

ESXi Resource Management

Focus on Resource Distribution

Valentin Bondzio

Sr. Staff TSE / GSS Premier Services

2020-01-09

Agenda

Distributed Resources

vCenter Level View

Host Resources

ESXi Level View

Consumers

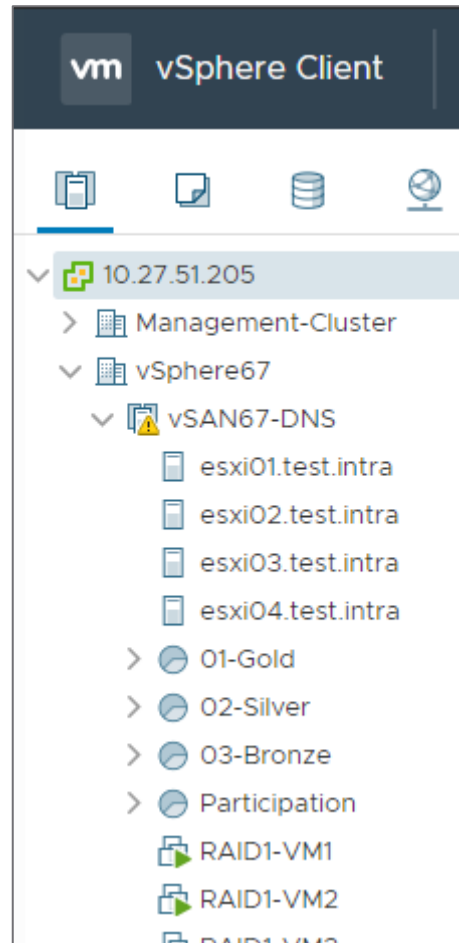
VMs

Consumers part 2

everything else

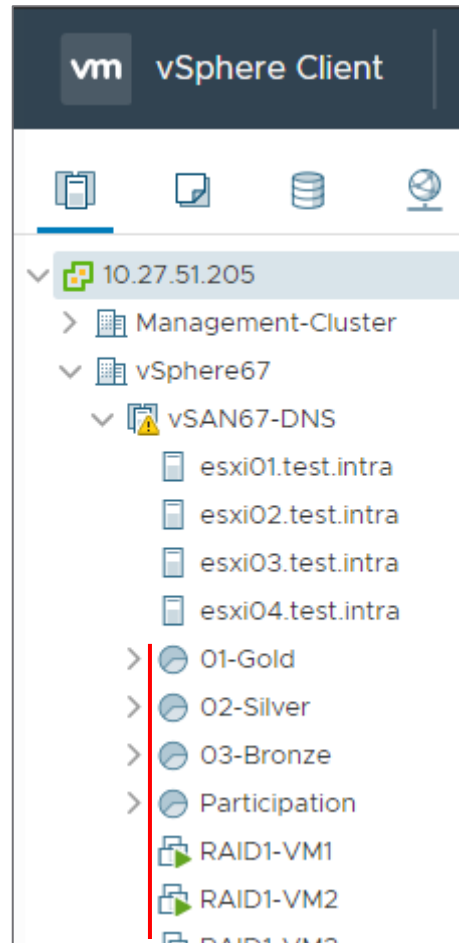
Resource Pools

Cluster Level



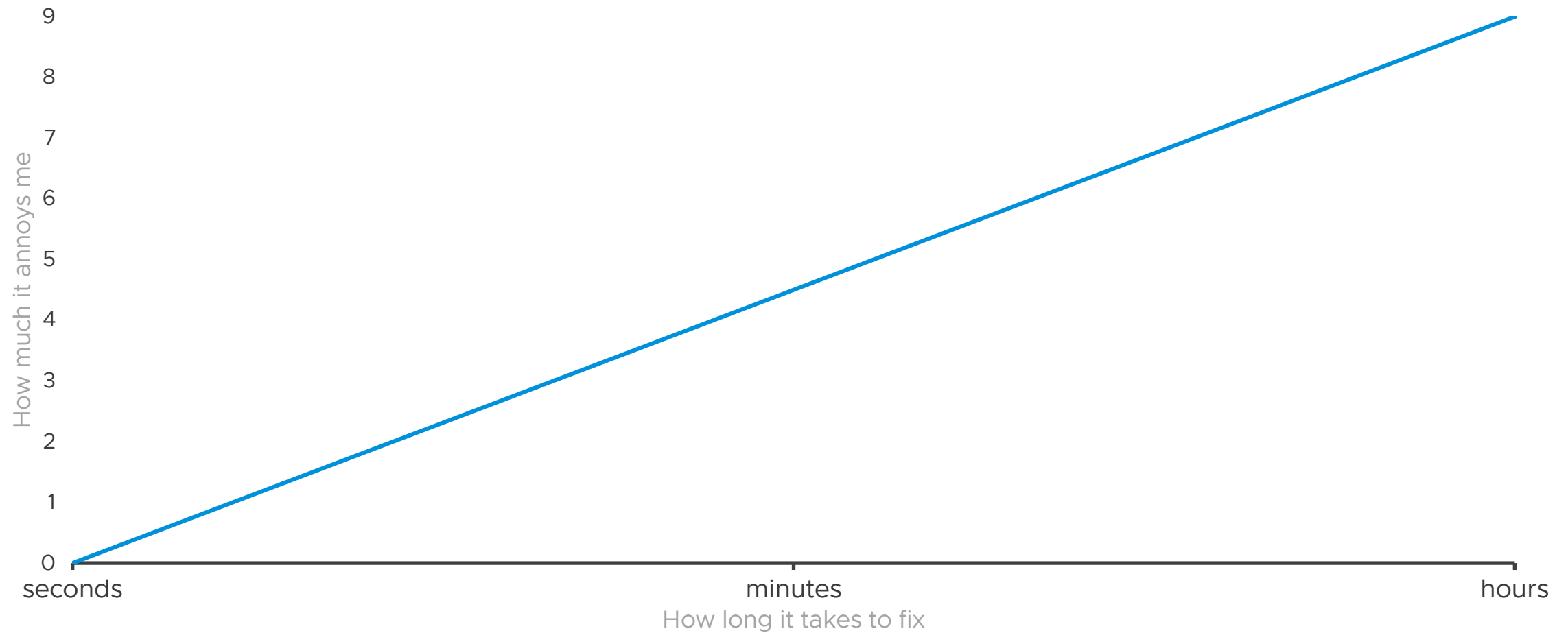
Resource Pools

Cluster Level



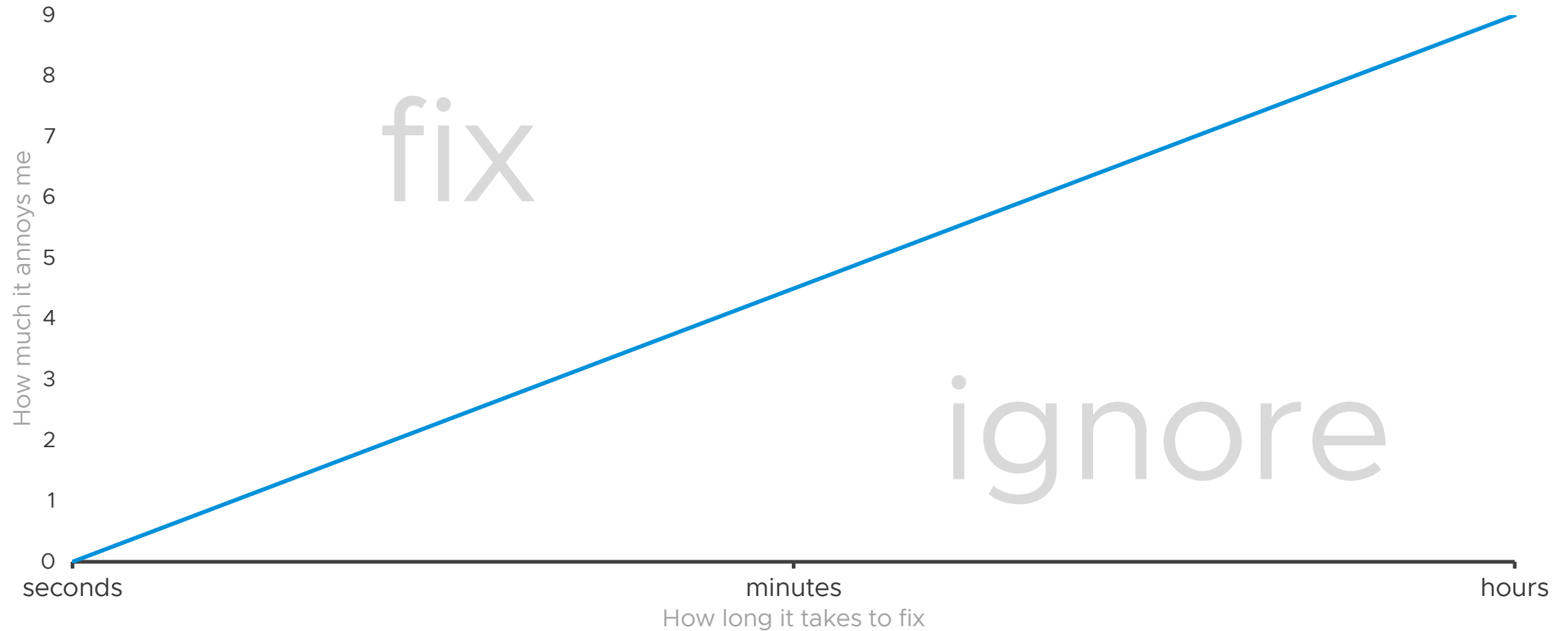
That warning triangle on the cluster ...

Yes it annoys me too



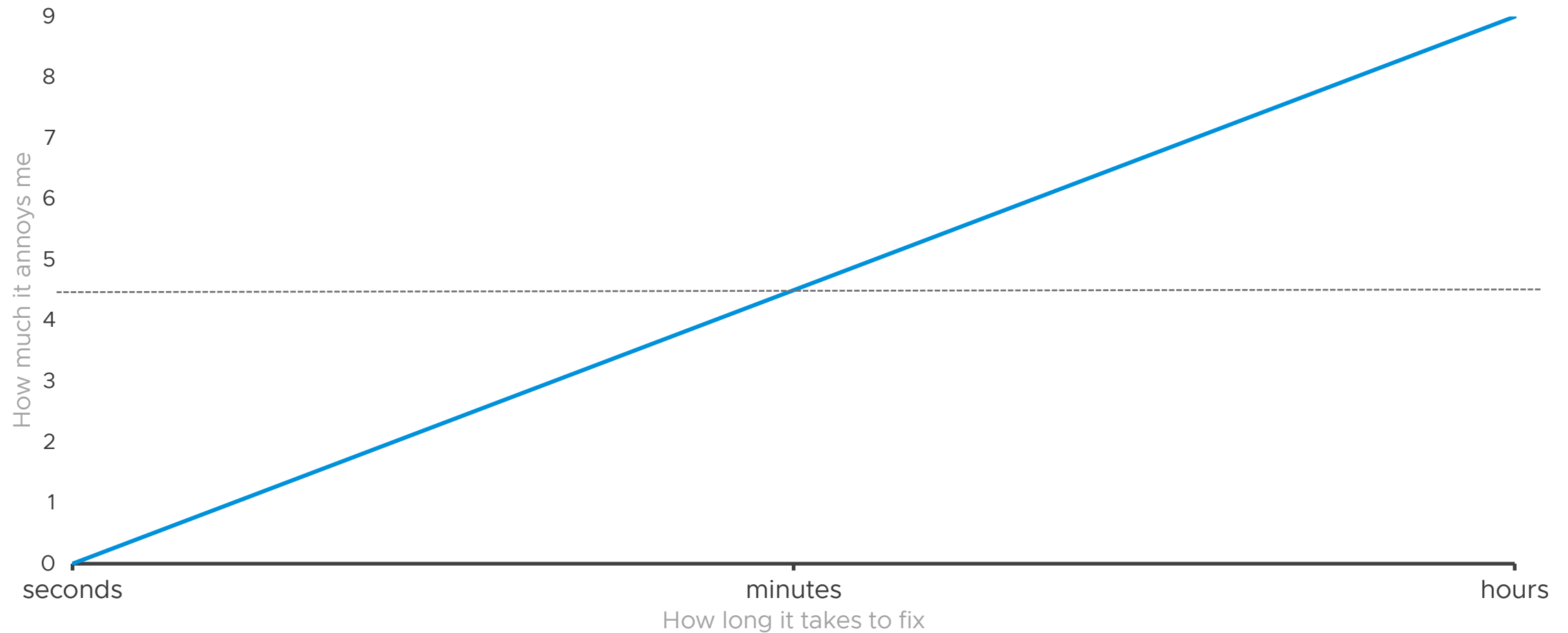
That warning triangle on the cluster ...

Yes it annoys me too



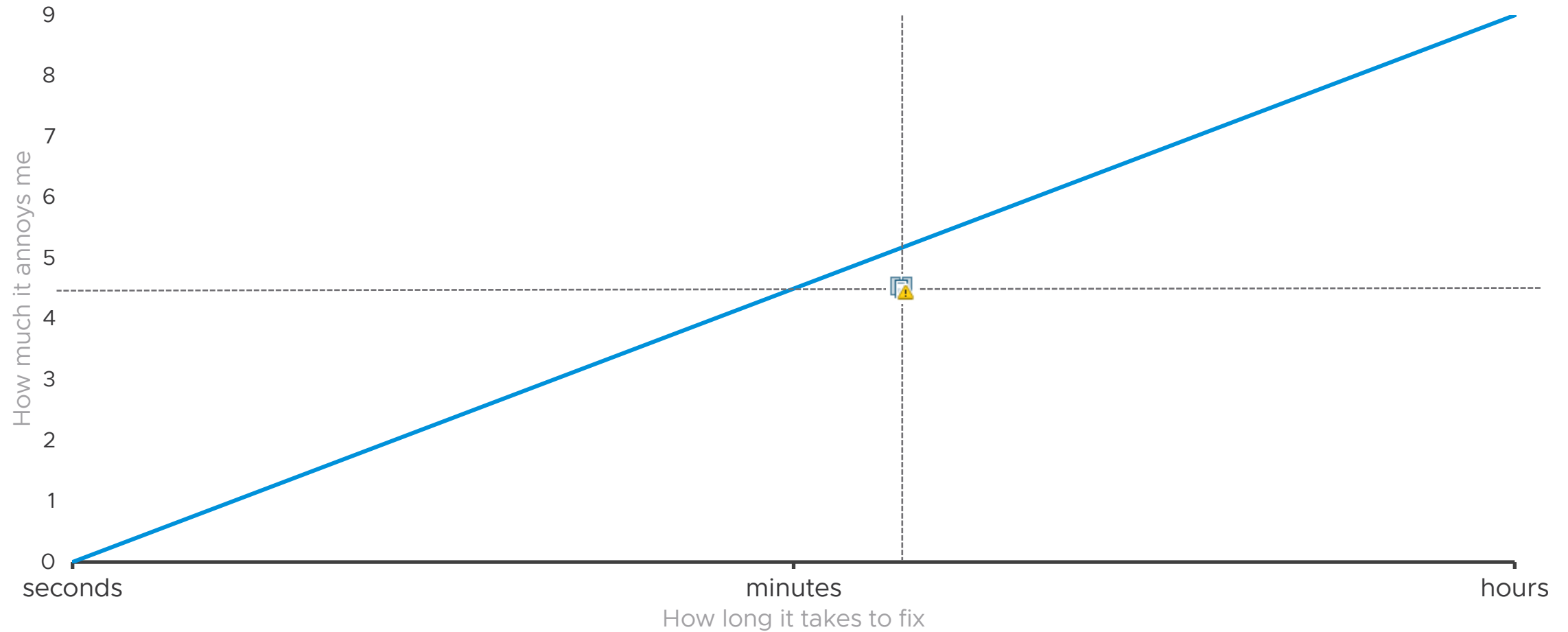
That warning triangle on the cluster ...

Yes it annoys me too



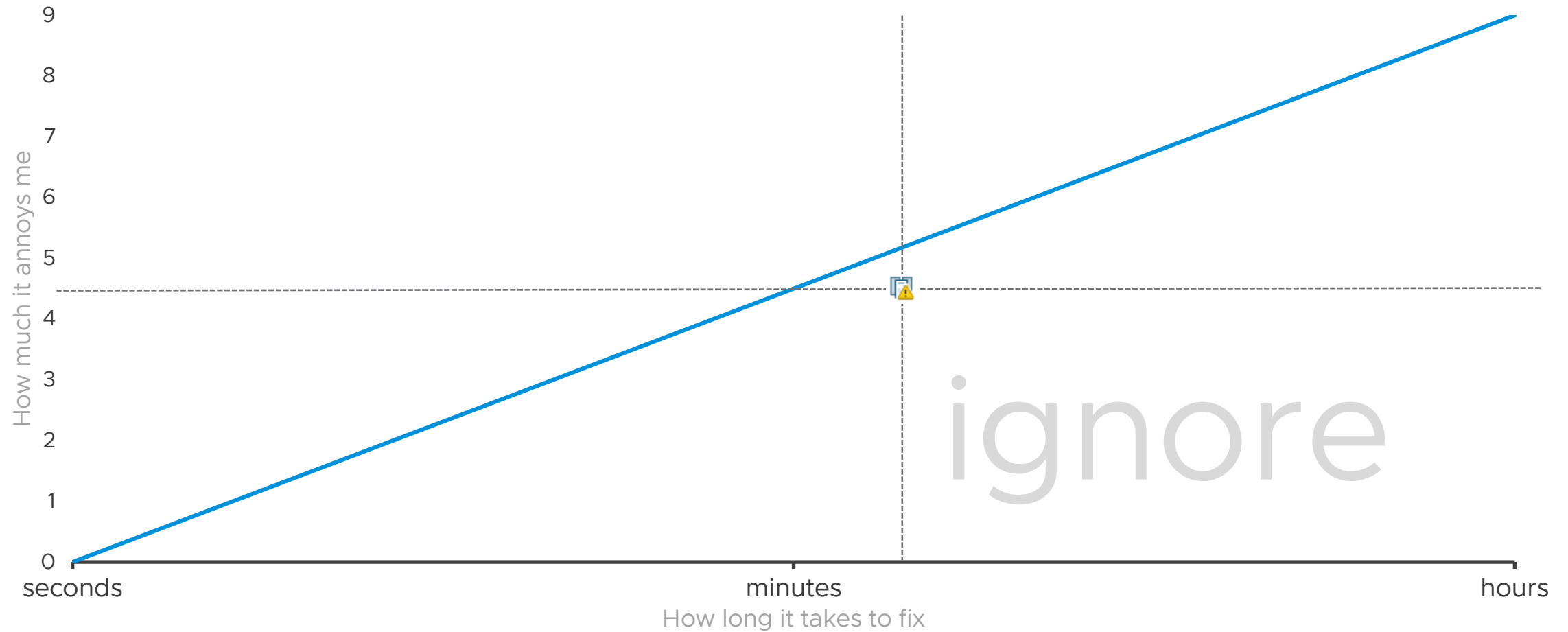
That warning triangle on the cluster ...

Yes it annoys me too



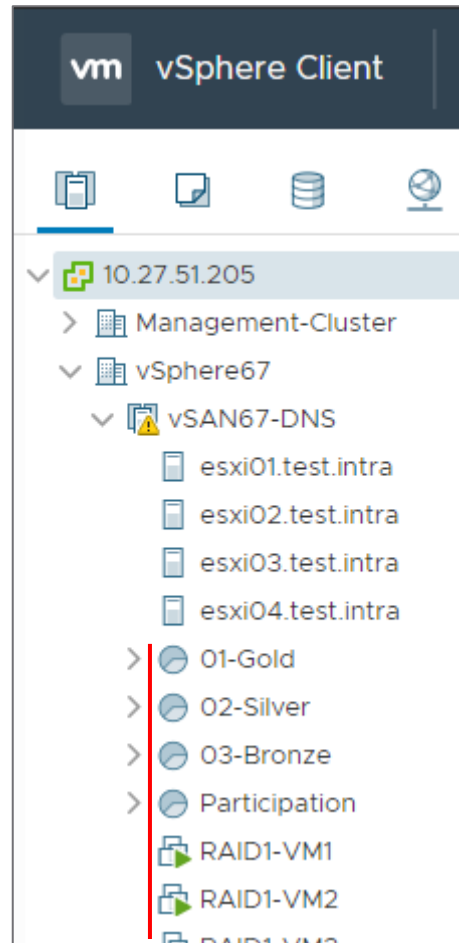
That warning triangle on the cluster ...

