

EC2 Auto-Start and Stop Schedule using IAM & Bash

Project Goal (Real-Time Purpose)

Goal:

To automatically start and stop EC2 instances (virtual servers on AWS) at specific times of the day to save costs and improve efficiency — without needing to manually log in every time.

This is very useful for real-world organizations that use EC2 servers only during working hours. For example:

- A company might only need their development or test servers from 9 AM to 6 PM.
- Keeping them running 24/7 wastes AWS billing hours.
- So, using a script + cron job, the instances can automatically:
 - Start at 9 AM (when employees start work)
 - Stop at 6 PM (when work ends)

This automation reduces manual work and cloud costs significantly.

Final Result:

Your EC2 instance will start at a specific time (like 9 AM) and stop automatically (like 6 PM) every day — no manual action needed!

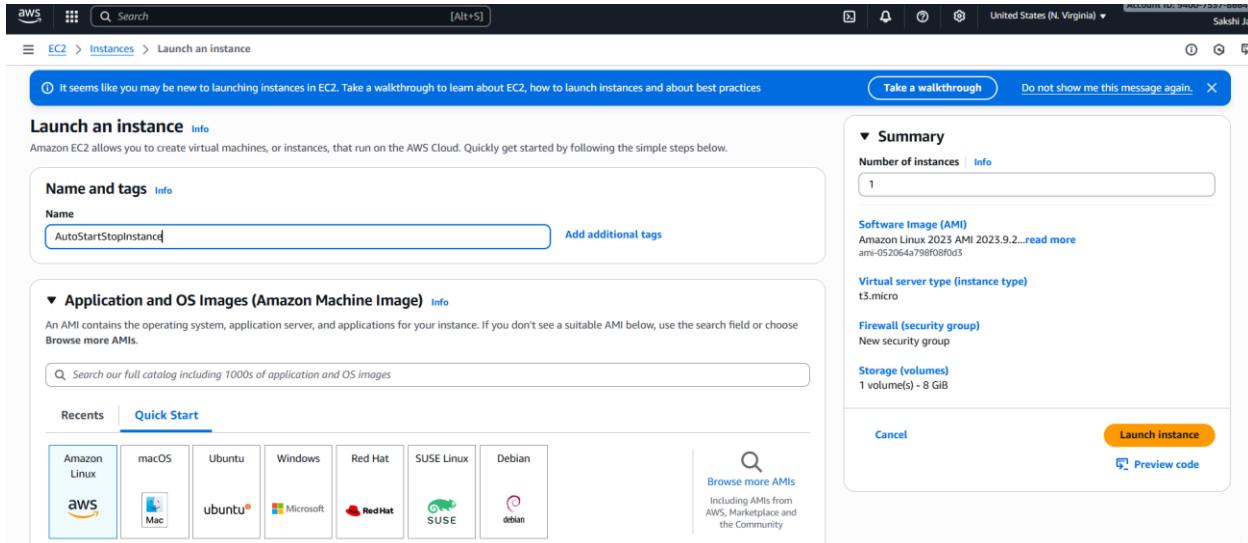
Step 1: Login to AWS

1. Go to <https://aws.amazon.com/console>
 2. Click **Sign in to the Console**.
 3. Use your AWS account credentials.
-



Step 2: Create an EC2 Instance

1. In the **AWS Management Console**, search **EC2** and open it.
2. Click **Launch instance**.
3. Fill details:
 - **Name** → AutoStartStopInstance
 - **Amazon Machine Image (AMI)** → Choose **Amazon Linux 2**
 - **Instance type** → t2.micro (Free Tier)
 - **Key pair** → Choose existing or create a new one (download .pem file).
 - **Security group** → Allow SSH (Port 22)
4. Click **Launch Instance**



Now your instance is running.

Step 3: Create an IAM Role for EC2 Control

This allows your scripts to control EC2 (start and stop).

