

If you have selected Ubuntu as the AMI when creating the EC2 instances, follow these two commands to install the AWS CLI and Git:

```
ubuntu@ip-172-31-40-176:~$ sudo snap install aws-cli
error: This revision of snap "aws-cli" was published using classic confinement and thus may perform
arbitrary system changes outside of the security sandbox that snaps are usually confined to,
which may put your system at risk.

If you understand and want to proceed repeat the command including --classic.
ubuntu@ip-172-31-40-176:~$ sudo snap install aws-cli --classic
aws-cli (v2/stable) 2.17.4 from Amazon Web Services (aws/) installed
ubuntu@ip-172-31-40-176:~$ aws s3 ls s3://amanullahbucket
2024-06-28 10:54:31      696880 2s.png
ubuntu@ip-172-31-40-176:~$ sudo apt install git
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
git is already the newest version (1:2.43.0-1ubuntu7).
git set to manually installed.
0 upgraded, 0 newly installed, 0 to remove and 0 not upgraded.
```

Or If you have selected Linux as the AMI when creating the EC2 instances, follow these two commands to install the AWS CLI and Git:

The first screenshot shows a terminal window on an Amazon Linux instance. It displays a warning about a newer release of Amazon Linux being available and provides instructions to upgrade using the 'dnf upgrade' command. The terminal output shows the command being executed and the dependencies being resolved.

The second screenshot shows a terminal window on the same Amazon Linux instance, displaying the output of the 'dnf install git' command. It lists the packages to be installed, their architectures, versions, repositories, and sizes. The output also shows the transaction summary, including the total download size and the packages being installed.

Package	Architecture	Version	Repository	Size
git	x86_64	2.40.1-1.amzn2023.0.3	amazonlinux	54 k
git-core	x86_64	2.40.1-1.amzn2023.0.3	amazonlinux	4.3 M
git-core-doc	noarch	2.40.1-1.amzn2023.0.3	amazonlinux	2.6 M
perl-Error	noarch	1:0.17029-5.amzn2023.0.2	amazonlinux	41 k
perl-File-Find	noarch	1.37-477.amzn2023.0.6	amazonlinux	26 k
perl-Git	noarch	2.40.1-1.amzn2023.0.3	amazonlinux	42 k
perl-TermReadKey	x86_64	2.38-9.amzn2023.0.2	amazonlinux	36 k
perl-lib	x86_64	0.65-477.amzn2023.0.6	amazonlinux	15 k

Login to GitHub using the CLI.

```
ubuntu@ip-172-31-40-176:~$ git config --global user.name"AmanWasti9"
ubuntu@ip-172-31-40-176:~$ git config --global user.email wastiaman123@gmail.com
```

Clone the repository you want from your GitHub using the `git clone` command, and then navigate to the repository using `cd`.

```
ubuntu@ip-172-31-40-176:~$ git clone https://github.com/AmanWasti9/AWS-Practice
Cloning into 'AWS-Practice'...
remote: Enumerating objects: 3, done.
remote: Counting objects: 100% (3/3), done.
remote: Total 3 (delta 0), reused 0 (delta 0), pack-reused 0
Receiving objects: 100% (3/3), done.
ubuntu@ip-172-31-40-176:~$ ls
AWS-Practice  snap
ubuntu@ip-172-31-40-176:~$ cd
ubuntu@ip-172-31-40-176:~$ cd AWS-Practice
```

Then download the file from your S3 bucket into your repository.

```
ubuntu@ip-172-31-40-176:~/AWS-Practice$ aws s3 cp s3://amanullahbucket . --recursive
download: s3://amanullahbucket/2s.png to ./2s.png
ubuntu@ip-172-31-40-176:~/AWS-Practice$ LS
LS: command not found
ubuntu@ip-172-31-40-176:~/AWS-Practice$ ls
2s.png  README.md
```

Open GitHub, then go to the Settings page. There, you will find 'Developer settings.' Click on that.

The screenshot shows the GitHub Settings page with the left sidebar containing various settings categories. The 'Developer settings' option is selected and highlighted. The main content area displays several optional settings that can be configured or linked. These include an ORCID iD field, a 'Connect your ORCID iD' button, a 'Social accounts' section with four 'Link to social profile' buttons, a 'Company' section with a text input field and a note about mentioning the company's GitHub organization, a 'Location' section with a text input field, and a checkbox for 'Display current local time' with a note about time differences. At the bottom, a disclaimer states that all fields are optional and can be deleted at any time.

