

## The Industry Standard in IT Infrastructure Monitoring

### Purpose

This document outlines how to setup host groups in conjunction with templates, and then how to use the API to automatically add new hosts.

### Target Audience

This document is intended for use by anyone looking to automate their setup in Nagios XI.

### Text Configuration

This is what it would look like, if we were to set this up in Core. This assumes you have the default template 'generic-service' for your services.

```
define hostgroup{
    hostgroup_name    test-hostgroup
    alias             Hostgroup for Test
}
define host{
    name              test-host
    hostgroups        test-hostgroup
    check_period      24x7
    check_interval    5
    retry_interval    1
    max_check_attempts 10
    check_command     check-host-alive
    notification_period 24x7
    notification_interval 30
    notification_options a
    notifications_enabled 1
    contact_groups    admins
    register          0
}
define service {
    use                generic-service
    hostgroup_name     test-hostgroup
    service_description PING
    check_command      check_ping!100.0,20%!500.0,60%
}
define service {
    use                generic-service
    hostgroup_name     test-hostgroup
    service_description HTTP
    check_command      check_http!-f follow
}
define service {
    use                generic-service
    hostgroup_name     test-hostgroup
    service_description TCP 3128
    check_command      check_http!3128
}
define host{
    host_name          test1
    address            1.2.3.4
    use                test-host
}
```

```

}
define host{
    host_name          test2
    address            1.2.3.4
    use                test-host
}
define host{
    host_name          test3
    address            1.2.3.4
    use                test-host
}
    
```

## Create a Host Group

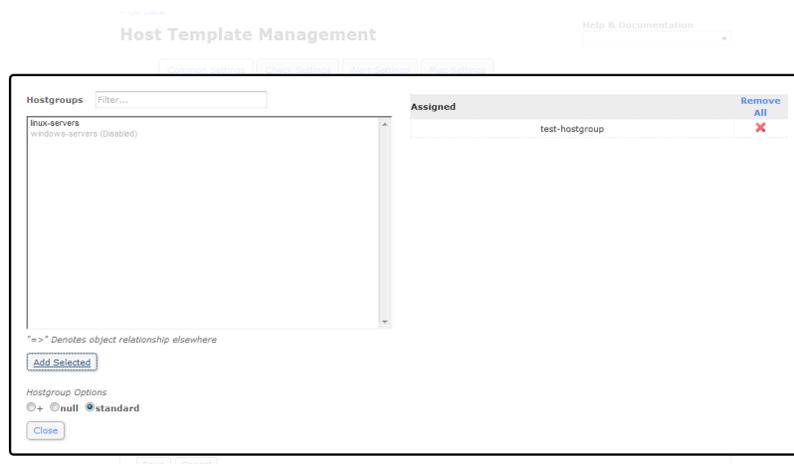
Navigate to Configure → Core Config Manager → Click 'Host Groups' on the left side. Then, click '+ Add New'.

Fill in the required information as done in the screenshot above, and click 'Save'.

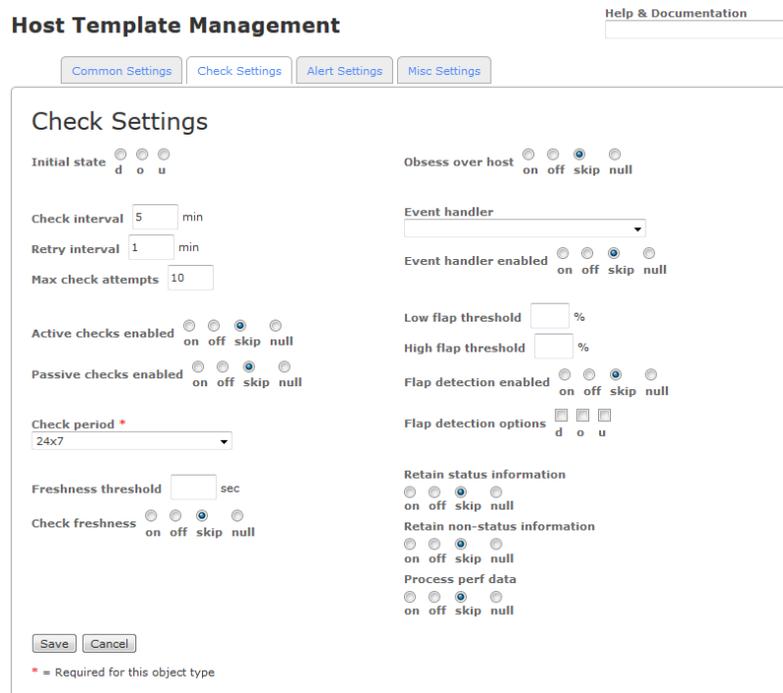
## Create a Host Template

Navigate to Configure → Core Config Manager → Click 'Host Templates' on the left side. Then, click '+ Add New'.

