box293_check_vmware

A Nagios Plugin To Monitor VMware Virtualization

By Troy Lea *aka* Box293

box293_check_vmware manual   *Page 1 / 85*

Table of Contents

Overview

This is the manual for the Nagios Plugin called "box293_check_vmware". The purpose of this Plugin is to monitor your VMware vCenter / ESX(i) environment using your Nagios monitoring solution. Why did I create this Plugin when there are already other Nagios Plugins which provide similar functionality? I was not satisfied with the existing solutions available and after identifying some annoying issues I decided to create my own solution.

My primary goals were:
1. Make the solution as easy as possible to implement
2. The plugin should not impact / cripple the Nagios Host

IMPORTANT
This Plugin is **NOT** designed to be run on your Nagios host, instead it is offloaded to the VMware vSphere Management Assistant (vMA). This is due to some performance issues that occur with the VMware SDK which can easily overload your Nagios host. How all of this works will be explained in the next couple of paragraphs and full instructions are provided to get you up and running as quickly as possible.

About The Instructions
The initial part of this manual provides you with steps to follow to get this solution working as quickly as possible. The following applies:
- Windows 7 is the operating system I use to implement these steps, hence there are some programs I will instruct you to use which are:
  - WinSCP is used for transferring files
  - Putty is used for establishing SSH sessions
- The steps listed below should cater to all types of end users, the goal is to make it as easy as possible
- There are some commands which are presented as follows:
  - Type **mkdir .ssh** and press Enter
  - The **bold text** is what you need to type into the window
  - **Copying** and **Pasting** the text can save spelling mistakes and it makes the process pretty simple

Assumed Nagios Knowledge
While this manual is aimed at all types of end users, it is assumed that you know how to use and configure Nagios host and service definitions.

VMware vMA

This Plugin runs on a VMware vMA appliance, a diagram showing how this works is as follows:

- Nagios Host establishes SSH session to vMA appliance and executes the plugin
- vMA appliance communicates with the vCenter Server performing the requested check
- vMA appliance returns the plugin result to the Nagios Host and the SSH session is disconnected

Nagios Host   vMA   vCenter Server   vCenter Resources: Guests, Hosts, Datastores

The reasons for using a vMA appliance are as follows:

- **VMware SDK will easily overload your Nagios host**
  - This Plugin uses the VMware SDK for Perl API
  - This is a very powerful API however it is also very resource intensive, consuming all the CPU resources it can get it's hands on (it likes memory too)
  - If the VMware SDK is installed on your Nagios host, then the checks executed by this plugin can easily cause your Nagios host to become slow and unresponsive, in turn crippling your Nagios monitoring solution
  - Hence by offloading the checks to a vMA appliance your Nagios monitoring solution is not affected
- **vMA is a ready to go platform**
  - Simply download the latest version of the vMA appliance and deploy it into your VMware Infrastructure
  - This includes everything required for the plugin to work straight away
  - No wasting time with confusing installation procedures and pre-requisites
  - Allows you to get your VMware monitoring solution up and working straight away

IMPORTANT:
If you do not use a vMA appliance then I will not be able to provide you with support.

box293_check_vmware manual   *Page 4 / 85*

Why am I using SSH sessions instead of an agent like NRPE?

Some of the checks in this plugin return a lot a information and performance data. NRPE has limitations in relation to the amount of data that it will return. By using SSH these issues do not occur. It also makes for a relatively simple solution and in turn allows you to spend less time configuring agents.

Does this plugin require vCenter? Will it work on stand alone ESXx(i) hosts?

- No, vCenter is not required
- Yes, this plugin will talk directly to ESX(i) hosts

If you have a vCenter server, then all of your checks should be issued through your vCenter server as you only require one set of credentials to query everything in your environment. If you choose to monitor each vCenter connected ESX(i) host, you will need to manage multiple sets of credentials. Also, any checks that are specific to vCenter can only be checked when issued through your vCenter server.

If you do not have vCenter then you can monitor each one of your ESX(i) hosts (and their guests) using this plugin.

What credentials are required?

- **vCenter solution**
  - All you require is a user account that has been assigned the Read-only role. It's best to assign this permission at the top of the vCenter Tree. This plugin is fully functional with an account that has the Read-only role.
- **ESX(i) solution**
  - I have not yet tested this plugin with locally created accounts on the ESX(i) host. This will be tested in the future.
  - Currently all testing has been performed using the root account however it is assumed that a local account assigned the Read-only role will work.
- You will also need to know the root password of your Nagios host

vMA Placement

It is recommended to place the vMA into the same subnet as your vCenter Server / ESX(i) Host(s).

box293_check_vmware manual   *Page 5 / 85*

vMA Sizing

Memory and CPU allocations for the vMA will depend on how much is being monitored. It is recommended that you monitor the vMA and add memory and CPU when required.

I recommended the vMA has 2 CPUs and 2GB Memory at a minimum.

Every time a check is executed it will consume memory and CPU, hence the check period of each service definition should be sensible. For example there is no reason to check the vCenter version every five minutes, this could be a 'once a day' check.

However it is helpful to have some real world numbers here. Occasionally I get feedback from end users who divulge their vMA appliance resource allocations along with how much monitoring they are doing. If you would like your environment added to this document please send them through.

Environment #1

vMA

- Memory: **8**GB

- # CPU Sockets / Cores : **4vCPU @ 2.4GHz**

Nagios:

- Approx number of box293_check_vmware checks: ~96-100 on one vMA and ~70-75 on another vMA

- Average check_interval defined in services: check interval is 5 on all checks except maybe 5-10 where the interval is once a week

Notes:

- Our VM admin also notes that performance on the server typically sits at around 20% CPU usage and 40-50% memory. Spikes occasionally, especially during backups.

Environment #2

vMA

- Memory: **8**GB

- # CPU Sockets / Cores : **4vCPU @ 2.6GHz**

Nagios:

Check	check_interval	Quantity	Notes
Guest_Status	60	84	
Guest_Disk_Usage	60	79	
Datastore_Usage	10	30	Maximum concurrent checks is 60
Host_Status	60	6	
Other	1440	19	

box293_check_vmware manual   *Page 6 / 85*

Initial Configurations

These are the steps required to get the box293_check_vmware plugin up and running.

Download and extract files

Download the box293_check_vmware.zip file to a location on your PC and extract the files.

TCP/IP Configuration

The vMA appliance will require a Static IP Address. It is recommended that it lives in the same subnet as your vCenter server / ESX(i) hosts. The following bits of information are required in the following steps:

- IP Address
- Subnet Mask
- Default Gateway
- DNS Servers

Deploy vMA Appliance

- Download the latest version of vMA from the VMware Website
- Deploy it into your VMware Virtual Infrastructure
- **Edit** the vMA appliance **settings**
  - **Options** tab > **vApp Options**
    - Select **Disable**
      - Answer **Yes** to deleting vApp settings
    - Click **OK**
- **Open** the vMA appliance **console**
- **Power on** the vMA appliance
- Once booted the console will display a menu of options
- Configure an IP Address
  - Type **6** and press Enter
    - Configure an IPv6 address?
      - Type **n** and press Enter *(you are not configuring IPv6)*
    - Configure an IPv4 address?
      - Type **y** and press Enter
    - Use a DHCPv4 server instead of a static IPv4 address?
      - Type **n** and press Enter
    - Type the **IP address** you want the vMA to have and press Enter *(for example 192.168.1.231)*
    - Type the **Netmask** press Enter *(for example 255.255.255.0)*
    - A summary is displayed
      - If the setting are correct type **y** and press Enter
    - *Wait while the IP Address is applied*

- Configure a Default Gateway
  - Type **2** and press Enter
  - Choose the interface for the default gateway
    - Type **0** and press Enter
  - Type the **IPv4 Default Gateway** you want the vMA to have and press Enter *(for example*

box293_check_vmware manual   *Page 7 / 85*

192.168.1.1)
- ○ For the IPv6 Default Gateway **press Enter** *(you are not configuring IPv6)*
- ○ *Wait while the Default Gateway is applied*

- • Configure DNS Servers
  - ○ Type **4** and press Enter
  - ○ DNS Server 1
    - ▪ Type your **local DNS server** here and press Enter *(for example 192.168.1.51)*
  - ○ DNS Server 2
    - ▪ Optional, if you don't require a second DNS server press Enter
  - ○ *Wait while the DNS Servers are applied*

- • Confirm our IP configuration
  - ○ Type **0** and press Enter
  - ○ Review your IP configuration and check all is correct
  - ○ If all is OK type **1** and press Enter

- • Define a password for the **vi-admin** user account
  - ○ Old Password
    - ▪ Type **vmware** and press Enter
  - ○ New password needs to be a complex password *(for example: A Str0ng P@ssw0rd)*
  - ○ **Type** the new password and press Enter
  - ○ **Retype** the new password and press Enter
- • This completes all the required wizard configurations
- • Wait while vMA completes booting (it will be a blue screen when done)
- • You can now close the vMA console as the remaining steps will be performed via an SSH session

Transfer the box293_check_vmware plugin to the vMA

- • Establish a **WinSCP session** to the **vMA** *(for example 192.168.1.231)*
  - ○ login as: **vi-admin**
  - ○ Password: The one you previously provided *(for example: A Str0ng P@ssw0rd)*
  - ○ In the **left pane** of WinSCP navigate to the directory you **extracted** box293_check_vmware.zip
  - ○ In the **right pane** of WinSCP **click** the **icon** that looks like a **house** (this changes to the home directory of vi-admin on the vMA)
  - ○ In the **left** pane drag **box293_check_vmware.pl** to the **right** pane
  - ○ This copies the box293_check_vmware.pl plugin to the vMA
  - ○ **Close** the WinSCP session

Create directory for ssh certificates AND configure plugin permissions

- • Establish an **SSH session** (using Putty) to the **vMA** *(for example 192.168.1.231)*
  - ○ login as: **vi-admin**
  - ○ Password: The one you previously provided *(for example: A Str0ng P@ssw0rd)*
  - ○ You will now be logged in and presented with: **vi-admin@localhost:~>**
  - ○ Type **mkdir .ssh** and press Enter
  - ○ Type **sudo chmod 0700 .ssh/** and press Enter
    - ▪ You are prompted for the vi-admin password *(for example: A Str0ng P@ssw0rd)*

box293_check_vmware manual   *Page 8 / 85*

- ○ Type **sudo chmod +rx box293_check_vmware.pl** and press Enter
- ○ Type **exit** and press Enter

Configure Nagios server

NOTE
The following example is performed a Nagios XI server which is running on CentOS. If you are using a different OS then some of these commands may be a little different. The main requirements here are:
- • openssh-clients (required for SSH)
- • Nagios Plugins 1.5 or later (this contains the check_by_ssh plugin which is required)
- • If you know you already have these components installed jump down to **Create Certificates** section

- • Establish an **SSH session** (using Putty) to your **Nagios Server**
  - ○ login as: **root**
  - ○ Password: *You should know your root password*
  - ○ You will now be logged in and presented with: [root@localhost ~]#

- • **Installing openssh-clients**
  - ○ Type **yum -y install openssh-clients** and press Enter
  - ○ *Wait while the files are downloaded and installed*

- • **Installing nagios-plugins**
  - ○ Type **cd /tmp** and press Enter
  - ○ Type **wget --no-check-certificate https://www.nagios-plugins.org/download/nagios-plugins-1.5.tar.gz** and press Enter
  - ○ *Wait while the files are downloaded*
  - ○ Type **tar zxvf nagios-plugins-1.5.tar.gz** and press Enter
  - ○ Type **cd nagios-plugins-1.5** and press Enter
  - ○ Type **./configure** and press Enter
  - ○ *Wait while the configuration occurs*
  - ○ Type **make all** and press Enter
  - ○ *Wait while the make occurs*
  - ○ Type **make install** and press Enter
  - ○ *Wait while the install occurs*
  - ○ Leave this SSH session open, you will use it in the next step

Create Certificates

- • **Create the certificates to use with vMA**
  - ○ *You are creating a certificate that allows the Nagios server to establish an SSH session with the vMA without using credentials*
  - ○ Type **su nagios** and press Enter *(this means the following steps will be performed as the 'nagios' user, as this is what account is used when the Nagios Monitoring Engine Executes the box293_check_vmware pluin)*
    - ▪ You will now be presented with: [nagios@localhost nagios-plugins-1.5]$
    - ▪ Type **cd ~** and press Enter *(puts you in the home directory of the nagios user)*
    - ▪ Type **ssh-keygen -t dsa** and press Enter
      - • Enter file in which to save the key (/home/nagios/.ssh/id_dsa):
        - ○ You will use the default location so **press Enter**

box293_check_vmware manual   *Page 9 / 85*

- Enter passphrase (empty for no passphrase):
  - You will use an empty passphrase so **press Enter**
- Enter same passphrase again:
  - You will use an empty passphrase so **press Enter**
- A randomart image is displayed, the certificate has been created
    - Leave this SSH session open, you will use it in the next step

- **Transfer certificates to the vMA**
    - The next command uses the IP Address or DNS name of the **vMA** *(for example 192.168.1.231)*
    - Type **cat ~/.ssh/id_dsa.pub | ssh vi-admin@192.168.1.231 'umask 077; cat >>~/.ssh/authorized_keys'** and press Enter *(all in one line)*
      - An authenticity message is displayed
      - Are you sure you want to continue connecting (yes/no)?
        - Type **yes** and press Enter
      - You are prompted for the vi-admin password *(for example: A Str0ng P@ssw0rd)*
        - Type *the password* and press Enter
      - The certificate file is transferred
    - Leave this SSH session open, you will use it in the next step

Configure vMA Credentials

- **Connect to the vMA**
  - Type **ssh vi-admin@192.168.1.231** and press Enter
    - You have now established an SSH session to the vMA without requiring credentials
    - Stay logged in as you will use this session in the next steps

- **Configure vMA credentials for connecting to vCenter Server(s) / ESX(i) host(s)**
    - The following steps use the vMA credentials repository for storing the user account credentials for accessing your vCenter Server(s) / ESX(i) host(s)
    - A full explanation of why you are doing this is in the "Credentials Overview" section of this manual

  - **vCenter Server Credentials**
    - In these steps:
      - I have a vCenter server using the IP Address 192.168.1.211
      - I have configured a user account called readonly which has been granted the Read-only role and permissions assigned in the vCenter hierarchy (it has the password: A V3ry Str0ng P@ssw0rd)
    - Type **/usr/lib/vmware-vcli/apps/general/credstore_admin.pl add --server 192.168.1.211 --username readonly** and press Enter
    - You will be prompted to type the password twice
    - You will be presented with > New entry added successfully
    - You can now perform a simple test with the plugin to ensure all is working OK
    - Type **~/box293_check_vmware.pl --server 192.168.1.211 --check vCenter_Name_Version** and press Enter
    - If the credentials were correct then you will get the response along the lines of
      - *OK: VMware vCenter Server 5.1.0 build-1123961*
    - If you have **multiple** vCenter servers, **repeat** the steps above to add the credentials for each vCenter server to the vMA credentials repository

box293_check_vmware manual   *Page 10 / 85*

- ○ You are finished on the vMA for the moment
- ○ Type **exit** and press Enter
- • You are now returned to your Nagios Host, as the user nagios
- • Type **exit** and press Enter
- ■ You are now back to your SSH session on your Nagios host as the user root
  - • Leave this SSH session open, you will use it in the next step
- ■ Proceed to the **Plugin Test From Nagios Host** section

- • **ESX(i) Host credentials**
  - ○ You only need to follow these steps if you **do not** have a vCenter server that manages your ESX(i) hosts
  - ○ In these steps:
    - ■ I have an ESX(i) host using the IP Address 192.168.1.210
    - ■ I am using the root user (it has the password: A V3ry Str0ng P@ssw0rd)
  - ○ Type **/usr/lib/vmware-vcli/apps/general/credstore_admin.pl add --server 192.168.1.210 --username root**
  - ○ You will be prompted to type the password twice
  - ○ You will be presented with > New entry added successfully
  - ○ You can now perform a simple test with the plugin to ensure all is working OK
  - ○ Type **~/box293_check_vmware.pl --server 192.168.1.210 --check Host_OS_Name_Version** and press Enter
  - ○ If the credentials were correct then you will get the response along the lines of
    - ■ *OK: VMware ESXi 5.1.0 build-1065491*
  - ○ If you have **multiple** ESX(i) hosts, **repeat** the steps above to add the credentials for each ESX(i) host to the vMA credentials repository
  - ○ You are finished on the vMA for the moment
  - ○ Type **exit** and press Enter
- • You are now returned to your Nagios Host, as the user nagios
- • Type **exit** and press Enter
- ■ You are now back to your SSH session on your Nagios host as the user root
  - • Leave this SSH session open, you will use it in the next step
- ■ Proceed to the **Plugin Test From Nagios Host** section

Plugin Test From Nagios Host

- ○ This will test the plugin in exactly the same way the Nagios monitoring engine would execute a check and confirm everything is configured correctly.
- ○ Continue on using your SSH session to your Nagios host logged in as root
- ○ Type **su nagios** and press Enter *(this means the steps will be performed as the 'nagios' user)*
  - ■ You will now be presented with: [nagios@localhost]$

- ○ **To test a vCenter Server**
  - ■ Type **/usr/local/nagios/libexec/check_by_ssh -E 1 -l vi-admin -H 192.168.1.231 -C "~/box293_check_vmware.pl --server 192.168.1.211 --check vCenter_Name_Version"** and press Enter
  - ■ If the command works as expected, you should get a response like:
    - • *OK: VMware vCenter Server 5.1.0 build-1123961*
  - ■ This means you have configured the Nagios Server to execute a check on the vMA host using box293_check_vmware without being prompted for credentials

box293_check_vmware manual   *Page 11 / 85*

- ◦ **To test an ESX(i) Host**
  - ▪ Type **/usr/local/nagios/libexec/check_by_ssh -E 1 -l vi-admin -H 192.168.1.231 -C "~/box293_check_vmware.pl --server 192.168.1.210 --check Host_OS_Name_Version"** and press Enter
  - ▪ If the command works as expected, you should get a response like:
    - • *OK: VMware ESXi 5.1.0 build-1065491*
  - ▪ This means you have configured the Nagios Server to execute a check on the vMA host using box293_check_vmware without being prompted for credentials

- ◦ Testing complete
  - ▪ Type **exit** and press Enter
- ◦ You are now back to your SSH session on your Nagios host as the user root
- ◦ Type **exit** and press Enter
- • You are now **logged off** your Nagios host

That completes all the configuration steps required to get the box293_check_vmware plugin working. The next steps will be to configure Nagios to use the plugin and create your hosts and services.

box293_check_vmware manual   *Page 12 / 85*

Configuring Nagios XI To Use The Plugin

If you use Nagios XI, there is a wizard available that does all the hard work of configuring Nagios to use this plugin.

The VMware Virtualization Wizard can be downloaded from here:

- http://exchange.nagios.org/directory/Addons/Configuration/Configuration-Wizards/VMware-Virtualization-Wizard/details

Specifically you should read this document that explains exactly how to get the wizard working (installing and configuring the vMA Settings Manager component for Nagios XI):

- http://exchange.nagios.org/components/com_mtree/attachment.php?link_id=6143&cf_id=29

IF you are using the Nagios XI VMware Virtualization Wizard then the next part of this manual *(Configuring Nagios Core To Use The Plugin)* does not need to be followed and can be skipped … however reading it will still be beneficial to understanding the plugin.

