# National MOTECH System (NMS)

Software Deployment-cum-Health Monitoring Strategy Document

# Revision History

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| # | Version | Date | Owner | Notes |
| 1. | 0.1 | 6/5/2015 | Manjot Singh | First version, Chapter 1 and Chapter 2 |
| 2 | 0.2 | 7/5/2015 | Abhishek | Added Chapter 3 |
| 3 | 0.3 | 8/5/2015 | Manjot Singh | Updated as per comments. |
| 4 | 0.4 | 11/5/2015 | Manjot Singh | Updated as per review comments |
| 5 | 0.5 | 27/5/2015 | Abhishek Srivastava | Appendix added for POC done with Nagios |
| 6 | 0.6 | 28/5/2015 | Manjot Singh | Updated Deployment document |
| 7 | 0.7 | 5/6/2015 | Manjot Singh | Document updated as per comments. |
| 8 | 0.8 | 11/6/2015 | Manjot Singh | Added complete System halt approach. |
| 9 | 1.0 | 12/6/2015 | Manjot Singh | Final Version |

|  |
| --- |
|  |

Table of Contents

[National MOTECH System (NMS) 1](#_Toc421310692)

[**Software Deployment-cum-Health Monitoring Strategy Document** 1](#_Toc421310693)

[Revision History 1](#_Toc421310694)

[Introduction 5](#_Toc421310695)

[Scope 5](#_Toc421310696)

[Assumptions 5](#_Toc421310697)

[Deployment 5](#_Toc421310698)

[Health Monitoring 5](#_Toc421310699)

[Open Issues 5](#_Toc421310700)

[Deployment 5](#_Toc421310701)

[Health Monitoring 5](#_Toc421310702)

[1. High level Architecture 6](#_Toc421310703)

[1.1 Web Tier 6](#_Toc421310704)

[1.2 Database Tier 7](#_Toc421310705)

[2. Deployment 7](#_Toc421310706)

[2.1 Roles 9](#_Toc421310707)

[2.1.1 MOTECH Webserver 9](#_Toc421310708)

[2.1.2 MOTECHdbserver 10](#_Toc421310709)

[2.1.3 Reporting Web Server 10](#_Toc421310710)

[2.1.4 ReportingDBServer 11](#_Toc421310711)

[3. Monitoring 13](#_Toc421310714)

[3.1 NMS Monitored Services 14](#_Toc421310715)

[3.2 Monitoring User 15](#_Toc421310716)

[4. Appendix 15](#_Toc421310717)

[4.1 Nagios POC 15](#_Toc421310718)

[4.1.1 Hosts Status Details For all Host Groups 16](#_Toc421310719)

[4.1.2 Service Status Details For all Hosts 16](#_Toc421310720)

[4.1.3 Service Overview for all Host Groups 17](#_Toc421310721)

[4.1.3.1 Tomcat 17](#_Toc421310722)

[4.1.3.2 Mysql 18](#_Toc421310723)

[4.1.3.3 Root Partition 18](#_Toc421310724)

[4.1.3.4 Current Users 19](#_Toc421310725)

[4.1.3.5 Total Processes 20](#_Toc421310726)

[4.1.3.6 Current Load 21](#_Toc421310727)

[4.1.3.7 Swap Usage 22](#_Toc421310728)

[4.1.3.8 SSH 23](#_Toc421310729)

# Introduction

The National MOTECH System includes set of BBC-Media applications based on MOTECH platform. This document tries to give overview of MOTECH software deployment and health monitoring strategy.

# Scope

Document covers the MOTECH software deployment and Health Monitoring. Document doesn’t cover IVR System deployment and its health monitoring. The Scope of the deployment script will be

1. Installing/Upgrading MOTECH ( Platform + NMS modules + Properties files)
2. Configuring and installing reporting server.
3. Installing SSH Keys between MOTECH and IVR
4. Installing SSL certificates.

# Assumptions

## Deployment

* There is no provision of Internet at the Railtel DC.
* Server provisioning is done before hand, i.e. all the tools installation/Configuration on NMS servers will be done at the Aricent.

## Health Monitoring

* Open source tool Nagios will be used to monitor the NMS system.

# Open Issues

## Deployment

* Reporting Tool is not yet finalized.
* It’s not yet decided on which server ETL will be hosted.

## Health Monitoring

* Shall IVR system be monitored? If yes, then
  + IVR system shall allow monitoring of its service
  + IVR system shall expose Web Service of SMS URL to be monitored.
* Default alarming level for each probe to be decided.

