Guided Lab: How to install CloudFormation helper scripts on Ubuntu

Description

CloudFormation helper scripts are essential tools provided by AWS to simplify and automate the setup and management of resources defined in CloudFormation templates. Unlike AWS-provided Linux AMIs, which come with pre-installed tools, an Ubuntu instance requires manual installation of these scripts. This guide will walk you through the process of installing CloudFormation helper scripts on Supported Ubuntu Versions.

- Ubuntu 16.04 LTS
- Ubuntu 18.04 LTS
- Ubuntu 20.04 LTS
- Ubuntu 22.04 LTS

Prerequisites

This lab assumes you have experience creating an Amazon EC2 Instance and has knowledge of basic fundamentals of the command line.

If you find any gaps in your knowledge, consider taking the following labs:

- Creating an Amazon EC2 instance (Linux)
- Setting up a Web server on an EC2 instance

Objectives

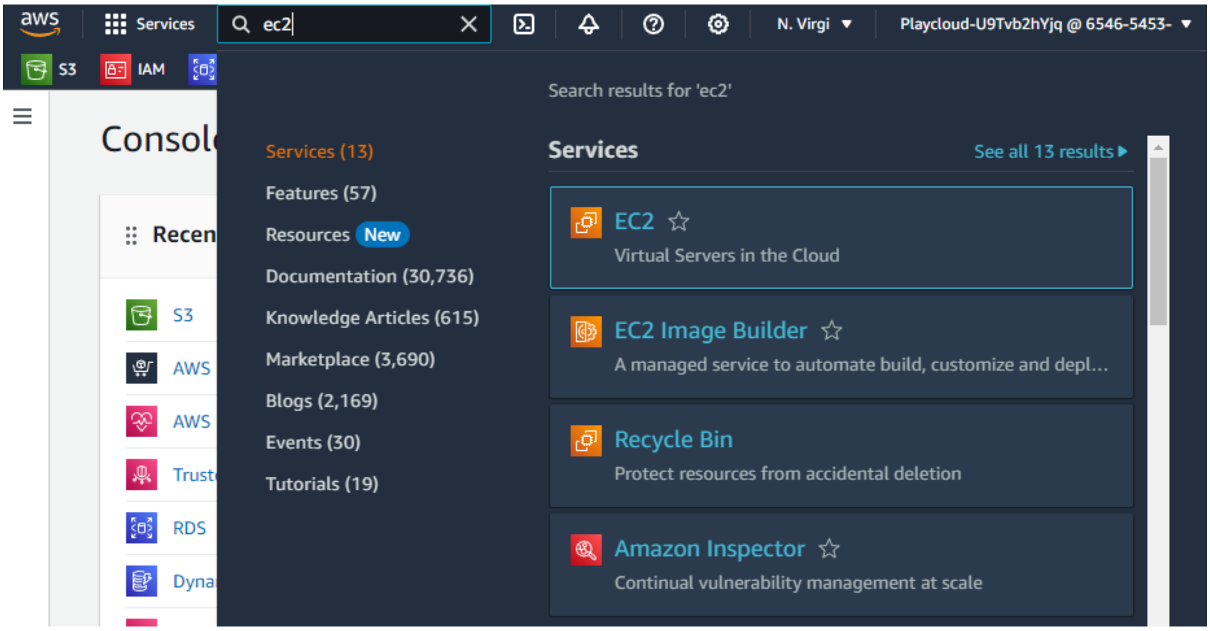
By the end of this guide, you will have successfully installed the AWS CloudFormation helper scripts on your Ubuntu instance, allowing you to leverage these tools for managing your CloudFormation stacks.

Subscribe to access AWS PlayCloud Labs

Lab Steps

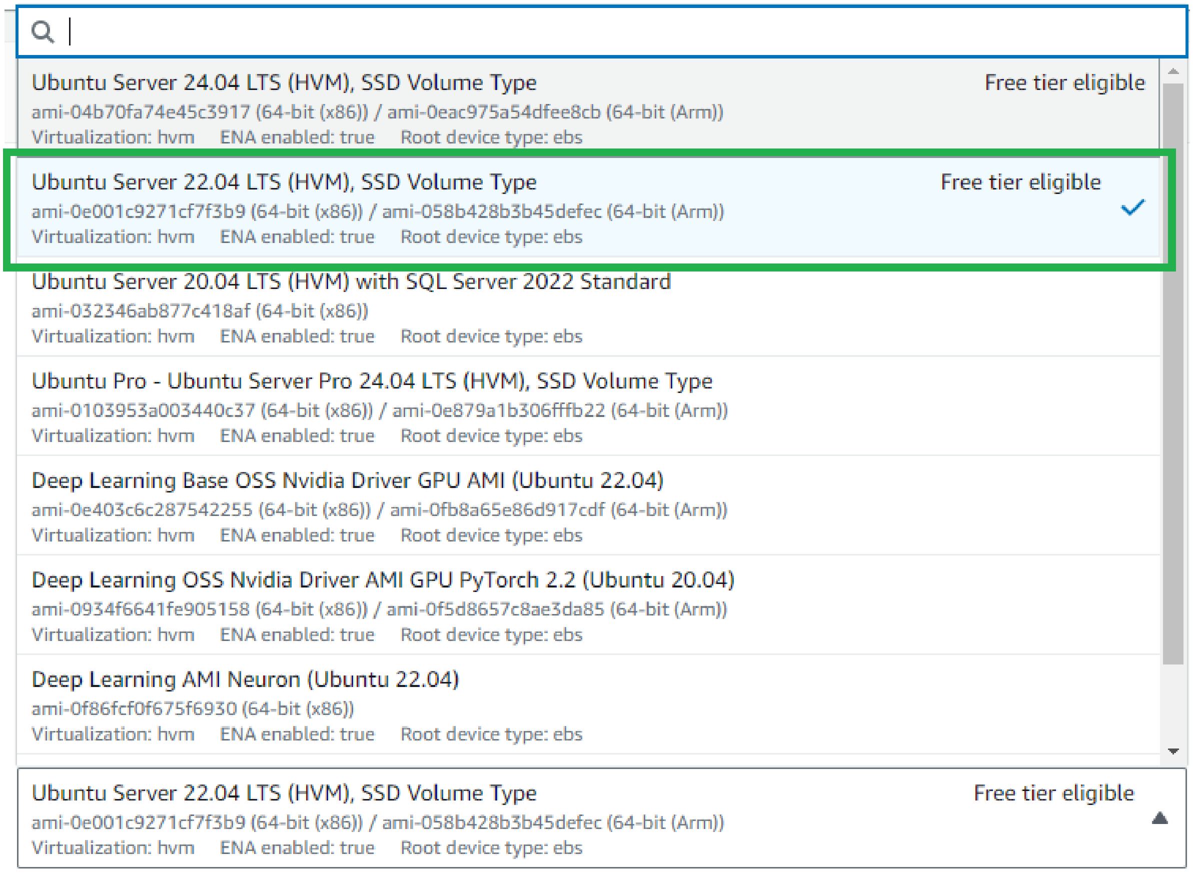
Launching your EC2 Instance

1. Search '*ec2*' in the AWS Management Console search bar. Click EC2 on the search results.



2. Launch EC2 instance with the following configurations.

- **Name: td-installing-cloudformation-helper**
- **AMI: Ubuntu Server 22.04 LTS**



Ubuntu Server 24.04 LTS (HVM), SSD Volume Type — Free tier eligible
ami-04b70fa74e45c3917 (64-bit (x86)) / ami-0eac975a54dfee8cb (64-bit (Arm))
Virtualization: hvm ENA enabled: true Root device type: ebs

Ubuntu Server 22.04 LTS (HVM), SSD Volume Type — Free tier eligible ✓
ami-0e001c9271cf7f3b9 (64-bit (x86)) / ami-058b428b3b45defec (64-bit (Arm))
Virtualization: hvm ENA enabled: true Root device type: ebs

Ubuntu Server 20.04 LTS (HVM) with SQL Server 2022 Standard
ami-032346ab877c418af (64-bit (x86))
Virtualization: hvm ENA enabled: true Root device type: ebs

Ubuntu Pro - Ubuntu Server Pro 24.04 LTS (HVM), SSD Volume Type
ami-0103953a003440c37 (64-bit (x86)) / ami-0e879a1b306fffb22 (64-bit (Arm))
Virtualization: hvm ENA enabled: true Root device type: ebs