Some of the newer features added to the plugin since the wizard was created are not currently available in the wizard. The wizard will be updated eventually, until then you will need to use CCM to use any feature in the plugin that is not present in the wizard.

box293_check_vmware manual   *Page 13 / 85*

Configuring Nagios Core To Use The Plugin

It is assumed you have knowledge of how Nagios works with command, host and service definitions. In particular the **check_command** line in service definitions. The format is:

- <command name>!$ARG1$!$ARG2$!$ARG3$!$ARG4$!$ARG5$!$ARG6$!$ARG7$!$ARG8$
- *Each argument is separated by an exclamation mark !*

Command Definition

The following command definition can be used for **any** check performed by the box293_check_vmware plugin. While only 8 $ARG*x*$ variables have been defined in the command, Nagios allows for up to 32.

```
define command {
        command_name        box293_check_vmware
        command_line        $USER1$/check_by_ssh -E 1 -t 90 -l vi-admin -H 192.168.1.231 -C "nice -n19
~/box293_check_vmware.pl --server $ARG1$ --check
$ARG2$ \"$ARG3$\" \"$ARG4$\" \"$ARG5$\" \"$ARG6$\" \"$ARG7$\" \"$ARG8$\""
        }
```

Broken down this means:

$USER1$	The location of your Nagios plugins, in my case this is /usr/local/nagios/libexec
check_by_ssh	The plugin used to execute the box293_check_vmware plugin via an SSH session
-E 1	This ignores the first line returned on errors
-t 90	Defines the timeout for check_by_ssh to be 90 seconds
-l vi-admin	This is a lower case L, it is telling the check_by_ssh plugin to connect to the vMA using the user account vi-admin *(which will authenticate using the certificate you setup)*
-H 192.168.1.231	This is the IP Address of the vMA which the check_by_ssh plugin will connect to
-C	The command to execute on the remote host • It is **VERY** important that the **ENTIRE** command is enclosed in **"double quotes"**

The command broken down:

nice -n19	• This makes the plugin execute at lower process schedule ○ Refer to the Advanced Topics section about nice
~/box293_check_vmware.pl	The box293_check_vmware plugin located in the vi-admin home directory
--server $ARG1$	The vCenter server or ESX(i) host the plugin is connecting to • $ARG1$ will use value defined in each service definition
--check $ARG2$	The type of check the plugin is performing • $ARG2$ will use value defined in each service
\"$ARG3$\" to \"$ARG8$\"	Depending on the check performed, service definition variables 3 to 8 will be populated with the required parameters. • The forward slash double quote (\") needs to surround each argument as this allows for arguments that have spaces or special characters to be correctly passed onto the plugin ○ Like a cluster named "HA Mission Critical"

box293_check_vmware manual   *Page 14 / 85*

Monitoring using a vCenter Server

Host Definition

- You will need to create a host definition for that vCenter server so you can assign the service definitions for each check you create
- I will not go into the steps required to do this as you should know how to
- For the following service definition examples the host is called "vCenter_Production"

NOTE
- The address you use for the host definition must be the same address you used when creating the certificates AND adding the user credentials to the vMA credentials repository:
  - If you used an IP Address, then the host definition must use that IP Address
  - If you used a DNS Name then the host definition must use that DNS name

Service Definitions

vCenter Name and Version

```
define service {
        host_name              vCenter_Production
        service_description    vCenter Version
        check_command          box293_check_vmware!$HOSTADDRESS$!vCenter_Name_Version!!!!!!
        initial_state          u
        max_check_attempts     3
        check_interval         1440
        retry_interval         7
        active_checks_enabled  1
        check_period           24x7
        register               1
        }
```

The command broken down:

$ARG1$ *(the vCenter server)*	$HOSTADDRESS$
$ARG2$ *(the check executed by the plugin)*	vCenter_Name_Version

Comments:
- $ARG1$ uses $HOSTADDRESS$, which will obtain the IP Address / DNS name of the vCenter server from the Host definition
  - As mentioned above, this must be the same address you used when creating the certificates AND adding the user credentials to the vMA credentials repository!
- The check_interval is 1440 (once a day) as it really isn't going to change very often

box293_check_vmware manual   *Page 15 / 85*

vCenter License Status

```
define service {
        host_name                       vCenter_Production
        service_description             vCenter License Status
        check_command                   box293_check_vmware!$HOSTADDRESS$!vCenter_License_Status!!!!!!
        initial_state                   u
        max_check_attempts              3
        check_interval                  1440
        retry_interval                  7
        active_checks_enabled           1
        check_period                    24x7
        register                        1
        }
```

The command broken down:

$ARG1$ *(the vCenter server)*	$HOSTADDRESS$
$ARG2$ *(the check executed by the plugin)*	vCenter_License_Status

That's pretty straight forward. Now for something a little more complicated.

Cluster Resource Information
I want to gather the cluster resource information with the output being in MHz for CPU *(normally GHz)* and
MB for Memory *(normally GB)*

```
define service {
        host_name                       vCenter_Production
        service_description             Cluster Resource Info - HA
        check_command                   box293_check_vmware!$HOSTADDRESS$!Cluster_Resource_Info!--
cluster!HA!--reporting_si!CPU_Speed:MHz,Memory_Size:MB!!
        initial_state                   u
        max_check_attempts              3
        check_interval                  60
        retry_interval                  7
        active_checks_enabled           1
        check_period                    24x7
        register                        1
        }
```

The command broken down:

$ARG1$ *(the vCenter server)*	$HOSTADDRESS$
$ARG2$ *(the check executed by the plugin)*	Cluster_Resource_Info
$ARG3$ *(the cluster argument)*	--cluster
$ARG4$ *(the cluster name)*	HA
$ARG5$ *(the reporting_si argument)*	--reporting_si
$ARG6$ *(the reporting_si values)*	CPU_Speed:MHz,Memory_Size:MB

I think you get the idea by now, read through the manual to see what checks are available.

box293_check_vmware manual   *Page 16 / 85*

Monitoring ESX(i) Host Managed by a vCenter Server

These checks are specific for an ESX(i) host managed by vCenter.

Host Definition

- You will need to create a host definition for that ESX(i) host so you can assign the service definitions for each check you want to create
- For my following service definition examples I will use the host called "ESXi001"

Service Definitions

Host OS Name and Version

```
define service {
        host_name                   ESXi001
        service_description         ESXi Version
        check_command               box293_check_vmware!192.168.1.211!Host_OS_Name_Version!--host!
192.168.1.210!!!!
        initial_state               u
        max_check_attempts          3
        check_interval              1440
        retry_interval              7
        active_checks_enabled       1
        check_period                24x7
        register                    1
        }
```

The command broken down:

$ARG1$ *(the vCenter server)*	192.168.1.211
$ARG2$ *(the check executed by the plugin)*	Host_OS_Name_Version
$ARG3$ *(the host argument)*	--host
$ARG4$ *(the ESXi host name in vCenter)*	192.168.1.210

Comments:
- $ARG4$ is the value of the host you want to check
  - NOTE: This is the NAME of the host in the vCenter inventory
  - My host has the name which happens to be the IP address of the host
  - If you use DNS names for your host you will need to use that name as the value (case sEnSaTiVe)

box293_check_vmware manual   *Page 17 / 85*

Host Storage Adapter Performance

```
define service {
        host_name                    ESXi001
        service_description          HBA Performance - vmhba1
        check_command                box293_check_vmware!192.168.1.211!
Host_Storage_Adapter_Performance!--host!192.168.1.210!--name!vmhba1!!
        initial_state                u
        max_check_attempts           3
        check_interval               5
        retry_interval               3
        active_checks_enabled        1
        check_period                 24x7
        register                     1
        }
```

The command broken down:

$ARG1$ *(the vCenter server)*	192.168.1.211
$ARG2$ *(the check executed by the plugin)*	Host_Storage_Adapter_Performance
$ARG3$ *(the host argument)*	--host
$ARG4$ *(the ESXi host name in vCenter)*	192.168.1.210
$ARG5$ *(the name argument)*	--name
$ARG6$ *(the name of the Storage Adapter)*	vmhba1

I think you get the idea by now, read through the manual to see what checks are available.

box293_check_vmware manual   *Page 18 / 85*

Monitoring Guests Managed by a vCenter Server

These are checks for virtual machines (guests) that are managed by vCenter.

Host Definition

- The guest I am going to monitor is the vCenter Server which happens to be a virtual machine in my environment
- The host definition for this guest was created earlier and it was called "vCenter_Production" (so this is the host that my service definitions will use)
- **NOTE:**
  - vCenter_ Production is the name it is called in Nagios, the actual name of the guest in my vCenter inventory is "VMware vCenter Server Appliance", so this is the name it will be referenced when using the --guest argument

Service Definitions

Guest CPU Usage

```
define service {
        host_name               vCenter_Production
        service_description     Guest CPU Usage
        check_command           box293_check_vmware!192.168.1.211!Guest_CPU_Usage!--guest!
VMware vCenter Server Appliance!!!!
        initial_state           u
        max_check_attempts      3
        check_interval          5
        retry_interval          3
        active_checks_enabled   1
        check_period            24x7
        register                1
        }
```

The command broken down:

$ARG1$ *(the vCenter server)*	192.168.1.211
$ARG2$ *(the check executed by the plugin)*	Guest_CPU_Usage
$ARG3$ *(the guest argument)*	--guest
$ARG4$ *(the guest name in vCenter)*	VMware vCenter Server Appliance

box293_check_vmware manual   *Page 19 / 85*

Guest Snapshots

```
define service {
        host_name                    vCenter_Production
        service_description          Guest Snapshots
        check_command                box293_check_vmware!192.168.1.211!Guest_Snapshot!--guest!VMware
vCenter Server Appliance!--warning!snapshot_age:5!--critical!snapshot_age:10
        initial_state                u
        max_check_attempts           3
        check_interval               240
        retry_interval               7
        active_checks_enabled        1
        check_period                 24x7
        register                     1
        }
```

The command broken down:

$ARG1$ *(the vCenter server)*	192.168.1.211
$ARG2$ *(the check executed by the plugin)*	Guest_Snapshot
$ARG3$ *(the guest argument)*	--guest
$ARG4$ *(the guest name in vCenter)*	VMware vCenter Server Appliance
$ARG5$ *(the warning argument)*	--warning
$ARG6$ *(the warning value)*	snapshot_age:5
$ARG7$ *(the critical argument)*	--critical
$ARG8$ *(the critical value)*	snapshot_age:10

I think you get the idea by now, read through the manual to see what checks are available.

box293_check_vmware manual   *Page 20 / 85*

Monitoring using an ESX(i) Host (without a vCenter Server)

NOTE
You would only monitor this way if you do not have a vCenter server that manages your ESX(i) host(s).

Host Definition

- You will need to create a host definition for that ESX(i) host so you can assign the service definitions for each check you want to create
- I will not go into the steps required to do this as you should know how to
- For the following service definition examples the host is called "ESXi002"

NOTE
- The address you use for the host definition must be the same address you used when creating the certificates AND adding the user credentials to the vMA credentials repository:
  - If you used an IP Address, then the host definition must use that IP Address
  - If you used a DNS Name then the host definition must use that DNS name

Service Definitions

Host OS Name and Version

```
define service {
        host_name                       ESXi002
        service_description             ESXi Version
        check_command                   box293_check_vmware!$HOSTADDRESS$!
Host_OS_Name_Version!!!!!!
        initial_state                   u
        max_check_attempts              3
        check_interval                  1440
        retry_interval                  7
        active_checks_enabled           1
        check_period                    24x7
        register                        1
        }
```

The command broken down:

$ARG1$ *(the ESX(i) host)*	$HOSTADDRESS$
$ARG2$ *(the check executed by the plugin)*	Host_OS_Name_Version

Comments:
- You are talking directly to an ESX(i) Host, so you do not need to use the --host argument
- $ARG1$ uses $HOSTADDRESS$, which will obtain the IP Address / DNS name of the ESX(i) host from the Host definition
  - As mentioned above, this must be the same address you used when creating the certificates AND adding the user credentials to the vMA credentials repository!
- The check_interval is 1440 (once a day) as it really isn't going to change very often

box293_check_vmware manual   *Page 21 / 85*

Host Storage Adapter Performance

```
define service {
        host_name                     ESXi002
        service_description           HBA Performance - vmhba1
        check_command                box293_check_vmware!$HOSTADDRESS$!
Host_Storage_Adapter_Performance!--name!vmhba1!!!!
        initial_state                 u
        max_check_attempts            3
        check_interval                5
        retry_interval                3
        active_checks_enabled         1
        check_period                  24x7
        register                      1
        }
```

The command broken down:

$ARG1$ *(the ESX(i) host)*	$HOSTADDRESS$
$ARG2$ *(the check executed by the plugin)*	Host_Storage_Adapter_Performance
$ARG3$ *(the name argument)*	--name
$ARG4$ *(the name of the Storage Adapter)*	vmhba1

I think you get the idea by now, read through the manual to see what checks are available.

box293_check_vmware manual   *Page 22 / 85*

Monitoring Guests Managed by an ESX(i) Host

These are checks for virtual machines (guests) that are managed by the ESX(i) Host.

Host Definition

- The guest is a File Server
- You will need to create a host definition for this guest so you can assign the service definitions for each check you want to create
- I will not go into the steps required to do this as you should know how to
- For the following service definition examples the guest is called "File_Server"
- **NOTE**
  - File_Server is the name it is called in Nagios, the actual name of the guest in my vCenter inventory is "File Server", so this is the name it will be referenced when using the --guest argument

Service Definitions

Guest CPU Usage

```
define service {
        host_name                       File_Server
        service_description             Guest CPU Usage
        check_command                   box293_check_vmware!192.168.1.41!Guest_CPU_Usage!--guest!File
Server!!!!
        initial_state                   u
        max_check_attempts              3
        check_interval                  5
        retry_interval                  3
        active_checks_enabled           1
        check_period                    24x7
        register                        1
        }
```

The command broken down:

$ARG1$ *(the ESX(i) host)*	192.168.1.41
$ARG2$ *(the check executed by the plugin)*	Guest_CPU_Usage
$ARG3$ *(the guest argument)*	--guest
$ARG4$ *(the guest name managed by ESX(i) host)*	File Server

box293_check_vmware manual   *Page 23 / 85*

<u>Guest Snapshots</u>

```
define service {
        host_name                       File_Server
        service_description             Guest Snapshots
        check_command                   box293_check_vmware!192.168.1.41!Guest_Snapshot!--guest!File
Server!--warning!snapshot_age:5!--critical!snapshot_age:10
        initial_state                   u
        max_check_attempts              3
        check_interval                  240
        retry_interval                  7
        active_checks_enabled           1
        check_period                    24x7
        register                        1
        }
```

The command broken down:

$ARG1$ *(the ESX(i) host)*	192.168.1.41
$ARG2$ *(the check executed by the plugin)*	Guest_Snapshot
$ARG3$ *(the guest argument)*	--guest
$ARG4$ *(the guest name managed by ESX(i) host)*	File Server
$ARG5$ *(the warning argument)*	--warning
$ARG6$ *(the warning value)*	snapshot_age:5
$ARG7$ *(the critical argument)*	--critical
$ARG8$ *(the critical value)*	snapshot_age:10

I think you get the idea by now, read through the manual to see what checks are available.

box293_check_vmware manual   *Page 24 / 85*

IP Addresses OR DNS Names

When you are configuring the plugin, credentials repository and service checks you will need to use either an IP Address OR a DNS Name. It is important that you use the same method throughout as this is how some of the magic happens. Let me explain the different areas where this comes into play.

Nagios Host establishing SSH session to the vMA without requiring credentials

- When check_by_ssh establishes an SSH session to the vMA, it will:
  - Connect to the vMA using the <u>address</u> you have specified in your service definition
  - It has a certificate fingerprint of the <u>address</u> of the vMA you are connecting to
    - The certificate fingerprint of an "IP Address" is different to a :DNS Name"
- If you use a DNS Name to connect to the vMA then your Nagios Host will need to be able to resolve this DNS Name

vMA connecting to vCenter Server or ESX(i) Host

- The plugin requires a value for the --server argument (which is the vCenter Server or ESX(i) Host)
- If you use a DNS Name then the vMA will need to be able to resolve this DNS Name

vMA Credentials Repository

- When you add entries to the Credentials Repository, they are tied to the --server argument
  - When you add a credential entry for the --server "192.168.1.85", it will only be used when the plugin connects to --server "192.168.1.85"
  - When you add a credential entry for the --server "vCenter.domain.Local", it will only be used when the plugin connects to --server "vCenter.domain.Local"

ESX(i) Host checks via a vCenter Server

- This only applies when you have a vCenter server that manages your ESX(i) host(s)
- When you perform specific host related checks, you need to provide a value for the --host argument
- The value you provide must match the <u>name</u> of the host as it appears in the vCenter inventory, for example:
  - Host001.domain.Local (case sensative)
  - 192.168.1.123

CaSe SeNsAtIve

- When you use DNS Names, they are case SenSaTivE

box293_check_vmware manual   *Page 25 / 85*

Credentials Overview

When executing a check, you will need to supply a username and password to access to the vCenter server or ESX(i) host. When you connect to a vCenter server, you do NOT need to supply additional credentials to perform checks against ESX(i) systems managed by that vCenter server.

There are a couple of options available for providing credentials when connecting to a vCenter server or ESX(i) host.

Using the Credentials Store

On the vMA appliance you can use the "Credentials Store" to save a username and passwords for each relevant vCenter server or ESX(i) host. This means that when the check is executed, vMA looks in the Credentials Store to see if it has saved credentials for that vCenter server or ESX(i) host.

The benefits of using the Credentials Store are:

- Makes it easier to change a password in the future, it only needs to be updated once on the vMA appliance and all checks from then on use the new password (which means you don't need to update any service definitions on your Nagios host)
- No credentials are exposed on the Nagios host for other admins to see

To ADD credentials to the credentials store:

- SSH to the vMA appliance as vi-admin
- Type **/usr/lib/vmware-vcli/apps/general/credstore_admin.pl add --server <vCenter server or ESX(i) host> --username readonly** and press Enter
- To **UPDATE** the password for that user simply run the same command and it will update the credential repository

To REMOVE credentials from the credentials store:

- SSH to the vMA appliance as vi-admin
- Type **/usr/lib/vmware-vcli/apps/general/credstore_admin.pl remove --server <vCenter server or ESX(i) host> --username readonly** and press Enter
- **NOTE:**
  - If your username has special characters like \ you will need to enclose the username in "double quotes"
    - Like: "vsphere.local\readonly"
- This will **REMOVE** the user from the credential repository

box293_check_vmware manual   *Page 26 / 85*

Plugin prompts me for the username even though I added the user to the credentials repository

This can happen when multiple entries exist for the same vCenter/ESX(i) server. The vMA appliance needs to know what username you want to connect to the server as. To see a **list** of users in the credentials repository:

- SSH to the vMA appliance as vi-admin
- Type **/usr/lib/vmware-vcli/apps/general/credstore_admin.pl list** and press Enter
- You should see something like:

Server	User Name
192.168.1.210	root
192.168.1.211	readonly
vcenter.box293.local	readonly
vcenter.box293.local	root

In this example you can see two entries for vcenter.box293.local.

To **RESOLVE** the issue simply remove the user from the credentials repository using the remove steps.

Alternatively if you need both credentials you can execute the plugin using the --username argument and it will then use the password for that account in the credentials repository.

Using the --username and --password arguments

If you do not want to use the credentials store you can provide the --username and --password arguments instead.
NOTE
- If your password has special characters like spaces or hash symbols you will need to enclose the password in "double quotes"

box293_check_vmware manual   *Page 27 / 85*

Note About Percentages

Some checks (such as datastore performance) will display the rate value followed by a percentage value.

Example:
- OK: Datastore 'ESXi 5.1' {Rate (Read:32 kBps / 21%)(Write:122 kBps / 79%)} {Number of (Reads:20)(Writes:109)} {Latency (Read:0 ms)(Write:1 ms)}|'Read Rate'=32kBps 'Write Rate'=122kBps 'Number of Reads'=20 'Number of Writes'=109 'Read Latency'=0ms 'Write Latency'=1ms
- Specifically {Rate (Read:32 kBps / **21%**)(Write:122 kBps / **79%**)}

The percentage values are relevant to READ vs WRITE.

For example:
- READ = 32/(32+122) * 100 = 21%
- WRITE = 122/(32+122) * 100 = 79%

- The point of including these numbers is to help assess read/write ratios when referring to best practice white papers
- These values will help you determine how your environment compares to the best practices

You will notice, after the pipe (|) in the performance data, there are no data sources for these percentage values. This is to avoid wasting disk space in the RRD files. Instead, using custom PNP graphs you can calculate the percentage values using the read and write datasources. Custom PNP graphs will be created a distributed with the box293_check_vmware plugin in the near future, stay tuned!

As of version 2016-03-24, functionality was introduced to allow percentages to be used for thresholds for specific checks. First you must enable the specific percentage via the --perfdata_option argument, example:
- --perfdata_option CPU_Free%:1

Then you need to define the warning or critical threshold:
- --warning cpu_free%:99 --critical cpu_free%:98

Example:
- box293_check_vmware.pl --server vcenter.box293.local --check Cluster_CPU_Usage --cluster "HA" --perfdata_option Cores:1,CPU_Free:1,CPU_Free%:1,CPU_Effective:1,CPU_Used:1,CPU_Used%:1,CPU_Total:1 --warning cpu_free%:99 --critical cpu_free%:98

- CRITICAL: 'HA' {Cores (Total: 4) (Available: 4)} {CPU (Free: 9.6 GHz) (Free: 98% (CRITICAL <= 98)) (Used: 0.2 GHz) (Used: 2%) (Effective: 9.8 GHz) (Total: 14 GHz)}|'Cores Total'=4 'Cores Available'=4 'CPU Free'=9.6GHz 'CPU Free %'=98%;99;98 'CPU Used'=0.2GHz 'CPU Used %'=2% 'CPU Effective'=9.8GHz 'CPU Total'=14GHz [Cluster_CPU_Usage]

Full details as to what checks support percentages and the required arguments are outlined in each check they can be used in.

So as to not cause problems with existing performance data files (RRD), percentages are optional in checks and to use them and have them report in performance data you will need to enable them using the --perfdata_option argument.

box293_check_vmware manual   *Page 28 / 85*

Note About "Double Quotes"

Throughout the manual there are references to using double quotes (").

