



GitLab

Lara Ianov, Ph.D.

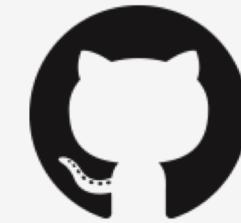
Neurodevelopmental Bioinformatics Initiative (NBI)
Civitan International Research Center
University of Alabama at Birmingham

Git



- Version control software
 - Tracks changes among all members
 - Maintains a history of all changes in an organized way and allows one to revert changes
 - A higher order method over the standard "copy and change name method": copy a file with changes to it and name it differently or place it in time-stamped directories - usually these scripts are filled with unnecessary comments documenting the changes
 - A git repository contains the full history of the project. Each clone will always mirror the full repository. Thus, git is a Distributed Version Control System

GitLab, GitHub, Bitbucket etc.



- Web-based services which can be used to host your git repositories in a streamlined manner. Here are some differences among them:

Free Plans	Public Repos	Private Repos	Collaborators	Storage Space	Hosting	Support
GitHub Public	Unlimited	* 0	Unlimited	N / A	Cloud	Email / Forum
Bitbucket Small teams	Unlimited	Unlimited (1Gb /project)	5	N / A	Cloud	Email / Forum
GitLab Cloud Hosted	Unlimited	Unlimited (10Gb / Project)	Unlimited	Unlimited	Cloud	Forum
GitLab Community Edition	Unlimited	Unlimited	Unlimited	N / A	Self-hosted	Forum

UAB RC hosts

* As a student (with .edu account), you can have private repos for free with access to "student developer plan". Details at: <https://help.github.com/articles/applying-for-a-student-developer-pack/>

Table source:

<https://medium.com/flow-ci/github-vs-bitbucket-vs-gitlab-vs-coding-7cf2b43888a1>

GitLab

- Sign-in with Blazer ID and password: https://gitlab.rc.uab.edu/users/sign_in
- Learning sources:
 - Git for beginners from UAB RC: https://docs.uabgrid.uab.edu/wiki/Git_For_Beginners
 - GitLab tutorial: <https://www.tutorialspoint.com/gitlab/index.htm>
 - GitLab docs: <https://docs.gitlab.com/>
 - Google ☺
- **Today's focus:** GitLab's graphical interface (with some emphasis on important commands)
- Basic git terminology:
 - Clone: copy a repository (repo)
 - Commit: record changes to the repo
 - Push: updates remote git repo with committed changes (if working outside of GitLab environment)
 - Branch: independent area of development from the primary set/branch
 - ... and more

GitLab groups



CIRC_NBI 

This group contains the NBI hosted pipelines and supplementary scripts

Global 

Filter by name... 

Last created  New project 

Labs  Lab specific scripts

  Matt_Alexander  Script(s) by the NBI in support of Dr. Matt Alexander's lab  0  1 week ago

  NBI_standard_RNA-seq_pipeline  One out of two RNA-seq pipelines offered by the NBI. Its name contain "standard" as it uses the common alignment meth...  1  2 months ago

  NBI_quasi_mapping_RNA-seq_pipeline  One out of two RNA-seq pipelines offered by the NBI. This pipeline utilizes the pseudoaligner, Salmon, for quasi-mapping.  1  2 months ago

  File_input_rename  Often times, cores provide raw data files in a manner which is not optimal as input for the NBI pipelines. This set of script r...  0  6 days ago

  DESeq2_sample_scripts  Sample scripts for RNA-seq statistical analysis  0  6 days ago

  NBI_RRBS_SE_pipeline  RRBS single-end pipeline.  1  2 months ago

 0  1  1  

N

NBI_standard_RNA-seq_pipeline

One out of two RNA-seq pipelines offered by the NBI. Its name contains "standard" as it uses the common alignment method (STAR), rather than pseudoaligners.

