Automated Test Bed

Release 1.0

Getting Started Guide

Contents

[Introduction 3](#_Toc466390128)

[Pre-requisite 3](#_Toc466390129)

[Typical test setup topology 3](#_Toc466390130)

[Setup Steps 3](#_Toc466390131)

[Testbed setup 4](#_Toc466390132)

[Ansible based Hadoop installation-User Guide 4](#_Toc466390133)

[Pre-requisite 4](#_Toc466390134)

[User inputs before running the script on Ansible node: 4](#_Toc466390135)

[Running the Ansible script: 5](#_Toc466390136)

[Autotestbed Application setup 6](#_Toc466390137)

[Splunk Application setup 6](#_Toc466390138)

[Collectd Installations In all nodes 10](#_Toc466390139)

[Collectd Configurations 10](#_Toc466390140)

[Logstash Installation 11](#_Toc466390141)

[Performance Monitoring 12](#_Toc466390142)

[Analytics application 12](#_Toc466390143)

[Main Dashboard 12](#_Toc466390144)

[Additional dashboards 12](#_Toc466390145)

[How to use Analytics App 12](#_Toc466390146)

[Performance details 13](#_Toc466390147)

[How to analyze the data. 14](#_Toc466390148)

# Introduction

Automated test bed project is a set of scripts, and Splunk based custom application for collection and analysis of performance metrics of various components of system. This project includes scripts for automation of setting up of Hadoop and running Hadoop based workload in a test setup. The collected data can be analyzed for the performance impact of running large Hadoop workload, on servers, and switches.

## Pre-requisite

* Network infrastructure with switches is setup
* Bare metal servers are setup and Ubuntu Linux is installed on these servers.

## Typical test setup topology

* Ansible
* Splunk/Elasticsearch
* Logstash
* Collectd
* N9K poller

Compute013

Compute015

Compute24

Compute023

Compute019

Compute021

Compute022

Compute014 192.168.101.14/ 10.23.248.38

Lacrosse-1 10.23.248.88

Lacrosse-2 10.23.248.90

1. Hadoop

2. Collectd

## Setup Steps

Below is a summary of steps in involved in automated test setup for Hadoop based workload.

* Setup infrastructure specifically, switches, server (with Linux OS), VM with Splunk, A VM for running Ansible and collector
* Setup Hadoop by using Ansible scripts
* Build Splunk app by running build.py
* Install Splunk app on Splunk server and restart Splunk
* Configure collectd-client.conf with collectd server/ Splunk server address and update these on the nodes from where metrics are collected.
* Add a Splunk Job from application dashboard and monitor the performance counters. Repeat test multiple times to take reasonable error free values.

# Testbed setup

## Ansible based Hadoop installation-User Guide

Ansible scripts for Hadoop installation are located under Hadoop installation directory. Copy these scripts to Ansible node (Node which has Ansible installed).

### Pre-requisite

1. SSH key from Ansible node should be copied from Ansible node to the cluster nodes.

### User inputs before running the script on Ansible node:

1. /etc/ansible/hosts files has to be updated with list of nodes for each service.
2. Update the hosts file with list of IP addresses and hostnames
3. Update settings.cfg file with network and hostname.
4. Subnet has to be given as an input to prepare\_nodes.yaml
5. Cluster\_name and subnet has to be given as input for Hadoop\_install.yaml for mapr installation.

/etc/ansible/hosts file will have the following format.

