HCP Zeppelin Guide 1

# Hortonworks Cybersecurity Package

**Date of Publish:** 2018-08-23

**http://docs.hortonworks.com**

# Contents

# Introduction to Using Zeppelin With HCP

The Zeppelin dashboard is intended for use by Security Operations Center (SOC) analysts and investigators.

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.

# Installing Zeppelin

You can install Zeppelin either using Ambari or manually.

## Install Apache Zeppelin Using Ambari

How to install Apache Zeppelin on an Ambari-managed cluster.

### Before you begin

Install Zeppelin on a node where Spark clients are already installed and running. This typically means that Zeppelin will be installed on a gateway or edge node.

Zeppelin requires the following software versions:

- HDP 3.0 or later.
- Apache Spark 2.0.
- Java 8 on the node where Zeppelin is installed.

The optional Livy server provides security features and user impersonation support for Zeppelin users. Livy is installed as part of Spark.

- After installing Spark, Livy, and Zeppelin, refer to "Configuring Zeppelin" in this guide for post-installation steps.

### Install Zeppelin Using Ambari

The Ambari installation wizard sets default values for Zeppelin configuration settings. Initially, you should accept the default settings. Later, when you are more familiar with Zeppelin, consider customizing the Zeppelin configuration settings.

To install Zeppelin using Ambari, add the Zeppelin service:

1. Click the ellipsis (…) symbol next to Services on the Ambari dashboard, then click Add Service.
2. On the Add Service Wizard under Choose Services, select Zeppelin Notebook, then click Next.
3. On the Assign Masters page, review the node assignment for Zeppelin Notebook, then click Next.
4. On the Customize Services page, review the default values, then click Next.
5. If Kerberos is enabled on the cluster, review the principal and keytab settings on the Configure Identities page, modify the settings if desired, then click Next.
6. Review the configuration on the Review page, then click Deploy to begin the installation.
7. The Install, Start, and Test page displays the installation status.
8. When the progress bar reaches 100% and a "Success" message appears, click Next.
9. On the Summary page, click Complete to finish installing Zeppelin.

To validate the Zeppelin installation, open the Zeppelin Web UI in a browser window. Use the port number configured for Zeppelin (9995 by default); for example:

```
http://<zeppelin-host>:9995
```

You can also open the Zeppelin Web UI by selecting Zeppelin Notebook > Zeppelin UI on the Ambari dashboard.

To check the Zeppelin version number, type the following command on the command line:

```
/usr/hdp/current/zeppelin-server/bin/zeppelin-daemon.sh --version
```

Zeppelin stores configuration settings in the /etc/zeppelin/conf directory. Note, however, that if your cluster is managed by Ambari you should not modify configuration settings directly. Instead, use the Ambari web UI.

Zeppelin stores log files in /var/log/zeppelin on the node where Zeppelin is installed.

## Import the Apache Zeppelin Notebook Manually

As an alternative to using Ambari to install Apache Zeppelin you can manually install the tool.

### Procedure

1. Use ssh to navigate to the host where you want to install Zeppelin.

   ```
   ssh $METRON_HOME
   ```

2. Use the following command to import the $METRON_HOME/config/zeppelin/metron-yaf-telemetry.json file onto your Zeppelin host.

   ```
   curl -s -XPOST https://github.com/apache/incubator-metron/blob/master/
   metron-platform/metron-indexing/src/main/config/zeppelin/metron/metron-
   yaf-telemetry.json/api/notebook/import -d @"$METRON_HOME/config/zeppelin/
   metron-yaf-telemetry.json"
   ```

3. Navigate to http://$ZEPPELIN_HOST:9995.

# Setting up Zeppelin to Run with HCP

You can import the Zeppelin Notebook using Ambari or manually. To complete your set up you'll need to use Zeppelin interpretors and load the telemetry information.

Setting up Zeppelin is very simple. To access Zeppelin, go to http://$ZEPPELIN_HOST:9995.

In addition to this documentation, there are two 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 additional information on using Zeppelin.