# High level Architecture

MOTECH WEB -1

MOTECH WEB -2

MOTECH WEB -3

MDS DB-1 ActiveMQ Broker 1

Virtual IP

Repl

Reporting WEB -1

Reporting WEB -1

Virtual IP

Report DB -1

Report DB -2

Health Monitoring

LOAD BALANCER

MDS DB-2ActiveMQ Broker2

LOAD BALANCER

Repl

ETL

Figure 1: High Level Architecture

MOTECH system consists of multiple nodes namely it can be divided into

## Web Tier

The responsibility of Web Tier can be divided in terms of interfaces which they cater.

1. IVR-MOTECH interface

MOTECH provides services to IVR using three MOTECH servers which are connected via Load Balancer, Load Balancer will schedule incoming HTTP request from IVR to any one of the three MOTECH Servers.

1. Public Reporting Interface

Reporting web servers will be configured to provide reporting services to the user in public network. Both the reporting web servers will be connected with Load balancer which will schedule load in between these servers.

## Database Tier

1. NMS-MDS Databases nodes

MDS database node provides services to MOTECH Web servers; it consists of two MOTECH (MDS) database servers which will run MySQL in MASTER-SLAVE mode.

MySQL based database will be accessible via Virtual IP (VIP) which is configured using Keepalived that provides Linux Virtual server services. The VIP will be assigned to one of the two database nodes which will be MySQL Master at that time. In case of any outage (MySQL service stop/restart, Keepalived service stop/restart) Virtual IP will be reassigned to the new active server.

ActiveMQ brokers which will be configured in JDBC Master Slave mode will also be hosted on these two servers that will be connected to MySQL data store.

1. NMS-Reporting Database nodes

Two Reporting Database servers (Master-Slave) which will provide services to Reporting Web Server and can be accessed via Virtual IP

1. ETL

ETL will be installed on MDS DB server or Reporting DB server, which will periodically extract reporting relevant information from MDS DB and populate reporting database.

# Deployment

Deployment will use Ansible which is Open source configuration management software. Ansible is agent less tool and requires one controlling server from where other nodes can be managed via SSH. Ansible remotely orchestrate the nodes which are in inventory configuration file based on the Playbooks written specific to the node. Ansible Playbooks execution is done on the basis of roles, for example webserver, mdsdbserver, reporting server etc.

In NMS, Health monitoring server shall be used as controlling server from where remote nodes shall be managed depending on their roles. Sample Directory structure of the Ansible will look like (Assuming Reporting server is Jasper)

Figure 2: Sample NMS-Deploy structure

NMS-Delivery module will be version controlled in git and will be copied to health monitoring server by VPN connectivity from Aricent.

Prerequisites

1. SSH KEYS

Ansible works on ssh( password less authentication), so before executing any playbook, ssh key of the health monitoring server will be installed on all the nodes manually.

1. SSH PORT

As per the requirement SSH on all severs will work on non-standard port, So Ansible host file will be configured to work on non-standard port.

## Roles

## MOTECH Webserver

MOTECH WEB server will host NMS application on the MOTECH Platform. Deployment script on the Web server shall be responsible for following configurations.

1. Tomcat configuration
2. server.xml, tomcat-users.xml
3. SSL Encryption certificate.
4. MOTECH configuration
5. MOTECH bootstrap configuration
6. MOTECH modules configuration files
7. NMS MOTECH configuration

MOTECH platform will be packaged in WAR file and related NMS Modules will be archived in JAR format. MOTECH war and NMS bundles will be delivered at the deployment center via Aricent VPN. Health monitoring server will be used to save MOTECH and NMS bundles where bundles will be saved with respective versions. Sample directory structure of health monitoring server will be

$REPO\_HOME/MOTECH/MOTECH{version}.war

$REPO\_HOME/NMS/MA{version}.jar

Ansible is idempotent system it will gather facts from all the hosts and will perform desired tasks only if mismatch is detected. User will have to specify MOTECH and other bundles version along with source path. NMS deployment will follow complete system halt approach, which means when ever new build upgrade is triggered all the Web servers will be stopped, So NMS services will not be accessible during this time.

Sample Playbook instructions

Variables

Motech Version

NMS modules version

Repository name

- hosts: webservers

- name: disable nagios alerts

-name: Stop Tomcat

-name: Wait for all Web server to Stop

-name: Copying of Tomcat configuration files at respective locations

-name: Creation of .MOTECH directory structure in $USER\_HOME

-name: Copying MOTECH properties to $USER\_HOME/.MOTECH/config

-name: Check if other than required MOTECH version is installed if yes perform clearance.

-name: Copying MOTECH package in $CATALINA\_HOME/webapps if required.

