

v1.1

Analytics Software

Installation Procedure

1.1

June 28, 2011

Copyright © 2011 SmithMicro®.

2023 Stierlin Ct, 2nd Floor

Mountain View, CA 94043

Voice 🞟 650.603.6600

Fax 🞟 650.603.6635

[support@smithmicro.com](mailto:support@smithmicro.com)

[www.smithmicro.com](http://www.smithmicro.com/)

**This document is the property of SmithMicro Software, Inc, Mountain View, California. It contains confidential information and may be used only for the express purpose defined at the time of delivery and only by the individual to whom it has been provided.**

Revision History

|  |  |  |
| --- | --- | --- |
| Date | Author | Comments |
| 06/03/11 | Luis Ramos | Initial Skeleton |
| 06/14/11 | Andy Kim | Add Logger and Portal install sections |
| 06/23/11 | Luis Ramos | Added Admin install sections. Added some validation under each install section. Removing chapter 4 validation & error to just common error. |
| 06/23/11 | Luis Ramos | Added example section with examples on how to load data and samples. |
| 06/28/11 | Luis Ramos | More validations and common errors. |
| 07/21/11 | Andy Kim | Update analytics\_bundle.sh description. |

Table of Contents

[1. Introduction 6](#_Toc296610330)

[1.1 Prerequisites 6](#_Toc296610331)

[1.1.1 Networking 6](#_Toc296610332)

[1.2 Server Types 6](#_Toc296610333)

[1.2.1 Masternode Server 6](#_Toc296610334)

[1.2.2 Slavenode Server 6](#_Toc296610335)

[1.2.3 Datanode Server 6](#_Toc296610336)

[1.2.4 Loggernode Server 7](#_Toc296610337)

[1.2.5 Webnode Server 7](#_Toc296610338)

[1.2.5.1 Admin Web Services 7](#_Toc296610339)

[1.2.5.2 Portal User Interface 7](#_Toc296610340)

[2. Software Installation 9](#_Toc296610341)

[2.1 About Puppet 9](#_Toc296610342)

[2.2 Getting Puppet to Install Analytics 9](#_Toc296610343)

[2.2.1 Download Installation Release 9](#_Toc296610344)

[2.2.1.1 System Config Installation Explained 9](#_Toc296610345)

[2.2.2 Upload to Analytics Server 9](#_Toc296610346)

[2.2.3 Bootstrap the Server 10](#_Toc296610347)

[2.2.3.1 Bootstrap Validation 10](#_Toc296610348)

[2.3 Installing Analytics Server Types 10](#_Toc296610349)

[2.3.1 Installing the Masternode 11](#_Toc296610350)

[2.3.1.1 Masternode Validation 12](#_Toc296610351)

[2.3.2 Installing the Slavenode 12](#_Toc296610352)

[2.3.2.1 Slavenode Validation 12](#_Toc296610353)

[2.3.3 Installing the Loggers 13](#_Toc296610354)

[2.3.3.1 Logger Verification 13](#_Toc296610355)

[Currently Showing: Most Recent 50 Lines 14](#_Toc296610356)

[2.3.4 Installing the Portal 14](#_Toc296610357)

[2.3.5 Installing the Admin Web Service 14](#_Toc296610358)

[2.3.5.1 Admin Verification 15](#_Toc296610359)

[3. Example Installations 17](#_Toc296610360)

[3.1 Development using Virtual Machines 17](#_Toc296610361)

[3.1.1 About 17](#_Toc296610362)

[3.2 Uploading Data to the Loggers 17](#_Toc296610363)

[3.2.1 About 17](#_Toc296610364)

[3.3 Loading Logger Data into Hive Database 17](#_Toc296610365)

[3.3.1 About 17](#_Toc296610366)

[3.3.1.1 Loading Pre-Populated Sample Data. 17](#_Toc296610367)

[4. Common Errors 20](#_Toc296610368)