Yes it annoys me too



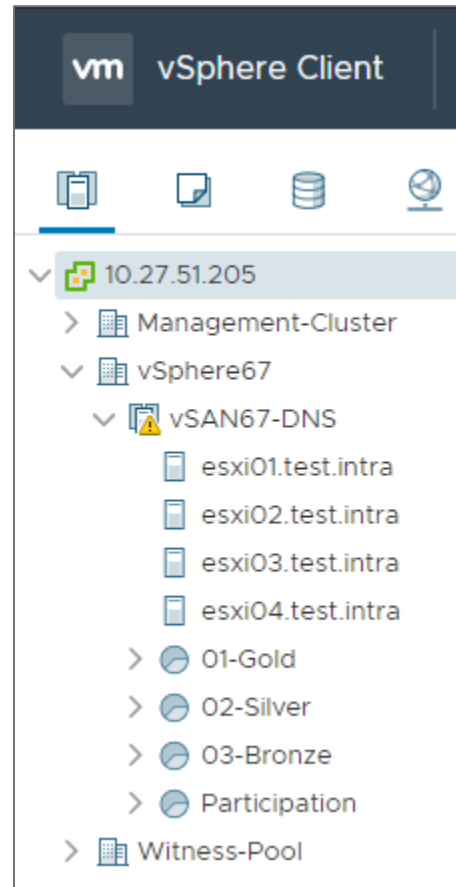
Resource Pools

Cluster Level*



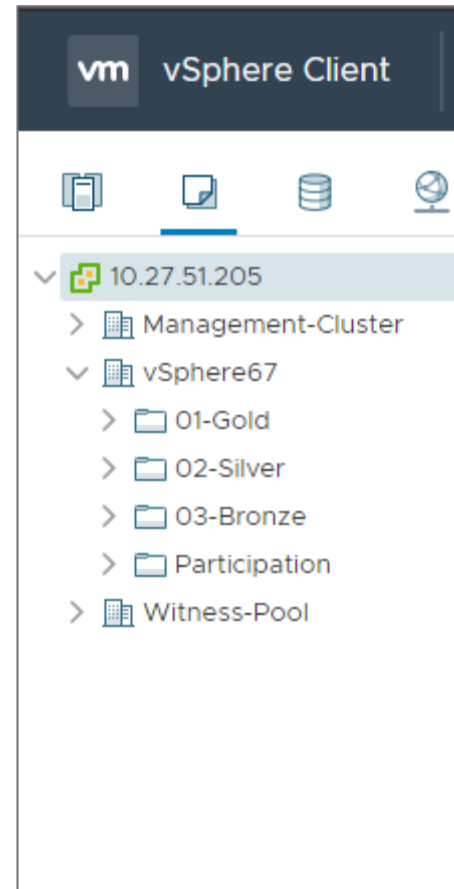
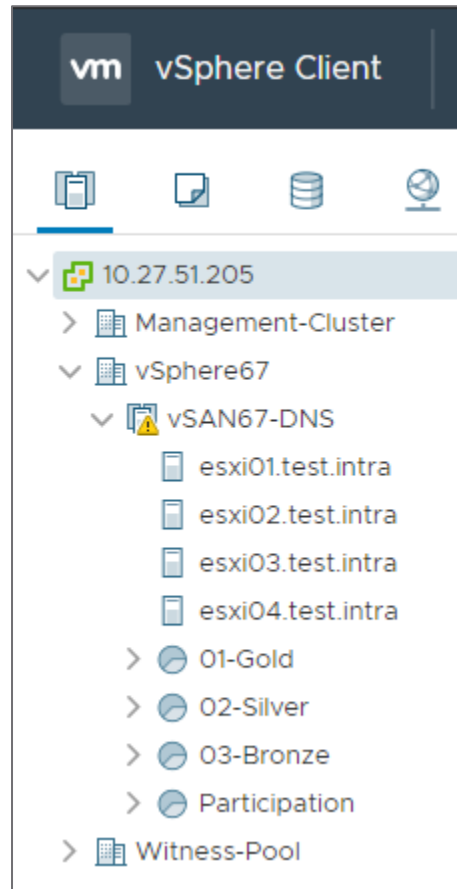
Resource Pools

Cluster Level*



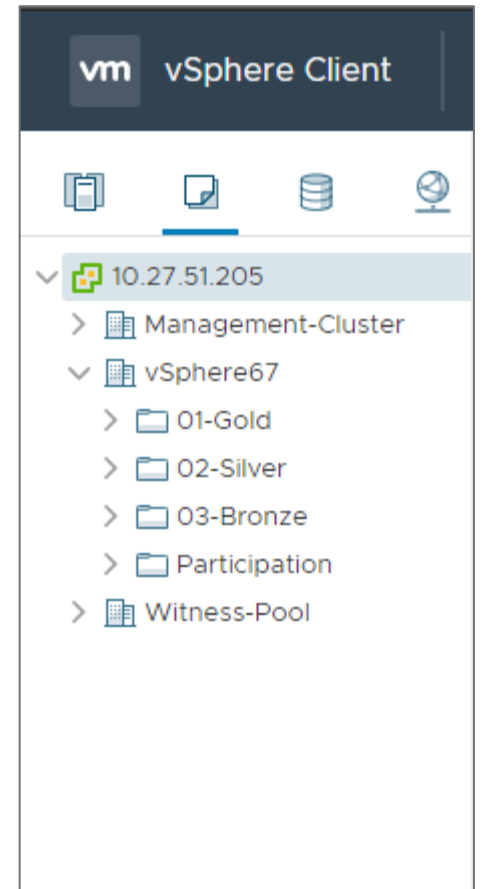
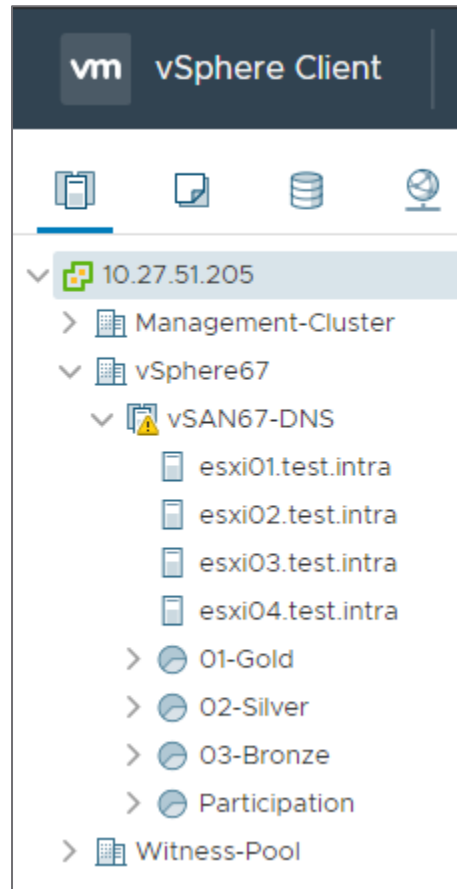
Resource Pools

Cluster Level*



Resource Pools

Cluster Level*

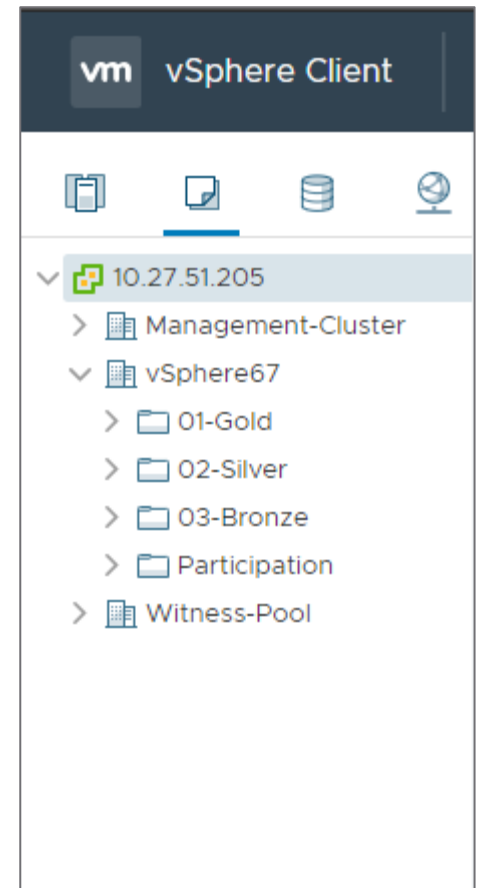
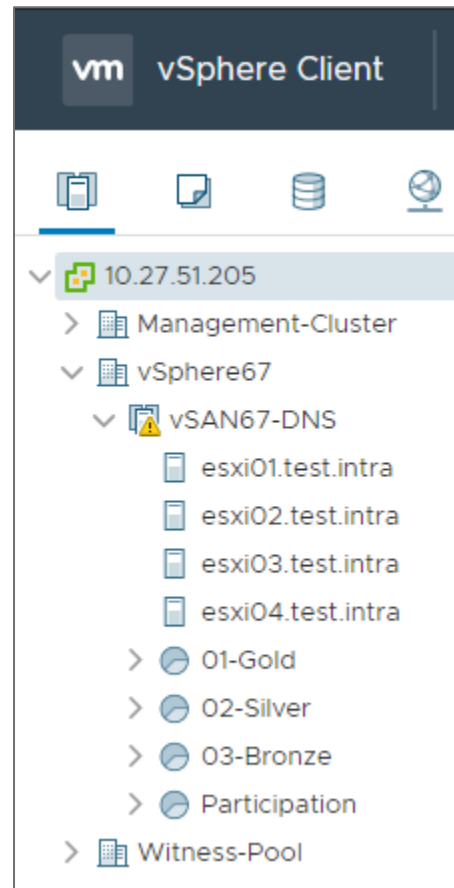


Resource Pools

Cluster Level*

Resource Pools

- Distribute
- Prioritize
- Guarantee
- Limit

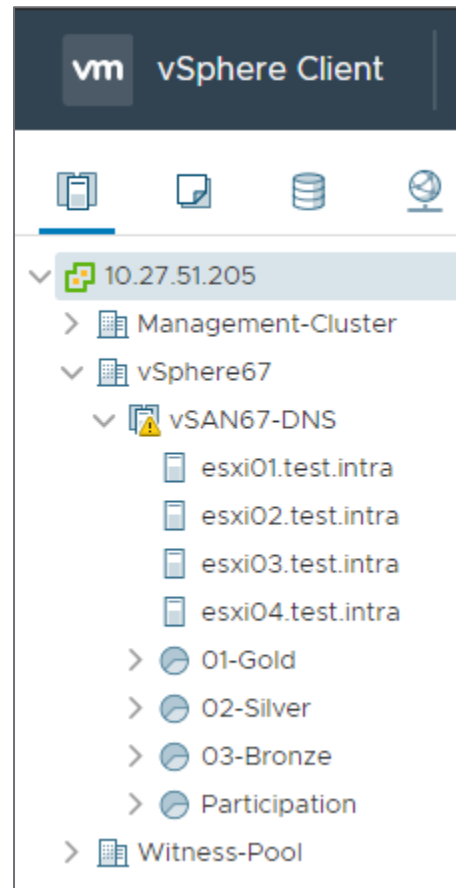


Resource Pools

Cluster Level*

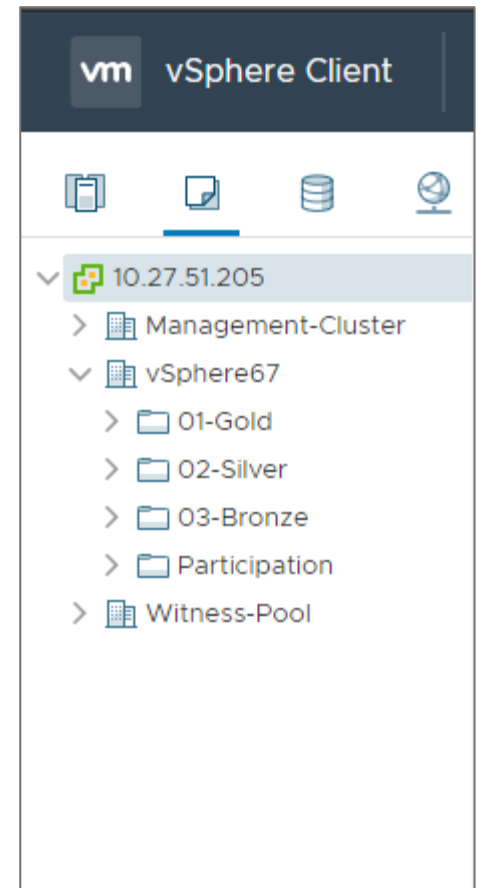
Resource Pools

- Distribute
- Prioritize
- Guarantee
- Limit



Folders

- Structure

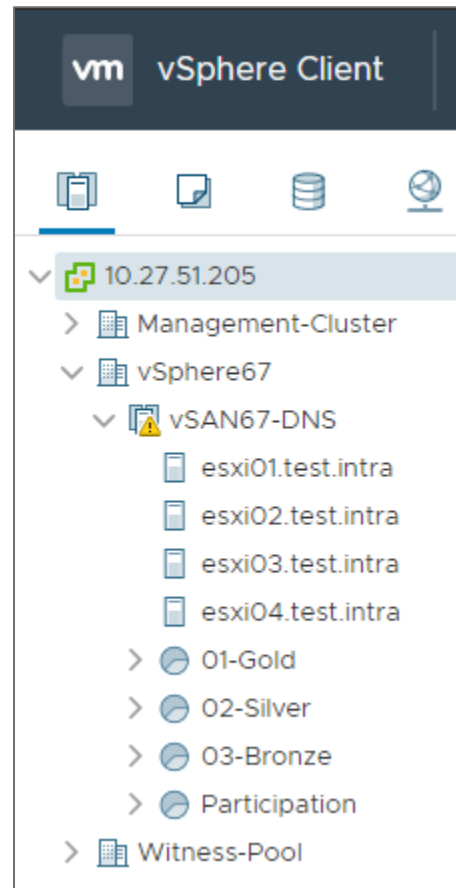


Resource Pools

Cluster Level*

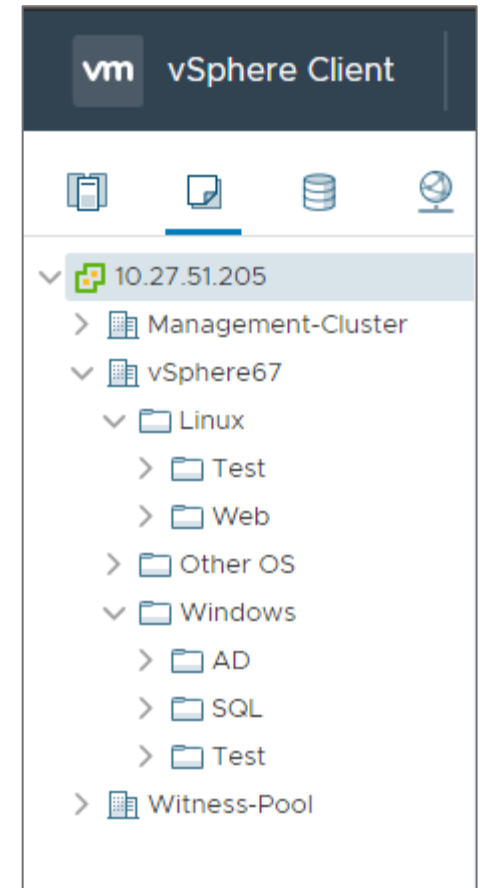
Resource Pools

- Distribute
- Prioritize
- Guarantee
- Limit



Folders

- Structure



Resource Pools

So what if I don't care about resources and just want some structure?

Resource Pools

So what if I don't care about resources and just want some structure?



- ▼ ● Linux
 - > ● Test
 - > ● Web
 - > ● Other OS
- ▼ ● Windows
 - > ● AD
 - > ● SQL
 - > ● Test

Resource Pools

So what if I don't care about resources and just want some structure?



- ▼ ● Linux
 - > ● Test
 - > ● Web
 - > ● Other OS
- ▼ ● Windows
 - > ● AD
 - > ● SQL
 - > ● Test



- ▼ □ Linux
 - > □ Test
 - > □ Web
 - > □ Other OS
- ▼ □ Windows
 - > □ AD
 - > □ SQL
 - > □ Test

Resource Pools

So what if I don't care about resources and just want some structure?



- ▼ ● Linux
 - > ● Test
 - > ● Web
 - > ● Other OS
- ▼ ● Windows
 - > ● AD
 - > ● SQL
 - > ● Test



- ▼ □ Linux *
 - > □ Test
 - > □ Web
 - > □ Other OS
- ▼ □ Windows
 - > □ AD
 - > □ SQL
 - > □ Test

* don't forget about tags

Resource Pools

Creation and Maximums

New Resource Pool | vSAN67-DNS

Name	New Resource Pool	
v CPU		
Shares	Normal	4000
Reservation	0	MHz
Max reservation: 99,504 MHz		
Reservation Type	<input checked="" type="checkbox"/> Expandable	
Limit	Unlimited	MHz
Max limit: 132,672 MHz		
v Memory		

New Resource Pool | vSAN67-DNS

Name	New Resource Pool	
v CPU		
v Memory		
Shares	Normal	163840
Reservation	0	MB
Max reservation: 214,563 MB		
Reservation Type	<input checked="" type="checkbox"/> Expandable	
Limit	Unlimited	MB
Max limit: 309,470 MB		

Resource Pools

Creation and Maximums

New Resource Pool | vSAN67-DNS

Name	New Resource Pool	
▼ CPU		
Shares	Normal	4000
Reservation	0	MHz
	Max reservation: 99,504 MHz	
Reservation Type	<input checked="" type="checkbox"/> Expandable	
Limit	Unlimited	MHz
	Max limit: 132,672 MHz	
> Memory		

New Resource Pool | vSAN67-DNS

Name	New Resource Pool	
> CPU		
▼ Memory		
Shares	Normal	163840
Reservation	0	MB
	Max reservation: 214,559 MB	
Reservation Type	<input checked="" type="checkbox"/> Expandable	
Limit	Unlimited	MB
	Max limit: 309,465 MB	

Clarity Dark Theme

Official CSS / [unsupported](#) extension

The screenshot displays the vSphere Client interface in a dark theme. The top navigation bar includes the VMware logo, 'vSphere Client', a 'Menu' dropdown, a search bar, a refresh icon, a help icon, and the user 'Administrator@VSPHERE.LOCAL'. The left sidebar shows a tree view of the environment, with the host '10.27.51.205' selected. The main content area shows the host's summary, including a green VMware logo, the host name '10.27.51.205', and an 'ACTIONS' dropdown. Below this, there are tabs for 'Summary', 'Monitor', 'Configure', 'Permissions', 'Datacenters', 'Hosts & Clusters', 'VMs', and 'Datasto'. The 'Summary' tab is active, displaying a large green VMware logo and the following statistics: 'Virtual Machines: 17' and 'Hosts: 6'. To the right, there are three resource usage bars: CPU (Used: 1.17 GHz, Free: 207.46 GHz, Capacity: 208.62 GHz), Memory (Used: 87.26 GB, Free: 552.08 GB, Capacity: 639.34 GB), and Storage (Used: 138.4 GB, Free: 8.04 TB, Capacity: 8.18 TB). At the bottom, there are two expandable sections: 'Custom Attributes' with a table with columns 'Attribute' and 'Value', and 'Tags' with a table with columns 'Assigned Tag', 'Category', and 'Description'.

Use at your own risk: <https://github.com/BeryJu/dark-vcenter>

Clarity Dark Theme

Fully Supported in 6.7 U1

New in 6.7 U1

vm vSphere Client | Menu | Search | Administrator@VSPHERE.LOCAL

10.27.51.205 | ACTIONS

Summary | Monitor | Configure | Permissions | Datacenters | Hosts & Clusters | VMs | Datasto

Virtual Machines: 17
Hosts: 6

CPU: Free: 207.46 GHz, Used: 1.17 GHz, Capacity: 208.62 GHz
Memory: Free: 552.08 GB, Used: 87.26 GB, Capacity: 639.34 GB
Storage: Free: 8.04 TB, Used: 138.4 GB, Capacity: 8.18 TB

Attribute	Value
-----------	-------

Assigned Tag	Category	Description
--------------	----------	-------------

Clarity Dark Theme

Fully Supported in 6.7 U1

New in 6.7 U1

The screenshot displays the vSphere Client interface in the Clarity Dark Theme. The top navigation bar includes the VMware logo, 'vSphere Client', a 'Menu' dropdown, a search bar, a refresh icon, a help icon, and the user 'Administrator@VSPHERE.LOCAL'. The left navigation pane shows a tree view with the following structure:

- 10.27.51.205
 - Management-Cluster
 - Intel-PoC
 - vSphere67
 - vSAN67-DNS
 - esxi01.test.intra
 - esxi02.test.intra
 - esxi03.test.intra
 - esxi04.test.intra
 - 01-Gold
 - 02-Silver
 - 03-Bronze
 - Participation
 - Witness-Pool

The main content area shows the 'Summary' tab for the host '10.27.51.205'. The summary includes a large green icon, statistics for Virtual Machines (17) and Hosts (6), and resource usage bars for CPU, Memory, and Storage.

