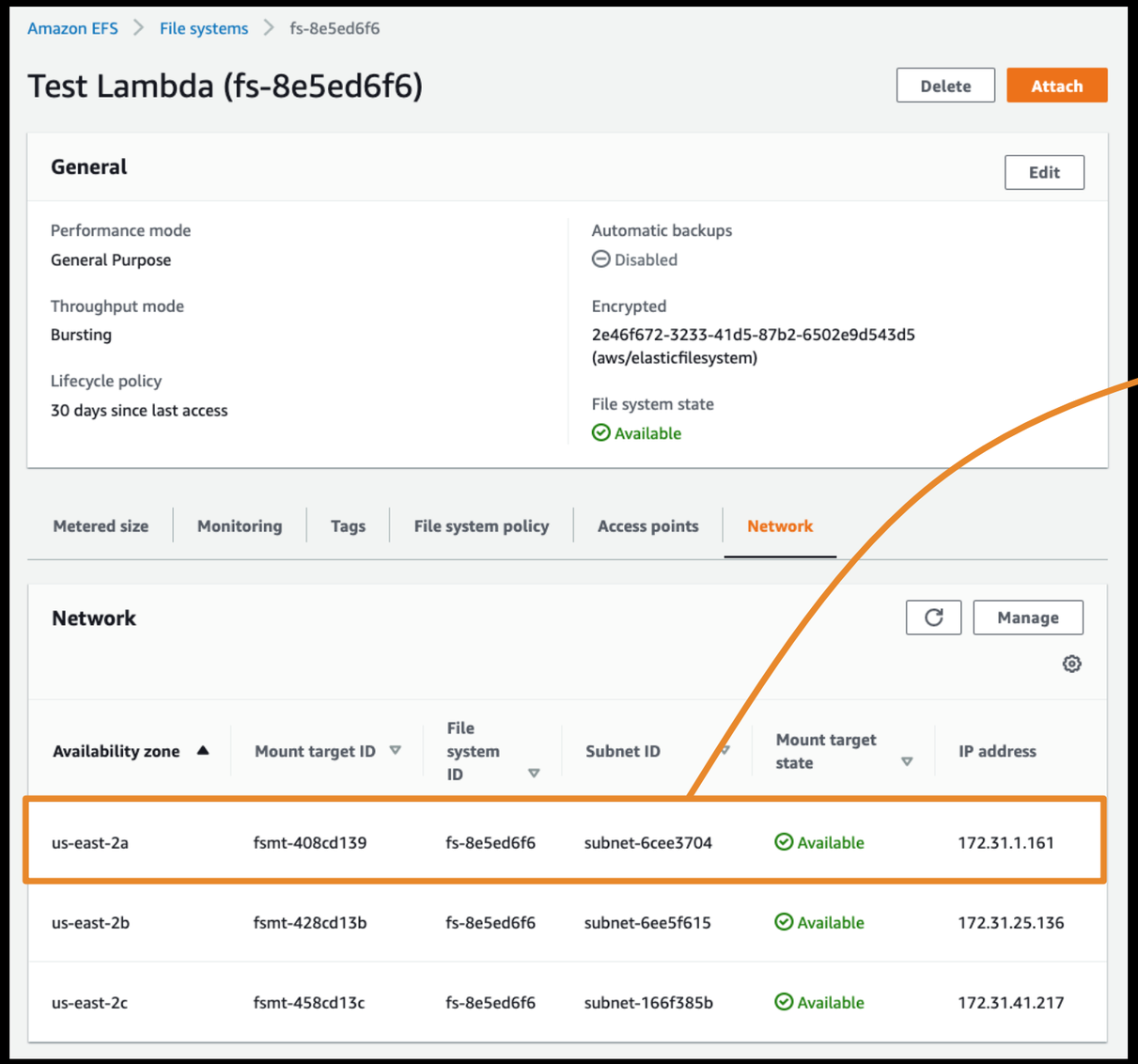
Access point enforces new root directory hiding other file system directories

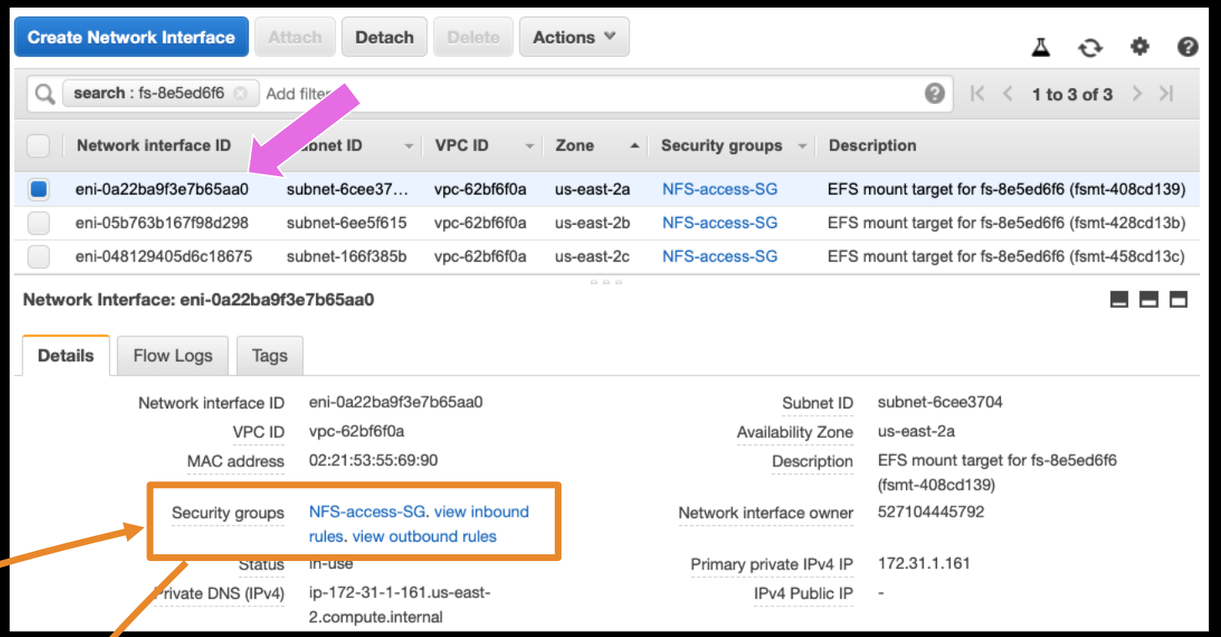
Directory is visible in the file system structure