[allnodes] #Should have all node IP addresses

192.168.101.13

192.168.101.15

192.168.101.19

192.168.101.21

192.168.101.22

192.168.101.23

192.168.101.21

[ansible-node]

192.168.101.14

[mapr-cldb]

192.168.0.13

[mapr-zookeeper]

192.168.101.13

192.168.101.15

192.168.101.19

[mapr-nfs]

192.168.101.13

[mapr-webserver]

192.168.101.13

192.168.101.15

192.168.101.19

[mapr-fileserver]

192.168.101.13

192.168.101.15

192.168.101.19

192.168.101.21

192.168.101.22

[mapr-jobtracker]

192.168.101.13

[mapr-resourcemanager]

192.168.101.13

192.168.101.15

192.168.101.19

[mapr-historyserver]

192.168.101.21

[mapr-nodemanager]

192.168.101.21

192.168.101.22

192.168.101.23

192.168.101.24

[mapr-tasktracker]

192.168.101.13

192.168.101.21

192.168.101.22

192.168.101.23

192.168.101.24

[squid3]

[ntp-server]

User should enter the list of IP’s where each service should run.

### Running the Ansible script:

1. Following script is used to install the pre-requisites

ansible-playbook prepare\_node.yaml --extra-vars="subnet=192.168.0.0/16"

1. Following script is used to install mapr packages

ansible-playbook hadoop\_install.yaml --extra-vars="subnet=192.168.0.0/16 cluster\_name=new1\_cluster"

1. Following script is used to add the license to the cluster

ansible-playbook add\_license.yaml --extra-vars=”path1=/home/cisco/req\_test cluster\_name=new1\_cluster”

1. Following script is used to validate the cluster after license is added

ansible-playbook validate.py

1. Following command is used cleanup/uninstall mapr packages on all the nodes

ansible-playbook hadoop\_cleanup.yaml

**Test Environment:**

These scripts are tested on a 2 node setup with Ubuntu as an operating system.

# Autotestbed Application setup

1. Download project files from github <https://github.com/pramurthy/autotestbed>
2. cd autotestbed-master
3. Do splunk application setup as below.

# Splunk Application setup

* Build custom Splunk application by running python script build.py. This script creates MonitoringSolutions.tar.gz
* Copy MonitoringSolutions.tar.gz to Splunk server.
* Untar the application under, /opt/splunk/etc/apps directory.
* cd /opt/splunk/etc/apps/MonitoringSolutions/bin.
* Copy all python files(\*.py) from github path autotestbed-master/collectd/collect\_pplugins/
* cd /opt/splunk/etc/MonitoringSolutions/bin.
* Open globals.py in editing mode and specify the log\_stash\_file\_name.(eg. /tmp/autotestbed7nodes.json)
* Restart splunk. Below is sample commands and output

cd /opt/splunk/etc/apps

sudo tar -zxf MonitoringSolutions.tar.gz

sudo /opt/splunk/bin/splunk restart

splunkd 32516 was not running.

Stopping splunk helpers...

Done.

Stopped helpers.

Removing stale pid file... done.

splunkd is not running.

Splunk> Take the sh out of IT.

Checking prerequisites...

Checking http port [8000]: open

Checking mgmt port [8089]: open

Checking appserver port [127.0.0.1:8065]: open

Checking kvstore port [8191]: Checking configuration... Done.

Checking critical directories... Done

Checking indexes...

Validated: \_audit \_internal \_introspection \_thefishbucket history main summary

Done

open

Checking filesystem compatibility... Done

Checking conf files for problems...

Done

Checking default conf files for edits...

Validating installed files against hashes from '/opt/splunk/splunk-6.4.0-f2c836328108-linux-2.6-x86\_64-manifest'

All installed files intact.

