

VMware vSphere Bitfusion User Guide

11 MAY 2021

Updated to include VMware vSphere Bitfusion 3.5
VMware vSphere Bitfusion 3.0

You can find the most up-to-date technical documentation on the VMware website at:

<https://docs.vmware.com/>

VMware, Inc.
3401 Hillview Ave.
Palo Alto, CA 94304
www.vmware.com

Copyright © 2020-2021 VMware, Inc. All rights reserved. [Copyright and trademark information.](#)

Contents

1	About <i>VMware vSphere Bitfusion User Guide</i>	5
2	Understanding VMware vSphere Bitfusion	6
3	Starting Applications in vSphere Bitfusion	10
	Starting Applications with the Run Command	10
	Allocating GPUs with the RUN Command	11
	Partitioning GPU Memory	11
	GPU Partitioning Examples	12
	Starting Applications with Reserved GPUs	13
4	Managing vSphere Bitfusion Servers	16
	Add Subsequent vSphere Bitfusion Servers	16
	Remove a vSphere Bitfusion Server	19
	Configuring the Network Settings of a vSphere Bitfusion Server	20
	Add a Network Interface	20
	Configure a Network Interface	21
	Remove a Network Interface	22
	vSphere Bitfusion vApp Properties	23
	Change the Settings of a vSphere Bitfusion Server	25
	Perform a Health Check of a vSphere Bitfusion Server	26
	vSphere Bitfusion Health Checks List	27
	Create vSphere Bitfusion Server Logs	28
	View vSphere Bitfusion Server Logs	28
	View GPU Information for a vSphere Bitfusion Server	29
5	Managing vSphere Bitfusion Clients	30
	Disable or Delete a vSphere Bitfusion Client	30
	Change the Settings of a vSphere Bitfusion Client	31
	View GPU Information for a vSphere Bitfusion Client	31
6	Managing vSphere Bitfusion	32
	Back Up a vSphere Bitfusion Cluster	33
	Restore a vSphere Bitfusion Cluster	34
	Start and Stop the vSphere Bitfusion Service	34
	Download vSphere Bitfusion Monitoring Data	35
	Set a Global Display Refresh Interval	36
	Use a Subset List of vSphere Bitfusion Servers	36

vSphere Bitfusion Configuration Files	37
vSphere Bitfusion Commands Reference	38
Monitoring vSphere Bitfusion in the vSphere Bitfusion Plug-In	41
Monitoring vSphere Bitfusion in the CLI	42

7 Troubleshooting vSphere Bitfusion 44

vSphere Bitfusion Client ID Changes	44
Deleted vSphere Bitfusion Clients Can Request GPUs	44
vSphere Bitfusion Client Cannot Connect to the vSphere Bitfusion Servers	45
vSphere Bitfusion Server Cannot Start	46

About *VMware vSphere Bitfusion User Guide*

1

The *VMware vSphere Bitfusion User Guide* provides information about using and configuring VMware vSphere[®] Bitfusion[®].

At VMware, we value inclusion. To foster this principle within our customer, partner, and internal community, we create content using inclusive language.

The *VMware vSphere Bitfusion User Guide* describes how to allocate, partition, and attach GPUs to workloads, and how to configure and monitor vSphere Bitfusion.

Intended Audience

This guide is intended for advanced users who are familiar with ESXi, vCenter Server, and command-line interface (CLI).

Understanding VMware vSphere Bitfusion

2

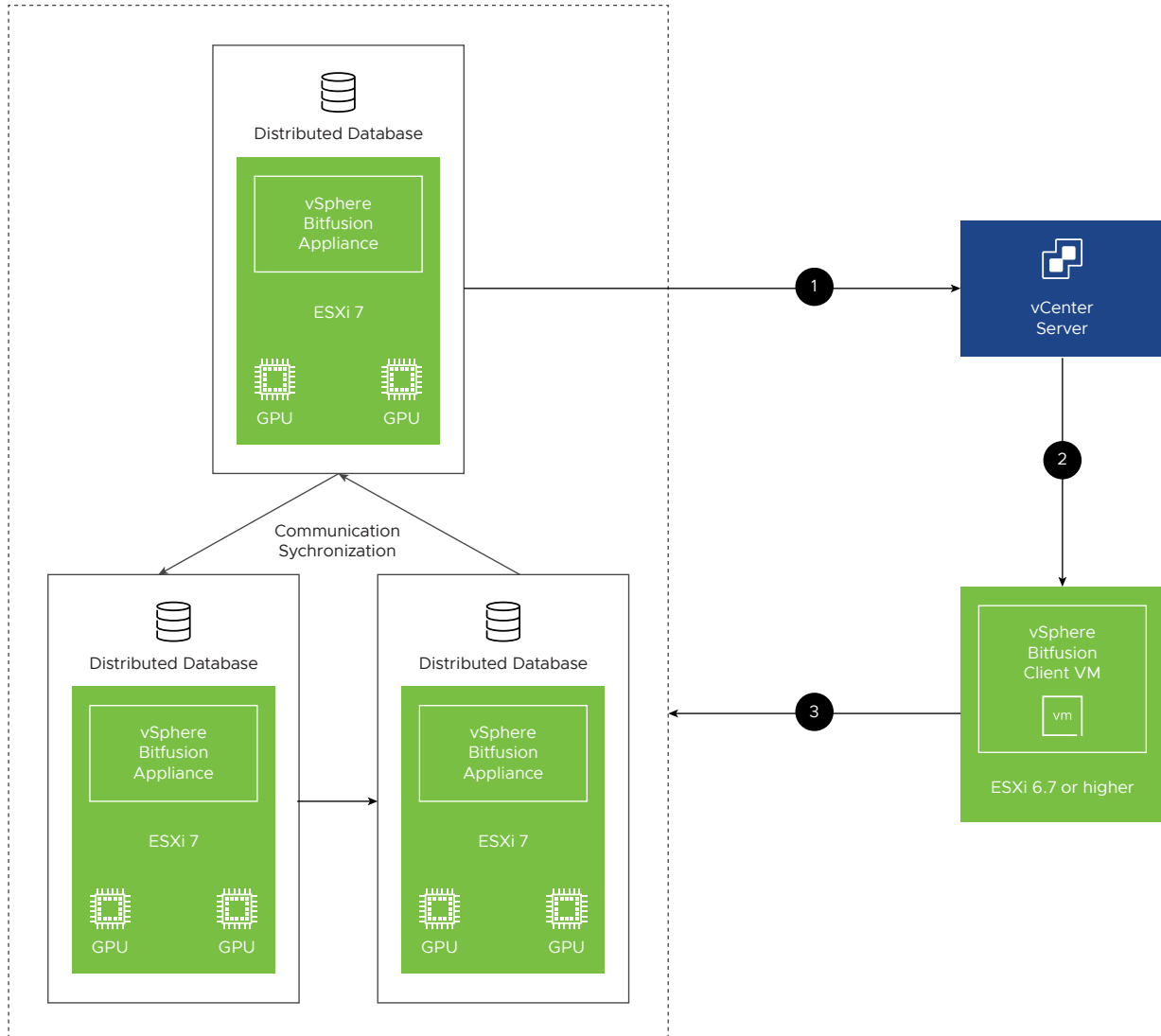
VMware vSphere Bitfusion virtualizes hardware accelerators such as graphical processing units (GPUs) to provide a pool of shared, network-accessible resources that support artificial intelligence (AI) and machine learning (ML) workloads.

vSphere Bitfusion Architecture

vSphere Bitfusion has a client-server architecture. The product allows multiple client virtual machines (VMs) running artificial intelligence (AI) and machine learning (ML) applications to share access to remote GPUs on virtual machines running vSphere Bitfusion server software. You run the applications on the vSphere Bitfusion client machines, while the GPUs that provide acceleration are installed on the vSphere Bitfusion server machines across a network. The applications can open files, allocate memory, and call CUDA as if operating on a machine with local GPUs.