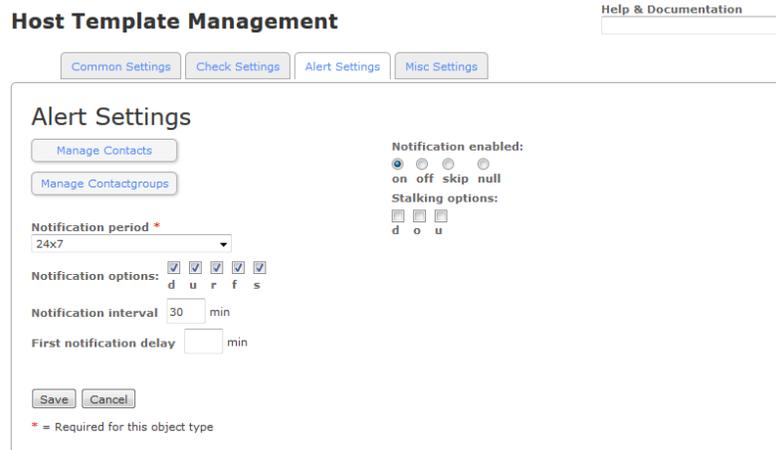
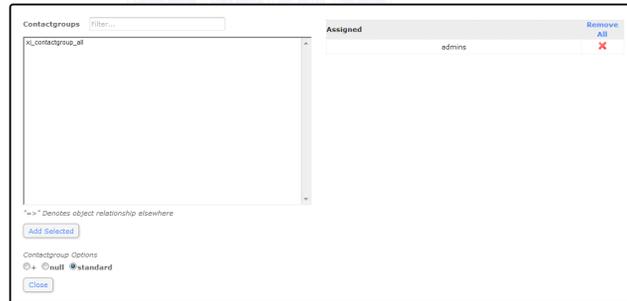
Fill out the required information on the first page, and click 'Manage Hostgroups'. Select 'test-hostgroup', and then 'Add Selected'.



Click 'Close', and then move on to the 'Check Settings' tab. Fill out the information as I've done below.



Now on the 'Alert Settings' tab, fill out the information as I've done below. Make sure to add 'admins' to the contactgroups.



Now click 'Save'.

## Create Services for this Host Group to use

Now that we've got a Host group, and host template created, we can add services that will be assigned to this host group. Usually, Nagios expects a host assigned to a service, but in this case we will be assigning a host group instead.

Navigate to Configure → Core Config Manager → Click 'Services' on the left side. Then, click '+ Add New'.

Click 'Manage Templates' → Select 'generic-service' → Click 'Add Selected'.

Then, Click 'Manage Hostgroups' → Select 'test-hostgroup' → Click 'Add Selected'

Now, assign a config name, description, and check\_command to this service. Once done, click 'Save'.

**Service Management** Help & Documentation

Common Settings | Check Settings | Alert Settings | Misc Settings

### Common Settings

Config Name \*  
test-service1

Description \*  
HTTP

Display name

Active ⓘ

\* = Required for this object type

Check command  
check\_http

Command view

```
$USER1$/check_http -I $HOSTADDRESS$ $ARG1$
```

\$ARG1\$   
\$ARG2\$   
\$ARG3\$   
\$ARG4\$   
\$ARG5\$   
\$ARG6\$   
\$ARG7\$   
\$ARG8\$

Here is an additional service that we will be monitoring as well, which I have also assigned the template, and to the 'test-hostgroup'.

**Service Management** Help & Documentation

Common Settings | Check Settings | Alert Settings | Misc Settings

### Common Settings

Config Name \*  
test-service2

Description \*  
PORT 3128

Display name

Active ⓘ

\* = Required for this object type

Check command  
check\_tcp

Command view

```
$USER1$/check_tcp -H $HOSTADDRESS$ -p $ARG1$ $ARG2$
```

\$ARG1\$ 3128  
\$ARG2\$   
\$ARG3\$   
\$ARG4\$   
\$ARG5\$   
\$ARG6\$   
\$ARG7\$   
\$ARG8\$

One more check, for ping.

