ESXi
Tuning Guide
Eli-v6.2.20

Bridgeworks
Unit 1, Aero Centre, Ampress Lane,
Ampress Park, Lymington,
Hampshire SO41 8QF
Tel: +44 (0) 1590 615 444
Email: support@4bridgeworks.com

Table of Contents

1 Introduction **3**

1.1 When to use this guide . . . 3

1.2 When not to use this guide . . . 3

1.3 Further reading . . . 4

2 Summary **5**

2.1 CPU Core Allocation: . . . 5

2.2 Core Affinity: . . . 5

2.3 Memory Allocation: . . . 5

2.4 Memory (NUMA - Non-Uniform Memory Access) Alignment: . . . 6

3 Requirements **7**

3.1 ESXi Tools . . . 7

3.1.1 ESXi built-in Web Client . . . 7

3.1.1.1 Accessing ESXi built-in Web Client . . . 7

3.1.2 ESXi vCenter Server Flash based Web Client . . . 10

3.1.2.1 Accessing ESXi vCenter Server Web Client . . . 10

3.1.3 vSphere client program (windows machine only) . . . 13

3.1.3.1 Accessing vSphere client program . . . 13

4 CPU Allocation **16**

4.1 Adjusting Core Allocation . . . 16

4.1.1 vSphere . . . 16

4.1.2 ESXi Built-in Web Client . . . 19

4.1.3 vCenter Server Flash Web Client . . . 20

5 Core Affinity **24**

5.1 Assessing Viability . . . 24

5.1.1 vSphere . . . 24

5.1.2 ESXi built-in Web Client . . . 25

5.1.3 vCenter Server Flash Web Client 27
5.2 Applying Affinity 29
5.2.1 vSphere 29
5.2.2 ESXi built-in Web Client 32
5.2.3 vCenter Server Flash Web Client 34

6 Memory Allocation **38**
6.1 Adjusting Allocation 39
6.1.1 vSphere 39
6.1.2 ESXi Built-in Web Client 40
6.1.3 vCenter Server Flash Web Client 41

7 Memory Alignment **42**
7.1 Setting Alignment 42
7.1.1 vSphere 42

1 Introduction

This document details the prerequisites for tweaking a deployed virtual Bridgeworks Node to improve resource use and maintain peak performance.

The suggested modifications included in this guide are given as a best practice approach to reduce the chance of having a sub-optimal virtual installation.

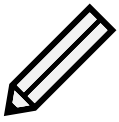
Most of the optimisation is targeted at ensuring that the virtual Bridgeworks Node can keep its CPU cores and RAM access aligned. Without alignment the RAM speeds can suffer, which can affect acceleration performance.

1.1 When to use this guide

Generally ESXi tuning is not required. The system does a good job of moving resources around to keep everything aligned and provided with enough memory and CPU time.

Key symptoms that could be improved by tuning the ESXi guest settings are:

- Reduced performance, these can be:
  - Reduced throughput. If it is proven that the Bridgeworks Node is not reaching the full speed of the network connection it has been given.
  - Inconsistent throughput. If the graphing in the *Transfer Statistics* page show that throughput is not consistent during a full bandwidth data a transfer in your typical network environment, and no other environmental variables are causing this.
  - Very high CPU usage. If the System info page is showing more than 80% CPU utilisation during a full bandwidth data transfer in your typical network environment.
  - Very high memory usage. If the System info page is showing more than 80% memory utilisation at any time.
- Conserving ESXi resources when performance is not being reduced*. Such as:
  - Very low CPU usage.
    If the System info page is showing less than 25% CPU utilisation during a full bandwidth data transfer in your typical network environment.
  - Very low memory usage.
    If the System info page is showing less than 25% memory utilisation during a full bandwidth data transfer in your typical network environment.



Note: *Reducing resources to the Bridgeworks Node should always take into account any resource headroom required to sustain performance if network quality deteriorates. See Chapter 4: CPU Allocation and Chapter 6: Memory Allocation sections for minimum supported resource settings.

1.2 When not to use this guide

Other factors can result in lower utilisation of network connection between Bridgework Nodes.

Possible other factors include:

- Onload/Offload storage I/O

  If the systems on either side of the Bridgeworks Nodes cannot provide or receive the data quickly enough a bottleneck will occur and the Nodes will throttle the connection appropriately.

- Shared network resources

  If either Bridgeworks Node shares the network connection with other systems then competition for network time can reduce throughput depending on the demand for the network bandwidth.

- Product bandwidth limits Each series of Bridgeworks node has a different network bandwidth capacity.

  - Series 100 = 125MB/s.
  - Series 200 = 250MB/s.
  - Series 400 = 1GB up to 10GBe dependant on hardware specification.

1.3 Further reading

For more detailed information regarding ESXi tuning, see VMwares own ”Performance Best Practices” documentation for your relevant version of ESXi.

- ESXi 6.5 Best Practices (2017)
- ESXi 6.7 Best Practices (2018)
- ESXi 7.0 Best Practices (2020)

2 Summary

2.1 CPU Core Allocation:

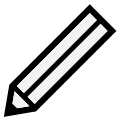
It is beneficial to keep core allocation conservative. Increasing the number of assigned cores will improve performance only when the existing setup is consistently experiencing CPU usage above 75%. The OVAs for Bridgeworks Nodes have a preset core allocation that is set to accommodate the throughput intended for that model. Scenarios that could result in needing more CPU power are; running encryption on top of full network speeds, or having a very large packet loss environment. Conversely, if the Node never sees high core usage, then reducing the allocation may help ensure that the host never places the Bridgeworks Node into a sub-optimal state.

In ESXi the core allocation is dictated by vCPUs. These cores are also presented in a settable number of sockets. This setting is for operating system compatibility only and does not provide a performance increase. The Bridgeworks Node should be given all vCPUs as individual sockets. Running the system with a 'multiple cores per socket' setup can have adverse effects on its performance.

2.2 Core Affinity:

Core affinity can have an immediate effect on acceleration performance, this is evident when the core affinity forces execution to be spread across the physical sockets of the host. In ESXi, core affinities can be set on a per guest basis. The only requirement is that the affinity gives at least the number of vCPUs given to the guest. The recommended practice is to give an affinity setting that ensures all execution of the Bridgeworks Node will happen on the same physical socket, this ensures that all memory access will be using the same caching and RAM modules.

In addition, when using a processor feature such as Intel Hyperthreading the underlying hardware will present 2 CPUs per physical processor core. ESXi will handle these as *logical cores*, which results in the ESXi scheduler load balancing the 2 logical cores across the physical core evenly*. This load balancing means that the effective performance of 1 logical core can be less than the actual performance of the physical core it is running on.



Note: *When ESXi CPU scheduling results in only 1 of the 2 available logical cores on a given physical core being utilised, the guest running on that logical core is given the full performance of the physical core. Logical core performance is reduced when both logical cores on the same physical core are being utilised by guests, at which time the performance of the physical core is split equally between the two logical cores presented by ESXi.

2.3 Memory Allocation:

Just like vCPUs, assigning the minimum necessary memory will reduce any chance of a sub-optimal setup. A smaller RAM footprint makes it a lot easier for the ESXi host to ensure that the memory is aligned to the same socket that the vCPUs reside on.

2.4 Memory (NUMA - Non-Uniform Memory Access) Alignment:

ESXi can split memory into NUMA pools, this is due to separate RAM access across separate physical sockets on a motherboard. In a worst case scenario a virtual guest could end up with all of its CPU resource being scheduled on physical CPU cores that are not directly attached to the RAM that the guest is using. Bandwidth testing has shown that memory performance can degrade significantly, slowdowns of over 20% were frequently seen. In addition, running a badly aligned Bridgeworks Node will result in a decrease to the total throughput attained on a like-for-like network environment. Memory alignment through NUMA affinity is similar to running CPU affinity. Manipulating CPU affinity is preferred as it can be adjusted on a live system.

3 Requirements

3.1 ESXi Tools

There are several methods for configuring virtual machines in an ESXi system:

- Basic settings via a built-in web client.
- More advanced settings using the ESXi vCenter Server system and its Flash based web client.
- Classic control using the now deprecated vSphere client on a windows machine.

The following sections list what each method can and cannot achieve with regards to the tuning solutions included in this guide.

3.1.1 ESXi built-in Web Client

ESXi installation software deploys a web based control GUI automatically.

This method can control:

- Core allocation and socket/core distribution.
- RAM amount and priority/reservation values.
- Core affinity.

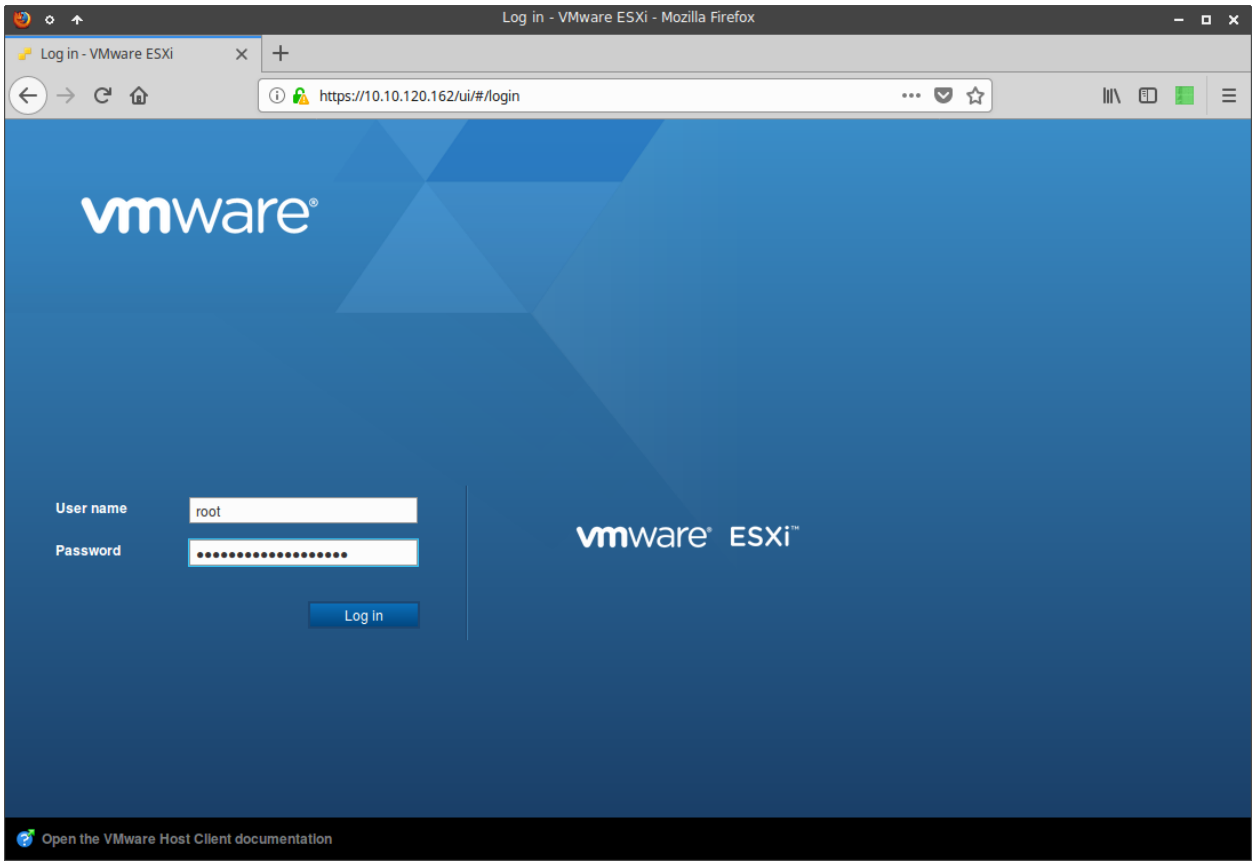
This method cannot control:

- RAM NUMA affinity.

3.1.1.1 Accessing ESXi built-in Web Client

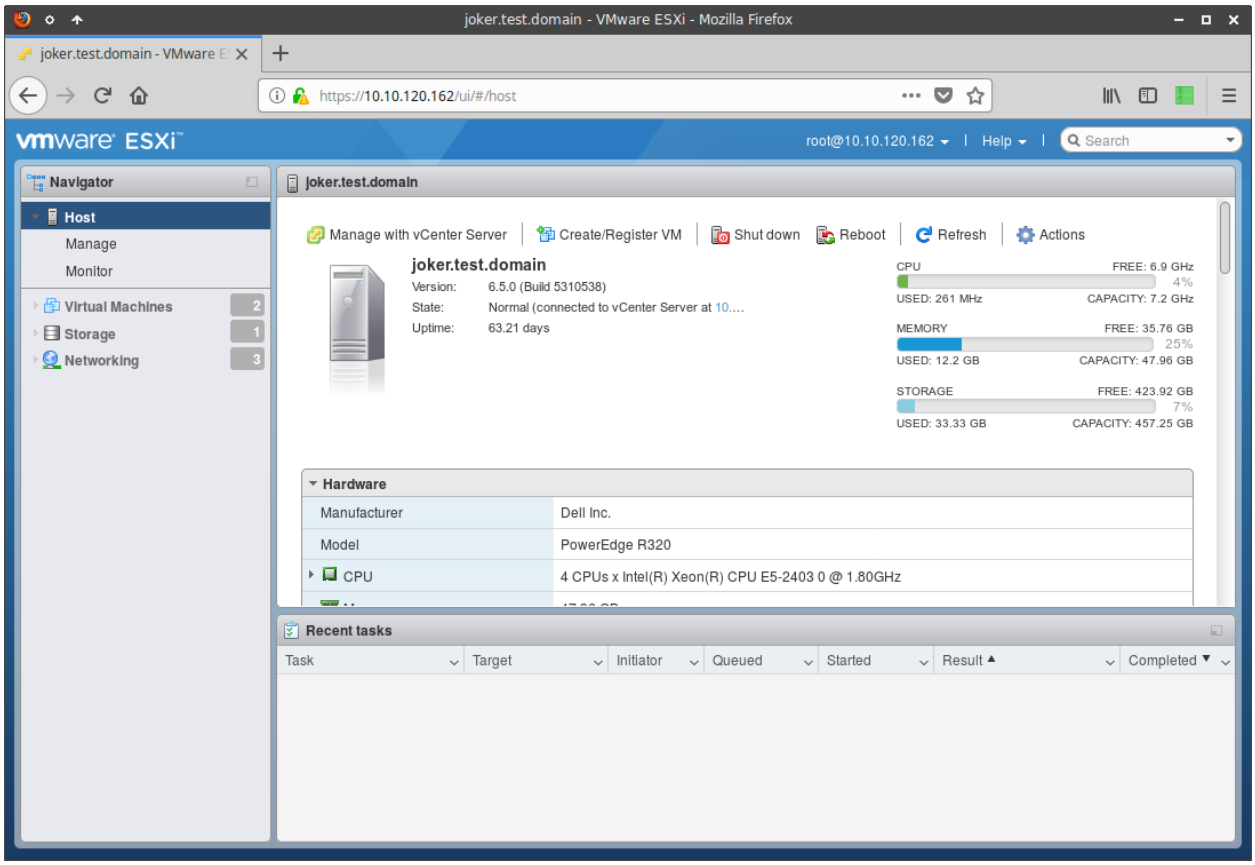
To connect to the built-in web client, open your browser of choice and navigate to the IP address or website name of the ESXi host machine.

This should present a login screen.

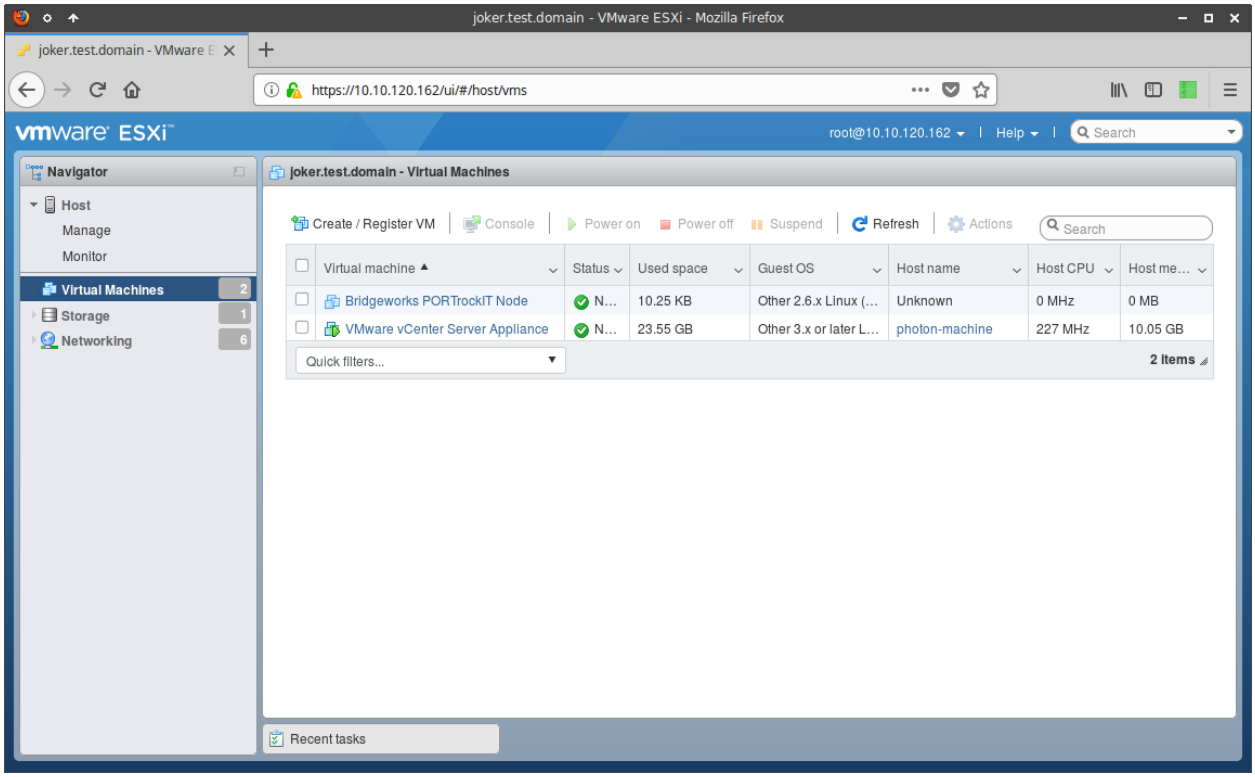


Enter the correct user and password and click ”Log in”.

At this stage the web client will show the host machine information page.



In the left hand pane there is a list of options. Click the ”Virtual Machines” entry. For tuning, only the Bridgeworks Node will be used.



