

# Metron Development Environment Setup Instructions

The following provides instructions on how to set up your dev environment so you can start contributing to the Apache Metron project quickly.

The below instructions is with the IntelliJ as the IDE. However, you can use whatever IDE you would like to contribute to Metron.

- [Pre-Requisites](#)
- [Setting up your Dev Env](#)
  - [Fork the Metron Repo](#)
  - [Clone the Metron repo to you local Dev Env](#)
  - [Setup your Cloned Repo as an IntelliJ project](#)
- [How to Start Contributing](#)
  - [Create a JIRA Ticket](#)
  - [Committing your changes to your Local Repo](#)
- [Pushing the Local Committed Changes to the Forked Github repo](#)
- [Create a Pull Request](#)

## Pre-Requisites

1. Create an account on [Github](#) if you don't already have an account.
2. Download the latest version of [IntelliJ](#)

## Setting up your Dev Env

The below provides instructions on how to setup your Dev environment with Github and IntelliJ

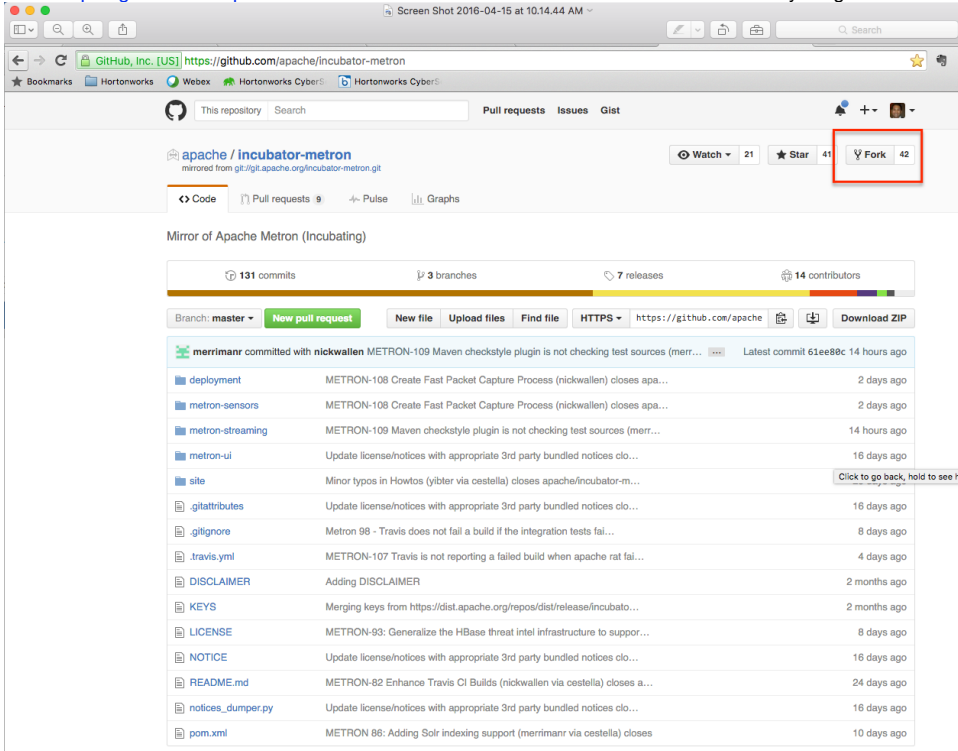
### Fork the Metron Repo

We need to first fork the Apache Metron repo. Forking a repository allows you to freely experiment with changes without affecting the original project. See [here](#) for details on forking.