# Using Zeppelin Interpreters

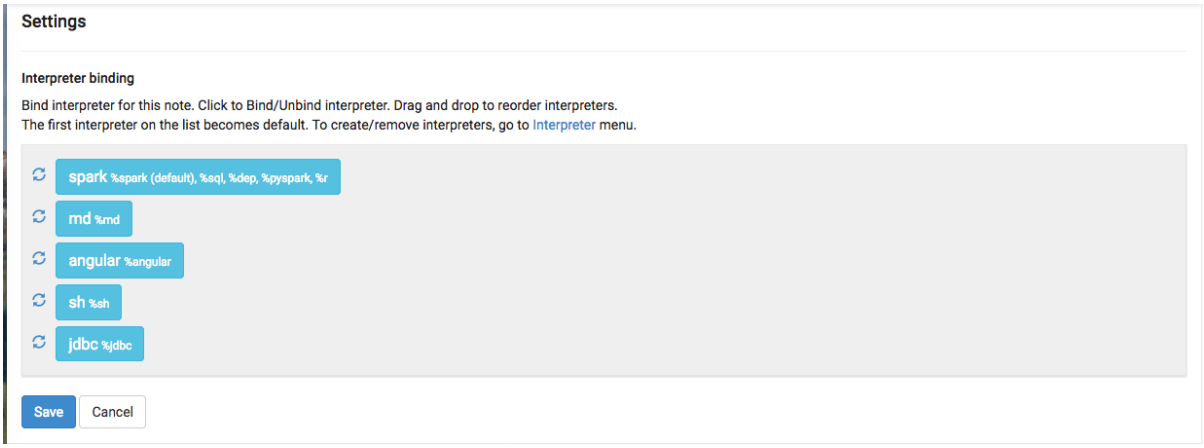
This section describes how to use Apache Zeppelin interpreters.

Before using an interpreter, ensure that the interpreter is available for use in your note:

1. Navigate to your note.

2. Click on “interpreter binding”:


3. Under "Settings", make sure that the interpreter you want to use is selected (in blue text). Unselected interpreters appear in white text:




4. To select an interpreter, click on the interpreter name to select the interpreter. Each click operates as a toggle.
5. You should unselect interpreters that will not be used. This makes your choices clearer. For example, if you plan to use %livy to access Spark, unselect the %spark interpreter.

Whenever one or more interpreters could be used to access the same underlying service, you can specify the precedence of interpreters within a note:

- Drag and drop interpreters into the desired positions in the list.
- When finished, click "Save".

Use an interpreter in a paragraph

To use an interpreter, specify the interpreter directive at the beginning of a paragraph, using the format %[INTERPRETER_NAME]. The directive must appear before any code that uses the interpreter.

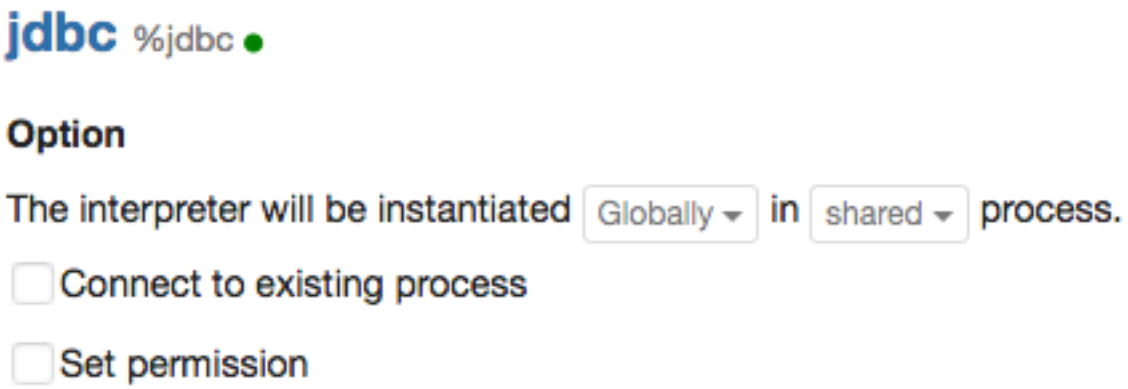
The following paragraph uses the %sh interpreter to access the system shell and list the current working directory:

```
%sh
pwd

home/zeppelin
```

Some interpreters support more than one form of the directive. For example, the %livy interpreter supports directives for PySpark, PySpark3, SparkR, Spark SQL.