3.1.2 ESXi vCenter Server Flash based Web Client

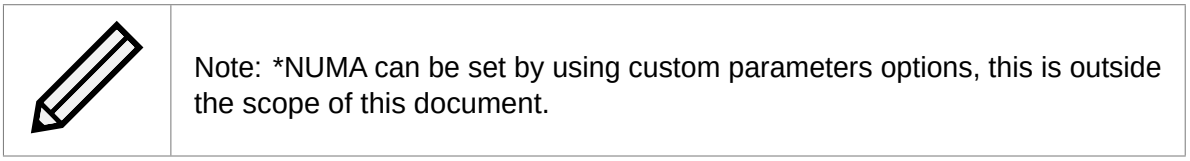
Optionally, a ”vCenter Server” system can be deployed and connected to your ESXi host. This allows far greater control than the basic ESXi web client.

This method can control:

- Core allocation and socket/core distribution.
- RAM amount and priority/reservation values.
- Core affinity.

This method cannot control:

- RAM NUMA affinity.*

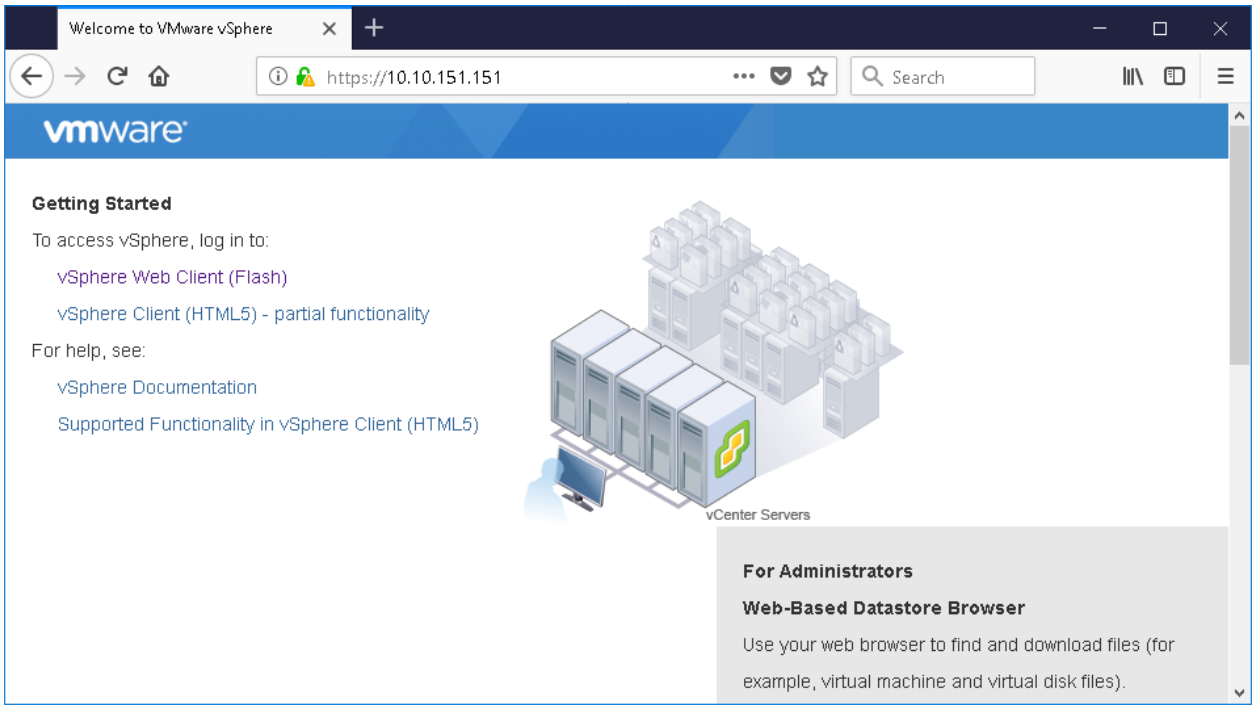


3.1.2.1 Accessing ESXi vCenter Server Web Client

To connect to the Flash based web client, open your browser of choice and navigate to the IP address or website name of the vCenter Server.

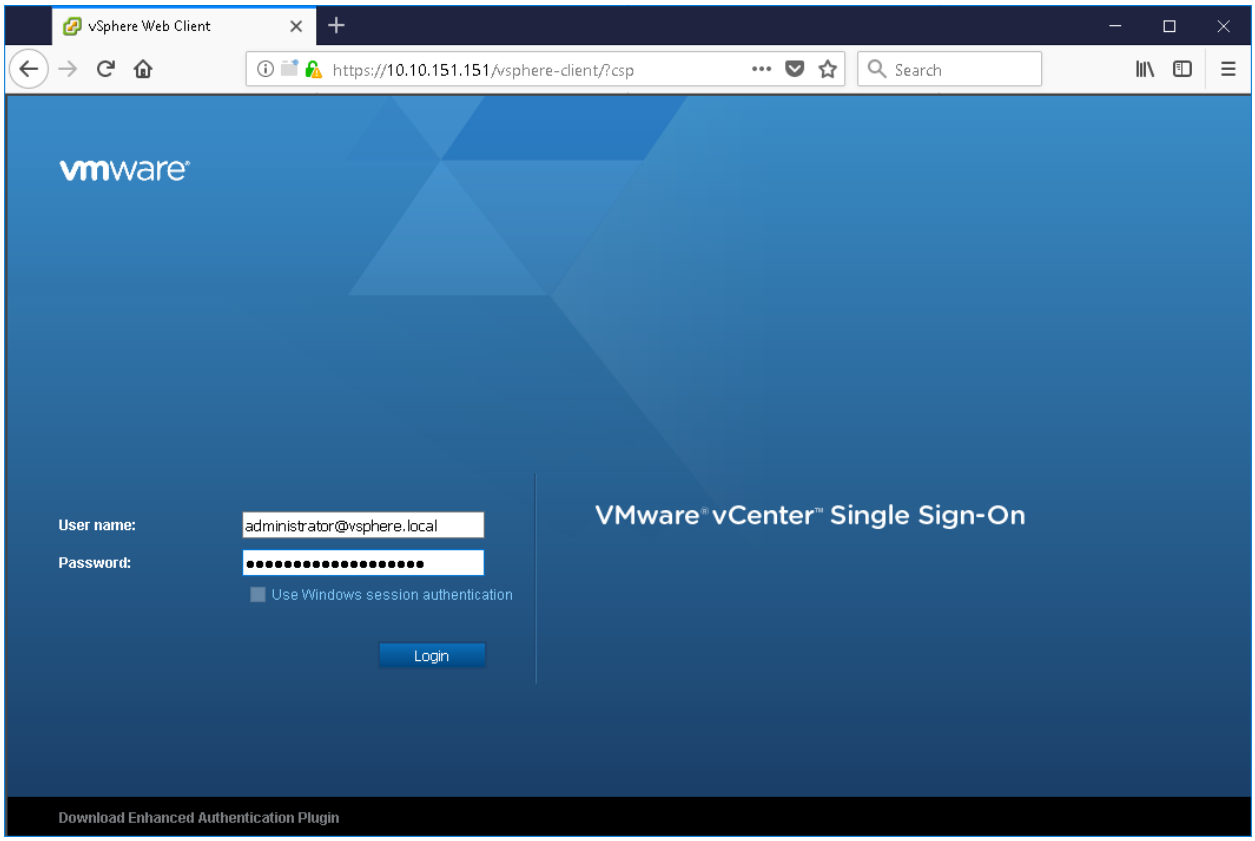
If you do not know the address for the vCenter Server you may be able to get it from the Host info page on the ESXi built-in web client. See above for instructions on accessing it. On the Host info page presented at login, look for ”State” in the main pane, to the right there should be a status, along with notification of ”connected to vCenter Server” and an IP address or website URL to the vCenter Server.

Upon accessing the vCenter Server there should be a page listing the access.

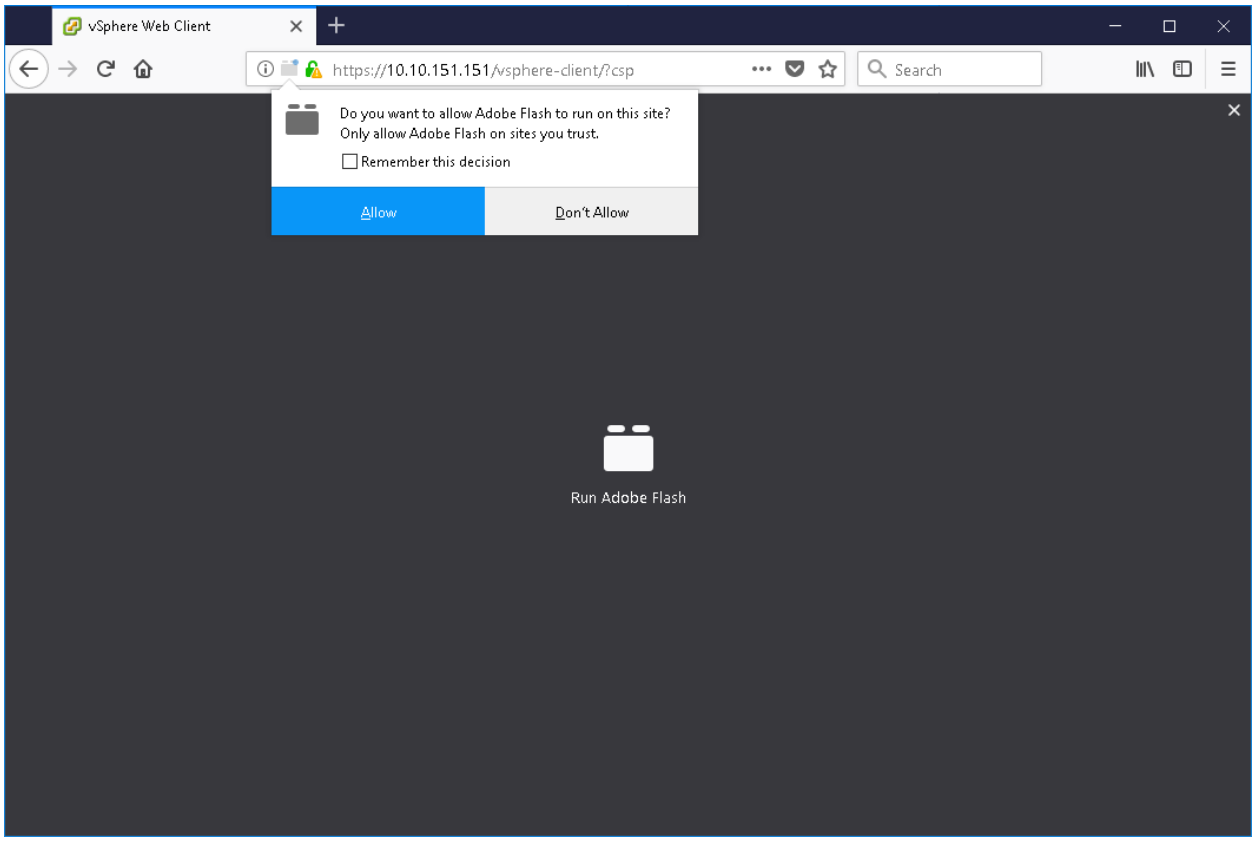


Select the ”vSphere Web Client (Flash)” link.

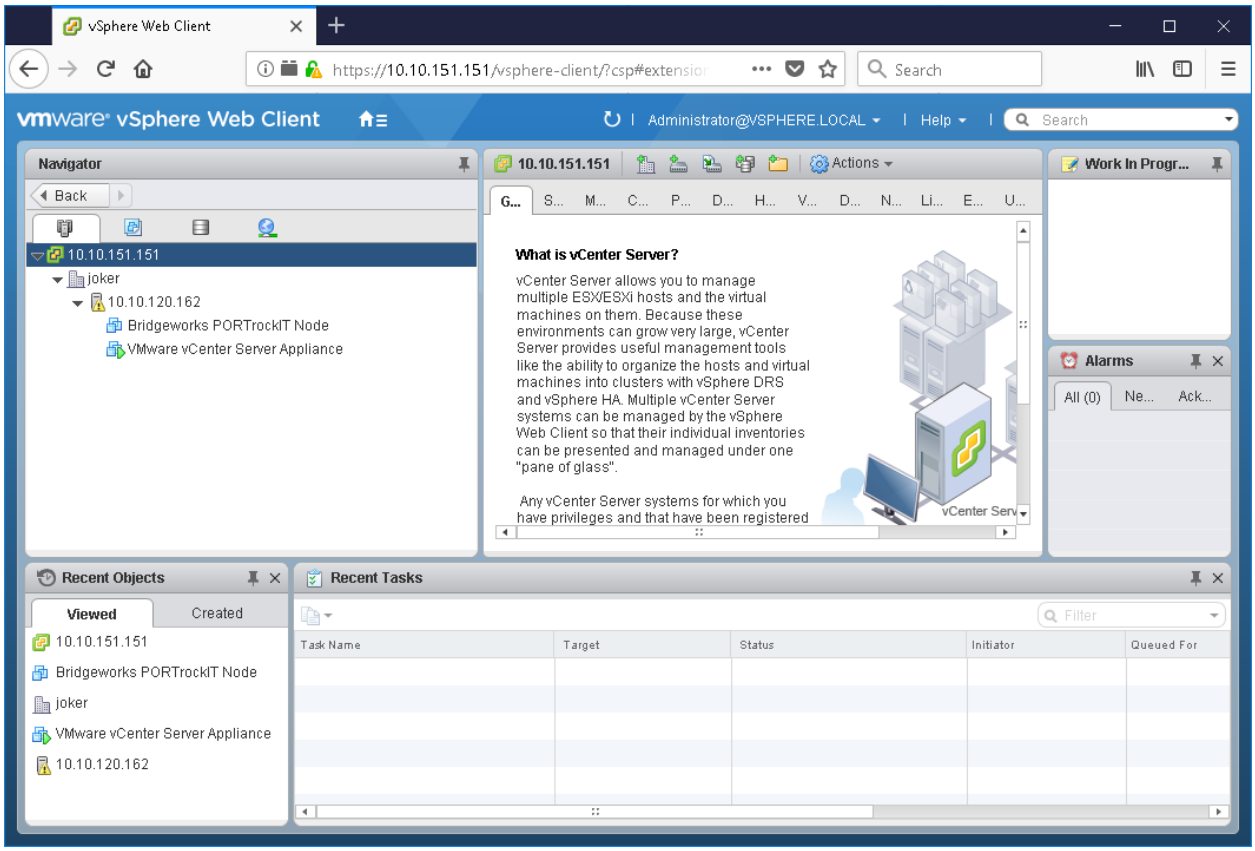
This should present a login screen. Enter the correct user and password and click ”Log in”.



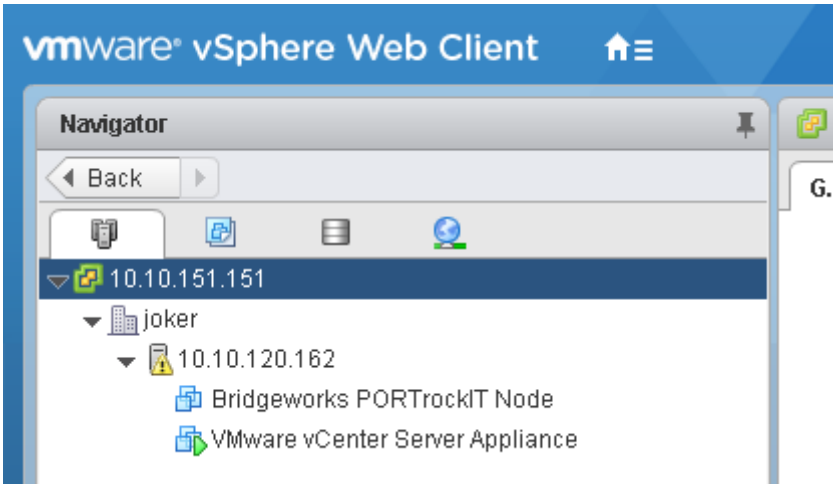
At this stage the web client will show the last page viewed by this account. If not, be sure that the browser supports Flash, and that it is enabled to run.



In the left hand pane there should be an entry for the target ESXi host. And expanding the ESXi host should reveal the target Bridgeworks Node.



For tuning, only the target Bridgeworks Node selection will be used.



3.1.3 vSphere client program (windows machine only)

As of ESXi 6.5, the vSphere Client is still supported by VMWare Inc. The vSphere Client does not support any new features past ESXi 5.1.

This method can control all parameters referred to by this document:

- Core allocation and socket/core distribution.
- RAM amount and priority/reservation values.
- Core affinity.
- RAM NUMA affinity.

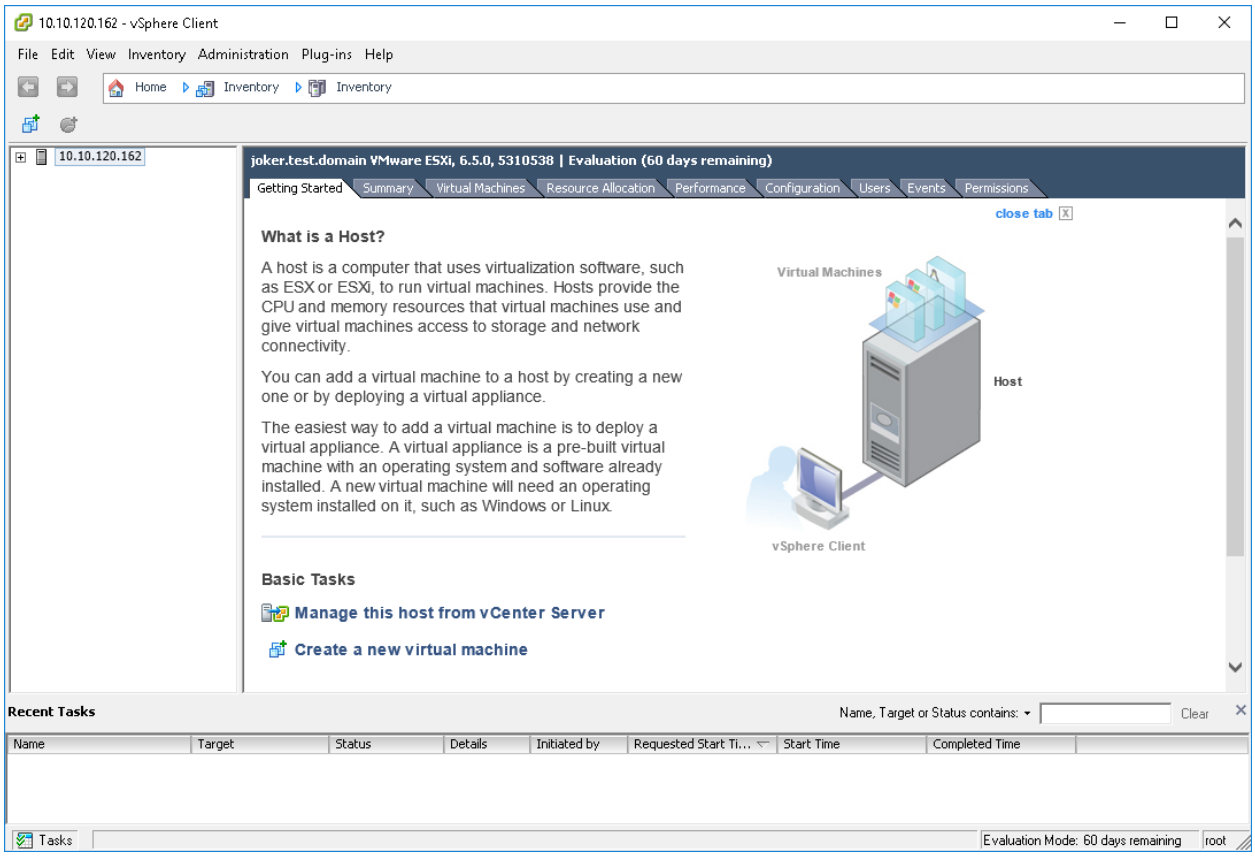
3.1.3.1 Accessing vSphere client program

On a windows machine, ensure that the ”VMWare vSphere Client” program is installed.

