**Group Project 3**

**CIT-470-001**

**Instructor: Darci Guriel**

**Group: 1**

**Name DRR**

**Members:**

**Dan Jones, Rick Replogle, Ryan Williams**

**Date Submitted: 06/21/2021**

**TABLE OF CONTENTS**

[**Introduction……………………………….……………….……………,….……3**](#Introduction)

[**Design………………………………..……….……….….…….………,…….….4**](#Design)

[**Procedure…………………………….………………….……….……,……..….5**](#Procedure)

[**Installing and configuring Monit with a bash shell script…,.………....5**](#InstallConfigMonit)

[**Testing Monit on servers with a bash shell script…,…………..…...….6**](#TestingServer)

[**Testing Monit on clients with a bash shell script………………....…….7**](#TestingClient)

[**Achievements…….……………………………………….……………….….…8**](#Achievements)

[**Conclusion…………..………………….………………………………….….…9**](#Conclusion)

**Introduction**

Every company has a need to be able to monitor critical systems for both servers and client computers. Most of the time there is an Information Technology Department that has to monitor and maintain these critical systems. It is an important task to keep these critical systems up and running with as little downtime as possible. This is one of the most important parts of the role of a system administrator. Monitoring server and client systems as well as automatic recovery of those systems is an invaluable process to a systems administrator and for the company.

This process will demonstrate how monitoring critical systems for both server and client can be easily setup and configured within a matter of minutes. In this demonstration we used a monitoring tool called **Monit**.

**Monit is** an open source monitoring utility tool for managing and monitoring linux systems. **Monit** performs automatic maintenance and repair and can execute commands to fix situational errors.

In this presentation we are going to show your company how to configure **Monit** to monitor a variety of systems on your servers and client computers throughout your company. For this demonstration we used the following:

1 VM Server (IP address: 10.2.7.66) and

4 VM clients (IP addresses: 10.2.6.2, 10.2.6.5, 10.2.6.9, 10.2.6.10)

**Monit** uses a simple configuration file located in the ***/etc*** directory with the name ***monitrc*** (***/etc/monitrc***). This configuration file is where you would setup **Monit** to monitor Processes, Services, Files, and Directories, Filesystems as well as CPU and Memory usages.

**Design**

It is important for a critical system monitoring utility to just work all the time. You should

be able to trust that it will do just that A system monitoring utility needs to be non-

intrusive and you should be able to forget about it once it's installed. Until it alerts you of

a problem and hopefully fixed that problem on its own.

**Monit** is designed to be an autonomous system. It does not need to depend on plugins or any special

libraries for it to run. Instead it works right out of the box. It can utilize existing infrastructure already

on your system.

For this demonstration we used bash shell scripts and preconfigured monitrc

configuration files for both the server and the clients. A bash shell script was used to

configure the server with an updated monitrc configuration file that is downloaded from

a GitHub project site. The script also writes information out to a log file that can be

reviewed later. A similar script was used to configure the clients. To test the **Monit**

monitoring capabilities a server testing script and client testing script were used. These

scripts ran tests to kill processes and attempt to restart them, overload the CPU and

Memory, and test the directories and filesystems on each system.

We used bash shell scripts because we are more familiar with those and it is easier for

our group to follow and debug them. The testing scripts were able to be slimed down to

9 or 10 lines each.

**Procedure**

**Installing and configuring Monit with a bash shell script:**

1. If the client is not already booted then do it now.
2. When shown a login prompt login as root.
3. Navigate to the home directory of root using cd ~
4. Use the wget command to download the a3.tar.bz2 file.

**wget http://github.com/NKU-CIT-470/Project3/raw/main/a3.tar.bz2**

1. Type the following command to extract the files out of the archive:

**tar xvf a3.tar.bz2**

1. Switch to the a3 directory with the command ***cd a3*** (where the extracted files were placed.)
2. Run the **Monit** configuration script ***monit\_config\_script***

**./monit\_config\_script <IP Address of Client to be configured > <IP Address of Server>**

This script will configure **Monit** with the desired critical systems to be monitored on the specified client system.

1. Repeat running this **Monit** script on all the needed client systems.

**Procedure**

**Testing Monit on servers with a bash shell script:**

**Testing on a server:**

1. If you are not already logged in to the server then login as root now.
2. Navigate to the home directory of root using cd ~
3. Use the wget command to download the a3.tar.bz2 file.

**wget** [**http://github.com/NKU-CIT-470/Project3/raw/main/a3.tar.bz2**](http://github.com/NKU-CIT-470/Project3/raw/main/a3.tar.bz2)

1. Type the following command to extract the files out of the archive:

**tar xvf a3.tar.bz2**

1. Switch to the a3 directory with the command ***cd a3*** (where the extracted files were placed.)
2. Run the **Monit** configuration script ***server\_testing.sh***

**./server\_testing.sh**

1. This script will test server systems and processes.

SSH

NFS

LDAP

syslog

CPU

Memory

Directory and Filesystem

**Procedure**

**Testing Monit on clients with a bash shell script:**

**Testing on a client:**

1. If you are not already logged in to the client then login as root now.
2. Navigate to the home directory of root using cd ~
3. Use the wget command to download the a3.tar.bz2 file.

**wget** [**http://github.com/NKU-CIT-470/Project3/raw/main/a3.tar.bz2**](http://github.com/NKU-CIT-470/Project3/raw/main/a3.tar.bz2)

1. Type the following command to extract the files out of the archive:

**tar xvf a3.tar.bz2**

1. Switch to the a3 directory with the command ***cd a3*** (where the extracted files were placed.)
2. Run the **Monit** configuration script c***lient\_testing.sh***

**./client\_testing.sh**

1. This script will test client systems and processes.

SSH

NFS

LDAP

syslog

CPU

Memory

Directory and Filesystem

**Achievements**

* Team DRR met all of the requirements needed to demonstrate the effectiveness of a monitoring utility program that is easy to setup and configure in minutes. This was done with the **Monit**.
* This solution demonstrated meet your company’s needs for setting up a monitoring utility for your servers and clients. This allows system administrators to be alerted if any critical systems fail and tried to recover the failed system automatically. A tar file was downloaded and files were extracted using wget and bzip2 and then a bash shell scripts were to configure clients and test critical systems on both the clients and servers.
* The server Monit configuration file was setup to monitor the following on the server:

SSH, syslog, LDAP, NFS, Apache (httpd)

* The client Monit configuration file was setup to monitor the following on the clients:

SSH, syslog, LDAP client, NFS client.

* Monit was also configured to monitor CPU usage, Disk capacity, and Memory on each system.
* All alert messages where sent to the server for archiving. Also an email was sent to a group member Gmail account with the alert messages.
* Two testing scripts were used for testing. One for the Server and one for the Clients.
* All files used in this demonstration were uploaded to canvas.
* All members of Team DRR were involved in the setup and troubleshooting of any issues that were found along the way.

**Conclusion**

This demonstration was about how to setup **Monit** within your company. **Monit** is a small Open Source utility for managing and monitoring Linux systems. **Monit** performs automatic monitoring, maintenance and repair and can execute actions in error situations.

Using **Monit** shows how your company can benefit from the monitoring of critical server and client systems with automatic recovery and alerts of failures send to logs for archiving and alerts sent to email accounts of the system administrators.

The design was mostly straight forward and met all of your requirements. The testing scripts were able to be condensed down to 9 or 10 lines of code. The script files met the requirements specifications that were asked for.