The below provides instructions on how to fork the Metron Repo:

1. Log into your github account

2. Go to <https://github.com/apache/incubator-metron> and click the Fork button and fork into your github account.

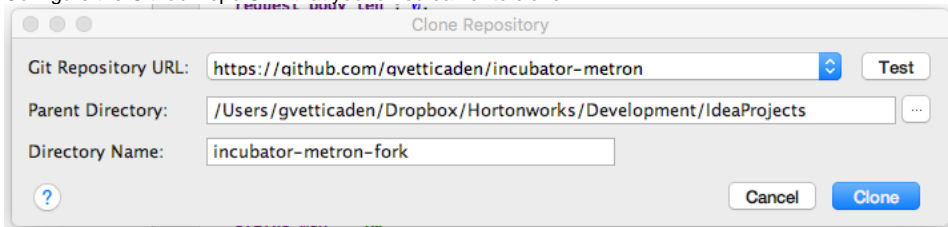


3. You should now have a fork like: [https://github.com/\[github\\_user\\_name\]/incubator-metron](https://github.com/[github_user_name]/incubator-metron)
- 4.

## Clone the Metron repo to you local Dev Env

After forking, clone the repo to your local dev workstation. See following instructions:

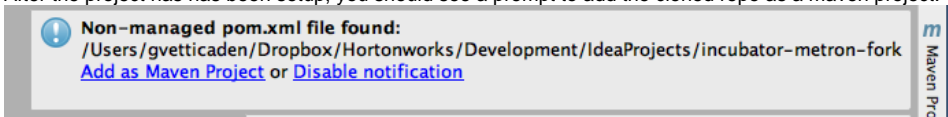
1. Start IntelliJ
2. Click File --> New --> Project from Version Control --> Github
3. Configure the Github Repo URL that you forked earlier to clone.



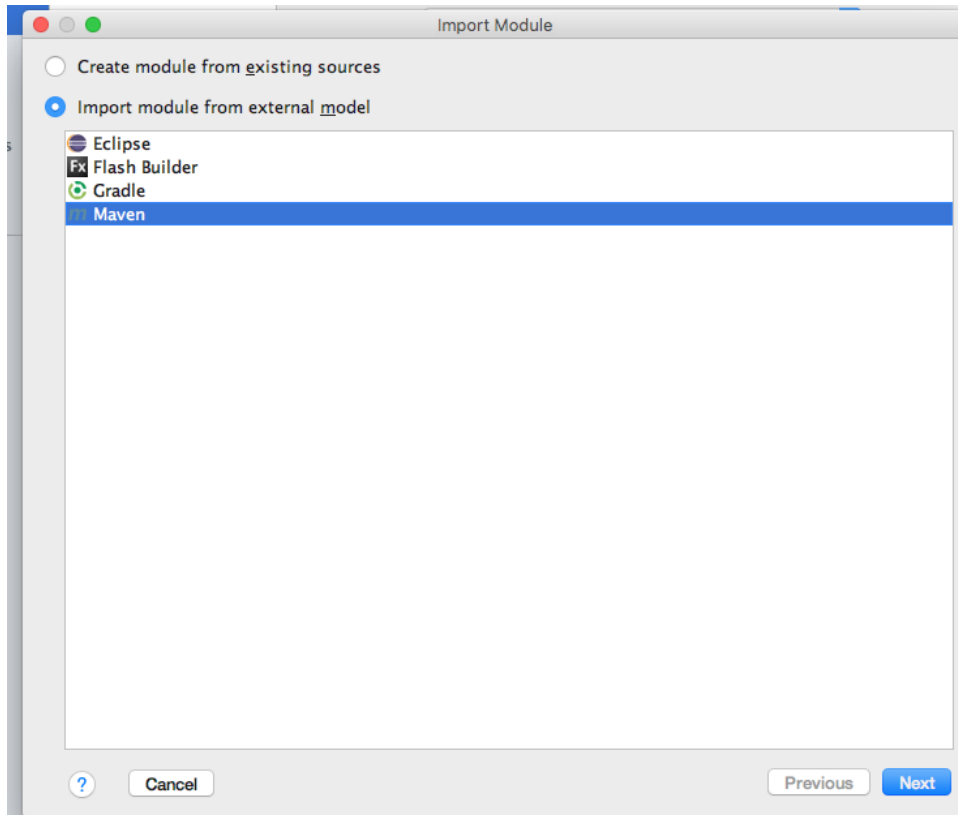
- 4.

