Hortonworks Cybersecurity Package

Apache Zeppelin Component Guide

(July 12, 2017)

docs.hortonworks.com

Hortonworks Cybersecurity Package: Apache Zeppelin Component Guide

Hortonworks Cybersecurity Package (HCP) is a modern data application based on Apache Metron, powered by Apache Hadoop, Apache Storm, and related technologies.

HCP provides a framework and tools to enable greater efficiency in Security Operation Centers (SOCs) along with better and faster threat detection in real-time at massive scale. It provides ingestion, parsing and normalization of fully enriched, contextualized data, threat intelligence feeds, triage and machine learning based detection. It also provides end user near real-time dashboards.

Based on a strong foundation in the Hortonworks Data Platform (HDP) and Hortonworks DataFlow (HDF) stacks, HCP provides an integrated advanced platform for security analytics.

Please visit the Hortonworks Data Platform page for more information on Hortonworks technology. For more information on Hortonworks services, please visit either the Support or Training page. Feel free to Contact Us directly to discuss your specific needs.

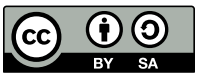


Table of Contents

List of Figures

1. Overview

This guide is intended for use by Security Operations Center (SOC) analysts and investigators. This guide describes how to set up the Apache Zeppelin dashboard and use runbooks generated with Zeppelin. Like the Metron dashboard, the Zeppelin dashboard can be used to view and analyze the enriched telemetry data provided by HCP. However Zeppelin can be used by a data scientist to create runbooks for recreatable investigations. These runbooks can be static, which require no input, or dynamic, which require you to enter or choose information.

This guide contains the following information:

2. Setting up Zeppelin to Run with HCP

To install Zeppelin with HCP, see the following sections:

- Importing Zeppelin Notebook Using Ambari
- Importing the Apache Zeppelin Notebook Manually

Setting up Zeppelin is very simple. To access Zeppelin, go to `http://$ZEPPELIN_HOST:9995`. To complete your set up, see the following sections:

- Using Zeppelin Interpreters [2]
- Loading Telemetry Information into Zeppelin [2]

In addition to this documentation, there are three other sources for Zeppelin information.

- The Zeppelin installation for HCP provides a couple sample notes including tutorials specific to Metron. These notes are listed on the left side of the **Welcome** screen and in the **Notebook** menu.
- Zeppelin documentation provides information on launching and using Zeppelin, and you can refer to the following links for this information:
  - Using Zeppelin Interpreters [2]
  - Loading Telemetry Information into Zeppelin [2]
- Apache Zeppelin documentation provides information on Zeppelin basic features, supported interpreters, and more. To view the Apache Zeppelin documentation, see Apache Zeppelin 0.7.0.

2.1. Using Zeppelin Interpreters

When you install Zeppelin on HCP the installation includes the interpreter for Spark. Spark is the main backend processing engine for Zeppelin. Spark is also a front end for Python, Scala, and SQL and you can use any of these languages to analyze the telemetry data.

2.2. Loading Telemetry Information into Zeppelin

Before you can analyze telemetry information in Zeppelin, you must first download it from Metron. Metron archives the fully parsed, enriched, and triaged telemetry for each sensor in HDFS. This archived telemetry information is simply raw JSON files which makes it simple to parse and analyze the information with Zeppelin. The following is an example of some Bro telemetry information.

```
%sh

hdfs dfs -ls -C -R /apps/metron/indexing/indexed/bro
/apps/metron/indexing/indexed/bro/enrichment-null-0-0-1484124296101.json
/apps/metron/indexing/indexed/bro/enrichment-null-0-0-1484128332104.json
/apps/metron/indexing/indexed/bro/enrichment-null-0-0-1484131460758.json
```

```
/apps/metron/indexing/indexed/bro/enrichment-null-0-1-1484217861096.json
/apps/metron/indexing/indexed/bro/enrichment-null-0-10-1484995461039.json
/apps/metron/indexing/indexed/bro/enrichment-null-0-11-1485081861043.json
/apps/metron/indexing/indexed/bro/enrichment-null-0-12-1485168261040.json
/apps/metron/indexing/indexed/bro/enrichment-null-0-13-1485254661040.json
/apps/metron/indexing/indexed/bro/enrichment-null-0-14-1485341061047.json
/apps/metron/indexing/indexed/bro/enrichment-null-0-15-1485427461040.json
/apps/metron/indexing/indexed/bro/enrichment-null-0-16-1485513861039.json
/apps/metron/indexing/indexed/bro/enrichment-null-0-17-1485600261045.json
/apps/metron/indexing/indexed/bro/enrichment-null-0-18-1485686661035.json
/apps/metron/indexing/indexed/bro/enrichment-null-0-19-1485773061037.json
/apps/metron/indexing/indexed/bro/enrichment-null-0-2-1484304261042.json
/apps/metron/indexing/indexed/bro/enrichment-null-0-20-1485859461037.json
/apps/metron/indexing/indexed/bro/enrichment-null-0-21-1485945861039.json
/apps/metron/indexing/indexed/bro/enrichment-null-0-22-1486032261036.json
```

