

Nagios Sevices, Hosts, and Host Groups

IN719 Systems Administration

Introduction

In the last lab we created a Nagios module to manage our server. In today's we will expand our monitoring by adding resources to our module's `config` class. In particular we're going to get our Nagios server to monitor network services on our hosts. To do this, we will define *Hosts*, *Host Groups*, and *Services*. Note that we aren't directly managing Nagios configuration files, but rather defining Nagios-specific resources and letting Puppet write and manage the files.

1 Create hosts

Last week we set up a `nagios_host` for our `db` server. Start by creating similar host entries for your other three servers in the same way. Note that your Nagios instance monitors its `localhost` as part of its default configuration. You may want to remove that eventually, but for the time being it's a useful example, so leave it in place. Even though we saw how to define a host resource in the last lab, we will look at that resource again below so that we can note a few specific properties.

```
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",
}
```

This resource will set up monitoring of a host named `db-x.foo.org.nz`. Nagios will look up that host's ip address and check that it is up using the check `check-host-alive`. We'll see later how these check are defined, but in this case it just pings the target server.

In particular, note the `target` and `mode` properties. The `target` attribute tells Puppet the file in which it should place this host element. By convention I like to use the `ppt` prefix so that we can tell which files are being managed by Puppet when we look at them on our Nagios server. The `target` value can be reused for other elements and they will all be placed in the same file. So in this case we could put all our hosts in the same file. The `mode` value is important because the Nagios process needs to be able to read the file.

You should already have a host resource like the one above in your module's `config` class. Now add three more for the remaining servers.

2 Host Groups

Ultimately we want to monitor services, but those services may run on multiple hosts. Ssh is an example of this. We'd like to handle all the hosts running ssh services in one go. We do this by creating a *host group*, then set up our ssh service monitoring to look at all the hosts in the appropriate group. Create a host group for our ssh servers by adding the following to the config class:

```
nagios_hostgroup {"my-ssh-servers":
  target => "/etc/nagios3/conf.d/ppt_hostgroups.cfg",
  mode => "0444",
  alias => 'My SSH servers',
}
```

```
members => 'group00db.foo.org.nz, group00app.foo.org.nz, <et al.>',  
}
```

Note that you have a preexisting host group for ssh servers from the default configuration. You will want to remove it at some point.

Once this is done, set up a similar host group for your database server. This host group should only have your db server as a member.

3 Services

Finally we will define *services* that we want to monitor. Nagios checks those services by trying to connect to them over the network to see if they respond properly. First, define a service to check ssh. (Again, this replaces an existing one, but it's a good example.)

```
nagios_service {"ssh":
    service_description => "ssh servers",
    hostgroup_name => "my-ssh-servers",
    target => "/etc/nagios3/conf.d/ppt_services.cfg",
    check_command => "check_ssh",
    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 => "admins",
}
```

Once this is done, define a similar service for MariaDB. The relevant `check_command` is `check_mysql`, and of course the host group should be the db host group you created.

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