Deep Learning Base OSS Nvidia Driver GPU AMI (Ubuntu 22.04)
ami-0e403c6c287542255 (64-bit (x86)) / ami-0fb8a65e86d917cdf (64-bit (Arm))
Virtualization: hvm ENA enabled: true Root device type: ebs

Deep Learning OSS Nvidia Driver AMI GPU PyTorch 2.2 (Ubuntu 20.04)
ami-0934f6641fe905158 (64-bit (x86)) / ami-0f5d8657c8ae3da85 (64-bit (Arm))
Virtualization: hvm ENA enabled: true Root device type: ebs

Deep Learning AMI Neuron (Ubuntu 22.04)
ami-0f86fcf0f675f6930 (64-bit (x86))
Virtualization: hvm ENA enabled: true Root device type: ebs

Ubuntu Server 22.04 LTS (HVM), SSD Volume Type — Free tier eligible
ami-0e001c9271cf7f3b9 (64-bit (x86)) / ami-058b428b3b45defec (64-bit (Arm))
Virtualization: hvm ENA enabled: true Root device type: ebs

- **Instance type: t2.micro**

- **Key pair:**
    - **Click on Create new key pair and name it myKeyPair**
      Take note where you downloaded your keypair
- **Network settings:**
    - **Allow SSH traffic from: (In production environment, normally, you choose My IP for best practice. However, for simplicity of this lab lets choose Anywhere)**
    - **Auto-assign public IP: Enable**

3. Click on Launch instance

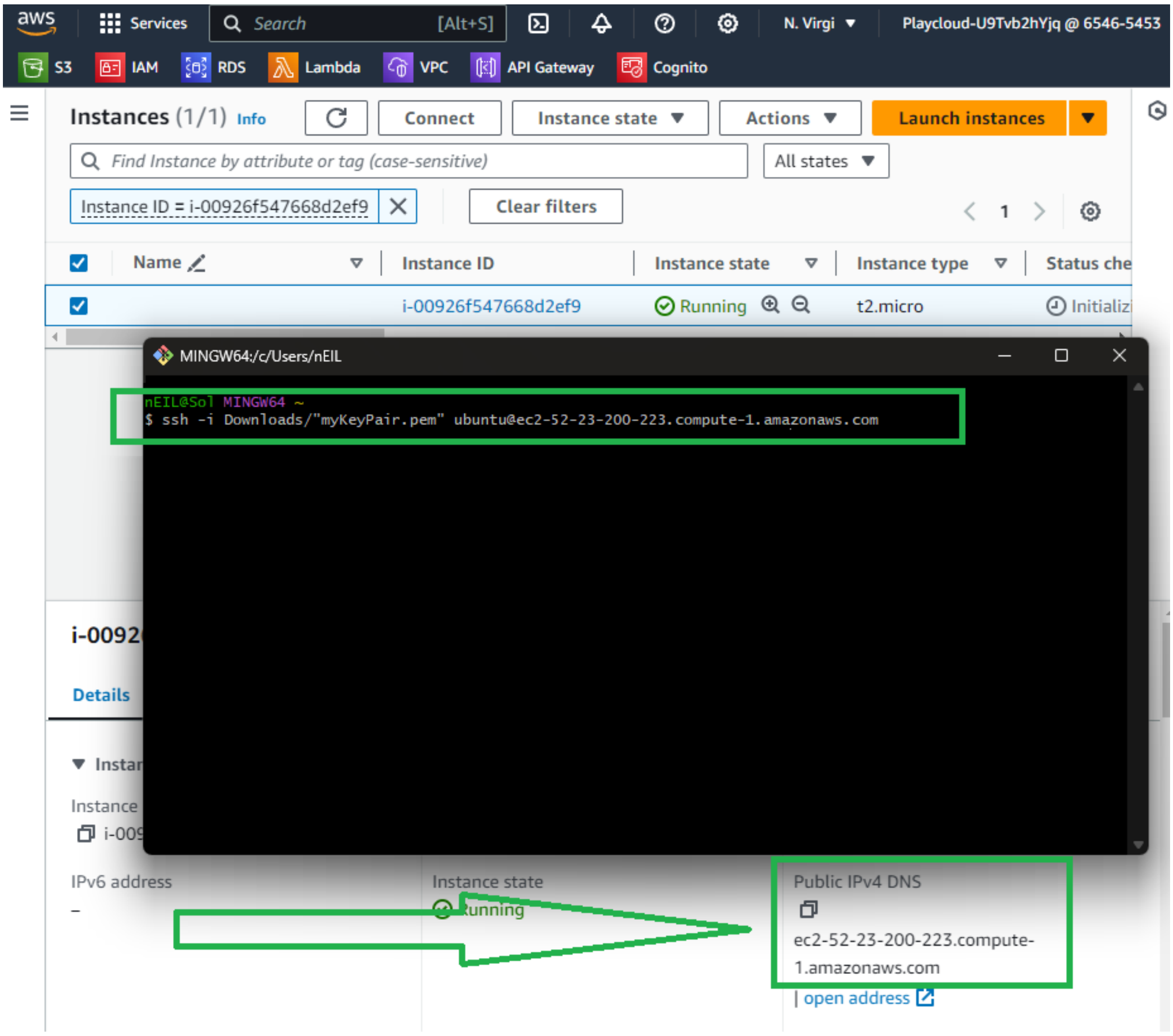
Installing the CloudFormation Helper Scripts

1. **Connect to Your EC2 Instance via SSH**

- Open your terminal/Git bash and connect to your EC2 instance using SSH:

```
ssh -i <directory_of_your_keypair>/"myKeyPair.pem"
ubuntu@<your_ec2_instance_Public_IPv4_DNS>
```

Do not forget to change the placheolders <directory_of_your_keypair> *and* <your_ec2_instance_Public_IPv4_DNS>



Instance ID = i-00926f547668d2ef9

☑	Name	Instance ID	Instance state	Instance type	Status che
☑		i-00926f547668d2ef9	Running	t2.micro	Initializi

MINGW64:/c/Users/nEIL

```
nEIL@Sol MINGW64 ~
$ ssh -i Downloads/"myKeyPair.pem" ubuntu@ec2-52-23-200-223.compute-1.amazonaws.com
```

i-0092

Instar

Instance
i-009

IPv6 address
–

Instance state
Running

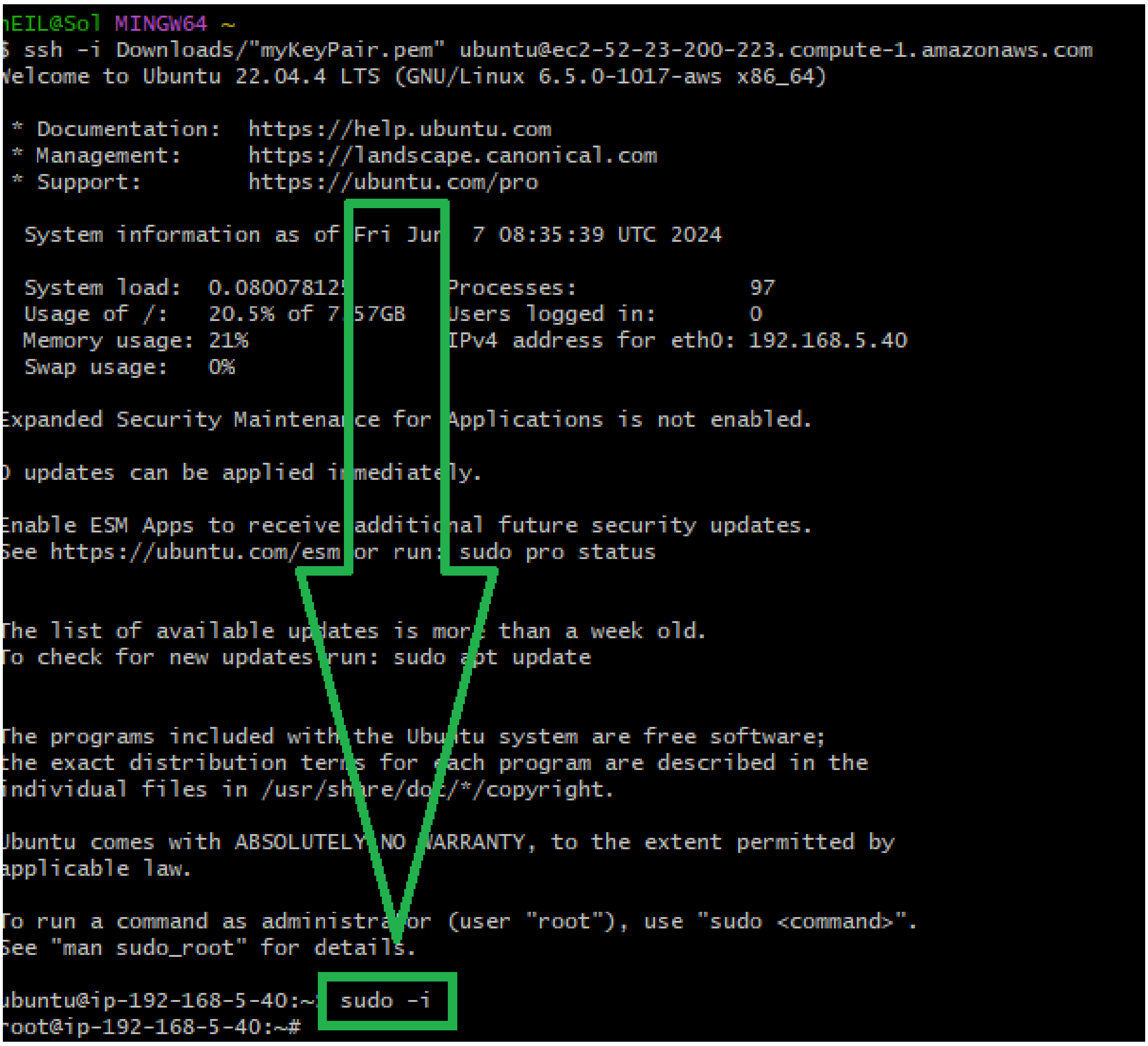
Public IPv4 DNS
ec2-52-23-200-223.compute-1.amazonaws.com