## Setup your Cloned Repo as an IntelliJ project

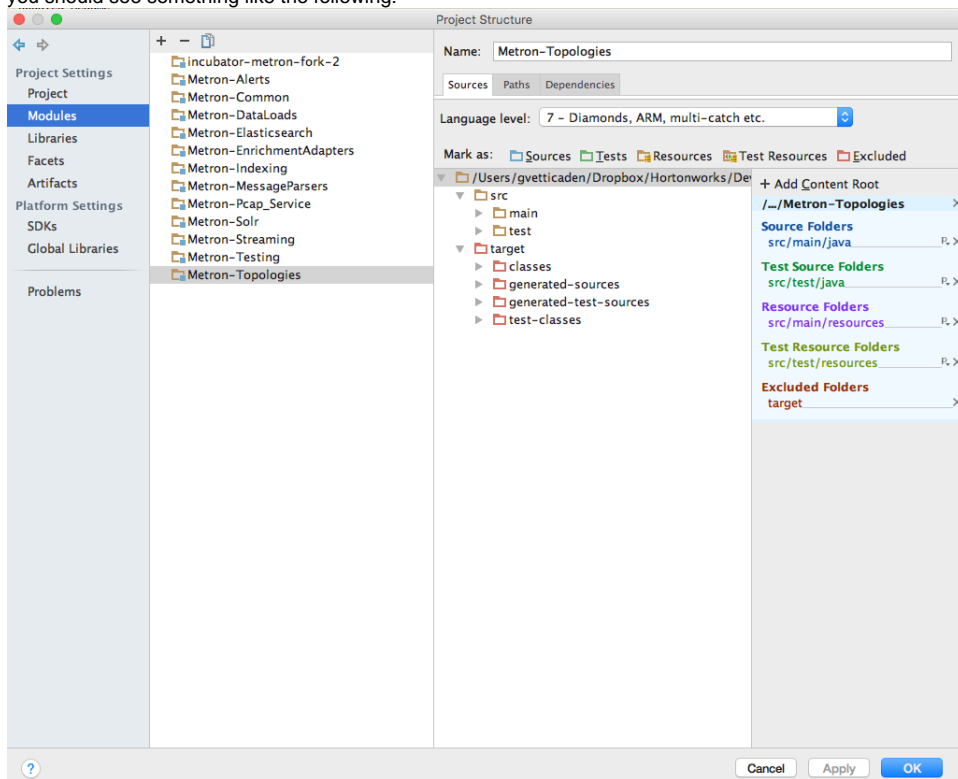
1. After the project has been setup, you should see a prompt to add the cloned repo as a maven project. Click the "Add as Maven Project"



- 2.
3. Setup the Java SDK as 1.8 (if you don't have oracle jdk 1.8, please install it first)
  - a. File -> Project Structure --> Project Settings --> Project --> Select 1.8 as SDK
4. Setup the Project Structure correctly
  - a. Go to File --> Project Structure --> Project Settings --> Module
  - b. select plus sign and select "Import Module"
  - c. Go to the location where you cloned your fork on your local workstation and select the metron-streaming folder
  - d. Check the checkbox "Import module as from External Folder" and Select Maven



- e.
- f. Leave the default setting on the import settings page
- g. Click Finish
- h. you should see something like the following:



- i.
- j. Do steps b - g for the following folders under the cloned repo directory: deployment, metron-sensors, metron-ui and site
- k. Build the entire project and ensure that you have no errors: Build --> Make Project

# How to Start Contributing

Now that you have setup your IDE environment and want to contribute on an enhancement, do the the following:

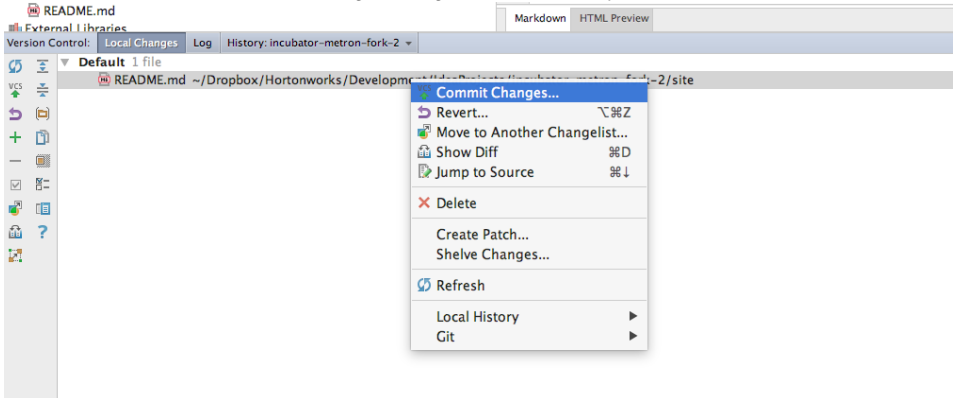
## Create a JIRA Ticket

1. Create JIRA ticket in [Metron JIRA](#) for the enhancement that you want to work on if it doesn't exist.
2. Do your work within your IntelliJ project

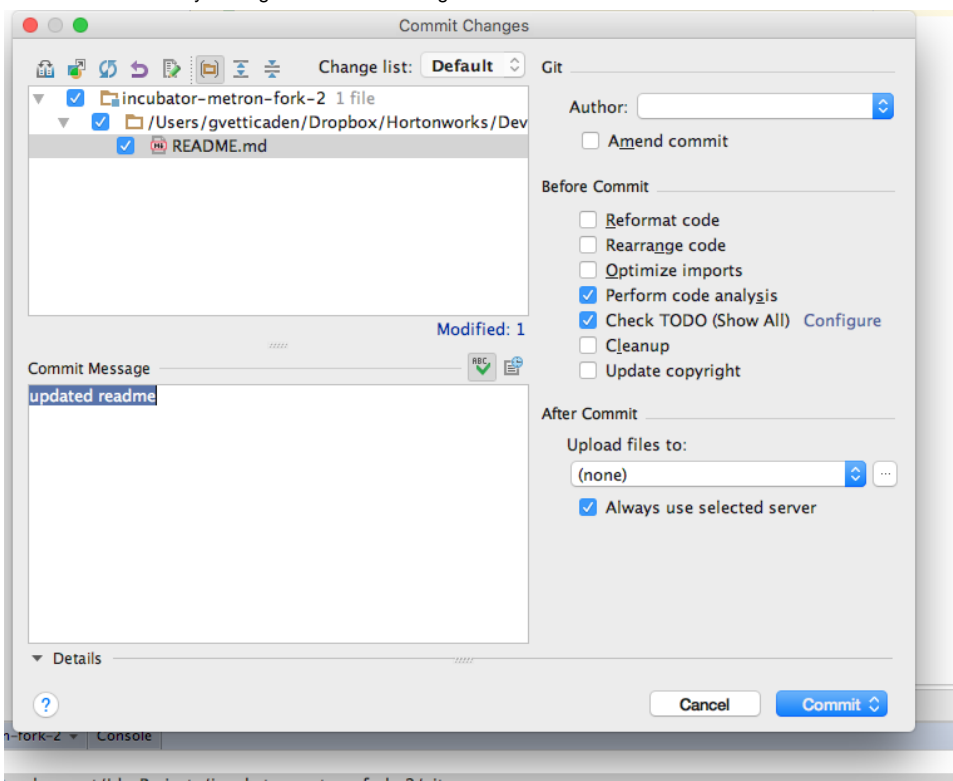
## Committing your changes to your Local Repo

After you have completd a small unit of work for the enhancement, you want to commit the changes to your lcoal repo. Do the following

1. Go to version control tab --> local changes --> right on the files that you modified or created --> select commit changes



- 2.
3. Confirm the commit by adding the commit message and the JIRA numebr associated with the commit.

