Git Commands -By Amar Palwankar

1. Install Git (if not already installed)

Check if Git is installed:

```
git --version
```

Expected Output: git version 2.45.0 (or similar)

If Git is not installed:

```
sudo dnf install git -y
```

Expected Output:

```
Last metadata expiration check: 0:00:05 ago...
Dependencies resolved.
Installing:
 git    x86_64    2.45.0-1.fc40
Complete!
```

Step 2: Configure Git

Set your name:

```
git config --global user.name "Your Name"
```

Expected Output: (No output - silent success)

Command:

```
git config --global user.name "Amar Famt"
```

Expected Output:

```
(No output - command succeeds silently)
```

Set Email:

```
git config --global user.email "your-email@example.com"
```

2. Setting Up Your Repository

Option A: Clone Existing Repository

```
cd ~/Documents
git clone https://github.com/amarfamt/your-repo-name.git
cd your-repo-name
```

Expected Output:

```
Cloning into 'your-repo-name'...
remote: Enumerating objects: 5, done.
```

```
remote: Counting objects: 100% (5/5), done.
remote: Total 5 (delta 0), reused 0 (delta 0), pack-reused 0
Unpacking objects: 100% (5/5), done.
```

Scenario 2: Initialize New Repository

```
# Create project folder
mkdir my-project
cd my-project

# Initialize git
git init

# Create README file
echo "# My Project" > README.md

# Add remote repository
git remote add origin https://github.com/amarfamt/your-repo-name.git
```

3. Configure Git (First Time Only)

Set your name:

```
git config --global user.name "Your Name"
```

Expected Output: No output (silent success)

Command:

```
git config --global user.email "your.email@example.com"
```

Expected Output: No output (command succeeds silently)

Verify Configuration

```
git config --list
```

Expected Output:

```
user.name=Your Name
user.email=your.email@example.com
...
```

3. Clone Your Repository

Command:

```
git clone https://github.com/amarfamt/repository-name.git
cd repository-name
```

Expected Output:

```
Cloning into 'repository-name'...
remote: Enumerating objects: 10, done.
remote: Counting objects: 100% (10/10), done.
remote: Compressing objects: 100% (8/8), done.
remote: Total 10 (delta 2), reused 10 (delta 2)
```

```
Unpacking objects: 100% (10/10), done.
```

4. Configure Git (First Time Setup)

Set your name

```
git config --global user.name "Your Name"
```

Expected Output: (No output, just completes)

Set your email

```
git config --global user.email "your.email@example.com"
```

Expected Output: (No output, just completes)

Verify configuration

```
git config --list
```

Expected Output:

```
user.name=Your Name
user.email=your.email@example.com
core.editor=nano
...
```

5. GitHub Personal Access Token (Required for Push)

Important: GitHub no longer accepts passwords for Git operations. You must use a Personal Access Token (PAT).

Step 1: Create a GitHub Token

1. Go to GitHub.com and log in
2. Click your profile picture → **Settings**
3. Scroll down to **Developer settings**
4. Click **Personal access tokens → Tokens (classic)**
5. Click **Generate new token → Generate new token (classic)**
6. Give it a name (e.g., "Fedora 40 Desktop")
7. Set expiration (90 days recommended)
8. Select scopes:
    - **repo** (Full control of repositories)
    - **workflow** (if using GitHub Actions)
9. Click **Generate token**
10. **COPY THE TOKEN IMMEDIATELY** - You won't see it again!

Token looks like:

```
ghp_xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
```

Keep this safe - treat it like a password!

Step 2: Using Token for Authentication

Option A: Store Token in Git Credential Helper (Recommended)

```
# Enable credential storage
git config --global credential.helper store

# Now when you push, enter token once - it will be saved
git push origin main
```

First time prompt:

```
Username for 'https://github.com': amarfamt
Password for 'https://amarfamt@github.com': [paste your token here]
```

Token will be saved in `~/.git-credentials` for future use

Option B: Include Token in Remote URL (Less Secure)

```
# Format: https://USERNAME:TOKEN@github.com/USERNAME/REPO.git
git remote set-url origin https://amarfamt:ghp_your_token_here@github.com/amarfamt/your-repo.git

# Verify the URL (token will be visible!)
git remote -v
```

```
origin  https://amarfamt:ghp_xxx...@github.com/amarfamt/your-repo.git (fetch)
origin  https://amarfamt:ghp_xxx...@github.com/amarfamt/your-repo.git (push)
```

6. Basic Git Workflow - Push a File

Complete Example: Create and Push a File

Step 1: Navigate to your repository

```
cd ~/projects/your-repo-name
pwd
```

```
/home/yourusername/projects/your-repo-name
```

Step 2: Create a new file

```
echo "# My Project" > README.md
cat README.md
```

```
# My Project
```

Step 3: Check status

```
git status
```

```
On branch main
Untracked files:
  (use "git add <file>..." to include in what will be committed)
        README.md

nothing added to commit but untracked files present (use "git add" to track)
```

Step 4: Add file to staging

```
git add README.md
git status
```

```
On branch main
Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
```

```
        new file:   README.md
```

Step 5: Commit the file

```
git commit -m "Add README file"
```

```
[main 1a2b3c4] Add README file
 1 file changed, 1 insertion(+)
 create mode 100644 README.md
```

Step 6: Push to GitHub (using token)

```
git push origin main
```

If using credential helper (first time):

```
Username for 'https://github.com': amarfamt
Password for 'https://amarfamt@github.com': [paste token here - no echo]

Enumerating objects: 3, done.
Counting objects: 100% (3/3), done.
Writing objects: 100% (3/3), 242 bytes | 242.00 KiB/s, done.
Total 3 (delta 0), reused 0 (delta 0), pack-reused 0
To https://github.com/amarfamt/your-repo.git
   abc123..def456  main -> main
```

If token is already stored:

```
Enumerating objects: 3, done.
Counting objects: 100% (3/3), done.
Writing objects: 100% (3/3), 242 bytes | 242.00 KiB/s, done.
Total 3 (delta 0), reused 0 (delta 0), pack-reused 0
To https://github.com/amarfamt/your-repo.git
   abc123..def456  main -> main
```

9. Quick Start Checklist

Complete Setup (First Time)

```
# 1. Update system
sudo dnf update -y

# 2. Install git
sudo dnf install git -y

# 3. Configure git
git config --global user.name "Your Name"
git config --global user.email "your@email.com"
git config --global credential.helper store

# 4. Create GitHub token (via GitHub website)
# Settings → Developer settings → Personal access tokens

# 5. Clone your repository
git clone https://github.com/amarfamt/your-repo.git
cd your-repo
```