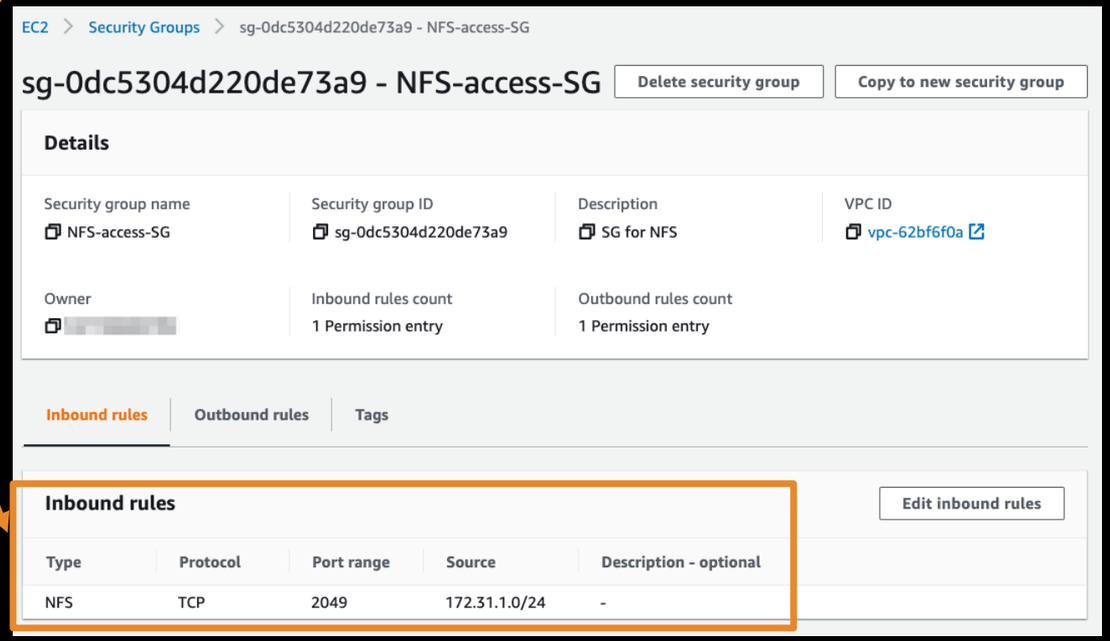
# Control network access



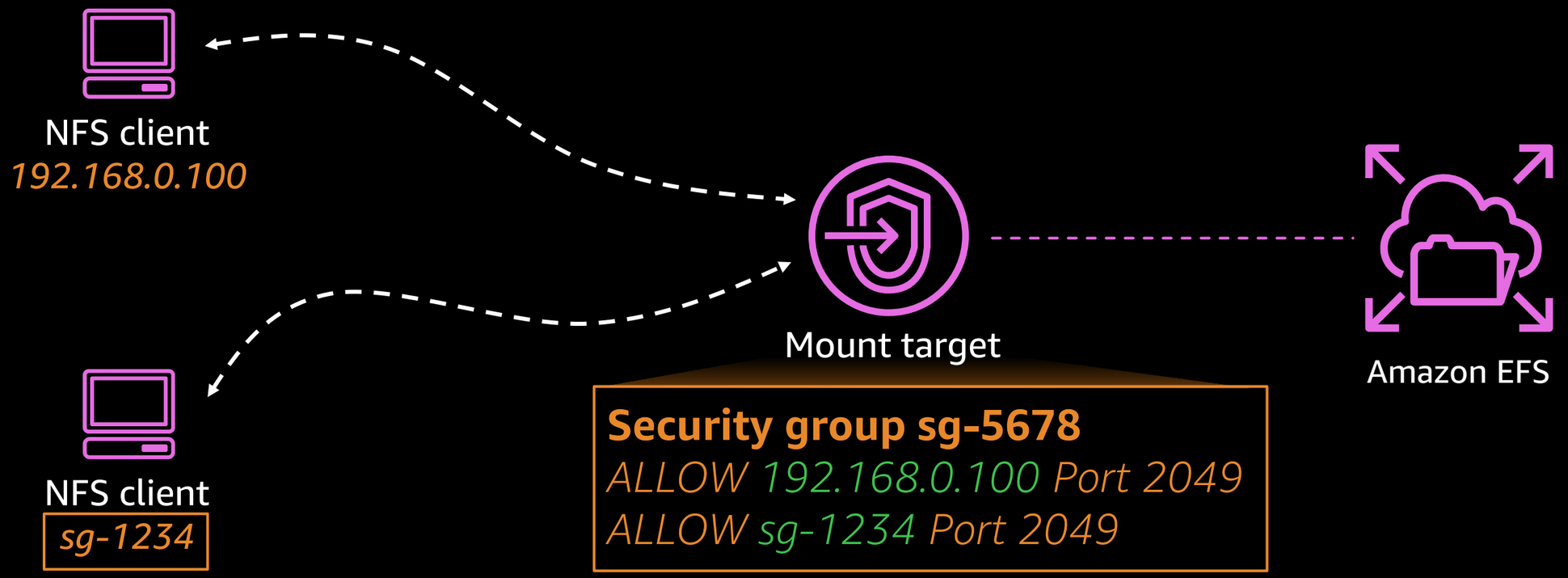
# Mount target is ENI



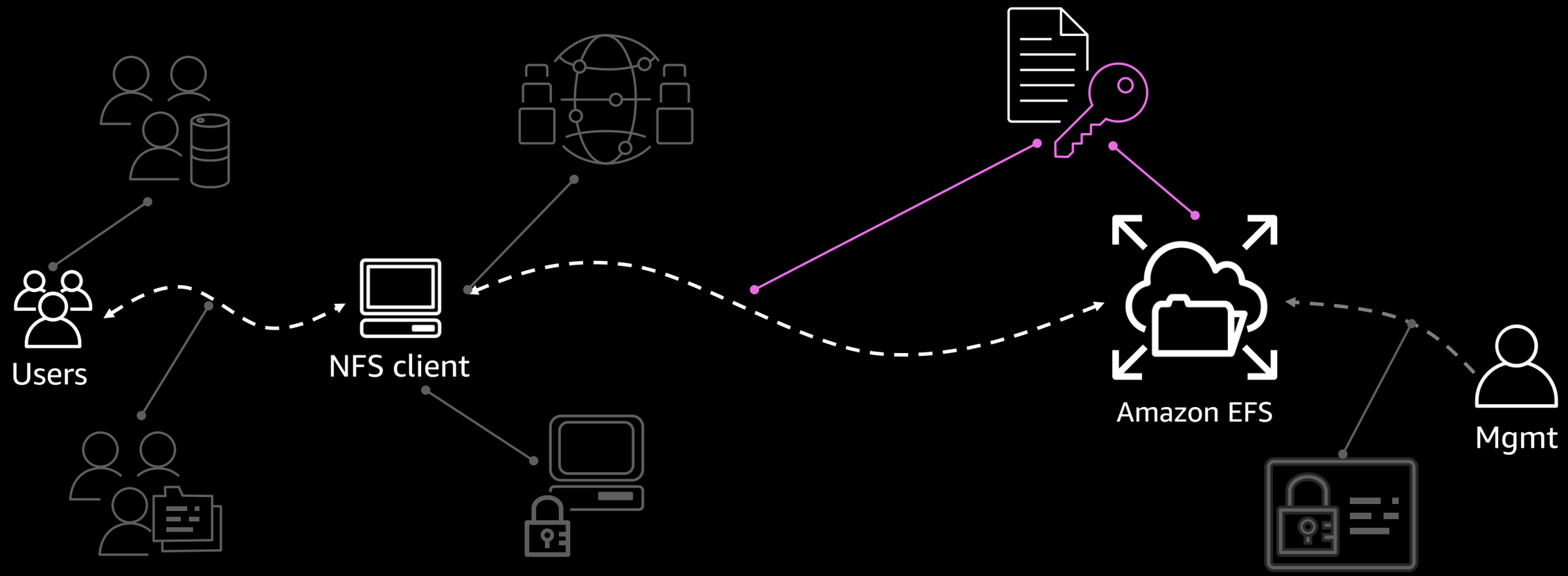



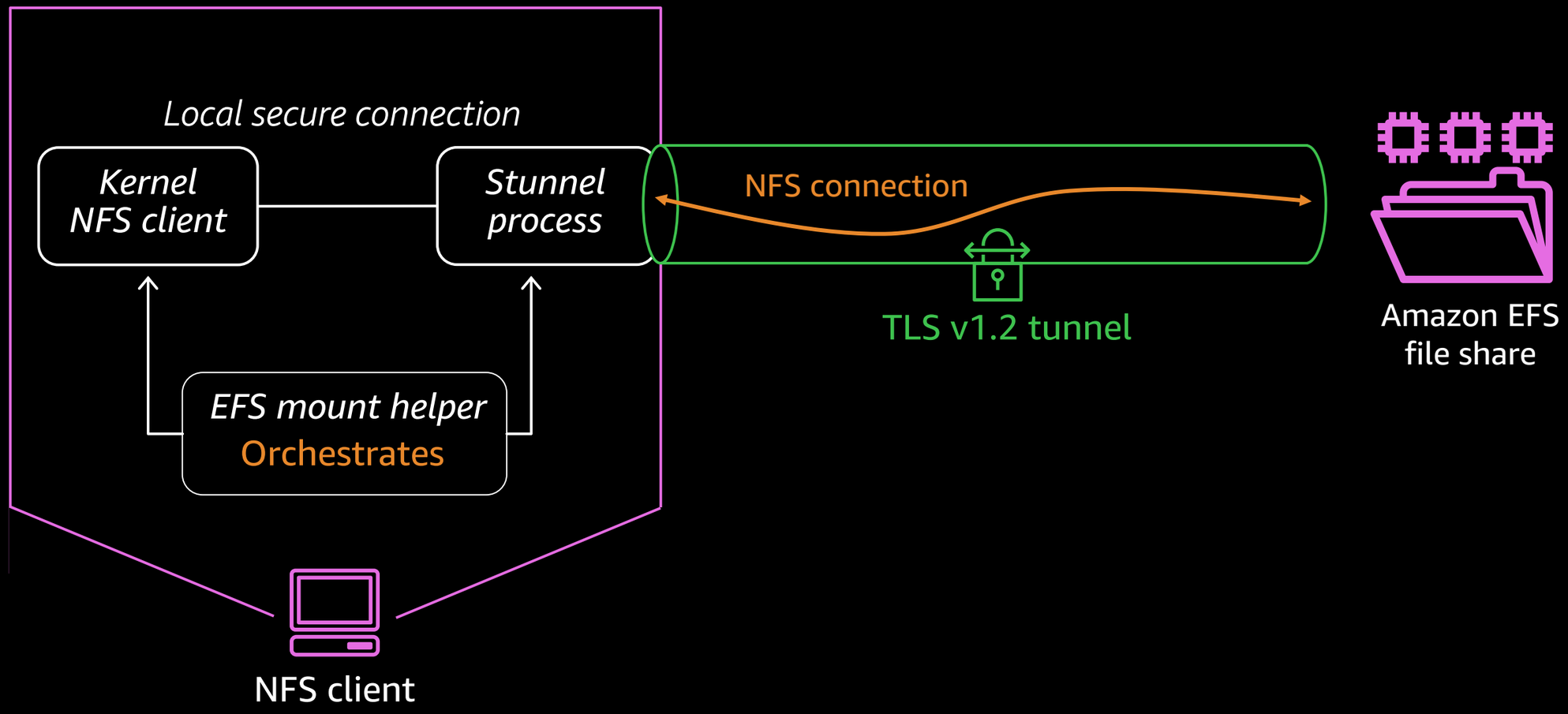
# Controlling network access with security groups