1. Go to **IAM** service from the console.
2. Click **Roles** → **Create role**.
3. Select **AWS service** → choose **EC2** → click **Next**.
4. Click **Create policy** (in a new tab):
 - Choose **JSON** tab.
 - Paste this:
 - {
 - "Version": "2012-10-17",
 - "Statement": [
 - {
 - "Effect": "Allow",
 - "Action": [
 "ec2:StartInstances",
 "ec2:StopInstances",
 "ec2:DescribeInstances"
],
 - "Resource": "*"
 - }
 -]
 - }
 - Click **Next**, name it **EC2StartStopPolicy**, then **Create policy**.
5. Go back to the role creation tab → **Refresh policies** → search **EC2StartStopPolicy** → select it.
6. Click **Next** → name the role **EC2StartStopRole** → **Create role**.

✓ Role ready!

Step 1 **Specify permissions**

Step 2 Review and create

Specify permissions Info

Add permissions by selecting services, actions, resources, and conditions. Build permission statements using the JSON editor.

Policy editor

```
1 ▼ {
2     "Version": "2012-10-17",
3     "Statement": [
4         {
5             "Effect": "Allow",
6             "Action": [
7                 "ec2:StartInstances",
8                 "ec2:StopInstances",
9                 "ec2:DescribeInstances"
10            ],
11            "Resource": "*"
12        }
13    ]
14 }
15 |
```

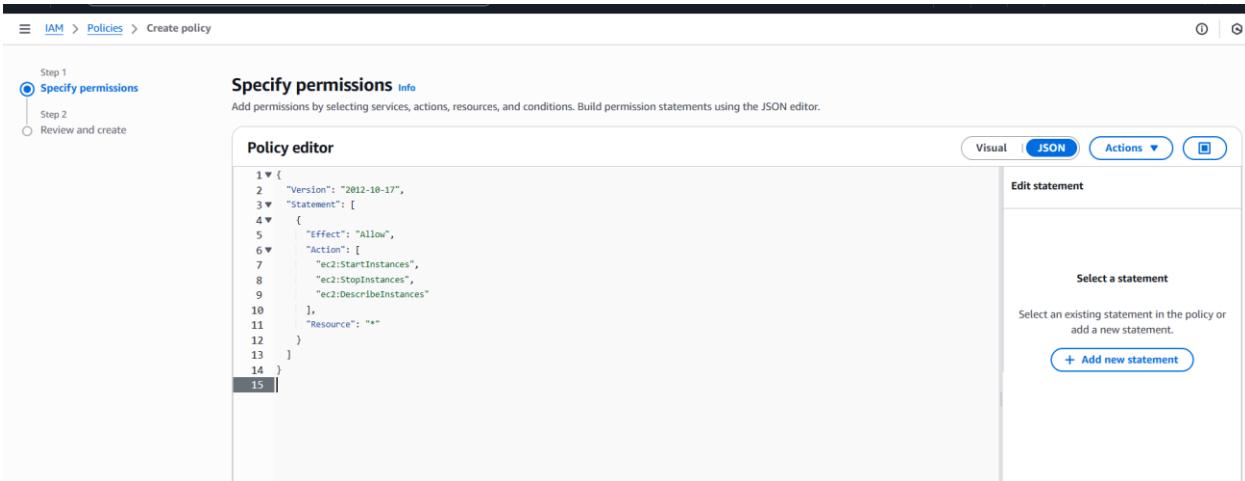
Visual **JSON** Actions ▾

Edit statement

Select a statement

Select an existing statement in the policy or add a new statement.

+ Add new statement



Step 1 **Specify permissions**

Step 2 **Review and create**

Review and create Info

Review the permissions, specify details, and tags.

Policy details

Policy name
Enter a meaningful name to identify this policy.

Maximum 128 characters. Use alphanumeric and '+-,@-_.' characters.

Description - optional
Add a short explanation for this policy.

Maximum 1,000 characters. Use alphanumeric and '+-,@-_.' characters.

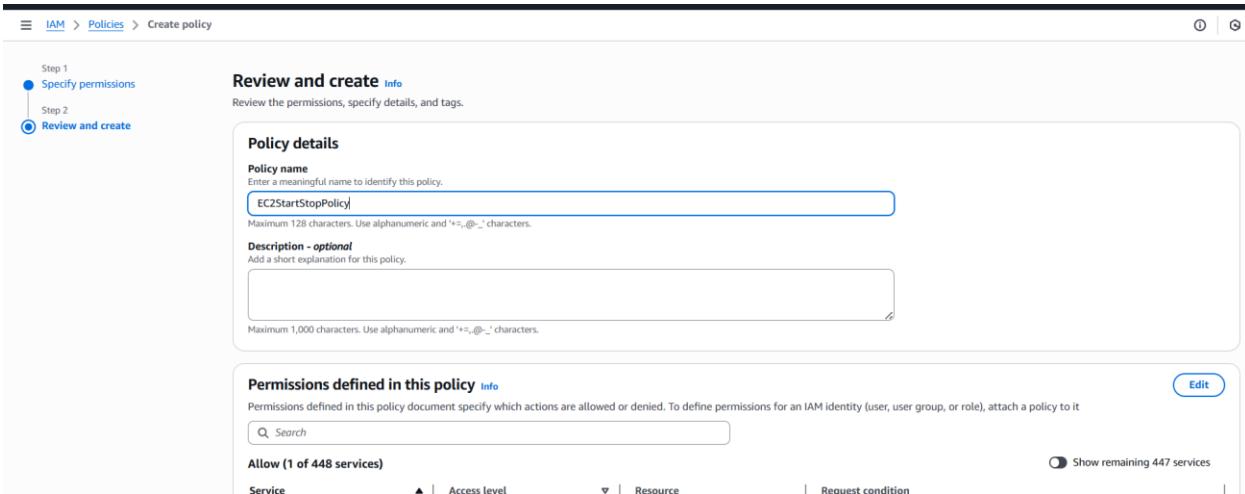
Permissions defined in this policy Info

Permissions defined in this policy document specify which actions are allowed or denied. To define permissions for an IAM identity (user, user group, or role), attach a policy to it.

Allow (1 of 448 services)

Search

Service Access level Resource **Edit**



The screenshot shows the 'Select trusted entity' step of the IAM role creation wizard. It includes a navigation bar with 'Search' and 'Account ID: 9400-7537-8664'. The main area has a sidebar with steps: 'Step 1 Select trusted entity' (selected), 'Step 2 Add permissions', and 'Step 3 Name, review, and create'. The 'Trusted entity type' section contains four options: 'AWS service' (selected), 'AWS account', 'SAML 2.0 federation', and 'Custom trust policy'. Below this is a 'Use case' section for EC2.

The screenshot shows the 'Add permissions' step of the wizard. The sidebar shows 'Step 1 Select trusted entity' (completed) and 'Step 2 Add permissions' (selected). The main area displays a 'Permissions policies (1/1082)' list with one item selected: 'EC2StartStopPolicy'. A 'Set permissions boundary - optional' section is present at the bottom.

The screenshot shows the 'Name, review, and create' step of the wizard. The sidebar shows 'Step 1 Select trusted entity' (completed) and 'Step 2 Add permissions' (completed). The main area includes a 'Role details' section with 'Role name' set to 'EC2StartStopRole' and a 'Description' box. Below is a 'Step 1: Select trusted entities' section with a 'Trust policy' code editor containing:

```
1+{| 2+ "Version": "2012-10-17", 3+ "Statement": [ 4+ { 5+ "Effect": "Allow", 6+ "Action": [ 7+ "sts:AssumeRole"
```

🔗 Step 4: Attach the Role to Your EC2 Instance

1. Go to **EC2 → Instances**.
2. Select your instance → click **Actions → Security → Modify IAM role**.
3. Choose your role: **EC2StartStopRole**.
4. Click **Update IAM role**.

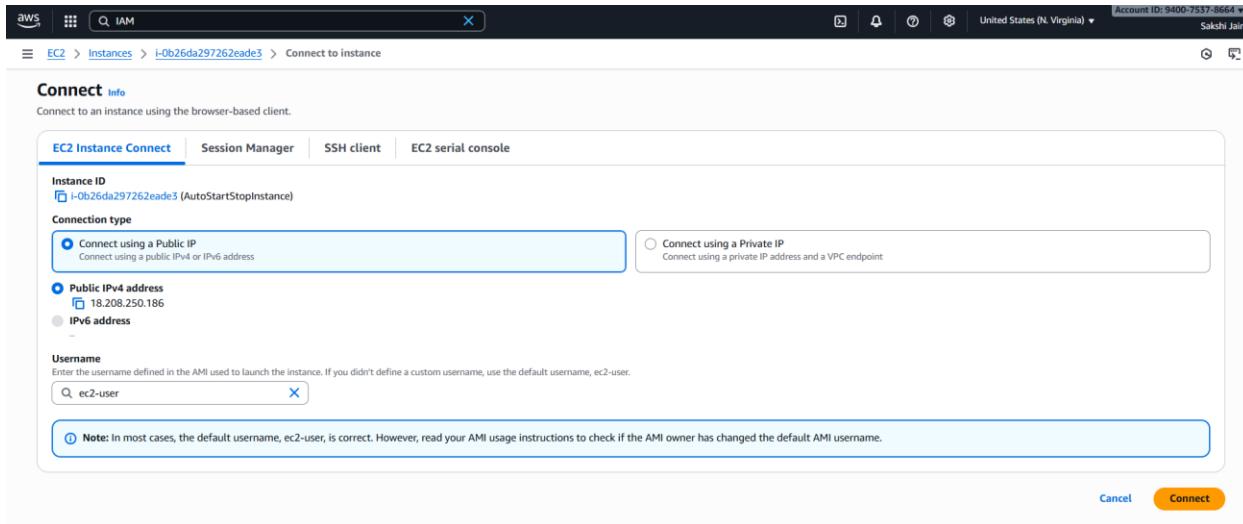
Now your instance has permission to start/stop other EC2 instances.

The screenshot shows the AWS EC2 Instances page. There are two instances listed: "AutoStartStop..." (Running, t3.micro, 3/3 checks passed, us-east-1d) and "Ec2-Backup" (Terminated, t2.micro). A context menu is open over the "AutoStartStop..." instance, with the "Modify IAM role" option highlighted. Below the main list, a modal dialog titled "Modify IAM role" is displayed, showing the "EC2StartStopRole" selected from a dropdown menu. The dialog also includes a "Create new IAM role" button and "Cancel" and "Update IAM role" buttons at the bottom.

Step 5: Connect to Your EC2 Instance

1. In EC2 Dashboard, select your instance → click **Connect**.
2. Choose **EC2 Instance Connect → Connect**.
OR
If using your PC terminal:
3. `ssh -i "your-key.pem" ec2-user@<your-public-ip>`

Now you're inside your EC2 terminal (Linux shell).



📋 Step 6: Install AWS CLI

If it's not already installed:

```
sudo yum update -y  
sudo yum install aws-cli -y
```

Check version:

```
aws --version
```

💻 Step 7: Create Start and Stop Scripts

Start Script

```
vim ~/start_ec2.sh
```

Paste:

```
#!/bin/bash  
INSTANCE_ID="i-0b26da297262eade3" # Replace with your actual instance ID  
aws ec2 start-instances --instance-ids $INSTANCE_ID  
echo "Instance $INSTANCE_ID started at $(date)" >> /home/ec2-user/ec2_log.txt
```

```
#!/bin/bash
INSTANCE_ID="i-0b26da297262eade3" # Replace with your actual instance ID
aws ec2 start-instances --instance-ids $INSTANCE_ID
echo "Instance $INSTANCE_ID started at $(date)" >> /home/ec2-user/ec2_log.txt
```

-- INSERT --

0,0-1 Al
0,1 Al
5,1 Al

Make it executable:

```
chmod +x ~/start_ec2.sh
```

Stop Script

```
vim ~/stop_ec2.sh
```

Paste:

```
#!/bin/bash
INSTANCE_ID="i-0b26da297262eade3" # Replace with your actual instance ID
aws ec2 stop-instances --instance-ids $INSTANCE_ID
echo "Instance $INSTANCE_ID stopped at $(date)" >> /home/ec2-user/ec2_log.txt
```

```
#!/bin/bash
INSTANCE_ID="i-0b26da297262eade3" # Replace with your actual instance ID
aws ec2 stop-instances --instance-ids $INSTANCE_ID
echo "Instance $INSTANCE_ID stopped at $(date)" >> /home/ec2-user/ec2_log.txt
```

-- INSERT --

0,0-1 Al
0,1 Al
5,0-1 Al

```
chmod +x ~/stop_ec2.sh
```



Tip:
You can find your instance ID in the **EC2 Dashboard** → **Instances** → **Instance ID** column.



Step 8: Automate with Cron Job

Now, you'll schedule the scripts to run automatically.

```
[root@ip-172-31-28-253 ~]# yum install cronie
amazon Linux 2023 Kernel Livepatch repository
Amazon Linux 2023 Kernel Livepatch repository
=====
Dependencies resolved.
=====
== Package                                Architecture      Version           Repository
e
=====
== Installing:
  cronie                               x86_64          1.5.7-1.amzn2023.0.2      amazonlinux
k
  Installing dependencies:
    cronie-anacron                      x86_64          1.5.7-1.amzn2023.0.2      amazonlinux
k
=====
Transaction Summary
```

Open cron editor:

```
crontab -e
```

At the bottom, add these lines (example times):

```
0 9 * * * /home/ec2-user/start_ec2.sh
30 12 * * * /home/ec2-user/stop_ec2.sh
```

- 0 9 * * * → Run every day at 9:00 AM
- 0 18 * * * → Run every day at 12:30 PM

The screenshot shows the AWS Lambda console interface. At the top, there's a navigation bar with the AWS logo, search bar, and account information (Account ID: 9400-7537-8664, United States (N. Virginia), Sakshi Jain). Below the navigation bar is a terminal window displaying the cron configuration. The terminal shows two scheduled tasks:

```
0 9 * * * /home/ec2-user/start_ec2.sh
30 12 * * * /home/ec2-user/stop_ec2.sh
```

The terminal also shows the status of the tasks, with some entries having green checkmarks and others having red error icons. At the bottom of the terminal, it says "INSERT --". On the right side of the terminal, there are status indicators: 0,0-1, 0,1, Al, and 3.

Check scheduled tasks:

```
crontab -l
```

```
[root@ip-172-31-28-253 ~]# crontab -l
0 9 * * * /home/ec2-user/start_ec2.sh
30 12 * * * /home/ec2-user/stop_ec2.sh
```

✳ Step 9: Verify

To test immediately:

```
./stop_ec2.sh # should stop instance
```

```
[root@ip-172-31-28-253 ~]# ./stop_ec2.sh
{
    "StoppingInstances": [
        {
            "InstanceId": "i-0b26da297262eade3",
            "CurrentState": {
                "Code": 64,
                "Name": "stopping"
            },
            "PreviousState": {
                "Code": 16,
                "Name": "running"
            }
        }
    ]
}
```

After start instance and connect instance again and run below command

```
./start_ec2.sh # should start instance
```

Then check on AWS Console → EC2 → your instance → it should **start/stop automatically**.

💻 Step 10: Check Logs

Open log file to see automatic entries:

```
cat /home/ec2-user/ec2_log.txt
```

```
[root@ip-172-31-28-253 ~]# cat /home/ec2-user/ec2_log.txt
Instance i-0b26da297262eade3 stopped at Wed Oct 15 06:44:47 UTC 2025
[root@ip-172-31-28-253 ~]# date
Wed Oct 15 06:52:17 UTC 2025
```

You'll see:

Instance i-0b26da297262eade3 started at Wed Oct 15 09:00:00 2025
Instance i-0b26da297262eade3 stopped at Wed Oct 15 18:00:00 2025