- You will receive a command response, saying:
  ”Are you sure you want to continue connecting (yes/no/[fingerprint])? “
-
  - Type “**yes**”

2. **Switch to Root User**

- Elevate your privileges by switching to the root user:

sudo -i

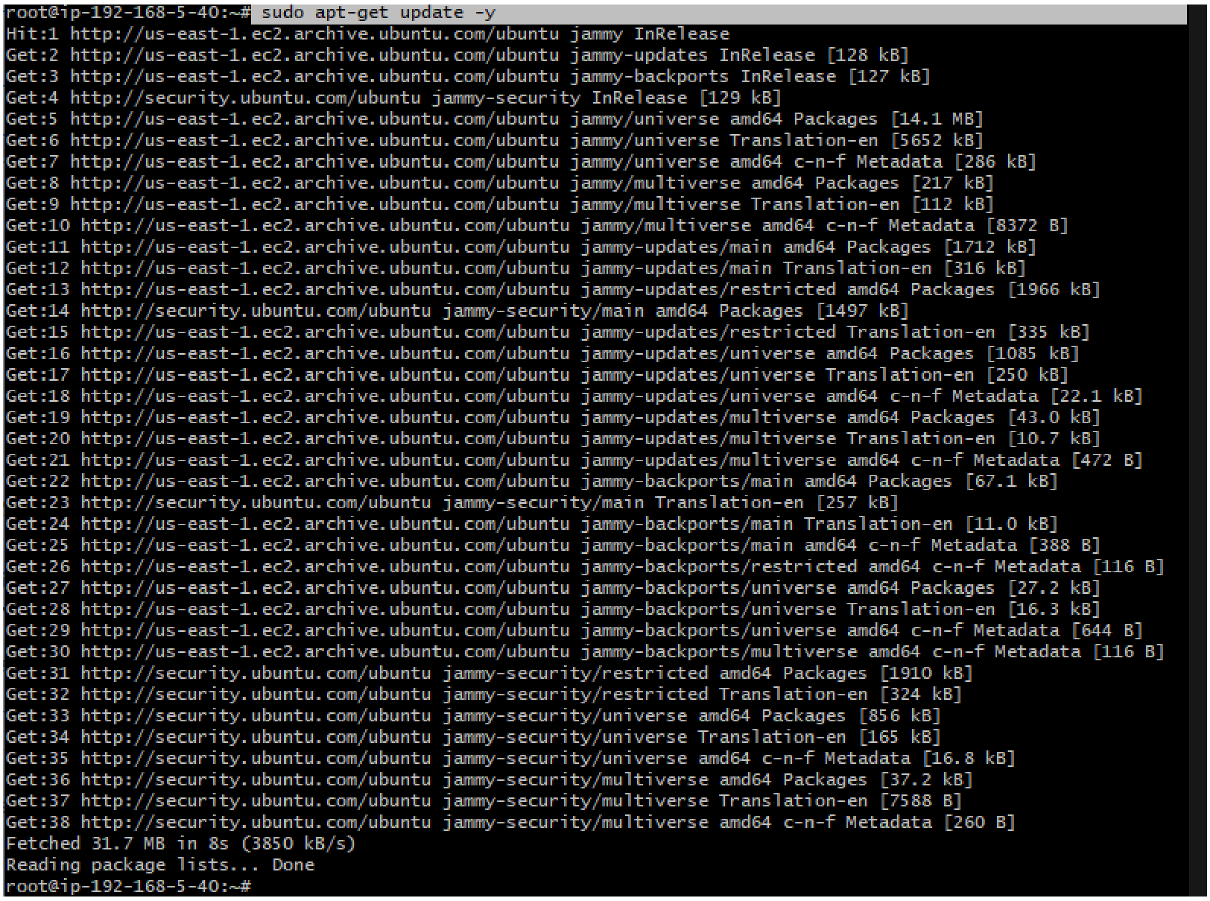


```
nEIL@Sol MINGW64 ~
$ ssh -i Downloads/"myKeyPair.pem" ubuntu@ec2-52-23-200-223.compute-1.amazonaws.com
Welcome to Ubuntu 22.04.4 LTS (GNU/Linux 6.5.0-1017-aws x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:     https://landscape.canonical.com
 * Support:        https://ubuntu.com/pro

  System information as of Fri Jun  7 08:35:39 UTC 2024

  System load:  0.080078125       Processes:             97
  Usage of /:   20.5% of 7.57GB   Users logged in:       0
  Memory usage: 21%               IPv4 address for eth0: 192.168.5.40
  Swap usage:   0%

Expanded Security Maintenance for Applications is not enabled.

0 updates can be applied immediately.

Enable ESM Apps to receive additional future security updates.
See https://ubuntu.com/esm or run: sudo pro status


The list of available updates is more than a week old.
To check for new updates run: sudo apt update


The programs included with the Ubuntu system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by
applicable law.

To run a command as administrator (user "root"), use "sudo <command>".
See "man sudo_root" for details.

ubuntu@ip-192-168-5-40:~$ sudo -i
root@ip-192-168-5-40:~#
```

3. **Update Package Repository:**

- Ensure that the package repository is up to date by running:

sudo apt-get update -y

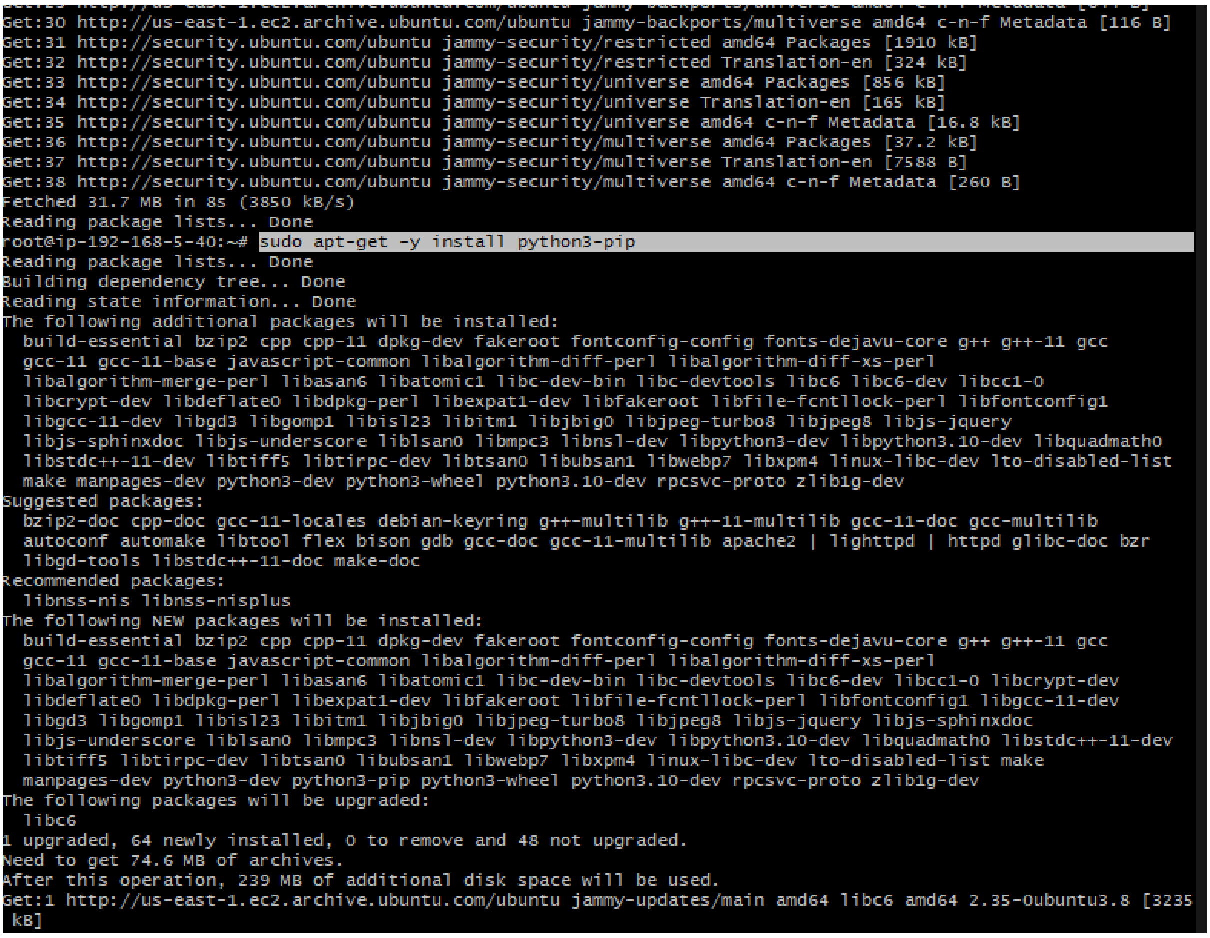


```
root@ip-192-168-5-40:~# sudo apt-get update -y
Hit:1 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy InRelease
Get:2 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy-updates InRelease [128 kB]
Get:3 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy-backports InRelease [127 kB]
Get:4 http://security.ubuntu.com/ubuntu jammy-security InRelease [129 kB]
Get:5 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy/universe amd64 Packages [14.1 MB]
Get:6 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy/universe Translation-en [5652 kB]
Get:7 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy/universe amd64 c-n-f Metadata [286 kB]
Get:8 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy/multiverse amd64 Packages [217 kB]
Get:9 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy/multiverse Translation-en [112 kB]
Get:10 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy/multiverse amd64 c-n-f Metadata [8372 B]
Get:11 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy-updates/main amd64 Packages [1712 kB]
Get:12 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy-updates/main Translation-en [316 kB]
Get:13 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy-updates/restricted amd64 Packages [1966 kB]
Get:14 http://security.ubuntu.com/ubuntu jammy-security/main amd64 Packages [1497 kB]
Get:15 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy-updates/restricted Translation-en [335 kB]
Get:16 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy-updates/universe amd64 Packages [1085 kB]
Get:17 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy-updates/universe Translation-en [250 kB]
Get:18 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy-updates/universe amd64 c-n-f Metadata [22.1 kB]
Get:19 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy-updates/multiverse amd64 Packages [43.0 kB]
Get:20 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy-updates/multiverse Translation-en [10.7 kB]
Get:21 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy-updates/multiverse amd64 c-n-f Metadata [472 B]
Get:22 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy-backports/main amd64 Packages [67.1 kB]
Get:23 http://security.ubuntu.com/ubuntu jammy-security/main Translation-en [257 kB]
Get:24 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy-backports/main Translation-en [11.0 kB]
Get:25 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy-backports/main amd64 c-n-f Metadata [388 B]
Get:26 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy-backports/restricted amd64 c-n-f Metadata [116 B]
Get:27 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy-backports/universe amd64 Packages [27.2 kB]
Get:28 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy-backports/universe Translation-en [16.3 kB]
Get:29 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy-backports/universe amd64 c-n-f Metadata [644 B]
Get:30 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy-backports/multiverse amd64 c-n-f Metadata [116 B]
Get:31 http://security.ubuntu.com/ubuntu jammy-security/restricted amd64 Packages [1910 kB]
Get:32 http://security.ubuntu.com/ubuntu jammy-security/restricted Translation-en [324 kB]
Get:33 http://security.ubuntu.com/ubuntu jammy-security/universe amd64 Packages [856 kB]
Get:34 http://security.ubuntu.com/ubuntu jammy-security/universe Translation-en [165 kB]
Get:35 http://security.ubuntu.com/ubuntu jammy-security/universe amd64 c-n-f Metadata [16.8 kB]
Get:36 http://security.ubuntu.com/ubuntu jammy-security/multiverse amd64 Packages [37.2 kB]
Get:37 http://security.ubuntu.com/ubuntu jammy-security/multiverse Translation-en [7588 B]
Get:38 http://security.ubuntu.com/ubuntu jammy-security/multiverse amd64 c-n-f Metadata [260 B]
Fetched 31.7 MB in 8s (3850 kB/s)
Reading package lists... Done
root@ip-192-168-5-40:~#
```

4. . **Install Python3 and pip:**

- CloudFormation helper scripts require Python3 and pip. Install them using the following command:

sudo apt-get -y install python3-pip



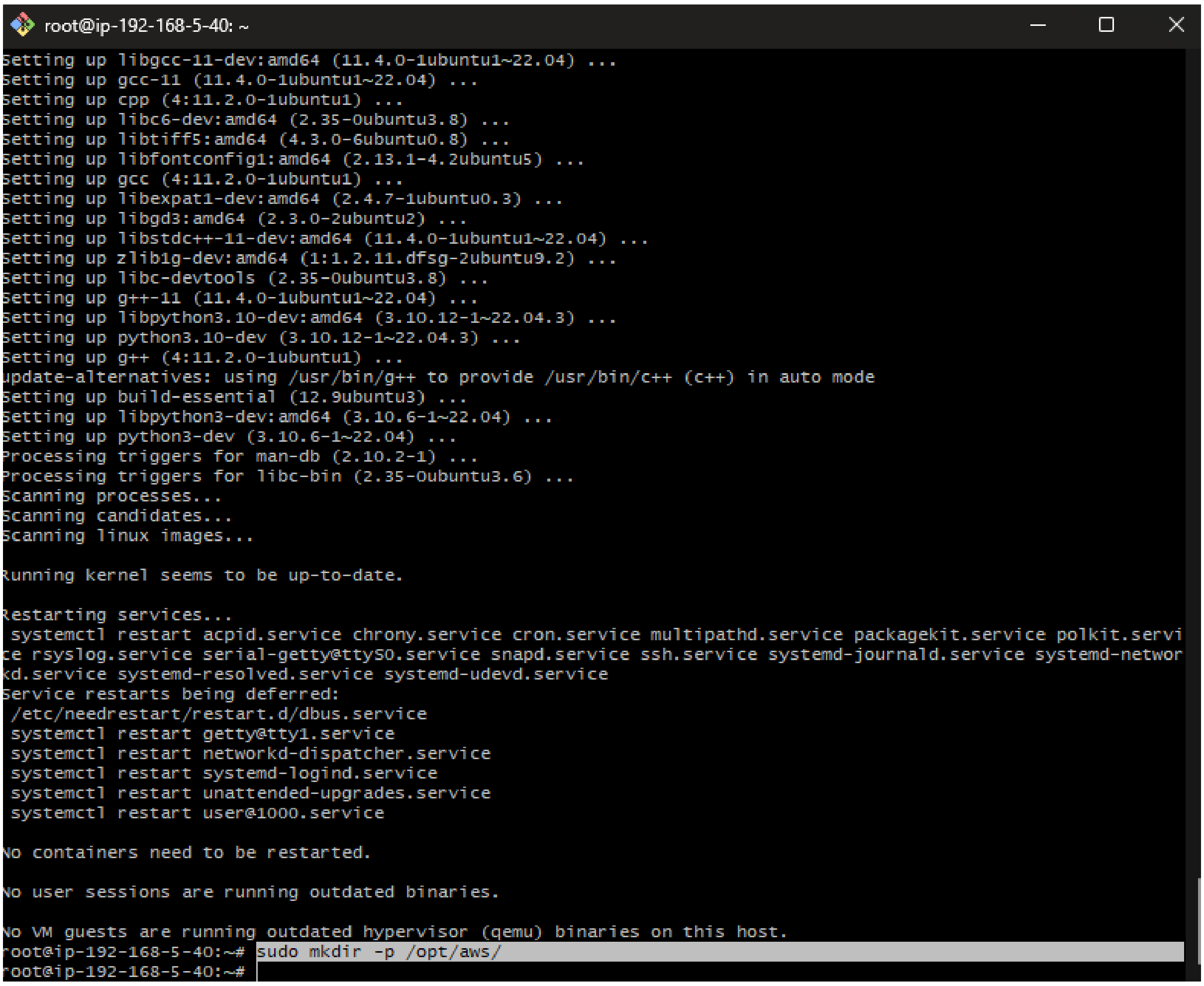
```
Get:30 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy-backports/multiverse amd64 c-n-f Metadata [116 B]
Get:31 http://security.ubuntu.com/ubuntu jammy-security/restricted amd64 Packages [1910 kB]
Get:32 http://security.ubuntu.com/ubuntu jammy-security/restricted Translation-en [324 kB]
Get:33 http://security.ubuntu.com/ubuntu jammy-security/universe amd64 Packages [856 kB]
Get:34 http://security.ubuntu.com/ubuntu jammy-security/universe Translation-en [165 kB]
Get:35 http://security.ubuntu.com/ubuntu jammy-security/universe amd64 c-n-f Metadata [16.8 kB]
Get:36 http://security.ubuntu.com/ubuntu jammy-security/multiverse amd64 Packages [37.2 kB]
Get:37 http://security.ubuntu.com/ubuntu jammy-security/multiverse Translation-en [7588 B]
Get:38 http://security.ubuntu.com/ubuntu jammy-security/multiverse amd64 c-n-f Metadata [260 B]
Fetched 31.7 MB in 8s (3850 kB/s)
Reading package lists... Done
root@ip-192-168-5-40:~# sudo apt-get -y install python3-pip
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following additional packages will be installed:
  build-essential bzip2 cpp cpp-11 dpkg-dev fakeroot fontconfig-config fonts-dejavu-core g++ g++-11 gcc
  gcc-11 gcc-11-base javascript-common libalgorithm-diff-perl libalgorithm-diff-xs-perl
  libalgorithm-merge-perl libasan6 libatomic1 libc-dev-bin libc-devtools libc6 libc6-dev libcc1-0
  libcrypt-dev libdeflate0 libdpkg-perl libexpat1-dev libfakeroot libfile-fcntllock-perl libfontconfig1
  libgcc-11-dev libgd3 libgomp1 libisl23 libitm1 libjbig0 libjpeg-turbo8 libjpeg8 libjs-jquery
  libjs-sphinxdoc libjs-underscore liblsan0 libmpc3 libnsl-dev libpython3-dev libpython3.10-dev libquadmath0
  libstdc++-11-dev libtiff5 libtirpc-dev libtsan0 libubsan1 libwebp7 libxpm4 linux-libc-dev lto-disabled-list
  make manpages-dev python3-dev python3-wheel python3.10-dev rpcsvc-proto zlib1g-dev
Suggested packages:
  bzip2-doc cpp-doc gcc-11-locales debian-keyring g++-multilib g++-11-multilib gcc-11-doc gcc-multilib
  autoconf automake libtool flex bison gdb gcc-doc gcc-11-multilib apache2 | lighttpd | httpd glibc-doc bzr
  libgd-tools libstdc++-11-doc make-doc
Recommended packages:
  libnss-nis libnss-nisplus
The following NEW packages will be installed:
  build-essential bzip2 cpp cpp-11 dpkg-dev fakeroot fontconfig-config fonts-dejavu-core g++ g++-11 gcc
  gcc-11 gcc-11-base javascript-common libalgorithm-diff-perl libalgorithm-diff-xs-perl
  libalgorithm-merge-perl libasan6 libatomic1 libc-dev-bin libc-devtools libc6-dev libcc1-0 libcrypt-dev
  libdeflate0 libdpkg-perl libexpat1-dev libfakeroot libfile-fcntllock-perl libfontconfig1 libgcc-11-dev
  libgd3 libgomp1 libisl23 libitm1 libjbig0 libjpeg-turbo8 libjpeg8 libjs-jquery libjs-sphinxdoc
  libjs-underscore liblsan0 libmpc3 libnsl-dev libpython3-dev libpython3.10-dev libquadmath0 libstdc++-11-dev
  libtiff5 libtirpc-dev libtsan0 libubsan1 libwebp7 libxpm4 linux-libc-dev lto-disabled-list make
  manpages-dev python3-dev python3-pip python3-wheel python3.10-dev rpcsvc-proto zlib1g-dev
The following packages will be upgraded:
  libc6
1 upgraded, 64 newly installed, 0 to remove and 48 not upgraded.
Need to get 74.6 MB of archives.
After this operation, 239 MB of additional disk space will be used.
Get:1 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy-updates/main amd64 libc6 amd64 2.35-0ubuntu3.8 [3235
 kB]
```

- If you receive a command response, just press **ENTER.**

5. **Create Directory for CloudFormation Helper Scripts:**

- Create a directory to store the CloudFormation helper scripts.

sudo mkdir -p /opt/aws/

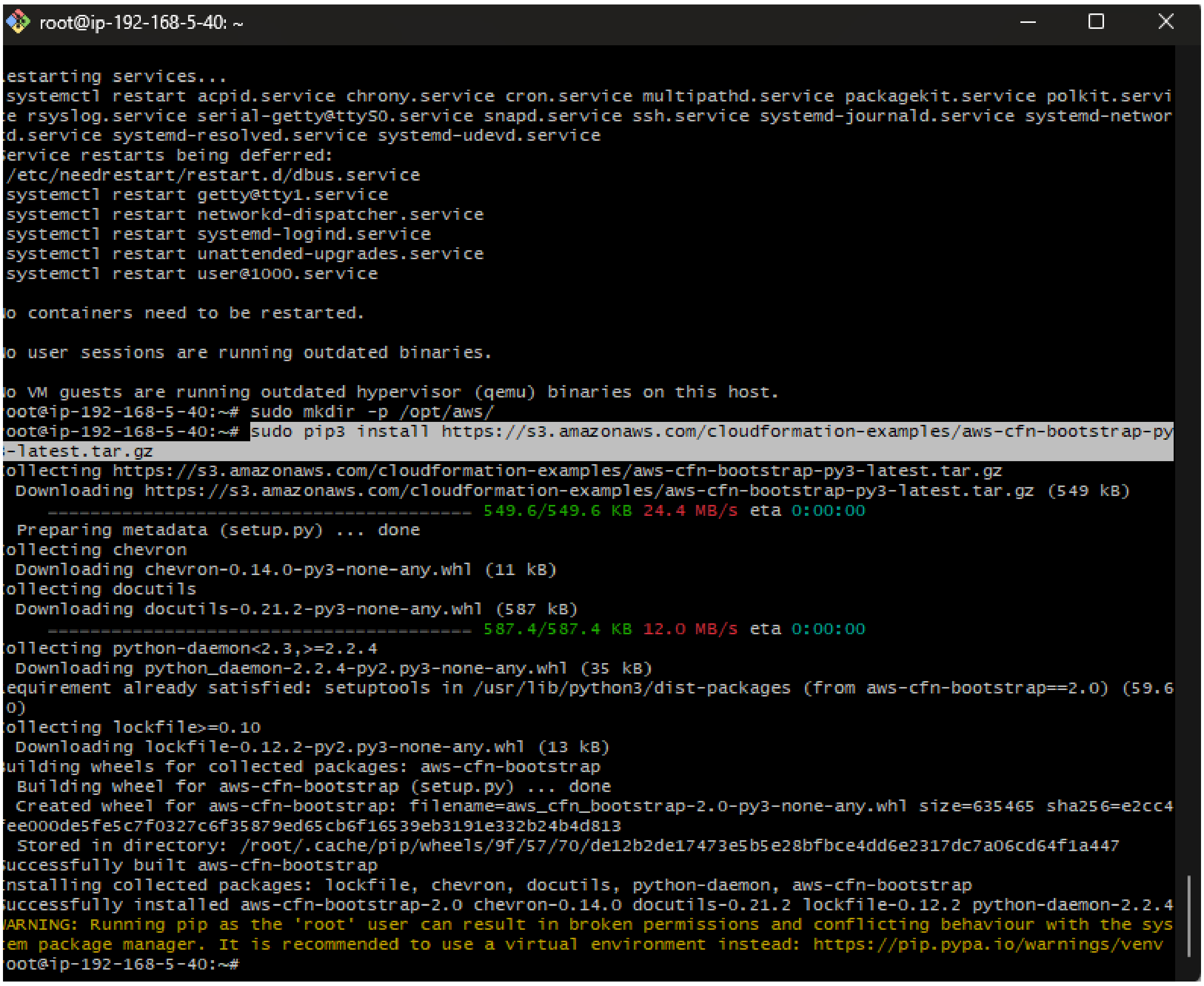


```
Setting up libgcc-11-dev:amd64 (11.4.0-1ubuntu1~22.04) ...
Setting up gcc-11 (11.4.0-1ubuntu1~22.04) ...
Setting up cpp (4:11.2.0-1ubuntu1) ...
Setting up libc6-dev:amd64 (2.35-0ubuntu3.8) ...
Setting up libtiff5:amd64 (4.3.0-6ubuntu0.8) ...
Setting up libfontconfig1:amd64 (2.13.1-4.2ubuntu5) ...
Setting up gcc (4:11.2.0-1ubuntu1) ...
Setting up libexpat1-dev:amd64 (2.4.7-1ubuntu0.3) ...
Setting up libgd3:amd64 (2.3.0-2ubuntu2) ...
Setting up libstdc++-11-dev:amd64 (11.4.0-1ubuntu1~22.04) ...
Setting up zlib1g-dev:amd64 (1:1.2.11.dfsg-2ubuntu9.2) ...
Setting up libc-devtools (2.35-0ubuntu3.8) ...
Setting up g++-11 (11.4.0-1ubuntu1~22.04) ...
Setting up libpython3.10-dev:amd64 (3.10.12-1~22.04.3) ...
Setting up python3.10-dev (3.10.12-1~22.04.3) ...
Setting up g++ (4:11.2.0-1ubuntu1) ...
update-alternatives: using /usr/bin/g++ to provide /usr/bin/c++ (c++) in auto mode
Setting up build-essential (12.9ubuntu3) ...
Setting up libpython3-dev:amd64 (3.10.6-1~22.04) ...
Setting up python3-dev (3.10.6-1~22.04) ...
Processing triggers for man-db (2.10.2-1) ...
Processing triggers for libc-bin (2.35-0ubuntu3.6) ...
Scanning processes...
Scanning candidates...
Scanning linux images...

Running kernel seems to be up-to-date.

Restarting services...
 systemctl restart acpid.service chrony.service cron.service multipathd.service packagekit.service polkit.servi
ce rsyslog.service serial-getty@ttyS0.service snapd.service ssh.service systemd-journald.service systemd-networ
kd.service systemd-resolved.service systemd-udevd.service
Service restarts being deferred:
 /etc/needrestart/restart.d/dbus.service
 systemctl restart getty@tty1.service
 systemctl restart networkd-dispatcher.service
 systemctl restart systemd-logind.service
 systemctl restart unattended-upgrades.service
 systemctl restart user@1000.service

No containers need to be restarted.

No user sessions are running outdated binaries.

No VM guests are running outdated hypervisor (qemu) binaries on this host.
root@ip-192-168-5-40:~# sudo mkdir -p /opt/aws/
root@ip-192-168-5-40:~#
```