Project ID: 446

Unstar 1 Fork 0 SSH git@gitlab.rc.uab.edu:CIRC_NBI Global

[Leave project](#)

Files (389 KB) Commits (35) Branch (1) Tags (11) Readme

[Add Changelog](#)

[Add License](#)

[Add Contribution guide](#)

[Add Kubernetes cluster](#)

[Set up CI/CD](#)



Auto DevOps

It will automatically build, test, and deploy your application based on a predefined CI/CD configuration.

[Learn more in the Auto DevOps documentation](#)

[Enable in settings](#)

SHA-1 hash is essential to git's
method of version control

master

NBI_standard_RNA-seq_pipeline

+

History

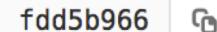
Find file

Web IDE



Updated dryrun to run from scratch in case of rerun Snakemake_submit_NBI.sh
Lara Ianov authored 2 months ago

fdd5b966



Access is given to specific users which have permission to see the pipeline, read the docs and clone to run the analysis at HPC.

They may also open "issues" with requests to add specific features etc.

Project members

You can add a new member to **NBI_standard_RNA-seq_pipeline** or share it with another group.

<p>Add member</p> <p>Select members to invite</p> <p>Search for members to update or invite</p> <p>Choose a role permission</p> <p>Guest</p> <p>Read more about role permissions</p> <p>Access expiration date</p> <p>Expiration date</p> <p>Add to project Import</p>	<p>Share with group</p>
--	-------------------------

Existing members and groups

Members of NBI_standard_RNA-seq_pipeline 7		Find existing members by name	Name, ascending
 Farah Lubin @flubin	Given access 2 months ago	Reporter	<input type="button" value="Delete"/>
 Jeremy Day @jjday	Given access 4 months ago	Reporter	<input type="button" value="Delete"/>
 Katherine Savell @savell	Given access 3 months ago	Reporter	<input type="button" value="Delete"/>
 Lara Ianov @lianov	It's you Given access 4 months ago	Maintainer	<input type="button" value="Leave"/>
 Rebecca M Hauser @rmhauser	Given access 2 months ago	Reporter	<input type="button" value="Delete"/>

See more member permissions at: <https://docs.gitlab.com/ee/user/permissions.html>

Action	Guest	Reporter	Developer	Maintainer	Owner
Create new issue	✓ ¹	✓	✓	✓	✓
Create confidential issue	✓ ¹	✓	✓	✓	✓
View confidential issues	(✓) ²	✓	✓	✓	✓
Leave comments	✓ ¹	✓	✓	✓	✓
See related issues	✓	✓	✓	✓	✓
See a list of jobs	✓ ³	✓	✓	✓	✓
See a job log	✓ ³	✓	✓	✓	✓
Download and browse job artifacts	✓ ³	✓	✓	✓	✓
View wiki pages	✓ ¹	✓	✓	✓	✓
View license management reports ?	✓ ¹	✓	✓	✓	✓
View Security reports ?	✓ ¹	✓	✓	✓	✓
Pull project code	1	✓	✓	✓	✓
Download project	1	✓	✓	✓	✓
Assign issues		✓	✓	✓	✓
Assign merge requests			✓	✓	✓
Label issues		✓	✓	✓	✓
Label merge requests			✓	✓	✓
Create code snippets		✓	✓	✓	✓
Manage issue tracker		✓	✓	✓	✓
Manage labels		✓	✓	✓	✓

GitLab – Starting a repository

- From GitLab:

Projects

Your projects Starred projects Explore projects Filter by name... Last updated New project

N	CIRC_NBI / NBI_standard_RNA-seq_pipeline <small>Owner</small>	★ 1  One out of two RNA-seq pipelines offered by the NBI. Its name contain "standard" as it uses the common alignment method (STAR), rather than pseudoaligners.	updated 1 day ago
G	Lara Ianov / General_HPC_setup <small>Maintainer</small>	★ 1  Useful alias to be added to .bashrc in HPC	updated 1 month ago
N	CIRC_NBI / NBI_WGBS_RRBS_PE_pipeline <small>Owner</small>	★ 1  WGBS and RRBS paired-end pipeline.	updated 2 months ago

[Blank project](#)[Create from template](#)[Import project](#)**Project path****Project name**

Want to house several dependent projects under the same namespace? [Create a group](#)

Project description (optional)

Description format

Path may be user profile or specific groups (e.g.: “CIRC_NBI”)

Visibility Level   Private

Project access must be granted explicitly to each user.

