

Using GIT on DISCOVER

- Make sure git version is recent, say ≥ 1.8
 - e.g. on discover: `module load git/2.24.0`
- Customize your git environment. Three places:
 - System `/etc/gitconfig` (need admin privileges)
 - User `$HOME/.gitconfig`
 - Project `my_project/.git/config`

Git customization

Git commands to edit configuration:

```
git config --system (system)
```

```
git config --global (user)
```

```
git config (project)
```

You should do – at least – the following:

```
$ git config --global user.name "[name]"
```

Sets the name you want attached to your commit transactions

```
$ git config --global user.email "[email address]"
```

Sets the email you want attached to your commit transactions

```
$ git config --global core.editor "vim"
```

Sets the editor you want attached to your commit transactions

```
$ git config --global color.ui auto
```

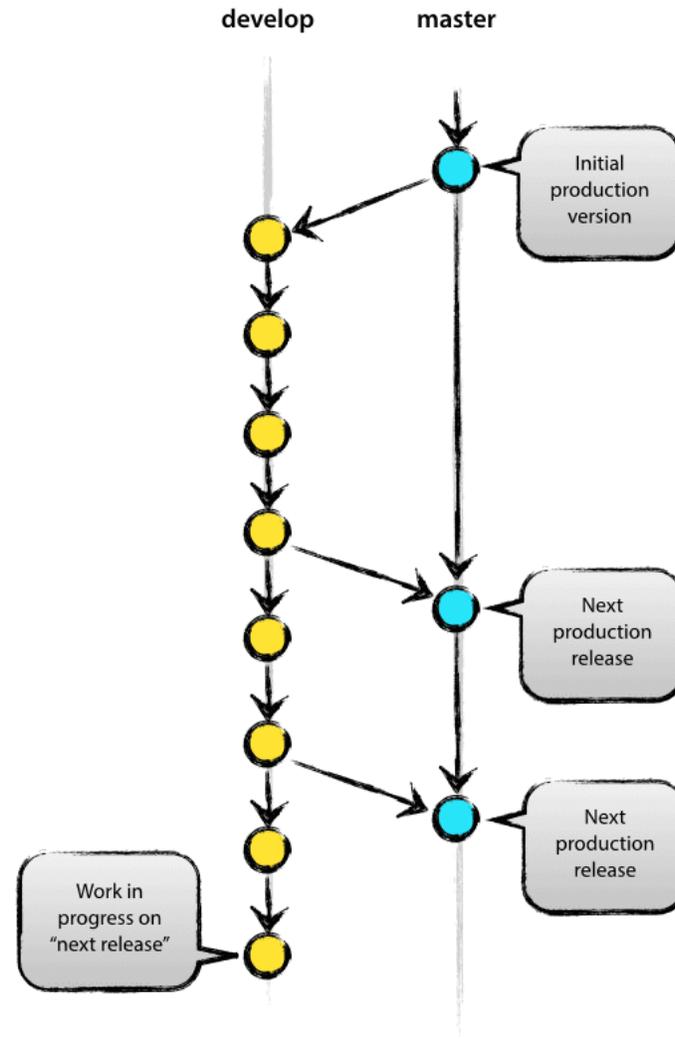
Enables helpful colorization of command line output

NU-WRF Git repository URL

- <https://git.mysmce.com/astg/nuwrf/nu-wrf-dev>
 - Note: This is not a public repo. To get access contact carlos.a.cruz@nasa.gov
- Clone repository
 - `git clone https://git.mysmce.com/astg/nuwrf/nu-wrf-dev`
 - List all branches:
 - `git branch -a`
 - In NU-WRF there are two branches: master and develop
 - All work should be done on `develop` branch or better yet, on a `feature` branch (see next page).

Git workflow

- In your project directory:
 1. Checkout develop branch
`git checkout develop`
 2. Create feature branch
`git branch my-feature-branch`
`git checkout my-feature-branch`
 3. Create, stage and commit a new file
`git add my-new-file`
`git commit my-new-file`
 4. When done with feature
`git checkout develop`
`git merge --no-ff my-feature-branch`
 5. Remove feature branch
`git branch -d my-feature-branch`
 6. Update (push) remote repository
`git push origin develop`



Common commands

Create repositories

\$ git init [project-name]

Creates a new local repository with the specified name

\$ git clone [url]

Downloads a project and its entire version history

Make changes

\$ git status

Lists all new or modified files to be committed

\$ git add [file]

Snapshots the file in preparation for versioning

\$ git reset [file]

Unstages the file, but preserve its contents

\$ git diff

Shows file differences not yet staged

\$ git diff --staged

Shows file differences between staging and the last file version

\$ git commit -m "[descriptive message]"

Records file snapshots permanently in version history

Group changes

\$ git branch

Lists all local branches in the current repository

\$ git branch [branch-name]

Creates a new branch

\$ git checkout [branch-name]

Switches to the specified branch and updates the working directory

\$ git merge [branch]

Combines the specified branch's history into the current branch

\$ git branch -d [branch-name]

Deletes the specified branch

Review history

\$ git log

Lists version history for the current branch

\$ git log --follow [file]

Lists version history for a file, including renames

\$ git diff [first-branch]...[second-branch]

Shows content differences between two branches

\$ git show [commit]

Outputs metadata and content changes of the specified commit