# Live AWS Resource manager Project using SHELL SCRIPTING for DevOps

**(https://www.youtube.com/watch?v=gx5E47R9fGk&list=PLdpzxOOAlwvIKMhk8WhzN1pYoJ1YU8Csa&index=13)**

Create a Linux image in the EC2 instance and connect to it.

pwd

/home/ec2-user

Vim aws_resource_tracker.sh
Click i to write then paste the script and type esc , :wq

Give the execute permission to the script
chmod 777 aws_resource_tracker.sh

For a Bash script used in a cron job, you should ensure the following permissions:

**Permissions for Bash Script:**

- **Read and execute** permissions for the user running the cron job.
- Typically, you'll want to use `chmod 700` to allow only the script owner to read, write, and execute the script.

bash
```
chmod 700 /path/to/your-script.sh
```

To schedule a cron job, you can edit the crontab using:

bash
```
crontab -e
```

**`Add the Cron Job`:**

- `Add the following line to schedule the script to run`
  bash

```
Copy code
* * * * * /home/ec2-user/aws-resource-tracker.sh >>
/home/ec2-user/log/aws-resource-ouput.txt 2>&1
```

Here's a breakdown of the cron job line:

### 🕒 `* * * * *`

- This defines the schedule for the cron job:
  - **First `*` (minute)**: Runs every minute.
  - **Second `*` (hour)**: Every hour.
  - **Third `*` (day of the month)**: Every day of the month.
  - **Fourth `*` (month)**: Every month.
  - **Fifth `*` (day of the week)**: Every day of the week.

So, this job will run **every minute**.

### 📜 `/home/ec2-user/aws-resource-tracker.sh`

- This is the path to the script that will be executed (`aws-resource-tracker.sh`).

### 📂 `>> /home/ec2-user/log/aws_resource-output.txt`

- The `>>` operator appends the output (stdout) of the script to the file `/home/ec2-user/log/aws_resource-output.txt`. If the file doesn't exist, it will be created.

### ⚠️ `2>&1`

- This redirects **stderr (2)** to the same file as **stdout (1)**. So, both standard output and error messages from the script will be logged into the same file.

### 🔄 Full Explanation

Every minute, the cron job runs the script `/home/ec2-user/aws-resource-tracker.sh`. Both the output and error messages from the script are logged to `/home/ec2-user/log/aws_resource-output.txt`.

Also, you have to create an IAM role that gives this permission to the EC2 instance.

| Policy name | Type | Attached entities |
| --- | --- | --- |
| AmazonEC2ReadOnlyAccess | AWS managed | 1 |
| AmazonRDSReadOnlyAccess | AWS managed | 1 |
| AmazonS3ReadOnlyAccess | AWS managed | 1 |
| IAMReadOnlyAccess | | |

**Attach the IAM Role to the EC2 Instance:**

- Go to the EC2 Management Console.
- Select the instance where you want to run the script.
- Click **Actions** → **Security** → **Modify IAM role**.
- Select the IAM role you created (`EC2-AWS-Resource-Tracker-Role`) from the dropdown and click **Update IAM role**.