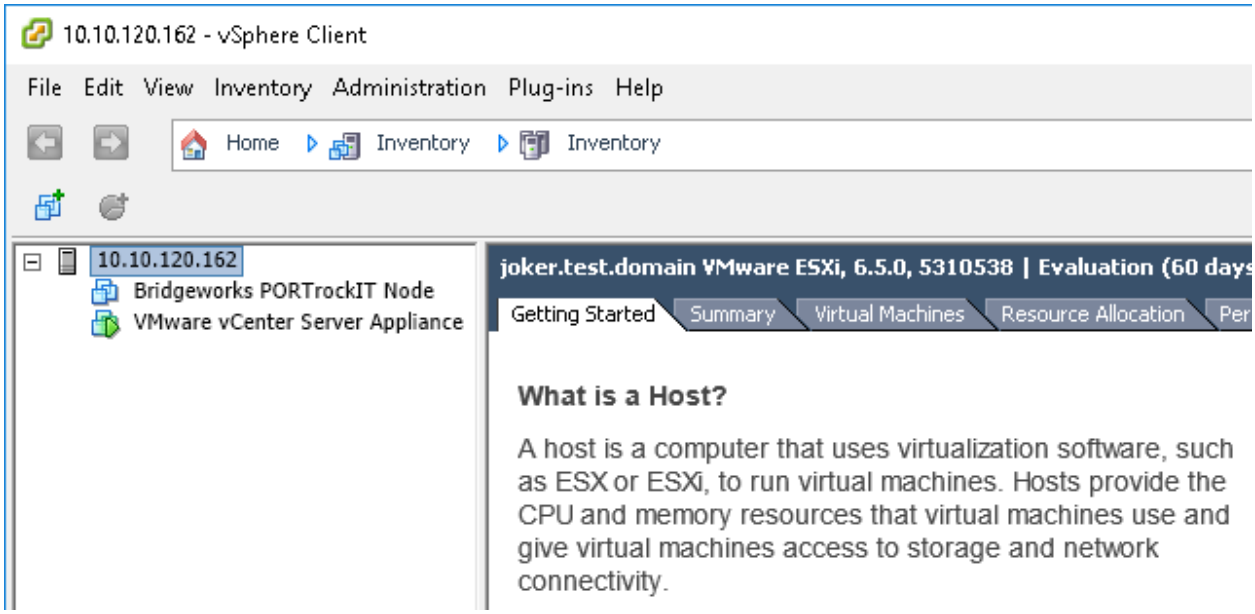
Run the program. A Login form will display.

Enter the IP address or server name of the target ESXi host. If applicable, fill out the username and password and click login.

At this stage the client will show the Getting Started page of the ESXi host.



In the left hand pane expand the entry showing the host name or IP.



For tuning, only the target Bridgeworks Node selection will be used. In this example the Node is name ”Bridgeworks PORTrockIT Node”.

4 CPU Allocation

ESXi allows provisioning of a huge number of vCPUs to a virtual system. Across multiple virtual guests, this core allocation can exceed the physical CPU count of the host.

In addition, vCPUs can be presented to a virtual guest's operating system as either multiple sockets, or as a multiple-cores-per-socket system. Generally the user should always choose multiple sockets, with multiple cores being a compatibility setting for operating systems that do not support multiple sockets.

Initial CPU cores for each system are:

- 100 Series = 2
- 200 Series = 3
- 400 Series = 4

These values can be changed to suit the individual applications. The number of cores can be reduced if core utilisation is consistently low, though less than 2 cores is not supported. Likewise CPU allocation can be increased if CPU utilisation is consistently high, this scenario could be due to overheads from excessively detrimental network conditions or enabling features that require CPU time, such as IPSec.

In most cases no more than 8 cores should be needed, with core clock speed being more important for performance than further increasing core allocation.

Core allocation should always try to be no more than that of the underlying hardware. Once vCPUs start sharing the resources of a hardware core, the scheduling for CPU resources can result in high latency and reduced performance. If an ESXi host needs to have oversaturation of the CPU cores, then consider using CPU affinity to ensure that the Bridgeworks Node has cores reserved. This ensures that network acceleration is not affected by other virtual guests competing for CPU time.

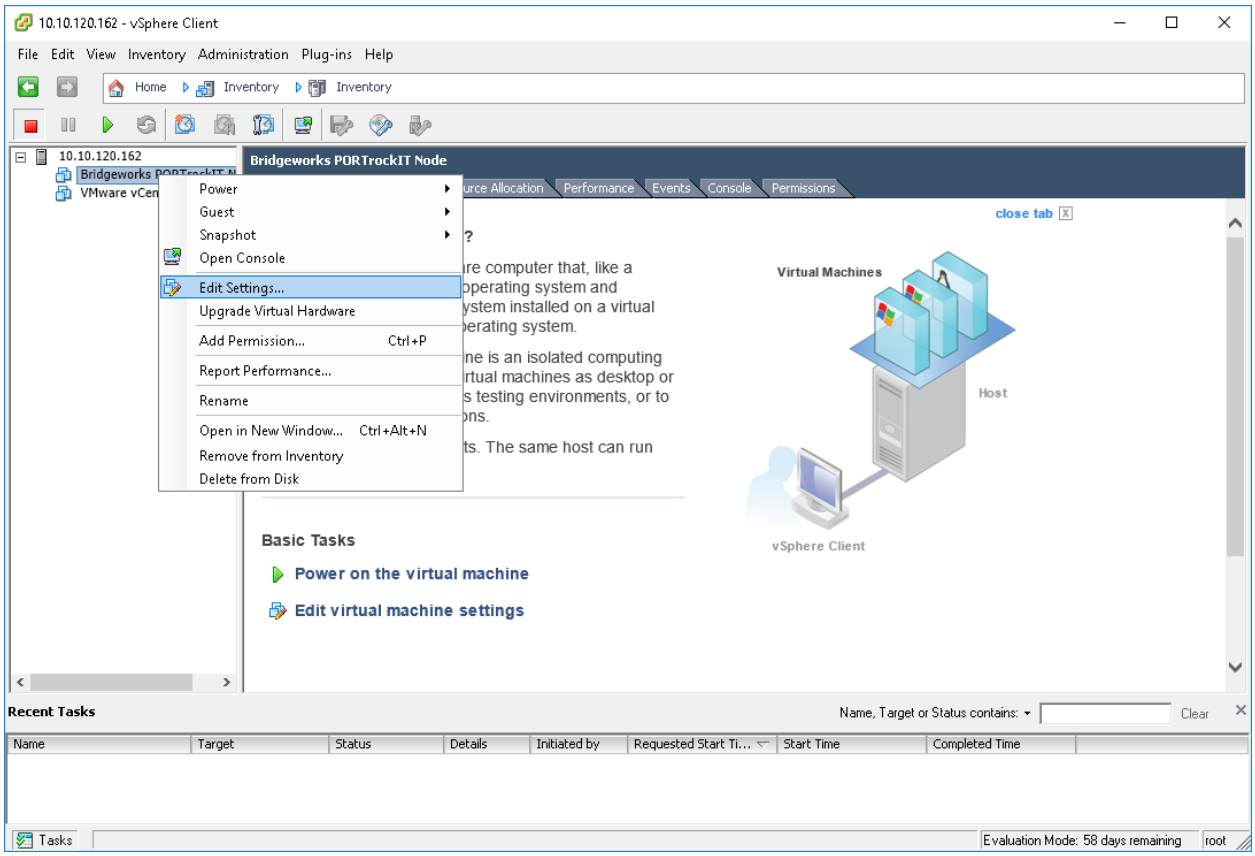
An ideal setup would allow the Bridgeworks Node to run entirely on a single physical socket, so when possible, the core allocation should be no more than the core count of one of the hosts CPU sockets.

When assessing core allocation requirements it is important to include the performance deficit from running on logical cores. This is not to be confused with the vCPUs terminology used in ESXi. On the Intel platform this logical core feature is known as "Intel Hyper-Threading Technology", this can be enabled/disabled in BIOS settings, and will double the detected cores of the underlying hardware. A logical core will not perform as well as a physical core; e.g. a dual core guest running on two logical (hyper-threaded) cores will not perform as well as a dual core guest running on two physical cores.

4.1 Adjusting Core Allocation

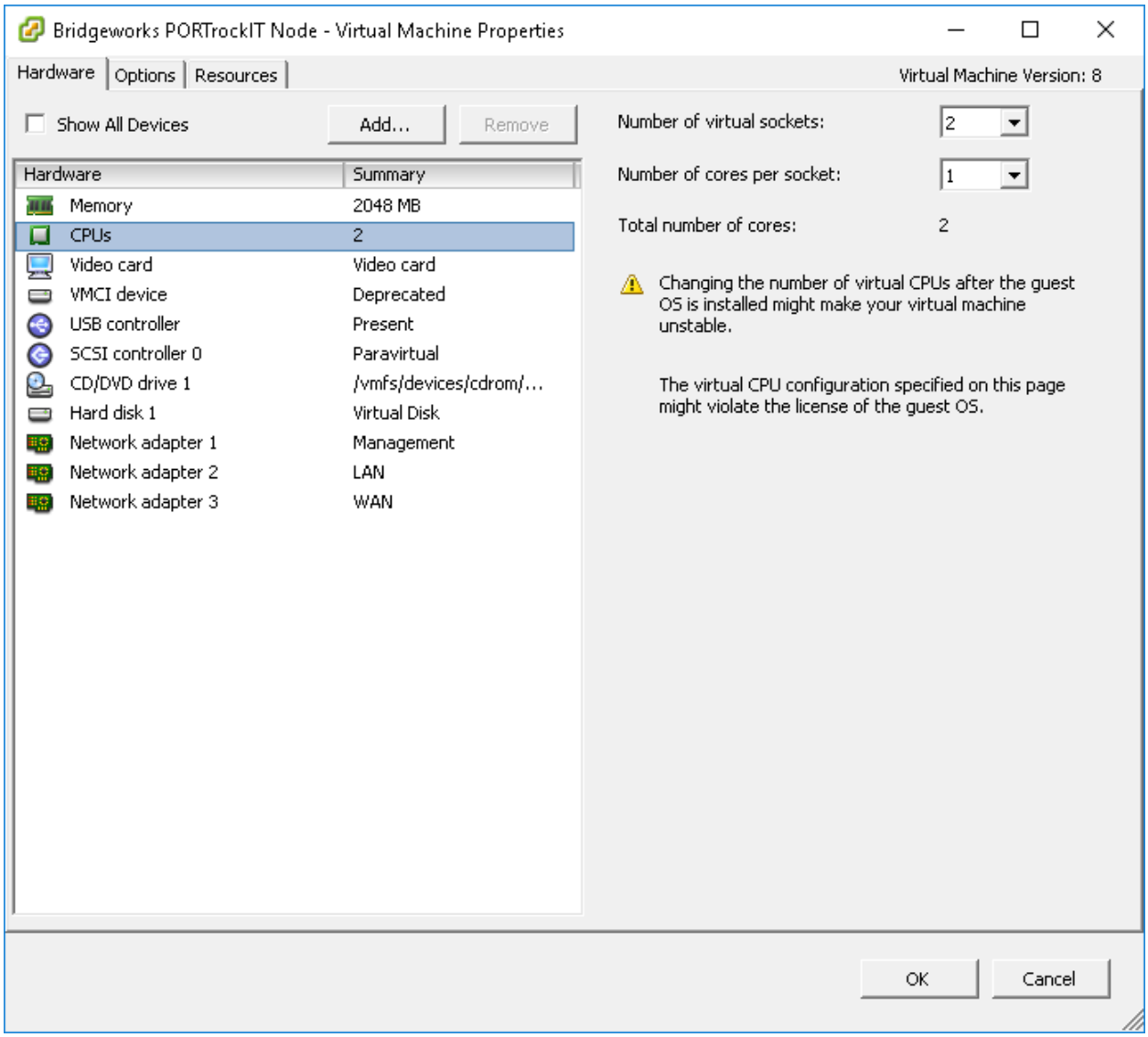
4.1.1 vSphere

From the vSphere client, access the *Edit Settings* screen for the Bridgeworks Node. To access the *Edit Settings* right click the Bridgeworks Node and left click *Edit Settings* from the context menu.

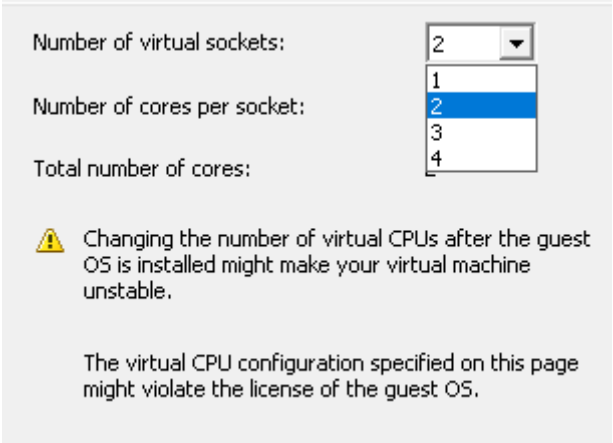


From the settings pane, select the *CPUs* entry from the list presented. Options relating to the CPU allocation will become available on the right side of the settings pane.

Here you have the option to change the number of virtual sockets and the number of cores per socket.



Edit the number field next to *Number of virtual sockets*. Click the dropdown arrow next to the number and select the desired socket count.

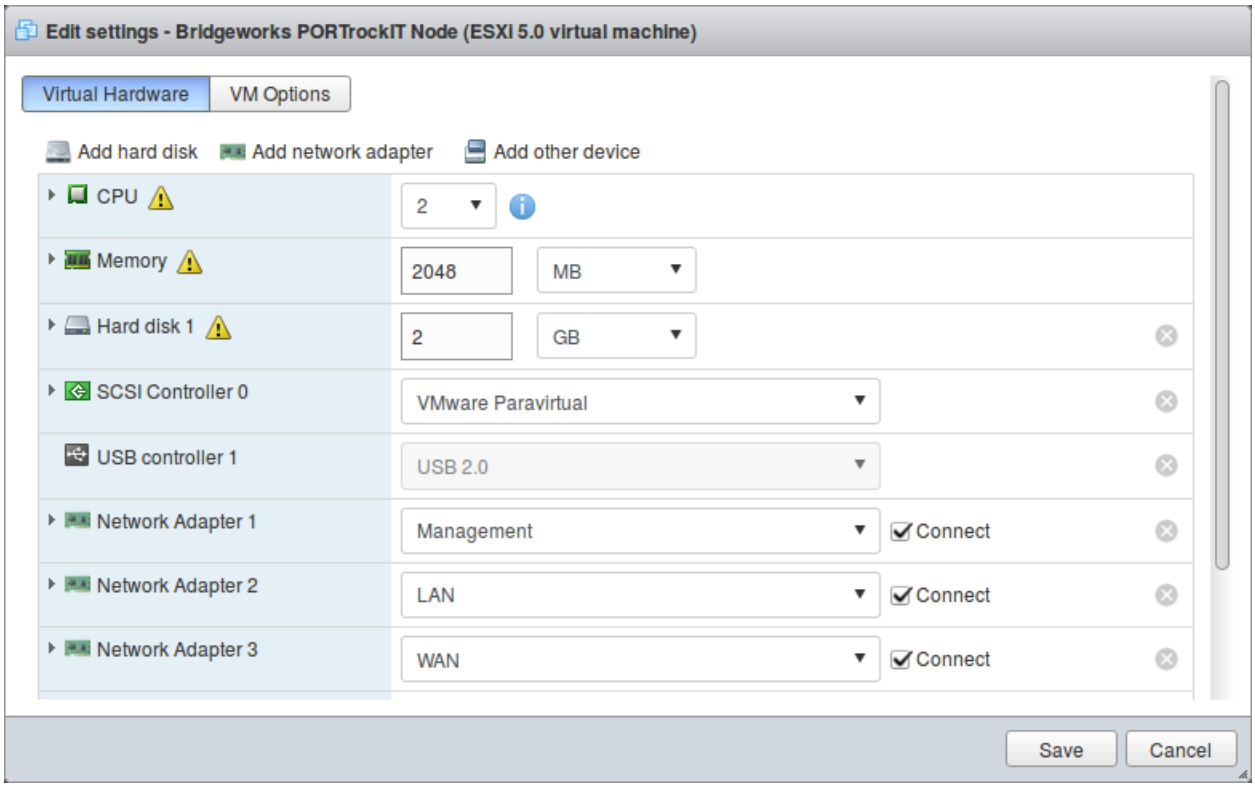


The maximum you can assign is limited to the number of logical cores presented by the physical processors to the ESXi host. This maximum may not be the number of physical cores the system has.