When reading the Plugin Checks section the example command will **include** double quotes.

box293_check_vmware.pl --check Cluster_EVC_Status --server 192.168.1.211 --cluster **"Empty Cluster"**

However when reading the Service Definition section the examples **do not** include double quotes.

```
define service {
        host_name                    vCenter_Production
        service_description          Cluster Resource Info - HA
        check_command                box293_check_vmware!$HOSTADDRESS$!
Cluster_Resource_Info!--cluster!Empty Cluster!--reporting_si!CPU_Speed:MHz,Memory_Size:MB!!
        ….......
        }
```

The service definition DOES NOT require double quotes. The **command definition** for box293_check_vmware has the double quotes defined.

When the Nagios monitoring engine executes the plugin, it will execute the command with the double quotes in the appropriate places.

box293_check_vmware manual   *Page 29 / 85*

Arguments Config File

Some arguments can be defined in a config file to make check commands simpler. The config file is checked when the plugin runs.

It is called **.visdkrc** and is located in the vi-admin home directory **/home/vi-admin** or ~ as it is also referenced. This file is a simple plain text file.

The current arguments that can be defined in the config file are:

- --concurrent_checks
  - VI_CONCURRENT_CHECKS=
- --server
  - VI_SERVER=
- --timeout
  - VI_TIMEOUT=

Example:
- VI_CONCURRENT_CHECKS=45
- VI_SERVER=vcenter.box293.local
- VI_TIMEOUT=120

When the plugin executes, it will use the values of the arguments defined in the config file. **HOWEVER** if the argument is also supplied to the plugin (like **--concurrent_checks 20**), the argument will override the value in the config file.

The main reason for introducing the config file had to do with making Nagios command / service definition simpler. Lets say you had defined 20 different command definitions in Nagios for the box293_vmware_plugin. At some point, you may need to increase the --conncurrent_checks or --timeout value, which would involve updating 20 separate commands. By using the config file, you only need to update the config file once and all Nagios checks from this point forward will use the updated argument value. No need to update Nagios or restart Nagios.

box293_check_vmware manual   *Page 30 / 85*

Plugin Syntax

box293_check_vmware.pl --check <check to be performed> --server <vCenter Server or ESX(i) Host> <other arguments as required>

Plugin Arguments

Required Arguments

--check
- The type of check you want to perform
- A list of checks are explained in detail in the **Plugin Checks** section

Credentials
- Refer to the Credentials Overview section.

--server
- The vCenter server or ESX(i) host to connect to
- Most checks will work on either however some checks that rely on vCenter will only work against a vCenter server (such as cluster checks)
- Can be defined in the arguments config file as:
    - VI_SERVER=xxx
    - Refer to the "Arguments Config File" section in the manual for more information

Optional Arguments

--concurrent_checks
- Maximum amount of concurrent checks that can run at any one time
- Default is 15
- This option helps prevent the vMA appliance from being overloaded, as the VMware SDK will consume lots of CPU and memory resources
- Can be defined in the arguments config file as:
    - VI_CONCURRENT_CHECKS=xxx
    - Refer to the "Arguments Config File" section in the manual for more information
    - Using the config file is **highly recommended** (instead of using --concurrent_checks)
- Example:
    - --concurrent_checks 25

--cluster
- This is the Cluster you wish to perform the check against
- If the name of the Cluster has spaces, enclose the name in "double quotes"
- Example:
    - --cluster "HA DRS"

box293_check_vmware manual   *Page 31 / 85*

--critical
- Allows you to provide a critical threshold for the check
- Each critical threshold is in the format <type>:<value> such as "cpu_free:10"
- The value is relative to the default metric used by the check OR the type defined using the --reporting_si argument
- Multiple thresholds can be defined as some checks have thresholds for different metrics (like "disk rate" and "latency")
- Multiple thresholds are separated with a comma such as "disk_rate:150,disk_latency:30"
- If the --critical argument is not supplied then it will not return a critical state
- Supplying the --critical argument does not require the --warning argument, however if both arguments are supplied then both thresholds are checked and triggered accordingly
- Example:
  - --critical "disk_rate:150,disk_latency:30"

--datacenter
- This is the Datacenter you wish to perform the check against
- If the name of the Datacenter has spaces, enclose the name in "double quotes"
- Example:
  - --datacenter "Box293 Production"

--debug
- Generates a LOT of verbose information about what the plugin is doing
- Creates the file /home/vi-admin/box293_check_vmware_debug_log.txt
- If the debug file exists it will be overwritten

--drs_automation_level
- Allows you to check what a DRS Clusters' Automation level is currently set to
- The options are:
  - manual
  - partiallyAutomated
  - fullyAutomated
  - AlwaysOK
    - When this option is provided the service state will always be OK
- If the Cluster does not match the supplied parameter then a critical state is triggered
- If the --drs_automation_level argument is not supplied it will check for "fullyAutomated"
- Example:
  - --drs_automation_level AlwaysOK

--drs_dpm_level
- Allows you to check how a Clusters' Power Management (DPM) is configured
- The options are:
  - off
  - manual
  - automated
  - AlwaysOK

box293_check_vmware manual   *Page 32 / 85*

- ▪ When this option is provided the service state will always be OK
- If the Cluster does not match the supplied parameter then a critical state is triggered
- If the --drs_dpm_level argument is not supplied it will check for "off"
- Example:
  - ○ --drs_dpm_level automated

--drs_state
- Allows you to check if a Cluster has the Distributed Resource Scheduler (DRS) enabled or disabled
- The options are:
  - ○ enabled
  - ○ disabled
- If the Cluster does not match the supplied parameter then a critical state is triggered
- If the --drs_state argument is not supplied it will check for "enabled"
- Example:
  - ○ --drs_state disabled

--exclude_issue
- Prevent certain HOST or CLUSTER event states from causing a warning or critical status (like enabling SSH)
- Exclude options are:
  - ○ ClusterOvercommittedEvent
  - ○ DasClusterIsolatedEvent
  - ○ DasHostFailedEvent
  - ○ DasHostIsolatedEvent
  - ○ HeartbeatDatastoreNotSufficient
  - ○ HostNoRedundantManagementNetworkEvent
  - ○ InsufficientFailoverResourcesEvent
  - ○ LocalTSMEnabledEvent
  - ○ RemoteTSMEnabledEvent
- You can supply multiple options by separating them with a comma like:
  - ○ LocalTSMEnabledEvent,RemoteTSMEnabledEvent
- Example:
  - ○ --exclude_issue LocalTSMEnabledEvent,RemoteTSMEnabledEvent

--exclude_snapshot
- Exclude snapshots that contain specific text, useful for backup products that create/remove snapshots freuqently
- Examples are GX_BACKUP or VEEAM
- You can supply multiple options by separating them with a comma like:
  - ○ GX_BACKUP,VEEAM
- NOTE: text is CaSe sEnSaTiVe!
- Example:
  - ○ --exclude_snapshot Before,before
- WARNING:
  - ○ Please take care using this argument
  - ○ If snapshots are left active indefinitely they will continue to consume more disk space on the back-

box293_check_vmware manual   *Page 33 / 85*

end datastore, this can have disastrous affects

--evc_mode
- Should a Clusters' Enhanced vMotion Compatibility (EVC) Mode be enabled or disabled?
- The options are:
  - enabled
  - disabled
- If the Cluster does not match the supplied parameter then a critical state is triggered
- Example:
  - --evc_mode enabled

--filter
- Allows you to provide a filter for specific checks
- Each filter is in the format:
  - <type>:<value>
- For example:
  - hours:24
  - days:10
- In the *Tasks_Events* check, you can specify the time range for which you want the check to apply for. If you used **days:3** the check would only apply for tasks and events in the past 3 days.
- Example:
  - --filter hours:30

--guest
- The name of the virtual machine you are performing a check against
- If the name of the Guest has spaces, enclose the name in "double quotes"
- Example:
  - --guest "File Server"

--guest_consolidation_state
- Allows you to define what service state (OK, WARNING, CRITICAL) should be returned if the guest disks require consolidation (true, false)
- The option is in the format <consolidation_state>:<service_state> such as "true:WARNING"
- Both states can be defined by separating with a comma such as "true:WARNING,false:OK"
- Default states are:
  - true:CRITICAL
  - false:OK
- Example:
  - --guest_consolidation_state true:WARNING,false:OK

--guest_power_state
- Allows you to define what service state (OK, WARNING, CRITICAL) should be returned for different guest power states (poweredOn, poweredOff, suspended)
- Each option is in the format <power_state>:<service_state> such as "poweredOff:CRITICAL"
- Multiple options are separated with a comma such as

"poweredOn:OK,poweredOff:CRITICAL,suspended:WARNING"
- Default states are:
  - poweredOn:OK
  - poweredOff:CRITICAL
  - suspended:CRITICAL
- Example:
  - --guest_power_state poweredOff:CRITICAL

--guest_tools_version_state
- Allows you to define what service state (OK, WARNING, CRITICAL) should be returned for different guest tools version status (guestToolsBlacklisted, guestToolsCurrent, guestToolsNeedUpgrade, guestToolsNotRunning, guestToolsSupportedNew, guestToolsSupportedOld, guestToolsTooNew, guestToolsTooOld, guestToolsUnmanaged)
- Each option is in the format <tools_state>:<service_state> such as "guestToolsUnmanaged:OK"
- Multiple options are separated with a comma such as "guestToolsNeedUpgrade:CRITICAL,guestToolsSupportedOld:CRITICAL"
- Default states are:
  - guestToolsBlacklisted:CRITICAL
  - guestToolsCurrent:OK
  - guestToolsNeedUpgrade:WARNING
  - guestToolsNotRunning:CRITICAL
  - guestToolsSupportedNew:OK
  - guestToolsSupportedOld:WARNING
  - guestToolsTooNew:CRITICAL
  - guestToolsTooOld:CRITICAL
  - guestToolsUnmanaged:OK
- Example:
  - --guest_tools_version_state guestToolsNeedUpgrade:CRITICAL

--ha_state
- Allows you to check if a Cluster has High Availability (HA) enabled or disabled
- The options are:
  - enabled
  - disabled
- If the Cluster does not match the supplied parameter then a critical state is triggered
- If the --ha_state argument is not supplied it will check for "enabled"
- Example:
  - --ha_state disabled

--ha_admission_control
- Should a HA Clusters' Admission Control option be enabled or disabled?
- The options are:
  - enabled
  - disabled
  - AlwaysOK
    - When this option is provided the service state will always be OK

box293_check_vmware manual   *Page 35 / 85*

- If the Cluster does not match the supplied parameter then a critical state is triggered
- If the --ha_admission_control argument is not supplied it will check for "enabled"
- Example:
  - --ha_admission_control enabled

--ha_host_monitoring
- Should a HA Clusters' Host Monitoring option be enabled or disabled?
- The options are:
  - enabled
  - disabled
  - AlwaysOK
    - When this option is provided the service state will always be OK
- If the Cluster does not match the supplied parameter then a critical state is triggered
- If the --ha_host_monitoring argument is not supplied it will check for "enabled"
- Example:
  - --ha_host_monitoring AlwaysOK

--help
- Display the help
- To see the help type:
  - box293_check_vmware.pl --help | more

--host
- The name of the ESX(i) host you are performing the check against
- If connecting directly to an ESX(i) host without going via a vCenter server the --host argument is not required, the value for the --server argument will be used instead
- **NOTE:**
  - This is the NAME of the host as it appears in the inventory such as "ESX10.local" or "192.168.1.210"
  - Using a Hosts' IP address will NOT work unless this is the NAME of the host as it appears in the inventory!
  - Also look at the --modifier argument which can adjust the values that have been sent

--hide_key
- Do not display the license key for Host_License_Status or vCenter_License_Status checks
- Used when this information is deemed highly sensitive

--license
- Display the GNU General Public License
- To see the license type:
  - box293_check_vmware.pl --license | more

box293_check_vmware manual   *Page 36 / 85*

--match
- Allows you to provide values and options for checks that perform string matching
- Each option is in the format
  - `<type>:<value>`
- For example:
  - operation:nomatch
  - string:'machine memory'
  - state:CRITICAL
- Options are separated with a comma such as:
  - operation:nomatch,string:'machine memory',state:CRITICAL
- The different options are explained as follows:
  - **operation**
    - This allows you to check if a specific string matches an event, or perhaps you don't want a match
    - Allowed Values:
      - match
        - *If the string matches, then it's good. If it doesn't we should trigger the state.*
      - nomatch
        - *If the string doesn't match, then it's good. If it does we should trigger the state.*
  - **string**
    - This is the string you want to perform the match / nomatch against
    - Allowed Values:
      - Plain text alphanumeric
  - **state**
    - What Nagios state do you want to return if there is a match / nomatch
    - Allowed Values:
      - OK
      - WARNING
      - CRITICAL
      - UNKNOWN
  - **time**
    - For "tasks", this allows you to specify the when task was queued, started or completed
    - Allowed Values:
      - completeTime
        - Tasks will be checked based on when they were completed
      - queuedTime
        - Tasks will be checked based on when they were queued
      - startedTime
        - Tasks will be checked based on when they were started
  - **type**
    - This allows you to check events, tasks or all
    - Allowed Values:
      - event
        - Events will be checked
      - task
        - Tasks will be checked
      - all
        - Events and Tasks will be matched
- Example:

box293_check_vmware manual   *Page 37 / 85*

- ○ --match operation:nomatch,string:'machine memory',state:CRITICAL

--modifier
- • The modifier argument allows manipulation of input and output values in Guest and Host checks
- • The *target audience* for this argument is Nagios administrators who want to create advanced Nagios object definitions using Nagios macros
  - ○ An example of a common Nagios macro is $HOSTADDRESS$
- • For example:
  - ○ In Nagios, the **address** defined in the host object directive has the value:
    - ▪ **windows10preview.box293.local**
  - ○ In vCenter the guest is called:
    - ▪ **windows10preview**
  - ○ The Nagios command/service definition uses the argument and value:
    - ▪ **--guest $HOSTADDRESS$**
  - ○ The --modifier argument allows you to cut **.box293.local** from the **--guest** value you provided when executing a Guest_XXX check:
    - ▪ **windows10preview.box293.local** minus **.box293.local** equals **windows10preview**
  - ○ Hence you can dynamically target objects in the vSphere API using Nagios object definitions which are similar but different
- • More detailed examples are explained in the **Advanced Topics** chapter
- • Argument format is:
  - ○ <type>:<operation>:<option>:<value>
- • Where:
  - ○ type = modify an inbound (request) value or outbound (response) value
    - ▪ request
    - ▪ response
  - ○ operation = modify the value by adding, removing or changing case
    - ▪ add
      - • Add a value to the end of the object
      - • If the value is already there it will not add it again, preventing **windows10preview.box293.local.box293.local** from occurring
    - ▪ remove
      - • Remove a value from the end of the object
    - ▪ reverseip
      - • Perform a reverse IP Address lookup to get a DNS Address object
    - ▪ reverseip_remove
      - • Perform a reverse IP Address lookup to get a DNS Address and then remove a value from the end of the DNS Address object
    - ▪ shift
      - • Change the cAsE of the object
  - ○ option = operations can target the CaSe of the values
    - ▪ upper
      - • the value will be converted to UPPERCASE
    - ▪ lower
      - • the value will be converted to lowercase
    - ▪ insensitive
      - • Don't alter the case in add, remove or reverseip_remove operations
  - ○ value = operations that modify the value by adding or removing a string will need a string to match

box293_check_vmware manual   *Page 38 / 85*

- **add**, **remove** and **reverseip_remove** operations **require** a value such as **.box293.local**
- **reverseip** and **shift** operations don't require a value
- For example:
  - request:remove:insensitive:.box293.local
  - request:add:insensitive:.box293.local
  - request:reverseip:insensitive
  - request:reverseip _remove:upper:.box293.local
  - response:remove:lower:.box293.local
  - response:shift:upper
- Multiple modifiers are separated with a comma such as:
  - request:remove:insensitive:.box293.local**,**response:shift:upper:shift
- Multiple modifiers of the same type are allowed such as:
  - request:remove:upper:.local,request:remove:upper:.box293
- IPv6 is not currently supported for the reverseip and reverseip_remove operations *(coming soon)*
- More detailed information and examples are provided in the **Advanced Topics** chapter
- Example:
  - --modifier request:reverseip_remove:upper:.box293.local

--mtu

- For vSwitch and NIC checks you can query the MTU size like: 1500 or 9000
- This will determine the service state, no default value
- Example:
  - --mtu 9000

--name

- You may need to provide a name for checks like:
  - Datastore
  - Host_pNIC_Status
  - Host_pNIC_Usage
  - Host_Storage_Adapter_Info
  - Host_Storage_Adapter_Performance
  - Host_Switch_Status
  - Host_vNIC_Status
- For switch checks, a host can have multiple vSwitches, so you can specify which vSwitches you want checked (otherwise all vSwitches will be checked)
- Same applies for pNIC, vNIC and Host Storage Adapter checks
- You can check multiple objects by separating them with a comma such as:
  - vmnic0,vmnic1
- If the name of the object has spaces, enclose the name in "double quotes"
- Example:
  - --name "15K RPM"

box293_check_vmware manual   *Page 39 / 85*

--nic_state
- For NIC checks (including NICs in a vSwitch) you can query if a NIC is connected or disconnected
- The options are:
  - connected
  - disconnected
- If the --nic_state argument is not supplied it will check for "connected"
- Example:
  - --nic_state connected

--nic_duplex
- For NIC checks (including NICs in a vSwitch) you can query the duplex setting which can be full or half
- The options are:
  - full
  - half
- If the --nic_duplex argument is not supplied it will check for "full"
- Example:
  - --nic_duplex half

--nic_speed
- For NIC checks (including NICs in a vSwitch) you can query the NIC speed
- The options can be:
  - 10
  - 100
  - 1000
  - 10000
  - 40000
  - *Any value is allowed*
- If the --nic_speed argument is not supplied then it will not be checked however the speed will still be reported
- Example:
  - --nic_speed 10000

--perfdata_option
- Allows you to modify the performance data string where applicable
- Argument format <option>:<value>
- Example:
  - post_check:disabled
  - Latency:1
  - Disk_Rate:1
- Multiple arguments are allowed, separated with a comma
- Example:
  - --perfdata_option Latency:1,post_check:disabled
- post_check
  - By default, checks that return a performance data string have the check name appended to the end of the performance data string in square brackets (PNP4Nagios uses this for templates)

box293_check_vmware manual   *Page 40 / 85*

- ○ post_check:disabled prevents this from happening as some monitoring systems like Centreon do not like this
  - ○ Example:
    - ▪ --perfdata_option post_check:disabled
- • Other options can be used to select what performance data you want checked / reported on
  - ○ Without using --perfdata_option:
    - ▪ Command:
      - • box293_check_vmware.pl --server vcenter.box293.local --check Host_CPU_Usage --host esxi001.box293.local
    - ▪ Output:
      - • OK: Host CPU {Free: 12.2 GHz} {Used: 3 GHz} {Total: 15.2 GHz}|'CPU Free'=12.2GHz 'CPU Used'=3GHz 'CPU Total'=15.2GHz [Host_CPU_Usage]
  - ○ Using --perfdata_option:
    - ▪ Command:
      - • box293_check_vmware.pl --server vcenter.box293.local --check Host_CPU_Usage --host esxi001.box293.local --perfdata_option CPU_Free:1
    - ▪ Output:
      - • OK: Host CPU {Free: 5.5 GHz}|'CPU Free'=5.5GHz [Host_CPU_Usage]
  - ○ Refer to each check for what options are specific to which check

--query_url
- • This is the URL of the Nagios server's objectjson.cgi
- • Required for the Guest_Host check
- • Example:
  - ○ --query_url http://xitest.box293.local/nagios/cgi-bin/objectjson.cgi
- • **NOTE:**
  - ○ You will be able to browse to this URL using your web browser

--query_username
- • This is the username for accessing the Nagios server's objectjson.cgi
- • Required for the Guest_Host check
- • Example:
  - ○ --query_username readonly

--query_password
- • This is the password for accessing the Nagios server's objectjson.cgi
- • Required for the Guest_Host check
- • Example:
  - ○ --query_password Str0ngP@ssw0rd

--reporting_si
- • The International System of Unit to use for results that are returned for checks like CPU Usage, Memory Usage etc
- • Argument format is:
  - ○ <type>:<SI>
- • Example:
  - ○ CPU_Speed:GHz

box293_check_vmware manual   *Page 41 / 85*

- ○ Datastore_Rate:kBps
- Multiple arguments are allowed, separated with a comma. For example:
  - ○ Datastore_Rate:kBps,Latency:ms
- This is an optional argument as all checks will use a default unit unless specified
- In the Checks section, each check explains what default unit is used
- The unit types are:
  - ○ Bytes: B, kB, MB, GB, TB, PB, EB
  - ○ Bytes Per Second: Bps, kBps, MBps, GBps, TBps, PBps, Ebps
  - ○ Hertz: Hz, kHz, MHz, GHz, THz
  - ○ Time: us, ms, s, m, h, d
- Example:
  - ○ --reporting_si Datastore_Rate:kBps,Latency:ms

--service_status_info
- The current "Status Information" of the Nagios service which is running the Guest_Host check (required for the Guest_Host check)
- In a Nagios service object definition use the macro $SERVICEOUTPUT$
- Example:
  - ○ --service_status_info "$SERVICEOUTPUT$"