  Internal

The project can be accessed by any logged in user.

  Public

The project can be accessed without any authentication.

 Initialize repository with a README

Allows you to immediately clone this project’s repository. Skip this if you plan to push up an existing repository.

[Create project](#)[Cancel](#)

- From the command line:
 - Very nice when uploading a directory of existing files

Command line instructions

Git global setup

```
git config --global user.name "Lara Ianov"
git config --global user.email "lianov@uab.edu"
```

Create a new repository

```
git clone git@gitlab.rc.uab.edu:lianov/gitlab_demo.git
cd gitlab_demo
touch README.md
git add README.md
git commit -m "add README"
git push -u origin master
```

} Or type a one-liner to stage and commit all files:
`git commit -a -m "message goes here"`

Existing folder

```
cd existing_folder
git init
git remote add origin git@gitlab.rc.uab.edu:lianov/gitlab_demo.git
git add .
git commit -m "Initial commit"
git push -u origin master
```

Commands to upload files when initial project (in this case named “gitlab_demo”) was created in GitLab

Existing Git repository

```
cd existing_repo
git remote rename origin old-origin
git remote add origin git@gitlab.rc.uab.edu:lianov/gitlab_demo.git
git push -u origin --all
git push -u origin --tags
```

- From the command line:
 - Or create the project directly from the command line and push all files to the project

```
cd existing_folder
git init
git add .
git commit -m "initial commit message"
# With SSH
git push -u git@gitlab.rc.uab.edu:<namespace>/<project_name>.git master

# OR with HTTPS
git push -u https://gitlab.rc.uab.edu/<namespace>/<project_name>.git master
```

FYI on setting up SSH for GitLab: <https://docs.gitlab.com/ee/ssh/README.html>

GitLab – Edit and commit

- GitLab contains an integrated development environment (IDE) – OK for small edits, but I do not use it for large projects/changes.

The screenshot shows the GitLab interface for the 'gitlab_demo' repository. At the top, there's a navigation bar with 'CIRC_NBI > gitlab_demo > Repository'. Below it, a dropdown menu shows 'master' and a path 'gitlab_demo / demo.sh'. To the right are buttons for 'Find file', 'Blame', 'History', and 'Permalink'. The main area displays a commit titled 'initial commit' by 'Lara Ianov' from 6 minutes ago. A green circular icon next to the author's name has a grid pattern. To the right of the commit are the commit hash 'fb8a78fc' and a copy icon. Below the commit, there's a preview of the 'demo.sh' file content:

```
1 #!/bin/bash
2
3 echo "demo test for gitlab"
4
```

Next to the file preview are several edit-related buttons: 'Edit' (highlighted with an orange border), 'Web IDE', 'Replace', and 'Delete'.

GitLab's IDE

The screenshot shows the GitLab IDE interface. On the left, there's a sidebar with project details ('gitlab_demo', 'CIRC_NBL/gitlab_demo'), a branch dropdown ('master'), and navigation links ('Review', 'Latest changes', 'README.md', 'demo.sh'). The main area has two code editors. The left editor shows a file named 'demo.sh' with the following content:

```
1 #!/bin/bash
2
3 echo "demo test for gitlab"
4
5
```

The right editor also shows 'demo.sh' with the same content, plus a new line at the bottom:

```
1 #!/bin/bash
2
3 echo "demo test for gitlab"
4
5 +echo "printing working directory, `pwd`"
6 |
```

A green bar highlights the new line '5+'. In the top right corner of the main area, there's a link 'Open in file view'.