6. **Install CloudFormation Helper Scripts:**

- Use pip to install the CloudFormation helper scripts.

sudo pip3 install https://s3.amazonaws.com/cloudformation-examples/aws-cfn-bootstrap-py3-latest.tar.gz

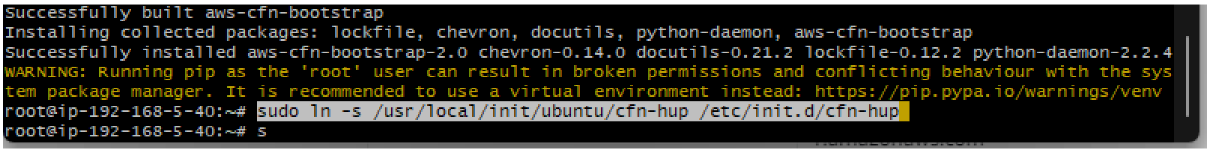


```
estarting services...
systemctl restart acpid.service chrony.service cron.service multipathd.service packagekit.service polkit.servi
e rsyslog.service serial-getty@ttyS0.service snapd.service ssh.service systemd-journald.service systemd-networ
d.service systemd-resolved.service systemd-udevd.service
ervice restarts being deferred:
/etc/needrestart/restart.d/dbus.service
systemctl restart getty@tty1.service
systemctl restart networkd-dispatcher.service
systemctl restart systemd-logind.service
systemctl restart unattended-upgrades.service
systemctl restart user@1000.service

o containers need to be restarted.

o user sessions are running outdated binaries.

o VM guests are running outdated hypervisor (qemu) binaries on this host.
oot@ip-192-168-5-40:~# sudo mkdir -p /opt/aws/
oot@ip-192-168-5-40:~# sudo pip3 install https://s3.amazonaws.com/cloudformation-examples/aws-cfn-bootstrap-py
-latest.tar.gz
ollecting https://s3.amazonaws.com/cloudformation-examples/aws-cfn-bootstrap-py3-latest.tar.gz
 Downloading https://s3.amazonaws.com/cloudformation-examples/aws-cfn-bootstrap-py3-latest.tar.gz (549 kB)
    ---------------------------------------- 549.6/549.6 KB 24.4 MB/s eta 0:00:00
 Preparing metadata (setup.py) ... done
ollecting chevron
 Downloading chevron-0.14.0-py3-none-any.whl (11 kB)
ollecting docutils
 Downloading docutils-0.21.2-py3-none-any.whl (587 kB)
    ---------------------------------------- 587.4/587.4 KB 12.0 MB/s eta 0:00:00
ollecting python-daemon<2.3,>=2.2.4
 Downloading python_daemon-2.2.4-py2.py3-none-any.whl (35 kB)
equirement already satisfied: setuptools in /usr/lib/python3/dist-packages (from aws-cfn-bootstrap==2.0) (59.6
0)
ollecting lockfile>=0.10
 Downloading lockfile-0.12.2-py2.py3-none-any.whl (13 kB)
uilding wheels for collected packages: aws-cfn-bootstrap
 Building wheel for aws-cfn-bootstrap (setup.py) ... done
 Created wheel for aws-cfn-bootstrap: filename=aws_cfn_bootstrap-2.0-py3-none-any.whl size=635465 sha256=e2cc4
ee000de5fe5c7f0327c6f35879ed65cb6f16539eb3191e332b24b4d813
 Stored in directory: /root/.cache/pip/wheels/9f/57/70/de12b2de17473e5b5e28bfbce4dd6e2317dc7a06cd64f1a447
uccessfully built aws-cfn-bootstrap
nstalling collected packages: lockfile, chevron, docutils, python-daemon, aws-cfn-bootstrap
uccessfully installed aws-cfn-bootstrap-2.0 chevron-0.14.0 docutils-0.21.2 lockfile-0.12.2 python-daemon-2.2.4
ARNING: Running pip as the 'root' user can result in broken permissions and conflicting behaviour with the sys
em package manager. It is recommended to use a virtual environment instead: https://pip.pypa.io/warnings/venv
oot@ip-192-168-5-40:~#
```

7. **Create Symbolic Link:**

- Create a symbolic link for the CloudFormation helper scripts to ensure they are accessible.