--standby_exit_state
- Applicable to the Host_Up_Down_State check
- If an ESX(i) host is in Standby mode, the default exit state is UP
- If you want it to report a DOWN state, use this argument with the value down
- Example:
  - ○ --standby_exit_state down

--swapfile_policy
- Should a Clusters' Swapfile Policy option be vmDirectory or hostLocal?
- The options are:
  - ○ vmDirectory
  - ○ hostLocal
- If the Cluster does not match the supplied parameter then a critical state is triggered
- If the --swapfile_policy argument is not supplied it will check for "vmDirectory"
- Example:
  - ○ --swapfile_policy hostLocal

box293_check_vmware manual   *Page 42 / 85*

--timeout
- Specify the duration a check is allowed to execute for
- 60 seconds by default
- Can be defined in the arguments config file as:
  - VI_TIMEOUT=xxx
  - Refer to the "Arguments Config File" section in the manual for more information
  - Using the config file is **highly recommended** (instead of using --timeout)
- Example:
  - --timeout 90

--version
- Reports the plugin version

--warning
- Allows you to provide a warning threshold for the check
- Each warning threshold is in the format:
  - <type>:<value>
- For example:
  - cpu_free:10
- The value is relative to the default metric used by the check OR the type defined using the --reporting_si argument
- Multiple thresholds can be defined as some checks have thresholds for different metrics (like "disk rate" and "latency")
- Multiple thresholds are separated with a comma, for example:
  - disk_rate:150,disk_latency:30
- If the --warning argument is not supplied then it will not return a warning state
- Supplying the --warning argument does not require the --critical argument, however if both arguments are supplied then both thresholds are checked and triggered accordingly
- Example:
  - --warning disk_rate:150,disk_latency:30

box293_check_vmware manual   *Page 43 / 85*

Plugin Checks

- When the plugin is executed a check needs to be defined using the --check argument
- Refer to the **Plugin Arguments → Required Arguments** section
- The following is a list of checks that can be performed, the arguments (required and optional) and example command syntax

Cluster_CPU_Usage
- Description:
  - Report the specified Clusters' CPU Usage
  - Returned as GHz by default
  - Effective vs Total vs Free:
    - Effective CPU resources available to run virtual machines. This is the aggregated effective resource level from all running hosts. Hosts that are in maintenance mode or are unresponsive are not counted. Resources used by the VMware Service Console are not included in the aggregate. HA clusters also reserve x amount of CPU resources to satisfy HA requirements.
    - Total is the aggregated CPU resources of all hosts
    - Free = Effective - Used
- Required Arguments:
  - --cluster
- Optional Arguments:
  - --perfdata_option post_check:disabled, Cores:1, CPU_Free:1, CPU_Free%:1, CPU_Effective:1, CPU_Used:1, CPU_Used%:1, CPU_Total:1
  - --reporting_si CPU_Speed:<Hertz>
  - --warning and --critical cpu_free:<Hertz>, cpu_free%:<value>, cpu_used:<Hertz>, cpu_used%:<value>

- Example 1:
  - box293_check_vmware.pl --check Cluster_CPU_Usage --server 192.168.1.211 --cluster HA
- Output for Example 1:
  - OK: 'HA' {Cores (Total: 6) (Available: 4)} {CPU (Free: 9.8 GHz) (Used: 0.9 GHz) (Effective: 10.7 GHz) (Total: 22.8 GHz)}|'Cores Total'=6 'Cores Available'=4 'CPU Free'=9.8GHz 'CPU Used'=0.9GHz 'CPU Effective'=10.7GHz 'CPU Total'=22.8GHz [Cluster_CPU_Usage]

- Example 2:
  - box293_check_vmware.pl --check Cluster_CPU_Usage --server 192.168.1.211 --cluster HA --reporting_si CPU_Speed:MHz --warning cpu_used:200
- Output for Example 2:
  - WARNING: 'HA' {Cores (Total: 6) (Available: 4)} {CPU (Free: 10,443 MHz) (Used: 273 MHz (WARNING >= 200)) (Effective: 10,716 MHz) (Total: 22,794 MHz)}|'Cores Total'=6 'Cores Available'=4 'CPU Free'=10443MHz 'CPU Used'=273MHz;200 'CPU Effective'=10716MHz 'CPU Total'=22794MHz [Cluster_CPU_Usage]

- Example 3:
  - box293_check_vmware.pl --server vcenter.box293.local --check Cluster_CPU_Usage --cluster "HA" --perfdata_option Cores:1,CPU_Free:1,CPU_Free%:1,CPU_Effective:1,CPU_Used:1,CPU_Used%:1,CPU_Total:1 --warning cpu_free%:99 --critical cpu_free%:98

box293_check_vmware manual   *Page 44 / 85*

- Output for Example 3:
  - CRITICAL: 'HA' {Cores (Total: 4) (Available: 4)} {CPU (Free: 9.6 GHz) (Free: 98% (CRITICAL <= 98)) (Used: 0.2 GHz) (Used: 2%) (Effective: 9.8 GHz) (Total: 14 GHz)}|'Cores Total'=4 'Cores Available'=4 'CPU Free'=9.6GHz 'CPU Free %'=98%;99;98 'CPU Used'=0.2GHz 'CPU Used %'=2% 'CPU Effective'=9.8GHz 'CPU Total'=14GHz [Cluster_CPU_Usage]

Cluster_DRS_Status
- Description:
  - Check the specified Distributed Resource Scheduler (DRS) Cluster
- Required Arguments:
  - --cluster
- Optional Arguments:
  - --drs_state
  - --drs_automation_level
  - --drs_dpm_level
  - --exclude_issue

- Example 1:
  - box293_check_vmware.pl --check Cluster_DRS_Status --server 192.168.1.211 --cluster "Empty Cluster"
- Output for Example 1:
  - OK: 'Empty Cluster' DRS: {Enabled} {Automation Level: Fully Automated} {Migration Threshold: 1} {DPM: Off}

- Example 2:
  - box293_check_vmware.pl --check Cluster_DRS_Status --server 192.168.1.211 --cluster "Empty Cluster" --drs_automation_level manual --drs_dpm_level on
- Output for Example 2:
  - CRITICAL: 'Empty Cluster' DRS: {Enabled} {Automation Level is fullyAutomated but should be manual} {Migration Threshold: 1} {DPM is off but should be on}

Cluster_EVC_Status
- Description:
  - Check the specified Clusters' Enhanced vMotion Compatibility (EVC) Mode
- Required Arguments:
  - --cluster
- Optional Arguments:
  - --evc_mode

- Example 1:
  - box293_check_vmware.pl --check Cluster_EVC_Status --server 192.168.1.211 --cluster "Empty Cluster"
- Output for Example 1:
  - OK: 'Empty Cluster' EVC: intel-westmere

- Example 2:
  - box293_check_vmware.pl --check Cluster_EVC_Status --server 192.168.1.211 --cluster HA --evc_mode enabled

box293_check_vmware manual   *Page 45 / 85*

- Output for Example 2:
  - CRITICAL: 'HA' EVC is disabled but should be enabled

Cluster_HA_Status
- Description:
  - Check the specified High Availability (HA) Cluster
- Required Arguments:
  - --cluster
- Optional Arguments:
  - --ha_state
  - --ha_host_monitoring
  - --ha_admission_control
  - --exclude_issue

- Example 1:
  - box293_check_vmware.pl --check Cluster_HA_Status --server 192.168.1.211 --cluster HA
- Output for Example 1:
  - OK: 'HA' HA: {Enabled} {Host Monitoring: Enabled} {Admission Control: Enabled, Policy: Tolerates 1 Host Failure} {VM Options: Restart Priority: Medium, Isolation Response: Leave Powered On} {VM Monitoring: Disabled}

- Example 2:
  - box293_check_vmware.pl --check Cluster_HA_Status --server 192.168.1.211 --cluster "Empty Cluster" --ha_admission_control disabled
- Output for Example 2:
  - OK: 'Empty Cluster' HA: {Enabled} {Host Monitoring: Enabled} {Admission Control: Disabled} {VM Options: Restart Priority: Low, Isolation Response: Shutdown} {VM Monitoring: VM Only}

Cluster_Memory_Usage
- Description:
  - Report the specified Clusters' Memory Usage
  - Returned as GB by default
  - Effective vs Total vs Free:
    - Effective Memory resources available to run virtual machines. This is the aggregated effective resource level from all running hosts. Hosts that are in maintenance mode or are unresponsive are not counted. Resources used by the VMware Service Console are not included in the aggregate. HA clusters also reserve x amount of Memory resources to satisfy HA requirements.
    - Total is the aggregated Memory resources of all hosts
    - Free = Effective - Used
- Required Arguments:
  - --cluster
- Optional Arguments:
  - --perfdata_option post_check:disabled, Memory_Effective:1, Memory_Free:1, Memory_Free%:1, Memory_Total:1, Memory_Used:1, Memory_Used%:1
  - --reporting_si Memory_Size:<Bytes>
  - --warning and --critical memory_free:<Bytes>, memory_free%:<value>, memory_used:<Bytes>, memory_used%:<value>

box293_check_vmware manual   *Page 46 / 85*

- Example 1:
  - box293_check_vmware.pl --check Cluster_Memory_Usage --server 192.168.1.211 --cluster HA
- Output for Example 1:
  - OK: 'HA' Memory {Free: 2.4 GB} {Used: 2 GB} {Effective: 4.4 GB} {Total: 12 GB}|'Memory Free'=2.4GB 'Memory Used'=2GB 'Memory Effective'=4.4GB 'Memory Total'=12GB [Cluster_Memory_Usage]

- Example 2:
  - box293_check_vmware.pl --check Cluster_Memory_Usage --server 192.168.1.211 --cluster HA --reporting_si Memory_Size:MB --warning memory_free:3000 --critical memory_free:2000
- Output for Example 2:
  - WARNING: 'HA' Memory {Free: 2,492 MB (WARNING <= 3,000)} {Used: 1,983 MB} {Effective: 4,475 MB} {Total: 12,286.5 MB}|'Memory Free'=2492MB;3000;2000 'Memory Used'=1983MB 'Memory Effective'=4475MB 'Memory Total'=12286.5MB [Cluster_Memory_Usage]

- Example 3:
  - box293_check_vmware.pl --server vcenter.box293.local --check Cluster_Memory_Usage --cluster "HA" --perfdata_option Memory_Effective:1,Memory_Free:1,Memory_Free%:1,Memory_Total:1,Memory_Used:1,Memory_Used%:1 --warning memory_free%:60 --critical memory_free%:30
- Output for Example 3:
  - WARNING: 'HA' Memory {Free: 2.4 GB} {Free: 55% (WARNING <= 60)} {Used: 2 GB} {Used: 45%} {Effective: 4.4 GB} {Total: 8 GB}|'Memory Free %'=55%;60;30 'Memory Used'=2GB 'Memory Used %'=45% 'Memory Effective'=4.4GB 'Memory Total'=8GB [Cluster_Memory_Usage]

Cluster_Resource_Info
- Description:
  - Reports the resources in a cluster such as Hosts, CPU and Memory
  - Always returns an OK state
  - Returned as GHz and GB by default
  - Effective vs Total vs Free:
    - Effective CPU and Memory resources available to run virtual machines. This is the aggregated effective resource level from all running hosts. Hosts that are in maintenance mode or are unresponsive are not counted. Resources used by the VMware Service Console are not included in the aggregate. HA clusters also reserve x amount of CPU and Memory resources to satisfy HA requirements.
    - Total is the aggregated CPU or Memory resources of all hosts
    - Free = Effective - Used
- Required Arguments:
  - --cluster
- Optional Arguments:
  - --perfdata_option post_check:disabled, Cores:1, CPU_Free:1, CPU_Free%:1, CPU_Effective:1, CPU_Used:1, CPU_Used%:1, CPU_Total:1, Hosts_Available:1, Hosts_Total:1, Memory_Effective:1, Memory_Free:1, Memory_Free%:1, Memory_Total:1, Memory_Used:1, Memory_Used%:1
  - --reporting_si CPU_Speed:<Hertz>, Memory_Size:<Bytes>

- Example 1:
  - box293_check_vmware.pl --check Cluster_Resource_Info --server 192.168.1.211 --cluster HA

box293_check_vmware manual   *Page 47 / 85*

- Output for Example 1:
  - OK: 'HA' [Hosts {Total: 3} {Available: 2}] [{Cores (Total: 6) (Available: 4)} {CPU (Free: 10.4 GHz) (Used: 0.3 GHz) (Effective: 10.7 GHz) (Total: 22.8 GHz)}] [Memory {Free: 2.4 GB} {Used: 2 GB} {Effective: 4.4 GB} {Total: 12 GB}]|'Hosts Total'=3 'Hosts Available'=2 'Cores Total'=6 'Cores Available'=4 'CPU Free'=10.4GHz 'CPU Used'=0.3GHz 'CPU Effective'=10.7GHz 'CPU Total'=22.8GHz 'Memory Free'=2.4GB 'Memory Used'=2GB 'Memory Effective'=4.4GB 'Memory Total'=12GB [Cluster_Resource_Info]

- Example 2:
  - box293_check_vmware.pl --check Cluster_Resource_Info --server 192.168.1.211 --cluster HA --reporting_si CPU_Speed:MHz,Memory_Size:MB
- Output for Example 2:
  - OK: 'HA' [Hosts {Total: 3} {Available: 2}] [{Cores (Total: 6) (Available: 4)} {CPU (Free: 10,541 MHz) (Used: 175 MHz) (Effective: 10,716 MHz) (Total: 22,794 MHz)}] [Memory {Free: 2,492 MB} {Used: 1,983 MB} {Effective: 4,475 MB} {Total: 12,286.5 MB}]|'Hosts Total'=3 'Hosts Available'=2 'Cores Total'=6 'Cores Available'=4 'CPU Free'=10541MHz 'CPU Used'=175MHz 'CPU Effective'=10716MHz 'CPU Total'=22794MHz 'Memory Free'=2492MB 'Memory Used'=1983MB 'Memory Effective'=4475MB 'Memory Total'=12286.5MB [Cluster_Resource_Info]

- Example 3:
  - box293_check_vmware.pl --server vcenter.box293.local --check Cluster_Resource_Info --cluster "HA" --perfdata_option Cores:1,CPU_Free:1,CPU_Free %:1,CPU_Effective:1,CPU_Used:1,CPU_Used %:1,CPU_Total:1,Hosts_Available:1,Hosts_Total:1,Memory_Effective:1,Memory_Free:1,Memory_ Free%:1,Memory_Total:1,Memory_Used:1,Memory_Used%:1
- Output for Example 3:
  - OK: 'HA' [Hosts {Total: 2} {Available: 2}] [{Cores (Total: 4) (Available: 4)} {CPU (Free: 9.6 GHz) (Free: 98%) (Used: 0.2 GHz) (Used: 2%) (Effective: 9.8 GHz) (Total: 14 GHz)}] [Memory {Free: 2.4 GB} {Free: 55%} {Used: 2 GB} {Used: 45%} {Effective: 4.4 GB} {Total: 8 GB}]|'Hosts Total'=2 'Hosts Available'=2 'Cores Total'=4 'Cores Available'=4 'CPU Free'=9.6GHz 'CPU Free %'=98% 'CPU Used'=0.2GHz 'CPU Used %'=2% 'CPU Effective'=9.8GHz 'CPU Total'=14GHz 'Memory Free %'=55% 'Memory Used'=2GB 'Memory Used %'=45% 'Memory Effective'=4.4GB 'Memory Total'=8GB [Cluster_Resource_Info]

Cluster_Swapfile_Status
- Description:
  - Checks the Clusters' Swapfile Policy
- Required Arguments:
  - --cluster
- Optional Arguments:
  - --swapfile_policy

- Example 1:
  - box293_check_vmware.pl --check Cluster_Swapfile_Status --server 192.168.1.211 --cluster HA
- Output for Example 1:
  - OK: 'HA' Swapfile Policy: Store In VM Directory

- Example 2:
  - box293_check_vmware.pl --check Cluster_Swapfile_Status --server 192.168.1.211 --cluster HA

box293_check_vmware manual   *Page 48 / 85*

--swapfile_policy hostLocal
- Output for Example 2:
  - CRITICAL: 'HA' Swapfile Policy is vmDirectory but should be hostLocal

Cluster_Time_Drift
- Description:
  - Checks all the hosts in a cluster to see if there is a NTP time drift between them
  - There must be a minimum of two hosts in the cluster for this check to work
- Required Arguments:
  - --cluster
- Optional Arguments:
  - --perfdata_option post_check:disabled
  - --reporting_si Time:<Time>
  - --warning and --critical time_drift:<Time>

- Example 1:
  - box293_check_vmware.pl --server vcenter.box293.local --check Cluster_Time_Drift --cluster DR
- Output for Example 1:
  - OK: 'DR' {Cluster Time Drift: 328 ms} {Hosts Total: 11}|'Cluster Time Drift'=328ms 'Hosts Total'=11 [Cluster_Time_Drift]

- Example 2:
  - box293_check_vmware.pl --server vcenter.box293.local --check Cluster_Time_Drift --cluster HA --warning time_drift:10 --critical time_drift:30 --reporting_si Time:s
- Output for Example 2:
  - CRITICAL: 'HA' {Cluster Time Drift: 90.12 s (CRITICAL >= 30), the two hosts with the biggest time drift are 'host601.box293.local' and 'host002.box293.local'} {Hosts Total: 3}|'Cluster Time Drift'=90.12s;10;30 'Hosts Total'=3 [Cluster_Time_Drift]

Cluster_vMotion_Info
- Description:
  - Reports the number of vMotions performed in the Cluster
  - Always returns an OK state
  - Useful for gathering performance data to observe trends over time
- Required Arguments:
  - --cluster
- Optional Arguments:
  - --perfdata_option post_check:disabled

- Example 1:
  - box293_check_vmware.pl --check Cluster_vMotion_Info --server 192.168.1.211 --cluster HA
- Output for Example 1:
  - OK: 'HA' vMotions: 283|'Number of vMotions'=283 [Cluster_vMotion_Info]

- Example 2:
  - box293_check_vmware.pl --check Cluster_vMotion_Info --server 192.168.1.211 --cluster HA --perfdata_option post_check:disabled

box293_check_vmware manual   *Page 49 / 85*

- Output for Example 2:
  - OK: 'HA' vMotions: 283|'Number of vMotions'=283

Datastore_Cluster_Status
- Description:
  - Check the Status of a Datastore Cluster and also reports it's configuration
  - Only valid in vSphere 5.0 onwards
- Required Arguments:
  - --name

- Example 1:
  - box293_check_vmware.pl --check Datastore_Cluster_Status --server 192.168.1.211 --name "15K Datastores"
- Output for Example 1:
  - WARNING: Datastore Cluster '15K Datastores' {Overall Status is yellow=WARNING} {Alarms Total: 2 (#1: NOT Acknowledged, yellow=WARNING, Age: 263 Days) (#2: NOT Acknowledged, yellow=WARNING, Age: 277 Days)} {Storage DRS is Enabled (Guest Disks: Keep Together) (I/O Load Balancing: Enabled, I/O Imbalance Threshold: 5, I/O Latency Threshold: 15 ms) (Automation Level: Manual) (Space Load Balancing Threshold: 80% Used) (Load Balancing Runs Every: 8 Hours) (Affinity Rule Behaviour During Maintenance: Ignored)}

Datastore_Cluster_Usage
- Description:
  - Check the Usage of a Datastore Cluster
  - Returned as TB by default
  - Only valid in vSphere 5.0 onwards
- Required Arguments:
  - --name
- Optional Arguments:
  - --perfdata_option post_check:disabled, Capacity:1, Children:1, Free:1, Free%:1, Used:1, Used%:1
  - --reporting_si Datastore_Cluster_Size:<Bytes>
  - --warning and --critical datastore_cluster_free:<Bytes>, datastore_cluster_free%:<value>, datastore_cluster_used:<Bytes>, datastore_cluster_used%:<value>

- Example 1:
  - box293_check_vmware.pl --check Datastore_Cluster_Usage --server 192.168.1.211 --name "7.2K Datastores"
- Output for Example 1:
  - OK: Datastore Cluster '7.2K Datastores' {Free Space: 11.2 TB} {Used Space: 18.8 TB} {Capacity: 30 TB} {Child Datastores: 15}|'Free Space'=11.2TB 'Used Space'=18.8TB 'Capacity'=30TB 'Child Datastores'=15 [Datastore_Cluster_Usage]

- Example 2:
  - box293_check_vmware.pl --check Datastore_Cluster_Usage --server 192.168.1.211 --name "7.2K Datastores" --critical datastore_cluster_free:12288 --reporting_si Datastore_Cluster_Size:GB
- Output for Example 2:
  - CRITICAL: Datastore Cluster '7.2K Datastores' {Free Space: 11,474.5 GB (CRITICAL <= 12,288)} {Used Space: 19,242.5 GB} {Capacity: 30,717 GB} {Child Datastores: 15}|'Free

box293_check_vmware manual   *Page 50 / 85*

Space'=11474.5GB;;12288 'Used Space'=19242.5GB 'Capacity'=30717GB 'Child Datastores'=15 [Datastore_Cluster_Usage]