### Service Management

Help & Documentation

Common Settings | Check Settings | Alert Settings | Misc Settings

#### Common Settings

Config Name \*  
test-service3

Description \*  
PING

Display name

Manage Hosts

Manage Templates

Manage Hostgroups

Manage Servicegroups

Active ⓘ

Save Cancel

\* = Required for this object type

Check command  
check\_ping

Command view

```
$USER1$/check_ping -H $HOSTADDRESS$ -w $ARG1$ -c $ARG2$ -p 5
```

\$ARG1\$ 100.0,20%

\$ARG2\$ 500.0,60%

\$ARG3\$

\$ARG4\$

\$ARG5\$

\$ARG6\$

\$ARG7\$

\$ARG8\$

Test Check Command

## Create Hosts and assign them to the Host Group

Since we've added services to the host group, now all we need to do is assign hosts to the host group. Fill out the information as done below, and make sure to assign the 'test-host' template to this host, 'test1'. It will inherit the hostgroup for 'test-hostgroup' from the 'test-host' template, so there is no need to add the member manually. Below are two examples.

### Host Management

Help & Documentation

Common Settings | Check Settings | Alert Settings | Misc Settings

#### Common Settings

Host Name \*  
test1

Description

Check command

Command view

### Host Management

Help & Documentation

Common Settings | Check Settings | Alert Settings | Misc Settings

#### Common Settings

Host Name \*  
test2

Description

Address \*  
4.3.2.1

Display name

Manage Parents

Manage Templates

Manage Hostgroups

Active ⓘ

Save Cancel

\* = Required for this object type

Check command

Command view

No command selected

\$ARG1\$

\$ARG2\$

\$ARG3\$

\$ARG4\$

\$ARG5\$

\$ARG6\$

\$ARG7\$

\$ARG8\$

## Overview of what was created

Now that you've saved everything in the Core Config Manager, it's time to click 'Apply Configuration'. This will write out your changes, and restart the nagios service. Upon doing so, click Home → Service Detail → Now, look for the host / services we just created. You will see something similar to what's below.

test1		HTTP	Critical	2m 22s	2/3	2016-05-25 13:03:21	CRITICAL - Socket timeout after 10 seconds
		PING	Critical	2m 0s	1/3	2016-05-25 13:01:49	PING CRITICAL - Packet loss = 100%
		PORT 3128	Critical	3m 8s	2/3	2016-05-25 13:02:37	CRITICAL - Socket timeout after 10 seconds
test2		HTTP	Critical	2m 13s	2/3	2016-05-25 13:03:36	CRITICAL - Socket timeout after 10 seconds
		PING	Critical	1m 52s	1/3	2016-05-25 13:01:57	PING CRITICAL - Packet loss = 100%
		PORT 3128	Critical	1m 43s	2/3	2016-05-25 13:02:14	CRITICAL - Socket timeout after 10 seconds

## Using the API

Now that we've created a hostgroup, assigned services to it, and created a host template for Nagios to use, we can now automate adding hosts by using the API. For reference, more information about the API can be found by navigating to Help → Config Reference.

We can now pass a CURL from the command line, to add in our 'test3' host by using the following command.

```
curl -XPOST "http://192.168.4.179/nagiosxi/api/v1/config/host?apikey=uii5lt5s&pretty=1" -d "host_name=test3&address=1.2.3.4&use=test-host&force=1"
```

Which will send back a response like this:

```
{
  "success": "Successfully added test3 to the system. Config imported but not yet applied."
}
```

After doing so navigate to Configure → Core Config Manager → Click 'Apply Configuration'. This will apply the host that was just added in the API. Alternatively, you can pass the `applyconfig=1` parameter to your API call, which will also do this.

The important parts to change for this CURL are going to be:

`apikey` = This is the API key assigned to your username, you can find it by clicking the user you're logged in as in the top right → look for the 'API Key' field.

`host_name` = This is the hostname we are assigning to the host, in this case it's 'test3'.

`address` = This is the IP address we are assigning to this host.

`use` = This is the template we are assigning to the host. Since the 'test-host' template is a member of the hostgroup 'test-hostgroup', this host will inherit it and become a member of the hostgroup as well.

If everything goes correctly with your API call, you will now see your third host added.

test3		HTTP	Critical	15m 53s	3/3	2016-05-25 13:30:43	CRITICAL - Socket timeout after 10 seconds
		PING	Critical	14m 14s	3/3	2016-05-25 13:32:30	PING CRITICAL - Packet loss = 100%
		PORT 3128	Critical	13m 2s	3/3	2016-05-25 13:23:45	CRITICAL - Socket timeout after 10 seconds