CPU
Free: 207.46 GHz
Used: 1.16 GHz
Capacity: 208.62 GHz

Memory
Free: 552.07 GB
Used: 87.27 GB
Capacity: 639.34 GB

Storage
Free: 8.04 TB
Used: 138.4 GB
Capacity: 8.18 TB

Custom Attributes

Attribute	Value
-----------	-------

Tags

Assigned Tag	Category	Description
--------------	----------	-------------

Resource Pools

Creation and Maximums

New Resource Pool | vSAN67-DNS

Name	New Resource Pool	
▼ CPU		
Shares	Normal ▼	4000
Reservation	0 ▼	MHz ▼
Max reservation: 99,504 MHz		
Reservation Type	<input checked="" type="checkbox"/> Expandable	
Limit	Unlimited ▼	MHz ▼
Max limit: 132,672 MHz		
▶ Memory		

New Resource Pool | vSAN67-DNS

Name	New Resource Pool	
▶ CPU		
▼ Memory		
Shares	Normal ▼	163840
Reservation	0 ▼	MB ▼
Max reservation: 214,563 MB		
Reservation Type	<input checked="" type="checkbox"/> Expandable	
Limit	Unlimited ▼	MB ▼
Max limit: 309,470 MB		

Resource Pools

Creation and Maximums

New Resource Pool | vSAN67-DNS

Name	New Resource Pool	
CPU		
Shares	Normal	4000
Reservation	0	MHz
Max reservation: 99,504 MHz		
Reservation Type	<input checked="" type="checkbox"/> Expandable	
Limit	Unlimited	MHz
Max limit: 132,672 MHz		
Memory		

vSAN67-DNS | ACTIONS

Summary | Monitor | Configure | Permissions | Hosts | VMs | Datastor

- Issues and Alarms
- Performance
- Tasks and Events
- vSphere DRS
- vSphere HA
- Resource Allocation
 - CPU
 - Memory
 - Storage
 - Utilization
 - Storage Overview
 - Security
 - vSAN
 - Health
 - Virtual Objects
 - Physical Disks

CPU Reservation Details

0 GHz | 132.67 GHz

Cluster Total Capacity: 153.57 GHz
Total Reservation Capacity: 132.67 GHz
Used Reservation: 33.17 GHz
Available Reservation: 99.5 GHz

Name ↑	Reservation (MHz)	Limit (MHz)
01-Gold	0	Unlimited
02-Silver	0	Unlimited
03-Bronze	0	Unlimited
Participation	0	Unlimited

Resource Pools

Creation and Maximums

New Resource Pool | vSAN67-DNS

Name: New Resource Pool

▼ CPU

Shares: Normal | 4000

Reservation: 0 MHz
Max reservation: 99,504 MHz

Reservation Type: Expandable

Limit: Unlimited MHz
Max limit: 132,672 MHz

> Memory

vSAN67-DNS | ACTIONS

Summary | **Monitor** | Configure | Permissions | Hosts | VMs | Datastor

- Issues and Alarms
- Performance
- Tasks and Events
- vSphere DRS
- vSphere HA
- ▼ Resource Allocation
 - CPU**
 - Memory
 - Storage
 - Utilization
 - Storage Overview
 - Security
 - ▼ vSAN
 - Health
 - Virtual Objects
 - Physical Disks

CPU Reservation Details

0 GHz | 132.67 GHz

Cluster Total Capacity: 153.57 GHz
Total Reservation Capacity: 132.67 GHz

Used Reservation: 33.17 GHz
Available Reservation: 99.5 GHz

Name ↑	Reservation (MHz)	Limit (MHz)
01-Gold	0	Unlimited
02-Silver	0	Unlimited
03-Bronze	0	Unlimited
Participation	0	Unlimited

Resource Pools

Creation and Maximums

New Resource Pool | vSAN67-DNS

Name	New Resource Pool	
CPU		
Shares	Normal	4000
Reservation	0	MHz
Max reservation: 99,504 MHz		
Reservation Type	<input checked="" type="checkbox"/> Expandable	
Limit	Unlimited	MHz
Max limit: 132,672 MHz		
Memory		

vSAN67-DNS | ACTIONS

Summary | Monitor | Configure | Permissions | Hosts | VMs | Datastor

- Issues and Alarms
- Performance
- Tasks and Events
- vSphere DRS
- vSphere HA
- Resource Allocation
 - CPU
 - Memory
 - Storage
- Utilization
- Storage Overview
- Security
- vSAN
 - Health
 - Virtual Objects
 - Physical Disks

CPU Reservation Details

0 GHz | 132.67 GHz

Cluster Total Capacity	153.57 GHz
Total Reservation Capacity	132.67 GHz
Used Reservation	33.17 GHz
Available Reservation	99.5 GHz

Name ↑	Reservation (MHz)	Limit (MHz)
01-Gold	0	Unlimited
02-Silver	0	Unlimited
03-Bronze	0	Unlimited
Participation	0	Unlimited

Resource Pools

What is consuming reservations? (CPU)

The screenshot shows the vSphere Web Client interface for vSAN67-DNS. The 'Monitor' tab is active, and the 'CPU' resource allocation section is selected in the left-hand navigation pane. The main content area displays 'CPU Reservation Details' with a progress bar showing 0 GHz used out of a 132.67 GHz total reservation capacity. Below the bar, a table lists reservation statistics: Cluster Total Capacity (153.57 GHz), Total Reservation Capacity (132.67 GHz), Used Reservation (33.17 GHz), and Available Reservation (99.5 GHz). At the bottom, a table lists resource pools with their respective reservation and limit values.

Name ↑	Reservation (MHz)	Limit (MHz)
01-Gold	0	Unlimited
02-Silver	0	Unlimited
03-Bronze	0	Unlimited
Participation	0	Unlimited

Resource Pools

What is consuming reservations? (CPU)

vSAN67-DNS | ACTIONS ▾

Summary | **Monitor** | Configure | Permissions | Hosts | VMs | Datastore

▶ Issues and Alarms
▶ Performance
▶ Tasks and Events
▶ vSphere DRS
▶ vSphere HA
▼ Resource Allocation

- CPU**
- Memory
- Storage
- Utilization
- Storage Overview
- Security
- ▼ vSAN
 - Health
 - Virtual Objects
 - Physical Disks

CPU Reservation Details

0 GHz | 132.67 GHz

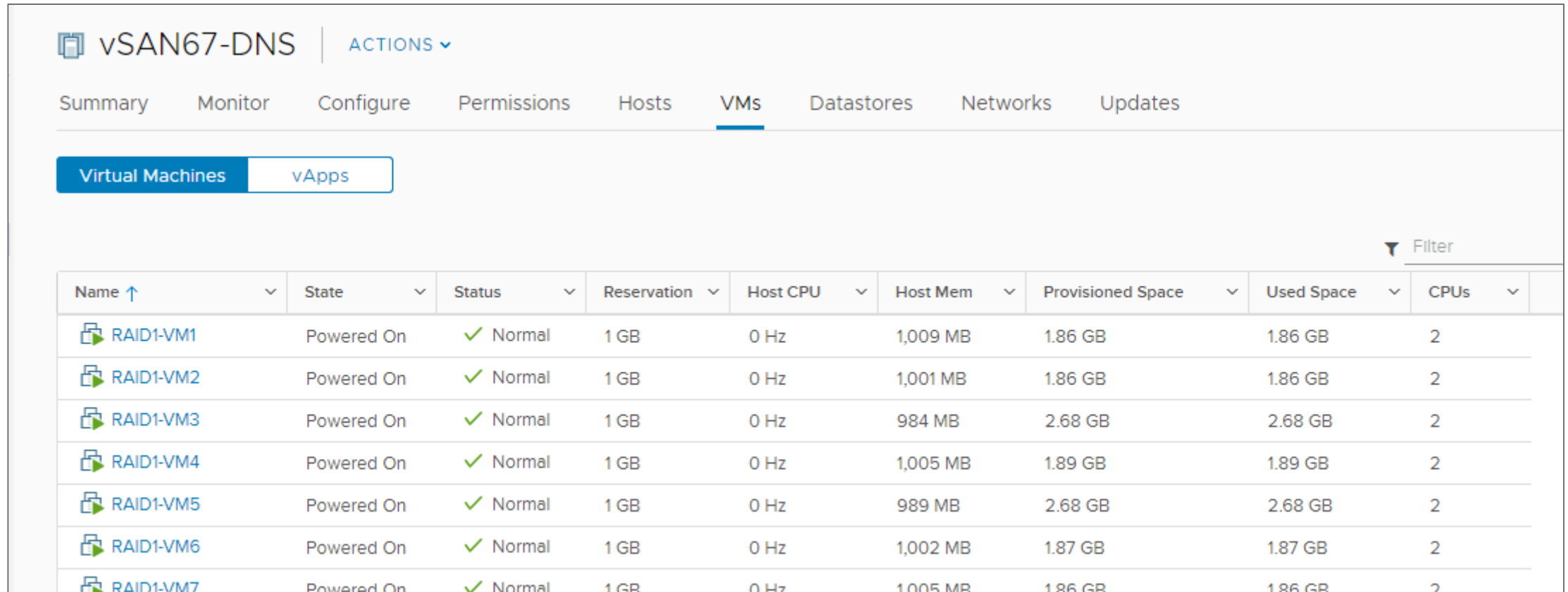
Cluster Total Capacity: 153.57 GHz
Total Reservation Capacity: 132.67 GHz

Used Reservation: 33.17 GHz
 Available Reservation: 99.5 GHz

Name ↑	Reservation (MHz)	Limit (MHz)
01-Gold	0	Unlimited
02-Silver	0	Unlimited
03-Bronze	0	Unlimited
Participation	0	Unlimited

Resource Pools

What is consuming reservations? (CPU)

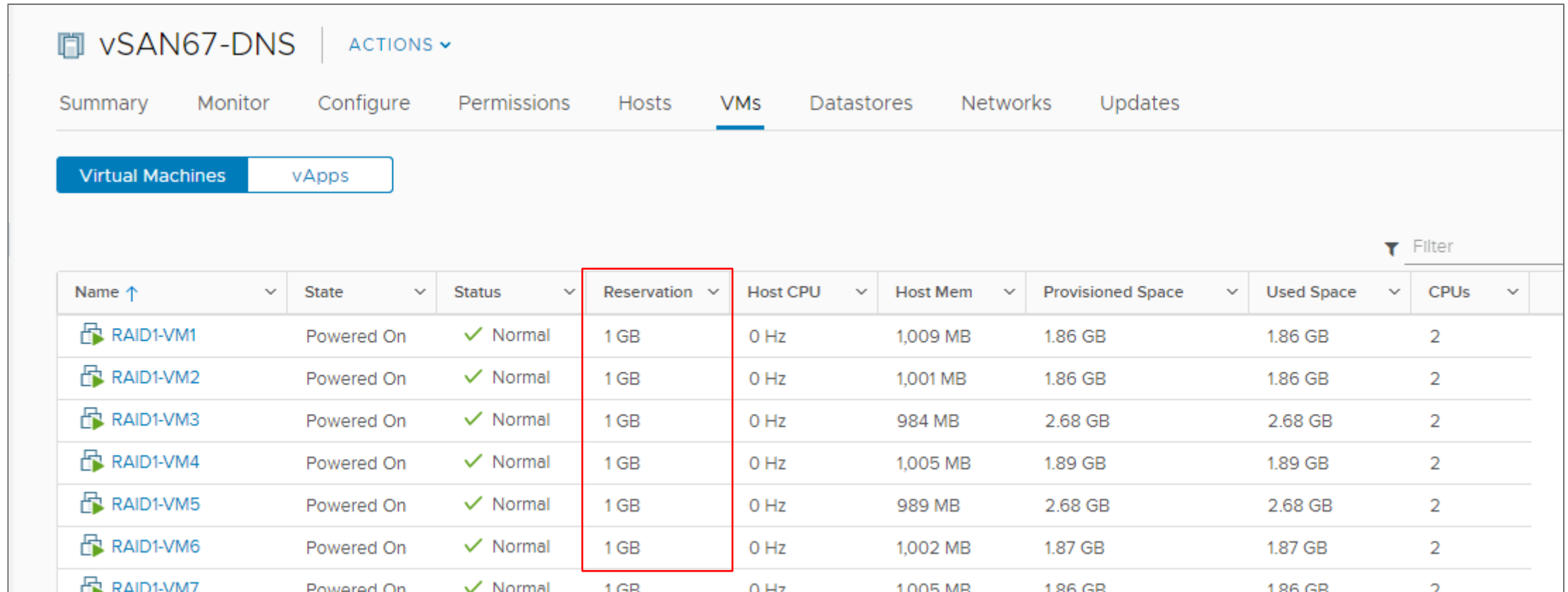


The screenshot shows the vSAN67-DNS VMS tab in a VMware interface. It displays a table of virtual machines with the following columns: Name, State, Status, Reservation, Host CPU, Host Mem, Provisioned Space, Used Space, and CPUs. The table lists seven RAID1-VMs, all of which are Powered On and in a Normal status. Each VM has a 1 GB reservation and 2 CPUs. The Host CPU usage is 0 Hz for all. The Host Mem usage varies, and the Provisioned Space and Used Space are also listed for each VM.

Name ↑	State	Status	Reservation	Host CPU	Host Mem	Provisioned Space	Used Space	CPUs
RAID1-VM1	Powered On	✓ Normal	1 GB	0 Hz	1,009 MB	1.86 GB	1.86 GB	2
RAID1-VM2	Powered On	✓ Normal	1 GB	0 Hz	1,001 MB	1.86 GB	1.86 GB	2
RAID1-VM3	Powered On	✓ Normal	1 GB	0 Hz	984 MB	2.68 GB	2.68 GB	2
RAID1-VM4	Powered On	✓ Normal	1 GB	0 Hz	1,005 MB	1.89 GB	1.89 GB	2
RAID1-VM5	Powered On	✓ Normal	1 GB	0 Hz	989 MB	2.68 GB	2.68 GB	2
RAID1-VM6	Powered On	✓ Normal	1 GB	0 Hz	1,002 MB	1.87 GB	1.87 GB	2
RAID1-VM7	Powered On	✓ Normal	1 GB	0 Hz	1,005 MB	1.86 GB	1.86 GB	2

Resource Pools

What is consuming reservations? (CPU)



The screenshot shows the vSAN67-DNS interface with the 'VMs' tab selected. A table lists seven virtual machines, all with a reservation of 1 GB. A red box highlights the 'Reservation' column for all rows.

Name ↑	State	Status	Reservation	Host CPU	Host Mem	Provisioned Space	Used Space	CPUs
RAID1-VM1	Powered On	✓ Normal	1 GB	0 Hz	1,009 MB	1.86 GB	1.86 GB	2
RAID1-VM2	Powered On	✓ Normal	1 GB	0 Hz	1,001 MB	1.86 GB	1.86 GB	2
RAID1-VM3	Powered On	✓ Normal	1 GB	0 Hz	984 MB	2.68 GB	2.68 GB	2
RAID1-VM4	Powered On	✓ Normal	1 GB	0 Hz	1,005 MB	1.89 GB	1.89 GB	2
RAID1-VM5	Powered On	✓ Normal	1 GB	0 Hz	989 MB	2.68 GB	2.68 GB	2
RAID1-VM6	Powered On	✓ Normal	1 GB	0 Hz	1,002 MB	1.87 GB	1.87 GB	2
RAID1-VM7	Powered On	✓ Normal	1 GB	0 Hz	1,005 MB	1.86 GB	1.86 GB	2


Resource Pools

What is consuming reservations? (CPU)








vSAN67-DNS | ACTIONS ▾

Summary Monitor Configure Permissions Hosts **VMs** Datastores Networks Updates

Virtual Machines vApps

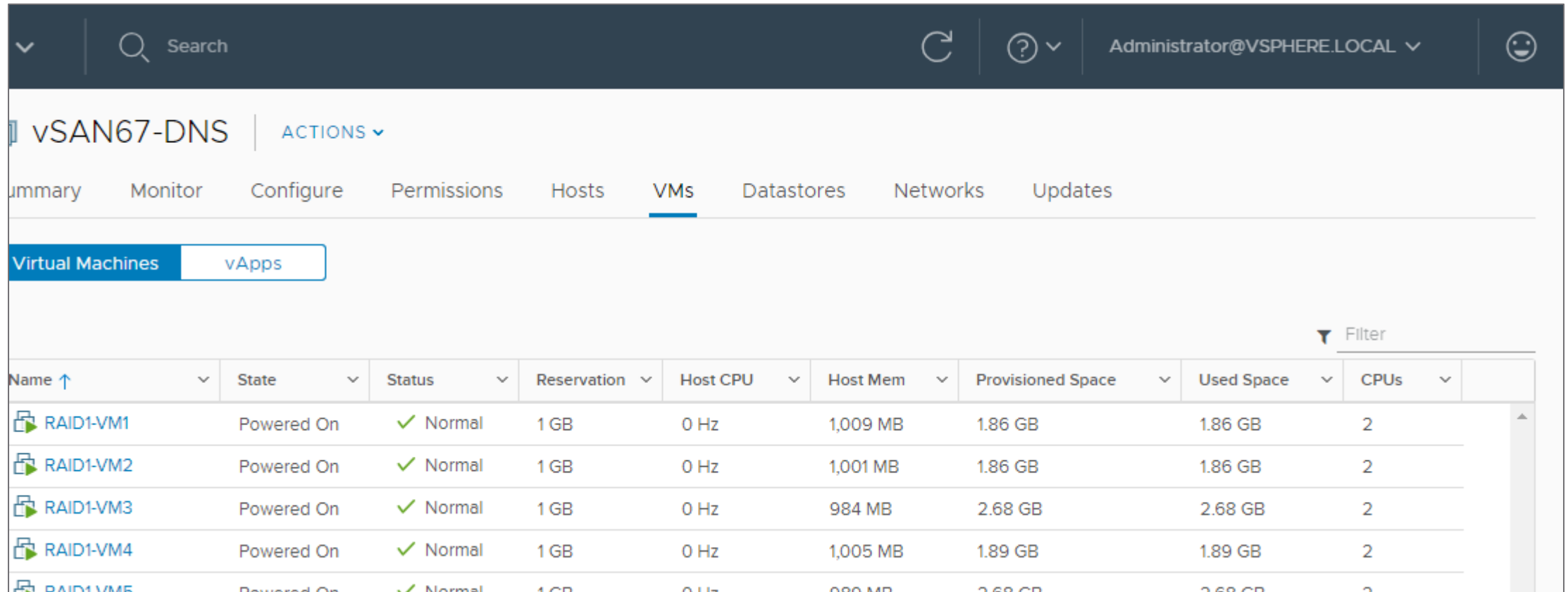


Filter

Name ↑ ▾	State ▾	Status ▾	Reservation ▾	Host CPU ▾	Host Mem ▾	Provisioned Space ▾	Used Space ▾	CPUs ▾
 RAID1-VM1	Powered On	✓ Normal	1 GB	0 Hz	1,009 MB	1.86 GB	1.86 GB	2
 RAID1-VM2	Powered On	✓ Normal	1 GB	0 Hz	1,001 MB	1.86 GB	1.86 GB	2
 RAID1-VM3	Powered On	✓ Normal	1 GB	0 Hz	984 MB	2.68 GB	2.68 GB	2
 RAID1-VM4	Powered On	✓ Normal	1 GB	0 Hz	1,005 MB	1.89 GB	1.89 GB	2
 RAID1-VM5	Powered On	✓ Normal	1 GB	0 Hz	989 MB	2.68 GB	2.68 GB	2
 RAID1-VM6	Powered On	✓ Normal	1 GB	0 Hz	1,002 MB	1.87 GB	1.87 GB	2
 RAID1-VM7	Powered On	✓ Normal	1 GB	0 Hz	1,005 MB	1.86 GB	1.86 GB	2

Resource Pools

What is consuming reservations? (CPU)

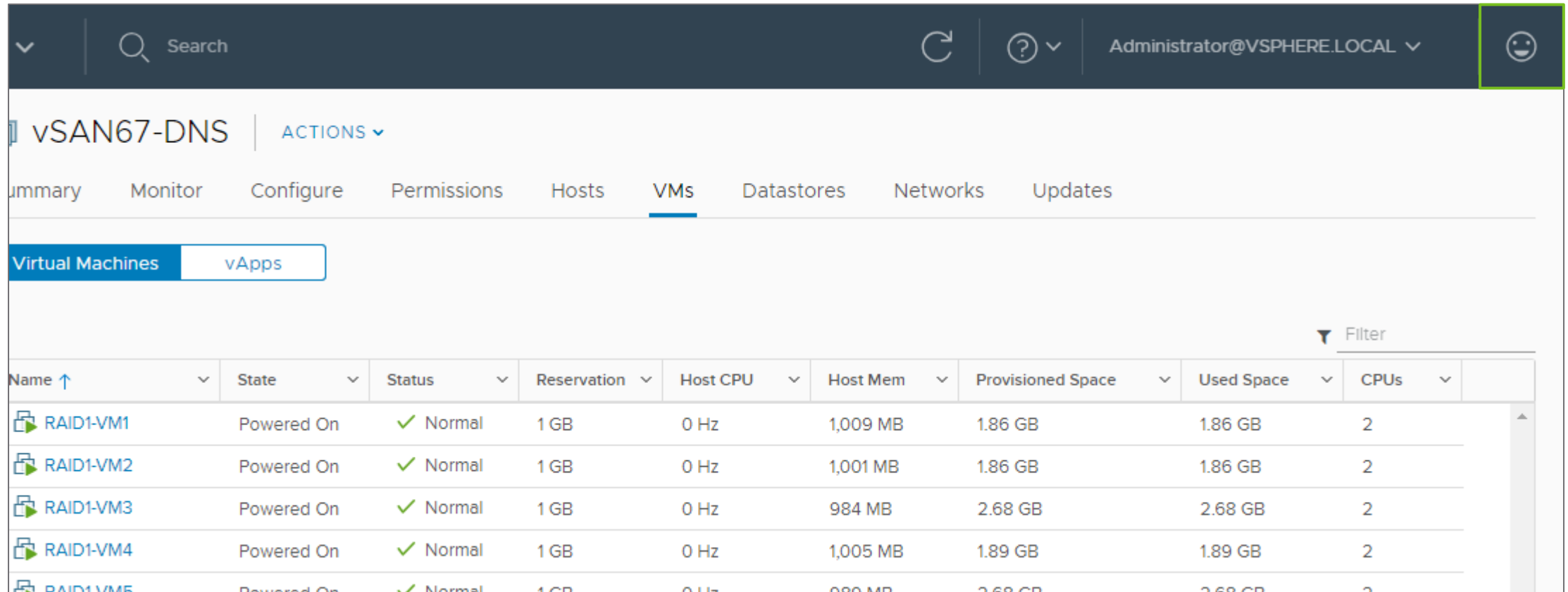


The screenshot shows the VMware vSphere interface for a resource pool named 'vSAN67-DNS'. The 'VMs' tab is selected, and the 'Virtual Machines' sub-tab is active. A table displays the following VMs:

Name	State	Status	Reservation	Host CPU	Host Mem	Provisioned Space	Used Space	CPUs
RAID1-VM1	Powered On	✓ Normal	1 GB	0 Hz	1,009 MB	1.86 GB	1.86 GB	2
RAID1-VM2	Powered On	✓ Normal	1 GB	0 Hz	1,001 MB	1.86 GB	1.86 GB	2
RAID1-VM3	Powered On	✓ Normal	1 GB	0 Hz	984 MB	2.68 GB	2.68 GB	2
RAID1-VM4	Powered On	✓ Normal	1 GB	0 Hz	1,005 MB	1.89 GB	1.89 GB	2
RAID1-VM5	Powered On	✓ Normal	1 GB	0 Hz	980 MB	2.68 GB	2.68 GB	2

Resource Pools

What is consuming reservations? (CPU)