Datastore_Performance
- Description:
  - Check the performance of a specific Datastore connected to a specific host
  - Metrics returned are:
    - Datastore Rate (Read and Write) as kBps
    - Number of (Reads and Writes) (no thresholds checked)
    - Device Latency (Read and Write) as ms
  - **NOTE:**
    - If the host is in Standby mode then the check will exit with an OK state
- Required Arguments:
  - --name
  - --host (the host that the datastore is connected to)
- Optional Arguments:
  - --modifier
  - --perfdata_option post_check:disabled, Datastore_Rate:1, Latency:1, Number_Of:1
  - --reporting_si Datastore_Rate:<Bytes Per Second>, Latency:<Time>
  - --warning and --critical datastore_rate:<Bytes Per Second>, datastore_latency:<Time>

- Example 1:
  - box293_check_vmware.pl --check Datastore_Performance --server 192.168.1.211 --host 192.168.1.210 --name "ESXi 5.1"
- Output for Example 1:
  - OK: Datastore 'ESXi 5.1' {Rate (Read:32 kBps / 21%)(Write:122 kBps / 79%)} {Number of (Reads:20) (Writes:109)} {Latency (Read:0 ms)(Write:1 ms)}|'Read Rate'=32kBps 'Write Rate'=122kBps 'Number of Reads'=20 'Number of Writes'=109 'Read Latency'=0ms 'Write Latency'=1ms [Datastore_Performance]

- Example 2:
  - box293_check_vmware.pl --check Datastore_Performance --server 192.168.1.211 --host 192.168.1.210 --name "ESXi 5.1" --reporting_si Datastore_Rate:Bps --warning datastore_rate:5000,datastore_latency:2 --critical datastore_rate:10000,datastore_latency:50
- Output for Example 2:
  - CRITICAL: Datastore 'ESXi 5.1' {Rate (Read:6,144 Bps / 7% (WARNING >= 5,000))(Write:95,232 Bps / 93% (CRITICAL >= 10,000))} {Number of (Reads:4) (Writes:175)} {Latency (Read:4 ms (WARNING >= 2))(Write:1 ms)}|'Read Rate'=6144Bps;5000;10000 'Write Rate'=95232Bps;5000;10000 'Number of Reads'=4 'Number of Writes'=175 'Read Latency'=4ms;2;50 'Write Latency'=1ms;2;50 [Datastore_Performance]

box293_check_vmware manual   *Page 51 / 85*

Datastore_Performance_Overall
- Description:
  - Check the Overall performance of a specific Datastore (for ALL connected hosts)
  - Metrics returned are:
    - Total Connected Hosts
    - Datastore Rate (Read and Write) as MBps
    - Number of (Reads and Writes) (no thresholds checked)
    - Device Latency (Read and Write) as ms
  - **NOTE:**
    - If ALL hosts are in Standby mode then the check will exit with an OK state
- Required Arguments:
  - --name
- Optional Arguments:
  - --perfdata_option post_check:disabled, Datastore_Rate:1, Hosts:1, Latency:1, Number_Of:1
  - --reporting_si Datastore_Rate:<Bytes Per Second>, Latency:<Time>
  - --warning and --critical datastore_rate:<Bytes Per Second>, datastore_latency:<Time>

- Example 1:
  - box293_check_vmware.pl --check Datastore_Performance_Overall --server 192.168.1.211 --name "ESXi 5.1"
- Output for Example 1:
  - OK: Datastore 'ESXi 5.1' {Total Connected Hosts: 1} {Rate (Read:38 MBps / 49%)(Write:40 MBps / 51%)} {Number of (Reads:24) (Writes:18)} {Latency (Read:0 ms)(Write:0 ms)}|'Total Connected Hosts'=1 'Read Rate'=38MBps 'Write Rate'=40MBps 'Number of Reads'=24 'Number of Writes'=18 'Read Latency'=0ms 'Write Latency'=0ms [Datastore_Performance_Overall]

- Example 2:
  - box293_check_vmware.pl --check Datastore_Performance_Overall --server 192.168.1.211 --name "ESXi 5.1" --reporting_si Datastore_Rate:Bps --warning datastore_rate:5000,datastore_latency:2 --critical datastore_rate:10000,datastore_latency:50
- Output for Example 2:
  - CRITICAL: Datastore 'ESXi 5.1' {Total Connected Hosts: 1} {Rate (Read:19,456 Bps / 49% (CRITICAL >= 10,000))(Write:20,480 Bps / 51% (CRITICAL >= 10,000))} {Number of (Reads:12) (Writes:9)} {Latency (Read:0 ms)(Write:0 ms)}|'Total Connected Hosts'=1 'Read Rate'=19456Bps;5000;10000 'Write Rate'=20480Bps;5000;10000 'Number of Reads'=12 'Number of Writes'=9 'Read Latency'=0ms;2;50 'Write Latency'=0ms;2;50 [Datastore_Performance_Overall]

Datastore_Usage
- Description:
  - Check the Usage of a specific Datastore
  - Returned as GB by default
- Required Arguments:
  - --name
- Optional Arguments:
  - --perfdata_option post_check:disabled, Datastore_Capacity:1, Datastore_Free:1, Datastore_Free%:1, Datastore_Used:1, Datastore_Used%:1
  - --reporting_si Datastore_Size:<Bytes>
  - --warning and --critical datastore_free:<Bytes>, datastore_free%:<value>, datastore_used:<Bytes>, datastore_used%:<value>

box293_check_vmware manual   *Page 52 / 85*

- Example 1:
  - box293_check_vmware.pl --check Datastore_Usage --server 192.168.1.211 --name "ESXi 5.1"
- Output for Example 1:
  - OK: Datastore 'ESXi 5.1' {Free Space: 67.3 GB} {Used Space: 67.7 GB} {Capacity: 135 GB}|'Free Space'=67.3GB 'Used Space'=67.7GB 'Capacity'=135GB [Datastore_Usage]

- Example 2:
  - box293_check_vmware.pl --check Datastore_Usage --server 192.168.1.211 --name "ESXi 5.1" --critical datastore_free:70
- Output for Example 2:
  - CRITICAL: Datastore 'ESXi 5.1' {Free Space: 67.3 GB (CRITICAL <= 70)} {Used Space: 67.7 GB} {Capacity: 135 GB}|'Free Space'=67.3GB;;70 'Used Space'=67.7GB 'Capacity'=135GB [Datastore_Usage]

- Example 3:
  - box293_check_vmware.pl --server vcenter.box293.local --check Datastore_Usage --name "VOL_02_03" --perfdata_option Datastore_Capacity:1,Datastore_Free:1,Datastore_Free %:1,Datastore_Used:1,Datastore_Used%:1 --warning datastore_free%:20
- Output for Example 3:
  - OK: Datastore 'VOL_02_03' {Free Space: 2,482.6 GB} {Free Space: 89%} {Used Space: 311.4 GB} {Used Space: 11%} {Capacity: 2,794 GB}|'Free Space'=2482.6GB 'Free Space %'=89%;20 'Used Space'=311.4GB 'Used Space %'=11% 'Capacity'=2794GB [Datastore_Usage]

Guest_CPU_Info
- Description:
  - Report Information about the Guests' CPU such as Cores, Total CPU, Reservation and Limit
  - Returned as MHz by default
  - Performance data can be useful for identifying when changes occurred, like adding CPU's or when a reservation or limit was defined
  - **NOTE:**
    - cpu_reservation and cpu_limit thresholds are either --warning OR --critical, NOT both
    - Number of cores only reported in vSphere 5.0 onwards
    - CPU Reservation only reported on directly connected ESXi hosts v 5.0 onwards ... via vCenter works for 4.0 onwards
- Required Arguments:
  - --guest
- Optional Arguments:
  - --modifier
  - --perfdata_option post_check:disabled, Cores:1 ,CPU_Limit:1, CPU_Reservation:1, CPU_Total:1
  - --reporting_si CPU_Speed:<Hertz>
  - --warning and --critical cpu_reservation:<Hertz>, cpu_limit:<Hertz>

- Example 1:
  - box293_check_vmware.pl --check Guest_CPU_Info --server 192.168.1.211 --guest "vSphere Management Assistant (vMA)"
- Output for Example 1:
  - OK: {CPU (Cores Total: 4) (Cores Per Socket: 2) (Sockets: 2)} {Total MHz: 15,196 MHz} {Reservation MHz: 0 MHz} {Limit MHz: 12,236 MHz}|'Cores Total'=4 'Cores Per Socket'=2

box293_check_vmware manual　　　　　　　　　　　　　　　　　　　　　　　　　　*Page 53 / 85*

'Sockets'=2 'CPU Total'=15196MHz 'CPU Reservation'=0MHz 'CPU Limit'=12236MHz [Guest_CPU_Info]

- Example 2:
  - box293_check_vmware.pl --check Guest_CPU_Info --server 192.168.1.211 --guest "vSphere Management Assistant (vMA)" --warning cpu_reservation:512 --critical cpu_limit:10240
- Output for Example 2:
  - CRITICAL: {CPU (Cores Total: 4) (Cores Per Socket: 2) (Sockets: 2)} {Total MHz: 15,196 MHz} {Reservation MHz: 0 MHz (WARNING != 512)} {Limit MHz: 12,236 MHz (CRITICAL != 10,240)}|'Cores Total'=4 'Cores Per Socket'=2 'Sockets'=2 'CPU Total'=15196MHz 'CPU Reservation'=0MHz;512 'CPU Limit'=12236MHz;;10240 [Guest_CPU_Info]

Guest_CPU_Usage
- Description:
  - Report the specified Guests' CPU Usage
  - Returned as MHz and ms by default
  - If the guest only has more than one core, performance data for each core is also gathered HOWEVER --warning and --critical thresholds will ONLY trigger on the total values (cpu_free, cpu_free%, cpu_used, cpu_used%, cpu_ready_time)
  - The value for total usage will not equal the individual cores added together. This has to do with the data pulled from the vSphere API, there will only be a small difference in the numbers.
- Required Arguments:
  - --guest
- Optional Arguments:
  - --modifier
  - --perfdata_option post_check:disabled, CPU_Free:1, CPU_Free%:1, CPU_Used:1, CPU_Used%:1, CPU_Available:1, CPU_Ready_Time:1
  - --reporting_si CPU_Speed:<Hertz> Time:<Time>
  - --warning and --critical cpu_free:<Hertz>, cpu_free%:<value>, cpu_used:<Hertz>, cpu_used%:<value>, cpu_ready_time:<Time>

- Example 1:
  - box293_check_vmware.pl --check Guest_CPU_Usage --server 192.168.1.211 --guest "vSphere Management Assistant (vMA)"
- Output for Example 1:
  - OK: {Free: 15,157 MHz} {Usage: (Total: 39 MHz) (CPU 0: 10 MHz) (CPU 1: 3 MHz) (CPU 2: 9 MHz) (CPU 3: 1 MHz)} {Total Available: 15,196 MHz} {Ready Time: (Total: 48 ms) (CPU 0: 14 ms) (CPU 1: 10 ms) (CPU 2: 15 ms) (CPU 3: 10 ms)}|'Total CPU Free'=15157MHz 'Total CPU Usage'=39MHz 'CPU 0: Usage'=10MHz 'CPU 1: Usage'=3MHz 'CPU 2: Usage'=9MHz 'CPU 3: Usage'=1MHz 'Total Available'=15196MHz 'Total Ready Time'=48ms 'CPU 0 Ready Time'=14ms 'CPU 1 Ready Time'=10ms 'CPU 2 Ready Time'=15ms 'CPU 3 Ready Time'=10ms [Guest_CPU_Usage]

- Example 2:
  - box293_check_vmware.pl --check Guest_CPU_Usage --server 192.168.1.211 --guest "vSphere Management Assistant (vMA)" --warning cpu_ready_time:10 --critical cpu_ready_time:20
- Output for Example 2:
  - CRITICAL: {Free: 15,061 MHz} {Usage: (Total: 135 MHz) (CPU 0: 16 MHz) (CPU 1: 4 MHz) (CPU 2: 97 MHz) (CPU 3: 2 MHz)} {Total Available: 15,196 MHz} {Ready Time: (Total: 54 ms

box293_check_vmware manual   *Page 54 / 85*

(CRITICAL >= 20)) (CPU 0: 18 ms) (CPU 1: 10 ms) (CPU 2: 16 ms) (CPU 3: 10 ms)}|'Total CPU Free'=15061MHz 'Total CPU Usage'=135MHz 'CPU 0: Usage'=16MHz 'CPU 1: Usage'=4MHz 'CPU 2: Usage'=97MHz 'CPU 3: Usage'=2MHz 'Total Available'=15196MHz 'Total Ready Time'=54ms;10;20 'CPU 0 Ready Time'=18ms 'CPU 1 Ready Time'=10ms 'CPU 2 Ready Time'=16ms 'CPU 3 Ready Time'=10ms [Guest_CPU_Usage]

- Example 3:
  - box293_check_vmware.pl --server vcenter.box293.local --check Guest_CPU_Usage --guest 'Windows Server' --perfdata_option CPU_Free:1,CPU_Free%:1,CPU_Used:1,CPU_Used %:1,CPU_Available:1,CPU_Ready_Time:1 --warning cpu_free%:20 --critical cpu_free%:10
- Output for Example 3:
  - OK: {Free: 6,749 MHz} {Free: 97%} {Usage: (Total: 233 MHz) (Total: 3%) (CPU 0: 233 MHz) (CPU 0: 3%) (CPU 1: 233 MHz) (CPU 1: 4%)} {Total Available: 6,982 MHz} {Ready Time: (Total: 55 ms) (CPU 0: 27 ms) (CPU 1: 28 ms)}|'Total CPU Free'=6749MHz 'Total CPU Free %'=97%;20;10 'Total CPU Usage'=233MHz 'Total CPU Usage %'=3% 'CPU 0: Usage'=92MHz 'CPU 0: Usage %'=3% 'CPU 1: Usage'=135MHz 'CPU 1: Usage %'=4% 'Total Available'=6982MHz 'Total Ready Time'=55ms 'CPU 0 Ready Time'=27ms 'CPU 1 Ready Time'=28ms [Guest_CPU_Usage]

Guest_Disk_Performance
- Description:
  - Check the Disk Performance of a Guests' virtual disk(s)
  - **NOTE:**
    - This check only works for guests running on ESX(i) hosts 4.1.0 onwards!
  - Metrics returned are:
    - Disk Rate (Read and Write) as kBps
    - Averaged Number of (Reads and Writes) (no thresholds checked)
    - Latency depending on ESX(i) host version:
      - Version 5.1.0 and later
        - Disk Latency (Read and Write) as us
      - Version less than 5.1.0
        - Total Latency (Read and Write) as ms
- Required Arguments:
  - --guest
- Optional Arguments:
  - --modifier
  - --perfdata_option post_check:disabled, Latency:1, Disk_Rate:1, Averaged:1
  - --reporting_si Disk_Rate:<Bytes Per Second>, Latency:<Time>
  - --warning and --critical disk_rate:<Bytes Per Second>, disk_latency:<Time>

- Example 1:
  - box293_check_vmware.pl --check Guest_Disk_Performance --server 192.168.1.211 --guest "VMware vCenter Server Appliance"
- Output for Example 1:
  - OK: [Hard disk 1 (scsi0:0) on 'ESXi 5.1' {Rate (Read:0 kBps / 0%)(Write:36 kBps / 100%)} {Averaged Number of (Reads:0) (Writes:2)} {Latency (Read:0 us)(Write:3,364 us)}], [Hard disk 2 (scsi0:1) on 'ESXi 5.1' {Rate (Read:0 kBps / 0%)(Write:9 kBps / 100%)} {Averaged Number of (Reads:0) (Writes:1)} {Latency (Read:0 us)(Write:407 us)}]|'Hard disk 1 (scsi0:0) Read Rate'=0kBps 'Hard disk 1 (scsi0:0) Write Rate'=36kBps 'Hard disk 1 (scsi0:0) Averaged Number of

box293_check_vmware manual   *Page 55 / 85*

Reads'=0 'Hard disk 1 (scsi0:0) Averaged Number of Writes'=2 'Hard disk 1 (scsi0:0) Read Latency'=0us 'Hard disk 1 (scsi0:0) Write Latency'=3364us 'Hard disk 2 (scsi0:1) Read Rate'=0kBps 'Hard disk 2 (scsi0:1) Write Rate'=9kBps 'Hard disk 2 (scsi0:1) Averaged Number of Reads'=0 'Hard disk 2 (scsi0:1) Averaged Number of Writes'=1 'Hard disk 2 (scsi0:1) Read Latency'=0us 'Hard disk 2 (scsi0:1) Write Latency'=407us [Guest_Disk_Performance]

- Example 2:
  ◦ box293_check_vmware.pl --check Guest_Disk_Performance --server 192.168.1.211 --guest "VMware vCenter Server Appliance" --warning disk_rate:2000 --critical disk_latency:3000
- Output for Example 2:
  ◦ CRITICAL: [Hard disk 1 (scsi0:0) on 'ESXi 5.1' {Rate (Read:0 kBps / 0%)(Write:36 kBps / 100%)} {Averaged Number of (Reads:0) (Writes:2)} {Latency (Read:0 us)(Write:3,175 us (CRITICAL >= 3,000))}], [Hard disk 2 (scsi0:1) on 'ESXi 5.1' {Rate (Read:0 kBps / 0%)(Write:25 kBps / 100%)} {Averaged Number of (Reads:0) (Writes:1)} {Latency (Read:12,673 us (CRITICAL >= 3,000)) (Write:717 us)}]|'Hard disk 1 (scsi0:0) Read Rate'=0kBps;2000 'Hard disk 1 (scsi0:0) Write Rate'=36kBps;2000 'Hard disk 1 (scsi0:0) Averaged Number of Reads'=0 'Hard disk 1 (scsi0:0) Averaged Number of Writes'=2 'Hard disk 1 (scsi0:0) Read Latency'=0us;;3000 'Hard disk 1 (scsi0:0) Write Latency'=3175us;;3000 'Hard disk 2 (scsi0:1) Read Rate'=0kBps;2000 'Hard disk 2 (scsi0:1) Write Rate'=25kBps;2000 'Hard disk 2 (scsi0:1) Averaged Number of Reads'=0 'Hard disk 2 (scsi0:1) Averaged Number of Writes'=1 'Hard disk 2 (scsi0:1) Read Latency'=12673us;;3000 'Hard disk 2 (scsi0:1) Write Latency'=717us;;3000 [Guest_Disk_Performance]

Guest_Disk_Usage
- Description:
  ◦ Check the Disk Usage of a Guests' virtual disk(s)
  ◦ Returned as GB by default
  ◦ Returns overall usage of all virtual disks and well as individual virtual disk usage
  ◦ Includes provisioning type (Thin, Thick Eager, Thick Lazy), swap files, suspend files and snapshot files
  ◦ For Thin provisioned disks, you can trigger thresholds on how much free disk space there is remaining (<total virtual disk size> - <current size on datastore>)
  ◦ Disk Total threshold is triggered on the overall total, not each individual disk
  ◦ **NOTE:** This is NOT checking the disk usage of the internal file system running inside the guest!
- Required Arguments:
  ◦ --guest
- Optional Arguments:
  ◦ --modifier
  ◦ --perfdata_option post_check:disabled, Disk_Capacity:1, Disk_Free:1, Disk_Free%:1, Disk_Size_On_Datastore:1, Disk_Snapshot_Space:1, Disk_Suspend_File:1, Disk_Swap_File:1, Disk_Swap_Userworld:1, Disk_Usage:1
  ◦ --reporting_si Disk_Size:<Bytes>
  ◦ --warning and --critical disk_free:<Bytes>, disk_free%:<value>, disk_total:<Bytes>

- Example 1:
  ◦ box293_check_vmware.pl --check Guest_Disk_Usage --server 192.168.1.211 --guest "VMware vCenter Server Appliance"
- Output for Example 1:
  ◦ OK: [Totals: {Disk Usage: 21.6 GB} {Swap File: 8 GB} {Userworld Swap File: 0.1 GB}], [Hard disk 1 (scsi0:0) on 'ESXi 5.1' {Provisioning: Thin} {Disk Capacity: 25 GB} {Size On Datastore: 8.6

box293_check_vmware manual *Page 56 / 85*

GB} {Free Space: 16.4 GB / 66%}], [Hard disk 2 (scsi0:1) on 'ESXi 5.1' {Provisioning: Thin} {Disk Capacity: 60 GB} {Size On Datastore: 4.9 GB} {Free Space: 55.1 GB / 92%}]|'Total Disk Usage'=21.6GB 'Swap File'=8GB 'Userworld Swap File'=0.1GB 'Hard disk 1 (scsi0:0) Capacity'=25GB 'Hard disk 1 (scsi0:0) Size On Datastore'=8.6GB 'Hard disk 1 (scsi0:0) Free Space'=16.4GB 'Hard disk 2 (scsi0:1) Capacity'=60GB 'Hard disk 2 (scsi0:1) Size On Datastore'=4.9GB 'Hard disk 2 (scsi0:1) Free Space'=55.1GB [Guest_Disk_Usage]

