Verifying Builds

Step 0 – Verify Release

Verify that the maven artifacts are the right version by inspecting the top level pom.xml and looking at the version.

Verify that the release bits are the signed properly by running:

```
gpg --import KEYS

export VERSION=<metron-release-version>
gpg --verify ./apache-metron_$VERSION.tar.gz.asc apache-metron_$VERSION.tar.gz
```

Note: on Mac you might not have gpg installed by default. You can add it via Homebrew with the command "brew install gpg"

Step 1 – Build Metron

```
cd apache-metron_$VERSION
mvn clean install
```

Verify that all tests are passing

Step 2 – Deploy metron as a single VM via vagrant and Ansible

```
cd metron-deployment/development/centos6/
vagrant plugin install vagrant-hostmanager
vagrant up
```

For a more complete set of instructions refer to:
https://github.com/apache/metron/tree/master/metron-deployment and
https://github.com/apache/metron/tree/master/metron-deployment/development

Verify metron is working:
- Check Ambari to make sure all the services are up by going to ambari in a browser at http://node1:8080

- Check Storm to make sure all the topologies are up
  From Ambari navigate to Storm -> Quick Links -> Storm UI
- Check that the enrichment topology has emitted some data (could take a few minutes to show up in the Storm UI)
- Check indexes to make sure indexing is done correctly and data is visualized in Kibana in a browser at http://node1:5000
- Check that some data is written into HDFS for at least one of the data sources
  Look in HDFS under /apps/metron/indexing/indexed/yaf|bro|snort

  This can be done by running hdfs dfs -ls /apps/metron/indexing/indexed/

- Test the Management UI at http://node1:4200/

Step 3 (optional) – Verify AWS Multi-Node Deploy with Ansible (NOTE: This will cost money to deploy AWS servers)

```
cd metron-deployment/amazon-ec2
./run.sh
```

For a more complete set of instructions refer to:
https://github.com/apache/metron/tree/master/metron-deployment

To verify the working build go through the same verifications as in Step2, but on AWS. Reference playbook.yml for location of the services.
Ambari-master contains Ambari, web contains Kibana and sensors.