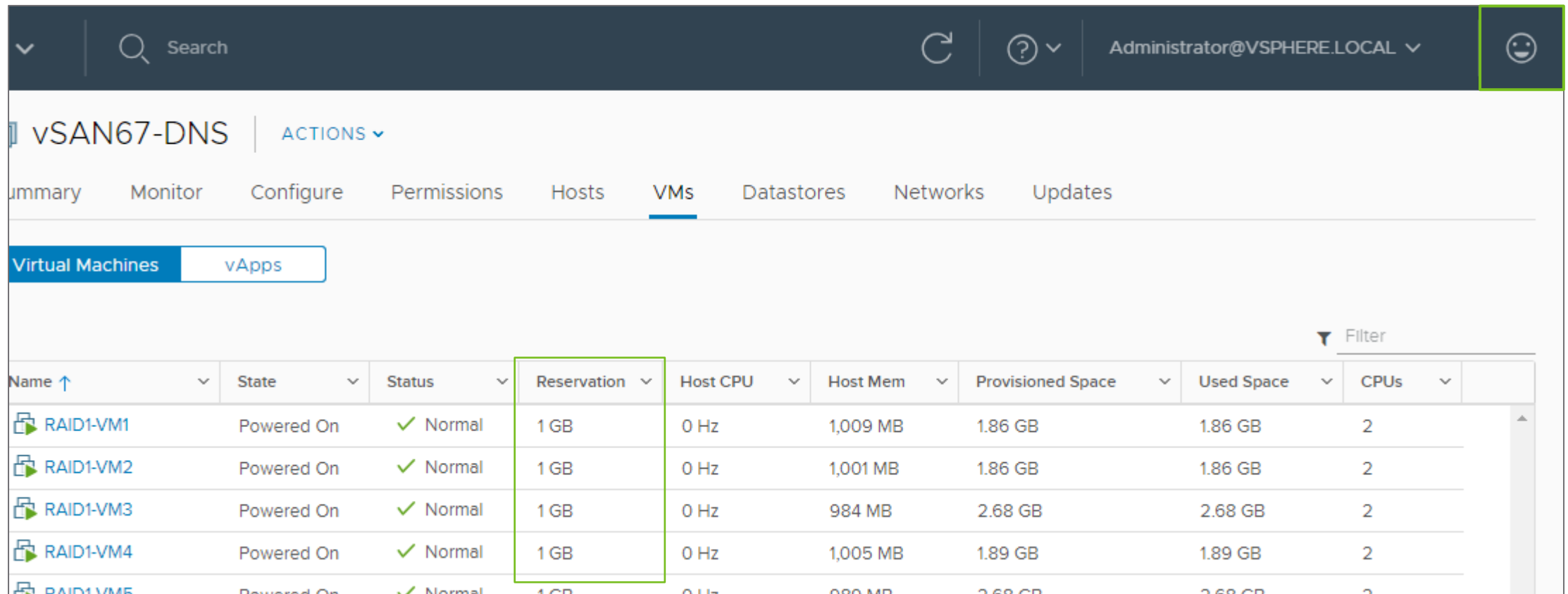


The screenshot shows the VMware vSphere interface for a resource pool named 'vSAN67-DNS'. The 'VMs' tab is selected, and the 'Virtual Machines' sub-tab is active. A table displays the following data:

Name	State	Status	Reservation	Host CPU	Host Mem	Provisioned Space	Used Space	CPUs
RAID1-VM1	Powered On	✓ Normal	1 GB	0 Hz	1,009 MB	1.86 GB	1.86 GB	2
RAID1-VM2	Powered On	✓ Normal	1 GB	0 Hz	1,001 MB	1.86 GB	1.86 GB	2
RAID1-VM3	Powered On	✓ Normal	1 GB	0 Hz	984 MB	2.68 GB	2.68 GB	2
RAID1-VM4	Powered On	✓ Normal	1 GB	0 Hz	1,005 MB	1.89 GB	1.89 GB	2
RAID1-VM5	Powered On	✓ Normal	1 GB	0 Hz	980 MB	2.68 GB	2.68 GB	2

Resource Pools

What is consuming reservations? (CPU)

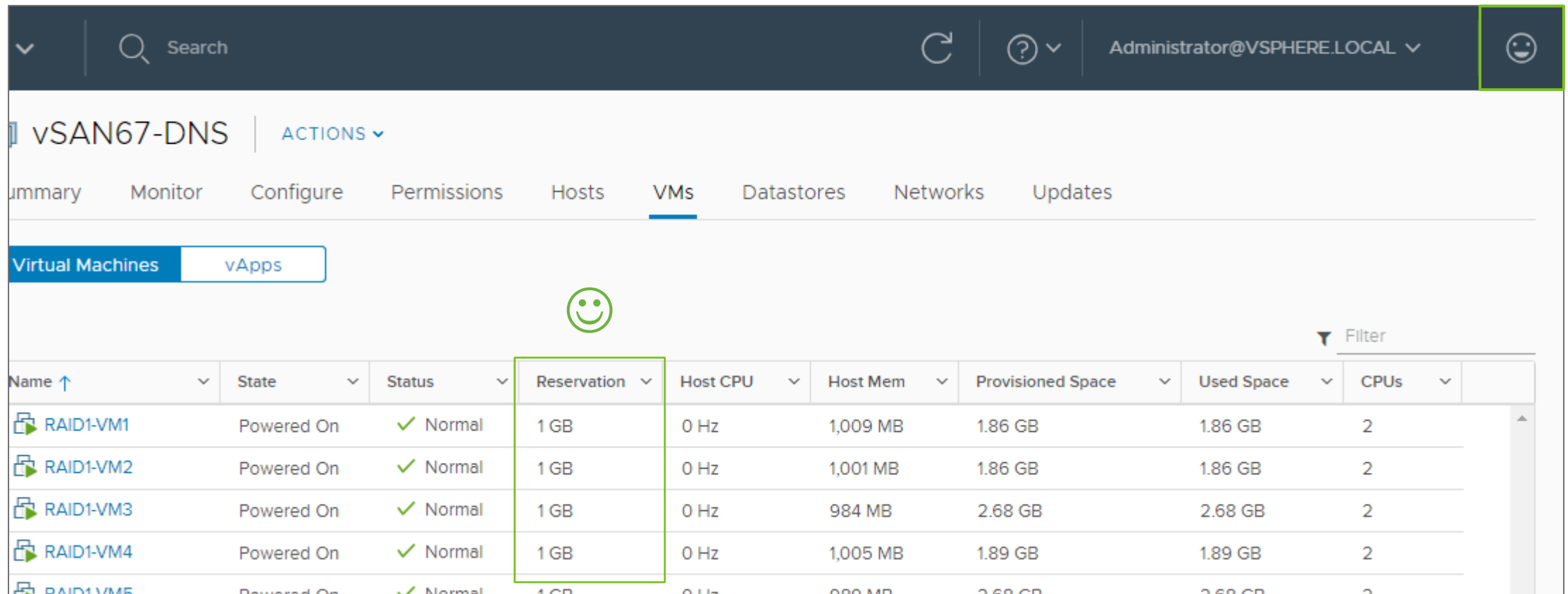


The screenshot shows the VMware vSphere interface for a resource pool named 'vSAN67-DNS'. The 'VMs' tab is selected, and the 'Virtual Machines' view is active. A table lists several VMs, with the 'Reservation' column highlighted by a green box. The 'Reservation' column shows that each VM has a reservation of 1 GB. The 'Host CPU' column shows 0 Hz for all VMs, indicating that the reservations are not being consumed.

Name	State	Status	Reservation	Host CPU	Host Mem	Provisioned Space	Used Space	CPUs
RAID1-VM1	Powered On	✓ Normal	1 GB	0 Hz	1,009 MB	1.86 GB	1.86 GB	2
RAID1-VM2	Powered On	✓ Normal	1 GB	0 Hz	1,001 MB	1.86 GB	1.86 GB	2
RAID1-VM3	Powered On	✓ Normal	1 GB	0 Hz	984 MB	2.68 GB	2.68 GB	2
RAID1-VM4	Powered On	✓ Normal	1 GB	0 Hz	1,005 MB	1.89 GB	1.89 GB	2
RAID1-VM5	Powered On	✓ Normal	1 GB	0 Hz	980 MB	2.68 GB	2.68 GB	2

Resource Pools

What is consuming reservations? (CPU)



The screenshot shows the VMware vSphere interface for a resource pool named 'vSAN67-DNS'. The 'VMs' tab is selected, and the 'Virtual Machines' view is active. A table lists several VMs, with the 'Reservation' column highlighted in green. A green smiley face icon is placed above the table, and another green smiley face icon is in the top right corner of the interface.

Name ↑	State	Status	Reservation	Host CPU	Host Mem	Provisioned Space	Used Space	CPUs
RAID1-VM1	Powered On	✓ Normal	1 GB	0 Hz	1,009 MB	1.86 GB	1.86 GB	2
RAID1-VM2	Powered On	✓ Normal	1 GB	0 Hz	1,001 MB	1.86 GB	1.86 GB	2
RAID1-VM3	Powered On	✓ Normal	1 GB	0 Hz	984 MB	2.68 GB	2.68 GB	2
RAID1-VM4	Powered On	✓ Normal	1 GB	0 Hz	1,005 MB	1.89 GB	1.89 GB	2
RAID1-VM5	Powered On	✓ Normal	1 GB	0 Hz	980 MB	2.68 GB	2.68 GB	2

Resource Pools

What is consuming reservations? (CPU)

```
PowerCLI C:\> Get-Cluster -Name "vSAN67-DNS" | Get-VM | Get-VMResourceConfiguration | Select-Object VM,CpuReservationMhz | Sort-Object VM
```

VM	CpuReservationMhz
RAID1-VM1	0
RAID1-VM2	0
RAID1-VM3	0
RAID1-VM4	0
RAID1-VM5	0
RAID1-VM6	0
RAID1-VM7	0
RAID1-VM8	0
RAID5-VM1	0
RAID5-VM2	0
RAID5-VM3	0
RAID5-VM4	0
RAID5-VM5	0
RAID5-VM6	0
RAID5-VM7	0
RAID5-VM8	0

```
PowerCLI C:\> 
```

powershell.exe[*64]:19544

PowerCLI C:\> Get-Cluster -Name "vSAN67-DNS" | Get-VM | Get-VMResourceConfiguration | Select-Object VM,CpuReservationMhz | Sort-Object VM

Resource Pools

What is consuming reservations? (Memory)

The screenshot displays the vSAN67-DNS interface, specifically the 'Monitor' tab. The left sidebar shows a navigation menu with 'Memory' selected under 'Resource Allocation'. The main content area is titled 'Memory Reservation Details' and features a bar chart showing 0 GB used and 303.01 GB available. Below the chart, a table lists reservation details for various tiers.

Name ↑	Reservation (MB)	Limit (MB)	Tier
01-Gold	0	Unlimited	E
02-Silver	0	Unlimited	E
03-Bronze	0	Unlimited	E
Participation	0	Unlimited	E

Summary of Memory Reservation Details:

- Cluster Total Capacity: 383.63 GB
- Total Reservation Capacity: 303.01 GB
- Used Reservation: 92.88 GB
- Available Reservation: 210.13 GB

Resource Pools

What is consuming reservations? (Memory)

vSAN67-DNS | ACTIONS ▾

Summary | **Monitor** | Configure | Permissions | Hosts | VMs | Datastore

▶ Issues and Alarms
▶ Performance
▶ Tasks and Events
▶ vSphere DRS
▶ vSphere HA
▼ Resource Allocation
 CPU
 Memory
 Storage
 Utilization
 Storage Overview
 Security
▼ vSAN
 Health
 Virtual Objects
 Physical Disks

Memory Reservation Details

0 GB 303.01 GB

Cluster Total Capacity	383.63 GB
Total Reservation Capacity	303.01 GB
<input checked="" type="checkbox"/> Used Reservation	92.88 GB
<input type="checkbox"/> Available Reservation	210.13 GB

Name ↑	Reservation (MB)	Limit (MB)	T...
01-Gold	0	Unlimited	E
02-Silver	0	Unlimited	E
03-Bronze	0	Unlimited	E
Participation	0	Unlimited	E

Resource Pools

What is consuming reservations? (Memory)

```
PowerCLI C:\> Get-Cluster -Name "vSAN67-DNS" | Get-VM | Get-VMResourceConfiguration | Select-Object VM,MemReservationGb | Sort-Object VM
```

VM	MemReservationGB
--	-----
RAID1-VM1	1
RAID1-VM2	1
RAID1-VM3	1
RAID1-VM4	1
RAID1-VM5	1
RAID1-VM6	1
RAID1-VM7	1
RAID1-VM8	1
RAID5-VM1	1
RAID5-VM2	1
RAID5-VM3	1
RAID5-VM4	1
RAID5-VM5	1
RAID5-VM6	1
RAID5-VM7	1
RAID5-VM8	1

```
PowerCLI C:\> 
```

powershell.exe[*64]:19544

PowerCLI C:\> Get-Cluster -Name "vSAN67-DNS" | Get-VM | Get-VMResourceConfiguration | Select-Object VM,MemReservationGb | Sort-Object VM

Resource Pools

What is consuming reservations? (Memory)

```
PowerCLI C:\> Get-Cluster -Name "vSAN67-DNS" | Get-VM | Get-VMResourceConfiguration | Measure-Object MemReservationGb -Sum | Select Sum

Sum
---
 16

PowerCLI C:\> █
powershell.exe[*64]:19544
```

Remember: 16 GB

```
PowerCLI C:\> Get-Cluster -Name "vSAN67-DNS" | Get-VM | Get-VMResourceConfiguration | Measure-Object MemReservationGb -Sum | Select-Object Sum
```

What could cause the difference? *

* 33.17 GHz vs 0 and 92.88 GB vs 16

Resource Pools

Admission Control

Edit Cluster Settings | vSAN67-DNS ✕

vSphere HA

Failures and responses | **Admission Control** | Heartbeat Datastores | Advanced Options

Admission control is a policy used by vSphere HA to ensure failover capacity within a cluster. Raising the number of potential host failures will increase the availability constraints and capacity reserved.

Host failures cluster tolerates
Maximum is one less than number of hosts in cluster.

Define host failover capacity by

Override calculated failover capacity.

Reserved failover CPU capacity: % CPU
Reserved failover Memory capacity: % Memory

Performance degradation VMs tolerate %

Resource Pools

Admission Control

Edit Cluster Settings | vSAN67-DNS ✕

vSphere HA

Failures and responses | **Admission Control** | Heartbeat Datastores | Advanced Options

Admission control is a policy used by vSphere HA to ensure failover capacity within a cluster. Raising the number of potential host failures will increase the availability constraints and capacity reserved.

Host failures cluster tolerates
Maximum is one less than number of hosts in cluster.

Define host failover capacity by

Override calculated failover capacity.

Reserved failover CPU capacity: % CPU
Reserved failover Memory capacity: % Memory

Performance degradation VMs tolerate %

Resource Pools

Without Admission Control

The screenshot shows the vSAN67-DNS Monitor page with the 'Monitor' tab selected. The left sidebar lists navigation options, with 'CPU' under 'Resource Allocation' highlighted. The main content area displays 'CPU Reservation Details' with a progress bar showing 0 GHz used out of 132.67 GHz total. Below the bar is a table of reservation details.

Name ↑	Reservation (MHz)	Limit (MHz)	...
01-Gold	0	Unlimited	...
02-Silver	0	Unlimited	...
03-Bronze	0	Unlimited	...
Participation	0	Unlimited	...

The screenshot shows the vSAN67-DNS Monitor page with the 'Monitor' tab selected. The left sidebar lists navigation options, with 'Memory' under 'Resource Allocation' highlighted. The main content area displays 'Memory Reservation Details' with a progress bar showing 0 GB used out of 303.01 GB total. Below the bar is a table of reservation details.

Name ↑	Reservation (MB)	Limit (MB)	...
01-Gold	0	Unlimited	...
02-Silver	0	Unlimited	...
03-Bronze	0	Unlimited	...
Participation	0	Unlimited	...

Resource Pools

Without Admission Control

The screenshot shows the vSAN67-DNS Monitor page with the CPU Reservation Details section. A red box highlights the 'Used Reservation' value of 0 GHz. The table below shows reservation details for various resource pools.

Name ↑	Reservation (MHz)	Limit (MHz)	...
01-Gold	0	Unlimited	...
02-Silver	0	Unlimited	...
03-Bronze	0	Unlimited	...
Participation	0	Unlimited	...

The screenshot shows the vSAN67-DNS Monitor page with the Memory Reservation Details section. A red box highlights the 'Used Reservation' value of 17.13 GB. The table below shows reservation details for various resource pools.

Name ↑	Reservation (MB)	Limit (MB)	...
01-Gold	0	Unlimited	...
02-Silver	0	Unlimited	...
03-Bronze	0	Unlimited	...
Participation	0	Unlimited	...

Resource Pools

Creation and Maximums

The screenshot shows the vSAN67-DNS Monitor page with the 'CPU' reservation details. A progress bar at the top indicates 0 GHz used out of a 132.67 GHz limit. A table below shows the cluster's total capacity and reservation capacity, with the reservation capacity highlighted in red. A secondary table lists reservation details for various resource pools.

Name ↑	Reservation (MHz)	Limit (MHz)	...
01-Gold	0	Unlimited	...
02-Silver	0	Unlimited	...
03-Bronze	0	Unlimited	...
Participation	0	Unlimited	...

CPU Reservation Details

0 GHz / 132.67 GHz

Cluster Total Capacity	153.57 GHz
Total Reservation Capacity	132.67 GHz

Used Reservation: 0 Hz
Available Reservation: 132.67 GHz

The screenshot shows the vSAN67-DNS Monitor page with the 'Memory' reservation details. A progress bar at the top indicates 0 GB used out of a 303.01 GB limit. A table below shows the cluster's total capacity and reservation capacity, with the reservation capacity highlighted in red. A secondary table lists reservation details for various resource pools.

Name ↑	Reservation (MB)	Limit (MB)	...
01-Gold	0	Unlimited	...
02-Silver	0	Unlimited	...
03-Bronze	0	Unlimited	...
Participation	0	Unlimited	...