-name: Check if other than required NMS bundles version is installed if yes perform clearance.

-name: Copying NMS bundles in $USER\_HOME/.MOTECH/bundles if required.

-name: UnBlocking HTTP/HTTPS tomcat Ports

-name: Start Tomcat

-name: Verification of Motech services.

-name: Enable Nagios alerts

## MOTECHdbserver

MOTECH DB server will host mysql DB, keepalived and ActiveMQ. Deployment script on the db server shall be responsible for following configurations:

1. MYSQL configuration
   1. Changing ‘root’ password for mysql
   2. Adding firewall exception so that mysql is accessible from remote
   3. Creating application user for accessing DB
   4. Setting replication in master-slave mode
2. Keepalived configuration
   1. Copy keepalived configuration to servers
3. ActiveMQ configuration
   1. Copying activemq configuration to servers

Sample playbook instructions

Variables:

Application DB username and password

Replication user and password

- hosts: dbservers

-name: Set root user password

-name: Create Application DB User

-name: Add firewall rule

-name: Copying of MYSQL configuration files at respective location

-name: Start replication

-name: Copying of keepalived configuration files at respective location

-name: Copying ActiveMQ configuration files at respective location notify to restart activemq

## Reporting Web Server

Reporting Web server will be hosted on tomcat server whether its Jasper or Pentaho, so the deployment of reporting web server can be divided in two parts

1. Tomcat server deployment
2. Reporting tool deployment

Tomcat server deployment will follow same approach as mentioned in MOTECH web server, So in this section will cover reporting specific playbook requirements

Reporting Tool Deployment

Jasper deployment

Jasper web server provides functionality to export reports, data input control and dbinfo in local files, and these files can be directly exported to the reporting web server using import script provided by the jasper buildomatic. From the development environment these files will be generated and then pushed to deployment center.

Sample Playbook

- hosts: reportwebserver

-name: Copying exported files in $JASPER\_HOME

-name: Stopping Tomcat server.

-name: Execution of Jasper import script at $JASPER\_HOME/ buildomatic/js-import.sh

-name: Add firewall rule

-nmae: Start Tomcat server

-name: Verfication that Jasper server is running.

Pentaho Deployment

Pentaho Deployment is similar to Jasper, Pentaho specific configuration will be copied to respective location and import script will be executed.

## ReportingDBServer

Reporting DB server deployment is similar to MOTECH database approach. We need to check if reporting DB migration is required or not.

# Monitoring

This section is intended to capture the monitoring proposal for NMS system.

Nagios tool shall be used to monitor the status of the infrastructural elements of NMS system that includes: Tomcat, MySQL, ActiveMq, and Connectivity to external services etc.

Nagios shall be setup on the Health Monitoring Server and it shall monitor all the servers.

## NMS Monitored Services

|  |  |  |  |
| --- | --- | --- | --- |
| **Services** | **Description** | **Server** | **Plugin** |
| Root Partition | Disk usage on root partition | MOTECH Web, Reporting Web, MDS-DB, Report-DB, Health Monitoring | check\_disk |
| Current Users | Number of users currently logged on the system and may raise WARNING alerts if the number of user exceeds some predefined number in the system | MOTECH Web, Reporting Web, MDS-DB, Report-DB, Health Monitoring | check\_users |
| Total Process | Number of processes running on the servers and may raise WARNING alerts if the number of processes exceeds some predefined number in the system | MOTECH Web, Reporting Web, MDS-DB, Report-DB, Health Monitoring | check\_procs |
| Current Load | CPU load on the server and may raise WARNING alert when load surpasses the amount specified by the system defined parameter. | MOTECH Web, Reporting Web, MDS-DB, Report-DB, Health Monitoring | check\_load |
| Swap Usage | Swap usage on servers and may raise WARNING when swap space goes below some predefine limit | MOTECH Web, Reporting Web, MDS-DB, Report-DB, Health Monitoring | check\_swap |
| SSH | SSH service on the server | MOTECH Web, Reporting Web, MDS-DB, Report-DB, Health Monitoring | check\_ssh |
| Tomcat | Tomcat service running on Servers | MOTECH Web | check\_tomcat.pl ([http://exchange.nagios.org/ directory/Plugins/Java-Applications-and-Servers/Apache-Tomcat/check\_tomcat-2Epl/details](http://exchange.nagios.org/%20directory/Plugins/Java-Applications-and-Servers/Apache-Tomcat/check_tomcat-2Epl/details)) |
| MySQL | MySQL running on servers | MDS-DB, Reporting-DB | check\_MySQL\_health |
| ActiveMQ | ActiveMQ running servers | MDS-DB | check\_activemq |
| Reporting tool | Reporting tool connectivity | Reporting Web | plugin needs to be written for NMS system |
| IVR system | IVR system connectivity | IVR | plugin needs to be written for NMS system |
| IVR SMS URL | SMS URL provided by IVR system | IVR | plugin needs to be written for NMS system |
| JVM stats | JVM stats for the JVM running on Servers | MOTECH Web, Reporting Web | JvmInspector is standalone tool + Nagios wrapper plugin (check\_jvm) |
| KK Service | Kilkari service running on Motech platform | MOTECH Web | GF to provide URL |
| MK Service | Mobile Kunji service running on Motech platform | MOTECH Web | GF to provide URL |
| MA Service | Mobile Academy service running on Motech platform | MOTECH Web | GF to provide URL |