To view interpreter directives and settings, navigate to the Interpreter page and scroll through the list of interpreters or search for the interpreter name. Directives are listed immediately after the name of the interpreter, followed by options and property settings. For example, the JDBC interpreter supports the %jdbc directive:




Note: The Interpreter page is subject to access control settings. If the Interpreters page does not list settings, check with your system administrator for more information.

Use interpreter groups

Each interpreter belongs to an interpreter group. Interpreters in the same group can reference each other. For example, if the Spark SQL interpreter and the Spark interpreter are in the same group, the Spark SQL interpreter can reference the Spark interpreter to access its SparkContext.

# Loading Telemetry Information into Zeppelin

Before you can analyze telemetry information in Zeppelin, you must first download it from Hortonworks Cybersecurity Platform (HCP).

HCP 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 look like the following:

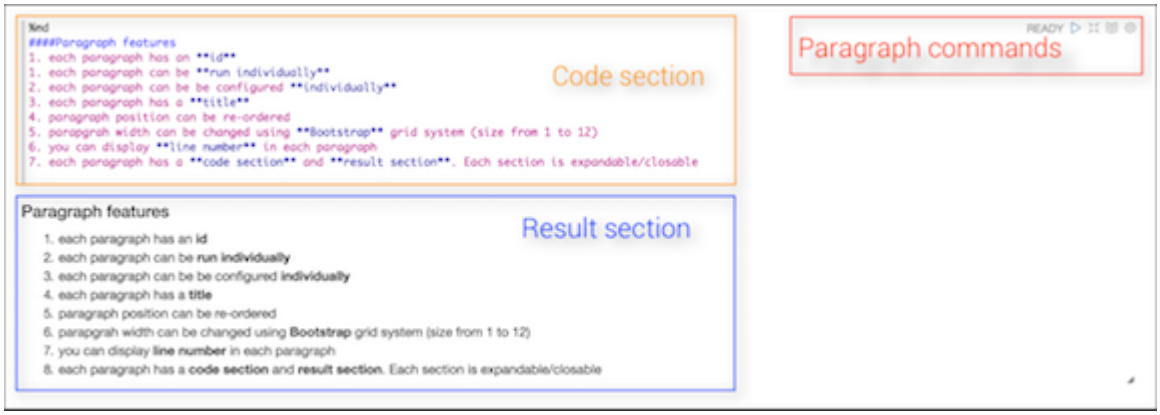
```
%spark
sqlContext.read.json("hdfs:///apps/metron/indexing/indexed/
bro").cache().registerTempTable("bro")
```

# Working with Zeppelin Notes

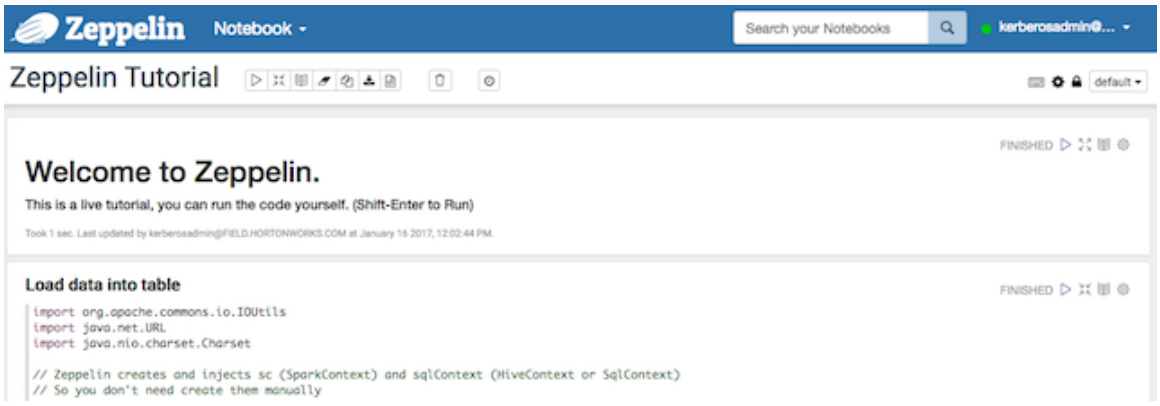
This section provides an introduction to Apache Zeppelin notes.