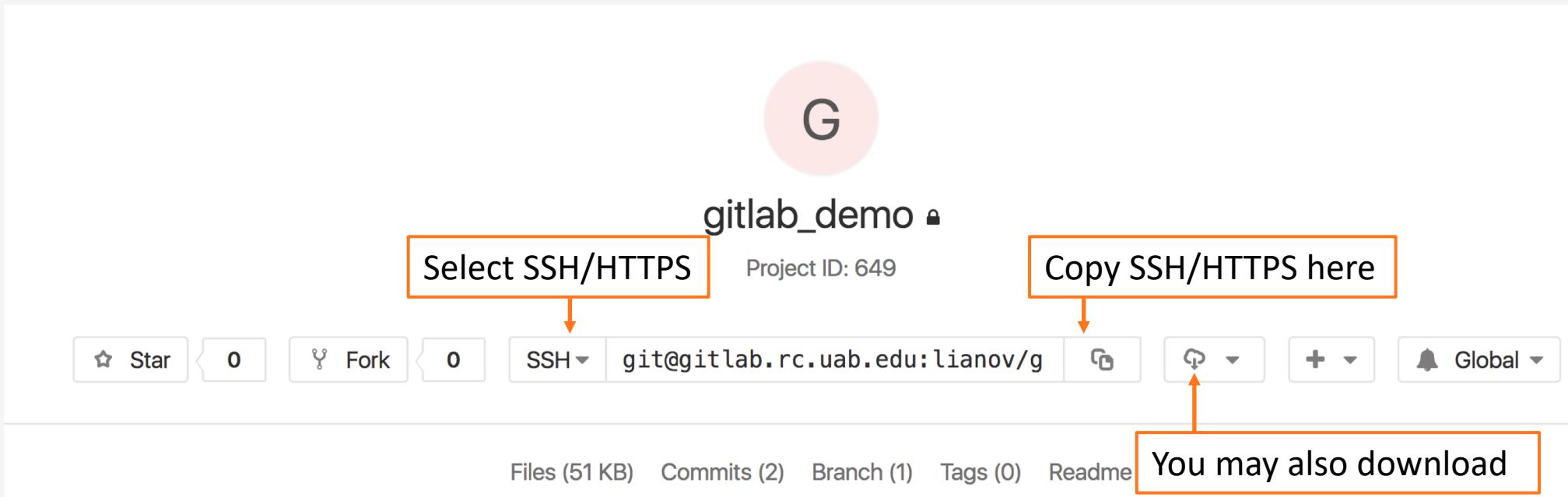
Hit “Stage and Commit” and include a message. Same as `git add . && git commit -m “message”`
No need to push as this is being done directly in GitLab.

The screenshot shows the 'Stage changes' dialog. It includes a 'Commit Message' field with the text 'Changed script to include working dir'. Below it are three radio button options: 'Commit to master branch' (selected), 'Create a new branch', and 'Create a new branch and merge request'. At the bottom are 'Stage & Commit' and 'Discard draft' buttons. A large bracket on the right side groups the commit message, options, and the 'Stage & Commit' button.

Note, the option of committing into the master branch or a new branch. A new branch would be wise if you are still testing the code or to work among collaborators where you would send (“create a merge request”) the reviewed code to the project’s maintainer.

GitLab – Edit and commit

- ✓ GitLab contains an integrated development environment (IDE) – OK for small edits, but I do not use it for large projects/changes.
- Cloning the repo to your local computer and using your favorite editor is usually best for large changes.
 - `git clone <SSH or HTTPS>`



GitLab – Edit and commit

- ✓ GitLab contains an integrated development environment (IDE) – OK for small edits, but I do not use it for large projects/changes.
- Cloning the repo to your local computer and using your favorite editor is usually best for large changes.
 - `git clone <SSH or HTTPS>`

```
demo.sh
1 #!/bin/bash
2
3 echo "demo test for gitlab"
4
5 echo "printing working directory, `pwd`"
6
7 echo "line added locally after cloning, and then committed and pushed to GitLab"
```

Line added in my editor of choice in my computer and then:
git add .
git commit -m "edit made locally in my editor"
git push -u origin master

GitLab – History

- Regardless of where changes were made (in GitLab IDE or locally and then pushed), all changes are easily tracked in GitLab:

The screenshot shows the GitLab interface for a repository named 'gitlab_demo'. The top navigation bar includes dropdowns for 'master' and 'gitlab_demo', a '+' button, and search/filter options like 'History', 'Find file', 'Web IDE', and a download icon. The 'History' button is highlighted with an orange border. Below the navigation, a commit card is displayed for a local edit by 'Lara Ianov' 17 minutes ago. At the bottom, a table lists files with their last commits and update times.