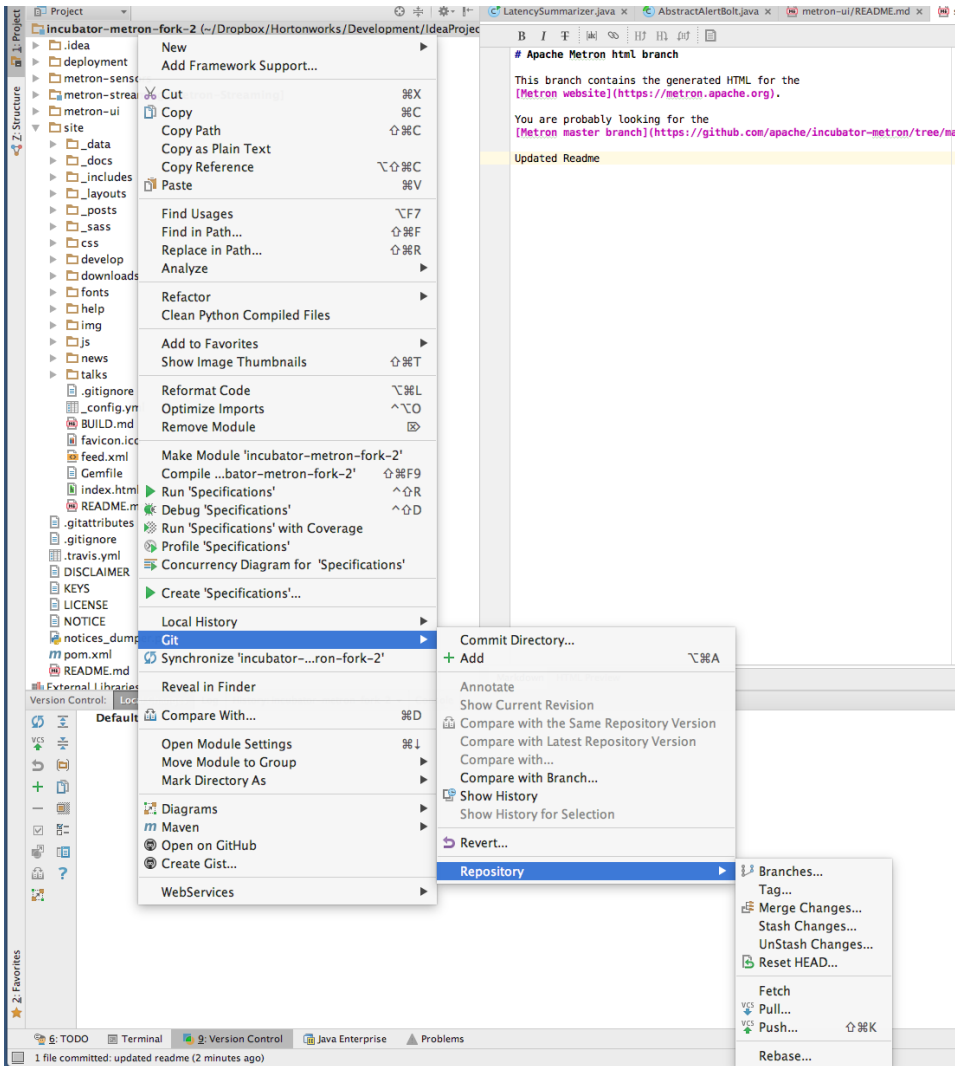


4. development/IdeaProjects/incubator-metron-fork-2/site

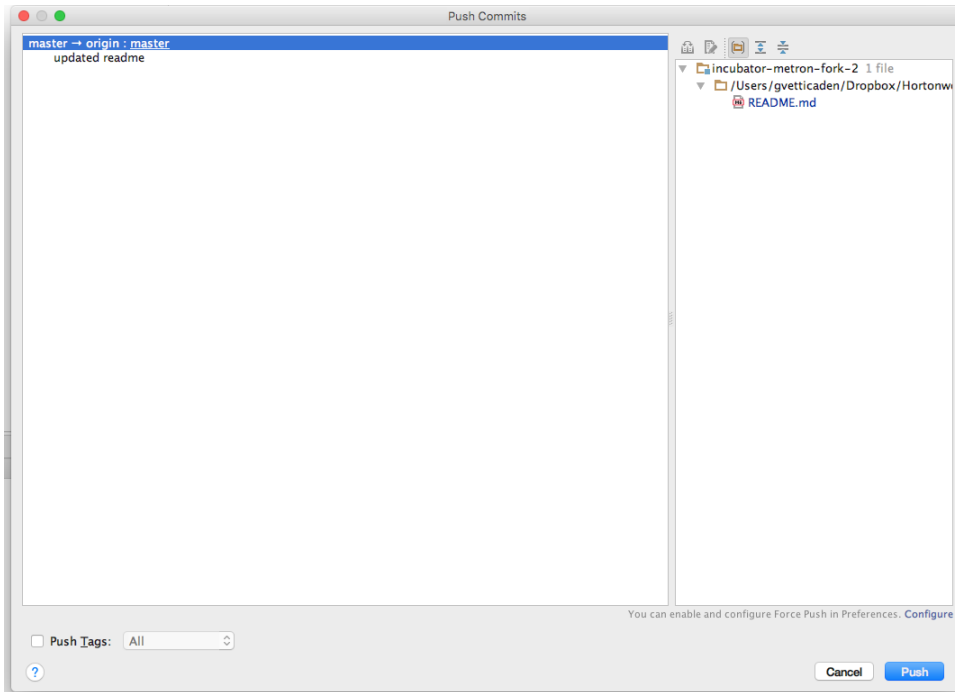
## Pushing the Local Committed Changes to the Forked Github repo

Now that the changes have been committed to local repo. We want to push and commit those changes to your forked github.

1. Right Click on on the project in IntelliJ --> Git --> Repository --> Push



2. Click Push



- 4.
5. You should now see the changes in the forked repo on your github account

## Create a Pull Request

Pull requests let you tell others about changes you've pushed to a GitHub repository. Once a pull request is sent, interested parties can review the set of changes, discuss potential modifications, and even push follow-up commits if necessary.

1. Log into your github account and go to your forked repo.
2. Click the button "new pull request"

gvticaden / incubator-metron

132 commits 3 branches 7 releases 14 contributors

Branch: master **New pull request** New file Upload files Find file HTTPS https://github.com/gvticaden/incubator-metron Download ZIP

This branch is 1 commit ahead of apache:master.

gvticaden updated readme Latest commit b288caf 2 hours ago

deployment	METRON-108 Create Fast Packet Capture Process (nickwallen) closes apa...	2 days ago
metron-sensors	METRON-108 Create Fast Packet Capture Process (nickwallen) closes apa...	2 days ago
metron-streaming	METRON-109 Maven checkstyle plugin is not checking test sources (merr...	15 hours ago
metron-ui	Update license/notices with appropriate 3rd party bundled notices clo...	16 days ago