The following figure is an example of a small vSphere Bitfusion cluster, such as a set of vSphere Bitfusion server-client machines and vCenter Server on a switched network. A minimal vSphere Bitfusion cluster configuration is one client, one server, and one vCenter Server. You can create large clusters with multiple clients and multiple servers.

Figure 2-1. Example of a small vSphere Bitfusion cluster



- 1 The primary vSphere Bitfusion server registers a vSphere Bitfusion Plug-in with vCenter Server.
- 2 The vSphere Bitfusion Plug-in enables a vSphere Bitfusion client VM.
- 3 The vSphere Bitfusion Client has authorized access to all vSphere Bitfusion servers in the vSphere Bitfusion cluster.

Note Before using VMware vSphere Bitfusion, you must deploy a vSphere Bitfusion server, and install and enable a vSphere Bitfusion client. For more information, see the *VMware vSphere Bitfusion Installation Guide*.

vSphere Bitfusion Functionality

When you start an AI or ML application on the vSphere Bitfusion client, vSphere Bitfusion intercepts the CUDA calls of the application, and sees the data and data pointers of the calls. The vSphere Bitfusion server does not require a connection to the data, but to the vSphere Bitfusion client only. The client transfers the data and the rest of the CUDA calls to the server. The vSphere Bitfusion server processes the calls and returns the results back to the client.

When you run AI and ML applications, vSphere Bitfusion can perform the following operations.

- Dynamically allocate and access GPU resources from vSphere Bitfusion servers.

Applications can share GPU resources that are not dedicated to individual machines and you can run each application on a configured machine, container, and environment. Applications consume GPU acceleration services from a pool of vSphere Bitfusion servers across a network, and consume the resources only for the length of time that an application or session runs. GPUs return to the pool when applications or sessions complete.

- Access partitions of GPU resources for concurrent sharing with other applications.

Another option to share GPUs is by partitioning the GPUs. The memory of a physical GPU can be divided into fractions of an arbitrary size and allocated to different applications at the same time. vSphere Bitfusion performs sharing with an interposition technology. vSphere Bitfusion intercepts API calls that normally address a local accelerator on a PCIe host bus and sends the API calls and related data across a network. vSphere Bitfusion provides sharing services for AI and ML applications, and supports the CUDA API to target NVIDIA GPUs.

vSphere Bitfusion Components

vSphere Bitfusion Server

vSphere Bitfusion server runs on an ESXi host with locally installed GPUs as a VMware appliance, which is a preconfigured virtual machine (VM) with prepackaged software and services. The server requires access to the local GPUs, usually through VMware vSphere® DirectPath I/O™.

vSphere Bitfusion Client

vSphere Bitfusion client runs on VMs which run the AI and ML applications.

vSphere Bitfusion Plug-In

The vSphere Bitfusion servers register a vSphere Bitfusion Plug-in with VMware vCenter Server. The plug-in provides monitoring and management of vSphere Bitfusion clients and servers.

vSphere Bitfusion Cluster

vSphere Bitfusion cluster is the set of all vSphere Bitfusion servers and clients in a vCenter Server instance.

vSphere Bitfusion Group

The vSphere Bitfusion client creates a vSphere Bitfusion group during the installation process. Only the members of the group can use vSphere Bitfusion. Certain configuration files are set up with appropriate permissions and the members of the group inherit appropriate limits to work effectively with vSphere Bitfusion.

vSphere Client

The vSphere Client lets you connect to vCenter Server instances by using a Web browser, so that you can manage your vSphere infrastructure. You access the vSphere Bitfusion Plug-in through the vSphere Client.

Command-Line Interface (CLI)

You can manage vSphere Bitfusion servers and clients by using command-line interface (CLI) commands.

vCenter Server

vCenter Server is the server management software that provides a centralized platform for controlling your vSphere environment.

Starting Applications in vSphere Bitfusion

3

You can run an application in the entire GPU memory or only in a dedicated partition of the memory. vSphere Bitfusion can allocate a GPU, run an application, and deallocate the GPU with a single CLI command or you can use individual commands to perform the same tasks.

This chapter includes the following topics:

- [Starting Applications with the Run Command](#)
- [Allocating GPUs with the RUN Command](#)
- [Partitioning GPU Memory](#)
- [GPU Partitioning Examples](#)
- [Starting Applications with Reserved GPUs](#)

Starting Applications with the Run Command

The vSphere Bitfusion client can run machine learning applications on remote shared GPUs. By using the `run` command, you can start a single application in vSphere Bitfusion.

The vSphere Bitfusion command to start an application is `run` with a mandatory argument for the number of the GPUs. To distinguish vSphere Bitfusion arguments from applications, you use a double-hyphen separator or place the application within quotes. You start an application in vSphere Bitfusion by replacing the placeholder values with actual values and running one of the following commands.

- `bitfusion run -n num_gpus other switches -- applications and arguments`
- `bitfusion run -n num_gpus other switches "applications and arguments"`

By running the `run` command, you can perform the following three tasks.

- 1 Allocate GPUs from the shared pool
- 2 Start an application in an environment that can access the GPUs when the application makes CUDA calls
- 3 Deallocate the GPUs when the application closes

The `run` command encapsulates the `request_gpus`, `client`, and `release_gpus` commands. You can use the individual commands to allocate GPUs and run multiple applications on the same GPUs. For more information, see [Starting Applications with Reserved GPUs](#).

Allocating GPUs with the RUN Command

You can run the `run` command to allocate GPUs for a single application. The application runs in the entire memory resource of the GPUs.

All GPUs that are requested by using the `run` command must be allocated from a single vSphere Bitfusion server, and the server must list the GPUs as separate devices with different PCIe addresses.

For example, the AI application, `asimov_i.py`, takes two arguments: the number of GPUs and a batch size.

- When the application expects 1 GPU, run `bitfusion run -n 1 -- python asimov_i.py --num_gpus=1 --batchsz=64`
- When the application expects 2 GPUs, run `bitfusion run -n 2 -- python asimov_i.py --num_gpus=2 --batchsz=64`

By default, vSphere Bitfusion waits for 30 minutes for enough GPUs to be available. To modify the default interval, use the `--timeout value`, `-t value` argument. Enter the timeout in seconds or time and unit, such as seconds (s), minutes (m), and hours (h).

For example, you can define the following values for the *value* argument.

<code>10</code>	10 seconds
<code>10s</code>	10 seconds
<code>10m</code>	10 minutes
<code>10h</code>	10 hours

Partitioning GPU Memory

You can run your application in a dedicated partition of a GPU's memory, and other applications can use the remaining GPU's memory.

The GPU partitioning arguments are optional `run` command arguments. You use the arguments to run your application in a partition of a GPU memory.

- The GPU partitioning process is dynamic. When you start a `run` command with an argument, vSphere Bitfusion allocates a partition before the application runs and deallocates the partition afterwards.
- The applications that are sharing GPUs concurrently are isolated from each other by using separate client processes, network streams, server processes, and memory partitions.

- vSphere Bitfusion partitions only the memory of the GPU and not the compute resource. An application is strictly contained to the assigned memory partition, but it can access the complete compute resource, if needed. When the same compute cells are required, the applications compete for compute resources, otherwise the applications run concurrently.

You can specify the partition size in MB or as a fraction of the total GPU memory.