Memory Reservation Details

0 GB / 303.01 GB

Cluster Total Capacity	383.63 GB
Total Reservation Capacity	303.01 GB

Used Reservation: 17.13 GB
Available Reservation: 285.88 GB

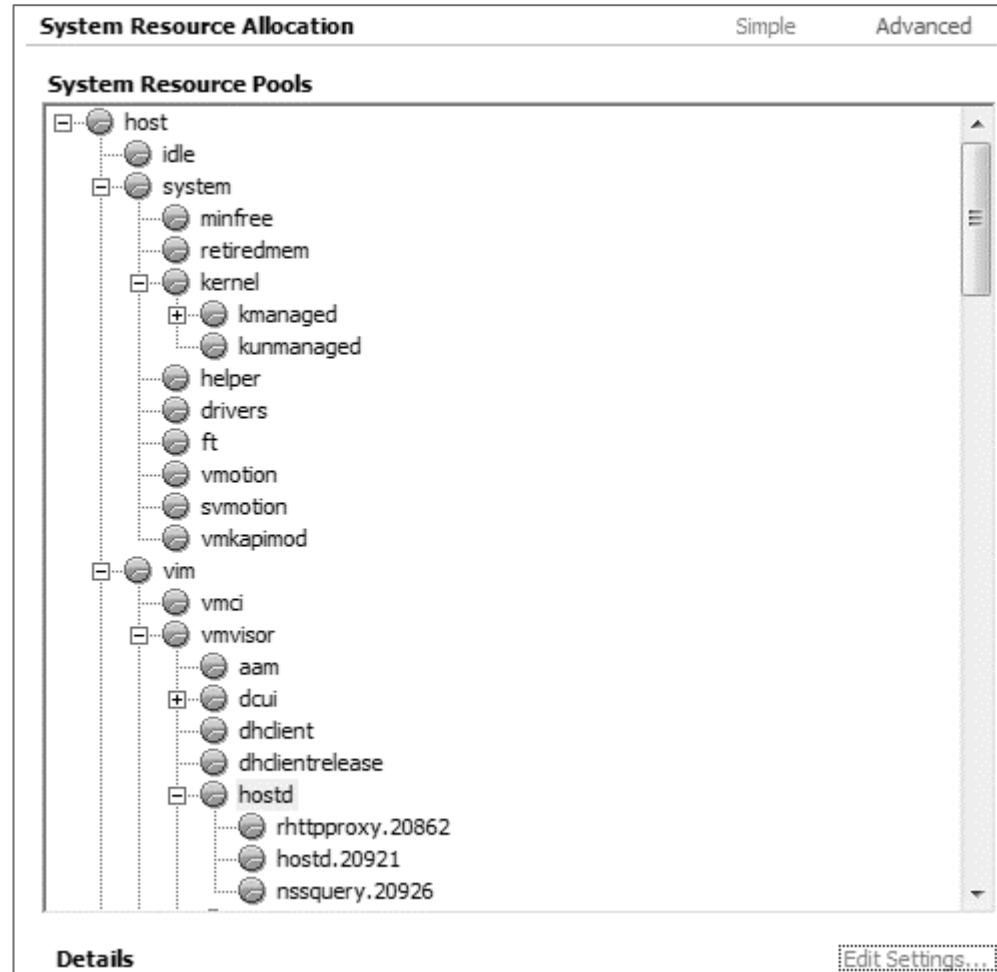
| 20.90 GHz
80.62 GB

Hierarchical Resource Groups

From an ESXi perspective

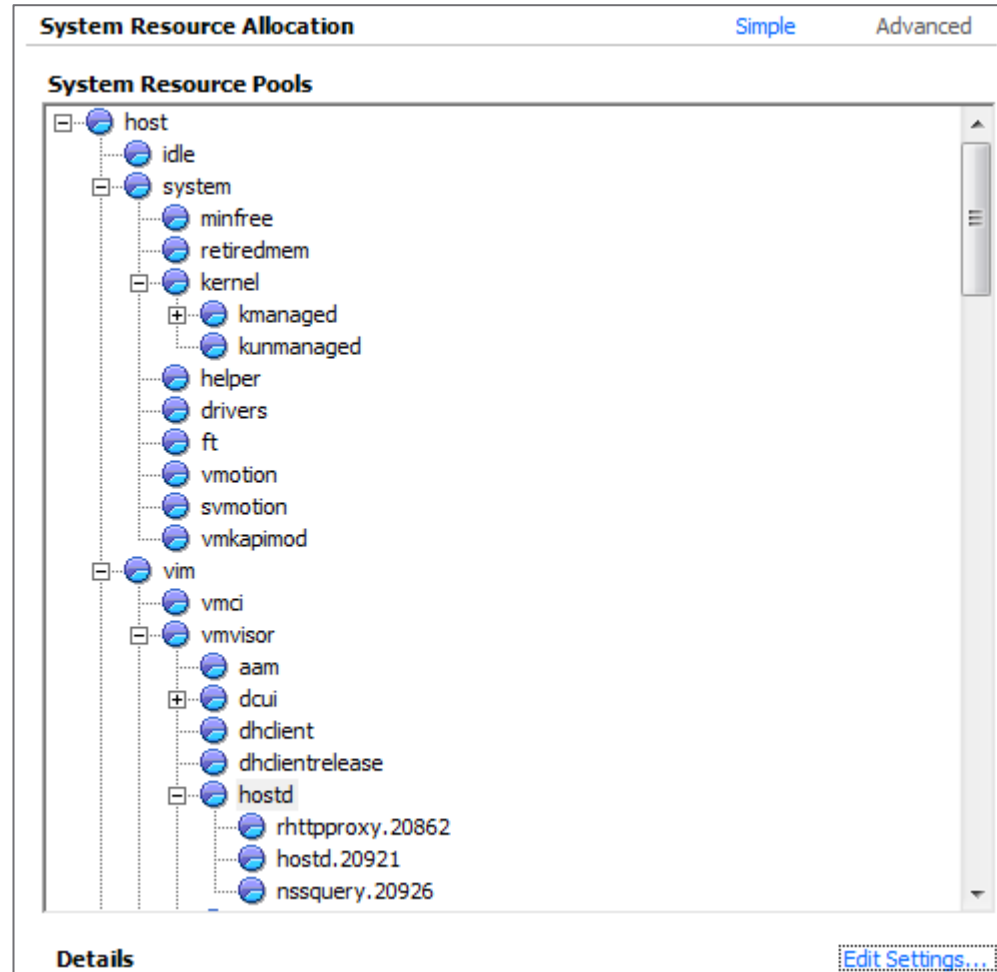
Hierarchical Resource Groups

From an ESXi perspective



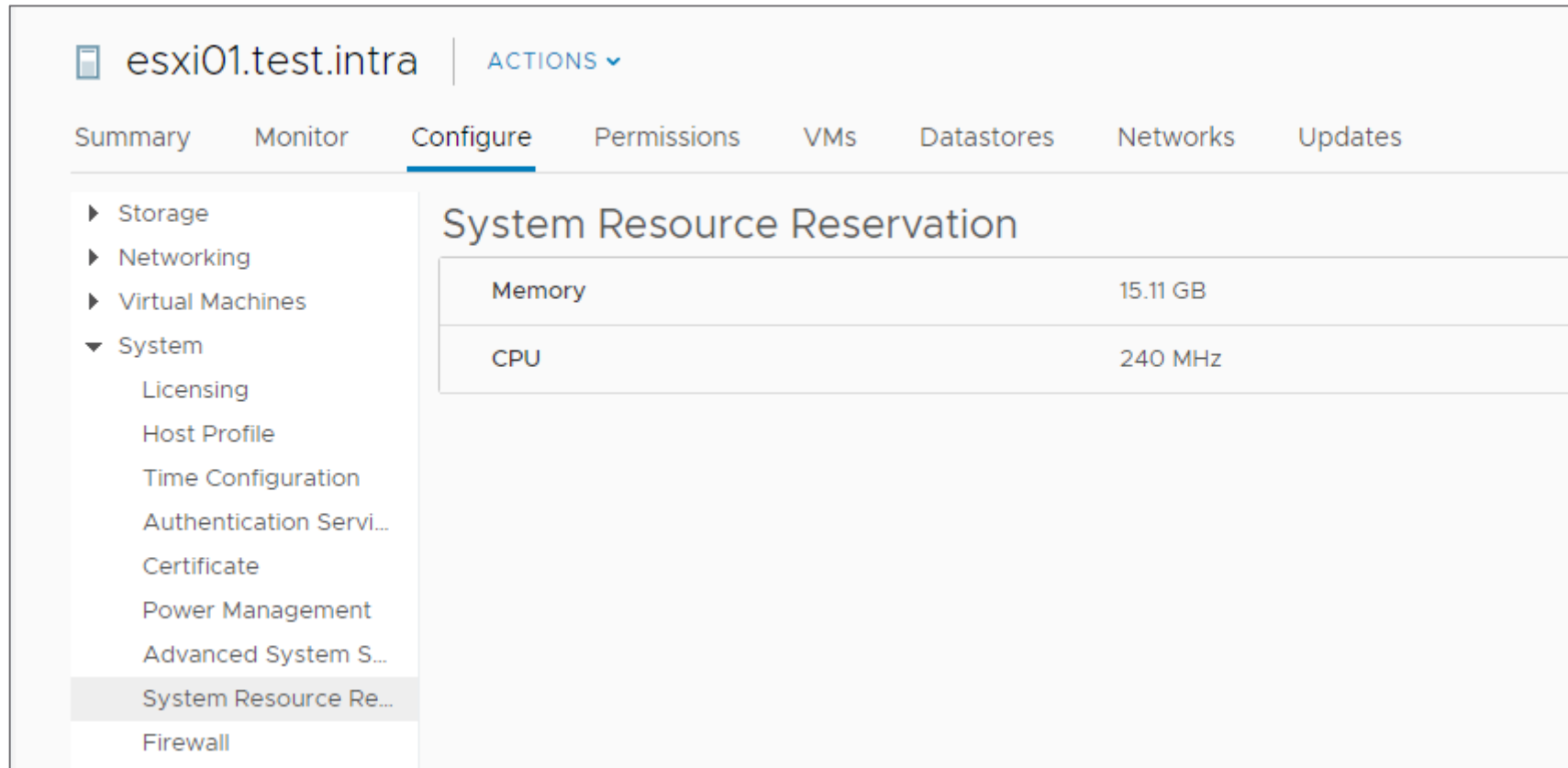
Hierarchical Resource Groups

From an ESXi perspective



Hierarchical Resource Groups

From an ESXi perspective



The screenshot displays the vSphere Client interface for the host 'esxi01.test.intra'. The 'Configure' tab is active, and the 'System Resource Reservation' configuration page is shown. The left-hand navigation pane lists various system settings, with 'System Resource Re...' selected. The main content area shows a table with the following data:

System Resource Reservation	
Memory	15.11 GB
CPU	240 MHz

Hierarchical Resource Groups

From an ESXi perspective

The screenshot shows the vSphere Client interface for host `esxi01.test.intra`. The `Configure` tab is active, and the `System Resource Reservation` configuration page is displayed. The left-hand navigation pane shows the `System` category expanded, with `System Resource Re...` selected. The main content area shows a table with the following data:

Resource	Value
Memory	15.11 GB
CPU	240 MHz

No hidden edit button off screen

Hierarchical Resource Groups

From an ESXi perspective

esxi01.test.intra | ACTIONS ▾

Summary Monitor **Configure** Permissions VMs Datastores Networks Updates

▶ Storage
▶ Networking
▶ Virtual Machines
▼ System
Licensing
Host Profile
Time Configuration
Authentication Servi...
Certificate
Power Management
Advanced System S...
System Resource Re...
Firewall

System Resource Reservation

Memory	15.11 GB
CPU	240 MHz
Memory	3.8 GB
CPU	240 MHz
Memory	3.79 GB
CPU	240 MHz
Memory	3.8 GB
CPU	240 MHz

No hidden edit button off screen

Hierarchical Resource Groups

From an ESXi perspective

esxi01.test.intra | ACTIONS ▾

Summary Monitor **Configure** Permissions VMs Datastores Networks Updates

- ▶ Storage
- ▶ Networking
- ▶ Virtual Machines
- ▼ System
 - Licensing
 - Host Profile
 - Time Configuration
 - Authentication Servi...
 - Certificate
 - Power Management
 - Advanced System S...
 - System Resource Re...**
 - Firewall

System Resource Reservation

Memory	15.11 GB
CPU	240 MHz

Memory	3.8 GB
CPU	240 MHz

Memory	3.79 GB
CPU	240 MHz

Memory	3.8 GB
CPU	240 MHz

No hidden edit button off screen



| 20.90 GHz
80.62 GB

20.90 GHz - 4 x 240 MHz

80.62 GB - 15.11 GB - 3 x 3.8 GB

20.90 GHz - 4 x 240 MHz $\neq 0$

80.62 GB - 15.11 GB - 3 x 3.8 GB $\neq 0$

Hierarchical Resource Groups

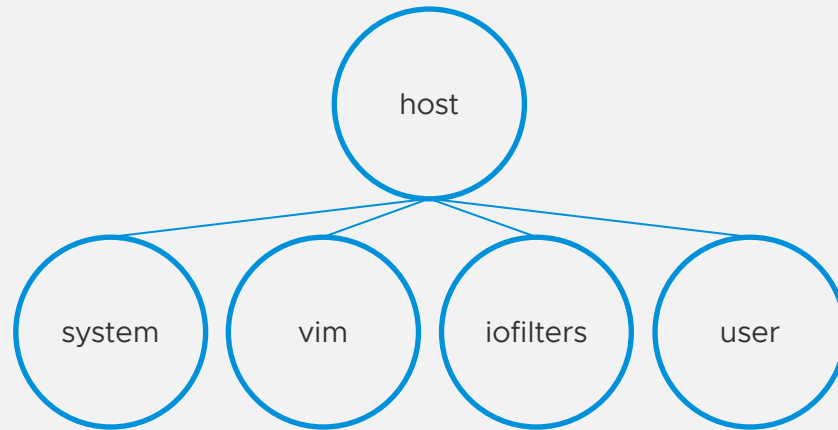
From an ESXi perspective



The host owns all resources

Hierarchical Resource Groups

From an ESXi perspective

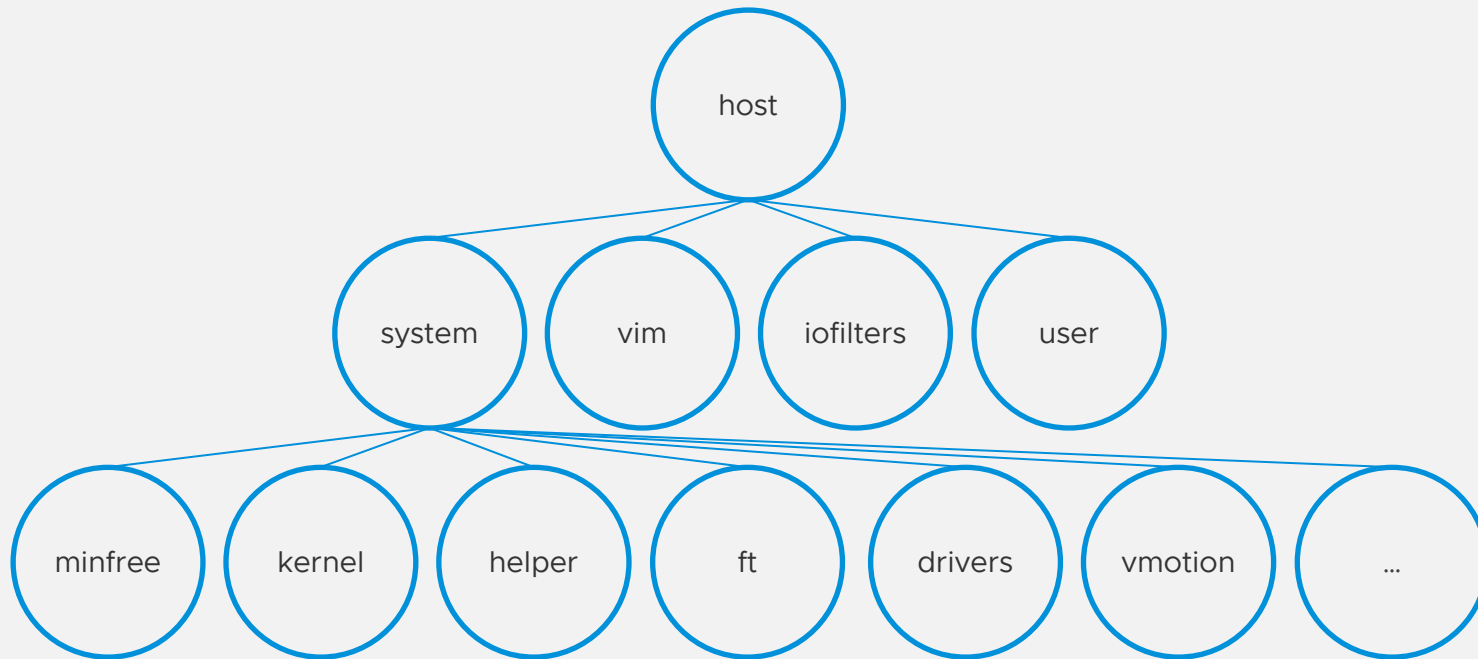


The host owns all resources

Those are distributed by hierarchical resource groups

Hierarchical Resource Groups

From an ESXi perspective

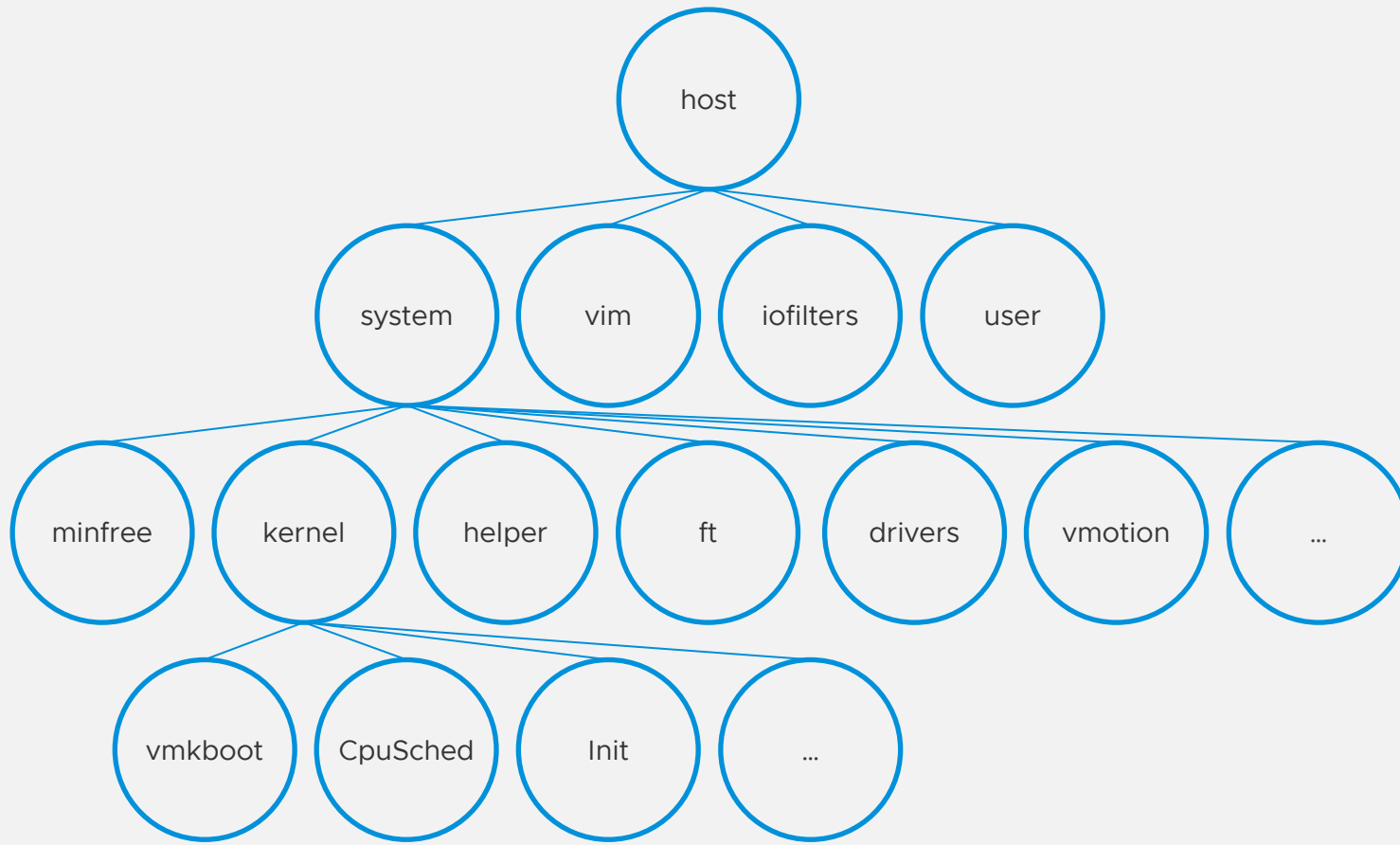


The host owns all resources

Those are distributed by hierarchical resource groups

Hierarchical Resource Groups

From an ESXi perspective

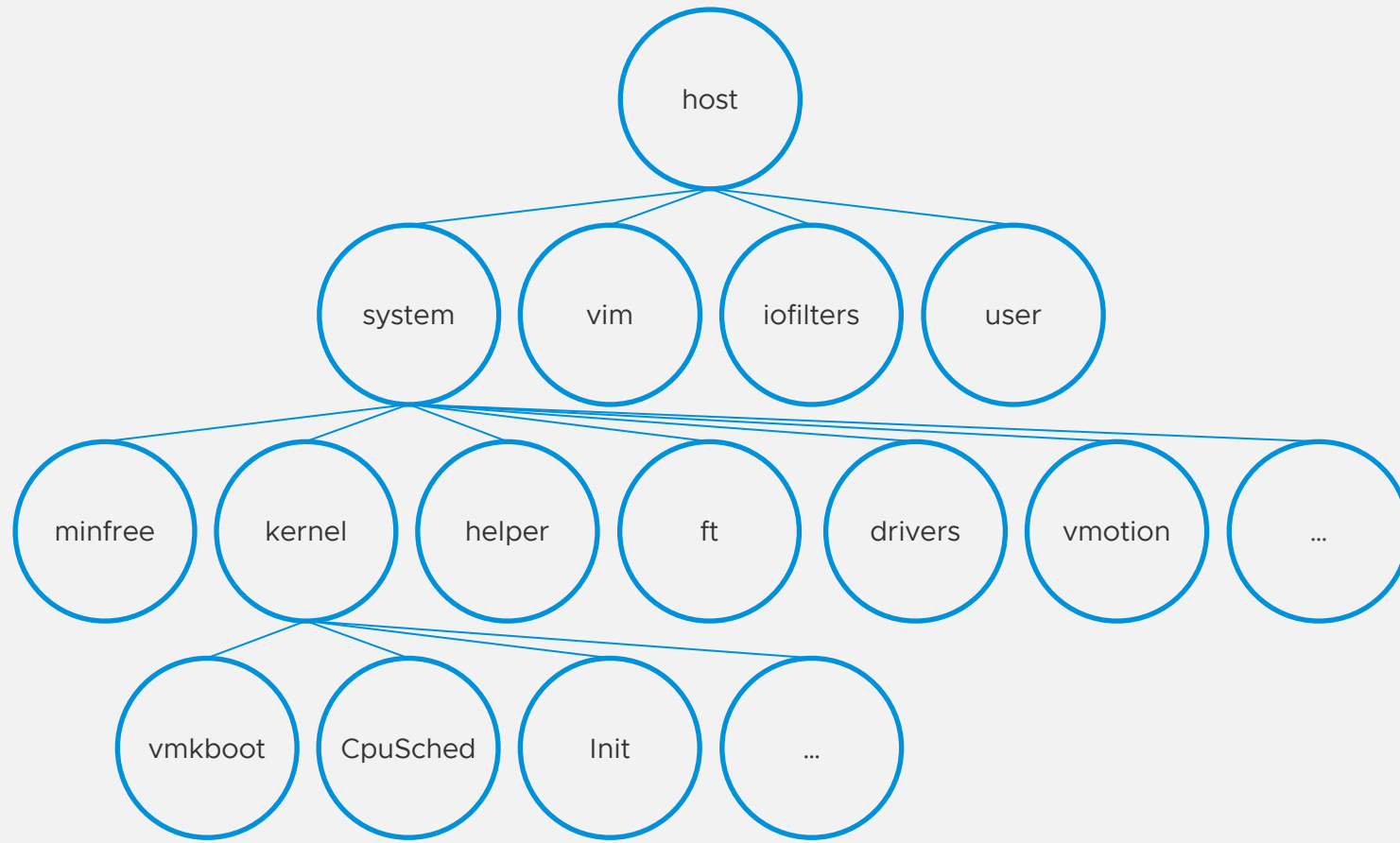


The host owns all resources

Those are distributed by hierarchical resource groups

Hierarchical Resource Groups

From an ESXi perspective



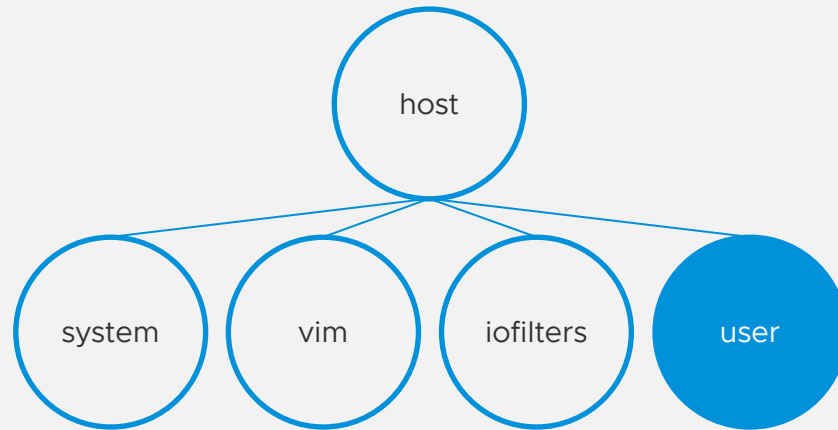
The host owns all resources

Those are distributed by hierarchical resource groups

Consumers can demand (request) resources

Hierarchical Resource Groups

From an ESXi perspective

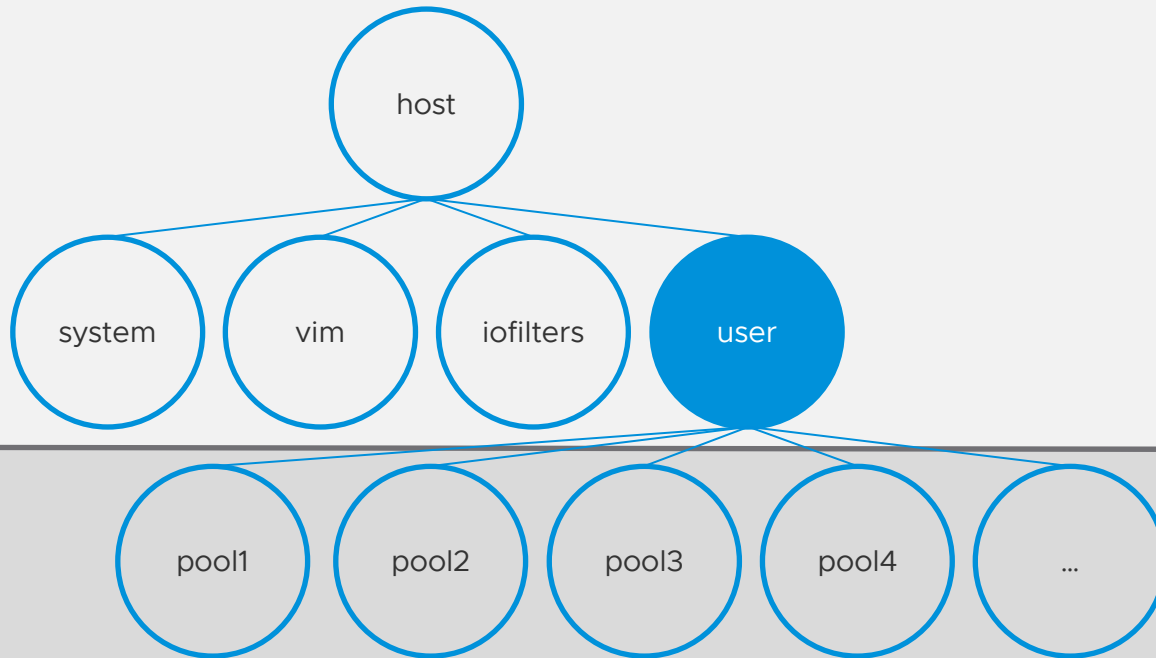


vCenter shows the sum of all **user** resources as:

Total Reservation Capacity

Hierarchical Resource Groups

From an ESXi perspective



vCenter shows the sum of all **user** resources as:

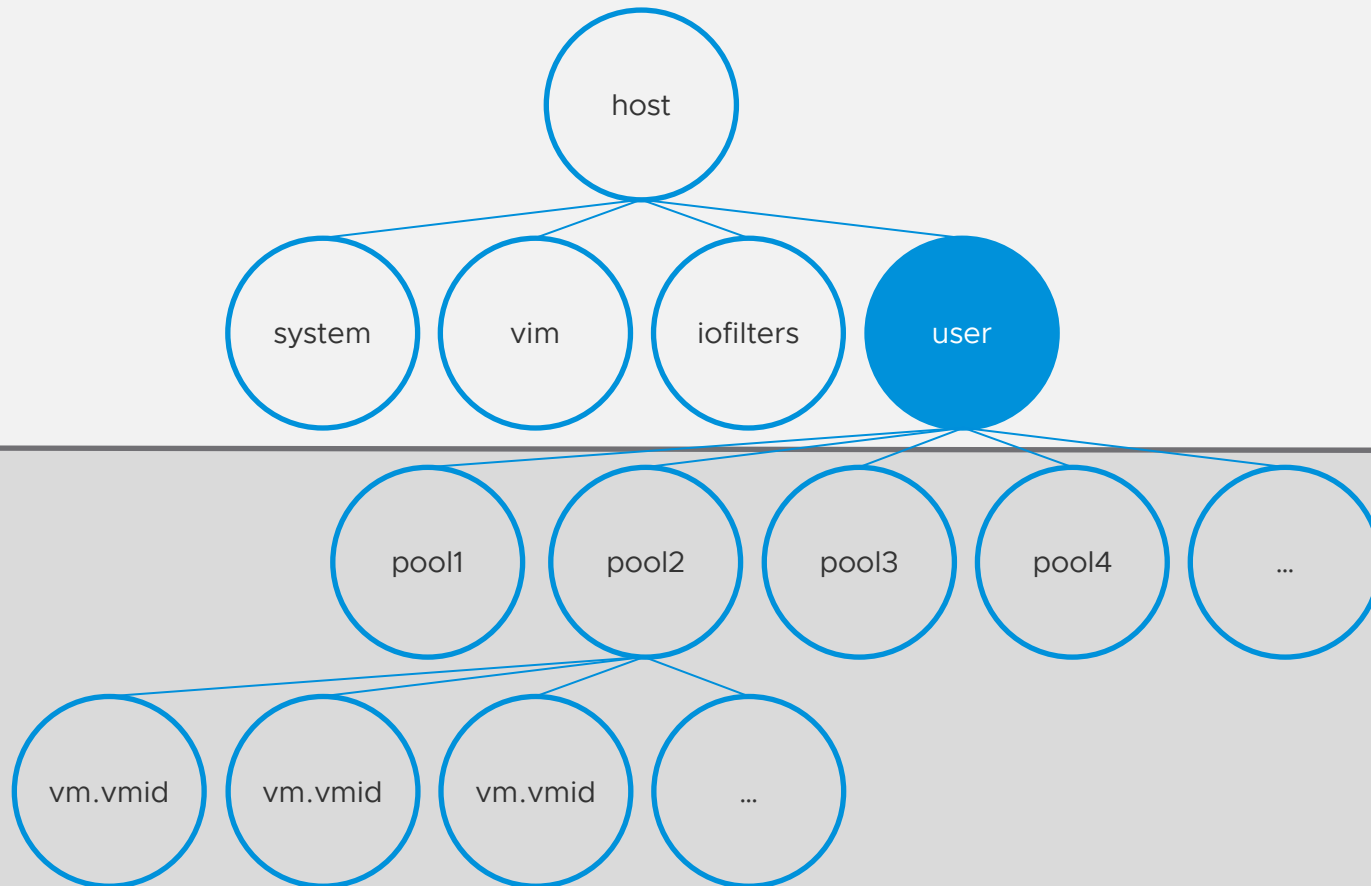
Total Reservation Capacity

Global Resource Pools are then distributed back to hosts into Local RPs

- Based on VMs demand

Hierarchical Resource Groups

From an ESXi perspective



vCenter shows the sum of all **user** resources as:

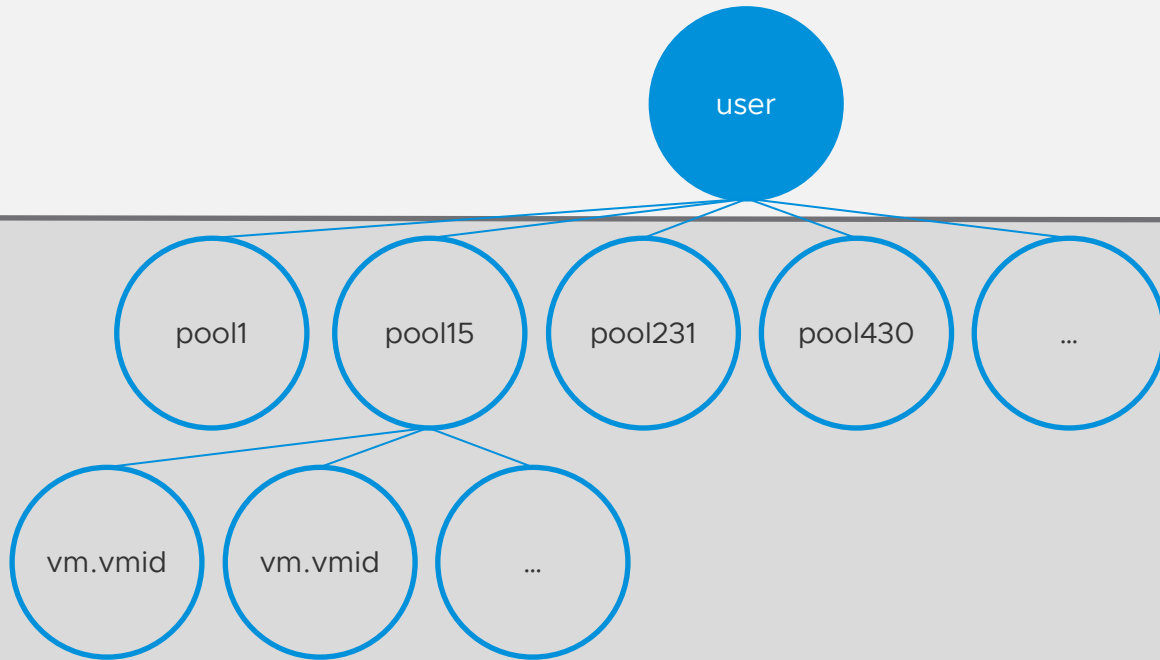
Total Reservation Capacity

Global Resource Pools are then distributed back to hosts into Local RPs

- Based on VMs demand

Hierarchical Resource Groups

From an ESXi perspective

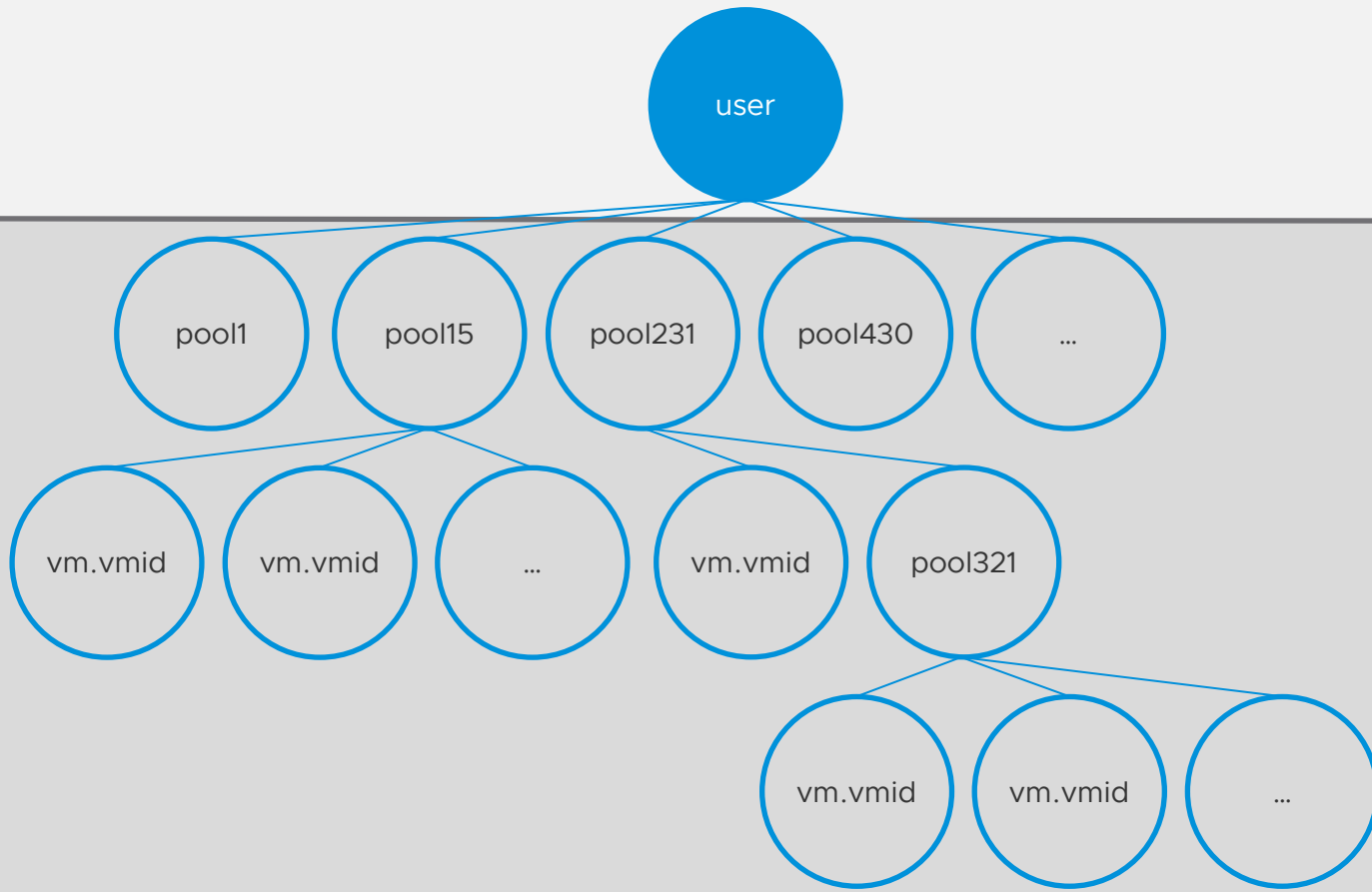


Local Resource Groups are created and incrementally numbered when clients are instantiated:

- VM starts / vMotions etc.
- Based on VMs demand

Hierarchical Resource Groups

From an ESXi perspective



Local Resource Groups are created and incrementally numbered when clients are instantiated:

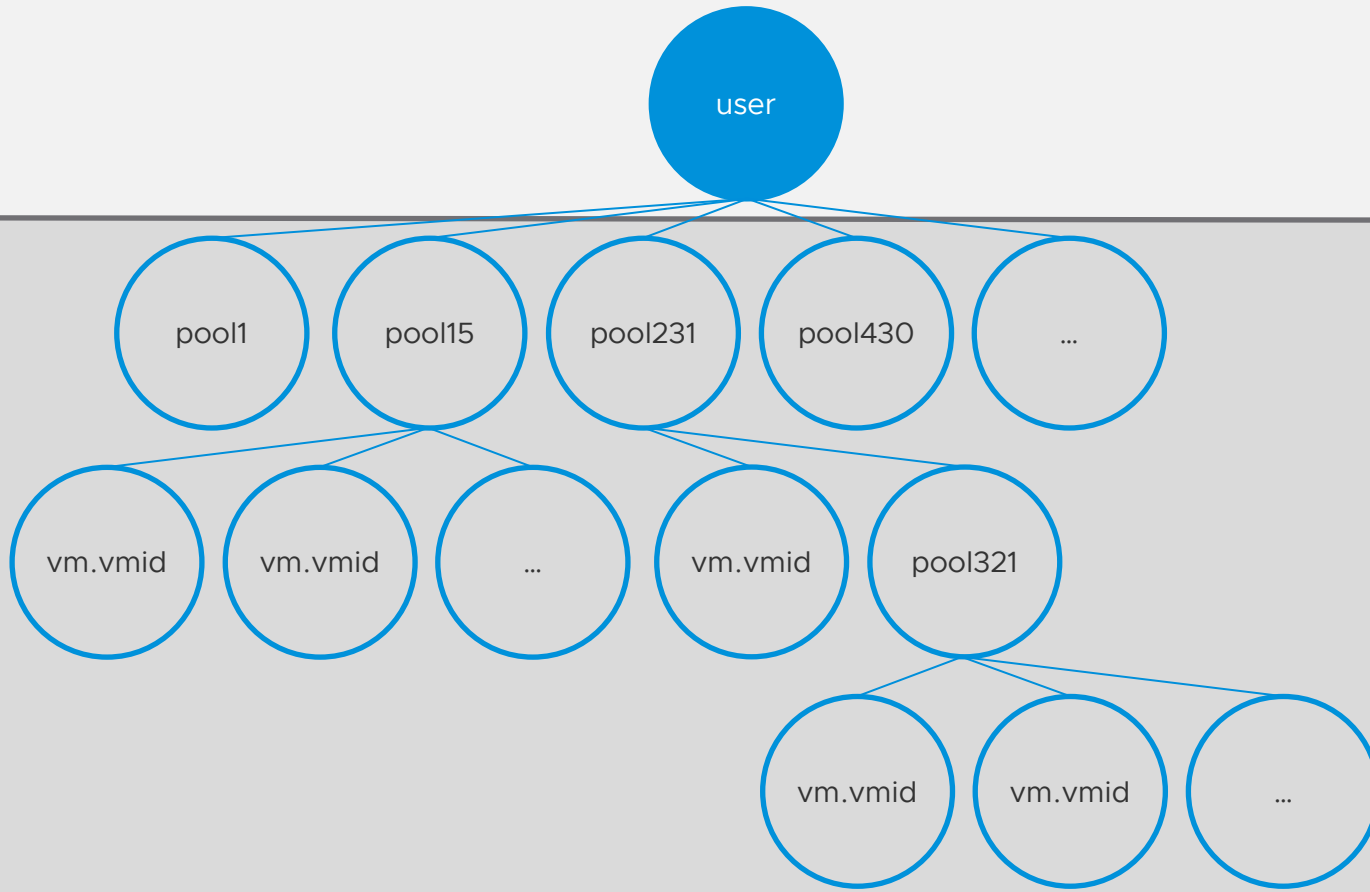
- VM starts / vMotions etc.
- Based on VMs demand

The local hierarchy is equal to the global one

- Check for VM / LRG siblings

Hierarchical Resource Groups

From an ESXi perspective



Local Resource Groups are created and incrementally numbered when clients are instantiated:

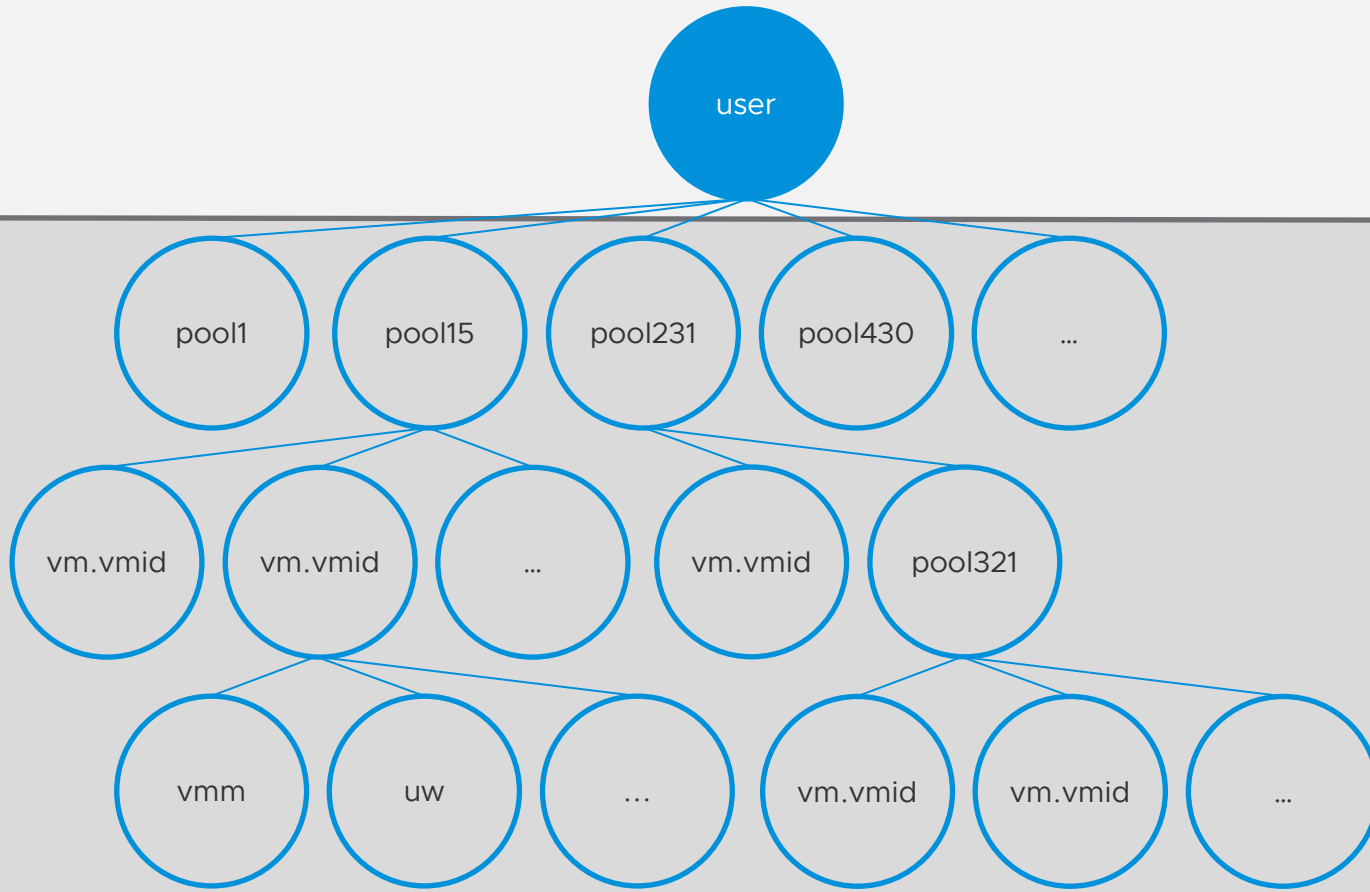
- VM starts / vMotions etc.
- Based on VMs demand

The local hierarchy is equal to the global one

- Check for VM / LRG siblings

Hierarchical Resource Groups

From an ESXi perspective



Local Resource Groups are created and incrementally numbered when clients are instantiated:

- VM starts / vMotions etc.
- Based on VMs demand

The local hierarchy is equal to the global one

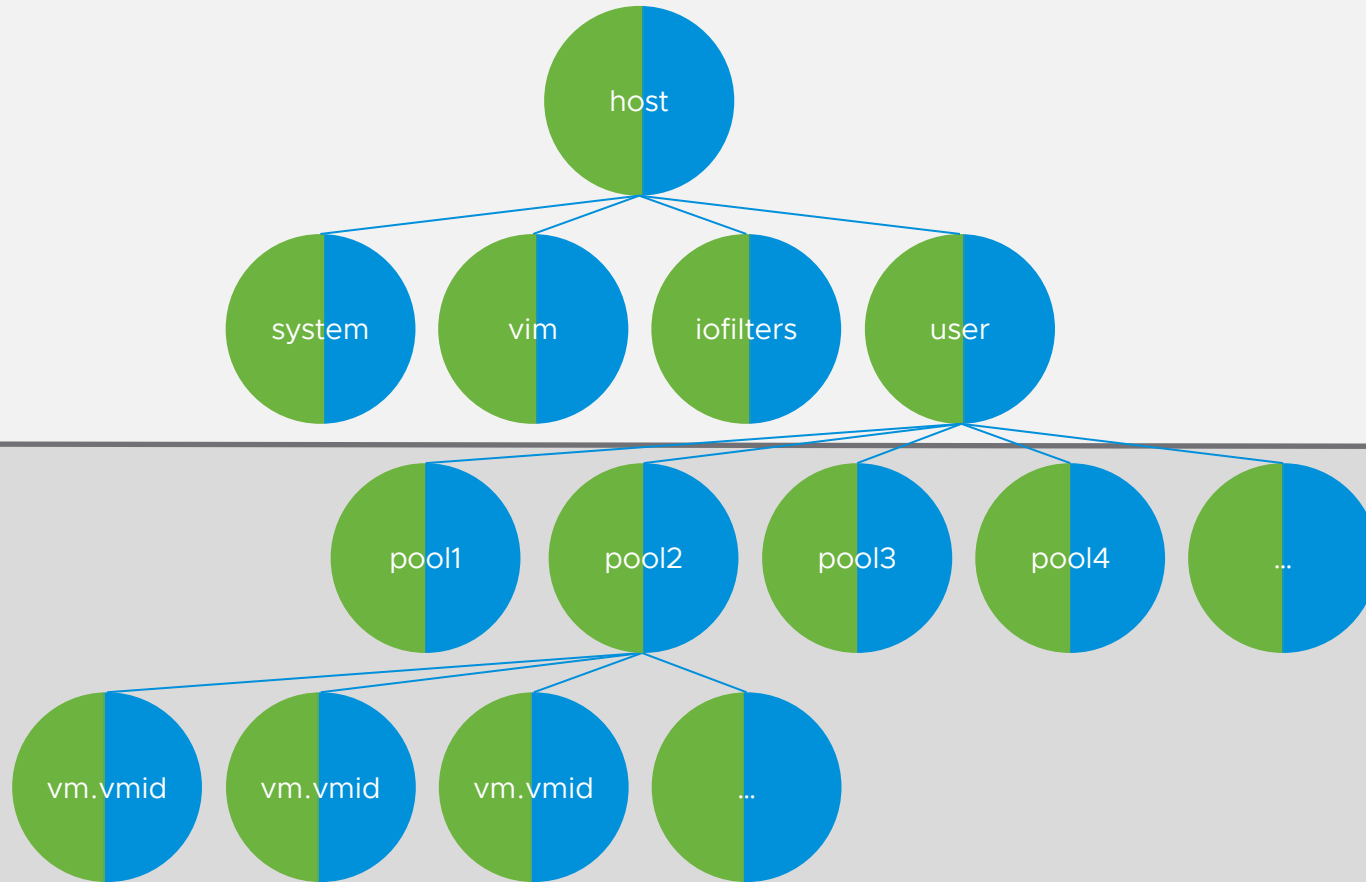
- Check for VM / LRG siblings

VM groups have multiple leaf consumers

- vmid is local, not global

Hierarchical Resource Groups

Both Memory and CPU resources



Memory

mem.resv	Reservation
mem.limit	Limit
mem.shares	Shares
mem.resvLimit	Expandable*

CPU

cpu.resv	Reservation
cpu.limit	Limit
cpu.shares	Shares
cpu.resvLimit	Expandable*

Tools

ESXi CLI (via SSH)



sched-stats

... for CPU



memstats

... for Memory



esxtop

... for comparison

Tools

cmdline for local groups (no VMs)

sched-stats

Tools

cmdline for local groups (no VMs)

sched-stats

```
# sched-stats -t groups | awk 'NR == 1'
```

Tools

cmdline for local groups (no VMs)

sched-stats

```
# sched-stats -t groups | awk 'NR == 1  
  || $2 ~ /^(vm\.|pool)[0-9]+/'
```

Tools

cmdline for local groups (no VMs)

sched-stats

```
# sched-stats -t groups | awk 'NR == 1  
|| $2 ~ /^(vm\.|pool)[0-9]+/  
|| /^ +[0-4] /
```

Tools

cmdline for local groups (no VMs)

sched-stats

```
# sched-stats -t groups | awk 'NR == 1
|| $2 ~ /^(vm\.|pool)[0-9]+/
|| /^ +[0-4] /
{printf ("% -10s% -12s% -9s% -6s% -6s% -6s% -9s% -6s% -9s% -9s% -10s\n"
,$1, $2, $3, $6, $8, $9, $10, $11, $12, $13, $14) }'
```

Tools

cmdline for local groups (no VMs)

```
# sched-stats -t groups | awk 'NR == 1
|| $2 ~ /^(vm\.|pool)[0-9]+/
|| /^ +[0-4] /
{printf ("% -10s% -12s% -9s% -6s% -6s% -6s% -9s% -6s% -9s% -9s% -10s\n"
,$1, $2, $3, $6, $8, $9, $10, $11, $12, $13, $14)}'
```

vmgid	name	pgid	vsmpps	amin	amax	minLimit	units	ashares	resvMHz	availMHz
0	host	0	933	1600	1600	1600	pct	4096000	5232	33168
1	system	0	659	10	-1	-1	pct	500	288	33168
2	vim	0	271	4944	-1	-1	mhz	500	4344	33768
3	iofilters	0	3	0	-1	-1	pct	1000	0	33168
4	user	0	0	0	-1	-1	pct	9000	0	33168

Tools

cmdline for local groups (no VMs)

```
# sched-stats -t groups | awk 'NR == 1
|| $2 ~ /^(vm\.|pool)[0-9]+/
|| /^ +[0-4] /
{printf ("% -10s% -12s% -9s% -6s% -6s% -6s% -9s% -6s% -9s% -9s% -10s\n"
,$1, $2, $3, $6, $8, $9, $10, $11, $12, $13, $14)}'
```

vmgid	name	pgid	vsmps	amin	amax	minLimit	units	ashares	resvMHz	availMHz
0	host	0	933	1600	1600	1600	pct	4096000	5232	33168
1	system	0	659	10	-1	-1	pct	500	288	33168
2	vim	0	271	4944	-1	-1	mhz	500	4344	33768
3	iofilters	0	3	0	-1	-1	pct	1000	0	33168
4	user	0	0	0	-1	-1	pct	9000	0	33168

Tools

cmdline for local groups (no VMs)

```
# sched-stats -t groups | awk 'NR == 1
|| $2 ~ /^(vm\.|pool)[0-9]+/
|| /^ +[0-4] /
{printf ("% -10s% -12s% -9s% -6s% -6s% -6s% -9s% -6s% -9s% -9s% -10s\n"
,$1, $2, $3, $6, $8, $9, $10, $11, $12, $13, $14)}'
```

vmgid	name	pgid	vsmps	amin	amax	minLimit	units	ashares	resvMHz	availMHz
0	host	0	933	1600	1600	1600	pct	4096000	5232	33168
1	system	0	659	10	-1	-1	pct	500	288	33168
2	vim	0	271	4944	-1	-1	mhz	500	4344	33768
3	iofilters	0	3	0	-1	-1	pct	1000	0	33168
4	user	0	0	0	-1	-1	pct	9000	0	33168

Tools

cmdline for local groups (no VMs)

sched-stats

```
# sched-stats -t groups | awk 'NR == 1
|| $2 ~ /^(vm\.|pool)[0-9]+/
|| /^ +[0-4] /
{printf ("% -10s% -12s% -9s% -6s% -6s% -6s% -9s% -6s% -9s% -9s% -10s\n"
,$1, $2, $3, $6, $8, $9, $10, $11, $12, $13, $14)}'
```

vmgid	name	pgid	vsmpps	amin	amax	minLimit	units	ashares	resvMHz	availMHz
0	host	0	933	1600	1600	1600	pct	4096000	5232	33168
1	system	0	659	10	-1	-1	pct	500	288	33168
2	vim	0	271	4944	-1	-1	mhz	500	4344	33768
3	iofilters	0	3	0	-1	-1	pct	1000	0	33168
4	user	0	0	0	-1	-1	pct	9000	0	33168

Tools

cmdline for local groups (with VMs)

sched-stats

vmgid	name	pgid	vsmps	amin	amax	minLimit	units	ashares	resvMHz	availMHz
0	host	0	995	1600	1600	1600	pct	4096000	5232	33168
1	system	0	664	10	-1	-1	pct	500	288	33168
2	vim	0	268	4944	-1	-1	mhz	500	4344	33768
3	iofilters	0	3	0	-1	-1	pct	1000	0	33168
4	user	0	60	0	-1	-1	pct	9000	0	33168
148269	vm.2118564	14605942	12	0	-1	-1	mhz	2000	0	33168
148279	vm.2118565	13787844	12	0	-1	-1	mhz	2000	0	33168
149779	vm.2118803	13787842	12	0	-1	-1	mhz	2000	0	33168
149810	vm.2118821	13787844	12	0	-1	-1	mhz	2000	0	33168
314753	vm.2139855	13787843	12	0	-1	-1	mhz	2000	0	33168
14605942	pool4	13787844	12	0	-1	-1	mhz	4000	0	33168
13787842	pool1	4	12	0	-1	-1	mhz	4000	0	33168
13787843	pool2	4	12	0	-1	-1	mhz	4000	0	33168
13787844	pool3	4	36	0	-1	-1	mhz	1142	0	33168

Tools

cmdline for local groups (with VMs)

sched-stats

vmgid	name	pgid	vsmps	amin	amax	minLimit	units	ashares	resvMHz	availMHz
0	host	0	995	1600	1600	1600	pct	4096000	5232	33168
1	system	0	664	10	-1	-1	pct	500	288	33168
2	vim	0	268	4944	-1	-1	mhz	500	4344	33768
3	iofilters	0	3	0	-1	-1	pct	1000	0	33168
4	user	0	60	0	-1	-1	pct	9000	0	33168
148269	vm.2118564	14605942	12	0	-1	-1	mhz	2000	0	33168
148279	vm.2118565	13787844	12	0	-1	-1	mhz	2000	0	33168
149779	vm.2118803	13787842	12	0	-1	-1	mhz	2000	0	33168
149810	vm.2118821	13787844	12	0	-1	-1	mhz	2000	0	33168
314753	vm.2139855	13787843	12	0	-1	-1	mhz	2000	0	33168
14605942	pool4	13787844	12	0	-1	-1	mhz	4000	0	33168
13787842	pool1	4	12	0	-1	-1	mhz	4000	0	33168
13787843	pool2	4	12	0	-1	-1	mhz	4000	0	33168
13787844	pool3	4	36	0	-1	-1	mhz	1142	0	33168

Tools

cmdline for local groups (with VMs)

sched-stats

vmgid	name	pgid	vsmps	amin	amax	minLimit	units	ashares	resvMHz	availMHz
0	host	0	995	1600	1600	1600	pct	4096000	5232	33168
1	system	0	664	10	-1	-1	pct	500	288	33168
2	vim	0	268	4944	-1	-1	mhz	500	4344	33768
3	iofilters	0	3	0	-1	-1	pct	1000	0	33168
4	user	0	60	0	-1	-1	pct	9000	0	33168
148269	vm.2118564	14605942	12	0	-1	-1	mhz	2000	0	33168
148279	vm.2118565	13787844	12	0	-1	-1	mhz	2000	0	33168
149779	vm.2118803	13787842	12	0	-1	-1	mhz	2000	0	33168
149810	vm.2118821	13787844	12	0	-1	-1	mhz	2000	0	33168
314753	vm.2139855	13787843	12	0	-1	-1	mhz	2000	0	33168
14605942	pool4	13787844	12	0	-1	-1	mhz	4000	0	33168
13787842	pool1	4	12	0	-1	-1	mhz	4000	0	33168
13787843	pool2	4	12	0	-1	-1	mhz	4000	0	33168
13787844	pool3	4	36	0	-1	-1	mhz	1142	0	33168

Tools

VM name / CPU information

sched-stats

```
# esxcli vm process list | grep -B3 -A3 2118564
RAID5-VM8
  World ID: 2118566
  Process ID: 0
  VMX Cartel ID: 2118564
  UUID: 42 09 74 3e 4d f5 23 86-41 82 44 84 d3 17 ef b3
  Display Name: RAID5-VM8
  Config File: /vmfs/volumes/vsan:(...)/loadtestvm-8.vmx
```

Tools

VM name / CPU information

sched-stats

```
# esxcli vm process list | grep -B3 -A3 2118564
RAID5-VM8
  World ID: 2118566
  Process ID: 0
  VMX Cartel ID: 2118564
  UUID: 42 09 74 3e 4d f5 23 86-41 82 44 84 d3 17 ef b3
  Display Name: RAID5-VM8
  Config File: /vmfs/volumes/vsan:(...)/loadtestvm-8.vmx
```

```
# sched-stats -t ncpus
32 PCPUs
16 cores
 2 packages
 1 NUMA nodes
```


Tools

VM name / CPU information

```
# esxcli vm process list | grep -B3 -A3 2118564
RAID5-VM8
  World ID: 2118566
  Process ID: 0
  VMX Cartel ID: 2118564
  UUID: 42 09 74 3e 4d f5 23 86-41 82 44 84 d3 17 ef b3
  Display Name: RAID5-VM8
  Config File: /vmfs/volumes/vsan:(...)/loadtestvm-8.vmx
```

```
# sched-stats -t ncpus
32 PCPUs
16 cores
 2 packages
 1 NUMA nodes
```

```
# vsish -e get /hardware/cpu/cpuModelName
Intel(R) Xeon(R) CPU E5-2630 v3 @ 2.40GHz
```

Tools

cmdline for specific VM group

sched-stats

```
# sched-stats -t cpu | awk 'NR == 1  
|| $(NF-5) ~ /^vm\.2118564/  
{printf ("%8s %5s %25s %5s %5s %7s %8s %11s %5s %5s\n"  
, $1, $3, $4, $13, $14, $15, $16, $17, $19, $22)}'
```

Tools

cmdline for specific VM group

sched-stats

```
# sched-stats -t cpu | awk 'NR == 1
|| $(NF-5) ~ /^vm\.2118564/
{printf ("%8s %5s %25s %5s %5s %7s %8s %11s %5s %5s\n"
,$1, $3, $4, $13, $14, $15, $16, $17, $19, $22)}'
```

vcpu	type	name	min	max	units	shares	group	cpu	affinity
2118564	U	vmx	0	-1	pct	1000	vm.2118564	16	0-31
2118568	SA	NUMASchedRemapEpochI	0	-1	pct	1000	vm.2118564	26	0-31
2118574	SA	vmast.2118566	0	-1	pct	100	vm.2118564	28	0-31
2118579	U	vmx-vthread-211	0	-1	pct	1000	vm.2118564	5	0-31
2118581	U	vmx-filtPoll:RAID5-V	0	-1	pct	1000	vm.2118564	19	0-31
2118582	U	vmx-vthread-211:RAID	0	-1	pct	1000	vm.2118564	26	0-31
2118586	U	vmx-mks:RAID5-VM8	0	-1	pct	1000	vm.2118564	28	0-31
2118588	U	vmx-svga:RAID5-VM8	0	-1	pct	1000	vm.2118564	18	0-31
2118590	U	vmx-vcpu-0:RAID5-VM8	0	-1	pct	2000	vm.2118564	15	0-15
2118593	U	vmx-vcpu-1:RAID5-VM8	0	-1	pct	2000	vm.2118564	6	0-15
2118591	SA	PVSCSI-2118566:0	0	-1	pct	1000	vm.2118564	6	0-31
2118592	SA	PVSCSI-2118566:1	0	-1	pct	1000	vm.2118564	13	0-31
2118567	SA	NetWorld-VM-2118566	0	-1	pct	2000	vm.2118564	17	0-31

Tools

cmdline for specific VM group

sched-stats

```
# sched-stats -t cpu | awk 'NR == 1
|| $(NF-5) ~ /^vm\.2118564/
{printf ("%8s %-5s %-25s %-5s %-5s %-7s %-8s %-11s %-5s %-5s\n"
,$1, $3, $4, $13, $14, $15, $16, $17, $19, $22)}'
```

vcpu	type	name	min	max	units	shares	group	cpu	affinity
2118564	U	vmx	0	-1	pct	1000	vm.2118564	16	0-31
2118568	SA	NUMASchedRemapEpochI	0	-1	pct	1000	vm.2118564	26	0-31
2118574	SA	vmast.2118566	0	-1	pct	100	vm.2118564	28	0-31
2118579	U	vmx-vthread-211	0	-1	pct	1000	vm.2118564	5	0-31
2118581	U	vmx-filtPoll:RAID5-V	0	-1	pct	1000	vm.2118564	19	0-31
2118582	U	vmx-vthread-211:RAID	0	-1	pct	1000	vm.2118564	26	0-31
2118586	U	vmx-mks:RAID5-VM8	0	-1	pct	1000	vm.2118564	28	0-31
2118588	U	vmx-svga:RAID5-VM8	0	-1	pct	1000	vm.2118564	18	0-31
2118590	U	vmx-vcpu-0:RAID5-VM8	0	-1	pct	2000	vm.2118564	15	0-15
2118593	U	vmx-vcpu-1:RAID5-VM8	0	-1	pct	2000	vm.2118564	6	0-15
2118591	SA	PVSCSI-2118566:0	0	-1	pct	1000	vm.2118564	6	0-31
2118592	SA	PVSCSI-2118566:1	0	-1	pct	1000	vm.2118564	13	0-31
2118567	SA	NetWorld-VM-2118566	0	-1	pct	2000	vm.2118564	17	0-31

VM Resource Control

Maximum Reservation

The screenshot shows the 'Edit Settings' window for a VM named 'RAID1-VM2'. The 'Virtual Hardware' tab is selected. Under the 'CPU' section, the 'Reservation' is set to 0 MHz and the 'Limit' is also set to 0 MHz. A tooltip is displayed over the 'Limit' field, showing the current value (0 MHz), the minimum (0 MHz), and the maximum (99,504 MHz). Other settings include 'Cores per Socket' (1), 'Sockets' (2), and 'CPU Hot Plug' (disabled). An 'ADD NEW DEVICE' button is visible in the top right. The 'CPUID Mask' section is partially visible at the bottom, showing 'Expose the NX/XD flag to guest' and an 'Advanced...' link.

Setting	Value
CPU	2
Cores per Socket	1
Sockets	2
CPU Hot Plug	Enable CPU Hot Add (disabled)
Reservation	0 MHz
Limit	0 MHz
Shares	
CPUID Mask	Expose the NX/XD flag to guest

VM Resource Control

Maximum Reservation

The screenshot shows the 'Edit Settings' window for a VM named 'RAID1-VM2'. The 'Virtual Hardware' tab is selected. Under the 'CPU' section, the 'Reservation' field is set to 0 MHz. A dropdown menu is open, showing the following values:

Current value:	0 MHz
Minimum:	0 MHz
Maximum:	99,504 MHz

The 'Maximum' value of 99,504 MHz is highlighted with a red box. Other settings visible include 'Cores per Socket' (1), 'Sockets' (2), 'CPU Hot Plug' (disabled), 'Limit' (0 MHz), 'Shares' (1), and 'CPUID Mask' (Expose the NX/XD flag to guest).

VM Resource Control

Maximum Reservation

The screenshot shows the 'Edit Settings' window for a VM named 'RAID1-VM2'. The 'Virtual Hardware' tab is selected. Under the 'CPU' section, the number of vCPUs is set to 2. The 'Reservation' field is set to 0 MHz. A tooltip is displayed over the 'Reservation' field, showing the current value (0 MHz), the minimum (0 MHz), and the maximum (99,504 MHz). The 'Maximum' value is highlighted with a red box. Other settings visible include 'Cores per Socket' (1), 'Sockets' (2), 'CPU Hot Plug' (disabled), 'Limit' (0 MHz), 'Shares' (0), and 'CPUID Mask' (Advanced...).

2 vCPUs x 2.4 GHz = **4.8 GHz** (maximum reservation the VM powers on with)

Tools

cmdline for identifying CPU reservations

sched-stats

```
# sched-stats -t groups | awk 'NR == 1
|| $8 !~ /^0/
{printf ("% -10s% -12s% -9s% -6s% -6s% -6s% -9s% -6s% -9s% -9s% -10s\n"
,$1, $2, $3, $6, $8, $9, $10, $11, $12, $13, $14)}'
```


Tools

cmdline for identifying CPU reservations

```
# sched-stats -t groups | awk 'NR == 1
|| $8 !~ /^0/
{printf ("%-10s%-12s%-9s%-6s%-6s%-6s%-9s%-6s%-9s%-9s%-10s\n"
,$1, $2, $3, $6, $8, $9, $10, $11, $12, $13, $14)}'
```

vmgid	name	pgid	vsmgs	amin	amax	minLimit	units	ashares	resvMHz	availMHz
0	host	0	995	1600	1600	1600	pct	4096000	5232	33168
1	system	0	664	10	-1	-1	pct	500	288	33168
2	vim	0	268	4944	-1	-1	mhz	500	4344	33768
8	helper	1	269	3	-1	-1	pct	1000	0	33240
9	lfhelper	1	0	3	-1	-1	pct	1000	0	33240
10	drivers	1	12	3	-1	-1	pct	1000	0	33240
15	vmvisor	2	263	35	-1	-1	pct	1000	4344	33768
271	vmkapimod	1	0	3	-1	-1	pct	1000	0	33240
758	hostd	15	53	3192	-1	-1	mhz	1000	0	36960
777	logging	15	4	432	-1	-1	mhz	1000	0	34200
809	vpva	15	39	720	-1	-1	mhz	1000	0	34488

Tools

cmdline for identifying CPU reservations

```
# sched-stats -t groups | awk 'NR == 1
|| $8 !~ /^0/
{printf ("%10s%-12s%-9s%-6s%-6s%-6s%-9s%-6s%-9s%-9s%-10s\n"
,$1, $2, $3, $6, $8, $9, $10, $11, $12, $13, $14)}'
```

vmgid	name	pgid	vsmgs	amin	amax	minLimit	units	ashares	resvMHz	availMHz
0	host	0	995	1600	1600	1600	pct	4096000	5232	33168
1	system	0	664	10	-1	-1	pct	500	288	33168
2	vim	0	268	4944	-1	-1	mhz	500	4344	33768
8	helper	1	269	3	-1	-1	pct	1000	0	33240
9	lfhelper	1	0	3	-1	-1	pct	1000	0	33240
10	drivers	1	12	3	-1	-1	pct	1000	0	33240
15	vmvisor	2	263	35	-1	-1	pct	1000	4344	33768
271	vmkapimod	1	0	3	-1	-1	pct	1000	0	33240
758	hostd	15	53	3192	-1	-1	mhz	1000	0	36960
777	logging	15	4	432	-1	-1	mhz	1000	0	34200
809	vpva	15	39	720	-1	-1	mhz	1000	0	34488

Why the high hostd / vpxa reservation?