[4.1 How to validate an Installation 20](#_Toc296610369)

[4.1.1 Look at individual sections under the Installation Chapter. 20](#_Toc296610370)

[4.2 Common Errors 20](#_Toc296610371)

[4.2.1 Java Dependencies 20](#_Toc296610372)

[5. Glossary 21](#_Toc296610373)

[5.1 Glossary 21](#_Toc296610374)

List of Figures

[Figure 1. Process - 1 **Error! Bookmark not defined.**](#_Toc220462051)

|  |
| --- |
| **Chapter 1** - Introduction |

# Introduction

## Prerequisites

Before continuing please refer to the Analytics Server Pre-installation Procedures:  
[AnalyticsPreInstallProcedure\_v1 1.pdf](file:///C:\Users\lramos\Documents\SMSI\Corporate\Analytics\Analytics_Software_Installation_Procedure_1.1.docx)

After completing the pre installation, you will end up with two different Analytics Server Hardware Configurations (**Processing** Server & **Logging** Server). However, it is important to note, that if absolutely necessary each of these configurations can support the different server types in the next section. Nonetheless you should follow the recommended configurations noted on the following section (1.2).

### Networking

All the servers in the cluster need to have proper hostnames (refer to Configuration Section). Each server also requires to be in the same domain as the rest of the cluster, each server should have the IP/Hostname of all other servers in it’s /etc/hosts files.

## Server Types

The following section explains the different servers used in Analytics and how collectively the servers form a harmonious Logging, Processing, and Service Cluster. Some servers may combine different types. Technically each server type can be installed on a single machine, but this is not functionally recommended.

**Minimum Cluster: 1 Masternode, 1 Slavenode, 2 Loggernode/Datanode**

### Masternode Server

Manages the Processing Servers, there can only be one masternode in a cluster and includes a secondary fail over node, as well as task and data nodes.

Analytics Pre-Install Hardware Configuration: **Processing**

### Slavenode Server

Part of the Processing Servers, a slavenode includes task and data nodes.

Analytics Pre-Install Hardware Configuration: **Processing**

### Datanode Server

Part of the Processing Servers, a datanode includes only the data node.

Analytics Pre-Install Hardware Configuration: **Any**

### Loggernode Server

The logger nodes accept events from analytic enabled devices. It is usually connected to the cluster via a data node.

Analytics Pre-Install Hardware Configuration: **Logging**

### Webnode Server

The web servers serve the backend admin and portal services.

Analytics Pre-Install Hardware Configuration: **Processing**

#### Admin Web Services

Analytics Backend API Service, this is required by the portal to communicate with the cluster.

#### Portal User Interface

Application for creating reports, charts and dashboard. As well as managing users, roles and data settings.

More details on each server type will be provided in chapter 2.

|  |
| --- |
| **Chapter 2** - Software Installation |

# Software Installation

## About Puppet

Puppet is a system for automating system administration tasks. We also use puppet as an automated installation tool for the analytics software. Currently our puppet system is use in standalone mode, meaning a copy of the installation package must be uploaded to each server.

More about Puppet: <http://docs.puppetlabs.com/guides/faq.html>

## Getting Puppet to Install Analytics

### Download Installation Release

You may download a release/build of the Analytics System Config (This will either be on Hudson or a Download Provider). Or you may also export a copy of the latest working directory or a tag using SVN:

**Trunk:**  
$ svn export <https://mmeng.smithmicro.net/repos/main/analytics/trunk/system_config>

**Tag:**  
$ svn export <https://mmeng.smithmicro.net/repos/main/analytics/tags/system_config-1.1>

#### System Config Installation Explained

The analytics system config includes two parts.

* “Bootstrap” – The scripts in this directory should be the first thing installed on a new system; Installs Ruby & Puppet and also updates the server’s sources for packages.
* “Puppet” – The scripts in this directory are the manifests used by analytics to install the different server types.