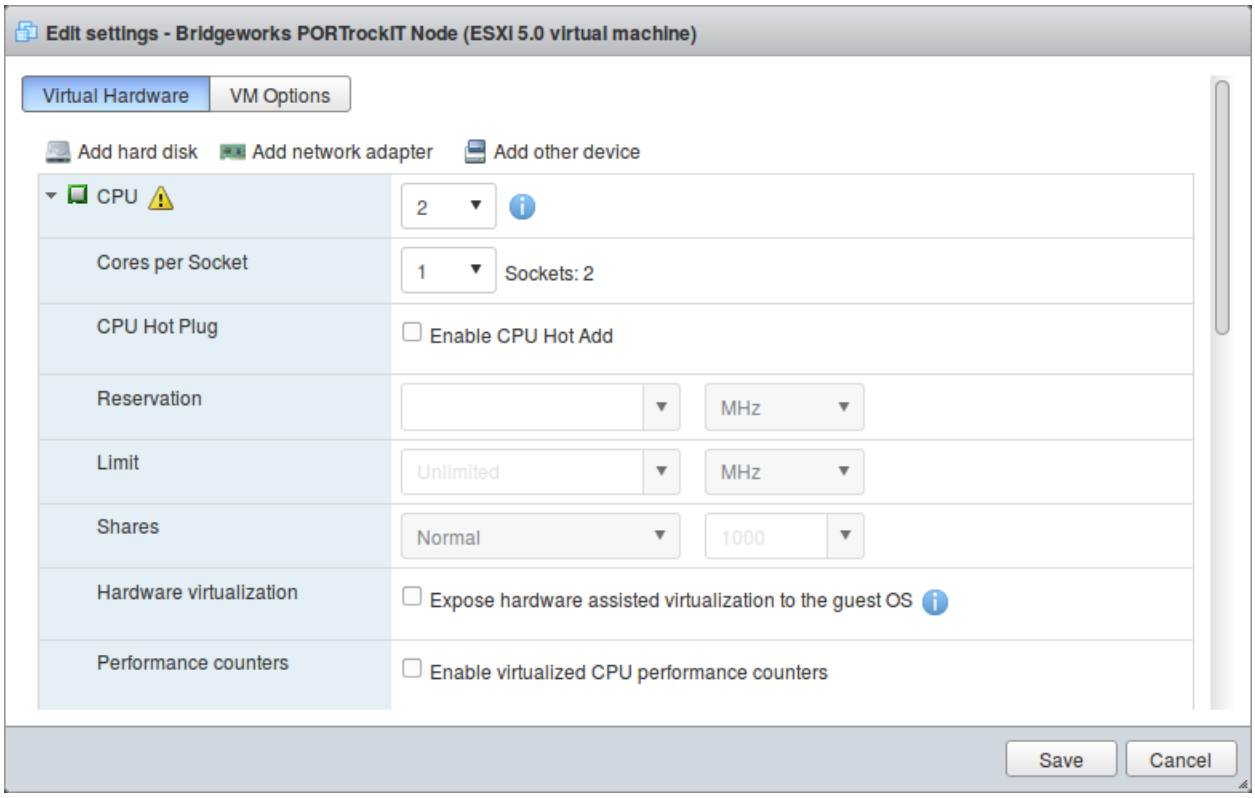
Click *OK* at the bottom of the pane.

4.1.2 ESXi Built-in Web Client

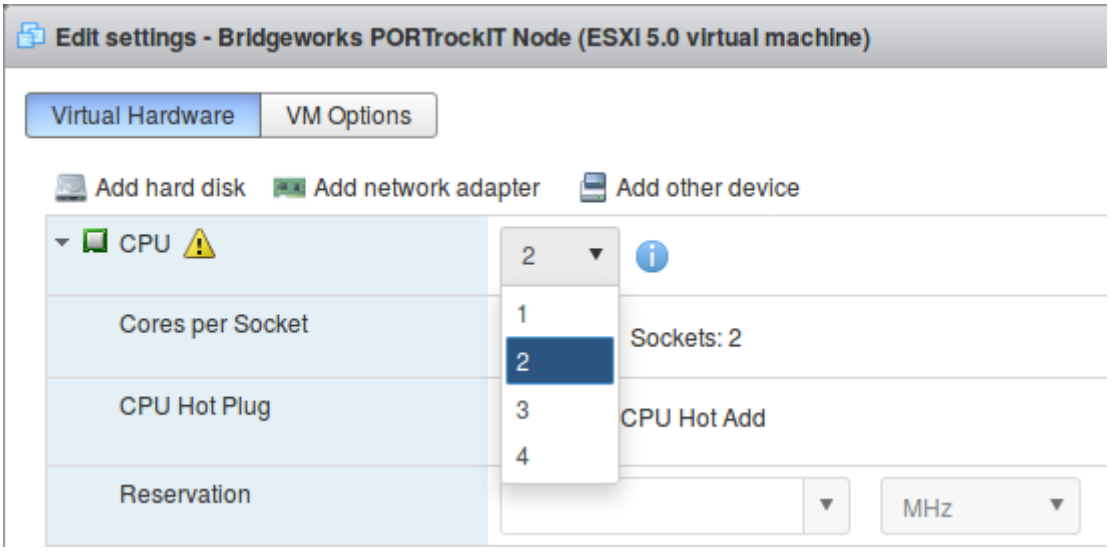
After logging in to the Web Client, navigate to the *Virtual Machines* page and bring up the *Edit settings* pane for the Bridgeworks Node.



In this section, click the arrow left of the *CPU* entry to expand the settings pane and ensure that the *Cores per Socket* is set to 1.



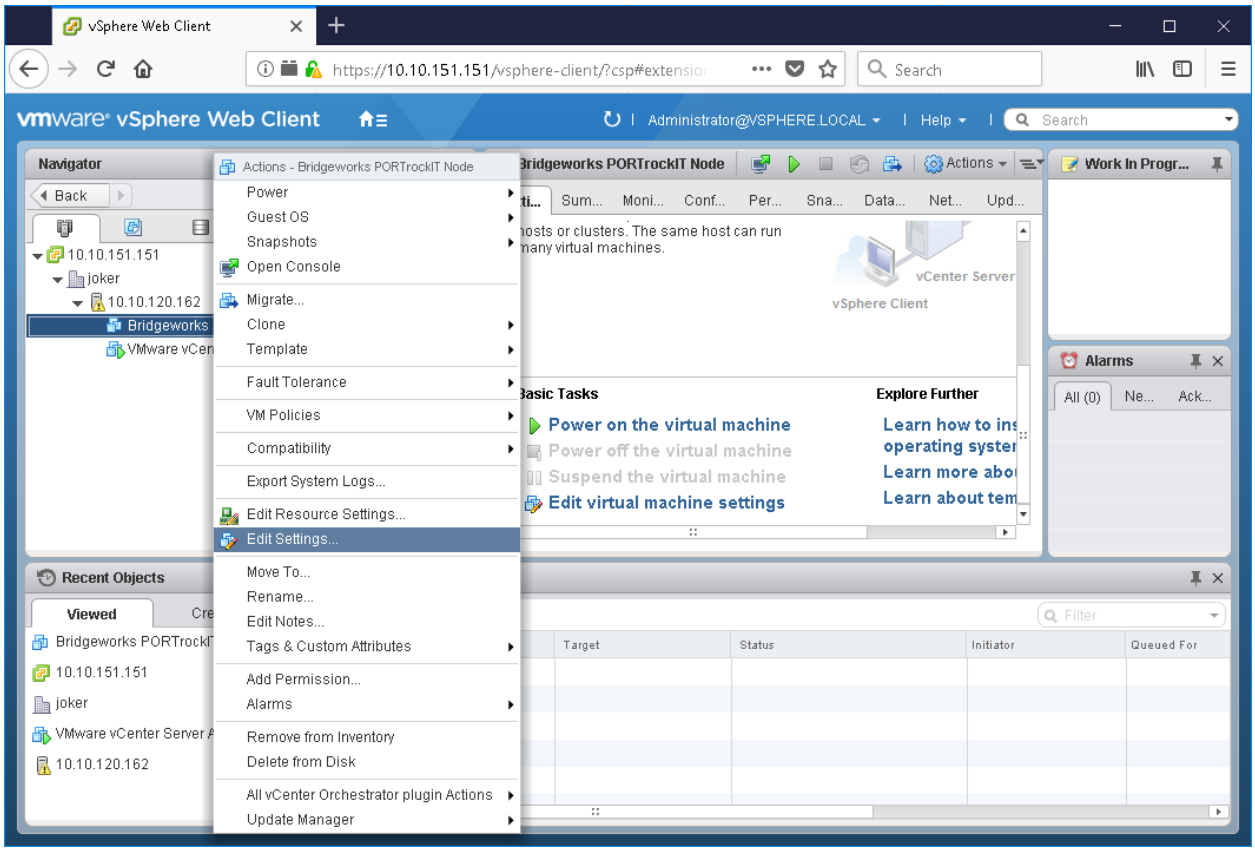
Click the number right of *CPU* and set the desired number of cores.



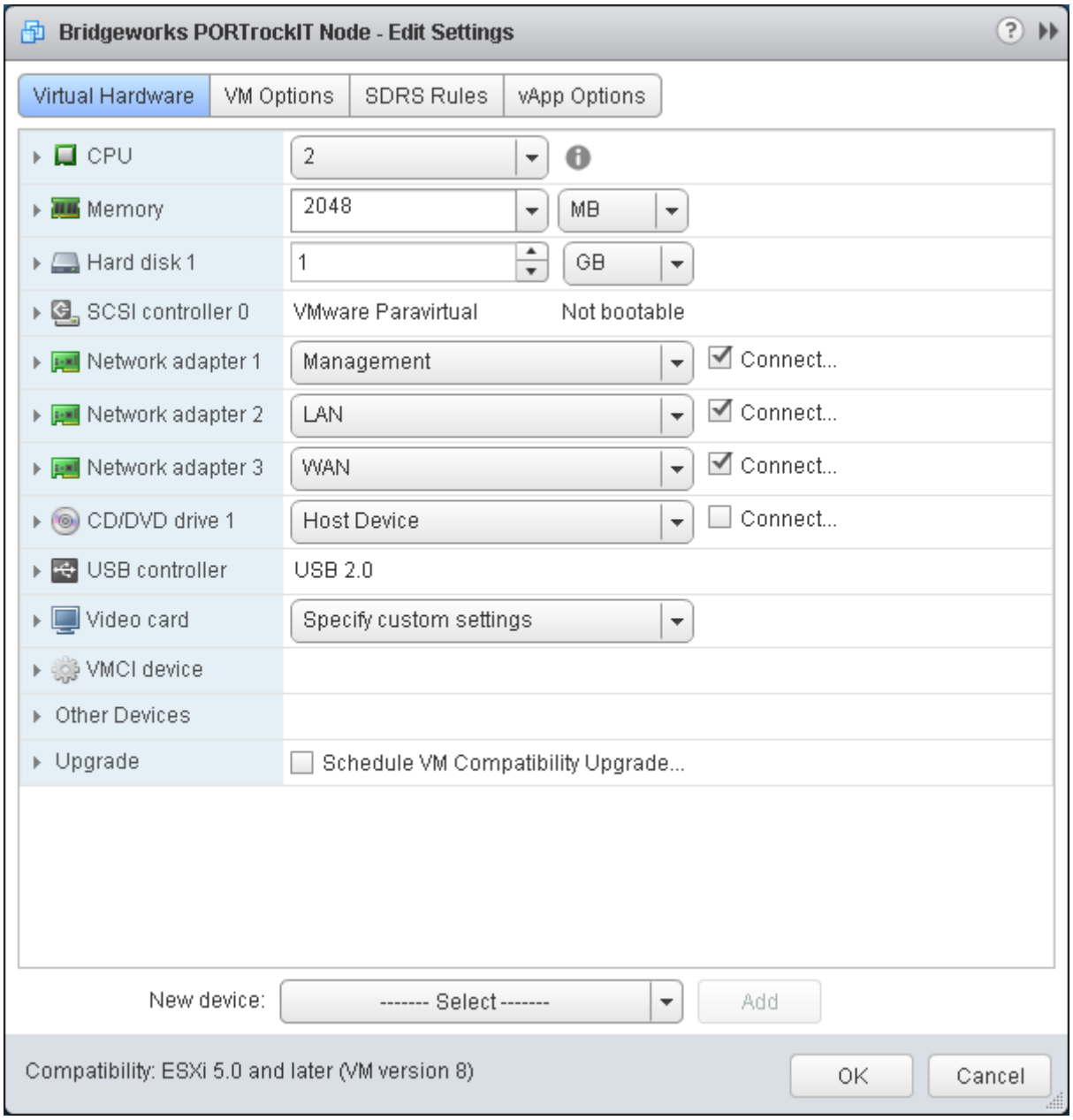
Click *Save* at the bottom of the pane.

4.1.3 vCenter Server Flash Web Client

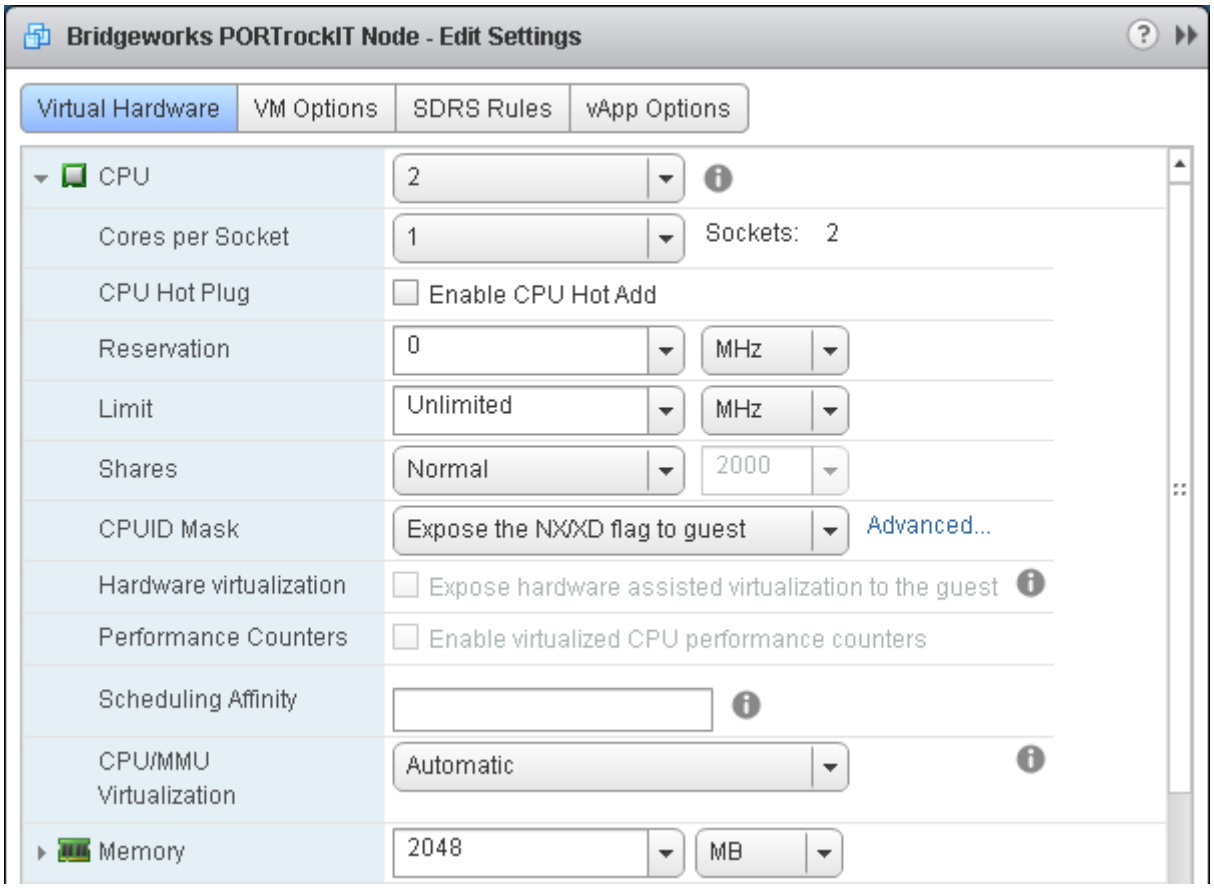
After logging in to the Web Client, navigate to the Bridgeworks Node and access the *Edit Settings* page by right clicking on the Bridgeworks Node and selecting *Edit Settings*.



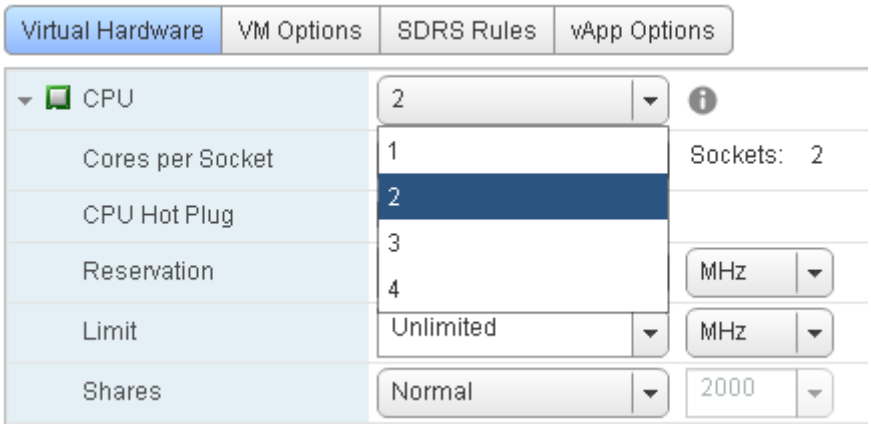
Expand the *CPU* section by clicking the arrow left of the entry.



Ensure that the *Core per Socket* entry is set to 1.



Click the number right of *CPU* and select the desired core count from the dropdown selection.



Click *OK* at the bottom of the pane.

5 Core Affinity

Affinity settings can be changed without having to bring down the guest system, so this can be applied during operation for immediate effect.

ESXi attempts to manage its guests to ensure that competition for any given core is reduced to a minimum, along with arranging memory resources to try to keep guest systems in an aligned setup. An aligned state is when the resources being used by the virtual guest are all locally addressable. For example, memory is allocated on RAM local to the CPU cores that the guest vCPUs are scheduled on. Generally ESXi can keep everything running well.

Issues can arise when resources on the host are overcommitted; multiple guest systems may queue for scheduling on CPU cores, or the ESXi host may have to re-arrange memory usage when host memory availability is running low. The result of resource overcommitment is that the memory allocation may fall over to RAM that is not local to the active vCPUs.

The Bridgeworks Node may suffer when physical CPUs become overcommitted, possibly resulting in reduced network acceleration capabilities.

If there is the possibility of a CPU overcommit, or execution getting moved over to a sub-optimal socket, then giving the Bridgeworks Node a dedicated affinity will help to mitigate slowdowns. Increasing reserve priority will also help if CPU cores cannot be 100% dedicated to the Bridgeworks Node.