Repositories

Codespaces

Packages

Copilot

Pages

Saved replies

Security

Code security and analysis

Integrations

Applications

Scheduled reminders

Archives

Security log

Sponsorship log

<> Developer settings

ORCID iD

ORCID provides a persistent identifier - an ORCID iD - that distinguishes you from other researchers. Learn more at [ORCID.org](https://orcid.org).

[Connect your ORCID iD](#)

Social accounts

[Link to social profile](#)

[Link to social profile](#)

[Link to social profile](#)

[Link to social profile](#)

Company

You can @mention your company's GitHub organization to link it.

Location

☐ Display current local time

Other users will see the time difference from their local time.

All of the fields on this page are optional and can be deleted at any time, and by filling them out, you're giving us

Click on 'Personal access tokens,' then click on 'Tokens (classic).'

GitHub Apps

OAuth Apps

Personal access tokens

Fine-grained tokens

Tokens (classic)

GitHub Apps

New GitHub App

Want to build something that integrates with and extends GitHub? [Register a new GitHub App](#) to get started developing on the GitHub API. You can also read more about building GitHub Apps in our [developer documentation](#).

Then generate new token classic

GitHub Apps

OAuth Apps

Personal access tokens

Fine-grained tokens

Tokens (classic)

Personal access tokens (classic)

Generate new token ▼ Revoke all

Tokens you have generated that can be used to access the [GitHub API](#).

Amantoken — admin:org, admin:org_hook, admin:public_key, admin:repo_hook, admin:ssh_signing_key, repo, user

Last used within the last week Delete

Expires on Mon, Jul 29 2024.

Personal access tokens (classic) function like ordinary OAuth access tokens. They can be used instead of a password for Git over HTTPS, or can be

Save that token, as it will be used later for `git push`.

GIT PUSH

Follow these commands, and when prompted for a password, use the token instead.

```
ubuntu@ip-172-31-40-176:~/AWS-Practice$ git push
Username for 'https://github.com': AmanWasti9
Password for 'https://AmanWasti9@github.com':
Everything up-to-date
ubuntu@ip-172-31-40-176:~/AWS-Practice$ ls
2s.png README.md
ubuntu@ip-172-31-40-176:~/AWS-Practice$ git add .
ubuntu@ip-172-31-40-176:~/AWS-Practice$ git commit -m "First commit aws"
[main ef57cbd] First commit aws
1 file changed, 0 insertions(+), 0 deletions(-)
create mode 100644 2s.png
ubuntu@ip-172-31-40-176:~/AWS-Practice$ git branch -M main
ubuntu@ip-172-31-40-176:~/AWS-Practice$ git remote add origin https://github.com/AmanWasti9/AWS-Practice.git
error: remote origin already exists.
ubuntu@ip-172-31-40-176:~/AWS-Practice$ git push -u origin main
Enumerating objects: 4, done.
Counting objects: 100% (4/4), done.
Delta compression using up to 2 threads
Compressing objects: 100% (3/3), done.
Writing objects: 100% (3/3), 671.37 KiB | 13.99 MiB/s, done.
```

GIT PULL

First add any file from your GitHub repository to this local repository, and then run the following commands.

```
ubuntu@ip-172-31-40-176:~/AWS-Practice$ git pull
remote: Enumerating objects: 4, done.
remote: Counting objects: 100% (4/4), done.
remote: Compressing objects: 100% (3/3), done.
remote: Total 3 (delta 0), reused 0 (delta 0), pack-reused 0
Unpacking objects: 100% (3/3), 499.98 KiB | 3.07 MiB/s, done.
From https://github.com/AmanWasti9/AWS-Practice
   ef57cbd..ba700d6  main       -> origin/main
Updating ef57cbd..ba700d6
Fast-forward
 3p.png | Bin 0 -> 520318 bytes
 1 file changed, 0 insertions(+), 0 deletions(-)
 create mode 100644 3p.png
ubuntu@ip-172-31-40-176:~/AWS-Practice$ ls
2s.png 3p.png README.md
ubuntu@ip-172-31-40-176:~/AWS-Practice$
```