Partitioning GPU memory size by fraction (number > 0.0 and <= 1.0, for example, 0.37)

```
bitfusion run -n num_gpus -p gpu_fraction -- applications and arguments
```

Partitioning GPU's memory size by MB

```
bitfusion run -n num_gpus -p MBs_per_gpu -- applications and arguments
```

For more information, see [GPU Partitioning Examples](#).

GPU Partitioning Examples

Multiple concurrent applications might use a GPU's computational capacity more efficiently than a single application. There are several ways you can partition the memory of your GPUs.

If you are using inference applications with smaller model sizes or small batches of work, such as number of images, you can run the applications concurrently on partitioned GPUs.

You can perform empirical testing to understand the memory size an application requires. Some applications expand to use all available memory, but they might not achieve better performance beyond a certain threshold.

The following examples presume knowledge of acceptable memory requirements with different batch sizes.

- When you expect that an application with a batch size of 64 requires 66% of GPU memory, run `bitfusion run -n 1 -p 0.66 -- python asimov_i.py --num_gpus=1 --batchsz=64`
- When you expect that an application with a batch size of 32 requires 5461 MB of GPU memory, run `bitfusion run -n 1 -m 5461 -- python asimov_i.py --num_gpus=1 --batchsz=32`

When you request multiple GPUs, all GPUs allocate the same amount of memory. The fraction size specification must be based on the GPU with the smallest amount of memory.

In the following example, the `-p` argument requests 33% of the memory of each of the two requested GPUs. The GPUs must physically reside on the same server. If the GPUs are 16 GB devices or if the smallest GPU is a 16 GB device, then approximately 5461 MB is allocated on each GPU. While no other applications are running, `asimov_i.py` can access the full compute power of the two GPUs.

```
Run bitfusion run -n 2 -p 0.33 -- python asimov_i.py --num_gpus=1 --batchsz=64
```

You can run multiple applications from a single client on the same GPU concurrently.

For example, to start two concurrent application instances in the background, run both these commands.

```
1 bitfusion run -n 1 -p 0.66 -- python asimov_i.py --num_gpus=1 --batchsz=64
  &
2 bitfusion run -n 1 -p 0.33 -- python asimov_i.py --num_gpus=1 --batchsz=32
  &
```

NVIDIA System Management Interface (nvidia-smi)

You can run the NVIDIA System Management Interface `nvidia-smi` monitoring application, for example, to check your GPU partition size or verify the resources available on a vSphere Bitfusion server. Typically, the application is provided on the server when you install the NVIDIA driver.

Applications that run on the vSphere Bitfusion clients do not require the NVIDIA driver, but might require the `nvidia-smi` application, for example, to understand the capabilities of the GPU or to determine the GPU memory sizing. To support such operations, since vSphere Bitfusion 3.0, the `nvidia-smi` application is provided on all vSphere Bitfusion clients. vSphere Bitfusion copies the application from the server to the client.

For example, to request a 1024 MB partition on a GPU, run `bitfusion run -n 1 -m 1024 -- nvidia-smi`.

The output of the `nvidia-smi` application displays the requested partition value of 1024MiB.

```
Requested resources:
Server List: 172.16.31.241:56001
Client idle timeout: 0 min
Wed Sep 23 15:21:17 2020

+-----+
| NVIDIA-SMI 440.100      Driver Version: 440.64.00    CUDA Version: 10.2     |
+-----+-----+-----+
| GPU   Name               Persistence-M| Bus-Id        Disp.A | Volatile Uncorr. ECC |
| Fan   Temp   Perf    Pwr:Usage/Cap|      Memory-Usage | GPU-Util  Compute M. |
|=====+=====+=====+
|    0   Tesla T4              Off      | 00000000:13:00.0 Off  |             0        |
| N/A   36C    P8      9W / 70W | 0MiB / 1024MiB |      0%      Default  |
+-----+-----+-----+

+-----+
| Processes:                                                       GPU Memory |
|  GPU       PID    Type    Process name                     Usage      |
|=====+=====+
|  No running processes found                                     |
+-----+
```

Starting Applications with Reserved GPUs

You can allocate a number of GPUs and run multiple applications on the same GPUs.

While the `run` command allocates GPU, runs applications, and deallocates GPU collectively, vSphere Bitfusion has three individual commands to perform the same tasks. By using the individual commands, you can use the same GPU for multiple applications and have greater control when you are integrating vSphere Bitfusion into other tools and workflows, such as the scheduling software, SLURM.

- To allocate GPUs, run `request_gpus`.
- To start applications in an environment that can access the GPUs when the application makes CUDA calls, run `client`.
- To deallocate the GPUs, run `release_gpus`.

Note The `request_gpus` command creates a file and environment variables that can be forwarded to other tools. The tools can run the `client` command with the same allocation configuration.

The arguments of the `run` command are split between the `request_gpus` and `client` commands.

To understand the use of the individual commands, see the following example workflow that is using the AI application `asimov_i.py`.

- 1 To allocate GPUs to start multiple and sequential applications, run `bitfusion request_gpus -n 1 -m 5461`.

```
Requested resources:
Server List: 172.16.31.241:56001
Client idle timeout: 0 min
```

- 2 To start an application by running the `client` command, run `bitfusion client nvidia-smi`.

```
Wed Sep 23 15:26:02 2020
+-----+
| NVIDIA-SMI 440.100      Driver Version: 440.64.00      CUDA Version: 10.2      |
+-----+-----+-----+
| GPU   Name           Persistence-M| Bus-Id        Disp.A | Volatile Uncorr. ECC |
| Fan   Temp   Perf    Pwr:Usage/Cap|      Memory-Usage | GPU-Util  Compute M. |
|=====+=====+=====+
|    0   Tesla T4            Off   | 00000000:13:00:0 Off  |            0         |
| N/A   36C    P8      10W / 70W   |      0MiB /  5461MiB |           0%      Default |
+-----+-----+-----+

+-----+
| Processes:                                     GPU Memory |
|  GPU       PID    Type    Process name                     Usage      |
|=====+=====+
| No running processes found                                     |
+-----+
+-----+
|
```

- 3 To start another application by running the `client` command, run `bitfusion client --python asimov_i.py --num_gpus=1 --batchsz=64`.
- 4 To deallocate the GPUs , run `bitfusion release_gpus`.

Managing vSphere Bitfusion Servers

4

By using the vSphere Bitfusion plug-in and CLI commands, you can add, remove, configure, and monitor the vSphere Bitfusion servers in your cluster.

This chapter includes the following topics:

- [Add Subsequent vSphere Bitfusion Servers](#)
- [Remove a vSphere Bitfusion Server](#)
- [Configuring the Network Settings of a vSphere Bitfusion Server](#)
- [Change the Settings of a vSphere Bitfusion Server](#)
- [Perform a Health Check of a vSphere Bitfusion Server](#)
- [vSphere Bitfusion Health Checks List](#)
- [Create vSphere Bitfusion Server Logs](#)
- [View vSphere Bitfusion Server Logs](#)
- [View GPU Information for a vSphere Bitfusion Server](#)

Add Subsequent vSphere Bitfusion Servers

You can add more servers to your vSphere Bitfusion cluster when you require more GPU resources.

After the primary vSphere Bitfusion server starts, vSphere Bitfusion registers a vSphere Bitfusion plug-in in the vCenter Server, resulting in a single vSphere Bitfusion cluster containing one vSphere Bitfusion server. After the vSphere Bitfusion plug-in is registered, you can add subsequent servers by following the steps in this procedure. The vSphere Bitfusion plug-in uses the primary server's configuration data, which allows faster deployment of the subsequent servers.

Alternatively, you can add a new server in your vSphere Bitfusion cluster by following the deploy procedure for the primary server. You deploy the vSphere Bitfusion appliance on a virtual machine (VM), customize the vSphere Bitfusion OVF template, pass through the GPUs to the vSphere Bitfusion server VM, and enable the VM as a vSphere Bitfusion server.