Name	Last commit	Last update
README.md	initial commit	1 hour ago
demo.sh	edit made locally in my editor	17 minutes ago

FYI command line version: git log

master gitlab_demo Filter by commit message

28 Nov, 2018 3 commits

 edit made locally in my editor
Lara Ianov authored 25 minutes ago

 Changed script to include working dir
Lara Ianov authored 1 hour ago

 initial commit
Lara Ianov authored 2 hours ago

Note the changes in SHA-1

0fd77486 0917c098 fb8a78fc

edit made locally in my editor

FYI command line version: git show <SHA-1>

-o parent 0917c098 ⌂ master
No related merge requests found

Showing 1 changed file ▾ with 2 additions and 0 deletions

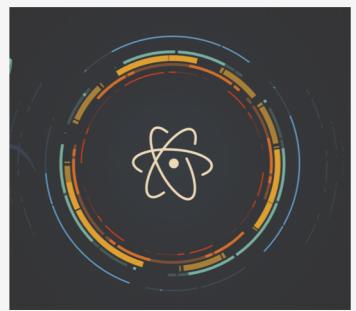
Hide whitespace changes Inline Side-by-side

demo.sh 100644 → 100755 ⌂

View file @ 0fd77486

```
... @@ -3,3 +3,5 @@
3 echo "demo test for gitlab"
4
5 echo "printing working directory, `pwd`"
... @@ -3,3 +3,5 @@
3 echo "demo test for gitlab"
4
5 echo "printing working directory, `pwd`"
6 +
7 + echo "line added locally after cloning, and then committed and pushed to GitLab"
```

GitLab – Atom integration



- Atom is a great editor for both GitHub and GitLab as it integrates the git commands, allowing you to perform tasks such as commit and push directly from the editor. Explore Atom's packages such as [this](#) to learn more

The screenshot shows the Atom code editor interface with GitLab integration. On the left, the file tree shows a directory named 'gitlab_demo' containing '.git', 'demo.sh', and 'README.md'. The main editor area contains a shell script (demo.sh) with several echo statements. A red arrow points from the text 'Highlights changes' in a callout box at the bottom left to the syntax highlighting of the script code. Another red box highlights the 'Stage All' button in the top right corner of the status bar. The status bar also shows 'Unstaged Changes' with 'demo.sh' listed, and 'Staged Changes' with 'No changes'. A callout box on the right highlights the commit history: 'Changes made in Atom and committed etc. directly' with a list including 'Commit to master' (24), 'Update README.md', 'edit made locally in my editor', and 'Changed script to include working dir'. At the bottom right, a callout box points to the 'Push 1' button in the status bar, with the text 'push icon is available after committing' and an arrow pointing to it.

1 `#!/bin/bash`
2
3 `echo "demo test for gitlab"`
4
5 `echo "printing working directory, `pwd`"`
6
7 `echo "line added locally after cloning, and then committed and pushed to GitLab"`
8
9 `# A good FYI about the Atom editor:`
10 `# changes below were made in Atom, staged, committed and pushed directly from Atom`
11 `echo "printing current date and time"`
12 `date`
13