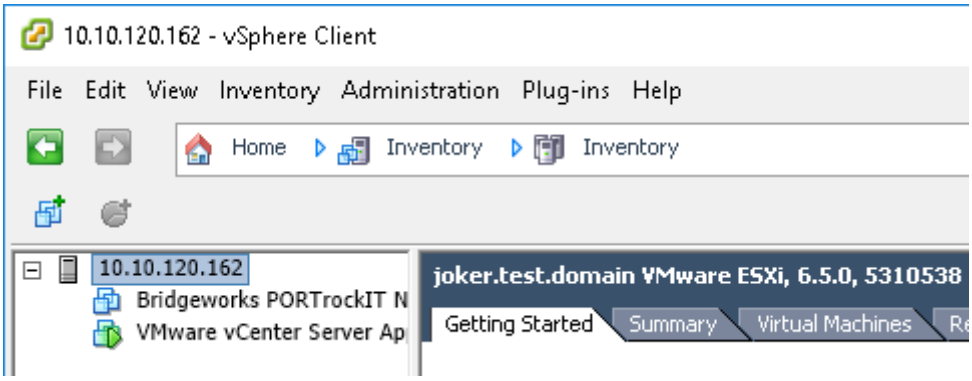
5.1 Assessing Viability

To setup an optimal CPU Affinity some information regarding the ESXi host hardware is required.

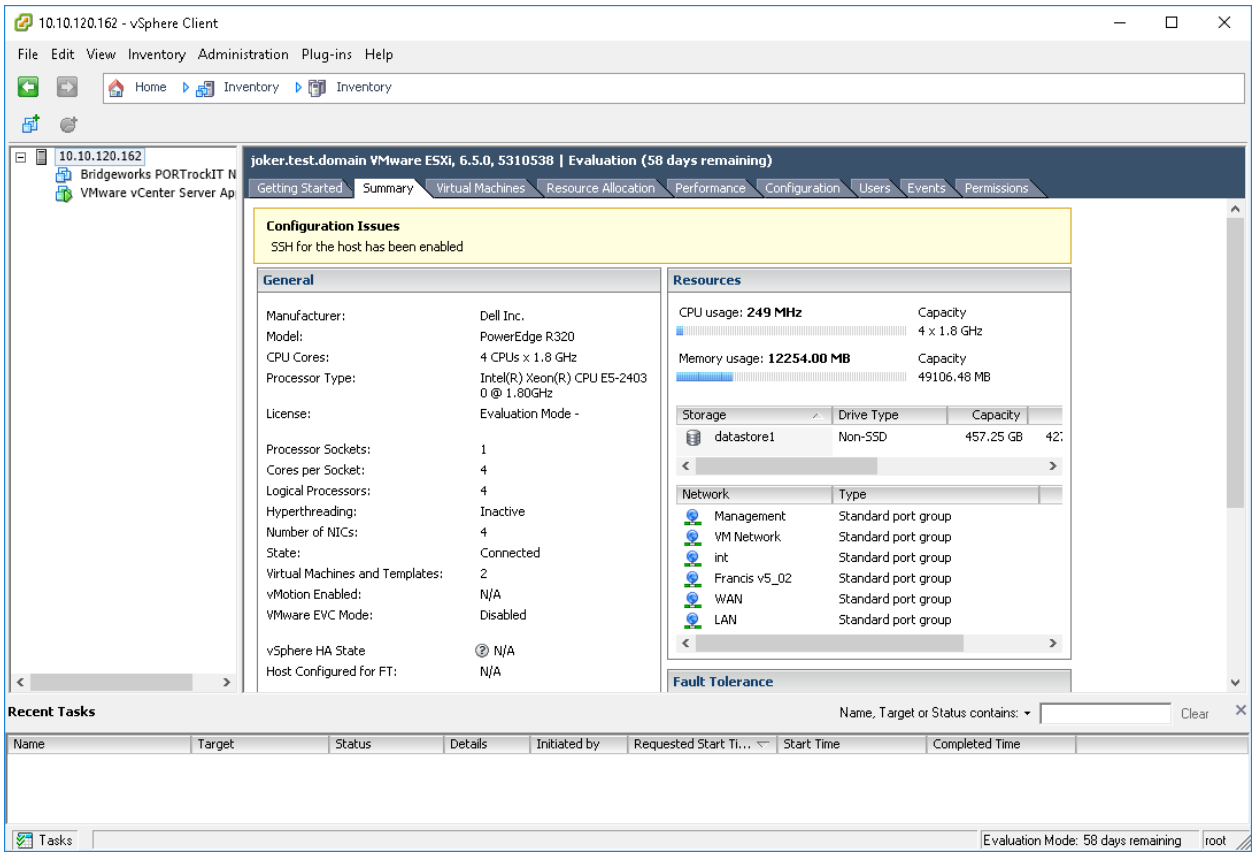
5.1.1 vSphere

In the vSphere client left click on the ESXi host, this will be either the Name or IP address of the host.

The host entry will be the parent of the Bridgeworks Node in the left pane.



Select the *Summary* tab from the entries along the top of the main pane.

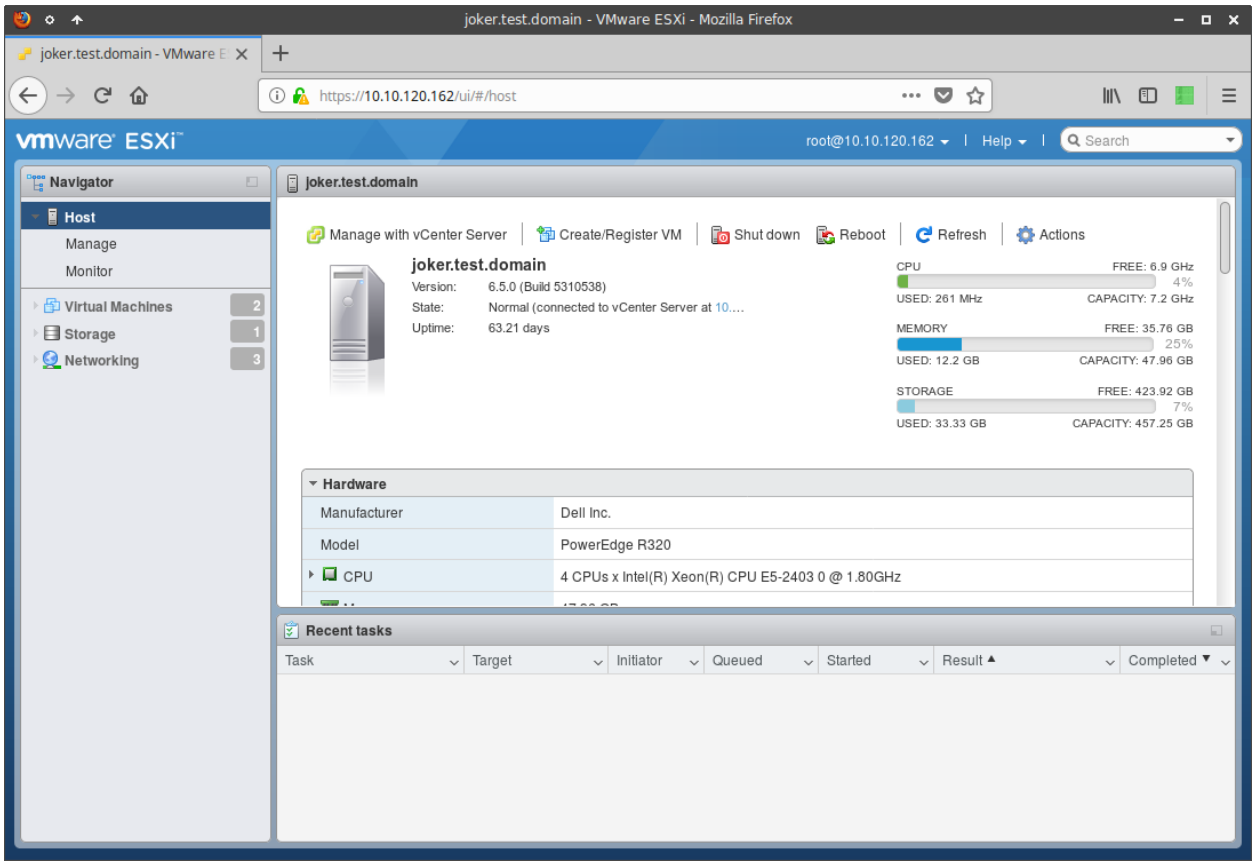


From this screen note the statistics about the processors. The most important entries for this guide are the *Cores per Socket* and *Hyperthreading*.

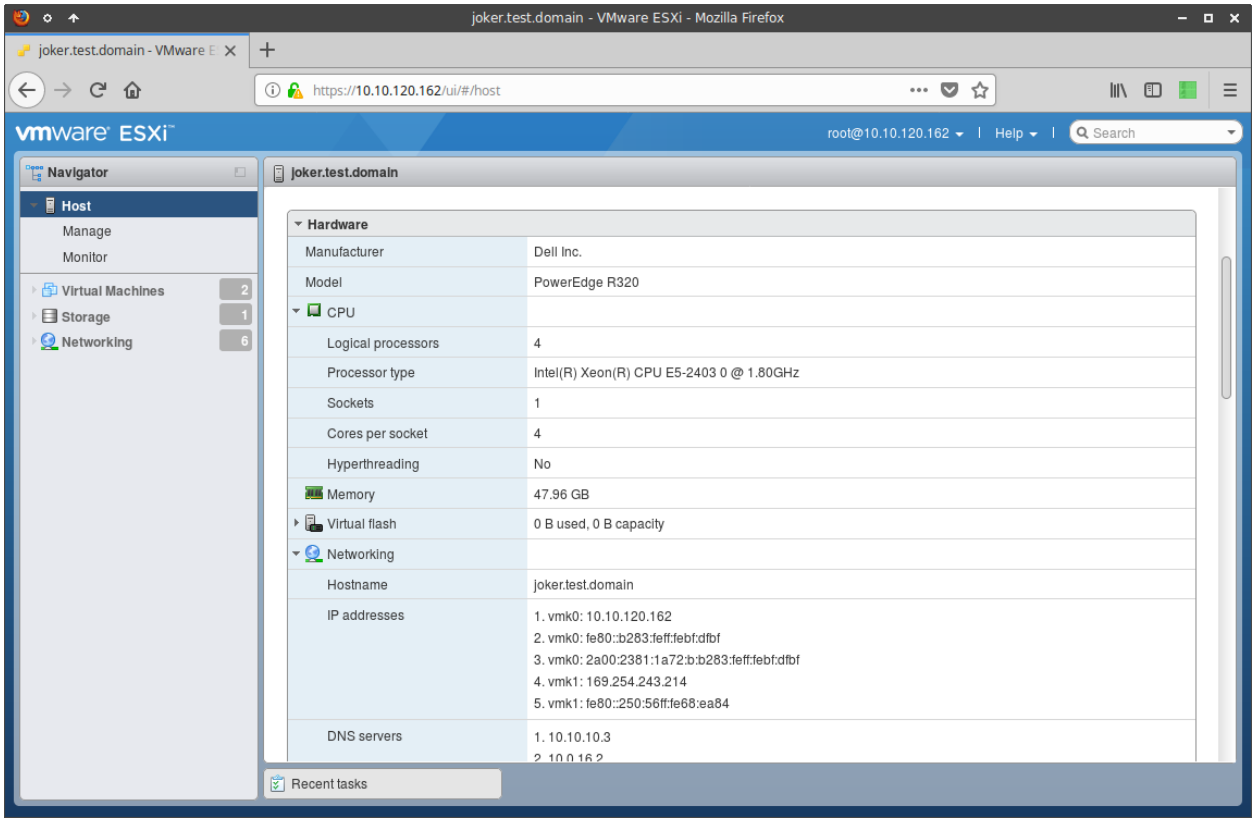
Processor Sockets:	1
Cores per Socket:	4
Logical Processors:	4
Hyperthreading:	Inactive

5.1.2 ESXi built-in Web Client

Upon successful login to the ESXi Web Client the page should navigate automatically to the *Host* page, otherwise select it from the pane at the left side of the browser.



On this page the CPU statistics can be found in the *CPU* section of the *Hardware* box in the main pane.



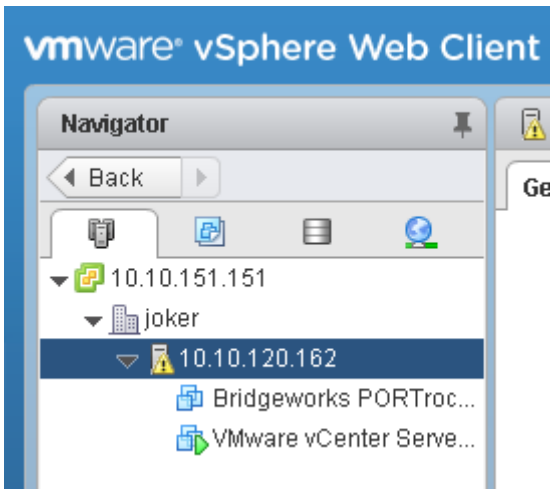
Left clicking on the arrow next to the *CPU* entry will expand the form and present the information needed.

CPU	
Logical processors	4
Processor type	Intel(R) Xeon(R) CPU E5-2403 0 @ 1.80GHz
Sockets	1
Cores per socket	4
Hyperthreading	No

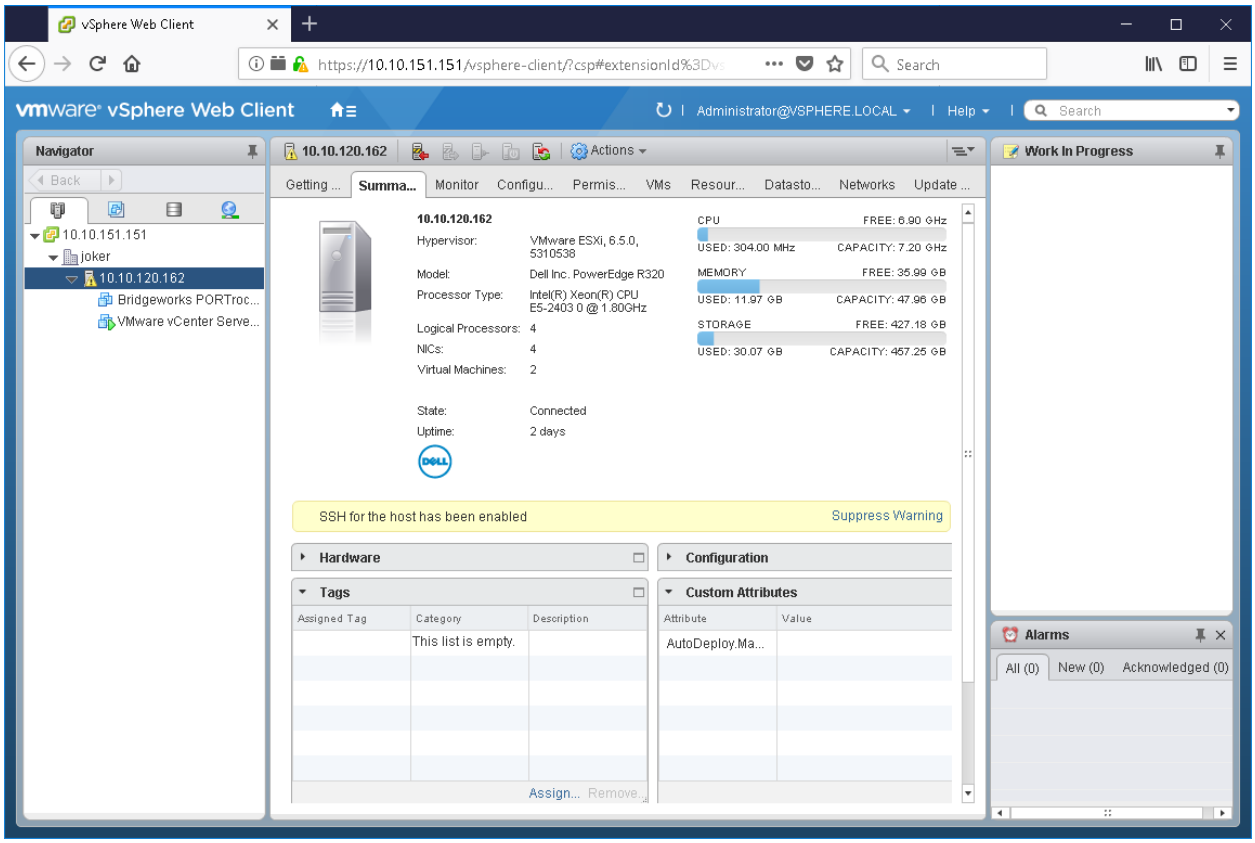
From this screen note the statistics about the processors. The most important entries for this guide are the *Cores per Socket* and *Hyperthreading*.

5.1.3 vCenter Server Flash Web Client

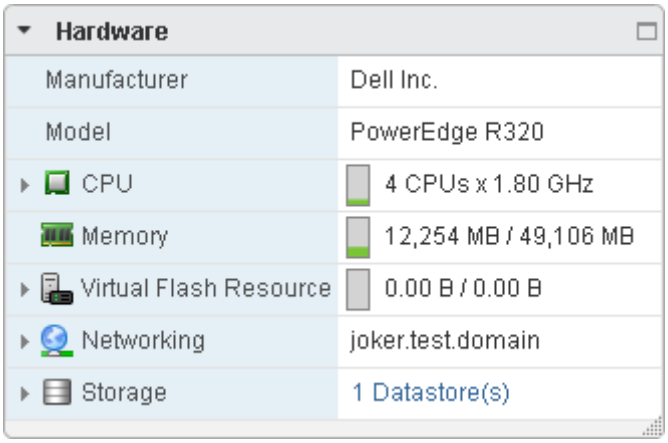
After successful login, use the left pane to navigate to the parent ESXi host of the Bridgeworks Node.



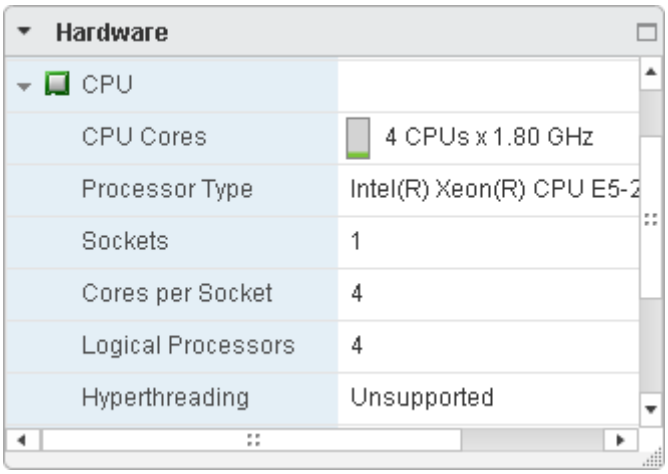
In the main pain, navigate to the *Summary* tab.



Expand the *Hardware* section in the main pane.



Further expand the *CPU* section.