Done

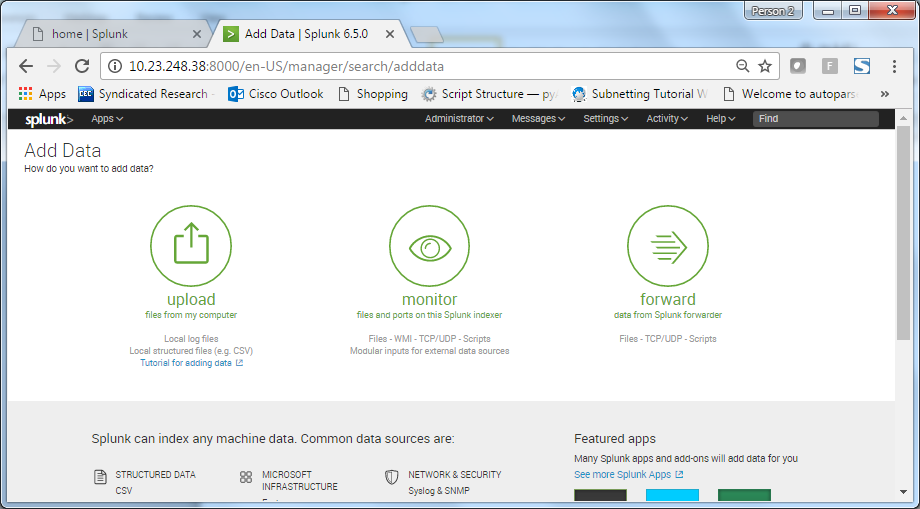
All preliminary checks passed.

Starting splunk server daemon (splunkd)...

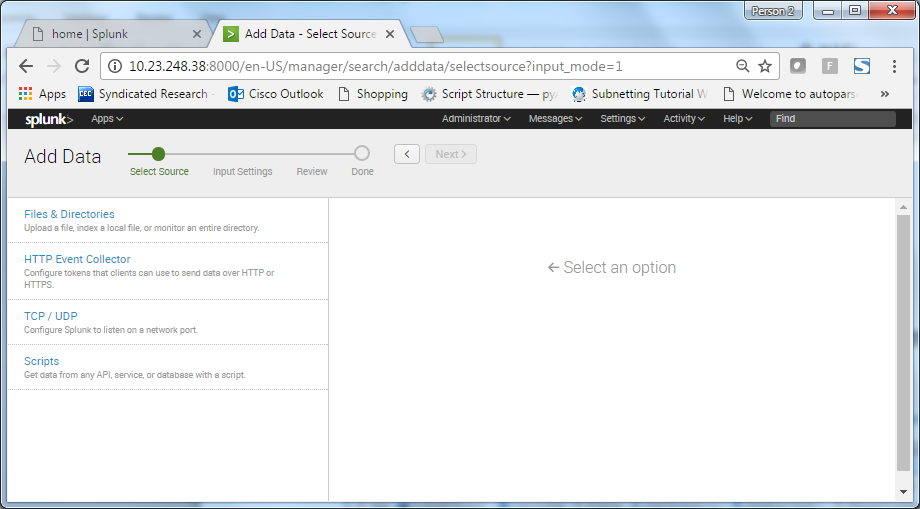
Done

Waiting for web server at http://127.0.0.1:8000 to be available.

* Open Splunk URL from browser.
* cd /tmp
* Create empty file “autotestbed7nodes.json” which shall be source file for splunkd server.
* In browser go to add data.
* Select monitoring