- Example 2:
  - box293_check_vmware.pl --check Guest_Disk_Usage --server 192.168.1.211 --guest "Windows 8 Development" --critical disk_free:60
- Output for Example 2:
  - CRITICAL: [Totals: {Disk Usage: 35.3 GB} {Suspend File: 4 GB} {All Snapshot Space: 4.1 GB}], [Hard disk 1 (scsi0:0) on 'RAID1 (1)(3)' {Provisioning: Thin} {Disk Capacity: 80.5 GB} {Size On Datastore: 27.2 GB} {Free Space: 53.3 GB / 67% (CRITICAL <= 60)}]|'Total Disk Usage'=35.3GB 'Suspend File'=4GB 'All Snapshot Space'=4.1GB 'Hard disk 1 (scsi0:0) Capacity'=80.5GB 'Hard disk 1 (scsi0:0) Size On Datastore'=27.2GB 'Hard disk 1 (scsi0:0) Free Space'=53.3GB;;60 [Guest_Disk_Usage]

- Example 3:
  - box293_check_vmware.pl --server vcenter.box293.local --check Guest_Disk_Usage --guest 'Windows Server' --perfdata_option Disk_Capacity:1,Disk_Free:1,Disk_Free %:1,Disk_Size_On_Datastore:1,Disk_Snapshot_Space:1,Disk_Suspend_File:1,Disk_Swap_File:1,Disk_Swap_Userworld:1,Disk_Usage:1 --warning disk_free%:20 Development" --critical disk_free:60
- Output for Example 3:
  - WARNING: [Totals: {Disk Usage: 1,441.4 GB} {Swap File: 3 GB} {Userworld Swap File: 0.2 GB}], [Hard disk 1 (scsi0:0) on 'VOL_04_05' {Provisioning: Thin} {Disk Capacity: 200.5 GB} {Size On Datastore: 178.8 GB} {Free Space: 21.7 GB} {Free Space: 11% (WARNING <= 20)}], [Hard disk 2 (scsi0:1) on 'VOL_04_05' {Provisioning: Thin} {Disk Capacity: 715.5 GB} {Size On Datastore: 640.5 GB} {Free Space: 75.0 GB} {Free Space: 11% (WARNING <= 20)}], [Hard disk 3 (scsi0:2) on 'VOL_04_05' {Provisioning: Thin} {Disk Capacity: 725.5 GB} {Size On Datastore: 8.1 GB} {Free Space: 717.4 GB} {Free Space: 99%}], [Hard disk 4 (scsi0:3) on 'VOL_04_05' {Provisioning: Thin} {Disk Capacity: 735.5 GB} {Size On Datastore: 610.8 GB} {Free Space: 124.7 GB} {Free Space: 17% (WARNING <= 20)}]|'Total Disk Usage'=1441.4GB 'Swap File'=3GB 'Userworld Swap File'=0.2GB 'Hard disk 1 (scsi0:0) Capacity'=200.5GB 'Hard disk 1 (scsi0:0) Size On Datastore'=178.8GB 'Hard disk 1 (scsi0:0) Free Space'=21.7GB 'Hard disk 1 (scsi0:0) Free Space %'=11%;20 'Hard disk 2 (scsi0:1) Capacity'=715.5GB 'Hard disk 2 (scsi0:1) Size On Datastore'=640.5GB 'Hard disk 2 (scsi0:1) Free Space'=75.0GB 'Hard disk 2 (scsi0:1) Free Space %'=11%;20 'Hard disk 3 (scsi0:2) Capacity'=725.5GB 'Hard disk 3 (scsi0:2) Size On Datastore'=8.1GB 'Hard disk 3 (scsi0:2) Free Space'=717.4GB 'Hard disk 3 (scsi0:2) Free Space %'=99%;20 'Hard disk 4 (scsi0:3) Capacity'=735.5GB 'Hard disk 4 (scsi0:3) Size On Datastore'=610.8GB 'Hard disk 4 (scsi0:3) Free Space'=124.7GB 'Hard disk 4 (scsi0:3) Free Space %'=17%;20 [Guest_Disk_Usage]

box293_check_vmware manual   *Page 57 / 85*

Guest_Host
- Description:
  - Compare the ESX(i) host the guest is running on against the parent_hosts defined for this Nagios host object
  - If ESX(i) and Nagios do not match then a WARNING state is triggered
  - This uses the Nagios objectjson.cgi to query Nagios directly to determine this
  - The purpose of this check is return the name of the ESX(i) host the guest is currently running on. IF a WARNING state is triggered then Nagios will execute the box293_event_handler to update the directive parent_hosts in this Nagios host object definition.
    - *NOTE: The box293_event_handler is currently a work in progress, stay tuned!*
- Required Arguments:
  - --guest
  - --query_url
  - --query_username
  - --query_password
  - --service_status_info
- Optional Arguments:
  - --modifier
- Other Requirements:
  - Nagios Core 4.0.8 is required on your Nagios server for the objectjson.cgi query to work *(it **may** work from 4.0.4 onwards however it has only been tested with 4.0.8)*

- Example 1:
  - box293_check_vmware.pl --server vcenter.box293.local --check Guest_Host --guest windows10preview.box293.local --query_url 'http://xitest.box293.local/nagios/cgi-bin/objectjson.cgi' --query_username 'readonly' --query_password 'AV3ryStr0ngP@ssw0rd' --service_status_info "Current_Parent=esxi001.box293.local"
- Output for Example 1:
  - Current_Parent=esxi001.box293.local

- Example 2:
  - box293_check_vmware.pl --server vcenter.box293.local --check Guest_Host --guest HOST001 --query_url 'http://xitest.box293.local/nagios/cgi-bin/objectjson.cgi' --query_username 'readonly' --query_password 'AV3ryStr0ngP@ssw0rd' --service_status_info ""
- Output for Example 2:
  - No_Parent_Defined_-_Should_Be=esxi001.box293.local
- **Note:** These two examples were executed at the command line. In a service definition, the --service_status_info argument would be followed with $SERVICEOUTPUT$
  - Example:
    - --service_status_info "$SERVICEOUTPUT$"

box293_check_vmware manual   *Page 58 / 85*

Guest_Memory_Info
- Description:
  - Report Information about the Guests' Memory such as Total, Reservation and Limit
    - Returned as MB by default
    - Performance data can be useful for identifying when changes occurred, like adding Memory or when a reservation or limit was defined
    - **NOTE:**
      - memory_reservation and memory_limit thresholds are either --warning OR --critical, NOT both
      - Memory Reservation only reported on directly connected ESXi hosts v 5.0 onwards ... via vCenter works for 4.0 onwards
- Required Arguments:
  - --guest
- Optional Arguments:
  - --modifier
  - --perfdata_option post_check:disabled, Memory_Limit:1, Memory_Reservation:1, Memory_Total:1
  - --reporting_si Memory_Size:<Bytes>
  - --warning and --critical memory_reservation:<Bytes>, memory_limit:<Bytes>

- Example 1:
  - box293_check_vmware.pl --check Guest_Memory_Info --server 192.168.1.211 --guest "Windows 8 Development"
- Output for Example 1:
  - OK: {Total: 4,096 MB} {Reservation MB: 0 MB} {Limit MB: 4,096 MB}|'Memory Total'=4096MB 'Memory Reservation'=0MB 'Memory Limit'=4096MB [Guest_Memory_Info]

- Example 2:
  - box293_check_vmware.pl --check Guest_Memory_Info --server 192.168.1.211 --guest "Windows 8 Development" --warning memory_limit:3096 --critical memory_reservation:1024
- Output for Example 2:
  - CRITICAL: {Total: 4,096 MB} {Reservation MB: 0 MB (CRITICAL != 1,024)} {Limit MB: 4,096 MB (WARNING != 3,096)}|'Memory Total'=4096MB 'Memory Reservation'=0MB;;1024 'Memory Limit'=4096MB;3096 [Guest_Memory_Info]

Guest_Memory_Usage
- Description:
  - Report the specified Guests' Memory Usage
  - Returned as MB and kBps by default
  - Memory Swapping is reported depending on ESX(i) host version:
    - version 4.0.0 and later
      - Swapping Rate (In and Out) as kBps
    - version less than 4.0.0
      - Swapped (In and Out) as MB
- Required Arguments:
  - --guest
- Optional Arguments:
  - --modifier
  - --perfdata_option post_check:disabled, Memory_Active:1, Memory_Ballooned:1,

box293_check_vmware manual   *Page 59 / 85*

Memory_Consumed:1, Memory_Consumed%:1, Memory_Free:1, Memory_Free%:1, Memory_Overhead:1, Memory_Shared:1, Memory_Swap:1, Memory_Total:1

- ○ --reporting_si Memory_Size:<Bytes>, Memory_Rate:<Bytes Per Second>
- ○ --warning and --critical memory_free:<Bytes>, memory_free%:<value>, memory_consumed:<Bytes>, memory_consumed%:<value>, memory_ballooned:<Bytes>, memory_swap_rate:<Bytes Per Second>, memory_swapped:<Bytes>

- • Example 1:
  - ○ box293_check_vmware.pl --check Guest_Memory_Usage --server 192.168.1.211 --guest "vSphere Management Assistant (vMA)"
- • Output for Example 1:
  - ○ OK: Guest Memory {Free: 3,468.0 MB} {Consumed: 628 MB} {Total: 4,096 MB} {Ballooned: 0 MB} {Overhead: 32.7 MB} {Active: 163.8 MB} {Shared: 0 MB} {Swap Rate (In: 0 kBps)(Out: 0 kBps)}|'Memory Free'=3468.0MB 'Memory Consumed'=628MB 'Memory Total'=4096MB 'Memory Ballooned'=0MB 'Memory Overhead'=32.7MB 'Memory Active'=163.8MB 'Memory Shared'=0MB 'Swapping Rate In'=0kBps 'Swapping Rate Out'=0kBps [Guest_Memory_Usage]

- • Example 2:
  - ○ box293_check_vmware.pl --check Guest_Memory_Usage --server 192.168.1.211 --guest "vSphere Management Assistant (vMA)" --warning memory_free:3500 --critical memory_free:2500
- • Output for Example 2:
  - ○ WARNING: Guest Memory {Free: 3,468.0 MB (WARNING <= 3,500)} {Consumed: 628 MB} {Total: 4,096 MB} {Ballooned: 0 MB} {Overhead: 32.7 MB} {Active: 81.9 MB} {Shared: 0 MB} {Swap Rate (In: 0 kBps)(Out: 0 kBps)}|'Memory Free'=3468.0MB;3500;2500 'Memory Consumed'=628MB 'Memory Total'=4096MB 'Memory Ballooned'=0MB 'Memory Overhead'=32.7MB 'Memory Active'=81.9MB 'Memory Shared'=0MB 'Swapping Rate In'=0kBps 'Swapping Rate Out'=0kBps [Guest_Memory_Usage]

- • Example 3:
  - ○ box293_check_vmware.pl --server vcenter.box293.local --check Guest_Memory_Usage --guest 'Windows Server' --perfdata_option Memory_Active:1,Memory_Ballooned:1,Memory_Consumed:1,Memory_Consumed%:1,Memory_Free:1,Memory_Free%:1,Memory_Overhead:1,Memory_Shared:1,Memory_Swap:1,Memory_Total:1 --warning memory_free%:20
- • Output for Example 3:
  - ○ WARNING: Guest Memory {Free: 0.0 MB} {Free: 0% (WARNING <= 20)} {Consumed: 3,072 MB} {Consumed: 100%} {Total: 3,072 MB} {Ballooned: 0 MB} {Overhead: 50.2 MB} {Active: 1,351.7 MB} {Shared: 0 MB} {Swap Rate (In: 0 kBps)(Out: 0 kBps)}|'Memory Free'=0.0MB 'Memory Free %'=0%;20 'Memory Consumed'=3072MB 'Memory Consumed %'=100% 'Memory Total'=3072MB 'Memory Ballooned'=0MB 'Memory Overhead'=50.2MB 'Memory Active'=1351.7MB 'Memory Shared'=0MB 'Swapping Rate In'=0kBps 'Swapping Rate Out'=0kBps [Guest_Memory_Usage]

box293_check_vmware manual   *Page 60 / 85*

Guest_NIC_Usage
- Description:
    - Check the Guests' Network Interface Card(s) (NIC) usage
    - Returned as kBps by default
    - If the guest has more than one NIC then each NIC will be reported on individually
    - Metrics returned are:
        - Rate (Received and Transmitted) as kBps
        - Packets (Received and Transmitted) (no thresholds checked)
            - Packets only reported for VMs running on ESXi hosts 5.0 onwards
- Required Arguments:
    - --guest
- Optional Arguments:
    - --modifier
    - --perfdata_option post_check:disabled, NIC_Rate:1, NIC_Packets:1
    - --reporting_si NIC_Rate:<Bytes Per Second>
    - --warning and --critical nic_rate:<Bytes Per Second>

- Example 1:
    - box293_check_vmware.pl --check Guest_NIC_Usage --server 192.168.1.211 --guest "vSphere Management Assistant (vMA)"
- Output for Example 1:
    - OK: {Rate (Rx:236 kBps / 96%)(Tx:11 kBps / 4%)} {Packets (Rx:3,669)(Tx:875)}|'Rate Rx'=236kBps 'Rate Tx'=11kBps 'Packets Rx'=3669 'Packets Tx'=875 [Guest_NIC_Usage]

- Example 2:
    - box293_check_vmware.pl --check Guest_NIC_Usage --server 192.168.1.211 --guest "vSphere Management Assistant (vMA)" --warning nic_rate:100 --critical nic_rate:200
- Output for Example 2:
    - WARNING: {Rate (Rx:192 kBps / 93% (WARNING >= 100))(Tx:16 kBps / 7%)} {Packets (Rx:3,166)(Tx:1,043)}|'Rate Rx'=192kBps;100;200 'Rate Tx'=16kBps;100;200 'Packets Rx'=3166 'Packets Tx'=1043 [Guest_NIC_Usage]

Guest_Snapshot
- Description:
    - Check if a guest has snapshots and trigger warning or critical states if snapshot is X days old
    - If a guest has multiple snapshots, the oldest snapshot is checked
    - You can target an individual Guest or multiple guests via a Host, Cluster or Datacenter
- Required Arguments:
    - --guest or --host or --cluster or --datacenter
- Optional Arguments:
    - --exclude_snapshot
    - --modifier
    - --perfdata_option post_check:disabled
    - --warning and --critical snapshot_age:<Day(s)>

- Example 1:
    - box293_check_vmware.pl --check Guest_Snapshot --server 192.168.1.211 --guest "vSphere Management Assistant (vMA)"

box293_check_vmware manual   *Page 61 / 85*

- Output for Example 1:
  - OK: No snapshots found

- Example 2:
  - box293_check_vmware.pl --check Guest_Snapshot --server 192.168.1.211 --datacenter Box293 --warning snapshot_age:5 --critical snapshot_age:15
- Output for Example 2:
  - CRITICAL: ['nagiosxi' (Notes: Tester) (Age: 2)], ['Windows XP' (Notes: Test) (Age: 15 (CRITICAL >= 15))], ['Windows Server 2003 x86' (Notes: Test) (Age: 20 (CRITICAL >= 15))], ['Debian 7 x86' (Notes: Another Test) (Age: 15 (CRITICAL >= 15))], ['Linux Mint 15' (Notes: asdas) (Age: 13 (WARNING >= 5))], ['vMA' (Notes: before configuring) (Age: 12 (WARNING >= 5))]

Guest_Status
- Description:
  - Check the status of a guest including Power State, Uptime, VMware Tools Version and Status, IP Address, Hostname, ESX(i) Host Guest Is Running On, Consolidation State and Guest Version
  - Warning or critical states can be triggered for Power State, Uptime, Consolidation State and VMware Tools Status
  - Uptime only reported for guests running on ESXi hosts 4.1 onwards
  - Consolidation state only reported for VMs running on ESXi hosts 5.0 onwards
- Required Arguments:
  - --guest
- Optional Arguments:
  - --modifier
  - --guest_consolidation_state
  - --guest_power_state
  - --guest_tools_version_state
  - --warning and --critical guest_uptime:<Day(s)>

- Example 1:
  - box293_check_vmware.pl --check Guest_Status --server 192.168.1.211 --guest "vSphere Management Assistant (vMA)"
- Output for Example 1:
  - OK: {State: poweredOn} {Uptime: 13 d} {Tools (Version: 2147483647) (Status: guestToolsUnmanaged)} {IP Address: 10.25.6.22} {Guest Hostname: vmademo} {Host: esxi001.box293.local} {Guest Version: vmx-07}

- Example 2:
  - box293_check_vmware.pl --check Guest_Status --server 192.168.1.211 --guest "vSphere Management Assistant (vMA)" --guest_tools_version_state guestToolsUnmanaged:WARNING --warning guest_uptime:15 --critical guest_uptime:1
- Output for Example 2:
  - WARNING: {State: poweredOn} {Uptime: 13 d (WARNING <= 15)} {Tools (Version: 2147483647) (Status: guestToolsUnmanaged (WARNING))} {IP Address: 10.25.6.22} {Guest Hostname: vmademo} {Host: esxi001.box293.local} {Guest Version: vmx-07}

box293_check_vmware manual   *Page 62 / 85*

Host_CPU_Info
- Description:
  - Report Information about the Hosts' CPU such as Model, Cores and Total CPU
  - Returned as GHz by default
  - **NOTE:**
    - If the host is in Standby mode then the check will exit with an OK state
- Required Arguments:
  - --host (if connected via a vCenter server)
- Optional Arguments:
  - --modifier
  - --reporting_si CPU_Speed:<Hertz>

- Example 1:
  - box293_check_vmware.pl --check Host_CPU_Info --server 192.168.1.211 --host 192.168.1.210
- Output for Example 1:
  - OK: AMD A10-5800K APU with Radeon(tm) HD Graphics, 4 cores @ 3.8GHz

- Example 2:
  - box293_check_vmware.pl --check Host_CPU_Info --server 192.168.1.211 --host 192.168.1.210 --reporting_si CPU_Speed:MHz
- Output for Example 2:
  - OK: AMD A10-5800K APU with Radeon(tm) HD Graphics, 4 cores @ 3799MHz

Host_CPU_Usage
- Description:
  - Report the specified Hosts' CPU Usage
  - Returned as GHz by default
  - **NOTE:**
    - If the host is in Standby mode then the check will exit with an OK state
- Required Arguments:
  - --host (if connected via a vCenter server)
- Optional Arguments:
  - --modifier
  - --perfdata_option post_check:disabled, CPU_Free:1, CPU_Free%:1, CPU_Total:1, CPU_Used:1, CPU_Used%:1
  - --reporting_si CPU_Speed:<Hertz>
  - --warning and --critical cpu_free:<Hertz>, cpu_free%:<value>, cpu_used:<Hertz>, cpu_used%:<value>

- Example 1:
  - box293_check_vmware.pl --check Host_CPU_Usage --server 192.168.1.211 --host 192.168.1.210
- Output for Example 1:
  - OK: Host CPU {Free: 12.9 GHz} {Used: 2.3 GHz} {Total: 15.2 GHz}|'CPU Free'=12.9GHz 'CPU Used'=2.3GHz 'CPU Total'=15.2GHz [Host_CPU_Usage]

- Example 2:
  - box293_check_vmware.pl --check Host_CPU_Usage --server 192.168.1.211 --host 192.168.1.210 --warning cpu_used:2 --critical cpu_used:10

box293_check_vmware manual   *Page 63 / 85*

- Output for Example 2:
  - WARNING: Host CPU {Free: 12.5 GHz} {Used: 2.7 GHz (WARNING >= 2)} {Total: 15.2 GHz}|'CPU Free'=12.5GHz 'CPU Used'=2.7GHz;2;10 'CPU Total'=15.2GHz [Host_CPU_Usage]

- Example 3:
  - box293_check_vmware.pl --server vcenter.box293.local --check Host_CPU_Usage --host esxi001.box293.local --perfdata_option CPU_Free:1,CPU_Free %:1,CPU_Total:1,CPU_Used:1,CPU_Used%:1 --warning cpu_used%:70
- Output for Example 3:
  - OK: Host CPU {Free: 19.4 GHz} {Free: 93%} {Used: 1.5 GHz} {Used: 7%} {Total: 20.9 GHz}|'CPU Free'=19.4GHz 'CPU Free %'=93% 'CPU Used'=1.5GHz 'CPU Used %'=7%;70 'CPU Total'=20.9GHz [Host_CPU_Usage]

Host_License_Status
- Description:
  - Reports the specified Hosts license status along with the license key
  - If the host is in evaluation mode and has less than 24 hours remaining a CRITICAL state is returned otherwise a WARNING state is returned
  - When host is queried using an account that has read only privileges, only a portion of the key is displayed
  - **NOTE:**
    - If the host is in Standby mode then the check will exit with an OK state
- Required Arguments:
  - --host (if connected via a vCenter server)
- Optional Arguments:
  - --modifier
  - --hide_key