# Encrypt data

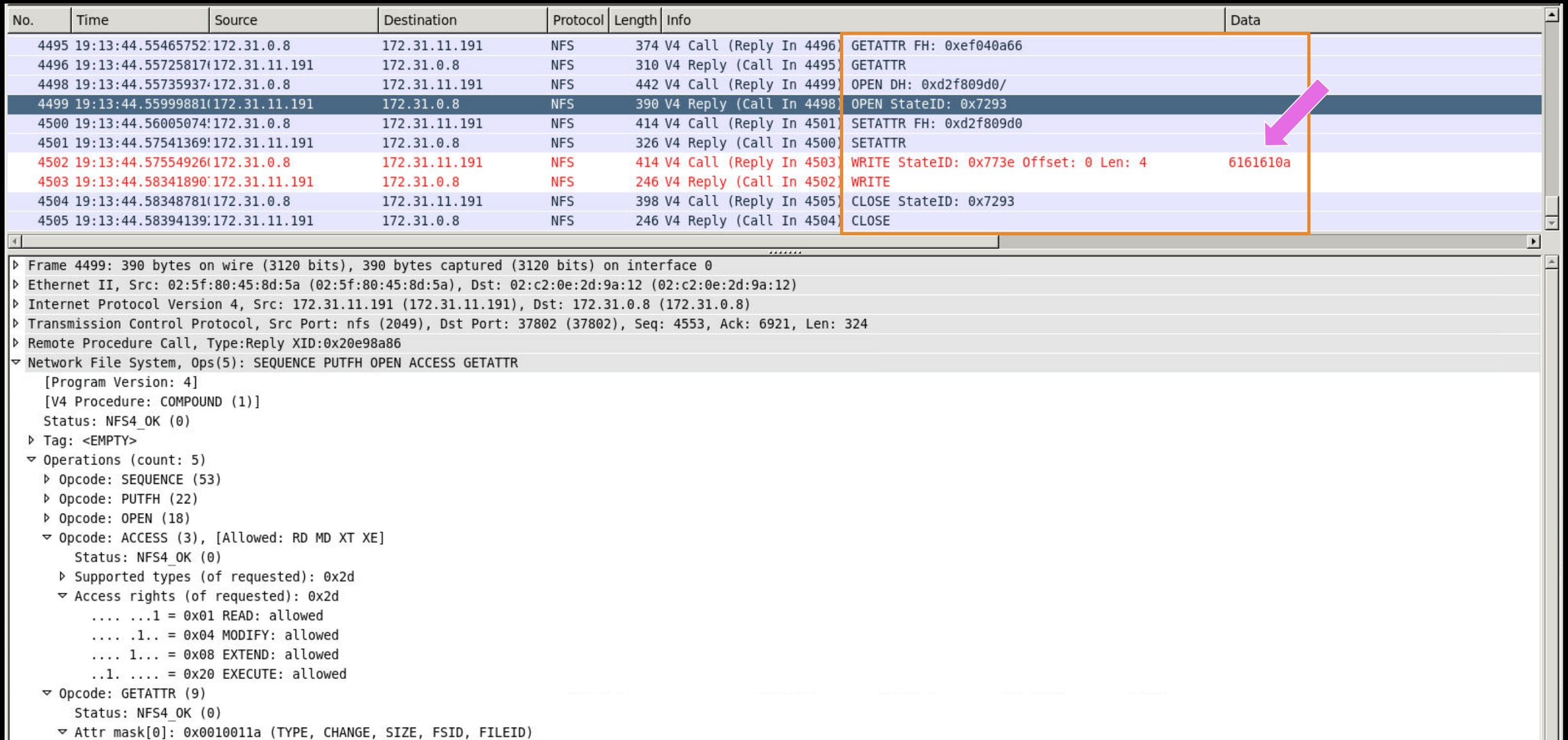


# Encryption in transit




*sudo mount -t efs -o tls fs-12345678:/ /mnt/efs*

# Example: Unencrypted write *(echo "aaa" > /efs/myfile)*



| No. | Time | Source | Destination | Protocol | Length | Info | Data |
|---|---|---|---|---|---|---|---|
| 4495 | 19:13:44.55465752 | 172.31.0.8 | 172.31.11.191 | NFS | 374 | V4 Call (Reply In 4496) | GETATTR FH: 0xef040a66 |
| 4496 | 19:13:44.55725817 | 172.31.11.191 | 172.31.0.8 | NFS | 310 | V4 Reply (Call In 4495) | GETATTR |
| 4498 | 19:13:44.55735937 | 172.31.0.8 | 172.31.11.191 | NFS | 442 | V4 Call (Reply In 4499) | OPEN DH: 0xd2f809d0/ |
| 4499 | 19:13:44.55999881 | 172.31.11.191 | 172.31.0.8 | NFS | 390 | V4 Reply (Call In 4498) | OPEN StateID: 0x7293 |
| 4500 | 19:13:44.56005074 | 172.31.0.8 | 172.31.11.191 | NFS | 414 | V4 Call (Reply In 4501) | SETATTR FH: 0xd2f809d0 |
| 4501 | 19:13:44.57541369 | 172.31.11.191 | 172.31.0.8 | NFS | 326 | V4 Reply (Call In 4500) | SETATTR |
| 4502 | 19:13:44.57554926 | 172.31.0.8 | 172.31.11.191 | NFS | 414 | V4 Call (Reply In 4503) | WRITE StateID: 0x773e Offset: 0 Len: 4 | 6161610a |
| 4503 | 19:13:44.58341890 | 172.31.11.191 | 172.31.0.8 | NFS | 246 | V4 Reply (Call In 4502) | WRITE |
| 4504 | 19:13:44.58348781 | 172.31.0.8 | 172.31.11.191 | NFS | 398 | V4 Call (Reply In 4505) | CLOSE StateID: 0x7293 |
| 4505 | 19:13:44.58394139 | 172.31.11.191 | 172.31.0.8 | NFS | 246 | V4 Reply (Call In 4504) | CLOSE |

```
▷ Frame 4499: 390 bytes on wire (3120 bits), 390 bytes captured (3120 bits) on interface 0
▷ Ethernet II, Src: 02:5f:80:45:8d:5a (02:5f:80:45:8d:5a), Dst: 02:c2:0e:2d:9a:12 (02:c2:0e:2d:9a:12)
▷ Internet Protocol Version 4, Src: 172.31.11.191 (172.31.11.191), Dst: 172.31.0.8 (172.31.0.8)
▷ Transmission Control Protocol, Src Port: nfs (2049), Dst Port: 37802 (37802), Seq: 4553, Ack: 6921, Len: 324
▷ Remote Procedure Call, Type:Reply XID:0x20e98a86
▽ Network File System, Ops(5): SEQUENCE PUTFH OPEN ACCESS GETATTR
    [Program Version: 4]
    [V4 Procedure: COMPOUND (1)]
    Status: NFS4_OK (0)
  ▷ Tag: <EMPTY>
  ▽ Operations (count: 5)
    ▷ Opcode: SEQUENCE (53)
    ▷ Opcode: PUTFH (22)
    ▷ Opcode: OPEN (18)
    ▽ Opcode: ACCESS (3), [Allowed: RD MD XT XE]
        Status: NFS4_OK (0)
      ▷ Supported types (of requested): 0x2d
      ▽ Access rights (of requested): 0x2d
          .... ...1 = 0x01 READ: allowed
          .... .1.. = 0x04 MODIFY: allowed
          .... 1... = 0x08 EXTEND: allowed
          ..1. .... = 0x20 EXECUTE: allowed
    ▽ Opcode: GETATTR (9)
        Status: NFS4_OK (0)
      ▽ Attr mask[0]: 0x0010011a (TYPE, CHANGE, SIZE, FSID, FILEID)
```