You can use Spark to load the archived information from HDFS into Zeppelin.

For example if you are loading information received from Bro, your command would like the following:

```
%spark
sqlContext.read.json("hdfs:///apps/metron/indexing/indexed/bro").cache().
registerTempTable("bro")
```

3. Working with Runbooks

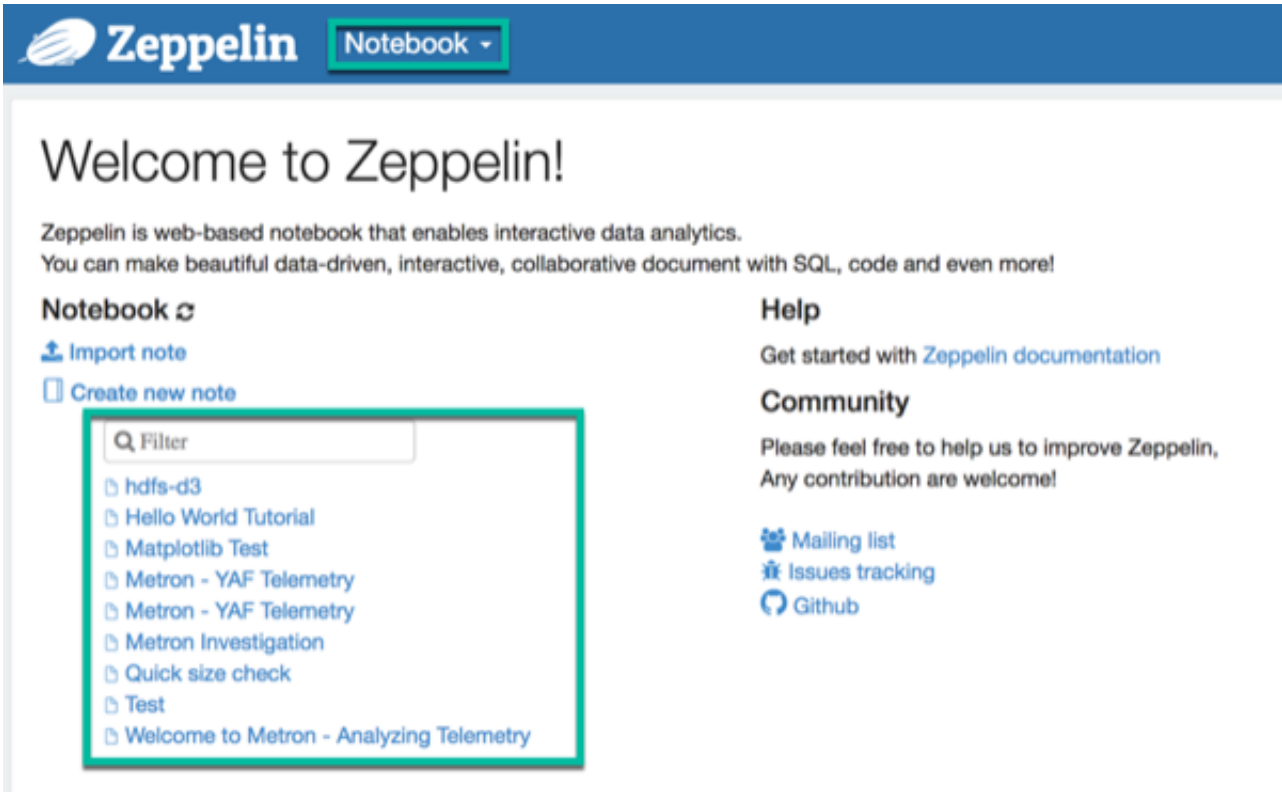
Runbooks created with Zeppelin consist of a note that is divided into paragraphs. Each paragraph consists of two sections: the code section and the result section. A runbook containing a recreatable investigation is already populated with the code and visualization choices that you will need to analyze data. You will need to choose or provide information if the runbook contains dynamic forms and run each paragraph to view and analyze the data provided in each results section. Dynamic fields that allow you to choose or enter information such as an IP address, allow you to reuse the runbook multiple times with different variables. For more information on using notes, see Working with Notes.