## Monitoring User

Monitoring users shall be created in Nagios Monitoring tool for sending email/sms in case of faults/alarms. NMS system shall maintain group of monitoring users. Any new user shall be added to this group through deployment script.

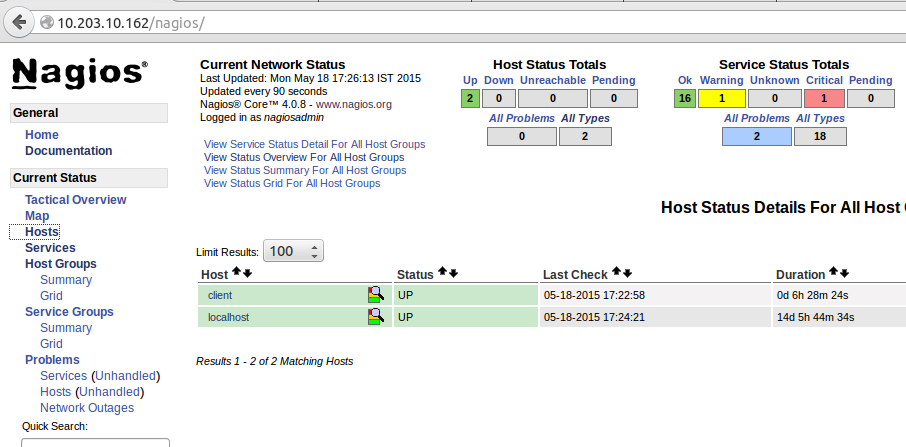
# Appendix

## Nagios POC

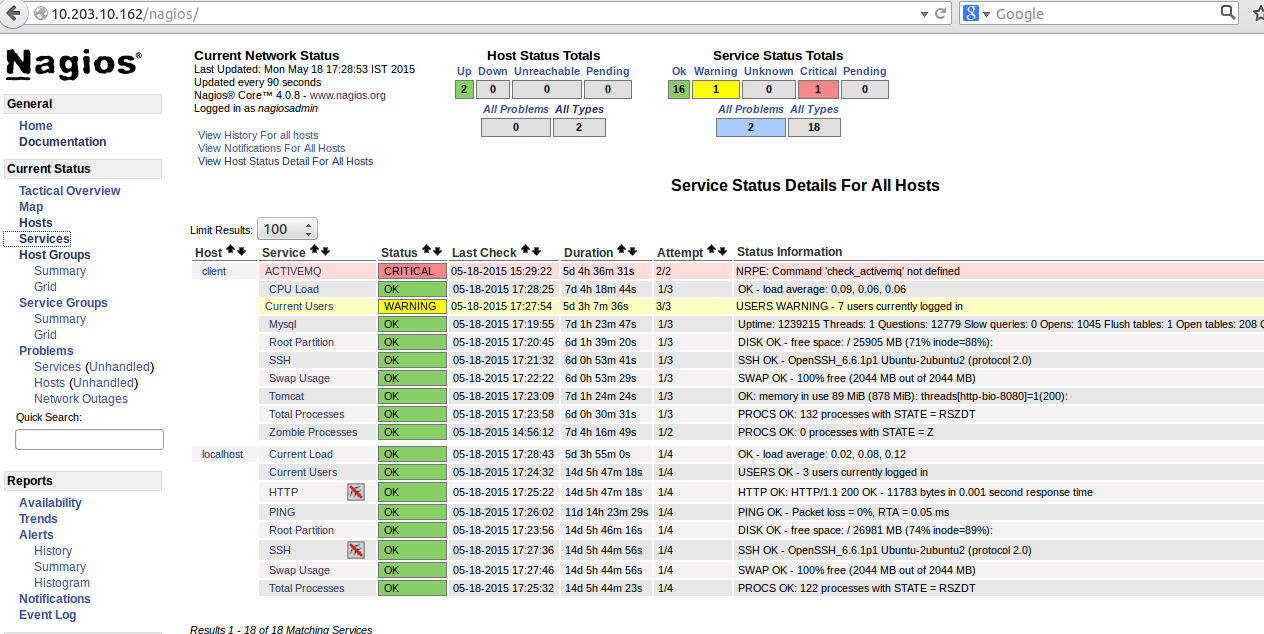
Nagios tool has been configured to monitor local and remote system as a POC.