Additional vSphere Bitfusion servers must be part of the same vCenter Server instance as the first vSphere Bitfusion server.

Prerequisites

- Verify you have installed a primary vSphere Bitfusion server.
- Verify that the vSphere Bitfusion is registered with vCenter Server server.

Procedure

- 1 From the **Hosts and Clusters** view in vCenter Server, right-click an ESXi host, and select **Bitfusion > Install Bitfusion server**.

The **Install Bitfusion server** dialog box appears.

- 2 On the **Select an OVA image** page, enter the URL of the vSphere Bitfusion OVA file or browse to the file, and click **Next**.
- 3 On the **Verify template details** page, review the OVA template details and click **Next**.
- 4 On the **Select a name and hostname** page, enter a name for the virtual machine and a hostname for the vSphere Bitfusion server, and click **Next**.

Optionally, you can specify a host ID for the vSphere Bitfusion server, for example, when you upgrade your vSphere Bitfusion server. If you skip this step, a host ID is generated and assigned automatically.

- 5 On the **Select storage** page, define where and how to store the files of the deployed VM, and click **Next**.
- 6 On the **Select networks** page, specify the networking configuration for Network Adapter 1 and click **Next**.

You must specify the configuration for Network Adapter 1 that is used for management and data traffic. Network Adapter 1 must be connected to a network that communicates with the vCenter Server instance.

If your vSphere Bitfusion server requires additional network adapters for data traffic, you can click **Add Network Adapter** and specify the network configuration for the additional adapter.

Option	Description
Network Adapter	Select a network from the drop-down menu.
Adapter Type	Select a network adapter to assign to the virtual machine. Note vSphere Bitfusion supports VMXNET3 and PVRDMA adapters.
DHCP/Fixed IP	Specify whether a DHCP server assigns the address of the network adapter or you use a fixed IPv4 address.
IPv4 Address	Enter the IPv4 address of the network adapter. If you are using DHCP, leave this text box blank. Note IPv6 is not supported.
Netmask	Select a netmask from the drop-down menu. For example, if your network uses a /24 netmask, select 24 (255.255.255.0).

Option	Description
Gateway	Enter the network gateway address to use with the appliance. If you are using DHCP, leave this text box blank.
MTU	Enter an MTU size. The default value is 1500. For optimal performance, specify an MTU size that is equal to the maximum MTU size supported by your network hardware. Note If you set an MTU size greater than 1500, verify that the network switches in your data center are enabled for jumbo frames.
DNS Servers	Enter the DNS server address to use with the appliance. If you are using DHCP, leave this text box blank.
DNS Search Domains	Enter the DNS search domain address to use with the appliance. If you are using DHCP, leave this text box blank.
NTP	Enter the NTP server address to use with the appliance. If you are using DHCP and the DHCP server supports sending NTP server information, leave this text box blank.

7 On the **Select GPUs** page, add GPUs to the subsequent server and click **Next**.

- a Click **Add GPU**.
- b Select a GPU from the **GPU Device** drop-down menu.
- c (Optional) Specify the total memory of the GPU.

The vSphere Bitfusion plug-in uses the aggregated GPU memory of all GPUs you add on the **Select GPUs** page to calculate the values for the minimum memory and the recommended memory mapped I/O size of the virtual machine of your vSphere Bitfusion server.

- d (Optional) To accept the NVIDIA license, select the **Download and Install NVIDIA Driver** check box.

By accepting the NVIDIA license, vSphere Bitfusion downloads and installs the NVIDIA driver, CUDA libraries, and NVIDIA Fabric Manager during the first boot of the virtual machine.

Note If you are operating vSphere Bitfusion in an environment without access to the Internet, for example, by using an air-gapped network, do not select the check box. You must manually download and install the NVIDIA software after deploying the vSphere Bitfusion appliance.

If your vSphere Bitfusion server requires additional GPUs, you can click **Add GPU Device** again and specify the settings for the GPU.

8 On the **Customize server** page, specify the vSphere Bitfusion server details and click **Next**.

- a Specify the number of CPUs for the virtual machine.
- b Specify the memory mapped I/O (MMIO) size of the virtual machine in GB.

- c (Optional) Enter a password for the customer account.

After the deployment is complete, you use the customer user account to log into the vSphere Bitfusion server by using the console shell or SSH. If you skip this step, you cannot log into the subsequent server.

- d (Optional) Select the **Power On VM After Create** check box.

You can deselect the check box, if you make changes to the virtual machine before powering it on.

- 9 On the **Summary** page, review the deployment details and click **Finish**.

Results

A new task for installing the vSphere Bitfusion server appears in the Recent Tasks pane. After the task finishes, the new appliance is created on the selected resources.

When a new vSphere Bitfusion server joins the cluster, vCenter Server supplies a token, a certificate, and a configuration to access the vSphere Bitfusion cluster.

Remove a vSphere Bitfusion Server

To perform troubleshooting or maintenance on a vSphere Bitfusion server, you must remove the server from the vSphere Bitfusion cluster.

When powering off a vSphere Bitfusion server for maintenance or to perform troubleshooting, the health status of the vSphere Bitfusion cluster changes. When the cluster is not in a healthy state, you cannot add vSphere Bitfusion servers or perform a cluster backup operation. If half of the servers are powered off, the cluster is inoperable. When powering off a server for a longer period of time, you can prevent any potential risk by removing the server from the cluster.

Performing the following procedure immediately removes the server from the vSphere Bitfusion cluster. Any running applications that are using the GPUs receive an immediate GPU failure and usually return an error condition.

Prerequisites

- Prevent new client connections to the specific server in the server settings.
- Verify that there are no running applications on the server.

Procedure

- 1 In the vSphere Client, select **Menu > Bitfusion**.
- 2 On the **Servers** tab, select a server from the list.
- 3 From the **Actions** drop-down menu, select **Delete**.
- 4 In the confirmation dialog box, click **Delete**.

- 5 Wait until the server is no longer listed on the **Servers** tab.

The delete operation can take up to 10 minutes and longer. During this time, the backing storage rebalances. Alternatively, you can verify that the delete operation is finished by running the `nodetool status` command in the terminal of a running server.

- 6 (Optional) Delete the server virtual machine (VM).

Accidentally powering on the removed VM may result in the vSphere Bitfusion plug-in and cluster information being overwritten.

Results

You have deleted the selected server from the vSphere Bitfusion cluster.

What to do next

To reuse the VM or the underlying hardware, you can perform one of the following tasks.

- If you deleted the server from the cluster without deleting the VM, delete the `/etc/bitfusion/bitfusion-manager.yaml` configuration file on the VM, reenabling the VM as a vSphere Bitfusion server, restart the vSphere Bitfusion service, and power on the VM. For more information, see *Enabling the vSphere Bitfusion Client* in the *VMware vSphere Bitfusion Installation Guide* and [Start and Stop the vSphere Bitfusion Service](#).
- If you deleted the server VM, you can reuse the underlying hardware as a vSphere Bitfusion server by creating a VM and deploying the vSphere Bitfusion server appliance. See [Add Subsequent vSphere Bitfusion Servers](#).

Configuring the Network Settings of a vSphere Bitfusion Server

After you deploy a vSphere Bitfusion server, you can connect your vSphere Bitfusion server to multiple networks by adding, removing, and modifying network interfaces.

You can connect the virtual machine of a vSphere Bitfusion server to up to 4 networks and if you do not use a DHCP server, you can set a network's IPv4 address, CIDR Prefix, and MTU size. You can also specify a network gateway address, DNS server address, DNS search domain address, and NTP server address for the management network.

