

# Lab 7.1: Nagios Setup

## IN719 Systems Administration

### Introduction

In this lab we will configure our Nagios system to monitor our MySQL database server. You should have already created a Puppet module for your Nagios server and used it to install Nagios on your `mgmt` server.

All tasks in this lab will be performed on your `mgmt` server.

### 1 Enable logins to your Nagios web console

When you installed the `nagios3` package it included a web based monitoring console that you can view at

`http://<your-mgmt-address>/nagios3`.

The only problem is that you can't log in. We need to set up a password file for Nagios with the command

```
sudo htpasswd -c /etc/nagios3/htpasswd.users nagiosadmin
```

You will be asked to supply a password. You will use that with the user name `nagiosadmin` to log onto the web console.

Now log in. Look in particular at Hosts and Hostgroups. Nagios is configured to monitor the local machine (`mgmt`) and nothing else.

### 2 Configure Nagios using Puppet

We want to be able to monitor the db server. We will modify and add resources to our nagios server puppet module's `config` class.

1. Create a contact for yourself (and others in your team).

```
nagios_contact { 'tclark':  
    target => '/etc/nagios3/conf.d/ppt_contacts.cfg',  
    alias => 'Tom Clark',  
    service_notification_period => '24x7',  
    host_notification_period => '24x7',  
    service_notification_options => 'w,u,c,r',  
    host_notification_options => 'd,r',  
    service_notification_commands => 'notify-service-by-email',  
    host_notification_commands => 'notify-host-by-email',  
    email => 'root@localhost',  
}
```

We will change that email address in a later lab, but for now we want to use `root@localhost`.

2. Contacts aren't used until they appear in a relevant contact group.

```
nagios_contactgroup { 'sysadmins':  
    target => '/etc/nagios3/conf.d/ppt_contactgroups.cfg',  
    alias => 'Systems Administrators',  
    members => 'tclark',
```

```
}
```

3. Now we will define a nagios host for our db server. This will cause Nagios to perform checks to see that our server is up.

```
nagios_host { 'db.sqrawler.com':  
    target => '/etc/nagios3/conf.d/ppt_hosts.cfg',  
    alias => 'db',  
    address => '10.25.1.18',  
    check_period => '24x7',  
    max_check_attempts => 3,  
    check_command => 'check-host-alive',  
    notification_interval => 30,  
    notification_period => '24x7',  
    notification_options => 'd,u,r',  
    contact_groups => 'sysadmins',  
}
```

You'll have to change the IP address to match the address of your db server.

4. Now define your service. This basically tells Nagios how you want MySQL servers to be monitored.

```
nagios_service {'MySQL':  
    service_description => 'MySQL DB',  
    hostgroup_name => 'db-servers',  
    target => '/etc/nagios3/conf.d/ppt_mysql_service.cfg',  
    check_command => 'check_mysql',  
    max_check_attempts => 3,  
    retry_check_interval => 1,  
    normal_check_interval => 5,  
    check_period => '24x7',  
    notification_interval => 30,  
    notification_period => '24x7',  
    notification_options => 'w,u,c',  
    contact_groups => 'sysadmins',  
}
```

And finally, define a host group that you will use to identify your MySQL servers to Nagios.

```
nagios_hostgroup{'db-servers':  
    target => '/etc/nagios3/conf.d/ppt_hostgroups.cfg',  
    alias => 'Database Servers',  
    members => 'db.sqrawler.com',  
}
```

### 3 Other steps

There are a few housekeeping details to get things working properly.

```
sudo chown root:puppet /etc/nagios3/conf.d  
sudo chmod 775 /etc/nagios3/conf.d
```

Have puppet configure your server and verify that the right files are in /etc/nagios3/conf.d. Sadly, the permissions will be wrong. Fix them with

```
sudo chmod 644 /etc/nagios3/conf.d/*
```

If you have done everything right, your db server will appear to be up in Nagios, but the MySQL service will appear to be down. We'll sort that out later.