- Example 1:
  - box293_check_vmware.pl --check Host_License_Status --server 192.168.1.211 --host 192.168.1.210
- Output for Example 1:
  - OK: Licensed {Version: VMware vSphere 5 Enterprise Plus}{Key: ABCDE-#####-#####-#####-VWXYZ}

- Example 2:
  - box293_check_vmware.pl --check Host_License_Status --server 192.168.1.43
- Output for Example 2:
  - WARNING: Evaluation Mode, Evaluation Period Remaining: 43 days

box293_check_vmware manual   *Page 64 / 85*

Host_Memory_Usage
- Description:
  - Report the specified Hosts' Memory Usage
  - Returned as GB by default
  - **NOTE:**
    - If the host is in Standby mode then the check will exit with an OK state
- Required Arguments:
  - --host (if connected via a vCenter server)
- Optional Arguments:
  - --modifier
  - --perfdata_option post_check:disabled, Memory_Free:1, Memory_Free%:1, Memory_Total:1, Memory_Used:1, Memory_Used%:1
  - --reporting_si Memory_Size:<Bytes>
  - --warning and --critical memory_free:<Bytes>, memory_free%:<value>, memory_used:<Bytes>, memory_used%:<value>

- Example 1:
  - box293_check_vmware.pl --check Host_Memory_Usage --server 192.168.1.211 --host 192.168.1.210
- Output for Example 1:
  - OK: Host Memory {Free: 9.2 GB} {Used: 22.8 GB} {Total: 32 GB}|'Memory Free'=9.2GB 'Memory Used'=22.8GB 'Memory Total'=32GB [Host_Memory_Usage]

- Example 2:
  - box293_check_vmware.pl --check Host_Memory_Usage --server 192.168.1.211 --host 192.168.1.210 --reporting_si Memory_Size:MB --warning memory_free:10000 --critical memory_free:5000
- Output for Example 2:
  - WARNING: Host Memory {Free: 8,549 MB (WARNING <= 10,000)} {Used: 23,396 MB} {Total: 31,945 MB}|'Memory Free'=8549MB;10000;5000 'Memory Used'=23396MB 'Memory Total'=31945MB [Host_Memory_Usage]

- Example 3:
  - box293_check_vmware.pl --server vcenter.box293.local --check Host_Memory_Usage --host esxi001.box293.local --perfdata_option Memory_Free:1,Memory_Free%:1,Memory_Total:1,Memory_Used:1,Memory_Used%:1 --warning memory_used%:70
- Output for Example 3:
  - OK: Host Memory {Free: 47.5 GB} {Free: 38%} {Used: 80.5 GB} {Used: 62%} {Total: 128 GB}|'Memory Free'=47.5GB 'Memory Free %'=38% 'Memory Used'=80.5GB 'Memory Used %'=62%;70 'Memory Total'=128GB [Host_Memory_Usage]

box293_check_vmware manual　　　　　　　　　　　　　　　　　　　　　*Page 65 / 85*

Host_OS_Name_Version
- Description:
  - Report the specified Hosts' product Name and Version
  - **NOTE:**
    - If the host is in Standby mode then the check will exit with an OK state
- Required Arguments:
  - --host (if connected via a vCenter server)
- Optional Arguments:
  - --modifier

- Example 1:
  - box293_check_vmware.pl --check Host_OS_Name_Version --server 192.168.1.211 --host 192.168.1.210
- Output for Example 1:
  - OK: VMware ESXi 5.1.0 build-1065491

Host_pNIC_Status
- Description:
  - Check the Status of the Hosts' Physical Network Interface Card(s) (pNIC)
  - All pNICs are returned by default however you can target specific pNIC(s)
  - If any pNIC is disconnected it will trigger a CRITICAL state
  - **NOTE:**
    - If the host is in Standby mode then the check will exit with an OK state
- Required Arguments:
  - --host (if connected via a vCenter server)
- Optional Arguments:
  - --modifier
  - --name
  - --nic_state
  - --nic_speed
  - --nic_duplex

- Example 1:
  - box293_check_vmware.pl --check Host_pNIC_Status --server 192.168.1.211 --host 192.168.1.210
- Output for Example 1:
  - CRITICAL: NICs [Total: 3, Connected: 1, Disconnected: 2], [vmnic1 on Local vSwitch 'vSwitch0', Driver: e1000e, NOT Connected], [vmnic0 on Local vSwitch 'vSwitch0', Driver: r8168, NOT Connected], [vmnic2 on Local vSwitch 'vSwitch0', Driver: e1000e, 1,000 MB, Full Duplex]

- Example 2:
  - box293_check_vmware.pl --check Host_pNIC_Status --server 192.168.1.211 --host 192.168.1.210 --name vmnic2
- Output for Example 2:
  - OK: NICs [Total: 3, Connected: 1], [vmnic2 on Local vSwitch 'vSwitch0', Driver: e1000e, 1,000 MB, Full Duplex]

box293_check_vmware manual   *Page 66 / 85*

Host_pNIC_Usage
- Description:
  - Check the Hosts' Physical Network Interface Card(s) (pNIC) usage
  - Returned as kBps by default
  - All pNICs are returned by default however you can target specific pNIC(s)
  - Metrics returned are:
    - Rate (Received and Transmitted) as kBps
    - Packets (Received and Transmitted) (no thresholds checked)
    - ESX(i) host version 5.0.0 and later also reports:
      - Packet Errors (Received and Transmitted)
  - TIP: group pNICs that are in the same vSwitch (like a dedicated iSCSI vSwitch) and create separate checks for different roles, this makes viewing performance data easier
  - **NOTE:**
    - If the host is in Standby mode then the check will exit with an OK state
- Required Arguments:
  - --host (if connected via a vCenter server)
- Optional Arguments:
  - --modifier
  - --name
  - --perfdata_option post_check:disabled, NIC_Rate:1, NIC_Packets:1, NIC_Packet_Errors:1
  - --reporting_si NIC_Rate:<Bytes Per Second>
  - --warning and --critical nic_rate:<Bytes Per Second>, packet_errors:<Number>

- Example 1:
  - box293_check_vmware.pl --check Host_pNIC_Usage --server 192.168.1.211 --host 192.168.1.21
- Output for Example 1:
  - OK: [vmnic0 {Rate (Rx:1 kBps / 100%)(Tx:0 kBps / 0%)} {Packets (Rx:138)(Tx:24)} {Packet Errors (Rx:0)(Tx:0)}]|'vmnic0 Rate Rx'=1kBps 'vmnic0 Rate Tx'=0kBps 'vmnic0 Packets Rx'=138 'vmnic0 Packets Tx'=24 'vmnic0 Packet Errors Rx'=0 'vmnic0 Packet Errors Tx'=0 [Host_pNIC_Usage]

- Example 2:
  - box293_check_vmware.pl --check Host_pNIC_Usage --server 192.168.1.211 --host 192.168.1.210 --name vmnic2 --reporting_si NIC_Rate:Bps
- Output for Example 2:
  - OK: [vmnic2 {Rate (Rx:2,048 Bps / 50%)(Tx:2,048 Bps / 50%)} {Packets (Rx:320)(Tx:283)} {Packet Errors (Rx:0)(Tx:0)}]|'vmnic2 Rate Rx'=2048Bps 'vmnic2 Rate Tx'=2048Bps 'vmnic2 Packets Rx'=320 'vmnic2 Packets Tx'=283 'vmnic2 Packet Errors Rx'=0 'vmnic2 Packet Errors Tx'=0 [Host_pNIC_Usage]

box293_check_vmware manual   *Page 67 / 85*

Host_Service
- Description:
  - Check the services running on a Host, their startup policy or if they are running
  - Startup Policy:
    - There are three startup policy options for a service
      - 0 = off = Start and stop manually
      - 1 = on = Start and stop with host
      - 2 = automatic = Start automatically if any ports are open, and stop when all ports are closed
    - The numbers are what will be used in the warning and/or critical arguments
  - Running
    - A service can be:
      - 0 = no = The service is NOT currently running
      - 1 = yes = The service IS currently running
    - The numbers are what will be used in the warning and/or critical arguments
- Services
  - Services can be targeted by the warning and critical arguments using the short name for the service and is CaSe sEnSaTivE
  - An ESXi host has the following default services:
    - DCUI = Direct Console UI
    - TSM = ESXi Shell
    - TSM-SSH = SSH
    - lbtd = Load-Based Teaming Daemon
    - lwsmd = Active Directory Service
    - ntpd = NTP Daemon
    - pcscd = PC/SC Smart Card Daemon
    - sfcbd-watchdog = CIM Server
    - snmpd = SNMP Server
    - vmsyslogd = Syslog Server
    - vprobed = VProbe Daemon
    - vpxa = VMware vCenter Agent
    - xorg = X.Org Server
- Thresholds
  - If you do not supply a warning or critical threshold, the check will return an OK state and will report on all the services
  - When you supply a warning or critical threshold, the check will ONLY report on the services being targeted by the thresholds
  - If you want to check the startup policy you use the format:
    - shortname_policy:<value>
    - Example: --critical ntpd_policy:1
  - If you want to check if a service is running you use the format:
    - shortname_running:<value>
    - Example: --warning TSM-SSH_running:0
  - If you supply both a warning and critical for the same threshold, critical will take precedence
- Required Arguments:
  - --host (if connected via a vCenter server)
- Optional Arguments:
  - --modifier
  - --warning and --critical (see notes about thresholds)

box293_check_vmware manual   *Page 68 / 85*

- Example 1:
  - box293_check_vmware.pl --server vcenter.box293.local --check Host_Service --host host601.box293.local
- Output for Example 1:
  - OK: Host Services [Direct Console UI (DCUI) {Policy: on} {Running: yes}] [ESXi Shell (TSM) {Policy: off} {Running: no}] [SSH (TSM-SSH) {Policy: off} {Running: no}] [Load-Based Teaming Daemon (lbtd) {Policy: on} {Running: yes}] [Active Directory Service (lwsmd) {Policy: off} {Running: no}] [NTP Daemon (ntpd) {Policy: automatic} {Running: no}] [PC/SC Smart Card Daemon (pcscd) {Policy: off} {Running: no}] [CIM Server (sfcbd-watchdog) {Policy: on} {Running: no}] [SNMP Server (snmpd) {Policy: on} {Running: no}] [Syslog Server (vmsyslogd) {Policy: on} {Running: yes}] [VProbe Daemon (vprobed) {Policy: off} {Running: no}] [VMware vCenter Agent (vpxa) {Policy: on} {Running: yes}] [X.Org Server (xorg) {Policy: on} {Running: no}]

- Example 2:
  - box293_check_vmware.pl --server vcenter.box293.local --check Host_Service --host host601.box293.local --warning ntpd_policy:2 --critical ntpd_running:1,TSM-SSH_policy:0,TSM-SSH_running:0
- Output for Example 2:
  - CRITICAL: Host Services [SSH (TSM-SSH) {Policy: off} {Running: no}] [NTP Daemon (ntpd) {Policy: automatic} {Running: no (CRITICAL should be yes)}]

Host_Status
- Description:
  - Check the overall status of a Host and report any known issues
  - Reports Triggered Alarms and will trigger warning and critical states if the alarms have not been acknowledged in vCenter
  - **NOTE:**
    - If the host is in Standby mode then the check will exit with an OK state
- Required Arguments:
  - --host (if connected via a vCenter server)
- Optional Arguments:
  - --modifier
  - --exclude_issue

- Example 1:
  - box293_check_vmware.pl --check Host_Status --server 192.168.1.211 --host 192.168.1.210
- Output for Example 1:
  - WARNING: Host has a YELLOW status {Local Tech Support Mode ENABLED, Remote Tech Support Mode ENABLED}

- Example 2:
  - box293_check_vmware.pl --check Host_Status --server 192.168.1.211 --host 192.168.1.210 --exclude_issue LocalTSMEnabledEvent,RemoteTSMEnabledEvent
- Output for Example 2:
  - OK: No problems detected

- Example 3:
  - box293_check_vmware.pl --check Host_Status --server vcenter.box293.local --host

box293_check_vmware manual   *Page 69 / 85*

esxi008.box293.local
- Output for Example 3:
  - CRITICAL: {Alarms Total: 2 (#1: NOT Acknowledged, red=CRITICAL, Age: 314 Days) (#2: NOT Acknowledged, red=CRITICAL, Age: 66 Days)}

Host_Storage_Adapter_Info
- Description:
  - Report Information about the Hosts' Storage Adapter(s) such as Model, Device Name and Driver Name (driver version not available at this point in time)
  - **NOTE:**
    - If the host is in Standby mode then the check will exit with an OK state
- Required Arguments:
  - --host (if connected via a vCenter server)
- Optional Arguments:
  - --modifier
  - --name

- Example 1:
  - box293_check_vmware.pl --check Host_Storage_Adapter_Info --server 192.168.1.211 --host 192.168.1.210
- Output for Example 1:
  - OK: [LSI 3ware 9750 (vmhba1) {Driver: 3w-sas}], [AMD Hudson SATA Controller [AHCI Mode] (vmhba0) {Driver: ahci}]

- Example 2:
  - box293_check_vmware.pl --check Host_Storage_Adapter_Info --server 192.168.1.211 --host 192.168.1.210 --name vmhba1
- Output for Example 2:
  - OK: [LSI 3ware 9750 (vmhba1) {Driver: 3w-sas}]

Host_Storage_Adapter_Performance
- Description:
  - Check the performance of a specific Hosts' Storage Adapter
  - Metrics returned are:
    - Rate (Read and Write) as kBps
    - Average Number of (Reads and Writes) (no thresholds checked)
    - Latency Total (Read and Write) as ms
  - You need to provide the Device Name of at least one Storage Adapter to check
  - **NOTE:**
    - If the host is in Standby mode then the check will exit with an OK state
    - This check will NOT work with hosts less than version 4.1
- Required Arguments:
  - --host (if connected via a vCenter server)
  - --name
- Optional Arguments:
  - --modifier
  - --perfdata_option post_check:disabled, Averaged:1, HBA_Latency:1, HBA_Rate:1

box293_check_vmware manual   *Page 70 / 85*

- ○ --reporting_si HBA_Rate:<Bytes Per Second>, Latency:<Time>
- ○ --warning and --critical hba_rate:<Bytes Per Second>, hba_latency:<Time>

- • Example 1:
  - ○ box293_check_vmware.pl --check Host_Storage_Adapter_Performance --server 192.168.1.211 --host 192.168.1.210 --name vmhba1
- • Output for Example 1:
  - ○ OK: [LSI 3ware 9750 (vmhba1) {Rate (Read:493 kBps / 23%)(Write:1,720 kBps / 77%)} {Averaged Number of (Reads:81) (Writes:168)} {Total Latency (Read:2 ms)(Write:0 ms)}]|'vmhba1 Rate Read'=493kBps 'vmhba1 Rate Write'=1720kBps 'vmhba1 Average Number of Reads'=81 'vmhba1 Average Number of Writes'=168 'vmhba1 Latency Total Read'=2ms 'vmhba1 Latency Total Write'=0ms [Host_Storage_Adapter_Performance]

- • Example 2:
  - ○ box293_check_vmware.pl --check Host_Storage_Adapter_Performance --server 192.168.1.211 --host 192.168.1.210 --name vmhba1 --warning hba_rate:500 --critical hba_rate:1000
- • Output for Example 2:
  - ○ CRITICAL: [LSI 3ware 9750 (vmhba1) {Rate (Read:227 kBps / 13%)(Write:1,595 kBps / 87% (CRITICAL >= 1,000))} {Averaged Number of (Reads:13) (Writes:157)} {Total Latency (Read:2 ms)(Write:0 ms)}]|'vmhba1 Rate Read'=227kBps;500;1000 'vmhba1 Rate Write'=1595kBps;500;1000 'vmhba1 Average Number of Reads'=13 'vmhba1 Average Number of Writes'=157 'vmhba1 Latency Total Read'=2ms 'vmhba1 Latency Total Write'=0ms [Host_Storage_Adapter_Performance]

Host_Switch_Status
- • Description:
  - ○ Check the Status of a Hosts' vSwitch or Distributed Switch including all pNICs that are connected to the switch
  - ○ **NOTE:**
    - ▪ If the host is in Standby mode then the check will exit with an OK state
- • Required Arguments:
  - ○ --host (if connected via a vCenter server)
- • Optional Arguments:
  - ○ --modifier
  - ○ --mtu
  - ○ --name
  - ○ --nic_state
  - ○ --nic_speed
  - ○ --nic_duplex

- • Example 1:
  - ○ box293_check_vmware.pl --check Host_Switch_Status --server 192.168.1.211 --host 192.168.1.22
- • Output for Example 1:
  - ○ OK: [vSwitch0 {Local}, Ports {Total: 128} {Available: 127}, MTU: {1,500}, NICs: {No Physical NICs}], [dvSwitch {Distributed}, Ports {Total: 256} {Available: 248}, MTU: {1,500}, NICs: {Total: 2, Connected: 2} {vmnic0, Driver: e1000, 1,000 MB, Full Duplex} {vmnic1, Driver: e1000, 1,000 MB, Full Duplex}]

- • Example 2:

box293_check_vmware manual   *Page 71 / 85*

- ◦ box293_check_vmware.pl --check Host_Switch_Status --server 192.168.1.211 --host 192.168.1.22 --name dvSwitch --mtu 9000
- • Output for Example 2:
  - ◦ CRITICAL: [dvSwitch {Distributed}, Ports {Total: 256} {Available: 248}, MTU is 1500 but should be 9000, NICs: {Total: 2, Connected: 2} {vmnic0, Driver: e1000, 1,000 MB, Full Duplex} {vmnic1, Driver: e1000, 1,000 MB, Full Duplex}]

Host_Up_Down_State
- • Description:
  - ◦ This check is to be used as a host object check, helpful for hosts can go in Standby Mode and you don't want to be alerted about this as Standby Mode is normal behaviour (a standard ping check will report the host as down when in standby mode)
  - ◦ If you want it to report a DOWN state, use the --standby_exit_state argument, see example below
  - ◦ You can also trigger a "critical
  - ◦ When host is UP, uptime performance data is output
  - ◦ When host is UP, the ESX(i) host version is also displayed
  - ◦ **NOTE:**
    - ▪ This check only works when you are checking the host through vCenter
    - ▪ NOTE: there is no uptime or performance data on hosts less than 4.1
- • Required Arguments:
  - ◦ --host
- • Optional Arguments:
  - ◦ --perfdata_option post_check:disabled
  - ◦ --reporting_si Time:<Time>
  - ◦ --standby_exit_state
  - ◦ --warning and --critical uptime:<Time>

- • Example 1:
  - ◦ box293_check_vmware.pl --check Host_Up_Down_State --server 192.168.1.211 --host 192.168.1.22
- • Output for Example 1:
  - ◦ UP: Host is Up, Uptime: 13.7 days, Version: VMware ESXi 5.5.0 build-2143827|'Uptime'=13.7d [Host_Up_Down_State]

- • Example 2:
  - ◦ box293_check_vmware.pl --check Host_Up_Down_State --server 192.168.1.211 --host 192.168.1.22
- • Output for Example 2:
  - ◦ STANDBY: Host in StandBy mode, in Maintenance Mode

- • Example 3:
  - ◦ box293_check_vmware.pl --check Host_Up_Down_State --server 192.168.1.211 --host 192.168.1.22 --standby_exit_state down
- • Output for Example 3:
  - ◦ DOWN: Host in StandBy mode, in Maintenance Mode

- • Special Behaviour
  - ◦ The --warning or --critical thresholds for "uptime" work with this check, useful if you want to be alerted if the host has been up for less than x amount of time (maybe it was rebooted and you want

box293_check_vmware manual   *Page 72 / 85*

to know about that)
- ○ HOWEVER in relation to a host Up/Down check, there is no warning or critical status, it will be reported in Nagios as DOWN.
- ○ Example 4:
  - ▪ box293_check_vmware.pl --check Host_Up_Down_State --server 192.168.1.211 --host 192.168.1.22 --warning uptime:27
- ○ Output for Example 4:
  - ▪ WARNING: Host is Up, Uptime: 13.7 days (WARNING <= 27), Version: VMware ESXi 5.5.0 build-2143827|'Uptime'=13.7d;27

Host_vNIC_Status
- • Description:
  - ○ Check the Status of the Hosts' Virtual Network Interface Card(s) (vNIC) and report the role(s)
  - ○ All vNICs are returned by default however you can target specific vNIC(s)
  - ○ **NOTE:**
    - ▪ If the host is in Standby mode then the check will exit with an OK state
- • Required Arguments:
  - ○ --host (if connected via a vCenter server)
- • Optional Arguments:
  - ○ --modifier
  - ○ --mtu
  - ○ --name

- • Example 1:
  - ○ box293_check_vmware.pl --check Host_vNIC_Status --server 192.168.1.211 --host 192.168.1.210
- • Output for Example 1:
  - ○ OK: [{Management Network (vmk0 on Local vSwitch 'vSwitch0')} {MTU: 1,500} {Roles: Management}], [{VMkernel test (vmk1 on Local vSwitch 'vSwitch1')} {MTU: 1,500} {Roles: vMotion, Management, Fault Tolerance}]

- • Example 2:
  - ○ box293_check_vmware.pl --check Host_vNIC_Status --server 192.168.1.211 --host 192.168.1.43 --name vswif0
- • Output for Example 2:
  - ○ OK: [{vswif0 on Distributed vSwitch 'dvSwitch'} {MTU: 1,500} {Roles: Service Console}]

box293_check_vmware manual   *Page 73 / 85*

Tasks_Events
- Description:
    - Check the Tasks or Events history for specific text strings
    - For example, make sure a task occurs daily that contains the word **backup** and if it's not found trigger the desired Nagios state
    - You can target a Guest, Host, Cluster or Datacenter
    - **Note**: Tasks are only available when checking objects via vCenter
    - The matches are defined as follows:
        - **Events**
            - Match performed against the **fullFormattedMessage** property
        - **Tasks**
            - Match performed against the **descriptionId** or **entityName** properties
- Required Arguments:
    - --guest or --host or --cluster or --datacenter
        - If querying a host directly, it will use the --server argument
    - --match
        - These options are required:
            - operation
            - string
            - state
- Optional Arguments:
    - --filter
        - These options have the default values:
            - hours
                - 24
    - ---match
        - These options have the default values:
            - type
                - all
            - time
                - queuedTime

- Example 1 (**match** operation):
- Check the guest **XI Test** for anything with the word **backup** in the past **30** hours … if there is a match then all is OK otherwise trigger the state
    - box293_check_vmware.pl --server vcenter.box293.local --check Tasks_Events --guest 'XI Test' --filter hours:30 --match operation:match,string:'backup',state:CRITICAL
- Output for Example 1:
    - CRITICAL: The string 'backup' was NOT matched!