### Upload to Analytics Server

The downloaded Analytics System Config Release must be uploaded to every Server that will be part of the cluster. The user that will install the software must either be Root or have Root Privileges (sudo).

A common way of copying a file/directory securely to a remote server:

$ scp system\_config-1.1.tar.gz root@remote.location:

You may also use tools like “wget” from the server if you know the download target.

### Bootstrap the Server

Each analytics server is required to be “bootstrapped”. Go to the directory of the uploaded Analytics System Config installation and then into the “bootstrap” folder. Usually this means using SSH to securely access the server. (You may need to unpackage the file if it is a tarball).

Assuming you have a file on the server named “system\_config-1.1.tar.gz”:

$ tar –xzf system\_config-1.1.tar.gz  
$ cd system\_config-1.1/  
$ cd bootstrap/  
(assuming you are on a debian system with apt and a user with root privileges)  
# bash ./install.sh

#### Bootstrap Validation

After the bootstrap is finished you should be able to execute the following:

$ /usr/bin/ruby –v  
 ruby 1.8.7 ….  
$ which puppet  
 /usr/bin/puppet  
$ /usr/bin/gem list  
 facter (…)  
 puppet (…)

Puppet is now installed and the next phase is to install the “Server Types” in section 2.3.

## Installing Analytics Server Types

**Requires: Puppet (look at section 2.2)**

The order you install the server types is usually important. For example, a slavenode and datanode, depend on a masternode, you should not install any type of node without first installing a masternode. If there is already a masternode present then additions are okay.

Each installation is also dependent on an environment. Currently we have several environments depending on your goal. Each may have unique settings/configurations.

Development: “dev”  
Production: “production”

Puppet installation scripts directory:

$ cd system\_config-1.1/  
$ cd puppet/  
$ cd scripts/

**C:\Program Files (x86)\Microsoft Office\MEDIA\CAGCAT10\j0293236.wmf** The following is a recommended installation example on a cluster of 4:

Processing Server 1: Masternode  
Processing Server 2: Slavenode, Portal, Admin Web Service  
Logger Server 1: Datanode & Loggernode  
Logger Server 2: Datanode & Loggernode

Each Server is a unique machine, in this example, you first install the Masternode, and then proceed with the rest of the server node types.

**C:\Program Files (x86)\Microsoft Office\MEDIA\CAGCAT10\j0293236.wmf** More about Analytics Environments:

During the installation process you may choose to install any of the server types under a certain environment. Each environment is recommended for different objectives:

**Development** (code: “dev”) – This should be used when attempting to install the analytics bundle for development/testing purposes. It uses less memory, and selects a single data location for tasks.  
**Staging** (code: “staging”) – This should be used when installing analytics bundle on hardware that is similar to that of production hardware. It shares some of the performance settings from production.  
**Production** (code: “bakrire” and “sprintcm”) – This should only be used when installing analytics bundle for one of these clients. The settings are configured to their specific hardware, you may also add remove nodes\*.

\* For more details, refer to nodes.pp under puppet/manifests.

### Installing the Masternode

|  |  |
| --- | --- |
| Detail | Description |
| Script | analytics\_bundle.sh |
| Environments | “dev”, Production: “bakrie”, “sprintcm” |
| Command | $ ./analytics\_bundle.sh –i masternode –e [ENV] [ENV] = “dev”, “production” |
| Post Install Components | Namenode, SecondaryNameNode, JobTracker, Tasktracker, Datanode, Hadoop, Hive, MySQL, FTP, RabbitMQ, JAVA, Python. |

To install the masternode, you simply initiate the script “analytics\_bundle.sh” (under system\_config-1.1/puppet/scripts/) with the desired arguments for your server type and environment.

All process of the installation is displayed as standard output in the user’s console. Please pay particular attention to Warnings or Errors, usually distinguished by Magenta colored text.