Adapter	Description
Network Adapter 1	This network is used for management and data traffic.
Network Adapter 2	This network is used for data traffic only.
Network Adapter 3	This network is used for data traffic only.
Network Adapter 4	This network is used for data traffic only.

Add a Network Interface

You can connect the virtual machine of your vSphere Bitfusion server to up to four networks.

During the deployment process of a vSphere Bitfusion server, you must configure at least Network Adapter 1, which is used for management and data traffic. Network adapters 2, 3, and 4 are optional and are used for data traffic only. To add network interfaces for data traffic after the deployment process of your server is finished, follow this procedure.

Note Each network adapter must be connected to a separate network. vSphere Bitfusion chooses the network that is most efficient for data transfers to the vSphere Bitfusion server.

Prerequisites

- Verify that you have the **Virtual machine.Configuration.Add or remove device** privilege.
- Verify that the virtual machine of the vSphere Bitfusion server is powered off.

Procedure

- 1 In the vSphere Client, right-click the virtual machine of a vSphere Bitfusion server and select **Edit Settings**.
- 2 On the **Virtual Hardware** tab, click the **Add New Device** button.
- 3 Under **Network**, select **Network Adapter**.
- 4 From the **New Network** drop-down menu, select a network to connect the virtual machine to.
- 5 Expand the **New Network** section and from the **Adapter Type** drop-down menu, select the network adapter to assign to the virtual machine.

vSphere Bitfusion supports VMXNET3 and PVRDMA adapters.
- 6 Click **OK**.

Results

You have added a new network adapter to the virtual machine of your vSphere Bitfusion server.

What to do next

- You can add up to four network adapters.
- Enable the adapter on the vSphere Bitfusion server and specify additional settings, if you are not using DHCP. See [Configure a Network Interface](#).

Configure a Network Interface

To configure a network adapter and specify the network's IPv4 address, CIDR Prefix, and MTU size, you must configure vApp properties. vSphere Bitfusion uses the values of these properties and configures the networking during the boot of the virtual machine.

The following procedure provides information on how to enable and set the network configuration for **Network Adapter 2** by configuring vApp properties. You can modify the configuration of the other network adapters by replacing the properties that is used in this procedure. For a list of all vApp properties that you can modify, see [vSphere Bitfusion vApp Properties](#).

Prerequisites

- Verify that you have the **vApp.vApp application configuration** privilege.
- Verify that the virtual machine of the vSphere Bitfusion server is powered off.

Procedure

- 1 From the **Hosts and Clusters** view in vCenter Server, select the virtual machine of a vSphere Bitfusion.
- 2 On the **Configure** tab, select **Settings > vApp Options**.
- 3 In the **Properties** pane, select the `guestinfo.bitfusion.host.net2.configure` property and click **Set Value**.
- 4 In the **Set value** dialog box, enable the toggle switch and click **OK**.
- 5 If you do not use DHCP, select a property and specify the value for **Network Adapter 2**.

Property	Value
<code>guestinfo.bitfusion.host.net2.ipv4address</code>	Enter an IPv4 address. For example, 192.168.200.111.
<code>guestinfo.bitfusion.host.net2.netmask</code>	Select a netmask value from the drop-down menu.
<code>guestinfo.bitfusion.host.net2.mtu</code>	Enter a valid MTU size. For example, 9000.

Results

You have configured **Network Adapter 2**.

What to do next

You can configure the other network adapters by replacing the correspondent properties and following the same procedure. See [vSphere Bitfusion vApp Properties](#).

Remove a Network Interface

You can remove a network adapter, if for example, the virtual machine of a vSphere Bitfusion server is no longer using a network.

Prerequisites

- Verify that you have the **Virtual machine.Configuration.Add or remove device** privilege.
- Verify that you have the **vApp.vApp application configuration** privilege.
- Verify that the virtual machine of the vSphere Bitfusion server is powered off.

Procedure

- 1 In the vSphere Client, select the virtual machine of a vSphere Bitfusion server and select **Edit Settings**.
- 2 On the **Virtual Hardware** tab, to delete a network interface, click the remove icon (⊗) next to the network adapter.

- 3 Click **OK**.
- 4 From the **Hosts and Clusters** view in vCenter Server, select the virtual machine of the vSphere Bitfusion.
- 5 On the **Configure** tab, select **Settings > vApp Options**.
- 6 In the **Properties** pane, select a property and click **Set Value**.
 - If you deleted **Network Adapter 2**, select `guestinfo.bitfusion.host.net2.configure`.
 - If you deleted **Network Adapter 3**, select `guestinfo.bitfusion.host.net3.configure`.
 - If you deleted **Network Adapter 4**, select `guestinfo.bitfusion.host.net4.configure`.
- 7 In the **Set value** dialog box, disable the toggle switch and click **OK**.

Results

You removed the network adapter and the virtual machine of your vSphere Bitfusion server is not connected to this network.

vSphere Bitfusion vApp Properties

A list of all vApp properties that you can modify by changing their values.

Bitfusion Server Setup

Property	Value
<code>guestinfo.bitfusion.host.hostname</code>	The hostname for the server. Valid characters for hostnames are the ASCII characters A through Z (both upper- and lower-case), the digits 0 to 9, and the hyphen (-). A hostname cannot start with a hyphen.
<code>guestinfo.bitfusion.server.vcenter-guid</code>	The vCenter Server GUID.
<code>guestinfo.bitfusion.server.vcenter-url</code>	The vCenter Server URL.
<code>guestinfo.bitfusion.server.vcenter-username</code>	The username for the vCenter Server instance.
<code>guestinfo.bitfusion.server.vcenter-password</code>	The password for the vCenter Server instance.
<code>guestinfo.bitfusion.host.install_nvidia_packages</code>	Slide the toggle button to ON position to download and install the NVIDIA software, or slide the toggle button to OFF position to skip this process.

Network Adapter 1 (Management and Data)

Property	Value
<code>guestinfo.bitfusion.host.net1.ipv4address</code>	The IPv4 address of the network adapter. If you are using DHCP, leave this value blank. Note IPv6 is not supported.
<code>guestinfo.bitfusion.host.net1.netmask</code>	The network Classless Inter-Domain Routing (CIDR) settings.

Property	Value
<code>guestinfo.bitfusion.host.net1.mtu</code>	<p>The MTU size. The default value is 1500. For optimal performance, specify an MTU size of 4000 or greater. You can leave this text box blank for default value.</p> <p>Note If you set an MTU size greater than 1500, verify that the network switches in your data center are enabled for jumbo frames.</p>
<code>guestinfo.bitfusion.host.net1.gateway</code>	The network gateway address to use with the appliance. If you are using DHCP, leave this text box blank.
<code>guestinfo.bitfusion.host.net1.dns</code>	The DNS server address to use with the appliance. If you are using DHCP, leave this text box blank.
<code>guestinfo.bitfusion.host.net1.domain</code>	The DNS search domain address to use with the appliance. If you are using DHCP, leave this text box blank.
<code>guestinfo.bitfusion.host.net1.ntp</code>	The NTP server address to use with the appliance. If you are using DHCP and the DHCP server supports sending NTP server information, leave this text box blank.