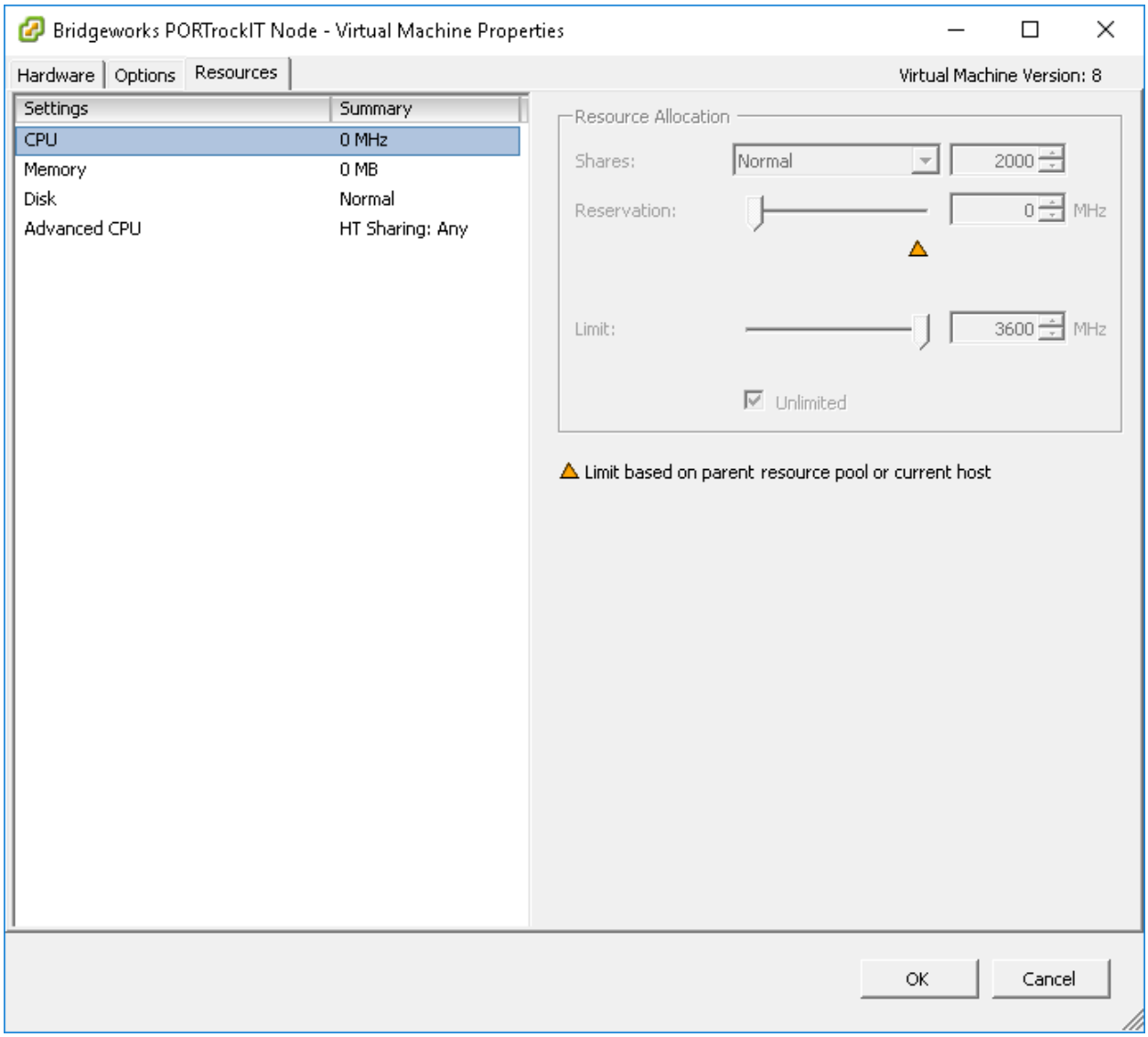


From this screen note the statistics about the processors. The most important entries for this guide are the *Cores per Socket* and *Hyperthreading*.

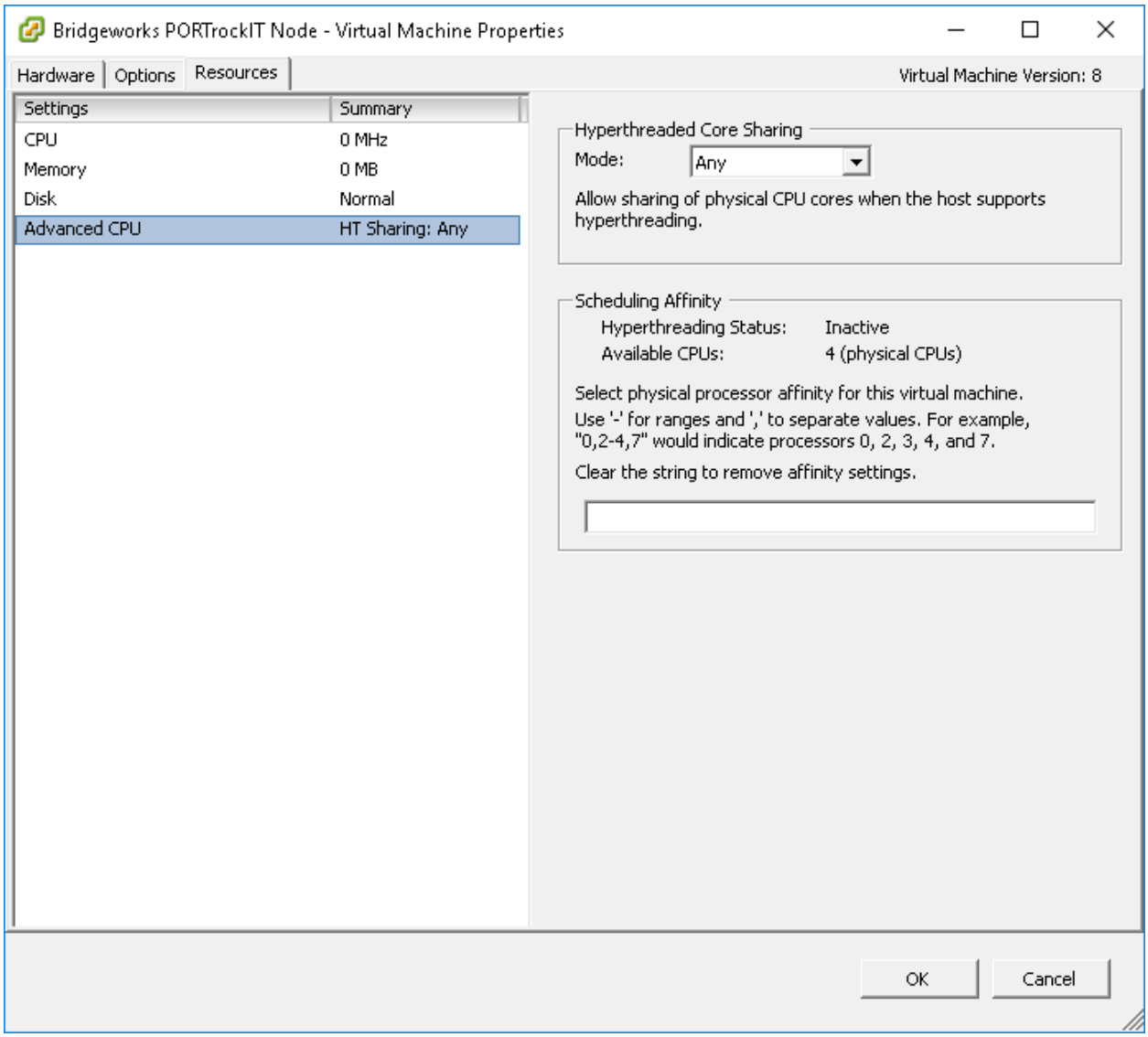
5.2 Applying Affinity

5.2.1 vSphere

In the *Edit Settings* pane, navigate to the *Resources* tab located near the top of the pane.



From here select the *Advanced CPU* entry from the list presented.

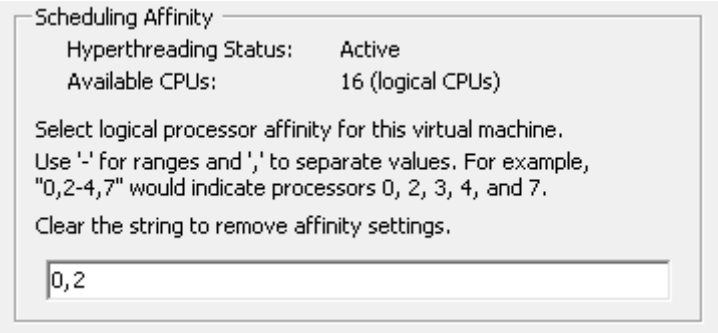


The advanced options available for CPU control are now displayed on the right side of the pane. For setting CPU affinity the text box in the *Scheduling Affinity* section will be used. The help text in this section describes how to input valid affinity.

At this stage the information from the Host processor setup becomes useful. The best affinity settings would have a CPU affinity that only selects a single logical core from each physical core, and only cores that exist on the same physical socket.

For example, the image below shows a 2 core Bridgeworks Node being given an affinity of cores 0 and 2. This is an optimal setup because the cores exist on a single socket of the dual socket host.

With Hyperthreading enabled, it is important to note that the logical cores 0 and 1 both share execution time on the first physical core of the first socket. Cores 2 and 3 would share the next physical core of the same socket.



Click *OK* at the bottom of the pane to apply the new memory settings.

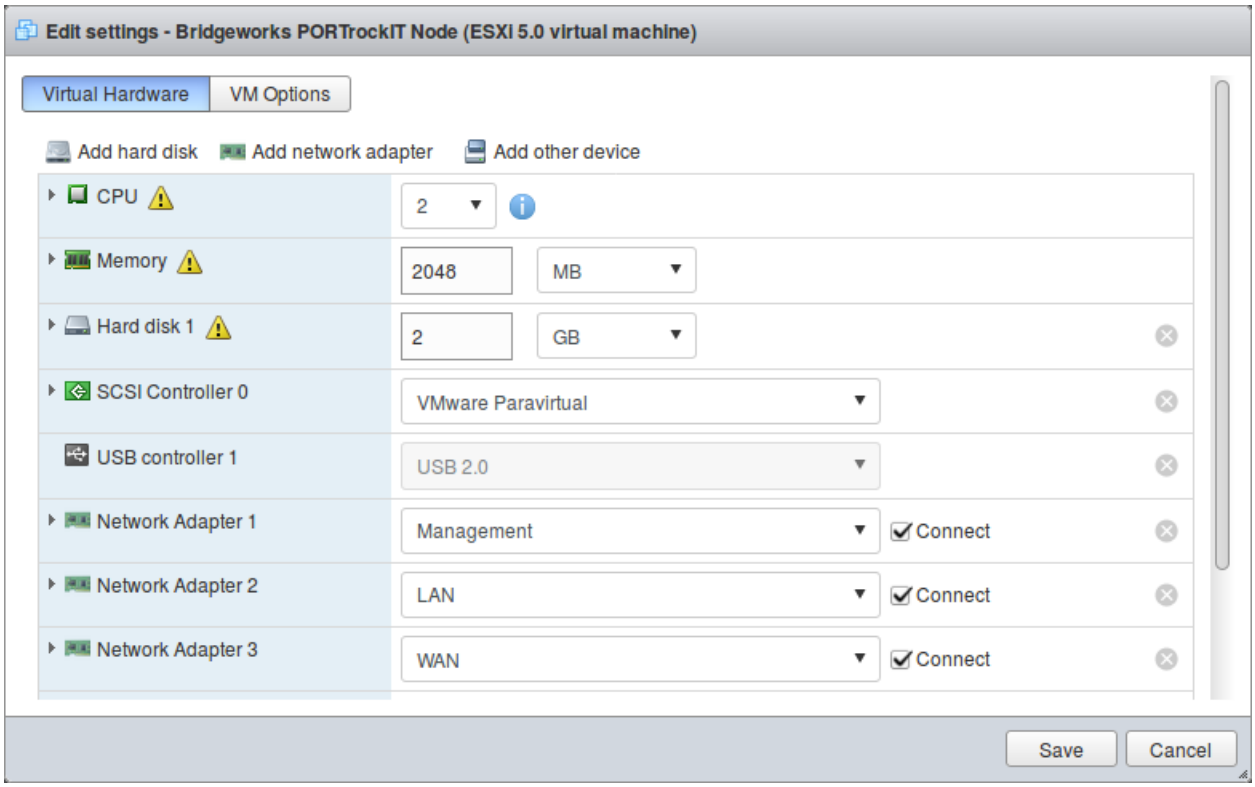
If memory affinity has been set (see Chapter 7: Memory Alignment), this option needs to line up with the same value. In this example, the memory affinity for the Bridgeworks Node will be set to only run on memory from the first memory node.

Summary of example:

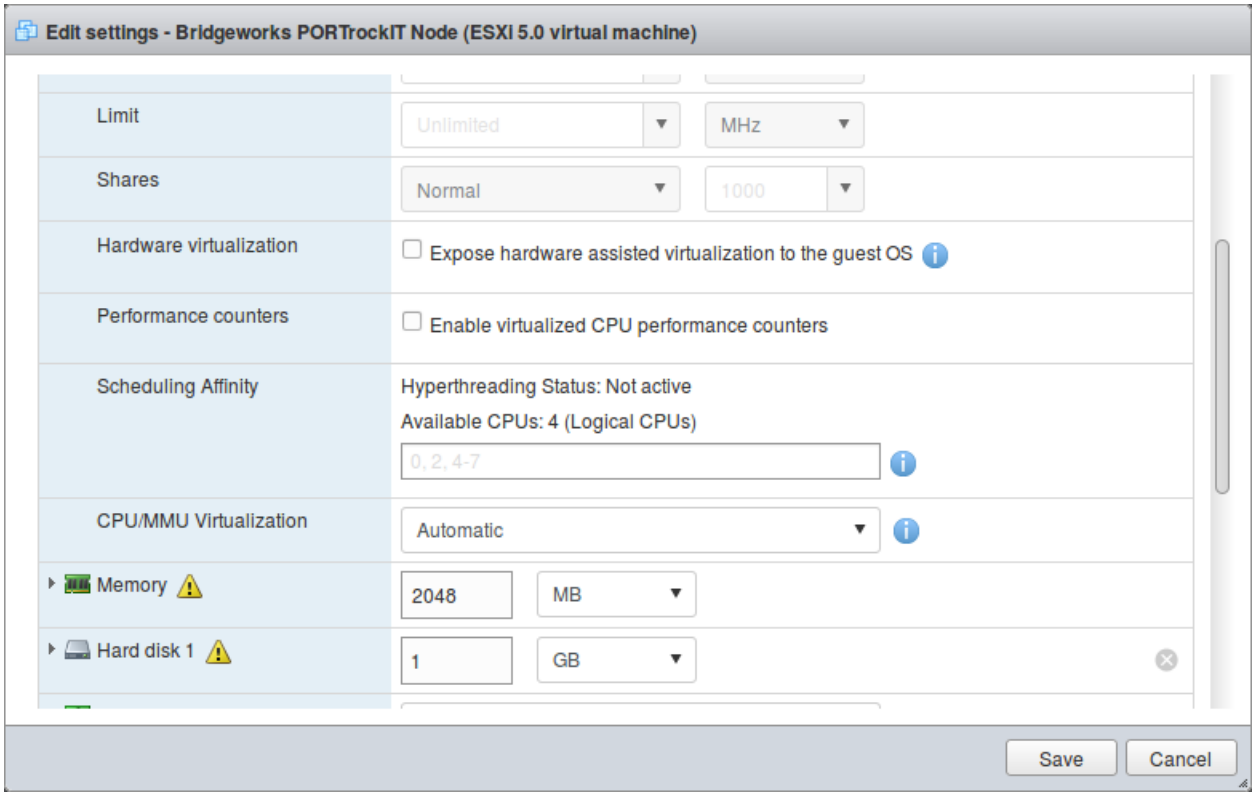
- An ESXi host has 2 sockets, each with 4 cores and Hyperthreading enabled. A total of 16 logical cores are available from the host.
- The Bridgeworks Node has been given memory affinity to run on the first memory node.
- It will be given 2 cores.
- With Hyperthreading enabled all consecutive logical cores are paired together on a single underlying physical core. E.g. 0 and 1 exist on the first physical core, 2 and 3 exist on the second core, and so on.
- Do not pick logical cores that exist on the same physical core, ”0,1” would result in lower performance when under load.
- The CPU affinity here could be input as ”0,2” to ensure that the logical cores given to the node do not compete with each other for the physical core resources.
- ”1,2” would also be a good setup, as 1 exists on the first physical, core but 2 exists on the second physical core.

5.2.2 ESXi built-in Web Client

In the *Edit Settings* pane, use the arrow to the left of the *CPU* entry to expand that section.



Scroll down to locate the *Scheduling Affinity* section.



For setting CPU affinity the text box in the *Scheduling Affinity* section will be used. The affinity input is formatted as a series of numbers representing the cores. The range is from 0 to one less than the total number of cores.

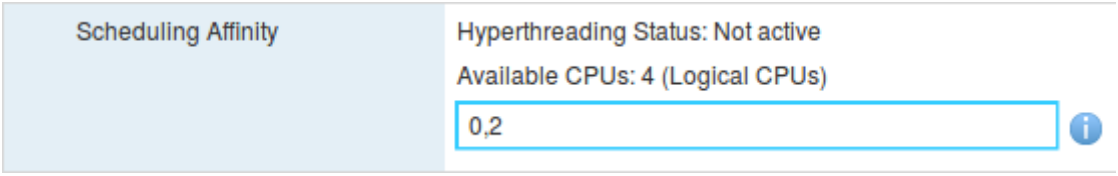
For example, an ESXi host running 12 cores will have a range of 0 to 11.

The numbers are separated by '-' or ',' where:

- '0-3' represents cores: 0,1,2,3.
- '0,3' represents only cores 0 and 3.
- The formatting can be mixed. E.g. '0,2-4' represents cores: 0,2,3,4.

At this stage the information from the Host processor setup becomes useful. The best affinity settings would have a CPU affinity that only selects physical cores that all exist on the same socket.

The image below shows a 2 core Bridgeworks Node being given an affinity of cores 0 and 2. The Host information states that Hyperthreading is not running on this ESXi system so all available cores are real physical CPU cores.