Highlights changes

Unstaged Changes
demo.sh

Stage All

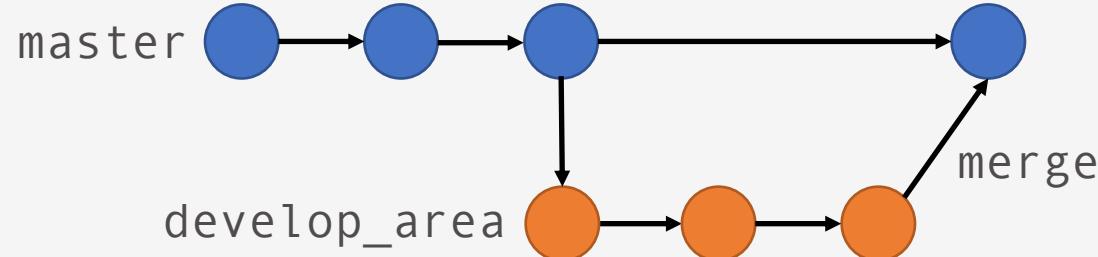
Staged Changes
No changes

Changes made in Atom and committed etc. directly

- Commit to master 24
- Update README.md
- edit made locally in my editor 2h
- Changed script to include working dir 3h

push icon is available after committing → ↵ master ↑ Push 1

GitLab - Branch



- By default, all projects start with one branch, called `master`. All commits I have done in this demo so far have been made to `master`.
- But what if you would like to work on the repo without committing to the main branch? Maybe testing code without overwriting the `master` branch?
 - This is where branching is very useful, especially when more than one person is working on the same code.
 - In the case of more than one person, a collaborator can branch, edit independently at the isolated branch and when finalized, send a merge request to the maintainer who can approve to merge all changes into the `master` branch.



gitlab_demo 🔒

Project ID: 649

Star

0

Fork

0

SSH

git@gitlab.rc.uab.edu:lianov/g...



Global

Files (92 KB)

Commits (5)

Branch (1)

Tags (0) Readme

Lara Ianov > gitlab_demo > Branches

Overview

Active

Stale

All

Filter by branch name

Delete merged branches

New branch

Protected branches can be managed in [project settings](#).

Active branches

master default protected

-o 25e27b03 · Changes made in Atom and committed etc. directly · 23 hours ago



New Branch

Branch name

Create from Existing branch name, tag, or commit SHA

Create branch Cancel



gitlab_demo

Project ID: 649

Star 0 Fork 0 SSH git@gitlab.rc.uab.edu:lianov/g...

Files (92 KB) Commits (5) **Branches (2)** Tags (0) Readme

You can now edit the code of the second branch and click on “branches” to compare and add a merge request

Overview Active Stale All

Filter by branch name

Delete merged branches

New branch

Protected branches can be managed in project settings.

Active branches

Y **develop_area**

-o [caeef48f](#) · Update demo.sh · 2 minutes ago

0 1

Merge request

Compare



Y **master** default protected

-o [25e27b03](#) · Changes made in Atom and committed etc. directly · 23 hours ago





Source

develop_area

...

Target

master

Compare

Create merge request

Commits (1)



Update demo.sh

Lara Ianov authored 3 minutes ago

caeef48f



Showing 1 changed file ▾ with 3 additions and 0 deletions

Hide whitespace changes

Inline

Side-by-side

▼ demo.sh



View file @ caeef48f

```
@@ -10,3 +10,5 @@ echo "line added locally after cloning, and then  
committed and pushed to GitLab"  
# changes below were made in Atom, staged, committed and pushed direct  
ly from Atom  
echo "printing current date and time"  
date  
...  
@@ -10,3 +10,5 @@ echo "line added locally after cloning, and then  
committed and pushed to GitLab"  
# changes below were made in Atom, staged, committed and pushed direct  
ly from Atom  
echo "printing current date and time"  
date  
13 +  
+ ----- This represents finalized code ready to be merged -----  
\\ No newline at end of file
```

GitLab - Branch

- Again, using GitLab's IDE may be OK for small changes but I prefer local for larger changes:

```
git clone <SSH or HTTPS>
cd <cloned_dir>
git branch -a
git checkout <branch_name> (switches the branch)
### modify code ####
git commit -am "message goes here" (adding in one-liner)
git push origin <branch_name>
```

- Atom's package offer good features to do this as well

Thanks!

A big thank you to UAB Research Computing for hosting GitLab and for taking the time to write documentation and train

My contact info:

lianov@uab.edu

CIRC 252C | P: 205-996-5871

[NBI](#)