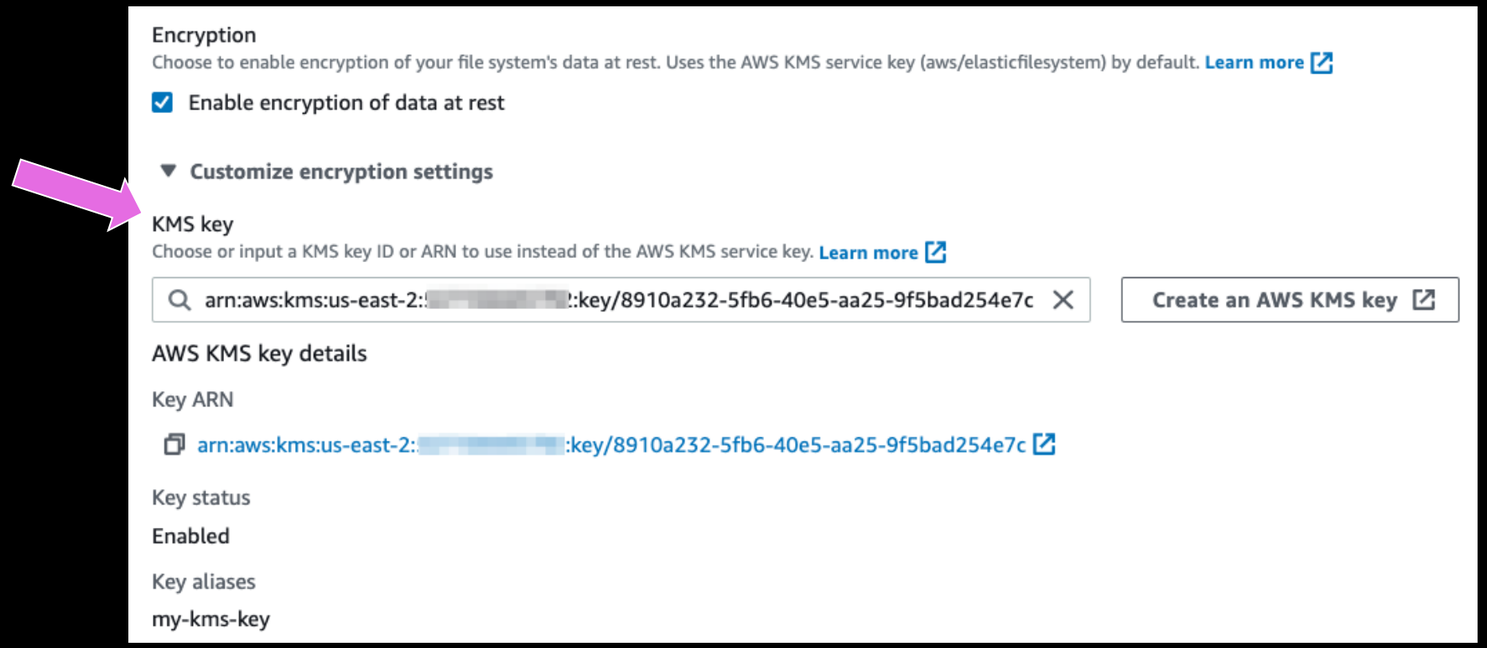
# Example: Encrypted with TLS *(echo "aaa" > /efs/myfile)*

| No. | Time | Source | Destination | Protocol | Length | Info | Data |
|---|---|---|---|---|---|---|---|
| 40631 | 19:49:58.27552522 | 172.31.0.8 | 172.31.11.191 | TCP | 403 | 38564 > nfs [PSH, ACK] Seq=11543 Ack=15427 Win=108544 Len=337 TSval=435 | |
| 40632 | 19:49:58.27604773 | 172.31.11.191 | 172.31.0.8 | TCP | 339 | nfs > 38564 [PSH, ACK] Seq=15427 Ack=11880 Win=106368 Len=273 TSval=267 | |
| 40633 | 19:49:58.27614416 | 172.31.0.8 | 172.31.11.191 | TCP | 403 | 38564 > nfs [PSH, ACK] Seq=11880 Ack=15700 Win=108544 Len=337 TSval=435 | |
| 40634 | 19:49:58.27803091 | 172.31.11.191 | 172.31.0.8 | TCP | 339 | nfs > 38564 [PSH, ACK] Seq=15700 Ack=12217 Win=106368 Len=273 TSval=267 | |
| 40635 | 19:49:58.27813319 | 172.31.0.8 | 172.31.11.191 | TCP | 403 | 38564 > nfs [PSH, ACK] Seq=12217 Ack=15973 Win=108544 Len=337 TSval=435 | |
| 40636 | 19:49:58.28010527 | 172.31.11.191 | 172.31.0.8 | TCP | 339 | nfs > 38564 [PSH, ACK] Seq=15973 Ack=12554 Win=106368 Len=273 TSval=267 | |
| 40637 | 19:49:58.28020711 | 172.31.0.8 | 172.31.11.191 | TCP | 403 | 38564 > nfs [PSH, ACK] Seq=12554 Ack=16246 Win=108544 Len=337 TSval=435 | |
| 40638 | 19:49:58.28226228 | 172.31.11.191 | 172.31.0.8 | TCP | 339 | nfs > 38564 [PSH, ACK] Seq=16246 Ack=12891 Win=106368 Len=273 TSval=267 | |
| 40639 | 19:49:58.28234676 | 172.31.0.8 | 172.31.11.191 | TCP | 403 | 38564 > nfs [PSH, ACK] Seq=12891 Ack=16519 Win=108544 Len=337 TSval=435 | |
| 40640 | 19:49:58.28428392 | 172.31.11.191 | 172.31.0.8 | TCP | 339 | nfs > 38564 [PSH, ACK] Seq=16519 Ack=13228 Win=106368 Len=273 TSval=267 | |
| 40641 | 19:49:58.28437670 | 172.31.0.8 | 172.31.11.191 | TCP | 403 | 38564 > nfs [PSH, ACK] Seq=13228 Ack=16792 Win=108544 Len=337 TSval=435 | |
| 40642 | 19:49:58.28620625 | 172.31.11.191 | 172.31.0.8 | TCP | 339 | nfs > 38564 [PSH, ACK] Seq=16792 Ack=13565 Win=106368 Len=273 TSval=267 | |
| 40643 | 19:49:58.28629265 | 172.31.0.8 | 172.31.11.191 | TCP | 427 | 38564 > nfs [PSH, ACK] Seq=13565 Ack=17065 Win=108544 Len=361 TSval=435 | |
| 40644 | 19:49:58.28835103 | 172.31.11.191 | 172.31.0.8 | TCP | 1719 | nfs > 38564 [PSH, ACK] Seq=17065 Ack=13926 Win=106368 Len=1653 TSval=26 | |
| 40645 | 19:49:58.32946370 | 172.31.0.8 | 172.31.11.191 | TCP | 66 | 38564 > nfs [ACK] Seq=13926 Ack=18718 Win=108544 Len=0 TSval=435855 TSe | |
| 40646 | 19:50:07.62205999 | 172.31.0.8 | 172.31.11.191 | TCP | 403 | 38564 > nfs [PSH, ACK] Seq=13926 Ack=18718 Win=108544 Len=337 TSval=445 | |
| 40647 | 19:50:07.62404090 | 172.31.11.191 | 172.31.0.8 | TCP | 339 | nfs > 38564 [PSH, ACK] Seq=18718 Ack=14263 Win=106368 Len=273 TSval=267 | |
| 40648 | 19:50:07.62405164 | 172.31.0.8 | 172.31.11.191 | TCP | 66 | 38564 > nfs [ACK] Seq=14263 Ack=18991 Win=108544 Len=0 TSval=445150 TSe | |
| 40649 | 19:50:07.62416039 | 172.31.0.8 | 172.31.11.191 | TCP | 403 | 38564 > nfs [PSH, ACK] Seq=14263 Ack=18991 Win=108544 Len=337 TSval=445 | |

```
▸ Frame 40603: 66 bytes on wire (528 bits), 66 bytes captured (528 bits) on interface 0
▸ Ethernet II, Src: 02:c2:0e:2d:9a:12 (02:c2:0e:2d:9a:12), Dst: 02:5f:80:45:8d:5a (02:5f:80:45:8d:5a)
▸ Internet Protocol Version 4, Src: 172.31.0.8 (172.31.0.8), Dst: 172.31.11.191 (172.31.11.191)
▾ Transmission Control Protocol, Src Port: 38564 (38564), Dst Port: nfs (2049), Seq: 7812, Ack: 11044, Len: 0
    Source port: 38564 (38564)
    Destination port: nfs (2049)
    [Stream index: 27]
    Sequence number: 7812    (relative sequence number)
    Acknowledgment number: 11044    (relative ack number)
    Header length: 32 bytes
  ▸ Flags: 0x010 (ACK)
```

# Encryption at rest

Data and metadata are automatically AES-256 encrypted before being written




As data and metadata are read, they are automatically decrypted

Because encryption is handled transparently, you don't need to modify your applications

# IAM in a nutshell

```
{
  "Statement": [
    {
      "Effect": "Allow",
      "Action": ["s3:Get*", "s3:List*"],
      "Resource": "*"
    }
  ]
}
```





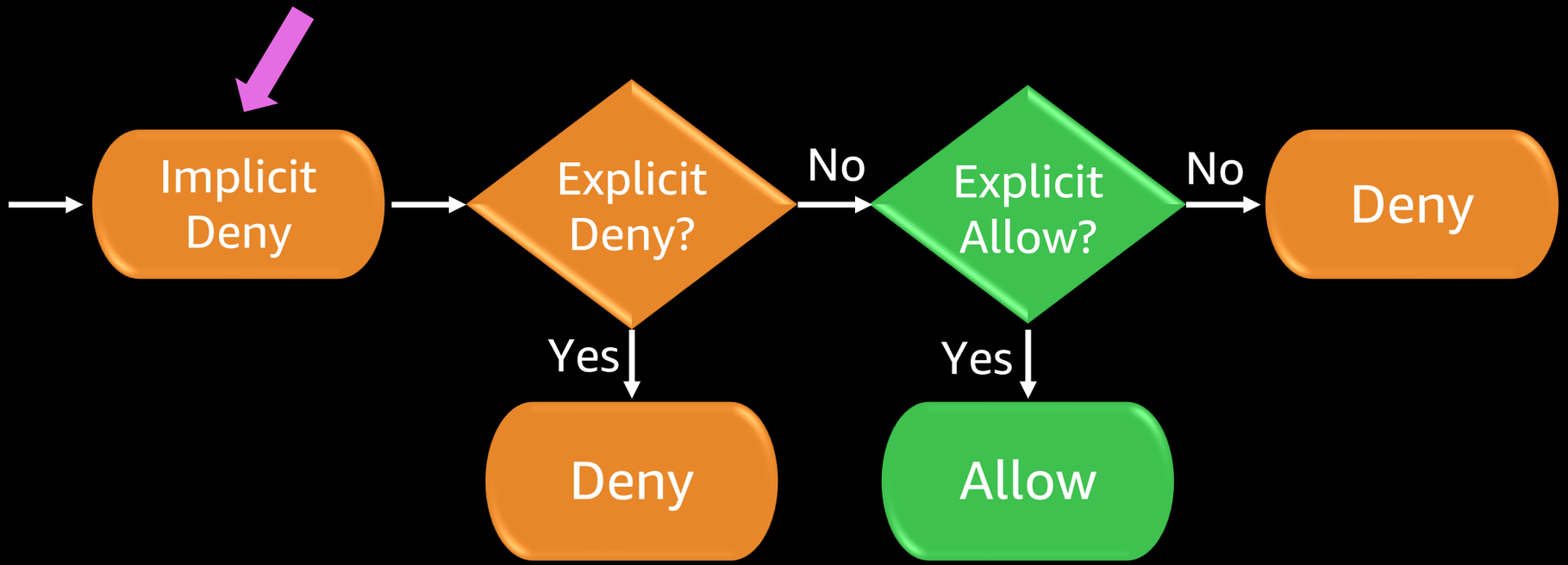
© 2020, Amazon Web Services, Inc. or its affiliates. All rights reserved.