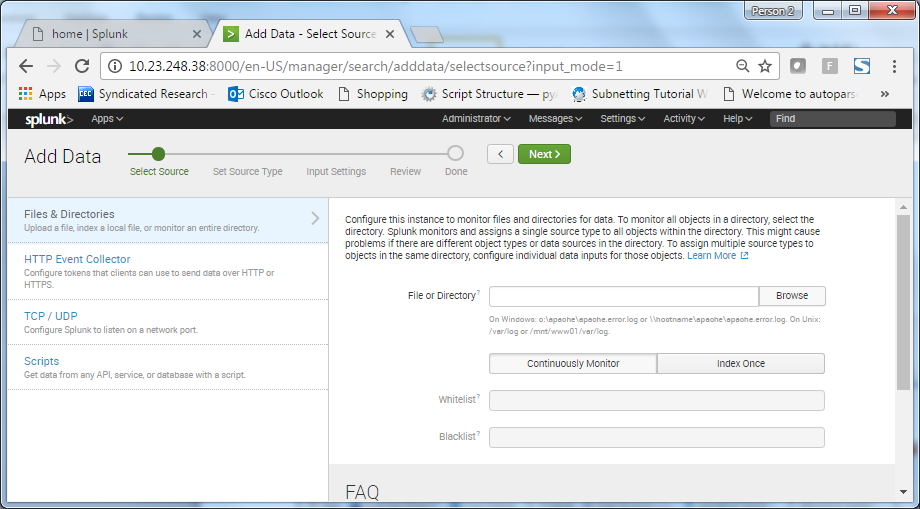


Select Files and Directories



Browse for /tmp

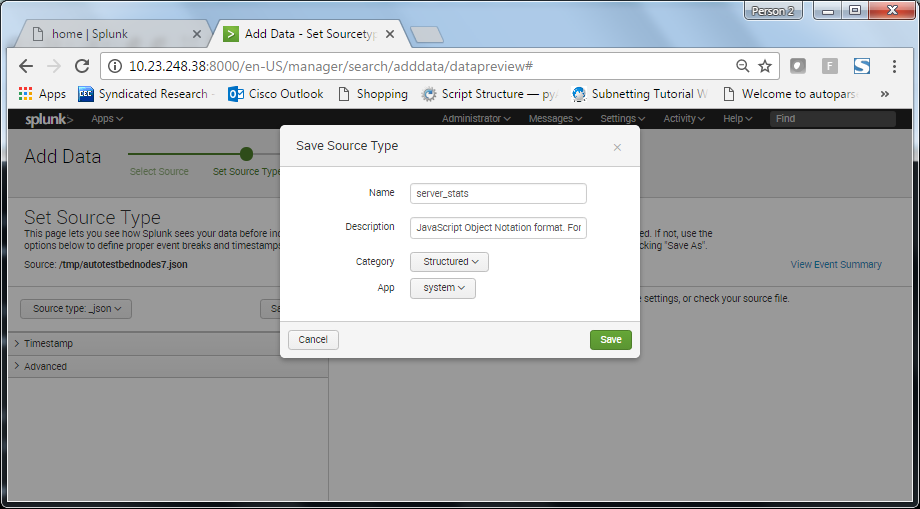
Add file name “/tmp/autotestbed7nodes.json”



Click next

Sourcetype -> saveas

Give name as server\_stats



Save and complete the steps

Once collectd is running on both server and clients, newfile.txt on server will be created.

(/opt/splunk/etc/apps/MonitoringSolutions/bin/newfile.txt)

Repeat above steps to add newfile.txt in splunk.

* Analytics application is listed on the left hand side.
* Click the application link and the dashboard comes up.
* Navigate the dashboard to start the test, view test metrics.

# Collectd Installations In all nodes

1. sudo apt-get update
2. sudo apt-get install g++
3. sudo apt-get install build-essential
4. cd /tmp
5. sudo wget <https://storage.googleapis.com/collectd-tarballs/collectd-5.6.1.tar.bz2>
6. sudo tar xf collectd-5.6.1.tar.bz2
7. cd collectd-5.6.1/src
8. open interface.c in editing mode (vi interface.c) and compare with autotestbed-master/collectd/collect\_src/collectd-5.5.0/src/interface.c
9. Do the necessary changes in /tmp/collectd-5.6.1/src/interface.c as per autotestbed-master/collectd/collect\_src/collectd-5.5.0/src/interface.c
10. cd /tmp/collectd-5.6.1
11. sudo ./configure
12. sudo make all install

# Collectd Configurations

1. **On server node(192.168.101.14/compute014)**
2. cd /opt/collectd/etc
3. open collectd.conf in editing mode (vi collectd.conf) and compare with autotestbed-master/collectd/collectd\_server.conf
4. Do the necessary changes in collectd.conf as per collectd\_server.conf
5. **On client nodes(compute013, compute015, compute019, compute021, compute022,compute023, compute024 )**
6. cd /opt/collectd/etc
7. open collectd.conf in editing mode(vi collectd.conf) and compare with autotestbed-master/collectd/collectd\_client.conf
8. Do the necessary changes in collectd.conf as per collectd\_client.conf