Having a look around ..

```
# egrep -o "host/vim/vmvisor/hostd\".*" /usr/lib/vmware/rp/bin/configRP
```

Why the high hostd / vpxa reservation?

Having a look around ..

```
# egrep -o "host/vim/vmvisor/hostd\".*" /usr/lib/vmware/rp/bin/configRP
```

```
host/vim/vmvisor/hostd",  
memMin=getHostdMemMax(),  
memMax=getHostdMemMax(),  
cpuShares=1000,  
cpuMin=getHostdCpuMin(),  
(...)
```

Why the high hostd / vpxa reservation?

Having a look around ..

```
# egrep -o "host/vim/vmvisor/hostd\".*" /usr/lib/vmware/rp/bin/configRP
```

```
host/vim/vmvisor/hostd",  
memMin=getHostdMemMax(),  
memMax=getHostdMemMax(),  
cpuShares=1000,  
cpuMin=getHostdCpuMin(),  
(...)
```

Why the high hostd / vpxa reservation?

Having a look around ..

```
# egrep -A4 "def getHostdCpuMin" /usr/lib/vmware/rp/bin/configRP
```

Why the high hostd / vpxa reservation?

Having a look around ..

```
# egrep -A4 "def getHostdCpuMin" /usr/lib/vmware/rp/bin/configRP

def getHostdCpuMin():
    """@return The cpu reservation for hostd
    """
    # PR562745
    return 25 + int(ceil(0.21 * getNumVMs()))
```

Why the high hostd / vpxa reservation?

Having a look around ..

```
# egrep -A4 "def getHostdCpuMin" /usr/lib/vmware/rp/bin/configRP
```

```
def getHostdCpuMin():  
    """@return The cpu reservation for hostd  
    """  
    # PR562745  
    return 25 + int(ceil(0.21 * getNumVMs()))
```

```
# egrep -A4 "def getNumVMs" /usr/lib/vmware/rp/bin/configRP
```


Why the high hostd / vpxa reservation?

Having a look around ..

```
# egrep -A4 "def getHostdCpuMin" /usr/lib/vmware/rp/bin/configRP
```

```
def getHostdCpuMin():  
    """@return The cpu reservation for hostd  
    """  
    # PR562745  
    return 25 + int(ceil(0.21 * getNumVMs()))
```

```
# egrep -A4 "def getNumVMs" /usr/lib/vmware/rp/bin/configRP
```

```
def getNumVMs():  
    """@return The number of supported VMs  
    """  
    return vsi.get("/system/supportedVMs")
```

Why the high hostd / vpxa reservation?

Having a look around ..

```
# egrep -A4 "def getHostdCpuMin" /usr/lib/vmware/rp/bin/configRP
```

```
def getHostdCpuMin():  
    """@return The cpu reservation for hostd  
    """  
    # PR562745  
    return 25 + int(ceil(0.21 * getNumVMs()))
```

```
# egrep -A4 "def getNumVMs" /usr/lib/vmware/rp/bin/configRP
```

```
def getNumVMs():  
    """@return The number of supported VMs  
    """  
    return vsi.get("/system/supportedVMs")
```

```
# vsish -e get /system/supportedVMs
```

Why the high hostd / vpxa reservation?

Having a look around ..

```
# egrep -A4 "def getHostdCpuMin" /usr/lib/vmware/rp/bin/configRP
```

```
def getHostdCpuMin():  
    """@return The cpu reservation for hostd  
    """  
    # PR562745  
    return 25 + int(ceil(0.21 * getNumVMs()))
```

```
# egrep -A4 "def getNumVMs" /usr/lib/vmware/rp/bin/configRP
```

```
def getNumVMs():  
    """@return The number of supported VMs  
    """  
    return vsi.get("/system/supportedVMs")
```

```
# vsish -e get /system/supportedVMs  
512
```

Why the high hostd / vpxa reservation?

Having a look around ..

| Usually never an issue *

We are looking into it though

Tools

cmdline for local groups (with VMs)

memstats

Tools

cmdline for local groups (with VMs)

memstats

```
# memstats -r group-stats
```

Tools

cmdline for local groups (with VMs)

memstats

```
# memstats -r group-stats  
-g0 -l2
```

Tools

cmdline for local groups (with VMs)

memstats

```
# memstats -r group-stats  
-g0 -l2  
-s gid:min:max:minlimit:shares:conResv:availResv:memSize
```


Tools

cmdline for local groups (with VMs)

memstats

```
# memstats -r group-stats  
-g0 -l2  
-s gid:min:max:minlimit:shares:conResv:availResv:memSize  
-u mb
```

Tools

cmdline for local groups (with VMs)

memstats

```
# memstats -r group-stats
-g0 -l2
-s gid:min:max:minlimit:shares:conResv:availResv:memSize
-u mb
| sed -n '/^-\+/,/.*\n/p'
```

Tools

cmdline for local groups (with VMs)

```
# memstats -r group-stats  
-g0 -l2  
-s gid:min:max:minlimit:shares:conResv:availResv:memSize  
-u mb  
| sed -n '/^-+/,/.*\n/p'
```

gid	min	max	minLimit	shares	conResv	availResv	memSize
0	97823	97823	97823	2147483647	28916	68908	24665
1	20023	-1	-1	500	20007	68924	18645
2	0	-1	-1	500	3378	68908	588
3	0	-1	-1	500	25	68908	9
4	0	-1	-1	9000	5490	68908	5425

Tools

cmdline for local groups (with VMs)

```
# memstats -r group-stats
-g0 -l2
-s gid:min:max:minlimit:shares:conResv:availResv:memSize
-u mb
| sed -n '/^-\\+/,/.*\\n/p'
```

gid	min	max	minLimit	shares	conResv	availResv	memSize
0	97823	97823	97823	2147483647	28916	68908	24665
1	20023	-1	-1	500	20007	68924	18645
2	0	-1	-1	500	3378	68908	588
3	0	-1	-1	500	25	68908	9
4	0	-1	-1	9000	5490	68908	5425

Tools

cmdline for local groups (with VMs)

```
# memstats -r group-stats
-g0 -l2
-s gid:min:max:minlimit:shares:conResv:availResv:memSize
-u mb
| sed -n '/^-+/,/.*\n/p'
```

gid	min	max	minLimit	shares	conResv	availResv	memSize
0	97823	97823	97823	2147483647	28916	68908	24665
1	20023	-1	-1	500	20007	68924	18645
2	0	-1	-1	500	3378	68908	588
3	0	-1	-1	500	25	68908	9
4	0	-1	-1	9000	5490	68908	5425

Tools

cmdline for local groups (with VMs)

```
# memstats -r group-stats
-g0 -l2
-s gid:min:max:minlimit:shares:conResv:availResv:memSize
-u mb
| sed -n '/^-\+/,/.*\n/p'
```

gid	min	max	minLimit	shares	conResv	availResv	memSize
0	97823	97823	97823	2147483647	28916	68908	24665
1	20023	-1	-1	500	20007	68924	18645
2	0	-1	-1	500	3378	68908	588
3	0	-1	-1	500	25	68908	9
4	0	-1	-1	9000	5490	68908	5425

Tools

cmdline for local groups (with VMs)

```
# memstats -r group-stats  
-g0 -l2  
-s gid:name:min:max::conResv:availResv  
-u mb  
| sed -n '/^-\\+/,/.*\\n/p'
```

gid	name	min	max	conResv	availResv
0	host	97823	97823	28917	68907
1	system	20024	-1	20008	68923
2	vim	0	-1	3378	68907
3	iofilters	0	-1	25	68907
4	user	0	-1	5490	68907

Tools

cmdline for local groups (with VMs)

```
# memstats -r group-stats
-g0 -l2
-s gid:name:min:max::conResv:availResv
-u mb
| sed -n '/^-\+/,/.*\n/p'
```

gid	name	min	max	conResv	availResv
0	host	97823	97823	28917	68907
1	system	20024	-1	20008	68923
2	vim	0	-1	3378	68907
3	iofilters	0	-1	25	68907
4	user	0	-1	5490	68907

Tools

cmdline for local groups (with VMs)

```
# memstats -r group-stats
-g0 -l2
-s gid:name:min:max::conResv:availResv
-u mb
| sed -n '/^-\+/,/.*\n/p'
```

gid	name	min	max	conResv	availResv
0	host	97823	97823	28917	68907
1	system	20024	-1	20008	68923
2	vim	0	-1	3378	68907
3	iofilters	0	-1	25	68907
4	user	0	-1	5490	68907

Tools

cmdline for local groups under user (with VMs)

```
# memstats -r group-stats  
-g4 -l2  
-s gid:name:min:max:conResv:availResv  
-u mb  
| sed -n '/^-\\+/,/.*\\n/p'
```

Tools

cmdline for local groups under user (with VMs)

```
# memstats -r group-stats  
-g4 -l2  
-s gid:name:min:max:conResv:availResv  
-u mb  
| sed -n '/^-\+/,/.*\n/p'
```

gid	name	min	max	conResv	availResv
4	user	0	-1	5490	68906
13787842	pool1	1098	-1	1065	68939
13787843	pool2	1098	-1	1065	68939
13787844	pool3	3294	-1	3227	68973

Tools

cmdline for local groups under user (with VMs)

```
# memstats -r group-stats  
-g4 -l2  
-s gid:name:min:max:conResv:availResv  
-u mb  
| sed -n '/^-\\+/,/\\. *\\n/p'
```

gid	name	min	max	conResv	availResv
4	user	0	-1	5490	68906
13787842	pool1	1098	-1	1065	68939
13787843	pool2	1098	-1	1065	68939
13787844	pool3	3294	-1	3227	68973

Tools

cmdline for local groups under user (with VMs)

memstats

```
# memstats -r group-stats  
-g4 -l3  
-s gid:name:min:max:conResv:availResv  
-u mb  
| sed -n '/^-\+/,/.*\n/p'
```

Tools

cmdline for local groups under user (with VMs)

```
# memstats -r group-stats
-g4 -l3
-s gid:name:min:max:conResv:availResv
-u mb
| sed -n '/^-\\+/,/.*\\n/p'
```

gid	name	min	max	conResv	availResv
4	user	0	-1	5490	68904
13787842	pool1	1098	-1	1065	68937
149779	vm.2118803	0	-1	1065	34
13787843	pool2	1098	-1	1065	68937
314753	vm.2139855	0	-1	1065	34
13787844	pool3	3294	-1	3227	68971
149810	vm.2118821	0	-1	1065	34
148279	vm.2118565	0	-1	1065	34
14605942	pool4	1098	-1	1065	69004

Tools

cmdline for local groups under user (with VMs)

```
# memstats -r group-stats  
-g4 -l3  
-s gid:name:min:max:conResv:availResv  
-u mb  
| sed -n '/^-\\+/,/.*\\n/p'
```

gid	name	min	max	conResv	availResv
4	user	0	-1	5490	68904
13787842	pool1	1098	-1	1065	68937
149779	vm.2118803	0	-1	1065	34
13787843	pool2	1098	-1	1065	68937
314753	vm.2139855	0	-1	1065	34
13787844	pool3	3294	-1	3227	68971
149810	vm.2118821	0	-1	1065	34
148279	vm.2118565	0	-1	1065	34
14605942	pool4	1098	-1	1065	69004

Tools

Per VM (“container”) limit / reservation

```
9:49:14am up 7 days 21:17, 1001 worlds, 5 VMs, 10 vCPUs; MEM overcommit avg: 0.00, 0.00, 0.00
PMEM /MB: 98208 total: 17053 vmk,5400 other, 75754 free
VMKMEM/MB: 97822 managed: 1592 minfree, 28915 rsvd, 68907 ursvd, high state
NUMA /MB: 49054 (36855), 49152 (38514)
PSHARE/MB: 68 shared, 51 common: 17 saving
SWAP /MB: 0 curr, 0 rclmtgt: 0.00 r/s, 0.00 w/s
ZIP /MB: 0 zipped, 0 saved
MEMCTL/MB: 0 curr, 0 target, 3238 max
```

GID	NAME	AMIN	AMAX	ASHRS	AMLMT	AUNITS
148269	RAID5-VM8	0	-1	-3	1124352	kb
149779	RAID1-VM2	0	-1	-3	1124352	kb
148279	RAID5-VM3	0	-1	-3	1124352	kb
314753	RAID1-VM3	0	-1	-3	1124352	kb
149810	RAID1-VM8	0	-1	-3	1124352	kb

Tools

Per VM (“container”) limit / reservation

```
9:49:14am up 7 days 21:17, 1001 worlds, 5 VMs, 10 vCPUs; MEM overcommit avg: 0.00, 0.00, 0.00
PMEM /MB: 98208 total: 17053 vmk,5400 other, 75754 free
VMKMEM/MB: 97822 managed: 1592 minfree, 28915 rsvd, 68907 ursvd, high state
NUMA /MB: 49054 (36855), 49152 (38514)
PSHARE/MB: 68 shared, 51 common: 17 saving
SWAP /MB: 0 curr, 0 rclmtgt: 0.00 r/s, 0.00 w/s
ZIP /MB: 0 zipped, 0 saved
MEMCTL/MB: 0 curr, 0 target, 3238 max
```

GID	NAME	AMIN	AMAX	ASHRS	AMLMT	AUNITS
148269	RAID5-VM8	0	-1	-3	1124352	kb
149779	RAID1-VM2	0	-1	-3	1124352	kb
148279	RAID5-VM3	0	-1	-3	1124352	kb
314753	RAID1-VM3	0	-1	-3	1124352	kb
149810	RAID1-VM8	0	-1	-3	1124352	kb

Tools

cmdline for a specific pool with flags

memstats

```
# memstats -r group-stats  
-g13787842 -l3  
-s gid:name:min:max:conResv:availResv  
-u mb  
-i pre:vm:vmm:ug  
| sed -n '/^-\+/,/.*\n/p'
```

Tools

cmdline for a specific pool with flags

```
# memstats -r group-stats
-g13787842 -l3
-s gid:name:min:max:conResv:availResv
-u mb
-i pre:vm:vmm:ug
| sed -n '/^-\+/,/.*\n/p'
```

gid	name	min	max	conResv	availResv
13787842	pool1	1098	-1	1065	68937
149779	vm.2118803	0	-1	1065	34
149789	vmm.2118803	1024	-1	0	1058

Tools

cmdline for a specific pool with flags

```
# memstats -r group-stats
-g13787842 -l3
-s gid:name:min:max:conResv:availResv
-u mb
-i pre:vm:vmm:ug
| sed -n '/^-\\+/,/\\.\\*\\n/p'
```

gid	name	min	max	conResv	availResv
13787842	pool1	1098	-1	1065	68937
149779	vm.2118803	0	-1	1065	34
149789	vmm.2118803	1024	-1	0	1058

Tools

cmdline for a specific pool with world flags

```
# memstats -r group-stats -g149779 -l2 -s gid:name:min:max:conResv:availResv -u mb | etc.
```

Tools

cmdline for a specific pool with world flags

```
# memstats -r group-stats -g149779 -l2 -s gid:name:min:max:conResv:availResv -u mb | etc.
```

gid	name	min	max	conResv	availResv
149779	vm.2118803	0	-1	1065	34
149780	worldGroup.2118803	0	-1	0	34
149781	uw.2118803	19	-1	0	52
149782	vsiHeap.2118803	0	-1	0	34
149783	pt.2118803	1	-1	0	35
149784	cartelheap.2118803	0	-1	0	34
149785	uwshmempt.2118803	0	-1	0	34
149786	uwAsyncRemapHeap.2118803	0	-1	0	34
149787	uwCrypt.2118803	4	4	0	4
149788	uwregbmp.2118803	1	-1	0	34
149789	vmm.2118803	1024	-1	0	1058
149790	vmmanon.2118803	17	-1	0	50
149791	vmcpt.2118804	1	-1	0	34
149792	vmmregbmp.2118803	0	-1	0	34
149793	vmmAsyncRemapHeap.2118803	0	-1	0	34

Tools

cmdline for a specific pool with world flags

```
# memstats -r group-stats -g149779 -l2 -s gid:name:min:max:conResv:availResv -u mb | etc.
```

gid	name	min	max	conResv	availResv
149779	vm.2118803	0	-1	1065	34
149780	worldGroup.2118803	0	-1	0	34
149781	uw.2118803	19	-1	0	52
149782	vsiHeap.2118803	0	-1	0	34
149783	pt.2118803	1	-1	0	35
149784	cartelheap.2118803	0	-1	0	34
149785	uwshmempt.2118803	0	-1	0	34
149786	uwAsyncRemapHeap.2118803	0	-1	0	34
149787	uwCrypt.2118803	4	4	0	4
149788	uwregbmp.2118803	1	-1	0	34
149789	vmm.2118803	1024	-1	0	1058
149790	vmmanon.2118803	17	-1	0	50
149791	vmcpt.2118804	1	-1	0	34
149792	vmmregbmp.2118803	0	-1	0	34
149793	vmmAsyncRemapHeap.2118803	0	-1	0	34

Tools

cmdline for any world's reservations

memstats

```
# memstats -r group-stats -l7 -s gid:name:min -u mb | sed -n '/^-\\+/,/.*\\n/p'  
| awk 'NR == 3 || $3 !~ /^0/'  
{printf ("%10s%-50s%-9s%-6s%-6s%-6s\\n", $1, $2, $3)}
```


Tools

cmdline for any world's reservations

memstats

```
# memstats -r group-stats -l7 -s gid:name:min -u mb | sed -n '/^-\\+/,/.*\\n/p'  
| awk 'NR == 3 || $3 !~ /^0/'  
{printf ("% -10s% -50s% -9s% -6s% -6s% -6s\\n", $1, $2, $3)}
```

Tools

cmdline for any world's reservations

```
# memstats -r group-stats -l7 -s gid:name:min -u mb | sed -n '/^-\+/,/.*\n/p'  
| awk 'NR == 3 || $3 !~ /^0/'  
{printf ("%10s%-50s%-9s%-6s%-6s%-6s\n", $1, $2, $3)}
```

```
-----  
gid          name                                          min  
-----  
0            host                                          97823  
1            system                                       20025  
(...)  
66644       dom-Owner-traceTblSlab-0x4300a6c8c140        67  
66645       dom-Owner-objSlab-0x4300a6c8c440            1463  
66646       dom-Owner-subscrEntrySlab-0x4300a6c8c740     201  
66647       dom-Owner-rdtHdrSlab-0x4300a6c8ca40         9  
(...)  
109524      LSOM_UnmapLsnEntrySlab-0x4300a6c94540       243  
109527      LSOMDiskGroup_002                           7737  
(...)
```

Tools

cmdline for any world's reservations

```
# memstats -r group-stats -l7 -s gid:name:min -u mb | sed -n '/^-\+/,/.*\n/p'  
| awk 'NR == 3 || $3 !~ /^0/  
{printf ("%10s%-50s%-9s%-6s%-6s%-6s\n", $1, $2, $3)}'
```

gid	name	min
0	host	97823
1	system	20025
(...)		
66644	dom-Owner-traceTblSlab-0x4300a6c8c140	67
66645	dom-Owner-objSlab-0x4300a6c8c440	1463
66646	dom-Owner-subscrEntrySlab-0x4300a6c8c740	201
66647	dom-Owner-rdtHdrSlab-0x4300a6c8ca40	9
(...)		
109524	LSOM_UnmapLsnEntrySlab-0x4300a6c94540	243
109527	LSOMDiskGroup_002	7737
(...)		

Tools

Why doesn't e.g. vSAN use CPU reservations?

```
# /tmp/foo.sh
host/system logSysAlert
host/system serialLogger
host/system netLegacyRx
host/system NetEventWorld
host/system netCoalesce2World
(...)
host/system VSAN_0x430e986f2fe8_Client
host/system VSAN_0x430e986ee298_Client
host/system VSAN_0x430e986efc18_Client
host/system VSAN_0x430fff7e3418_Owner
(...)
host/system VSAN_0x430f24818f18_PLOG
host/system VSAN_0x430f2481d618_PLOG
host/system VSAN_0x430f25070598_LSOMLLOG
host/system VSAN_0x430f25070918_LSOMLLOG
```

Tools

Why doesn't e.g. vSAN use CPU reservations?

```
# cat /tmp/foo.sh
for vcpuid in $(vsish -e ls /sched/Vcpus/)
do
    priority=$(vsish -e get /sched/Vcpus/${vcpuid}stats/summaryStats | (linebreak)
    sed -n 's/^    highPriority:\([0-9]\)$/\1/p') 2> /dev/null
    if [ ${priority} ] && [ ${priority} == 1 ]
    then
        gpname=$(vsish -e get /sched/Vcpus/${vcpuid}groupPathName)
        wname=$(vsish -e get /sched/Vcpus/${vcpuid}worldName)
        echo -e "${gpname} ${wname}"
    fi
done
```

Tools

Why doesn't e.g. vSAN use CPU reservations?

```
# cat /tmp/foo.sh
for vcpuid in $(vsish -e ls /sched/Vcpus/)
do
    priority=$(vsish -e get /sched/Vcpus/${vcpuid}stats/summaryStats | (linebreak)
    sed -n 's/^ highPriority:\([0-9]\)\$/\1/p') 2> /dev/null
    if [ ${priority} ] && [ ${priority} == 1 ]
    then
        gpname=$(vsish -e get /sched/Vcpus/${vcpuid}groupPathName)
        wname=$(vsish -e get /sched/Vcpus/${vcpuid}worldName)
        echo -e "${gpname} ${wname}"
    fi
done
```

Changes in vSphere 6.7

New in 6.7

Resource Pool entitlement is now calculated with consumed then active

Changes in vSphere 6.7

New in 6.7

Resource Pool entitlement is now calculated with consumed then active

Global to Local distribution happens every minute

Changes in vSphere 6.7

New in 6.7

Resource Pool entitlement is now calculated with consumed then active

Global to Local distribution happens every minute

Local Pool reservations are distributed more eagerly

Changes in vSphere 6.7

New in 6.7

Resource Pool entitlement is now calculated with consumed then active

Global to Local distribution happens every minute

Local Pool reservations are distributed more eagerly

A CPU reservation now ensures core throughput if behind vtime



Thank You