User Data and Metadata

User Data and Metadata

In AWS, **User Data** and **Metadata** are two important concepts used for configuring and managing EC2 instances.

1. User Data (Instance Bootstrapping)

- **User Data** is a script that runs automatically when an EC2 instance is launched.
- It is used for **bootstrapping** (initial configuration), such as:
  - Installing software
  - Configuring settings
  - Downloading files
  - Running startup scripts

Example: User Data for Installing Apache

When launching an EC2 instance, you can provide the following script in the **User Data** section:

Amazon Linux / RHEL

```
#!/bin/bash
sudo yum update -y
sudo yum install httpd -y
sudo systemctl start httpd
sudo systemctl enable httpd
sudo chown ec2-user:ec2-user /var/www/html
echo "Hello from Apache Web Server!" > /var/www/html/index.html
sudo yum install java-1.8.0-amazon-corretto-devel
```

Ubuntu/Debian:

```
#!/bin/bash
sudo apt update -y
sudo apt install apache2 -y
sudo systemctl start apache2
sudo systemctl enable apache2
echo "Hello from Apache Web Server!" > /var/www/html/index.html
```

How to Add User Data?

- Go to **AWS Console → EC2 → Launch Instance**
- Under **Advanced details**, find the **User Data** section
- Paste the script and launch the instance

How to View User Data of a Running Instance?

```
curl http://169.254.169.254/latest/user-data
```

1. Metadata (Instance Information)

- **Metadata** provides information about the running EC2 instance.
- It can be accessed from **inside the instance** via the **metadata service**.
- The metadata URL:
- http://169.254.169.254/latest/meta-data/
- No authentication is needed since it is available only inside the instance.

Example: Get EC2 Metadata from terminal after connecting to Instance via SSH and run below code

```
TOKEN=$(curl -X PUT "http://169.254.169.254/latest/api/token" -H "X-aws-ec2-metadata-token-ttl-seconds: 21600" -s)
curl -H "X-aws-ec2-metadata-token: $TOKEN" -s http://169.254.169.254/latest/meta-data/
```

This will return metadata categories like:

```
ami-id
instance-id
public-ipv4
local-ipv4
hostname
security-groups
```

Fetching Specific Metadata

Instance ID

```
curl -H "X-aws-ec2-metadata-token: $TOKEN" -s http://169.254.169.254/latest/meta-data/instance-id
```

```
TOKEN=$(curl -X PUT "http://169.254.169.254/latest/api/token" -H "X-aws-ec2-metadata-token-ttl-seconds: 21600" -s)
curl -H "X-aws-ec2-metadata-token: $TOKEN" -s http://169.254.169.254/latest/meta-data/
```

This code is used to securely fetch metadata about an AWS EC2 instance by utilizing **IMDSv2**

Step 1: Request a Token:

```
TOKEN=$(curl -X PUT "http://169.254.169.254/latest/api/token" \
 -H "X-aws-ec2-metadata-token-ttl-seconds: 21600" -s)
```

• curl -X PUT "http://169.254.169.254/latest/api/token" → Sends a PUT request to obtain an IMDSv2 token.
• -H "X-aws-ec2-metadata-token-ttl-seconds: 21600" → Sets the token's time-to-live (TTL) to **21,600 seconds (6 hours)**.
• -s → Runs the command **silently**, suppressing progress output.
• TOKEN=$(...) → Stores the obtained token in a variable named TOKEN.

Step 2: Use the Token to Fetch Metadata:

```
curl -H "X-aws-ec2-metadata-token: $TOKEN" -s http://169.254.169.254/latest/meta-data/
```

• -H "X-aws-ec2-metadata-token: $TOKEN" → Sends the retrieved token as a **header** for authentication.
• http://169.254.169.254/latest/meta-data/ → Queries the metadata service for instance-related information.
• -s → Runs silently.

Code Output:

1. **Retrieves an IMDSv2 token** (to ensure secure access to instance metadata).
2. **Uses the token** to fetch instance metadata securely.
3. **Outputs metadata information**, such as:
   - ami-id → AMI ID of the instance
   - instance-id → Unique instance ID
   - instance-type → Type of EC2 instance
   - public-ipv4 → Public IP address (if assigned)
   - local-ipv4 → Private IP address