sudo ln -s /usr/local/init/ubuntu/cfn-hup /etc/init.d/cfn-hup

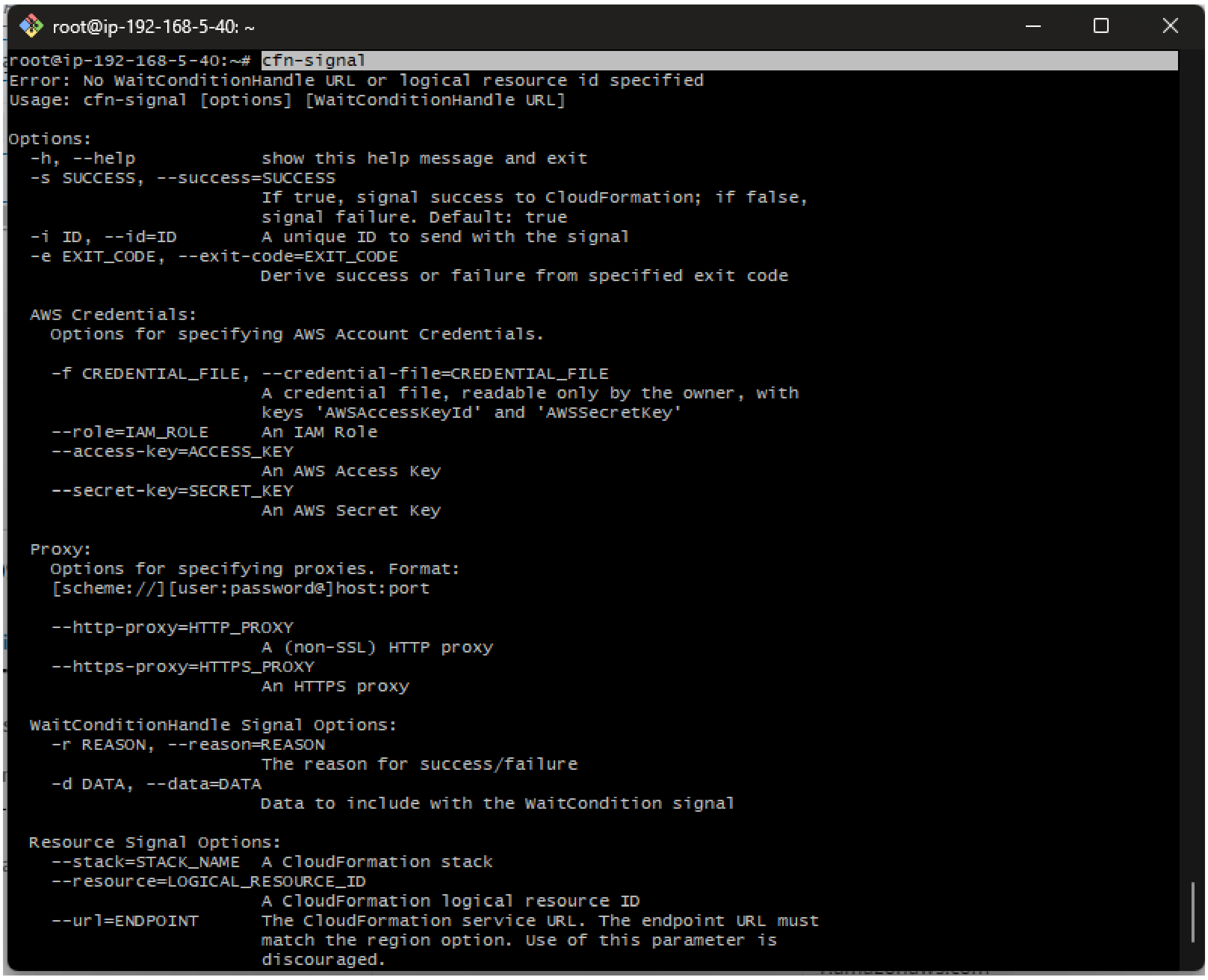


```
Successfully built aws-cfn-bootstrap
Installing collected packages: lockfile, chevron, docutils, python-daemon, aws-cfn-bootstrap
Successfully installed aws-cfn-bootstrap-2.0 chevron-0.14.0 docutils-0.21.2 lockfile-0.12.2 python-daemon-2.2.4
WARNING: Running pip as the 'root' user can result in broken permissions and conflicting behaviour with the sys
tem package manager. It is recommended to use a virtual environment instead: https://pip.pypa.io/warnings/venv
root@ip-192-168-5-40:~# sudo ln -s /usr/local/init/ubuntu/cfn-hup /etc/init.d/cfn-hup
root@ip-192-168-5-40:~# s
```

8. **Verification:**

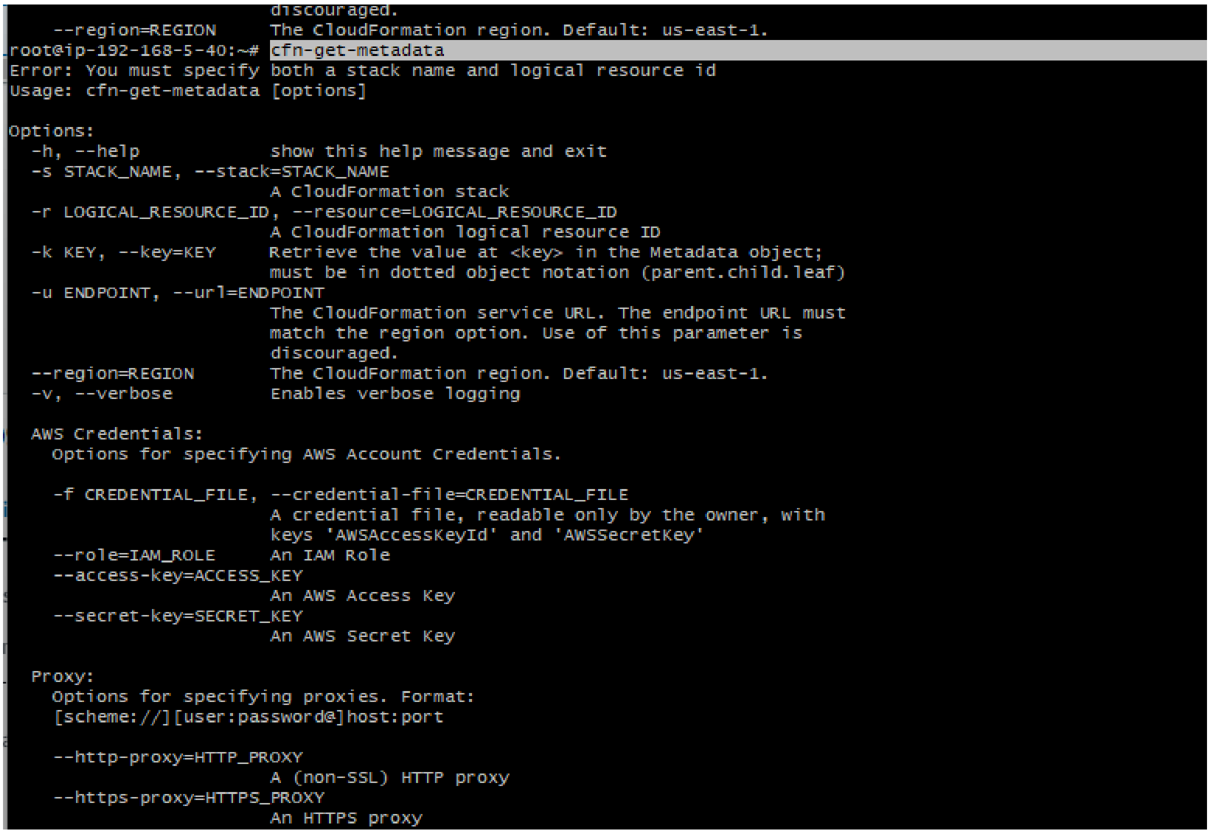
- To verify that the installation was successful, try running the following commands:

cfn-signal



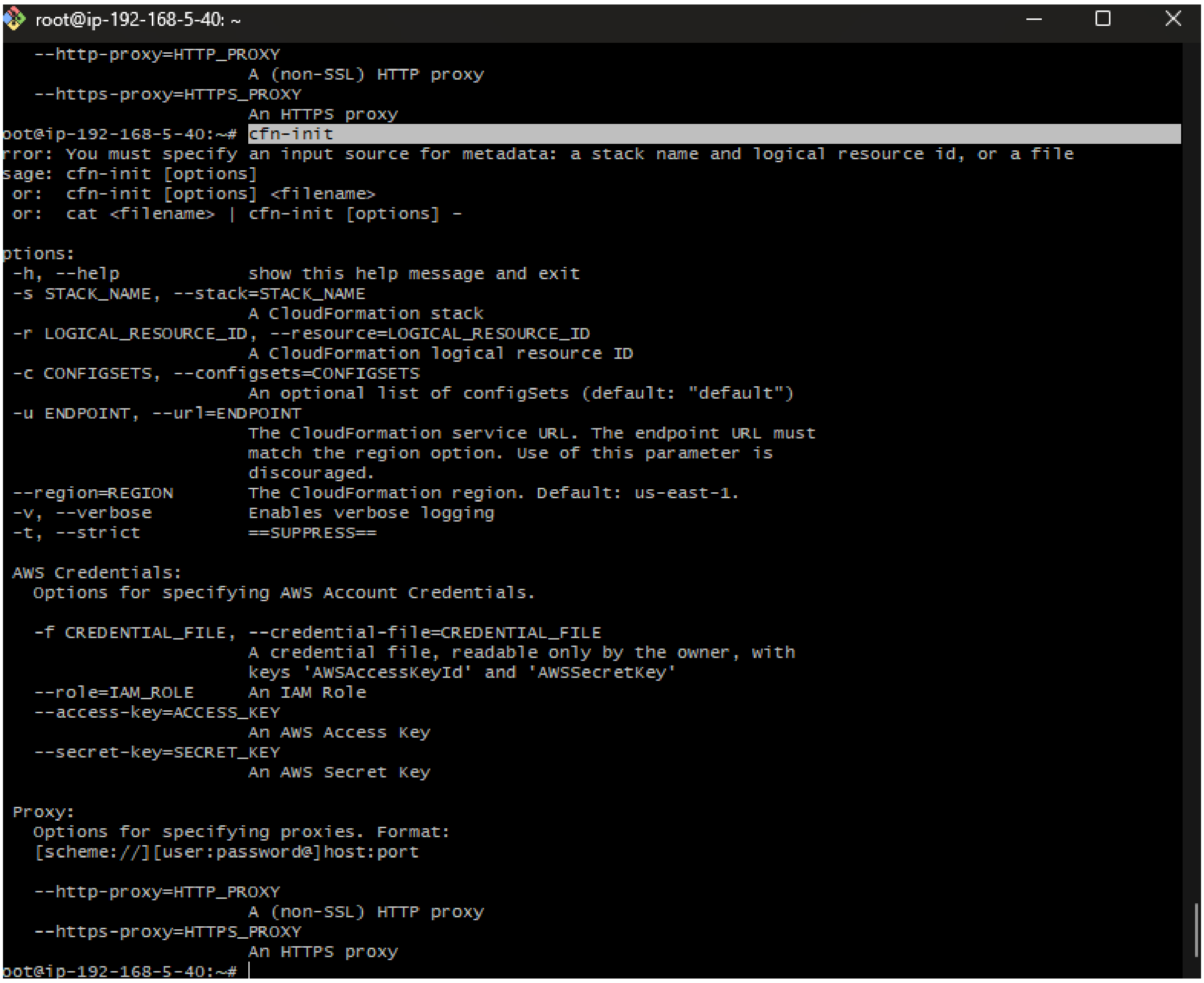
```
root@ip-192-168-5-40:~# cfn-signal
Error: No WaitConditionHandle URL or logical resource id specified
Usage: cfn-signal [options] [WaitConditionHandle URL]

Options:
  -h, --help            show this help message and exit
  -s SUCCESS, --success=SUCCESS
                        If true, signal success to CloudFormation; if false,
                        signal failure. Default: true
  -i ID, --id=ID        A unique ID to send with the signal
  -e EXIT_CODE, --exit-code=EXIT_CODE
                        Derive success or failure from specified exit code

  AWS Credentials:
    Options for specifying AWS Account Credentials.

    -f CREDENTIAL_FILE, --credential-file=CREDENTIAL_FILE
                        A credential file, readable only by the owner, with
                        keys 'AWSAccessKeyId' and 'AWSSecretKey'
    --role=IAM_ROLE     An IAM Role
    --access-key=ACCESS_KEY
                        An AWS Access Key
    --secret-key=SECRET_KEY
                        An AWS Secret Key

  Proxy:
    Options for specifying proxies. Format:
    [scheme://][user:password@]host:port

    --http-proxy=HTTP_PROXY
                        A (non-SSL) HTTP proxy
    --https-proxy=HTTPS_PROXY
                        An HTTPS proxy

  WaitConditionHandle Signal Options:
    -r REASON, --reason=REASON
                        The reason for success/failure
    -d DATA, --data=DATA
                        Data to include with the WaitCondition signal

  Resource Signal Options:
    --stack=STACK_NAME  A CloudFormation stack
    --resource=LOGICAL_RESOURCE_ID
                        A CloudFormation logical resource ID
    --url=ENDPOINT      The CloudFormation service URL. The endpoint URL must
                        match the region option. Use of this parameter is
                        discouraged.
```

cfn-get-metadata



```
                        discouraged.
    --region=REGION     The CloudFormation region. Default: us-east-1.
root@ip-192-168-5-40:~# cfn-get-metadata
Error: You must specify both a stack name and logical resource id
Usage: cfn-get-metadata [options]

Options:
  -h, --help            show this help message and exit
  -s STACK_NAME, --stack=STACK_NAME
                        A CloudFormation stack
  -r LOGICAL_RESOURCE_ID, --resource=LOGICAL_RESOURCE_ID
                        A CloudFormation logical resource ID
  -k KEY, --key=KEY     Retrieve the value at <key> in the Metadata object;
                        must be in dotted object notation (parent.child.leaf)
  -u ENDPOINT, --url=ENDPOINT
                        The CloudFormation service URL. The endpoint URL must
                        match the region option. Use of this parameter is
                        discouraged.
  --region=REGION       The CloudFormation region. Default: us-east-1.
  -v, --verbose         Enables verbose logging

  AWS Credentials:
    Options for specifying AWS Account Credentials.

    -f CREDENTIAL_FILE, --credential-file=CREDENTIAL_FILE
                        A credential file, readable only by the owner, with
                        keys 'AWSAccessKeyId' and 'AWSSecretKey'
    --role=IAM_ROLE     An IAM Role
    --access-key=ACCESS_KEY
                        An AWS Access Key
    --secret-key=SECRET_KEY
                        An AWS Secret Key

  Proxy:
    Options for specifying proxies. Format:
    [scheme://][user:password@]host:port

    --http-proxy=HTTP_PROXY
                        A (non-SSL) HTTP proxy
    --https-proxy=HTTPS_PROXY
                        An HTTPS proxy
```

cfn-init



```
  --http-proxy=HTTP_PROXY
                        A (non-SSL) HTTP proxy
  --https-proxy=HTTPS_PROXY
                        An HTTPS proxy
oot@ip-192-168-5-40:~# cfn-init
rror: You must specify an input source for metadata: a stack name and logical resource id, or a file
sage: cfn-init [options]
 or:  cfn-init [options] <filename>
 or:  cat <filename> | cfn-init [options] -

ptions:
 -h, --help            show this help message and exit
 -s STACK_NAME, --stack=STACK_NAME
                       A CloudFormation stack
 -r LOGICAL_RESOURCE_ID, --resource=LOGICAL_RESOURCE_ID
                       A CloudFormation logical resource ID
 -c CONFIGSETS, --configsets=CONFIGSETS
                       An optional list of configSets (default: "default")
 -u ENDPOINT, --url=ENDPOINT
                       The CloudFormation service URL. The endpoint URL must
                       match the region option. Use of this parameter is
                       discouraged.
 --region=REGION       The CloudFormation region. Default: us-east-1.
 -v, --verbose         Enables verbose logging
 -t, --strict          ==SUPPRESS==

 AWS Credentials:
   Options for specifying AWS Account Credentials.

   -f CREDENTIAL_FILE, --credential-file=CREDENTIAL_FILE
                       A credential file, readable only by the owner, with
                       keys 'AWSAccessKeyId' and 'AWSSecretKey'
   --role=IAM_ROLE     An IAM Role
   --access-key=ACCESS_KEY
                       An AWS Access Key
   --secret-key=SECRET_KEY
                       An AWS Secret Key

 Proxy:
   Options for specifying proxies. Format:
   [scheme://][user:password@]host:port

   --http-proxy=HTTP_PROXY
                       A (non-SSL) HTTP proxy
   --https-proxy=HTTPS_PROXY
                       An HTTPS proxy
oot@ip-192-168-5-40:~# |
```

That's it. Now you can fully leverage the capabilities of your Ubuntu instance installed with CloudFormation helper scripts within your CloudFormation templates. One common use case is creating an Amazon Machine Image (AMI) out of this instance, which you can then utilize in your EC2 resource declarations. This ensures consistency and repeatability in your deployments.

Alternatively, you can integrate the manual steps we've performed here into your EC2 resource using User Data. By embedding these commands directly into your EC2 instance's User Data section, you can automate the setup process entirely within your CloudFormation template. This approach streamlines deployment and reduces manual intervention, enhancing the efficiency of your infrastructure management.

Whether you choose to create an AMI or utilize User Data, the installation of CloudFormation helper scripts empowers you to automate tasks, configure resources dynamically, and maintain a standardized deployment process across your AWS environment.

One last thing! It is a good practice to clean up the resources created during this lab. Not only will it make you a better professional, but you will also become a more organized person. Happy learning!