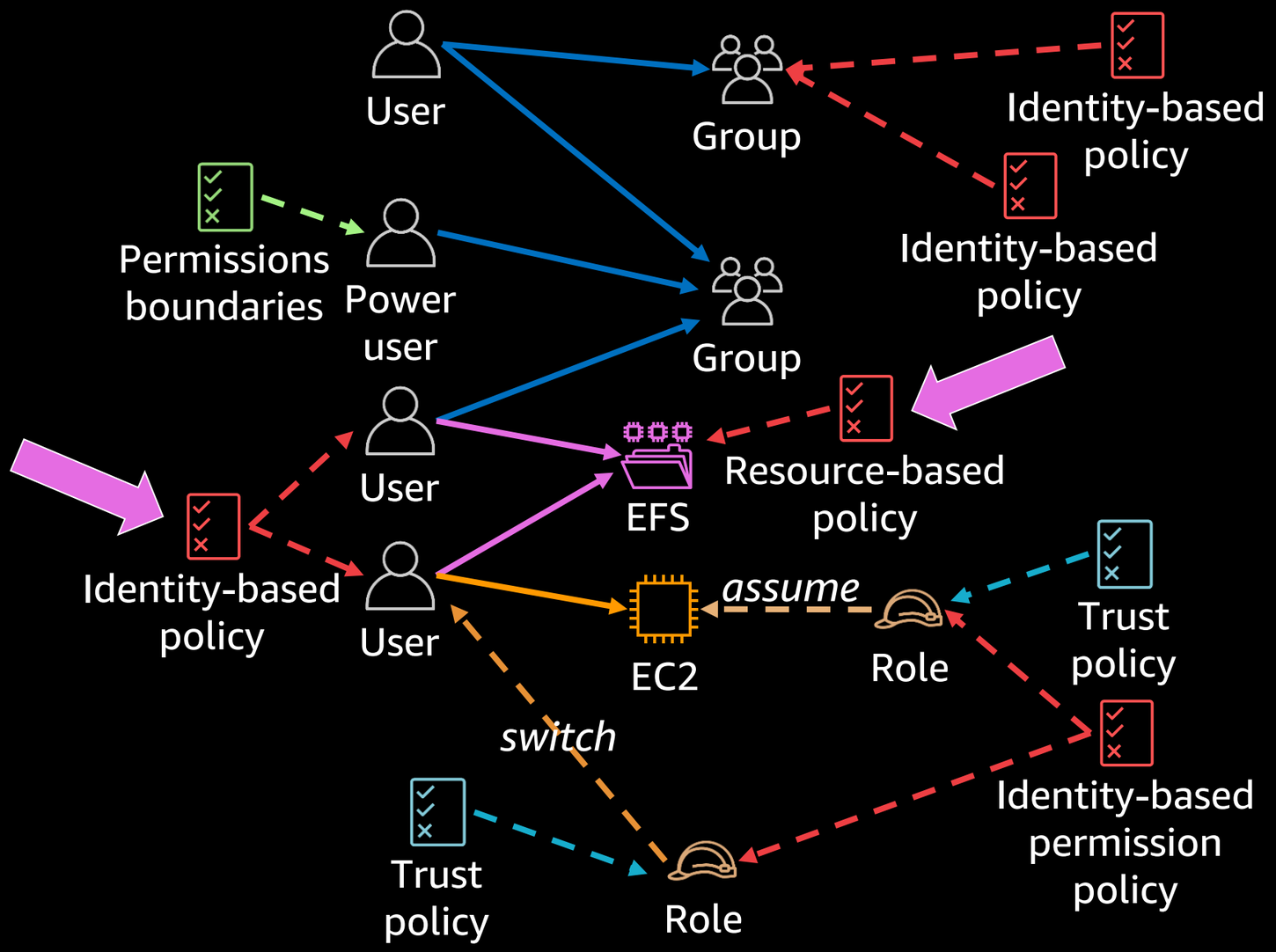
**E**ffect
**P**rincipal
**A**ction
**R**esource
**C**ondition

```
"Statement":[
   {
      "Effect":”Allow/Deny",
      "Principal":"arn",
      "Action":"elasticfilesystem:CreateTags",
      "Resource":"arn",
      "Condition":{
         "condition":{"key":"value" }
      }
   },
   {...}
]
```

# Policy evaluation logic

## By default, all requests are denied




User
Group
Identity-based policy
Permissions boundaries
Power user
Identity-based policy
Group
User
Resource-based policy
EFS
Identity-based policy
User
assume
Trust policy
EC2
Role
switch
Identity-based permission policy
Trust policy
Role