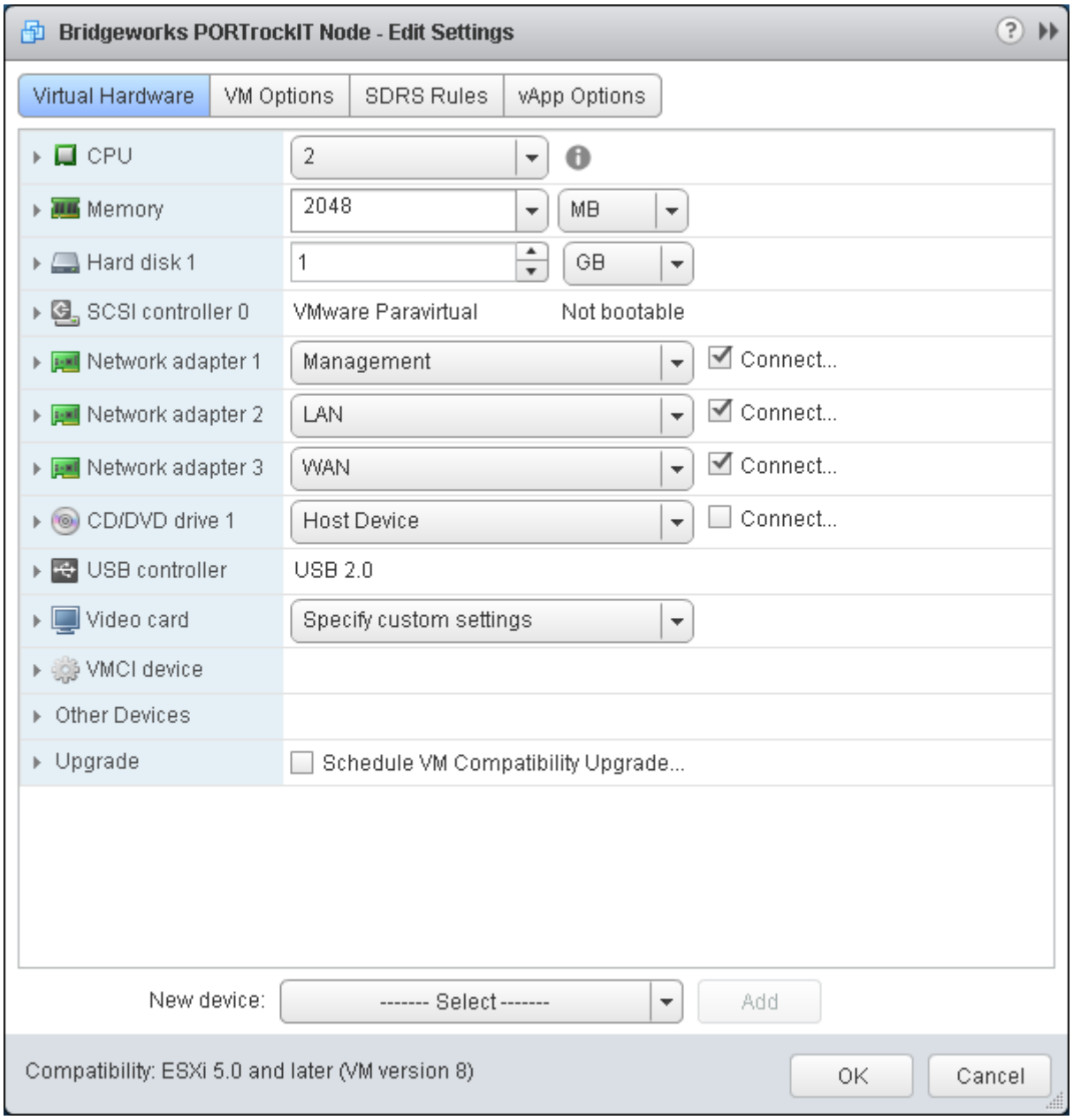


If Hyperthreading were enabled, it would be important to note that 0 and 1 would exist on the first physical core of the first socket. Likewise 2 and 3 would be the second physical core of the same socket.

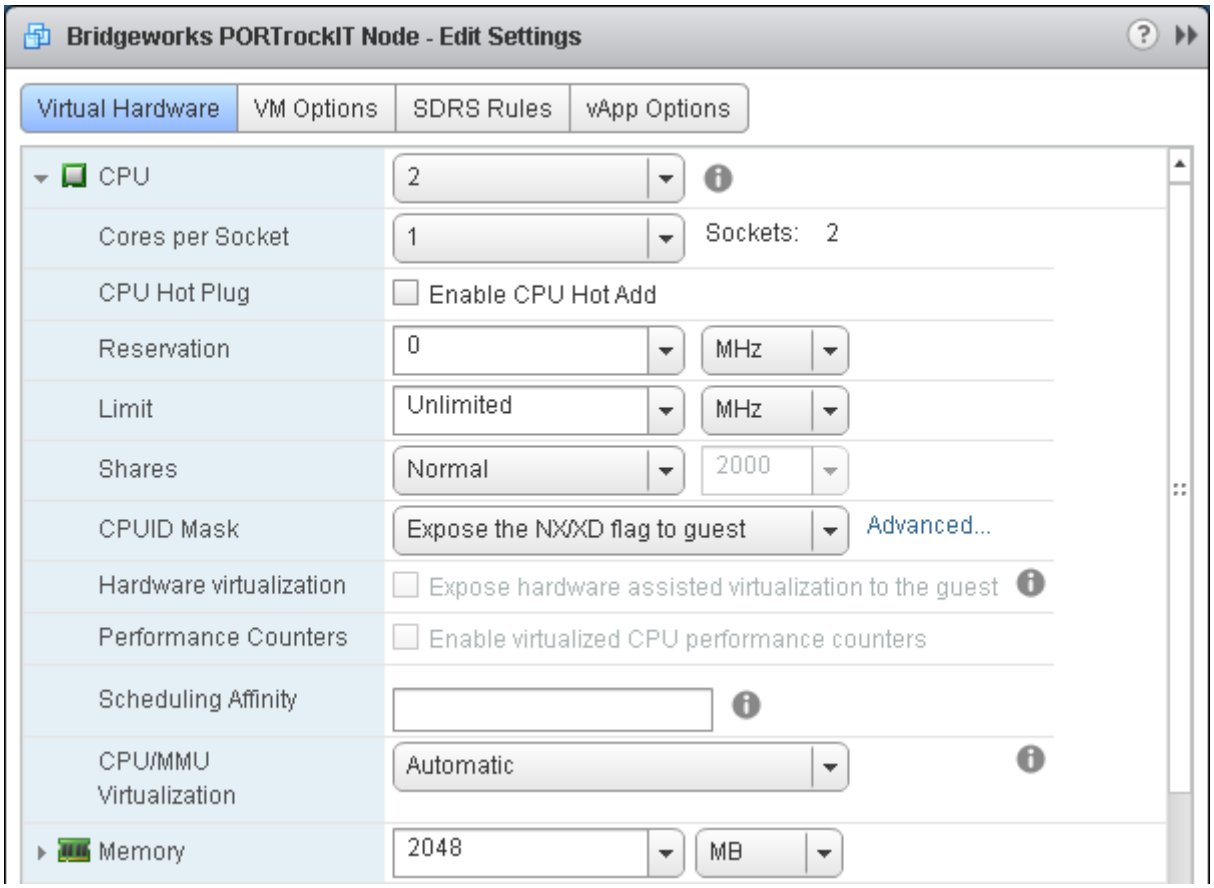
Click *Save* at the bottom of the pane to apply the new memory settings.

5.2.3 vCenter Server Flash Web Client

In the *Edit Settings* pane, use the arrow to the left of the *CPU* entry to expand that section.



Scheduling Affinity is located at the bottom of the expanded section.



For setting CPU affinity the text box in the *Scheduling Affinity* section will be used. The affinity input is formatted as a series of numbers representing the cores. The range is from 0 to one less than the total number of cores.

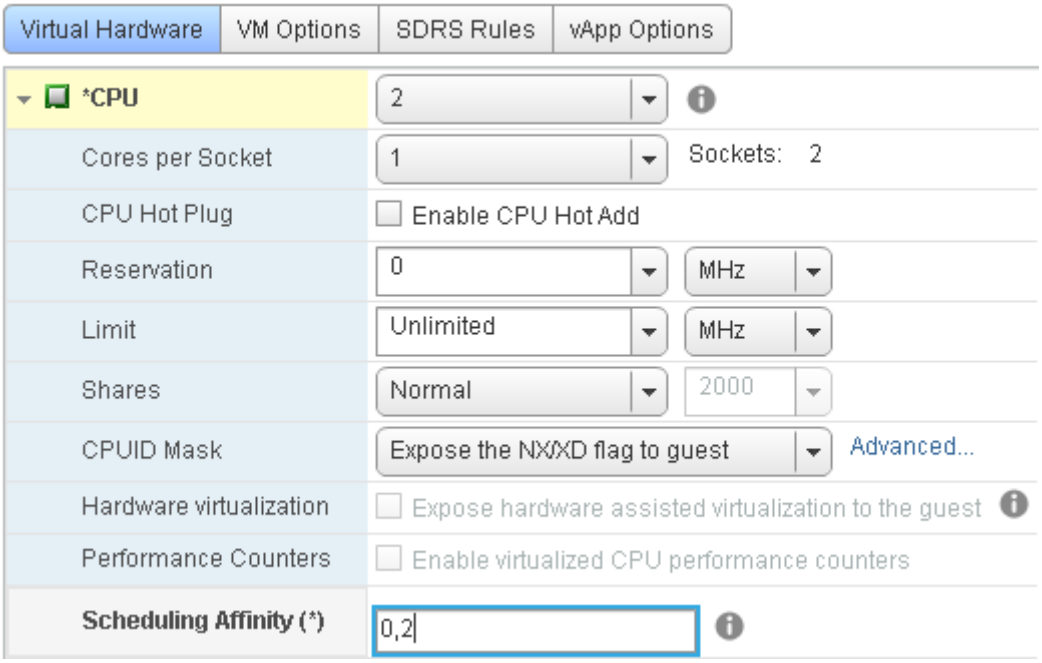
For example, an ESXi host running 12 cores will have a range of 0 to 11.

The numbers are separated by '-' or ',' where:

- '0-3' represents cores: 0,1,2,3.
- '0,3' represents only cores 0 and 3.
- The formatting can be mixed. E.g. '0,2-4' represents cores: 0,2,3,4.

At this stage the information from the Host processor setup becomes useful. The best affinity settings would have a CPU affinity that only selects physical cores that all exist on the same socket.

The image below shows a 2 core Bridgeworks Node being given an affinity of cores 0 and 2. The Host information states that Hyperthreading is not running on this ESXi system so all available cores are real physical CPU cores.



If Hyperthreading were enabled, it would be important to note that 0 and 1 would exist on the first physical core of the first socket. Likewise 2 and 3 would be the second physical core of the same socket.

Click *OK* at the bottom of the pane to apply the new memory settings.

6 Memory Allocation

Initial memory allocations for each system are:

- 100 Series = 2GB
- 200 Series = 4GB
- 400 Series = 16GB

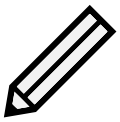
These values are calculated on the basis of how much temporary pool data is utilised during a full bandwidth transfer across a potentially highly utilised network environment.

Ideally, the ESXi host will be able to accommodate the entire memory allocation on the physical RAM slots for a single socket without any competition associated with other guest systems.

If the memory allocation results in RAM starvation of the ESXi host or memory being split across the physical sockets (see Chapter 7: Memory Alignment), then the options are:

- Assess whether the RAM allocation of the Bridgeworks Node can be reduced.

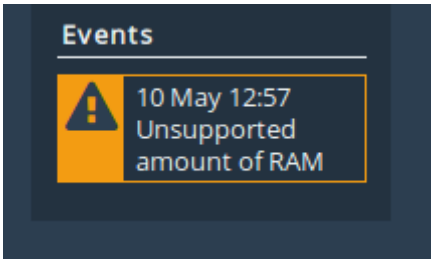
  The System Information page on the web GUI for the node shows the current memory usage, during typical data acceleration this can give a good indication of the memory allocation requirements.
- Consider reordering other virtual guests on the ESXi host to redistribute memory usage. This could free up enough memory on a single socket to fit the Bridgeworks node.
- If the previous options aren't possible then either move the Bridgeworks Node to another ESXi host entirely, or leave the memory in the sub-optimal setup.



Note: The minimum supported memory allocation for a Bridgeworks node is:

- 100 Series = 2GB
- 200 Series = 4GB
- 400 Series = 12GB

If the Bridgeworks Node is set below the minimum supported memory allocation the Web GUI will display a warning to the user:

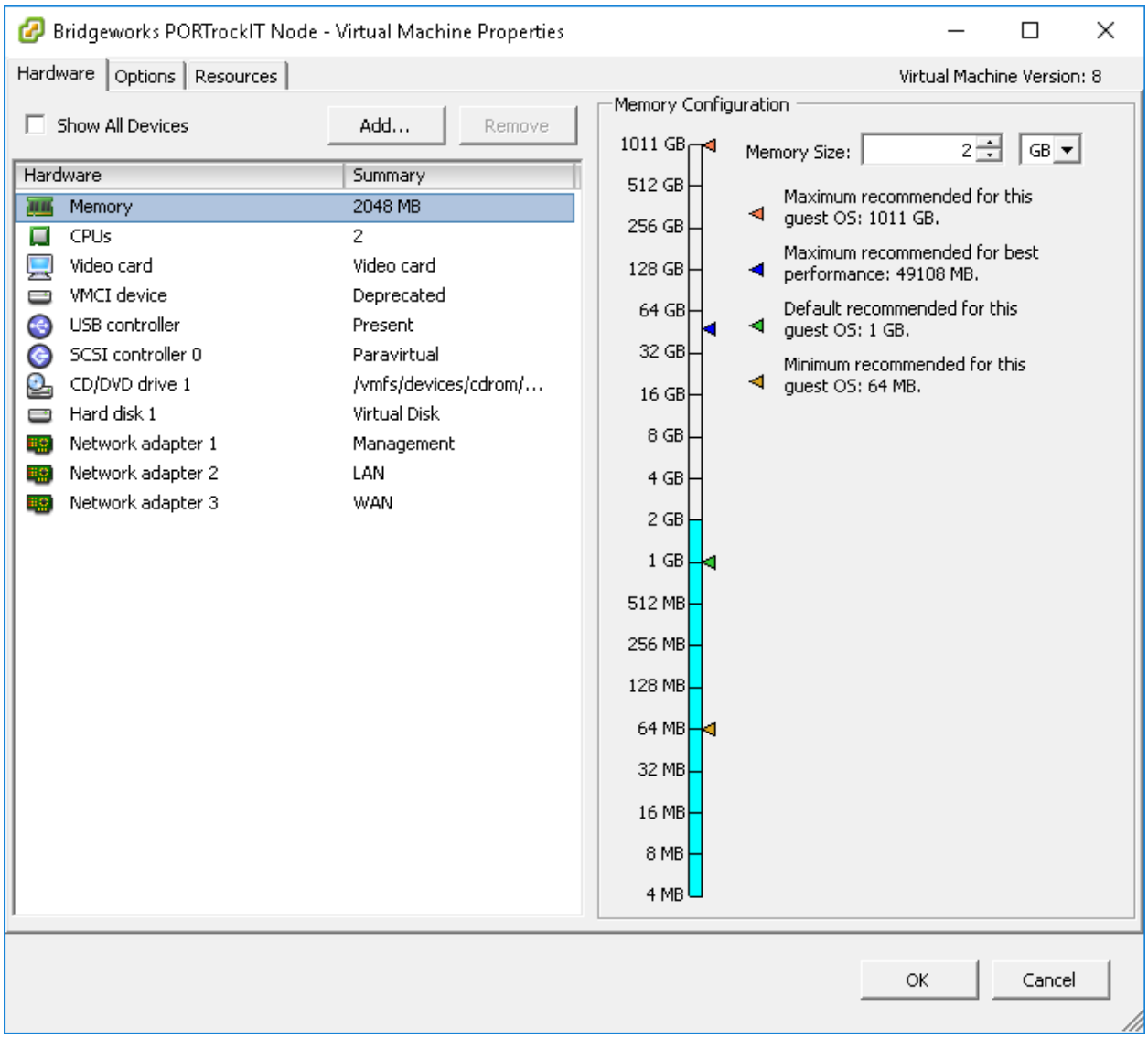


6.1 Adjusting Allocation

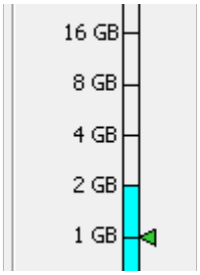
6.1.1 vSphere

In the *Edit Settings* pane, click the *Memory* option from the list.

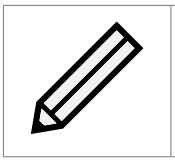
The right side of the pane will show the current memory settings.



The easiest option to change the memory allocation is to click on the numbers along the left side of the scale. This entry method is easy to repeat.



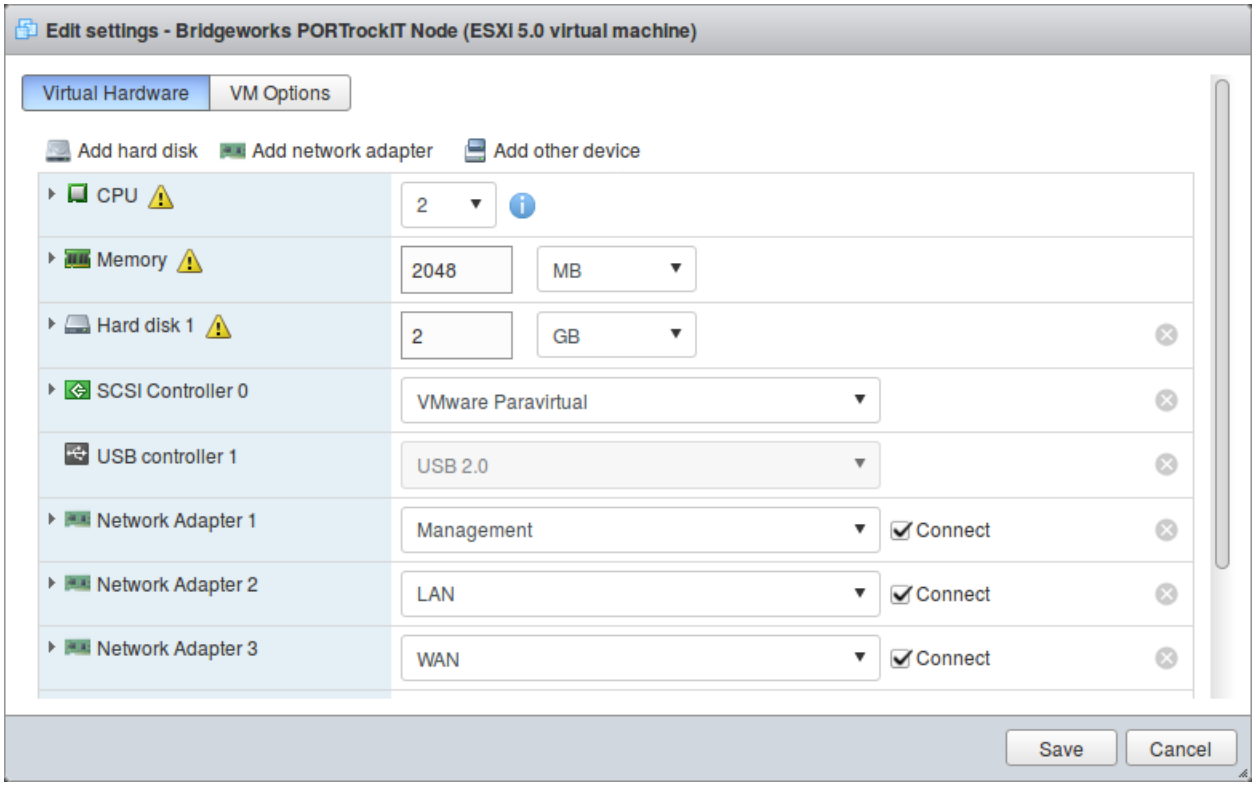
Click *OK* at the bottom of the pane to apply the new memory settings.