site	updated readme	2 hours ago
.gitattributes	Update license/notices with appropriate 3rd party bundled notices clo...	16 days ago
.gitignore	Metron 98 - Travis does not fail a build if the integration tests fai...	8 days ago
.travis.yml	METRON-107 Travis is not reporting a failed build when apache rat fai...	4 days ago
DISCLAIMER	Adding DISCLAIMER	2 months ago
KEYS	Merging keys from https://dist.apache.org/repos/dist/release/incubato...	2 months ago
LICENSE	METRON-93: Generalize the HBase threat intel infrastructure to suppor...	8 days ago
NOTICE	Update license/notices with appropriate 3rd party bundled notices clo...	16 days ago
README.md	METRON-82 Enhance Travis CI Builds (nickwallen via cestella) closes a...	24 days ago
notices_dumper.py	Update license/notices with appropriate 3rd party bundled notices clo...	16 days ago
pom.xml	METRON 88: Adding Solr indexing support (merrimann via cestella) closes	10 days ago
README.md		

- 3.
4. Please be very careful to preface your PR with the full name of the JIRA so that the comments from github are replicated to the JIRA system for posterity. The name must be exact. If there is a JIRA, METRON-777, that is titled "Fix all the things!", please create your PR with the title: "METRON-777: Fix all the things!"
5. Please fill out the PR template completely. For code submissions it will ask you not only to run full Unit Test, but also the integration test module, via
  - a.

```
mvn -q clean integration-test install && build_utils/verify_licenses.sh
```