Following snapshot have been captured to demonstrate the tool for monitoring services running on local and remote server machines.

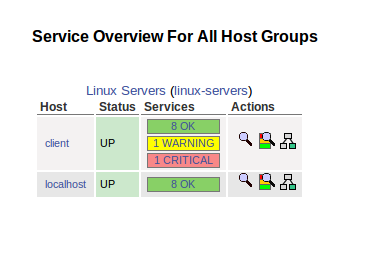
## Hosts Status Details For all Host Groups



## Service Status Details For all Hosts



## Service Overview for all Host Groups



## Tomcat

**Command**:

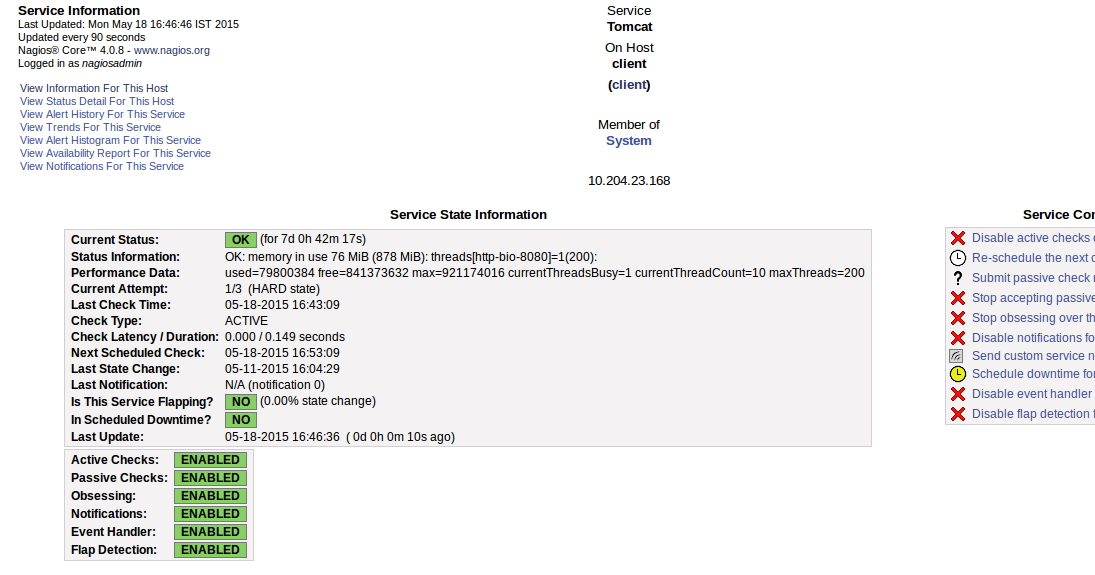
command[check\_tomcat]=/usr/lib/nagios/plugins/check\_tomcat.pl -p8080 -l\*abhishek\* -a\*abhishek\* -w20 -c20

**Purpose**:

This probe checks to see if Tomcat server is running on remote server. It issues CRITICAL alert when it is down.

This service is in System servicegroup. And the host group name is linux-servers.