If something goes wrong, look at the validation section or common error procedures. Some errors are safe to ignore, usually errors telling you a file or directory already exists.

#### Masternode Validation

Once masternode installation is finished you should be able to execute the following (under a root user) with similar output:

# jps

6960 **NameNode**  
7291 **TaskTracker**   
7120 **DataNode**   
7043 **SecondaryNameNode**  
7369 **JobTracker**  
23250 Jps

This assures that each individual java service is running on the server. A masternode consists of a NameNode, SecondaryNameNode, JobTracker, DataNode and TaskTracker.

**Verification URLs**:  
http://<masternode-host>:50030/jobtracker.jsp  
http://<masternode-host>:50070/dfshealth.jsp   
You should be able to see the page load and give initial information, including 1 node.

#### Masternode Validation

Check HDFS Folders:  
“sudo –u hadoop hadoop dfs –lsr /cache/” should output:  
/cache/geotagger  
/cache/geotagger/\_\_init\_\_.py  
/cache/geotagger/geolocation.py  
/cache/geotagger/hook.py

### Installing the Slavenode

|  |  |
| --- | --- |
| Detail | Description |
| Script | analytics\_bundle.sh |
| Environments | “dev”, “production” |
| Command | $ ./analytics\_bundle.sh –i slavenode –e [ENV] [ENV] = “dev”, “production” |
| Post Install Components | Tasktracker, Datanode, Hadoop, MySQL, JAVA, Python. |

To install the masternode, you simply initiate the script “analytics\_bundle.sh” (under system\_config-1.1/puppet/scripts/) with the desired arguments for your server type and environment.

All process of the installation is displayed as standard output in the user’s console. Please pay particular attention to Warnings or Errors, usually distinguished by Magenta colored text.

If something goes wrong, look at the validation section or common error procedures. Some errors are safe to ignore, usually errors telling you a file or directory already exists.

#### Slavenode Validation

Once slavenode installation is finished you should be able to execute the following (under a root user) with similar output:

# jps  
7291 **TaskTracker**   
7120 **DataNode**   
23250 Jps

This assures that each individual java service is running on the server. A slavenode consists of a DataNode and TaskTracker.

**Verification URLs**:  
http://<masternode-host>:50030/jobtracker.jsp  
http://<masternode-host>:50070/dfshealth.jsp   
You should be able to see the page load and aggregate the nodes information, you should also see more than one node total.

### Installing the Loggers

To install the loggers, you simply initiate the script “analytics\_logger\_bakrie.sh” (under system\_config-1.1/puppet/scripts/). Su (Sudo) privileges are required.

|  |  |
| --- | --- |
| Detail | Description |
| Script | analytics\_bundle.sh |
| Environments | All-In-One |
| Command | $ ./analytics\_bundle.sh –i logger –e [ENV] |
| Post Install Components | Apache, Passenger, Logger, … |

$ ./analytics\_logger\_bakrie.sh

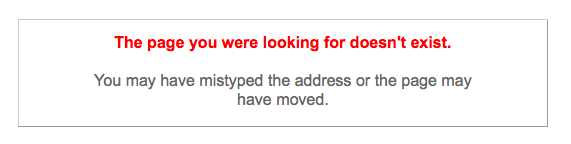
#### Note: You Will Also Want to Install a Datanode for Logger Features.