# Identity-based



Users / groups

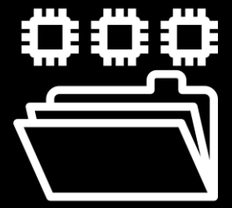


Roles

- **E**ffect
- ~~Principal~~
- **A**ction
- **R**esource
- **C**ondition



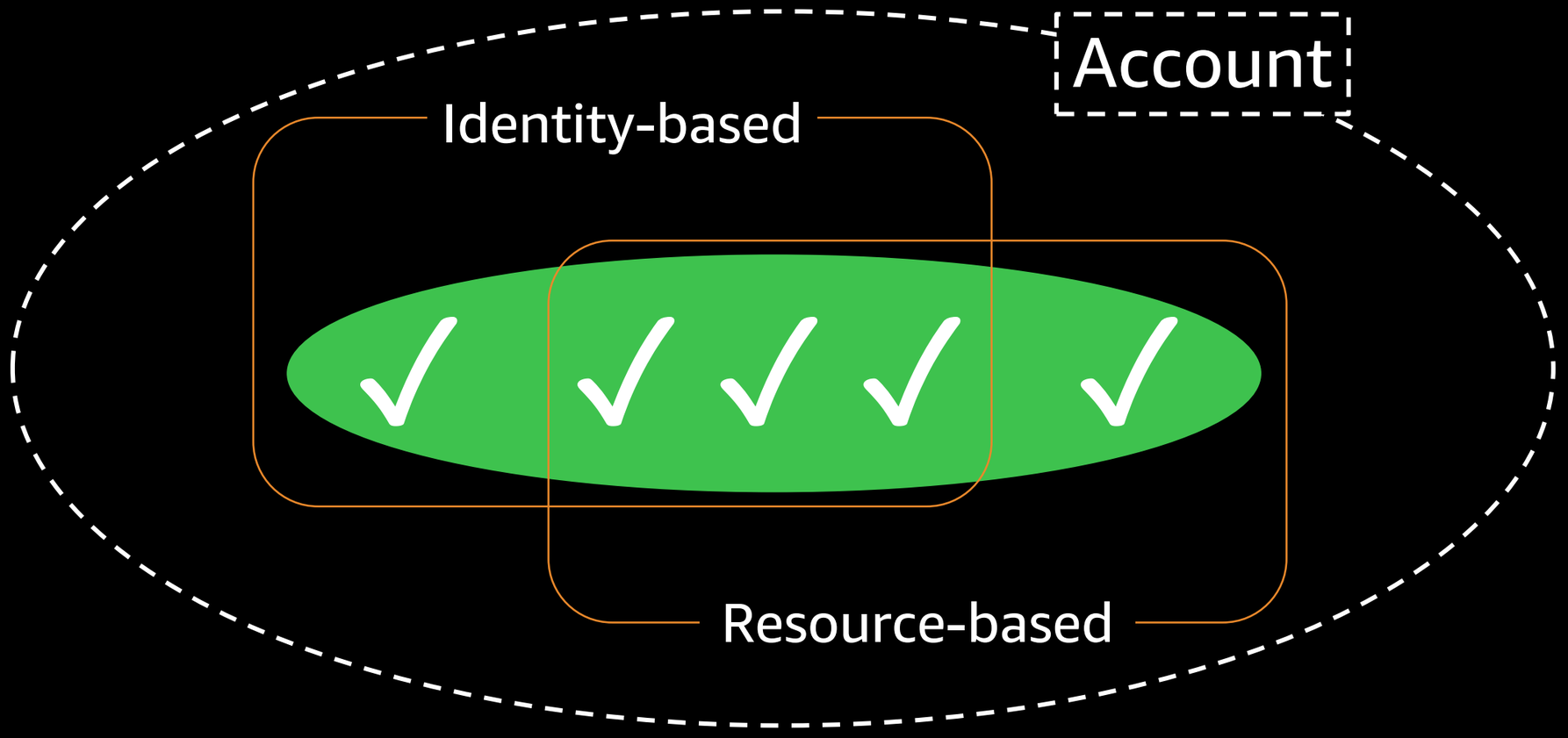
# Resource-based



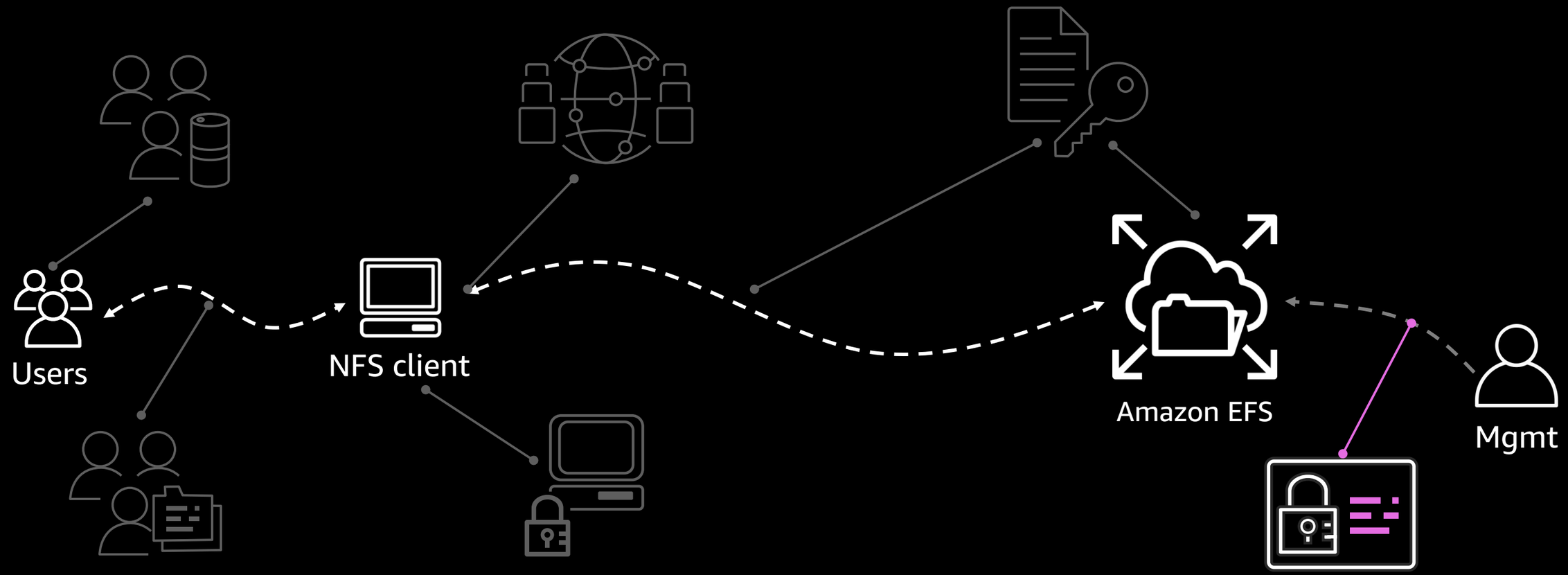
Resources

- **E**ffect
- **P**rincipal
- **A**ction
- ~~Resource~~
- **C**ondition

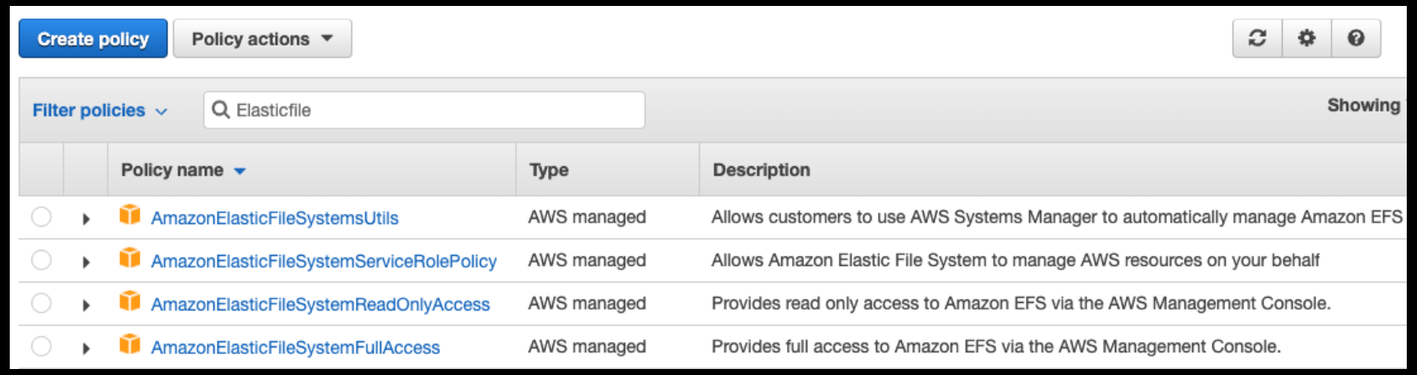
# The resulting permissions are the total permissions of the two types



# Control API access



# Predefined policies

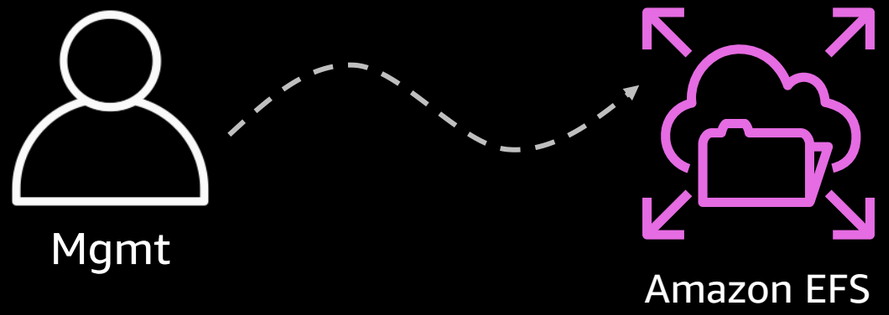



AmazonElasticFileSystemFullAccess

- CreateFileSystem
- UpdateFileSystem
- DeleteFileSystem

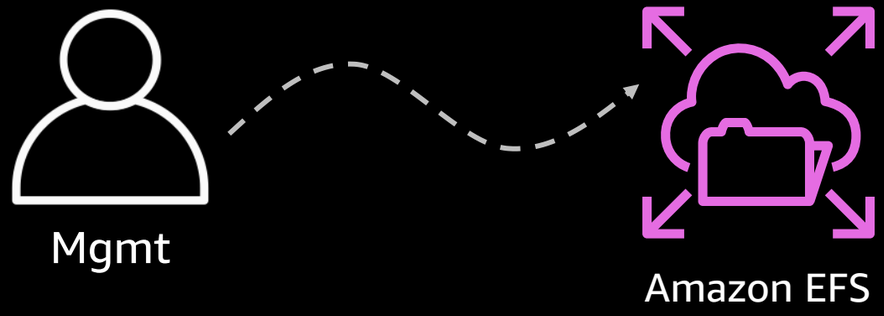
AmazonElasticFileSystemReadOnlyAccess

- DescribeFileSystems
- DescribeMountTargets
- DescribeTags

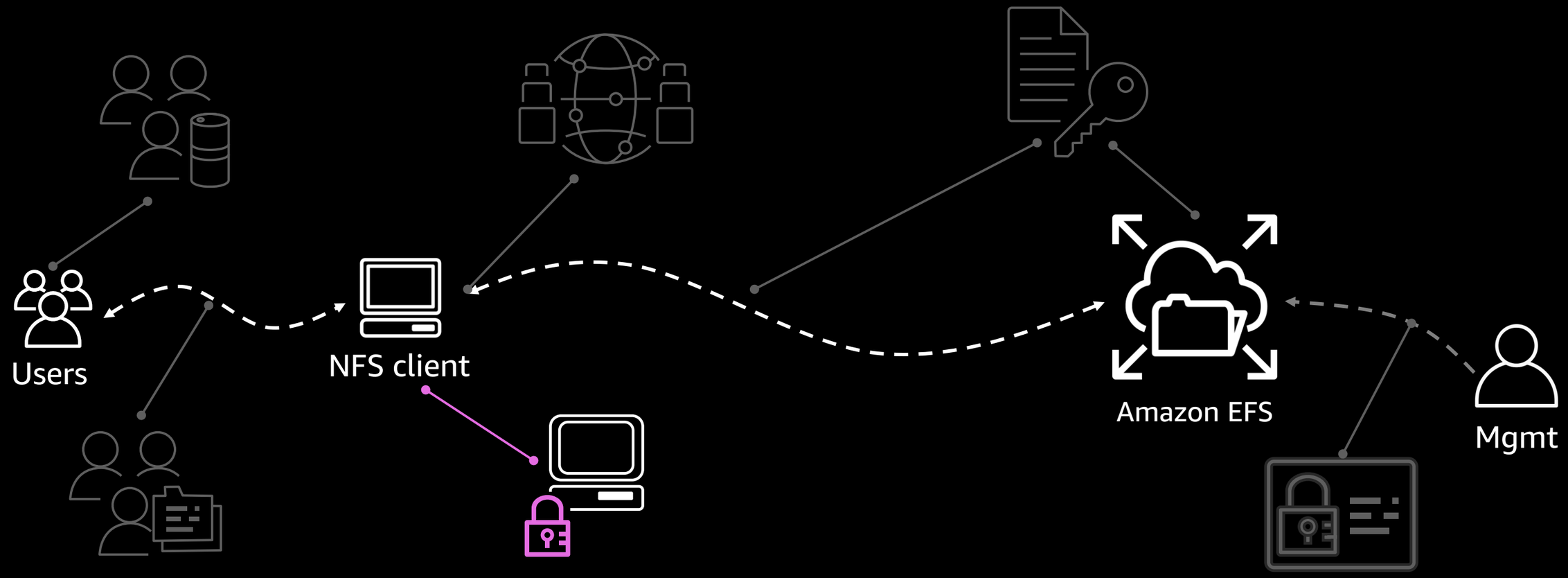


# Customer managed policies

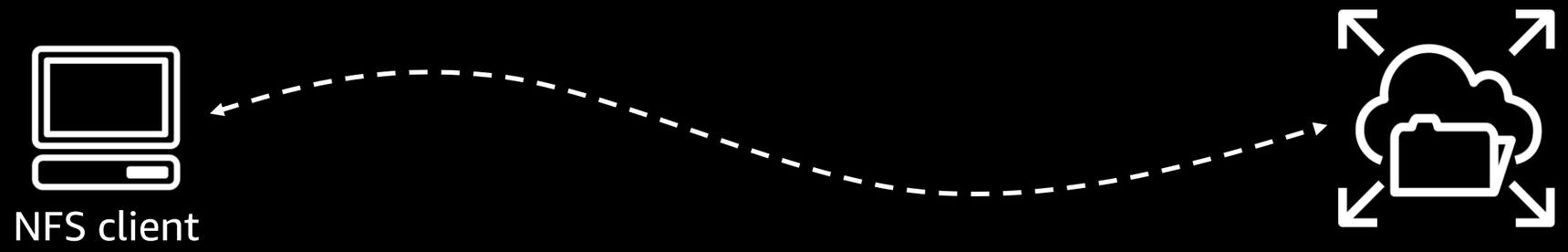
```
"Statement": [
 {
    "Effect": "Allow",
    "Action": [
        "elasticfilesystem:CreateFileSystem",
        "elasticfilesystem:CreateMountTarget",
    ],
    "Resource": "arn:aws:elasticfilesystem:us-west-2:12345:file-system/file-system-ID"
 },
 {
    "Effect": "Allow",
    "Action": [
        "ec2:DescribeSubnets",
        "ec2:CreateNetworkInterface",
        "ec2:DescribeNetworkInterfaces"
    ],
    "Resource": "*"
 }
]
```



# Control client access



# Restricting access with resource-based policy




File system resource policy

```
"Statement" : {
   "Effect" : "allow",
   "Principal" : "*",
   "Action" : [
      "elasticfilesystem:ClientMount",
      "elasticfilesystem:ClientWrite"
   ]
}
```

# File system policy options



NFS client



Amazon EFS > File systems > fs-8e5ed6f6 > policy

## File system policy

### Policy options

Select one or more of these common policy options, or create a more advanced policy using the editor. Learn more

- [x] Prevent root access by default*
- [x] Enforce read-only access by default*
- [x] Enforce in-transit encryption for all clients

* Identity-based policies can override these default permissions.

### Policy editor

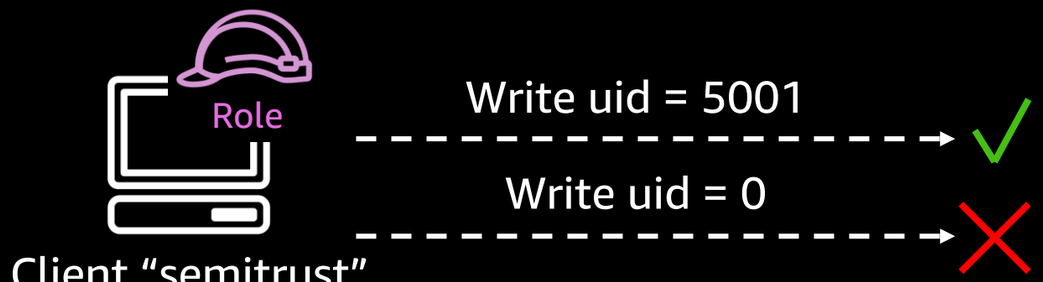
Clear

```
{
    "Version": "2012-10-17",
    "Id": "efs-policy-wizard-d968ad5c-2cad-4015-8442-70b037b633df",
    "Statement": [
        {
            "Sid": "efs-statement-5343b047-5523-4c2f-a026-f55abc82cfa8",
            "Effect": "Allow",
            "Principal": {
                "AWS": "*"
            },
            "Action": [
                "elasticfilesystem:ClientMount"
            ]
        },
        {
            "Sid": "efs-statement-bab56f82-ceb5-4d31-84db-4a75e5288ded",
            "Effect": "Deny",
            "Principal": {
                "AWS": "*"
            },
            "Action": "*",
            "Condition": {
                "Bool": {
                    "aws:SecureTransport": "false"
                }
            }
        }
    ]
}
```

# Anonymous access



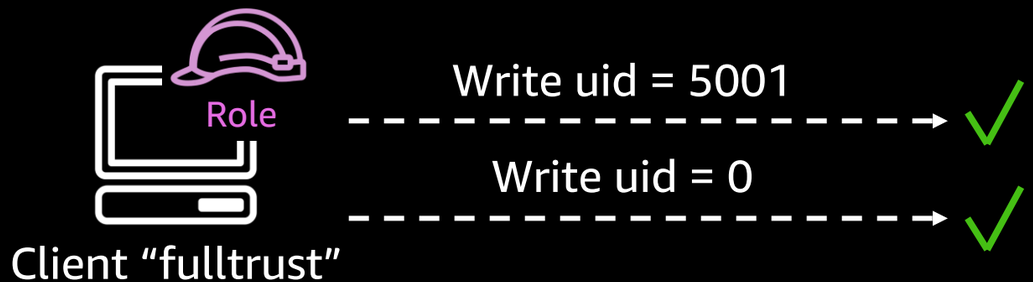
# Grant permissions to an individual IAM role







```
“Effect” : “allow”,
“Action” : [ “elasticfilesystem:ClientMount”,
             “elasticfilesystem:ClientWrite” ],
“Principal” : { “AWS”: “semitrust” }
```



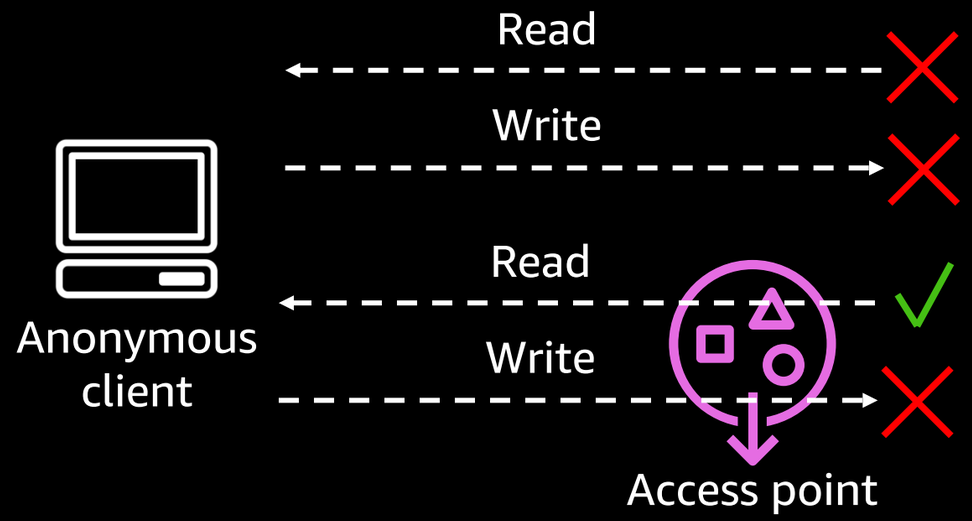




```
“Effect” : “allow”,
“Action” : [“elasticfilesystem:ClientMount”,
             “elasticfilesystem:ClientWrite”,
             “elasticfilesystem:ClientRootAccess ],
“Principal” : { “AWS”: “fulltrust” }
```

** Identity-based policies can override these default permissions*

# Restrict access with a specific access point

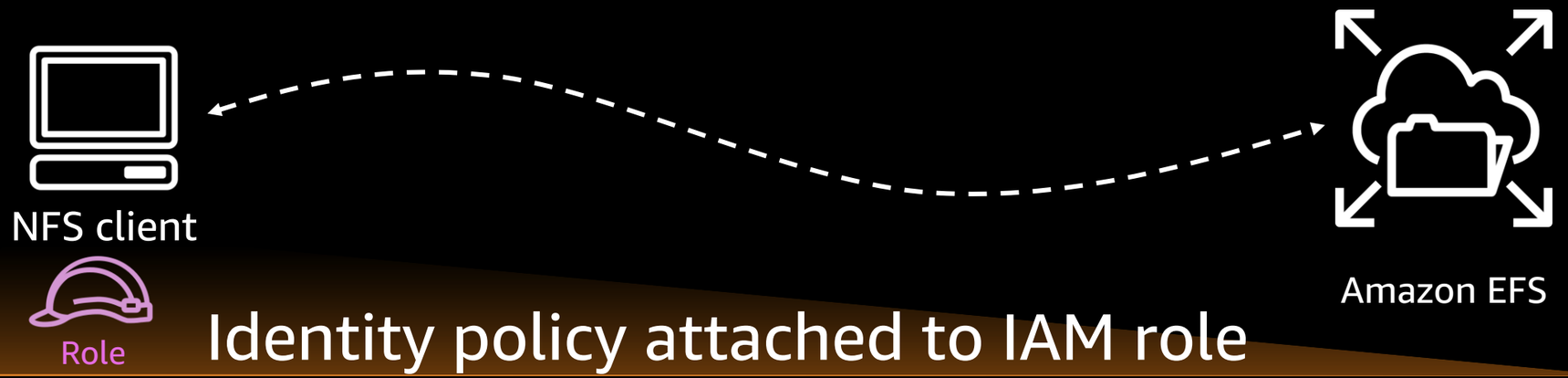






```
“Effect” : “allow”,
“Action” : “elasticfilesystem:ClientMount”,
“Principal” : “*”
“Condition” : {
   "StringEquals": {
      "elasticfilesystem :AccessPointArn" :
            "arn:aws:elasticfilesystem:us-east-2:
            1234:access-point/fsap-0204ce67"
}
```

# Restricting access with identity-based policy



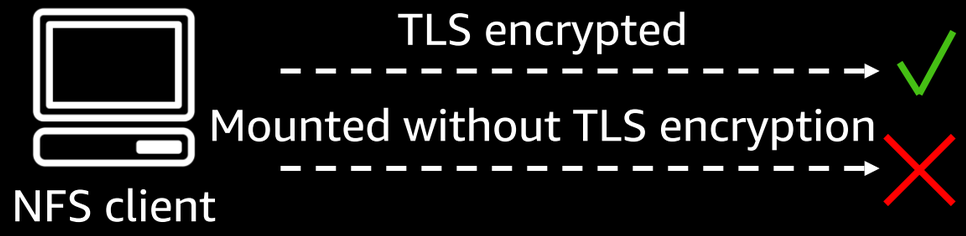

Identity policy attached to IAM role

```
“Statement” : {
   “Effect” : “allow”,
   “Action” : “elasticfilesystem:Client*”,
   "Resource": ” arn:aws:elasticfilesystem:us-east-2:123:file-system/fs-d1188b58"
}
```

```
sudo mount -t efs -o iam,tls fs-d1188b58 /efs
```

# Reinforce encryption

```
sudo mount -t efs -o tls fs-d1188b58 /efs
```







```
“Effect” : “deny”,
“Action” : “elasticfilesystem:Client*”,
“Principal” : “*”,
“Condition” : {
    “Bool”: {“aws:SecureTransport”, “true”}
}
```

```
"Effect": "Allow",
"Action": "elasticfilesystem:CreateFileSystem",
"Resource": "*"
"Condition": {
    "Bool": {
        "elasticfilesystem:Encrypted": "true"
    }
}
```









# Compliance



AWS re:Invent

© 2020, Amazon Web Services, Inc. or its affiliates. All rights reserved.

# Compliance



HIPAA | GDPR | PCI-DSS | SOC | ISO | IRAP | FedRAMP | C5

✓ ✓ ✓ ✓ ✓ ✓ ✓ ✓

OSPAR | ENS High | DoD CC SRG | HITRUST CSF | MTCS | K-ISMS

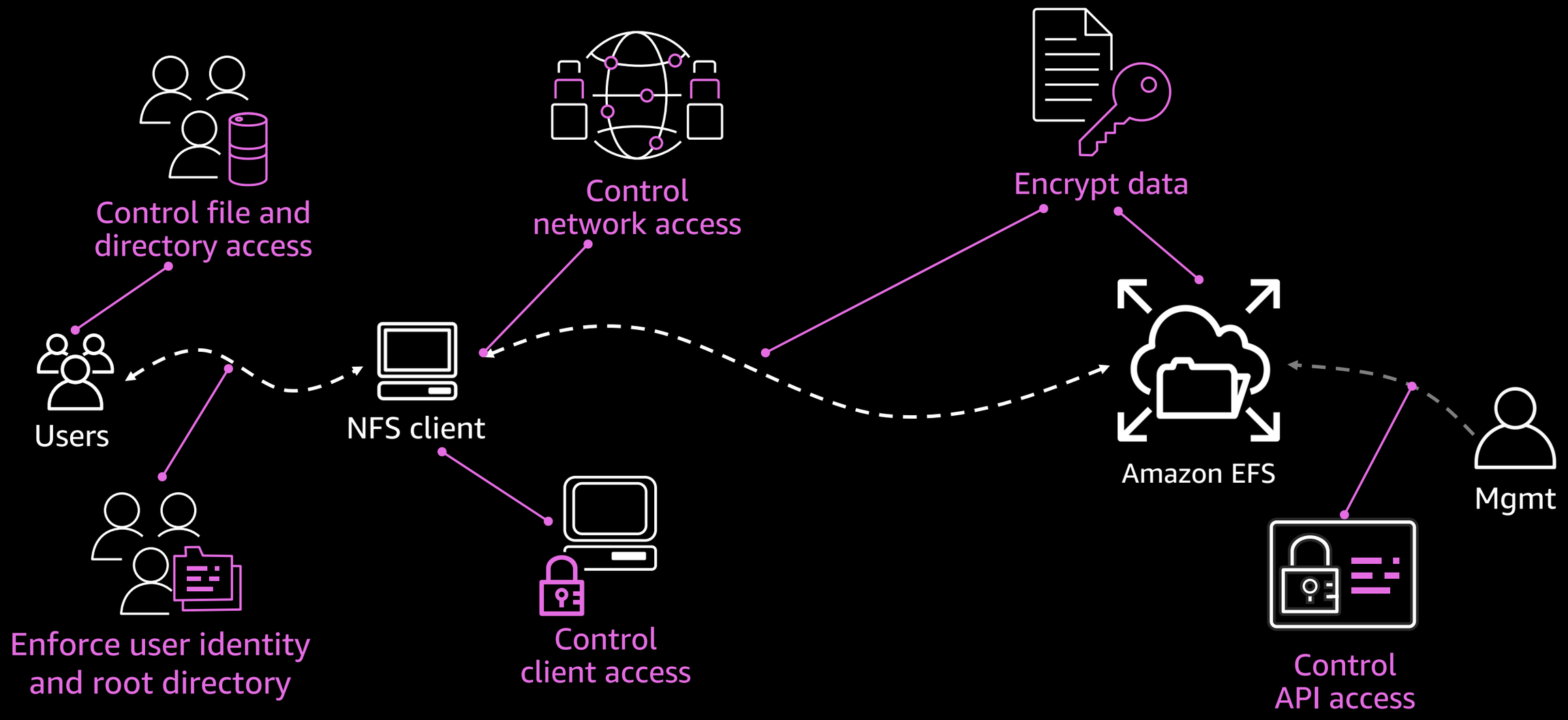
✓ ✓ ✓ ✓ ✓ ✓

# Summary





© 2020, Amazon Web Services, Inc. or its affiliates. All rights reserved.


Control file and directory access
Control network access
Encrypt data
Users
NFS client
Amazon EFS
Mgmt
Enforce user identity and root directory
Control client access
Control API access

# Key takeaways

- Use access points, even if single app per file system
- Always control network access with security groups
- Always enable encryption in transit end encryption at rest
- Use IAM restricting access to in-cloud resources



# Amazon EFS sessions

**STG213** What's new with Amazon EFS

**STG311** Optimizing Amazon EFS for cost and performance, using best practices

**STG307** Deep dive on Amazon EFS

**STG310** Modernize your applications with AWS Lambda and Amazon EFS

**STG312** Modernize your applications with containers using Amazon EFS

**STG306** Security best practices with Amazon EFS

**STG313** Reduce TCO for your Linux file-based applications

**STG314** Amazon EFS performance best practices for Linux workloads

# Thank you!



AWS re:Invent

© 2020, Amazon Web Services, Inc. or its affiliates. All rights reserved.



# Please complete the session survey

AWS re:Invent

© 2020, Amazon Web Services, Inc. or its affiliates. All rights reserved.