Network Adapter 2 (Data)

Property	Value
<code>guestinfo.bitfusion.host.net2.configure</code>	Enable or disable the toggle switch to configure or not to configure this interface.
<code>guestinfo.bitfusion.host.net2.ipv4address</code>	<p>The IPv4 address of the network adapter. If you are using DHCP, leave this value blank.</p> <p>Note IPv6 is not supported.</p>
<code>guestinfo.bitfusion.host.net2.netmask</code>	The network Classless Inter-Domain Routing (CIDR) settings.
<code>guestinfo.bitfusion.host.net2.mtu</code>	<p>The MTU size. The default value is 1500. For optimal performance, specify an MTU size of 4000 or greater. You can leave this text box blank for default value.</p> <p>Note If you set an MTU size greater than 1500, verify that the network switches in your data center are enabled for jumbo frames.</p>

Network Adapter 3 (Data)

Property	Value
<code>guestinfo.bitfusion.host.net3.configure</code>	Enable or disable the toggle switch to configure or not to configure this interface.
<code>guestinfo.bitfusion.host.net3.ipv4address</code>	<p>The IPv4 address of the network adapter. If you are using DHCP, leave this value blank.</p> <p>Note IPv6 is not supported.</p>

Property	Value
guestinfo.bitfusion.host.net3.netmask	The network Classless Inter-Domain Routing (CIDR) settings.
guestinfo.bitfusion.host.net3.mtu	The MTU size. The default value is 1500. For optimal performance, specify an MTU size of 4000 or greater. You can leave this text box blank for default value. Note If you set an MTU size greater than 1500, verify that the network switches in your data center are enabled for jumbo frames.

Network Adapter 4 (Data)

Property	Value
guestinfo.bitfusion.host.net4.configure	Enable or disable the toggle switch to configure or not to configure this interface.
guestinfo.bitfusion.host.net4.ipv4address	The IPv4 address of the network adapter. If you are using DHCP, leave this value blank. Note IPv6 is not supported.
guestinfo.bitfusion.host.net4.netmask	The network Classless Inter-Domain Routing (CIDR) settings.
guestinfo.bitfusion.host.net4.mtu	The MTU size. The default value is 1500. For optimal performance, specify an MTU size of 4000 or greater. You can leave this text box blank for default value. Note If you set an MTU size greater than 1500, verify that the network switches in your data center are enabled for jumbo frames.

Change the Settings of a vSphere Bitfusion Server

You can change server-specific settings from the vSphere Bitfusion Plug-in, such as allowing new client connections and entering a metrics interval.

The following procedure changes the settings for a specific vSphere Bitfusion server only. You can change the global settings for all vSphere Bitfusion servers in the **Settings > Global Server Defaults** tab.

Procedure

- 1 In the vSphere Client, select **Menu > Bitfusion**.
- 2 On the **Servers** tab, select a server from the list.
- 3 From the **Actions** drop-down menu, select **Settings**.
- 4 Change one or more server settings as required.
 - Select or deselect the **Allow new client connections** check box.

To shut down a vSphere Bitfusion server gracefully and perform maintenance or troubleshooting, you can deselect the **Allow new client connections** check box. This action prevents vSphere Bitfusion clients from running new applications on the GPUs of the vSphere Bitfusion servers while you wait for all running applications to finish.

- To set a frequency to collect server statistics, enter a **Metrics interval** value in seconds.
- To use the global server settings for this vSphere Bitfusion server, click **Match Defaults**.

5 Click **Save**.

Perform a Health Check of a vSphere Bitfusion Server

You can check the performance, stability, system resources, and software versions of a vSphere Bitfusion server by performing a health check.

You can check the health status of a selected vSphere Bitfusion server and if needed, perform troubleshooting. The health check examines the performance, stability, system resources, and software versions of a selected vSphere Bitfusion server and the server's surrounding vCenter Server environment. Each health check can return a pass, marginal, or fatal status.

For example, the health check verifies that all nodes are running, that there is enough free space, and that the connection to vCenter Server is working. To view the list of all available health checks, see [vSphere Bitfusion Health Checks List](#).

By disabling a health check in the following procedure, you change the health check settings for the specific vSphere Bitfusion server only. A disabled health check is still performed in the background, but the status of the check is not changing the overall health status of the server that is displayed on the **Servers** tab. You can change the global health check settings for all vSphere Bitfusion servers on the **Settings > Global Health Check Defaults** tab.

Procedure

- 1 In the vSphere Client, select **Menu > Bitfusion**.
- 2 On the **Servers** tab, select a server from the list.
- 3 From the **Actions** drop-down menu, select **Health**.

The **Health logs** dialog box appears and the results of the health checks are displayed. You see the status, type, name, and details of the check.

- 4 (Optional) To disable a specific health check, click the toggle button.
- 5 Click **Save and Exit**.

What to do next

- [View vSphere Bitfusion Server Logs](#)
- [Back Up a vSphere Bitfusion Cluster](#)

vSphere Bitfusion Health Checks List

vSphere Bitfusion performs the following checks when a health check of a server is initiated from the vSphere Bitfusion Plug-in.

Health Checks List

Name	Type	Description
cass_buckets	Stability	Validates the bucketing used by Cassandra to store data for utilization and other items.
cass_node_num	Stability	Confirms that Cassandra and Bitfusion see the same number of servers in the cluster.
cass_nodetool	Stability	Confirms that Cassandra sees that the cluster is in a healthy state.
cass_replication	Stability	Confirms the replication factor.
compute_mode	Stability	Confirms that the GPUs have compute mode set appropriately.
network	Stability	Verifies if there are dropped packets on the network.
ecc	Stability	Verifies if there are any ECC errors on the GPUs.
gpu_api	Stability	Confirms that the GPU APIs are matching.
pci_nvml	Stability	Confirms that all GPUs can be enumerated.
pci_p2p	Stability	Verifies that PCIe P2P is supported.
temperature	Stability	Verifies that the GPUs temperature is below 100 degrees celsiuses.
vcenter_check	Stability	Validates that the server can connect to vCenter Server.
xid	Stability	Verifies if there are any GPU Xid failures.
bogomips	Performance	Validates performance. The metric is used by the Linux kernel.
hostmem	Performance	Validates that there is enough host memory on the system.
iface_compat	Performance	Validates that the network configuration is valid.
memops	Performance	Verifies that <code>memops</code> is enabled for the GPUs.
mtu	Performance	Verifies that jumbo frames are enabled for the network.
nvidia_stats	Performance	Validates the statistics for the GPUs.
nvidia_topo	Performance	Validates the host topology.
pci_width	Performance	Validates that the GPUs are using the maximum PCIe lane capacity.
ulimit_n	Performance	Verifies that the maximum file descriptors limit is appropriate.
diskspace	System Resource	Confirms the free space on the server.
install	System Resource	Validates the Bitfusion installation.

Name	Type	Description
pciinfo	System Resource	Validate the PCI configuration.
shadow_mem	System Resource	Verifies that there is at least the same amount of system memory as there is frame buffer memory on the GPUs.
cuda_version	Software Version	Verifies the CUDA version.
libdep	Software Version	Verifies that the software dependencies for Bitfusion are installed.
driver_version	Software Version	Verifies the NVIDIA driver version.

Create vSphere Bitfusion Server Logs

Since vSphere Bitfusion 2.5, you can run a support script that gathers essential server information and creates a bundle of the server logs. These logs provide important information when troubleshooting a server.

The support script gathers information about the setup and configuration of vSphere Bitfusion, driver versions, health check output, status of all servers and the server's hardware, Cassandra configuration and status, and others. Typically, you need the support bundle when working with VMware support.

By completing the following procedure, you create a support bundle by using the vSphere Bitfusion plug-in. Alternatively, you can create a server logs by using the command-line interface (CLI). You can log into the terminal of a running vSphere Bitfusion server and run the `sudo bitfusion-supportbundle.sh` command. The support bundle is created in `/tmp/bitfusion-supportbundle.tar.gz`

Procedure

- 1 In the vSphere Client, select **Menu > Bitfusion**.
- 2 On the **About** tab, under **Support**, click **Generate Support Bundle**.
- 3 Select a location to save the `.tar` file on your local machine.

View vSphere Bitfusion Server Logs

Server logs can provide useful insights when troubleshooting a vSphere Bitfusion server.