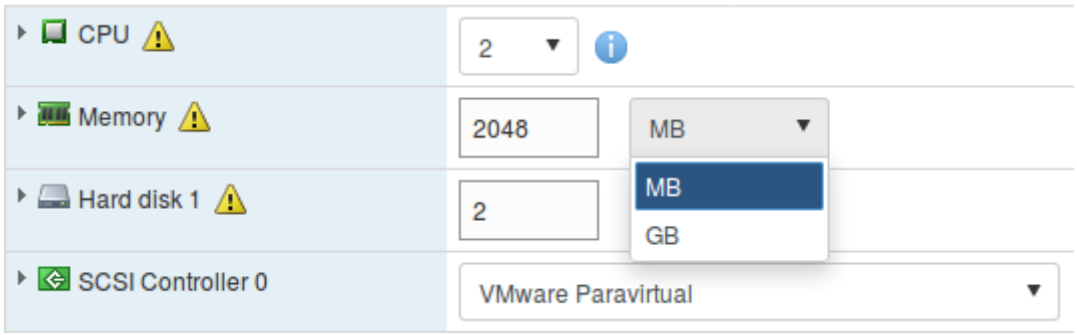
Note: The coloured arrows are suggestions by the ESXi host based entirely on the host systems memory resource size and are not tailored for the Bridgeworks Node use case.

6.1.2 ESXi Built-in Web Client

In the *Edit Settings* pane, navigate to the *Virtual Hardware* tab and select the value for the *Memory* entry.



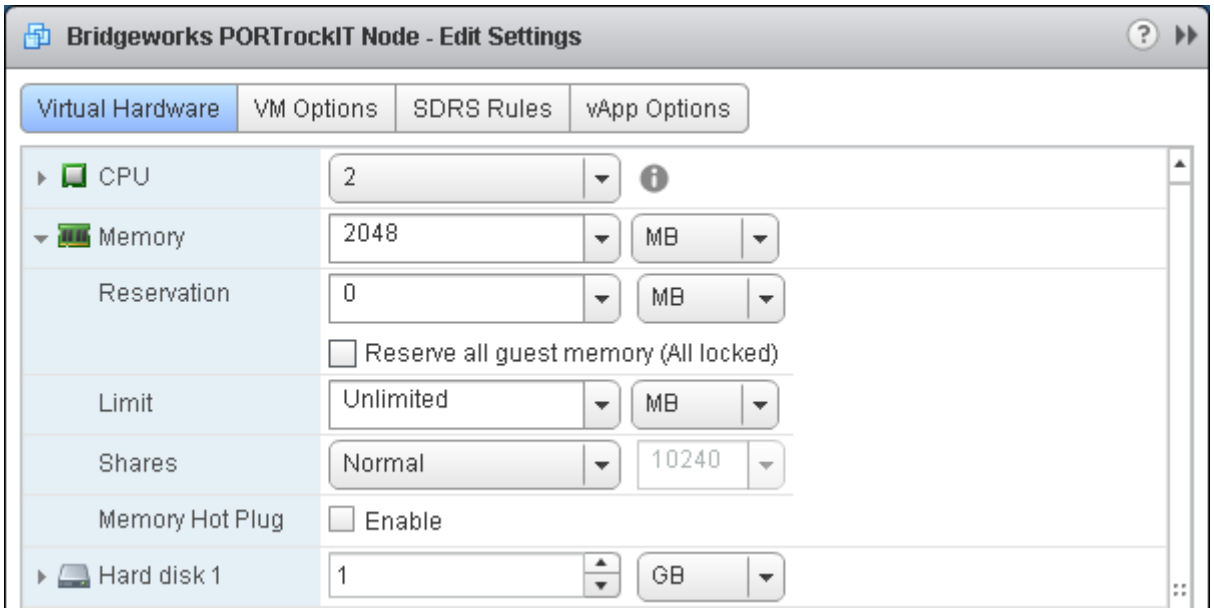
Enter the desired memory allocation. Changing the value to the right from *MB* to *GB* will be easier. Scale the number to suit the change.



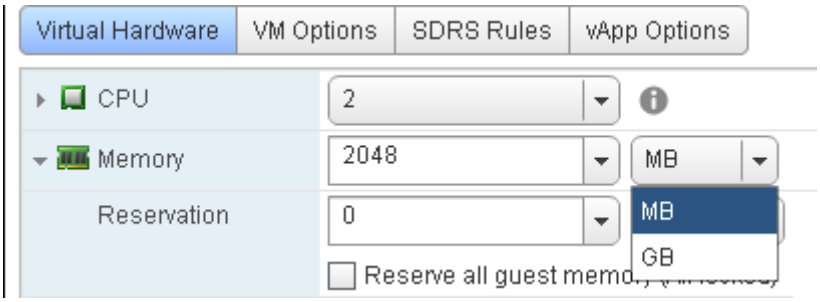
Click *Save* at the bottom of the pane to apply the new memory settings.

6.1.3 vCenter Server Flash Web Client

In the *Edit Settings* pane, navigate to the *Virtual Hardware* tab and select the value for the *Memory* entry.



Changing the value to the right from *MB* to *GB* will be easier.



Enter the desired memory allocation. Scale the number to suit the change.

Click *Save* at the bottom of the pane to apply the new memory settings.

7 Memory Alignment

A lack of memory available can cause a drop in performance. In some cases an operating system can become unresponsive while it reverts to pagefile or swap space to maintain operation.

Unlike a memory deficit, memory alignment will not be as catastrophic. Although performance can be reduced by as much as 20-30%.

Ensuring memory is aligned will reduce: CPU usage, memory usage, and increase overall performance.

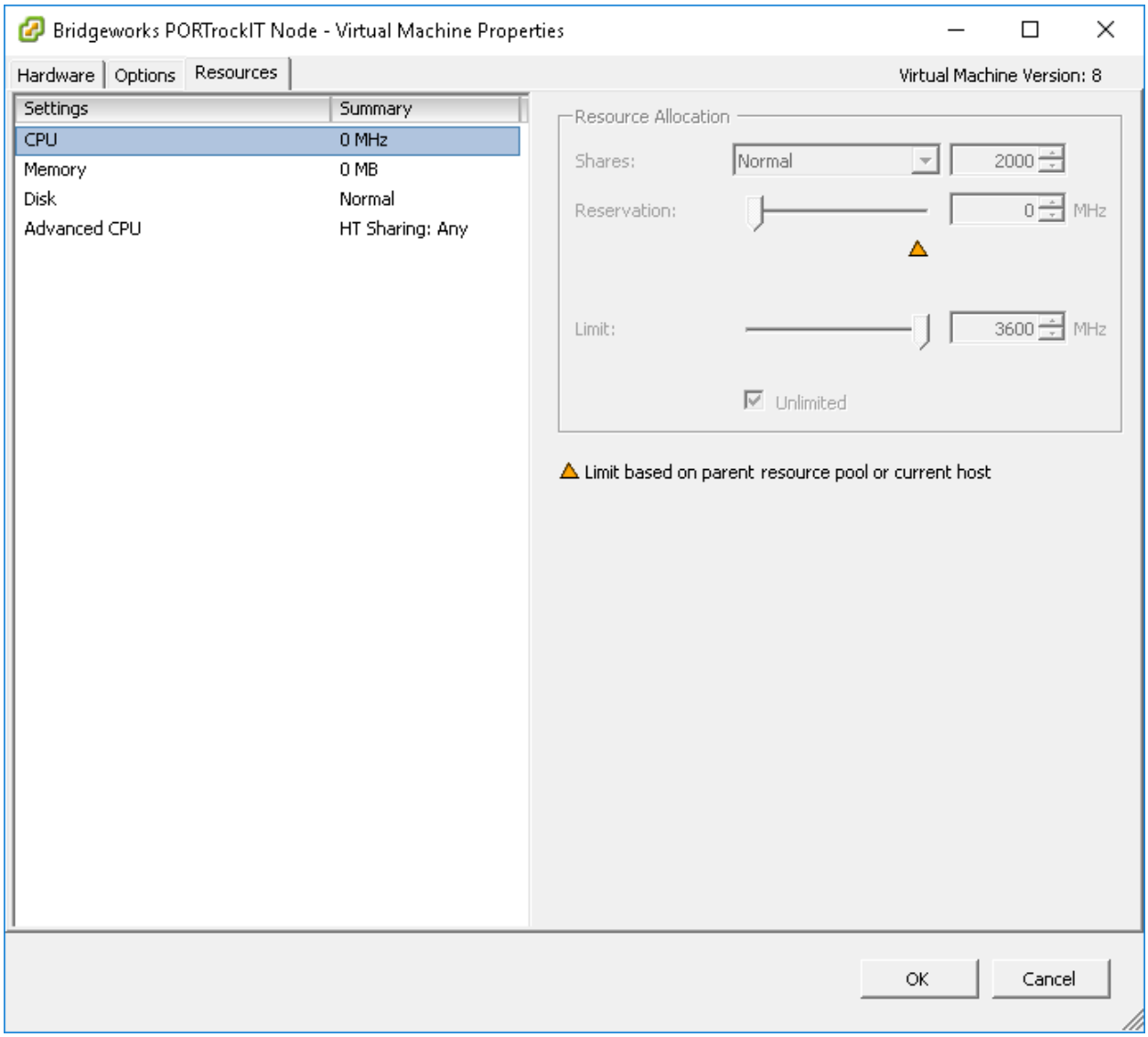
The same considerations as the memory allocation section apply here. If the ESXi host does not have enough memory in a single NUMA node (see Section 2.4: Memory (NUMA - Non-Uniform Memory Access) Alignment:) for the Bridgeworks Node,then memory misalignment can occur.

Consider shrinking the memory allocation, or reordering the virtual guests on the host, in order to ensure that all of the Bridgeworks memory allocation exists on physical RAM that is local to the CPU cores used by the Bridgeworks Node.

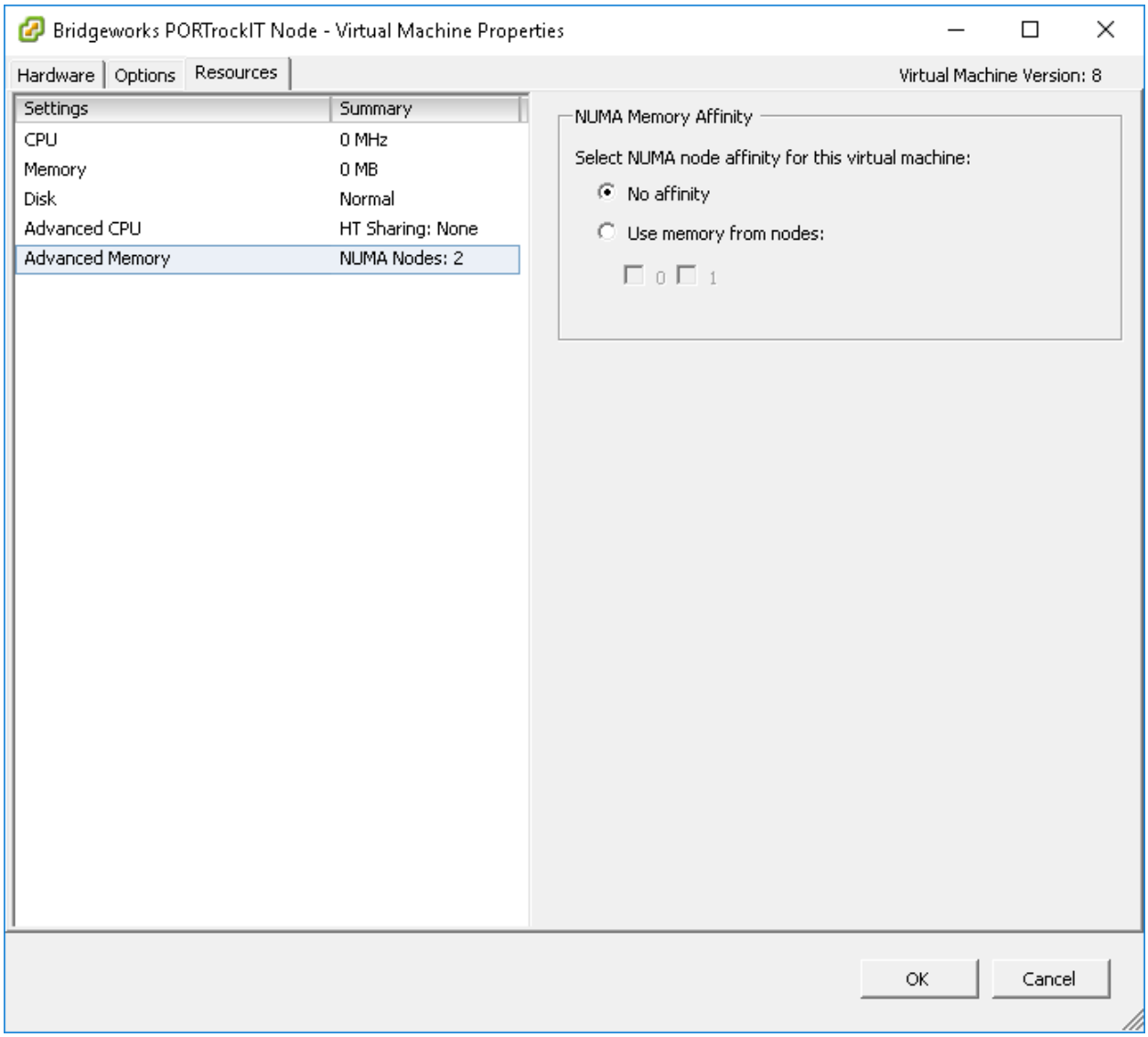
7.1 Setting Alignment

7.1.1 vSphere

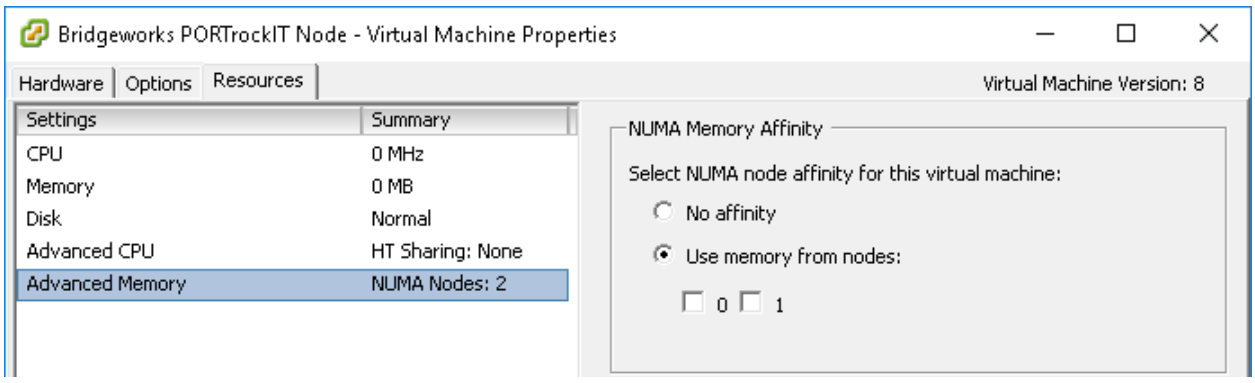
In the *Edit Settings* pane, navigate to the *Resources* tab located near the top of the pane.



From here select the *Advanced Memory* entry from the list presented.



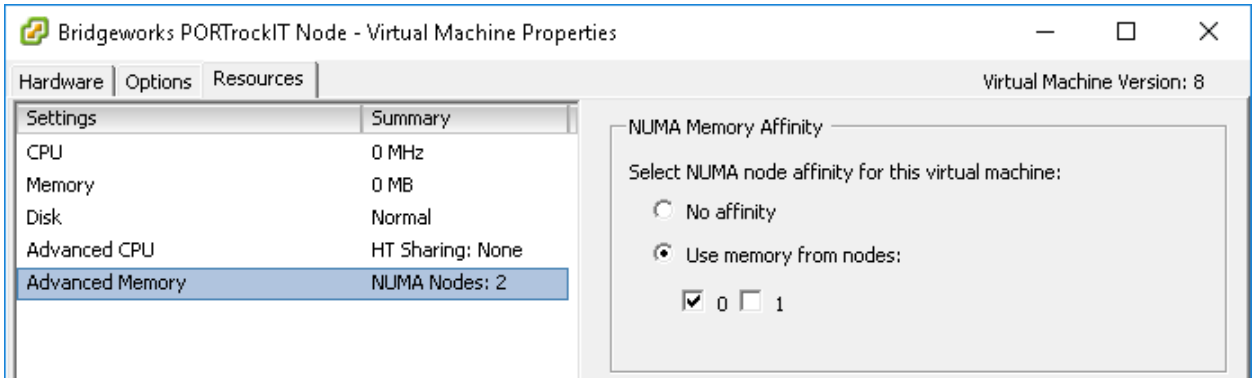
The right side of the pane is now displaying *NUMA Memory Affinity*, select the option labelled *Use memory from nodes:*.



The next step is to map the correct memory module. Each checkbox represents the physical memory attached to each socket. The example images depict an ESXi host with 2 sockets, therefore 2 memory nodes are available.

This option needs to line up with the CPU affinity that has been set. In this example the CPU affinity

for the Bridgeworks Node has been set to run on only cores from the first socket.



Summary of example:

- An ESXi host has 2 sockets, each with 4 cores and Hyperthreading enabled. A total of 16 logical cores on host.
- The Bridgeworks Node has been given CPU affinity to run on the *last* 2 physical cores of the *first* socket. The CPU affinity could be set to any of the following; ”4,6”, ”5,6”, ”4,7”, ”5,7”. Any of these values select 2 logical cores on 2 different physical cores.
- Since the Bridgeworks Node in this example is on the first socket, the *NUMA Memory Affinity* would be the checkbox labelled *0*.