#### Logger Verification

Verify that there are no errors in the Puppet output. Then open the logger URLs in a web browser to make sure the logger instances are running:

http://<logger1>/events

http://<logger2>/events

You should see the following error page.You may also verify the administrative pages for each logger:

http://<logger1>:8080/admin/events  
http://<logger2>:8080/admin/events

You should see a mostly blank page with the following message:

### Currently Showing: Most Recent 50 Lines

-Top of Form



Bottom of Form

### Installing the Portal

**C:\Program Files (x86)\Microsoft Office\MEDIA\CAGCAT10\j0293236.wmf** Portal and Admin Web Services Note:  
The portal can be used with or without an admin web service running. However the portal will only be able to use data it has already populated, or switch to use local files. Without the Admin Web Services (section 2.3.5) the portal will not be able to communicate with the masternode cluster.

|  |  |
| --- | --- |
| Detail | Description |
| Script | analytics\_bundle.sh |
| Environments | All-In-One |
| Command | $ ./analytics\_bundle.sh –i portal –e [ENV] |
| Post Install Components | Apache, Passenger, Portal, … |

To install the Portal UI, run the script “analytics\_portal.sh” on the same server as Admin Web Service. This is typically the Slavenode server. This should have already created your MySQL database user (analytics\_portal) and schema (analytics\_portal) for you. Once this process completes, go to your Portal root and import data into your database. Note this is ONLY for a NEW INSTALL.

$ cd /var/www/analytics\_portal/bakrie/current  
$ bundle install  
$ RAILS\_ENV=production bundle exec rake db:drop db:create db:migrate db:seed  
$ RAILS\_ENV=production bundle exec rake app:demo\_users

$ RAILS\_ENV=production bundle exec rake db:data:load\_dir

(If Error, PATH: /opt/ruby-enterprise-1.8.7-2011.03/bin/)

Verify your instance by hitting the Portal URL:

http://<slavenode>/

You should see the login page for the Portal.

#### Portal Common Errors

Try “/etc/init.d/apache2 restart” this sometimes can give you more clues. If you see an error like:  
“Could not load mod\_passenger.so”  
Solution:  
“gem uninstall passenger”  
“re-run portal installer”

OR  
  
“/opt/ruby-enterprise-1.8.7.2011.03/bin/passenger-install-apache2-module –a”  
“/etc/init.d/apache2 restart”

Error: 000-default, Message: “It Works”.  
Solution: delete the file in /etc/apache2/sites-enable/000-default and restart apache “/etc/init.d/apache2 restart”

### Installing the Admin Web Service

|  |  |
| --- | --- |
| Detail | Description |
| Script | analytics\_bundle.sh |
| Environments | All-In-One |
| Command | $ ./analytics\_bundle.sh –i admin –e [ENV] |
| Post Install Components | Tomcat, JRuby Binaries, Admin, … |

To install the Admin Web Service, run the script “analytics\_admin.sh” on the same server as the Portal. This is typically the Slavenode server. This should have already created your MySQL database user (analytics\_admin) and schema (analytics\_admin) for you. Once this process completes, tomcat should have deployed the admin application. Simply run the following in the terminal:

$ ./analytics\_admin.sh

#### Admin Verification

To verify the admin is properly installed and running. You should be able to navigate to the following pages:

http://<slavenode>:3001/  
You should see:  
**{"java\_version":"1.6.0\_24-b07"}**

http://<slavenode>:3001/analytics  
You should see:  
**Java Classpath - …  
Java xBootclasspath - …  
Java Tomcat Classpath - …**

http://<slavenode>:3001/hive  
You should see:  
**Hive Service Online: true**

#### Admin Common Errors

If verification fails, check the tomcat logs: /srv/tomcat/analytics\_admin/logs/catalina.out.

In most cases restarting tomcat fixes things: /etc/init.d/tomcat-analytics\_admin restart

Other error is a servletcontext error, delete/move java servlet-api jars from /usr/share/java/

Another thing to watch is for the WAR file to expand correctly in tomcat, verify you see JAR files in /srv/tomcat/analytics\_admin/webapps/analytics-admin-trunk/WEB-INF/lib/

|  |
| --- |
| **Chapter 3** - Example Installations |

# Example Installations

## Development using Virtual Machines

### About

## Uploading Data to the Loggers

### About

This section is about how data gets uploaded to the logger servers, please refer to the SmithMicro wiki pages:

<https://wiki.smithmicro.net/display/ALYT/Analytics+Logger>

## Loading Logger Data into Hive Database

### About

This section describes how data that has been logged by clients can be manually pushed into the Hive Masternode Database. Under normal circumstances the data is automatically loaded/pushed into the database on a day by day basis (configurable). The data being populated by the logger is generally a file named events.log or similar.

#### Loading Pre-Populated Sample Data.

**C:\Program Files (x86)\Microsoft Office\MEDIA\CAGCAT10\j0293236.wmf** Warning:  
Before continuing with loading data into the system, make sure you have at least one (1) masternode and one (1) slavenode in the cluster. If you have followed the installation procedures, you should already have this.

To load sample data you much log into the masternode using either a super user, or hadoop. You may also use SUDO command to run as other users (as per examples).

Once inside the system you must first “upload” the data into the directory belonging to the customer or environment. For example, using the environment “bakrie” you will have a directory: “/user/hadoop/bakrie”, this is not a File System Directory, rather, it is in HadoopDFS. You may accomplish this with the following command:

$ sudo -u hadoop hadoop dfs -put $ANALYTICS\_LIB/sample\_data/events.log.bakrie.custom /user/hadoop/bakrie

What this is doing is, using user hadoop (sudo –u hadoop), run the hadoop –put command with the following destination (data) and target (/user/hadoop/bakrie). You may verify the data is there by running the hadoop –ls command:

$ sudo -u hadoop hadoop dfs -ls /user/hadoop/bakrie/

drwxr-xr-x - hadoop hadoop 0 2011-05-28 /user/hadoop/bakrie/events.log.custom

Once you verify the data file is in the appropriate directory, you can manually ask hadoop to run the daily expander process:

$ sudo -u hadoop bash $ANALYTICS\_LIB/scripts/run\_daily\_expander.sh "\*.log\*" bakrie "bakrie-test1" "your-email@smithmicro.com"

-- Start run\_daily\_expander.sh --------------------------------------  
LOGFILE: \*.log\*  
GROUPID: bakrie  
REQUESTID: bakrie-test1  
MAILTO: [your-email@smithmicro.com](mailto:your-email@smithmicro.com)  
---------------------------------------------------------------------  
…  
11/06/23 16:30:26 INFO streaming.StreamJob: map 0% reduce 0%  
11/06/23 16:30:40 INFO streaming.StreamJob: map 2% reduce 0%  
…  
11/06/23 16:35:26 INFO streaming.StreamJob: map 100% reduce 99%  
11/06/23 16:35:40 INFO streaming.StreamJob: map 100% reduce 100%

The command is asking the server to process the files for the directory “bakrie” that match the pattern “\*.log\*”, this process can take a very long time, depending on the data being processed, it is normally about 30-60 seconds per date/event combination.

When the expansion process is done, all the data should be available on the database (data gets loaded into the database in increments). To verify if there is data in the database you may run hive:

$ sudo -u hadoop hive –e “use bakrie; select \* from logevents limit 50;”

You should see data in the console output. This data is also available from the portal and admin web services.

|  |
| --- |
| **Chapter 4** - Common Errors |

# Common Errors

## How to validate an Installation

### Look at individual sections under the Installation Chapter.

## Common Errors

### Java Dependencies

**Error Description**:

Java Dependencies (sun-java6-bin, sun-java6-jre, sun-java6-jkd) will sometimes require a user to accept the agreement, and cause puppet to fail.

**Actual Error:**

Package[sun-java6-bin]/ensure: change from purged to present failed.  
Package[sun-java6-jre]/ensure: change from purged to present failed.  
Package[sun-java6-jdk]/ensure: change from purged to present failed.

**Solution**:

The solution is printed on the screen. (apt-get –f install) to gather the correct accept agreement plugins. Then re-run puppet.

|  |
| --- |
| **Chapter 5** - Glossary |

# Glossary

## Glossary