To use a runbook, complete the following steps:

1. Find the runbook note that you want to use.

   Zeppelin lists available notes on the left side of the welcome screen and in the **Notebook** menu.

   Figure 3.1. Zeppelin Welcome Screen

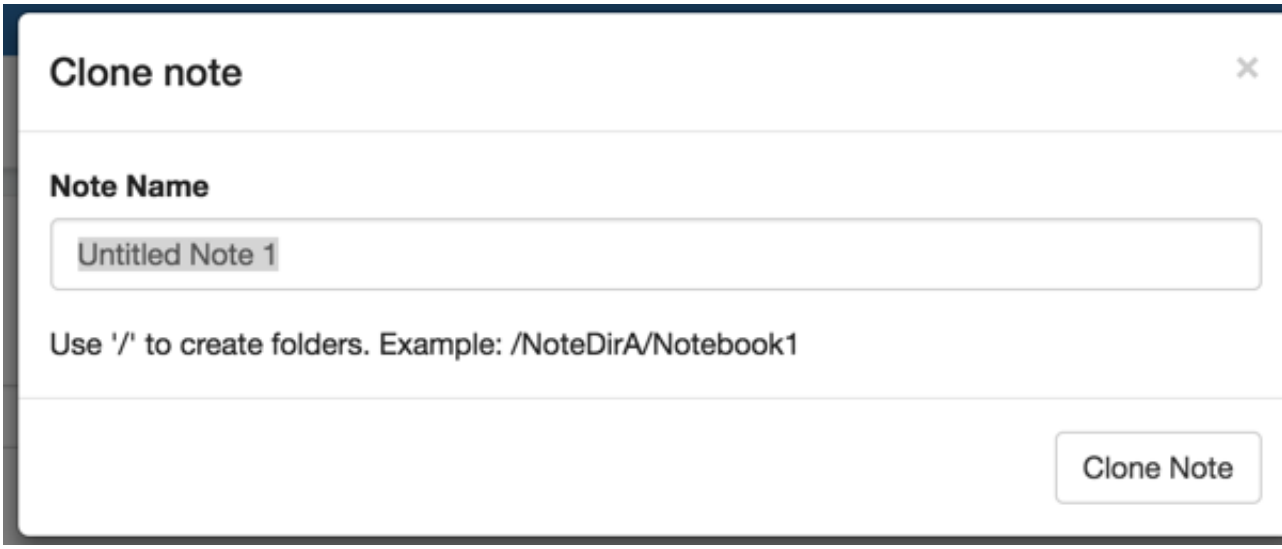


2. Clone the note.

Figure 3.2. Zeppelin Clone Icon

Zeppelin displays a **Clone note** dialog box:

Figure 3.3. Zeppelin Clone Note

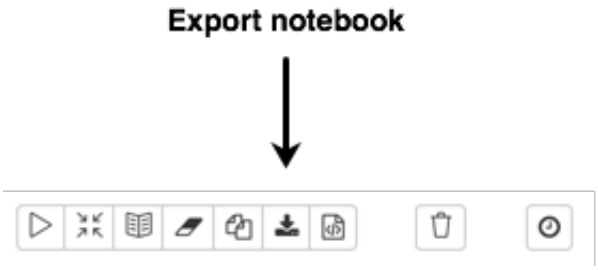


3. Enter the new name for the note in the **Note Name** field, then click **Clone Note**.
4. Now you can start to use the runbook. For each paragraph in the note, enter or choose any relevant information, then run the paragraph by clicking the triangle button in the cell that contains the code:

Figure 3.4. Zeppelin Triangle

5. Review the results for any suspicious behavior.
6. If you find suspicious issues that you would like to share, you can save the current version of the note.

Figure 3.5. Zeppelin Export Icon



a. From the Zeppelin toolbar, click the **Export the notebook** icon.

   By default, Zeppelin downloads the notebook as a JSON file that you can then share .