To investigate any possible issues with vSphere Bitfusion, you can view the activity log of a specific vSphere Bitfusion server. For example, you can check the logs for thumbprint problems or vCenter Server GUID problems, that have occurred during the vSphere Bitfusion Plug-in registration process.

Procedure

- 1 In the vSphere Client, select **Menu > Bitfusion**.

- 2 On the **Servers** tab, select a server from the list.
- 3 From the **Actions** drop-down menu, select **Logs**.

View GPU Information for a vSphere Bitfusion Server

You can view GPU-related information, such as driver version, partition size, and available resources for your vSphere Bitfusion servers.

The information displayed is similar to the output of the `nvidia-smi` application. For example, you can view the GPU temperature, fan speed, the currently running processes, and the resources available on a vSphere Bitfusion server.

If you want to view the allocated and partial GPUs for a specific vSphere Bitfusion client, see [View GPU Information for a vSphere Bitfusion Client](#).

Procedure

- 1 In the vSphere Client, select **Menu > Bitfusion**.
- 2 On the **Servers** tab, select a server from the list.
- 3 From the **Actions** drop-down menu, select **GPUs**.

Managing vSphere Bitfusion Clients

5

By using the vSphere Bitfusion plug-in, you can remove, configure, and monitor the vSphere Bitfusion clients in your cluster.

This chapter includes the following topics:

- [Disable or Delete a vSphere Bitfusion Client](#)
- [Change the Settings of a vSphere Bitfusion Client](#)
- [View GPU Information for a vSphere Bitfusion Client](#)

Disable or Delete a vSphere Bitfusion Client

You can stop a client from starting new application jobs or immediately prevent the client from accessing all vSphere Bitfusion servers.

Procedure

- 1 In the vSphere Client, select **Menu > Bitfusion**.
- 2 On the **Clients** tab, select a client from the list.
- 3 Disable or delete a vSphere Bitfusion client.
 - a From the **Actions** drop-down menu, select **Disable**
 - b In the confirmation dialog box, click **Disable**.

This option prevents the client from starting new applications and allows running applications to finish. After the client is disabled, you can still view the historical client data and re-enable the client later.

- a From the **Actions** drop-down menu, select **Delete**.
- b In the confirmation dialog box, click **Delete**.

This option immediately stops the client from accessing all vSphere Bitfusion servers. After the client is deleted, you can view only the historical client data in the vSphere Bitfusion server's database.

Change the Settings of a vSphere Bitfusion Client

You can change client-specific settings from the vSphere Bitfusion Plug-in, such as current GPU quota, auto disconnect, and auto-shutdown idle interval.

The following procedure changes the settings for a specific vSphere Bitfusion client only. You can change the global settings for all vSphere Bitfusion clients in the **Settings > Global Client Defaults** tab.

Procedure

- 1 In the vSphere Client, select **Menu > Bitfusion**.
- 2 On the **Clients** tab, select a client from the list.
- 3 From the **Actions** drop-down menu, select **Settings**.
- 4 Change one or more client settings as required.
 - Enter a **Current GPU Quota**.

The quota is the maximum number of GPUs that a vSphere Bitfusion client can allocate for all client applications. You can use non-integer values. For example, a quota of 3.5 allows a client to run simultaneously one application on two GPUs and a second application on 3 half-sized GPUs. The default value of 64 is enough to approximate an unlimited quota.
 - Select the **Auto Disconnect Idle GPUs** check box and enter an idle interval in minutes.

This option allows vSphere Bitfusion to deallocate client GPUs and return the GPUs to the pool if the auto-shutdown idle interval is reached.
 - To use the global client settings for this vSphere Bitfusion client, click **Match Defaults**.
- 5 Click **Save**.

View GPU Information for a vSphere Bitfusion Client

You can view the number of GPUs, that are allocated fully and partially for a specific vSphere Bitfusion client. Also, the GPU model and allocated memory are displayed.

To view GPU information related to a specific server, see [View GPU Information for a vSphere Bitfusion Server](#).

Procedure

- 1 In the vSphere Client, select **Menu > Bitfusion**.
- 2 On the **Clients** tab, select a client from the list.
- 3 From the **Actions** drop-down menu, select **GPUs**.

Managing vSphere Bitfusion

6

You can manage vSphere Bitfusion by using the vSphere Bitfusion plug-in, CLI commands, and the vSphere Client. For example, you can back up, restore, configure, and monitor vSphere Bitfusion.

Back up and Restore a vSphere Bitfusion Cluster

You can back up and restore your vSphere Bitfusion database. The distributed database includes the configuration, connectivity, health state, and history data of your vSphere Bitfusion cluster. The backup operation saves a snapshot of the database information and if your cluster fails, the restore operation recovers the cluster to a previously healthy state.

Managing vSphere Bitfusion with the vSphere Bitfusion Plug-In

After a vSphere Bitfusion server starts for the first time, the server registers a plug-in with vCenter Server. The vSphere Bitfusion plug-in provides a graphical user interface (GUI) in the main navigation pane and the drop-down menu of vCenter Server. The GUI displays the following data.

- GPU allocation
- Memory and compute resources use
- Network traffic
- Logging reports
- Health reports

You can use the plug-in to manage allocation limits and idle intervals. You can also perform other management functions, such as ending client connections, gracefully taking servers offline, and removing hosts from the vSphere Bitfusion cluster.

Managing vSphere Bitfusion with the vSphere Client

- You can create a snapshot of a vSphere Bitfusion server virtual machine (VM), but first you must power off the VM. Taking a snapshot while the VM is powered on may result in failure of the operation due to the pass-through devices that are connected to the server.
- You can perform a graceful shutdown or restart of a server VM by using the **Shut Down Guest OS** and **Restart Guest OS** options in the vSphere Client. Using the power on, power off, suspend, and reset options may result in failure of the vSphere Bitfusion appliance.

Managing vSphere Bitfusion with CLI commands

You can start and stop the vSphere Bitfusion service, start applications with an alternative vSphere Bitfusion server list that is a subset of the primary server list of GPU servers, and monitor vSphere Bitfusion by using CLI commands.

This chapter includes the following topics:

- [Back Up a vSphere Bitfusion Cluster](#)
- [Restore a vSphere Bitfusion Cluster](#)
- [Start and Stop the vSphere Bitfusion Service](#)
- [Download vSphere Bitfusion Monitoring Data](#)
- [Set a Global Display Refresh Interval](#)
- [Use a Subset List of vSphere Bitfusion Servers](#)
- [vSphere Bitfusion Configuration Files](#)
- [vSphere Bitfusion Commands Reference](#)
- [Monitoring vSphere Bitfusion in the vSphere Bitfusion Plug-In](#)
- [Monitoring vSphere Bitfusion in the CLI](#)

Back Up a vSphere Bitfusion Cluster

You can back up your vSphere Bitfusion database and save a snapshot of the configuration, connectivity, health state, and history data of your vSphere Bitfusion cluster.

You can save a snapshot of your vSphere Bitfusion cluster database and download the backup copy to a local machine.

Prerequisites

Verify that your cluster is in a healthy state and all vSphere Bitfusion servers are reachable.

Procedure

- 1 In the vSphere Client, select **Menu > Bitfusion**.

- 2 On the **Settings** tab, click **Backup/Restore**.
- 3 Click **New Backup**, enter a backup name, and click **Take Backup**.
The backup is listed in the table.
- 4 Select the backup and click **Download**.
- 5 (Optional) Select a location for the backup file on your local machine.

Restore a vSphere Bitfusion Cluster

You can restore your vSphere Bitfusion database and recover the cluster to a previously healthy state.