**Screenshot**:



## Mysql

**Command**:

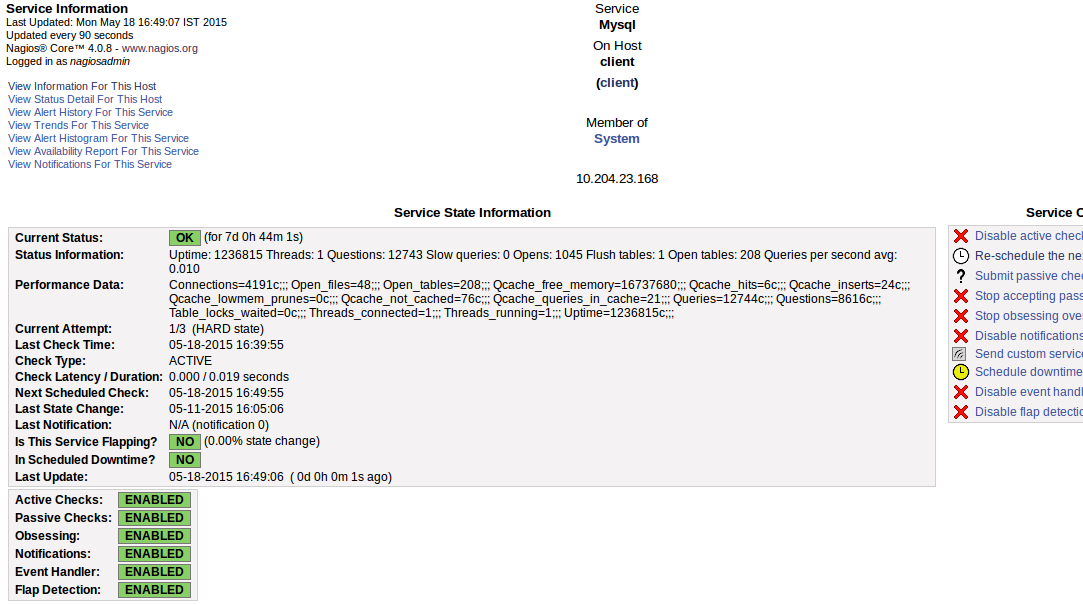
command[check\_mysql]=/usr/lib/nagios/plugins/check\_mysql -d motech\_data\_services -u root -p abhishek

**Purpose**:

This probe checks to see if mysql server is running on remote server. It issues a CRITICAL alert when it is down.

This service is in System servicegroup. And the host group name is linux-servers.

**Screenshot**:



## Root Partition

**Command**:

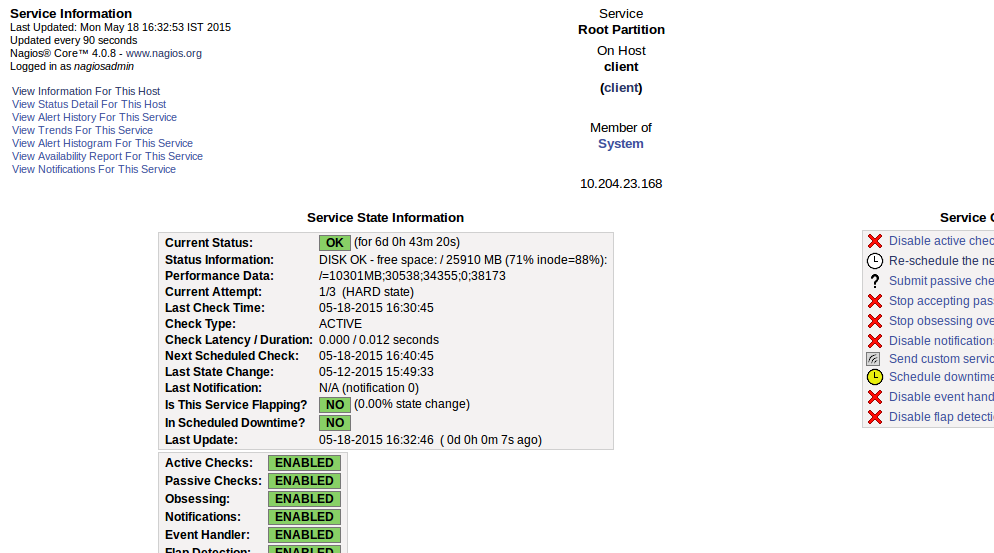
command[check\_disk]=/usr/lib/nagios/plugins/check\_disk -w 20% -c 10% -p /

**Purpose**:

This probe is to monitor the disk usage on root partition of remote server. It issues a WARNING alert if free disk space has gone below 20% and a CRITICAL alert if it has gone below 10%.

This service is in System servicegroup. And the host group name is linux-servers.

