# Lab 05.2: Set Up Nagios With Puppet
# IN719 Systems Administration

## Introduction

Next week we will start working with Nagios to monitor our systems. We will introduce Nagios itself then, but we can already start using Puppet to install Nagios and to perform some basic configuration.

There are two things about this that can be confusing. Nagios has its own configuration files with their own syntax, but for now we are only going to use Puppet and it manifest format to configure Nagios. We can do this because we won't use Puppet to manage Nagios' configuration *files*. Instead we will prepare native Puppet resources that represent elements of Nagios configuration. Puppet has features that directly support Nagios.

Note that everything in today's lab will take place on our `mgmt` servers. There is no reason to touch any other servers for this lab.

## 1 Set up a Nagios Module

Following the example of our MariaDB module, create a new module for Nagios with an `install`, `config`, and `service` class. Most of the work in the coming weeks will be in the config class, but the other two are what we need to get set up first.

We will apply this module to **only** our `mgmt` servers.

### 1.1 `install` class

The `install` class should create a `nagios` group and a `nagios` user. It should install the `nagios3` package.

### 1.2 `service` class

The `service` class should enable the `nagios3` service and ensure that it is running. When the service is running you should be able to reach its web interface by visit `http://<your server ip>/nagios3` with a web browser. Note that this is only accessible while on the OP campus network.

### 1.3 `config` class

Nagios requires a lot of configuration and we will be working on in over the next few weeks. We will get started today by creating a `config` class that includes the following:

Create a `file` resource that manages `/etc/nagios3/nagios.cfg`. You can use the default configuration file installed by the package for this. After installing you will find a copy of this file in `/etc/nagios3/nagios.cfg`.

Create a second `file` resource to manage `/etc/nagios3/htpasswd.users`. This file contains usernames and passwords used to log into the Nagios web interface. You will create this file with the `htpasswd` command, invoked in this manner:

```
htpasswd -c <path to you module's file directory>/htpasswd.users nagiosadmin
```

You will be prompted for a password which will be used with the user name `nagiosadmin`. You must use this user name, at least for now.

Nagios uses a directory, `/etc/nagios3/conf.d`. Create a file resource that ensures the directory is present, sets its group ownership to `puppet`, and its mode to `0775`.

Finally, we will set up one nagios-specific resource, a `nagios_host`. This will direct Nagios to monitor the status of our database server. Your `nagios_host` resource should look like this:

```
nagios_host { "db-x.foo.org.nz":
                 target => "/etc/nagios3/conf.d/ppt_hosts.cfg",
                 alias => "db",
                 check_period => "24x7",
                 max_check_attempts => 3,
                 check_command => "check-host-alive",
                 notification_interval => 30,
                 notification_period => "24x7",
                 notification_options => "d,u,r",
                 mode => 0444,
}
```

We will discuss Nagios resources in class, but note especially that this resource writes a file, `/etc/nagios3/conf.d/ppt_hosts.cfg`. It places configuration directives in that file that we specify in the resource definition.

## 2 Apply the Module

Use Puppet to apply this module to our management servers **only**. If sanity is a thing you're into, don't do this by trying to write the complete module and applying it in one go. Write the module piece by piece, applying and debugging it successively until the module is complete and working. When it is done, you should be able to log into your Nagios web dashboard and find the status of your management and database servers.