- **How to add new users**
    - **sudo useradd username**

- **How to set a password for this user**
    - **sudo passwd username**

- **How to add user to group**
    - **sudo usermod -aG groupname username**

- **How to remove user from group**
    - **sudo gpasswd -d username groupname**

- **How to create a new group**
    - **sudo groupadd <group name>**

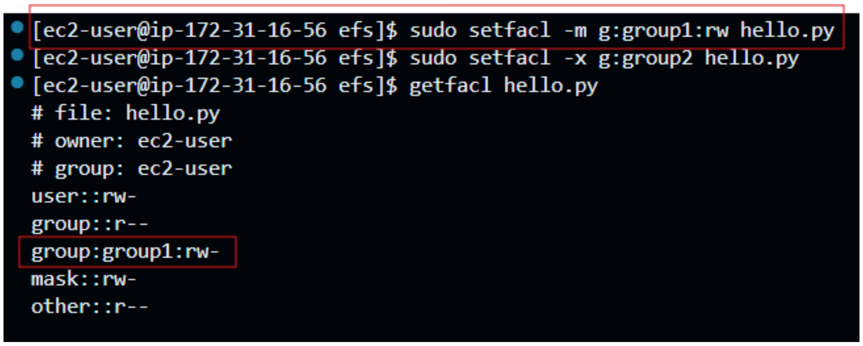
- **How to check members in the group.**
    - **getent group <group name>**

- **How to check the users in the group**
    - **getent group <group name>**

- **Assigning permissions to groups**
    - **Changes the group ownership of a file or directory.**
        - **sudo chgrp groupname /efs/directory**

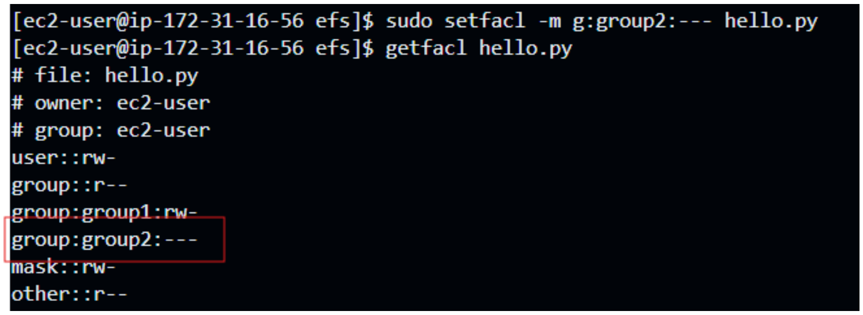
      or

    - **Change the Access Control Lists (ACLs)**
        - **Setting ACL for group1**
            - **sudo setfacl -m g:group1:rw hello.py**

              ```
              [ec2-user@ip-172-31-16-56 efs]$ sudo setfacl -m g:group1:rw hello.py
              [ec2-user@ip-172-31-16-56 efs]$ sudo setfacl -x g:group2 hello.py
              [ec2-user@ip-172-31-16-56 efs]$ getfacl hello.py
              # file: hello.py
              # owner: ec2-user
              # group: ec2-user
              user::rw-
              group::r--
              group:group1:rw-
              mask::rw-
              other::r--
              ```

            -
        - **Remove permissions all for group2**
            - **sudo setfacl -m g:group2:--- /efs/hello.py**

- 

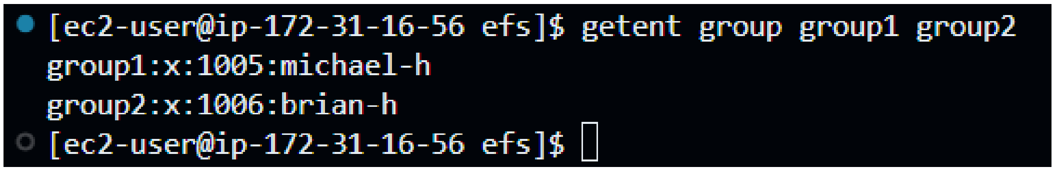
```
[ec2-user@ip-172-31-16-56 efs]$ sudo setfacl -m g:group2:--- hello.py
[ec2-user@ip-172-31-16-56 efs]$ getfacl hello.py
# file: hello.py
# owner: ec2-user
# group: ec2-user
user::rw-
group::r--
group:group1:rw-
group:group2:---
mask::rw-
other::r--
```

- **Establish two groups with different permissions.**

- **Group 1 with full permission**
  - **User in group 1 ————————> Input: cat hello.py ————————> Output: Hello, World!**

- **Group 2 without reading permission**
  - **User in group 2 ————————> Input: cat hello.py ————————> Output: Permission denied**

- **Example**



```
[ec2-user@ip-172-31-16-56 efs]$ getent group group1 group2
group1:x:1005:michael-h
group2:x:1006:brian-h
[ec2-user@ip-172-31-16-56 efs]$ 
```

- **Upload sensitive test data into EFS ✅**
  - **Example password.csv:**
    - **username,password**
    - **user1,pass1234**
    - **user2,abc$5678**
    - **user3,xyz@9012**
    - **user4,qwerty!234**
    - **user5,secure@567**

- **AWS Comprehend**

Amazon Comprehend provides APIs for detecting sensitive data such as Personally Identifiable Information. To use it EFS, you can mount the EFS file system on your EC2 instances or containerized applications, allowing them to process files stored on EFS and analyze them with Comprehend.

- **Amazon Macie (S3) ❌**

Amazon Macie can discover, classify, and protect sensitive data in your AWS environment. It can scan S3 buckets for sensitive data. For data stored in EFS, you would need to transfer the data to S3 or use a custom integration.

- **Amazon Textract**

Amazon Textract can extract text and data from scanned documents. It can be used in conjunction with EFS by first reading the documents stored in EFS, processing them with Textract, and then analyzing the extracted text.

- **Google Cloud Natural Language API**

Google Cloud Natural Language API can analyze text for sentiment, entities, and syntax. While it doesn't natively support EFS, you can read data from EFS and send it to the API for analysis.

- **Azure Text Analytics**

Azure Text Analytics provides a suite of tools for text analysis, including entity recognition, sentiment analysis, and language detection. Similar to Google Cloud, you can integrate it with EFS by reading the data and sending it to Azure for processing.

- **OpenAI GPT Models**

OpenAI's GPT-3 and GPT-4 are powerful language models that can be used for a wide range of NLP tasks, including text generation, summarization, and entity recognition. You can use these models via the OpenAI API.