**Screenshot**:



## Current Users

**Command**:

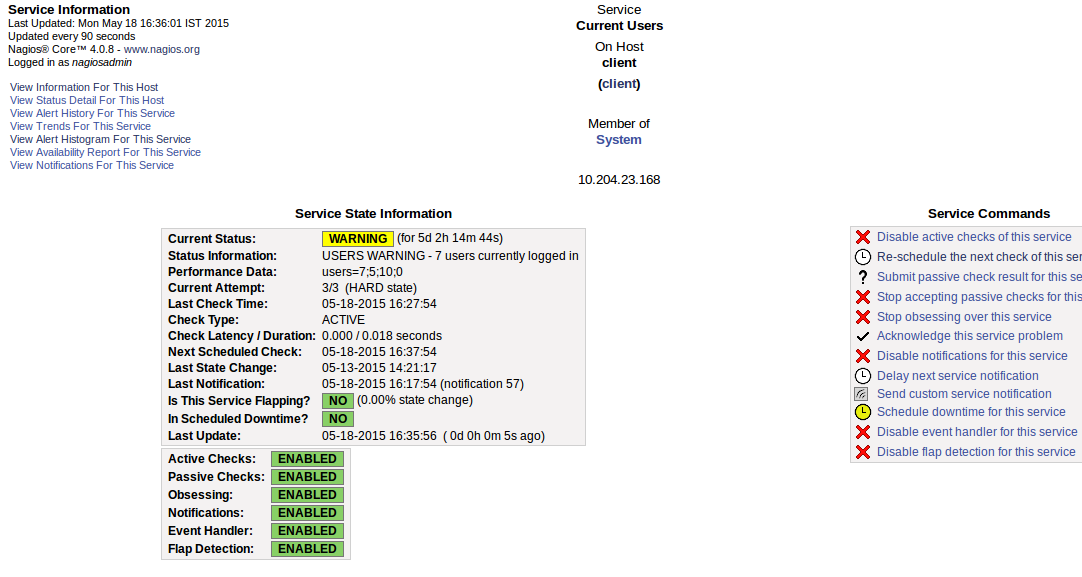
command[check\_users]=/usr/local/nagios/libexec/check\_users -w 5 -c 10

**Purpose**:

This probe monitors the number of users logged into remote server. It issues a WARNING alert when they are more than 5 in number and a CRITICAL alert when more than 10.

This service is in System servicegroup. And the host group name is linux-servers.

**Screenshot**:



## Total Processes

**Command**:

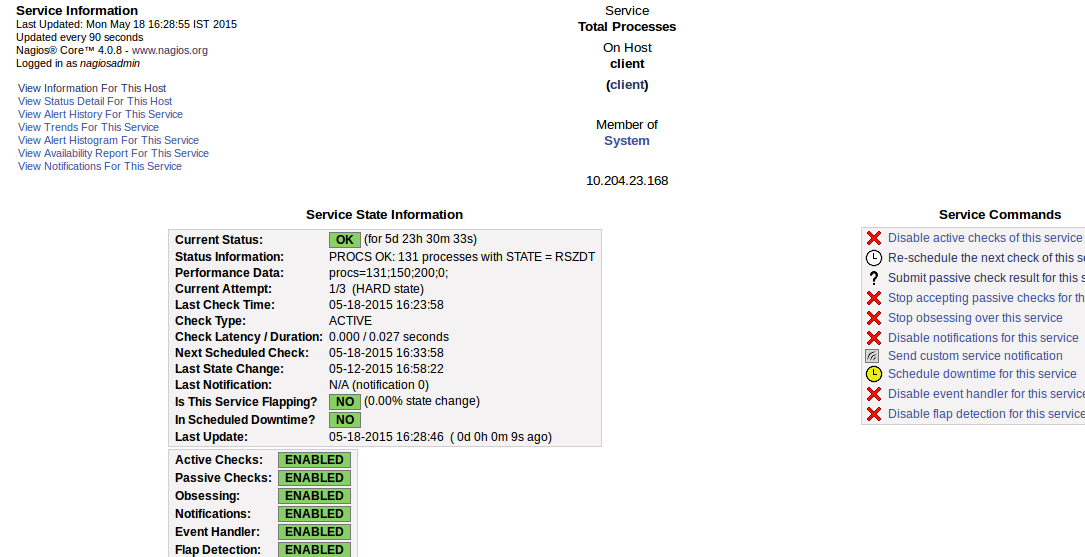
command[check\_total\_procs]=/usr/local/nagios/libexec/check\_procs -w 150 -c 200 RSZDT

**Purpose**:

This probe monitors the number of processes with status flag in RSZDT running currently on remote server. It issues a WARNING alert when they are more than 300 in number and a CRITICAL alert when more than 400.

This service is in System servicegroup. And the host group name is linux-servers.