# Logstash Installation

* sudo wget <https://download.elastic.co/logstash/logstash/logstash-all-plugins-2.4.0.tar.gz>
* tar xvzf logstash-all-plugins-2.4.0.tar.gz –C /opt
* cd /opt/logstash-2.4.0/bin
* mkdir conf.d
* copy all files(\*.conf) from autotestbed-master/logstash to /opt/logstash-2.4.0/bin/conf.d/
* cd /opt/logstash-2.4.0/bin/conf.d/
* open output.conf in editing mode and specify the path.

(eg. /tmp/autotestbed7nodes.json)

# 

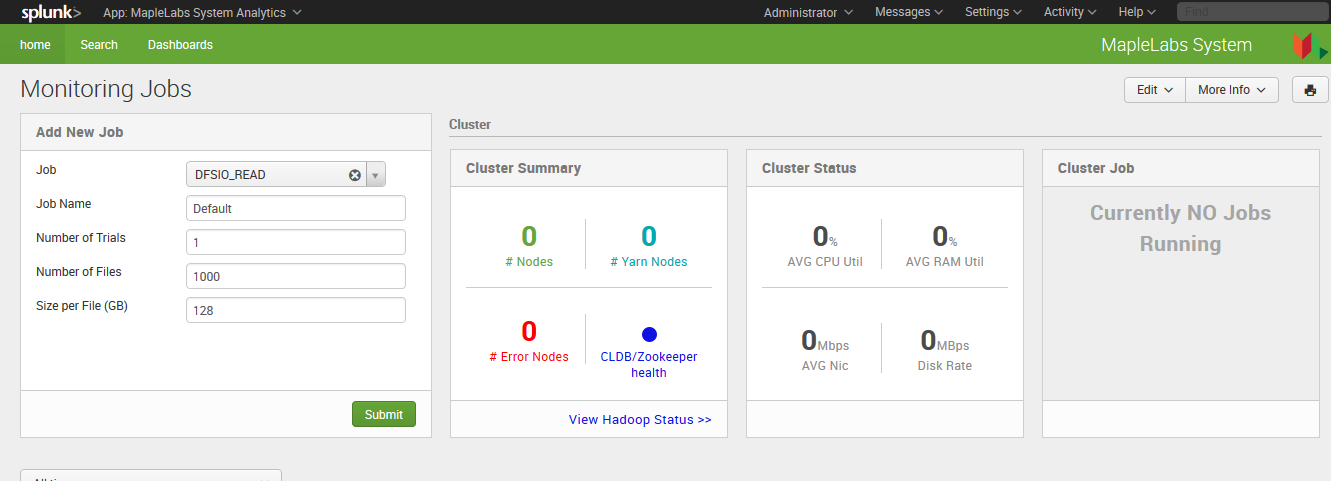
# Performance Monitoring

## Analytics application

Once you setup infrastructure, application and Splunk servers. Visit the Splunk application URL,

http://<splunk server url>

On the left side select Analytics application. This will load the main dashboard of the application.



## Main Dashboard

Main dashboard helps you to start a new job. Standard Hadoop jobs are prepopulated. This also has sections for providing information about the cluster summary and details, let of jobs running, a table consisting of job details.

## Additional dashboards

Select the link Dashboards to see other dashboards. These dashboards provides performance metrics for servers, switches and Hadoop job.

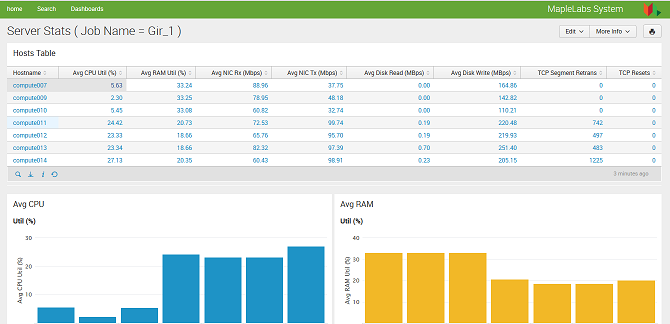
## How to use Analytics App

* Go to Main dashboard.
* Make sure no jobs are running on your test bed.
* Select specific job type. These are standard Hadoop jobs. Below are the job types.
  + Terragen
  + Terrasort
  + DFSIO Read
  + DFSIO write
* Provide a name to job.
* Number of trails to run
* Provide job specific configuration such as file size, number of files to create.
* Submit
  + Once job is submitted, job is executed on the test setup. The performance metrics are collected as the job executes.

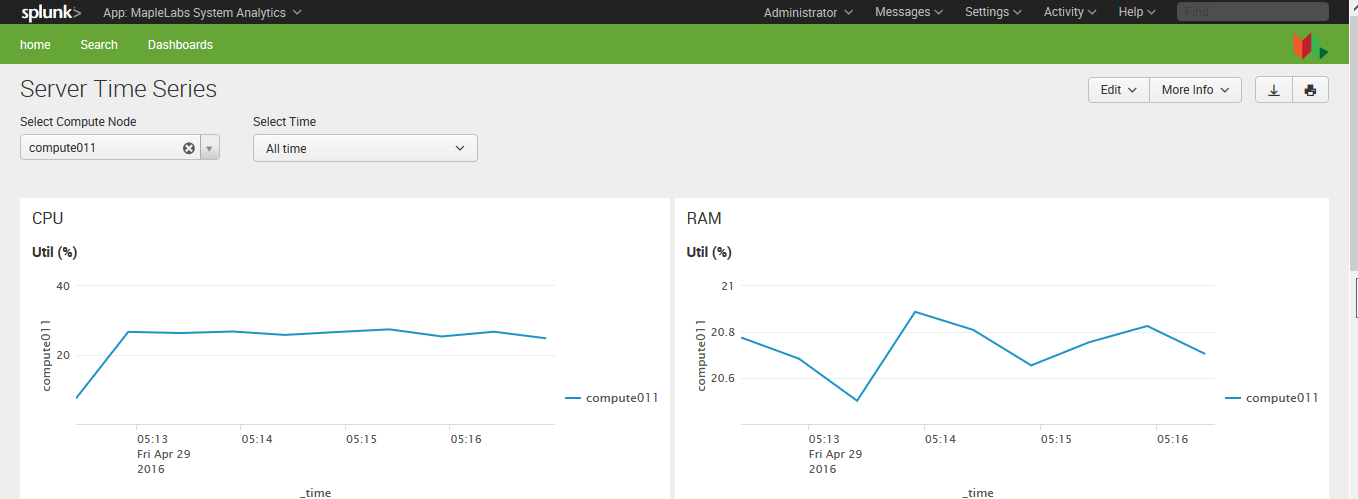
### Performance details

Job details table provides summary information about the jobs, job status, completion time, total maps, total reduces, average CPU utilization, Average RAM utilization, Average disk utilization, Average NIC RX and TX.

Select link server status to view the summary of all servers. Following is a sample server summary.



Click specific server to view server specific metrics. Below is a sample charts for a server.



Similarly select as switch to view switch performance metrics.

From the cluster summary section, click on view Hadoop stats to view Hadoop specific performance metrics.

## How to analyze the data.

* Start a Hadoop job as explained in previous section.
* View the cluster summary and cluster status.
  + If there are errors or non uniform utilization of CPU, memory or network, then view the status of servers, switches to make sure there is uniform load across all nodes.
  + View the server/switch parameters, specifically, CPU utilization, RAM utilization, Network utilization, switch buffer utilization to analyze which parameter is becoming bottle neck
  + Run the test multiple times to collect data and arrive at average values for benchmarking results.