- Example 2 (**nomatch** operation):
- Check the host **esxi001.box293.local** for the string **performance has deteriorated** in the past **3** days … if there is NO match then all is OK otherwise trigger the state
    - box293_check_vmware.pl --server vcenter.box293.local --check Tasks_Events --host esxi001.box293.local --filter days:3 --match operation:nomatch,string:'performance has deteriorated',state:WARNING
- Output for Example 2:
    - WARNING: The string 'performance has deteriorated' WAS matched!: {Device eui.001b4d21103dd801 performance has deteriorated. I/O latency increased from average value of

box293_check_vmware manual   *Page 74 / 85*

707 microseconds to 23155 microseconds. @ 2016-03-22T23:26:50.329999Z}

vCenter_License_Status
- Description:
  - Reports the specified vCenters' license status along with the license key
  - If vCenter is in evaluation mode and has less than 24 hours remaining a CRITICAL state is returned otherwise a WARNING state is returned
  - When vCenter is queried using an account that has read only privileges, only a portion of the key is displayed
- Required Arguments:
  - --server
- Optional Arguments:
  - --hide_key

- Example 1:
  - box293_check_vmware.pl --check vCenter_License_Status --server 192.168.1.211
- Output for Example 1:
  - OK: Licensed {Version: vCenter Server 5 Standard} {Key: ABCDE-#####-#####-#####-UVWXY}

- Example 2:
  - box293_check_vmware.pl --check vCenter_License_Status --server 192.168.1.211 --hide_key
- Output for Example 2:
  - OK: Licensed {Version: vCenter Server 5 Standard}

vCenter_Name_Version
- Description:
  - Report the specified vCenter product Name and Version
- Required Arguments:
  - --server

- Example 1:
  - box293_check_vmware.pl --check vCenter_Name_Version --server 192.168.1.211
- Output for Example 1:
  - VMware vCenter Server 5.1.0 build-1123961

box293_check_vmware manual   *Page 75 / 85*

vSphere_Desktop_License
- Description:
  - Report the vSphere Desktop Host license usage
- Required Arguments:
  - --server
- Optional Arguments:
  - --perfdata_option post_check:disabled, License_Free:1, License_Total:1, License_Used:1
  - --warning and --critical license_free:<value>, license_used:<value>

- Example 1:
  - box293_check_vmware.pl --server vcenter.box293.local --check vSphere_Desktop_License --warning license_free:20 --critical license_free:10
- Output for Example 1:
  - OK: vSphere Desktop License {Free: 246} {Used: 4} {Total: 250}|'License Free'=246;20;10 'License Used'=4 'License Total'=250 [vSphere_Desktop_License]

box293_check_vmware manual   *Page 76 / 85*

Advanced Topics

--modifier Argument

As explained earlier, the --modifier argument allows manipulation of input and output values in Host and Guest checks.

This is a useless option when used alone at the command line, as why would you want to modify an input value when you could have typed it correctly in the first place!

The power behind the --modifier argument becomes apparent when you start using Nagios macros in your command definitions. Utilising Nagios macros allows you to create a single service definition that will work for hundreds of Nagios host objects. So instead of having 200 individual service definitions for a guest CPU_Usage check you can have just 1 that applies to 200 hosts. Need to change an argument or threshold? Change it once and all 200 hosts get that updated service definition.

How does the --modifier argument help? It allows you to marry up different standards. Here are different scenarios which demonstrate how it can be used.

Scenario 1

- In your vCenter environment you have 300 guests
  - They all follow a naming standard which is the hostname of the server in **UPPERCASE**
    - FILESERVER01
    - SQLSERVER11
- In your Nagios server, you are already doing some basic monitoring of these guests
  - The existing Nagios host object definitions are using a different naming standard the which is to use the hostname of the server in the host_name directive ($HOSTNAME$) in **lowercase**
    - fileserver01
    - sqlserver11
- In this scenario, the --modifier argument can be used to take the $HOSTNAME$ value received from Nagios and shift it to UPPERCASE
- For example:
  - box293_check_vmware.pl --server vcenter.box293.local --check Guest_CPU_Usage --guest $HOSTNAME$ --modifier request:shift:upper:shift

Scenario 2

- In your vCenter environment you have 300 guests
  - They all follow a naming standard which is the hostname of the server in **UPPERCASE**
    - FILESERVER01
    - SQLSERVER11
- In your Nagios server, you are already doing some basic monitoring of these guests
  - The existing Nagios host object definitions are using a different naming standard the which is to use the [hostname + fully qualified domain name] of the server in the host_name directive ($HOSTNAME$) in **lowercase**
    - fileserver01.box293.local
    - sqlserver11.box293.local
- In this scenario, the --modifier argument can be used to take the $HOSTNAME$ value received from

box293_check_vmware manual   *Page 77 / 85*

Nagios, REMOVE the .box293.local part AND convert it to UPPERCASE
- For example:
  - box293_check_vmware.pl --server vcenter.box293.local --check Guest_CPU_Usage --guest $HOSTNAME$ --modifier request:remove:upper:.box293.local

Scenario 3

- In your vCenter environment you have 300 guests
  - They all follow a naming standard which is the hostname of the server in **UPPERCASE**
    - FILESERVER01
    - SQLSERVER11
- In your Nagios server, you are already doing some basic monitoring of these guests
  - The existing Nagios host object definitions are using a different naming standard the which is to use the [hostname + fully qualified domain name] of the server in the host_name directive ($HOSTNAME$) in **lowercase**
  - You have multiple fully qualified domain names in your Nagios host object defintions
    - fileserver01.box293.local
    - sqlserver11.box562.local
- In this scenario, the --modifier argument can be used to take the $HOSTNAME$ value received from Nagios, REMOVE the .box293.local part OR the .box562.local part AND convert it to UPPERCASE
- For example:
  - box293_check_vmware.pl --server vcenter.box293.local --check Guest_CPU_Usage --guest $HOSTNAME$ --modifier request:remove:upper:.box293.local,request:remove:upper:.box562.local

Scenario 4

- In your vCenter environment you have 300 guests
  - They all follow a naming standard which is the [hostname + fully qualified domain name] of the server in **UPPERCASE**
    - FILESERVER01.BOX293.LOCAL
    - SQLSERVER11.BOX293.LOCAL
- In your Nagios server, you are already doing some basic monitoring of these guests
  - The existing Nagios host object definitions are using a different naming standard the which is to use the hostname of the server in the host_name directive ($HOSTNAME$) in **UPPERCASE**
    - FILESERVER01
    - SQLSERVER11
- In this scenario, the --modifier argument can be used to take the $HOSTNAME$ value received from Nagios and ADD the value .BOX293.LOCAL
- For example:
  - box293_check_vmware.pl --server vcenter.box293.local --check Guest_CPU_Usage --guest $HOSTNAME$ --modifier request:add:insensitive:.BOX293.LOCAL

Scenario 5

- In your vCenter environment you have 30 ESXi hosts
  - They all follow a naming standard which is the [hostname + fully qualified domain name] of the server in **lowercase**
    - esxi001.box293.local

box293_check_vmware manual   *Page 78 / 85*

- - esxi002.box293.local
- In your Nagios server, you are already doing some basic monitoring of these hosts
  - The existing Nagios host object definitions have the IP Address of the ESXi host in the address directive ($HOSTADDRESS$)
    - 192.168.20.88
    - 192.168.20.89
- In this scenario, the --modifier argument can be used to take the $HOSTADDRESS$ value received from Nagios, perform a reverse DNS lookup on the address AND convert it all to lowercase
- For example:
  - box293_check_vmware.pl --server vcenter.box293.local --check Host_CPU_Usage --host $HOSTADDRESS$ --modifier request:reverseip:lower

Scenario 6

- In your vCenter environment you have 300 guests
  - They all follow a naming standard which is the hostname of the server in **UPPERCASE**
    - FILESERVER01
    - SQLSERVER11
- In your Nagios server, you are already doing some basic monitoring of these guests
  - The existing Nagios host object definitions have the IP Address of the guests in the address directive ($HOSTADDRESS$)
    - 192.168.30.132
    - 192.168.30.133
- In this scenario, the --modifier argument can be used to take the $HOSTADDRESS$ value received from Nagios, perform a reverse DNS lookup on the address, REMOVE the .box293.local part of the address AND convert it to UPPERCASE
- For example:
  - box293_check_vmware.pl --server vcenter.box293.local --check Guest_CPU_Usage --guest $HOSTNAME$ --modifier request:reverseip_remove:upper:.box293.local

Coming Soon: examples here about the response option, which will be used by the Guest_Host option!

box293_check_vmware manual   *Page 79 / 85*

Running box293_check_vmware using the 'nice' command

One thing that has really become apparent with this plugin is that it really loves to use as much CPU as it can get it's hands on. This starts to cause problems as you ramp up the amount of monitoring you are performing and hence how many instances of box293_check_vmware can be run concurrently.

I spent some time investigating how this could be limited. After much reading on the internet I discovered a simple program called 'nice'. Nice allows you to execute other programs with specific process scheduling levels. From the help:

nice --help
Usage: nice [OPTION] [COMMAND [ARG]...]
Run COMMAND with an adjusted niceness, which affects process scheduling.

I changed one of my test environments so that all instances of box293_check_vmware were executed using nice and it was clear that the vMA became more stable. A lot of CPU is still used, but the problems I had with the vMA becoming unresponsive went away.

Nice is included on the vMA appliance which is great because it means no extra steps are required to make it work.

Henceforth this document has been updated to include the nice command as part of the Nagios configuration examples and is now the recommended way to implement this plugin.

How do you use nice?

Here's a command before using nice:
- ~/box293_check_vmware.pl --server vcenter.box293.local --check Host_CPU_Usage --host esxi001.box293.local

Here's nice being used to execute that command:
- nice -n19 ~/box293_check_vmware.pl --server vcenter.box293.local --check Host_CPU_Usage --host esxi001.box293.local

It's that simple. As you can see, all that was added was **'nice -n19 '** to the beginning of the command.

box293_check_vmware manual   *Page 80 / 85*

Known Problems / Issues / Troubleshooting

Nagios XI does not display full command output

It has been observed that in Nagios XI, the GUI only displays the first 256 characters of the command output. Example:

However this is only a limitation in the GUI. You can see the entire information when **clicking** on *See this service in Nagios Core* on the **Advanced** tab. Solution is on the next page.

Example:

You will also notice that this same behaviour occurs for the performance data, however this is only a GUI limitation, all performance data is received and processed accordingly (and viewable in Nagios Core).

box293_check_vmware manual   *Page 81 / 85*

Solution:
The Nagios XI tables in the mysql database need to be increased. At the CLI of the Nagios XI host execute the following four commands:

VERY IMPORTANT NOTE:
- These four commands are one entire command. When copying and pasting, carriage returns may be added and cause them to not execute correctly.
- Simply copy and paste them into a text editor and remove the carriage returns so you have one long command (for each of the four commands). Then you can paste each command into an SSH session.

echo "use nagios;alter table nagios_servicestatus modify output varchar(65535) not null;alter table nagios_servicestatus modify long_output varchar(65535) not null;alter table nagios_servicestatus modify perfdata varchar(65535) not null;" | mysql -pnagiosxi

echo "use nagios;alter table nagios_hoststatus modify output varchar(65535) not null;alter table nagios_hoststatus modify long_output varchar(65535) not null;alter table nagios_hoststatus modify perfdata varchar(65535) not null;" | mysql -pnagiosxi

echo "use nagios;alter table nagios_servicechecks modify output varchar(65535) not null;alter table nagios_servicechecks modify long_output varchar(65535) not null;alter table nagios_servicechecks modify perfdata varchar(65535) not null;" | mysql -pnagiosxi

echo "use nagios;alter table nagios_hostchecks modify output varchar(65535) not null;alter table nagios_hostchecks modify long_output varchar(65535) not null;alter table nagios_hostchecks modify perfdata varchar(65535) not null;" | mysql -pnagiosxi

Once these commands have been executed, the next time the check is run the full output of the check will be shown in the Nagios XI GUI.

Plugin fails to find objects such as Datastores

It has been observed that in some instances the plugin cannot locate datastores. For example:

Command:
- ./box293_check_vmware.pl --server vcenter.box293.local --check Datastore_Usage --name VOL_01

Output:
- UNKNOWN: Datastore 'VOL_01' not found

The problem is with the permissions that have been applied in the vSphere Client. For some reason assigning the permissions at the top level of "Inventory > Hosts and Clusters" is not enough. You need to also assign the permissions at the top level of "Inventory > Datastores and Datastore Clusters".

Test if permissions are correct

The best way to test if the permissions have been correctly assigned in vSphere Client is to log into the vSphere Client using the read only account you've assigned the permissions to. If you don't see an object in the inventory while logged in using this read only account then the plugin will not work. To resolve, find the location in the inventory where the permissions are not being applied and correctly apply them.

box293_check_vmware manual   *Page 82 / 85*

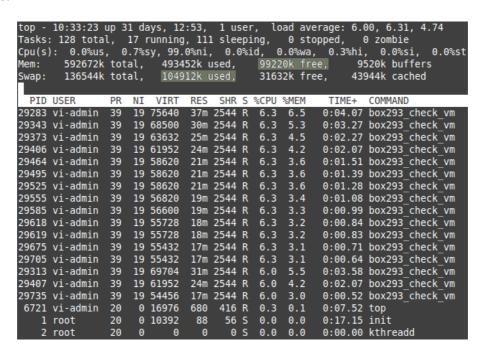## *UNKNOWN - check_by_ssh: Remote command 'xxxxxx' returned status 255*

Whenever you see an issue like this reported from your Nagios host, establish an SSH session to your vMA appliance as vi-admin and run the check directly on the vMA server. This will give you a more detailed explanation of what the cause of the error is.

Timeouts

If Nagios is reporting a lot of *'Service Check Timed Out'* then a likely reason is that the vMA appliance is not sized correctly.

Using the command 'top' when logged onto the vMA you can get an idea of the resources being used.

Here is an example:

```
top - 10:33:23 up 31 days, 12:53,  1 user,  load average: 6.00, 6.31, 4.74
Tasks: 128 total,  17 running, 111 sleeping,   0 stopped,   0 zombie
Cpu(s):  0.0%us,  0.7%sy, 99.0%ni,  0.0%id,  0.0%wa,  0.3%hi,  0.0%si,  0.0%st
Mem:    592672k total,   493452k used,    99220k free,     9520k buffers
Swap:   136544k total,   104912k used,    31632k free,    43944k cached

  PID USER      PR  NI  VIRT  RES  SHR S %CPU %MEM   TIME+  COMMAND
29283 vi-admin  39  19 75640  37m 2544 R  6.3  6.5  0:04.07 box293_check_vm
29343 vi-admin  39  19 68500  30m 2544 R  6.3  5.3  0:03.27 box293_check_vm
29373 vi-admin  39  19 63632  25m 2544 R  6.3  4.5  0:02.27 box293_check_vm
29406 vi-admin  39  19 61952  24m 2544 R  6.3  4.2  0:02.07 box293_check_vm
29464 vi-admin  39  19 58620  21m 2544 R  6.3  3.6  0:01.51 box293_check_vm
29495 vi-admin  39  19 58620  21m 2544 R  6.3  3.6  0:01.39 box293_check_vm
29525 vi-admin  39  19 58620  21m 2544 R  6.3  3.6  0:01.28 box293_check_vm
29555 vi-admin  39  19 56820  19m 2544 R  6.3  3.4  0:01.08 box293_check_vm
29585 vi-admin  39  19 56600  19m 2544 R  6.3  3.3  0:00.99 box293_check_vm
29618 vi-admin  39  19 55728  18m 2544 R  6.3  3.2  0:00.84 box293_check_vm
29619 vi-admin  39  19 55728  18m 2544 R  6.3  3.2  0:00.83 box293_check_vm
29675 vi-admin  39  19 55432  17m 2544 R  6.3  3.1  0:00.71 box293_check_vm
29705 vi-admin  39  19 55432  17m 2544 R  6.3  3.1  0:00.64 box293_check_vm
29313 vi-admin  39  19 69704  31m 2544 R  6.0  5.5  0:03.58 box293_check_vm
29407 vi-admin  39  19 61952  24m 2544 R  6.0  4.2  0:02.07 box293_check_vm
29735 vi-admin  39  19 54456  17m 2544 R  6.0  3.0  0:00.52 box293_check_vm
 6721 vi-admin  20   0 16976  680  416 R  0.3  0.1  0:07.52 top
    1 root      20   0 10392   88   56 S  0.0  0.0  0:17.15 init
    2 root      20   0     0    0    0 S  0.0  0.0  0:00.00 kthreadd
```

I have highlighted two numbers in this screenshot, Memory Free and Swap Used.

So this VM has the default 600MB assigned to it and currently has about 96MB Free. This isn't a lot really and needs to be increased, I suggest giving the vMA 2GB Memory.

The other number highlighted is Swap. Here you can see that about 102MB of Memory has been swapped to disk. This slows down the computer considerably as disks are much, much slower than Memory. Hence increasing the size of the Memory in the vMA will stop it from swapping.

box293_check_vmware manual   *Page 83 / 85*

Sponsors

This section is here to acknowledge third parties that have helped make this plugin what it is.

Nagios Enterprises, LLC

Website: www.nagios.com

UbiStor, Inc.

Website: www.ubistor.com

Subscribe For Update Notifications

I have a mailing list that I will send an email to when I update this plugin. This way you can find out as soon as a new version of this plugin is available.

Subscribe

- Send an email to updates+subscribe@box293.com
- You will receive an email with a link you need to follow to create a subscription request
- Click the link to open it in a web browser
- You will need to type your email address and click submit
- You will receive another email with a link you need to follow to complete your subscription
- You will now be subscribed!
- Check your spam folder if the emails are not received

Un-subscribe

- Send an email to updates+unsubscribe@box293.com
- You will receive an email informing you that you are no longer subscribed to this group

box293_check_vmware manual   *Page 84 / 85*

My Nagios Projects

I have created a handful of projects for Nagios. Some of them are very useful indeed so I suggest you check them out, you can view them all here via this link:
- http://exchange.nagios.org/directory/Owner/Box293/1

My Nagios Website

I have a website that has Nagios and Nagios related information. I do a lot of test and development, so I use this website to record procedures like installing core from source on specific operating systems. It saves me re-inventing the wheel.
- http://sites.box293.com/nagios

Nagios XI Discount Offer

Get a 10% discount on your Nagios XI Unlimited Nodes license when purchased through this link:
- http://www.nagios.com/nagiosxi10?ref=box293

Support

Please contact me for support via the email address:
- plugins@box293.com

I always announce product updates with a Tweet, my Twitter name is @Box293
- Please do NOT send support requests through Twitter

Please keep in mind that I do not charge for development and use of this plugin so support is provided on a best effort basis. Sometimes I will get back to you in a day, sometimes in a week!

Feedback and Contributions

Any feedback or contributions are welcome as it improves the plugin for others.

About Me

My name is Troy and I am an IT Consultant and Nagios Developer based in Deniliquin, Australia. I can be contacted a few ways:
- Through the Nagios Support Forums as the user Box293
- Via my email address plugins@box293.com
- Twitter: @Box293

box293_check_vmware manual   *Page 85 / 85*