**Screenshot**:



## Current Load

**Command**:

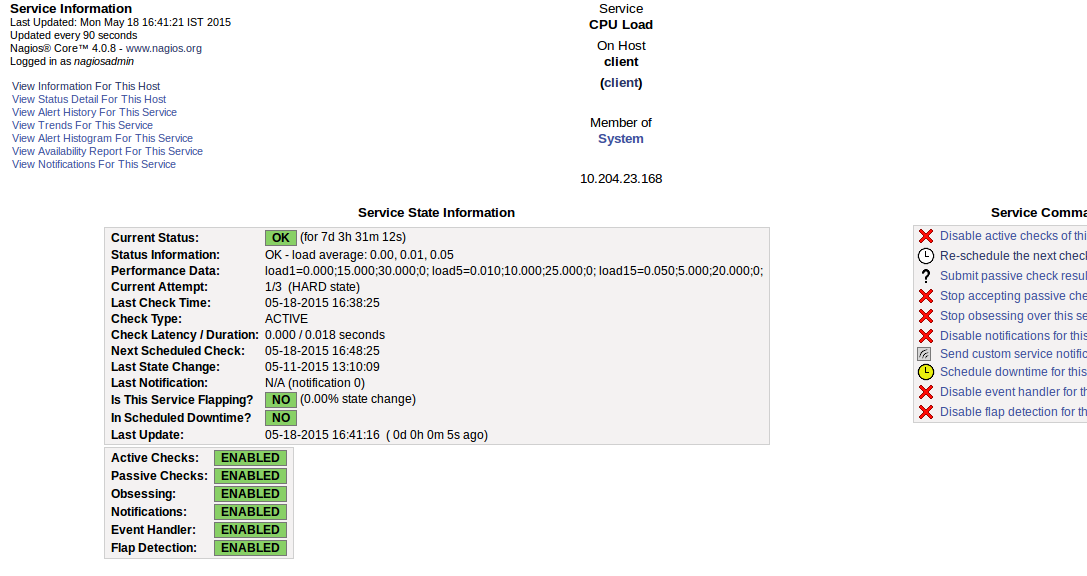
command[check\_load]=/usr/local/nagios/libexec/check\_load -w 15,10,5 -c 30,25,20

**Purpose**:

This probe monitors the CPU load avg across 5, 10 and 15 mins respectively and issues a WARNING/CRITICAL alert when the load avg surpasses the amounts specified by the –w (warning) and –c (critical) options.

This service is in System servicegroup. And the host group name is linux-servers.

**Screenshot**:



## Swap Usage

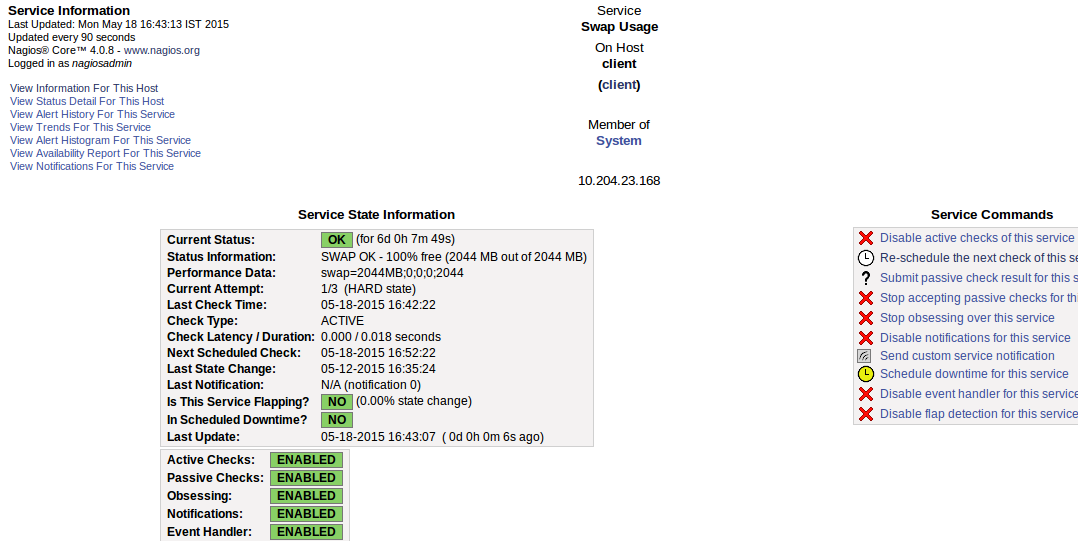
**Command**:

command[check\_swap]=/usr/local/nagios/libexec/check\_swap -w 20 -c 10

**Purpose**:

This probe monitors the swap usage on remote server. It issues WARNING when swap space goes below 20% and CRITICAL when it goes below 10%.

This service is in System servicegroup. And the host group name is linux-servers.

**Screenshot**:

## SSH

**Command**:

command[check\_SSH]=/usr/local/nagios/libexec/check\_ssh -p 22 <remote-serverip>

**Purpose**:

This probe monitors connectivity from secondary to primary via SSH. It issues CRITICAL when secondary is not able to SSH login into primary.

This service is in System servicegroup. And the host group name is linux-servers.

**Screenshot**:

