Devops

AWS CLI Full Guide

Two Ways to Login to EC2 Instance

Method 1: AWS Console (Web-based)

Steps:

1. Go to **AWS Console** → **EC2 Dashboard**
2. Click on **Instances (running)**
3. Select your **Instance**
4. Click **Instance ID**
5. Click **"Connect"** button at the top
6. Choose **"EC2 Instance Connect"** tab
7. Click **"Connect"**

✅ Opens a **browser-based terminal** directly to your instance

Pros:

- No setup needed
- Works from anywhere
- No key file needed

Cons:

- Requires internet browser
- Less flexible than terminal

Method 2: Terminal (SSH via MobaXterm/Local Terminal)

For Linux/Mac/MobaXterm:

bash

chmod 600 /path/to/your-key.pem

ssh -i /path/to/your-key.pem ubuntu@<IP-ADDRESS>

Example:

bash

chmod 600 ~/Downloads/my-ec2-key.pem

ssh -i ~/Downloads/my-ec2-key.pem ubuntu@54.123.45.67

Breakdown:

- chmod 600 - Sets proper permissions (owner read/write only)
- -i - Specifies identity/key file
- ubuntu - Username (varies by AMI)
- @54.123.45.67 - Public IP address of EC2 instance

Pros:

- More control and flexibility
- Can use multiple terminal windows
- Better for scripting and automation

Managing EC2 Instance Lifecycle

Stop an Instance:

1. Select your **instance**
2. Click **"Instance state"** dropdown
3. Select **"Stop instance"**

💡 **Stopped instance:**

- No longer running (not charged for compute)
- Data preserved on EBS volume
- Can restart later

Terminate an Instance:

1. Select your **instance**
2. Click **"Instance state"** dropdown
3. Select **"Terminate instance"**

⚠️ **Terminated instance:**

- Permanently deleted
- Cannot be restarted

- All data lost (unless EBS volume configured to persist)

Automating with AWS CLI

Step 1: Install AWS CLI

Download AWS CLI:

- **Windows:** Download installer from aws.amazon.com/cli
- **Mac:** brew install awscli
- **Linux:** sudo apt install awscli or pip install awscli

Verify installation:

bash

aws --version

Step 2: Create AWS Access Keys

Problem: AWS CLI doesn't know your account credentials

Solution: Create Access Keys

Steps:

1. Go to **AWS Console**
2. Click your **name** (top right corner)
3. Select **"Security credentials"**
4. Scroll down to **"Access keys"** section
5. Click **"Create access key"**
6. Choose **"Command Line Interface (CLI)"**
7. Check the acknowledgment box
8. Click **"Create access key"**

You'll receive:

- **Access Key ID** (e.g., AKIAIOSFODNN7EXAMPLE)
- **Secret Access Key** (e.g., wJalrXUtnFEMI/K7MDENG/bPxRfiCYEXAMPLEKEY)

⚠️ **IMPORTANT:**

- **Download** or copy these immediately
- Secret key is **shown only once**
- Keep them **secure** (never commit to Git!)

Step 3: Configure AWS CLI

Run in Terminal:

bash

aws configure

```

**You'll be prompted for:**

```

AWS Access Key ID [None]: AKIAIOSFODNN7EXAMPLE

AWS Secret Access Key [None]: wJalrXUtnFEMI/K7MDENG/bPxRfiCYEXAMPLEKEY

Default region name [None]: us-east-1

Default output format [None]: json

Configuration stored in:

- ~/.aws/credentials (Linux/Mac)
- C:\Users\USERNAME\.aws\credentials (Windows)

Step 4: Test AWS CLI

List EC2 Instances:

bash

aws ec2 describe-instances

Create EC2 Instance:

bash

aws ec2 run-instances \

 --image-id ami-0c55b159cbfafe1f0 \

--instance-type t2.micro \

--key-name my-ec2-key \

--count 1

Stop an Instance:

bash

aws ec2 stop-instances --instance-ids i-1234567890abcdef0

Terminate an Instance:

bash

aws ec2 terminate-instances --instance-ids i-1234567890abcdef0

AWS CLI vs Console

Action	AWS Console	AWS CLI
Create 1 EC2	2-3 minutes, many clicks	10 seconds, one command
Create 100 EC2	Hours of manual work	10 seconds, one command
Automation	❌ Not possible	✅ Fully scriptable
Efficiency	Low	**High**

Security Best Practices

Access Keys:

1. **Never hardcode** in scripts
2. **Never commit** to Git/GitHub
3. **Rotate regularly** (change every 90 days)
4. **Use IAM roles** when possible (for EC2 instances)
5. **Delete unused keys**

PEM Files:

1. Always use chmod 600 on .pem files
2. Store securely (not in public directories)

3. Back up in secure location

Next Steps: Automation Examples

Create 10 EC2 instances in one command:

bash

```
aws ec2 run-instances \
  --image-id ami-0c55b159cbfafe1f0 \
  --instance-type t2.micro \
  --key-name my-ec2-key \
  --count 10
```

Script to auto-create EC2 when requested:

bash

```
#!/bin/bash
echo "Creating EC2 instance..."
aws ec2 run-instances \
  --image-id ami-0c55b159cbfafe1f0 \
  --instance-type t2.micro \
  --key-name my-ec2-key \
  --count 1
echo "EC2 instance created!"
```

This is the **DevOps way** - automate everything! 🚀