An Apache Zeppelin note consists of one or more paragraphs of code, which you can use to define and run snippets of code in a flexible manner.

A paragraph contains code to access services, run jobs, and display results. A paragraph consists of two main sections: an interactive box for code, and a box that displays results. To the right is a set of paragraph commands. The following graphic shows paragraph layout.




Zeppelin ships with several sample notes, including tutorials that demonstrate how to run Spark scala code, Spark SQL code, and create visualizations.




To run a tutorial:

1. Navigate to the tutorial: click one of the Zeppelin tutorial links on the left side of the welcome page, or use the Notebook pull-down menu.
2. Zeppelin presents the tutorial, a sequence of paragraphs prepopulated with code and text.
3. Starting with the first paragraph, click the triangle button at the upper right of the paragraph. The status changes to PENDING, RUNNING, and then FINISHED when done.
4. When the first cell finishes execution, results appear in the box underneath your code. Review the results.
5. Step through each cell, running the code and reviewing results.

## Create and Run a Note

Use the following steps to create and run an Apache Zeppelin note.

To create a note:

1. Click "Create new note" on the welcome page, or click the "Notebook" menu and choose "+ Create new note."
2. Type your commands into the blank paragraph in the new note.

When you create a note, it appears in the list of notes on the left side of the home page and in the Notebook menu. By default, Zeppelin stores notes in the $ZEPPELIN_HOME/notebook folder.

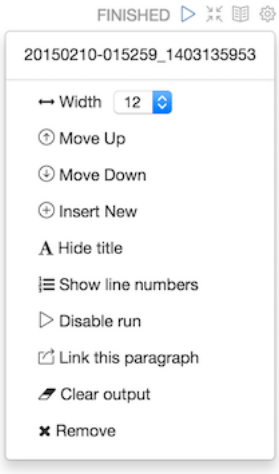
To run your code:

1. Click the triangle button in the cell that contains your code:

2. Zeppelin displays status near the triangle button: PENDING, RUNNING, ERROR, or FINISHED.
3. When finished, results appear in the result section below your code.

The settings icon (outlined in red) offers several additional commands:


These commands allow you to perform several note operations, such as showing and hiding line numbers, clearing the results section, and deleting the paragraph.




## Import a Note

Use the following steps to import an Apache Zeppelin note.

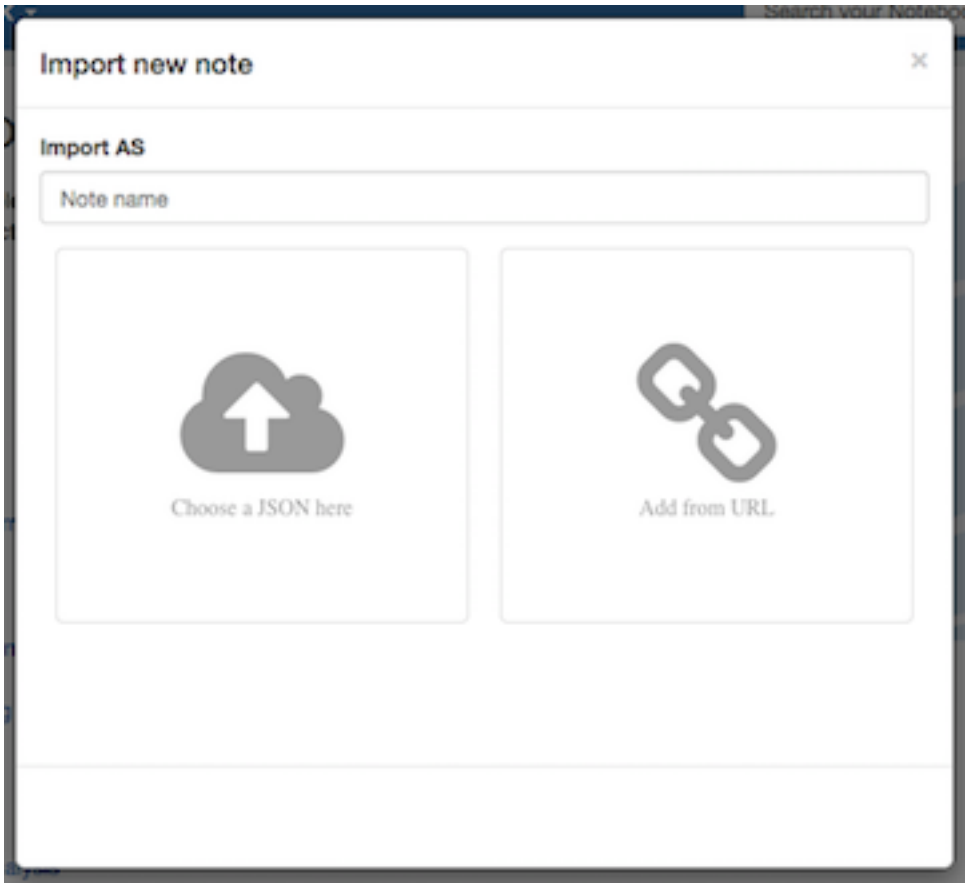
### About this task

To import a note from a URL or from a JSON file in your local file system:

### Procedure

1. Click "Import note" on the Zeppelin home page:


2. Zeppelin displays an import dialog box:



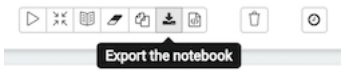

3. To upload the file or specify the URL, click the associated box.

   By default, the name of the imported note is the same as the original note. You can rename it by providing a new name in the "Import AS" field.

## Export a Note

Use the following steps to export an Apache Zeppelin note.

To export a note to a local JSON file, use the export note icon in the note toolbar:




Zeppelin downloads the note to the local file system.

Note: Zeppelin exports code and results sections in all paragraphs. If you have a lot of data in your results sections, consider trimming results before exporting them.

## Using the Note Toolbar

This section describes how to use the Apache Zeppelin Note toolbar.

At the top of each note there is a toolbar with buttons for running code in paragraphs and for setting configuration, security, and display options:

There are several buttons in the middle of the toolbar:

These buttons perform the following operations:

- Execute all paragraphs in the note sequentially, in the order in which they are displayed in the note.
- Hide or show the code section of all paragraphs.
- Hide or show the result sections in all paragraphs.
- Clear the result section in all paragraphs.
- Clone the current note.
- Export the current note to a JSON file.

  Note that the code and result sections in all paragraphs are exported. If you have extra data in some of your result sections, trim the data before exporting it.
- Commit the current note content.
- Delete the note.
- Schedule the execution of all paragraphs using CRON syntax. This feature is not currently operational. If you need to schedule Spark jobs, consider using Oozie Spark action.

There are additional buttons on the right side of the toolbar:


These buttons perform the following operations (from left to right):

- Display all keyboard shortcuts.
- Configure interpreters that are bound to the current note.
- Configure note permissions.
- Switch display mode:
  - Default: the notebook can be shared with (and edited by) anyone who has access to the notebook.
  - Simple: similar to default, with available options shown only when your cursor is over the cell.
  - Report: only your results are visible, and are read-only (no editing).

  Note: Zeppelin on HDP does not support sharing a note by sharing its URL, due to lack of proper access control over who and how a note can be shared.