To recover your vSphere Bitfusion cluster from a failure or unhealthy state, you can use a backup file to restore the configuration, connectivity, health state, and history data of the cluster.

Prerequisites

- Verify that you have a backup of your vSphere Bitfusion environment.
- Verify that your vSphere Bitfusion cluster has one vSphere Bitfusion server in a healthy state.


Procedure

- 1 In the vSphere Client, select **Menu > Bitfusion**.
- 2 On the **Settings** tab, click **Backup/Restore**.
- 3 Restore the cluster.
 - a Click **Restore From Download**.
 - b Select a backup file.
 - c Click **Restore From Backup**.

Results

The restore operation might need several minutes to complete. During the process, a **Restore in progress** notification is displayed in the **Backup/Restore** pane.

What to do next

After the operation is completed and the vSphere Bitfusion plug-in is registered again with vCenter Server, click the refresh icon () to update all data in the current vSphere Client view.

Start and Stop the vSphere Bitfusion Service

You can stop and start vSphere Bitfusion to make a configuration change or perform debugging.

vSphere Bitfusion runs as a regular application on both vSphere Bitfusion servers and clients. A `systemd` service starts the vSphere Bitfusion server software when the vSphere Bitfusion server starts. To stop, start, and restart the vSphere Bitfusion service or check the service log, you must access a vSphere Bitfusion server by using command line. The `systemd` file is in `/lib/systemd/system/bitfusion-manager.service`.

Note Typically, administrators and users do not interact with the vSphere Bitfusion server from the CLI. The interaction must be performed by using the vSphere Bitfusion Plug-in.

Procedure

- 1 Open a terminal application and run `ssh customer@ip_address`.

You can obtain the vSphere Bitfusion server IP address from the vSphere Bitfusion Plug-in.

- 2 Enter the customer password that you specified during the deployment of the vSphere Bitfusion open virtual appliance (OVA).
- 3 Start, stop, or monitor the vSphere Bitfusion service.

You can use the alias `bitfusion` for `bitfusion-manager.service`.

Action	CLI Command
Check the Bitfusion service	<code>sudo systemctl status bitfusion</code>
Stop the Bitfusion service	<code>sudo systemctl stop bitfusion</code>
Start the Bitfusion service	<code>sudo systemctl start bitfusion</code>
Restart the Bitfusion service	<code>sudo systemctl restart bitfusion</code>
Check the Bitfusion service log	<code>sudo journalctl -u bitfusion-manager.service</code>
	Note You cannot use an alias.

Download vSphere Bitfusion Monitoring Data

You can download monitoring data of your vSphere Bitfusion cluster, servers, and clients in the vSphere Bitfusion Plug-in.

By exporting monitoring data, you can use external tools to review and troubleshoot your vSphere Bitfusion environment. The **Download CSV** button on each tab in the vSphere Bitfusion plug-in, provides you with a different monitoring data set. You can download monitoring data for the past five minutes, one hour, 24 hours, and 30 days.

Procedure

- 1 In the vSphere Client, select **Menu > Bitfusion**.
- 2 Select a time period for the monitoring data.

3 Download the required monitoring data.

Option	Action
Cluster data	To save the cluster GPU allocation data, on the Cluster tab, click Download CSV .
Servers data	To save the data that is displayed for the selected server and pane, on the Servers tab, click Download CSV .
Clients data	To save the data that is displayed for the selected client and pane, on the Clients tab, click Download CSV .

4 (Optional) Select a location for the `.csv` file on your local machine.

Set a Global Display Refresh Interval

You can configure the vSphere Bitfusion Plug-in to refresh the data that it displays for clusters, servers, and clients regularly.

The refresh interval controls how often the vSphere Bitfusion Plug-in refreshes the displayed information. Alternatively, you can disable the automatic refresh in the GUI and manually press the **Refresh** button or navigate to a new tab.

Procedure

- 1 In the vSphere Client, select **Menu > Bitfusion**.
- 2 On the **Settings** tab, click **Application settings**.
- 3 Set a global refresh interval.
 - a Select the **Enable Refresh** check box.
 - b Enter a **Refresh Interval**.
The value is in seconds.
- 4 Click **Save**.

Use a Subset List of vSphere Bitfusion Servers

You can run the `run` command with an alternative vSphere Bitfusion server list that is a subset of the primary server list of GPU servers maintained by vSphere Bitfusion in the `~/.bitfusion/servers.conf` file.

vSphere Bitfusion supports IPv4 addresses only.

Procedure

- ◆ Perform one of the following steps.
 - To use a subset file of the primary servers, run `bitfusion run --servers value, -s value`.

You must change the *value* argument with a filepath to a `servers.conf` file.

- To create a subset list of vSphere Bitfusion servers, run `bitfusion run --server_list value, -l value`.

You must change the *value* argument to a "*ip_address:port;ip_address:port*" format. Enclose the list within quotes, because a semicolon is used as a separator when you list multiple addresses and the command-line interpreter can parse the list as multiple commands.

vSphere Bitfusion Configuration Files

After you start a vSphere Bitfusion server instance, vSphere Bitfusion creates and maintains `servers.conf` and `bitfusion-limits.conf` configuration files on the client virtual machines (VMs). The client VMs must be deployed on ESXi hosts that are part of the same vCenter Server environment as the vSphere Bitfusion server instance.

Servers Configuration File

vSphere Bitfusion creates a high-priority user-specific file in `~/.bitfusion/servers.conf`. Alternatively, you can create a system file `/etc/bitfusion/servers.conf`, which vSphere Bitfusion uses with a lower priority than the user-specific file. You use the `cat` command to display a server list.

To understand the command use, see the following example.

```
cat ~/.bitfusion/servers.conf
```

The servers configuration file lists the IPv4 addresses of all vSphere Bitfusion servers and ports that a vSphere Bitfusion client can access. The default port 56001 is not listed.

```
172.31.51.20
172.31.51.26:56003
172.31.51.42 56003
```

Limits Configuration File

The following limits apply to members of the vSphere Bitfusion group. Any user of the vSphere Bitfusion client must be a member of the vSphere Bitfusion group.

The `bitfusion-limits.conf` configuration file is installed on the vSphere Bitfusion client in `/etc/security/limits.d/bitfusion-limits.conf` by the client package. The file contains the following settings, which you can view and enforce by using the standard Linux utility, `ulimit`.

- Maximum number of open files

```
@bitfusion soft nofile 100000
@bitfusion hard nofile 100000
```

- Unlimited locked-in-memory address space

```
@bitfusion soft memlock unlimited
@bitfusion hard memlock unlimited
```

- Unlimited maximum resident set size

```
@bitfusion soft rss unlimited
@bitfusion hard rss unlimited
```

Note If the resource limit for open files is too low, vSphere Bitfusion might receive a connection error: `Cannot allocate memory error`. To resolve this issue, set the open files limit to 4096 or higher by running the `ulimit -n 4096` command.

vSphere Bitfusion Commands Reference

This section lists the most important vSphere Bitfusion CLI commands and their tasks. Additional CLI commands can be provided by the VMware support team.

Allocate GPUs

To allocate a number of GPUs for a single application, run the `run` command.

To allocate a number of GPUs and start a session, wherein you can run multiple applications on the same GPUs, run the `request_gpus`.

Start Applications in the vSphere Bitfusion Environment Accessing the GPUs

To start a single application, run the `run` command.

To start multiple applications in a session started with the `request_gpus` command, run the `client` command.

Deallocate the GPUs

To deallocate GPUs in a session started with the `request_gpus` command, run the `release_gpus` command.

List Available GPUs

To verify a vSphere Bitfusion server installation and find a list of available GPUs, run the `list_gpus` command.

```
- server 0 [172.31.51.20:56001]: running 0 tasks
|- GPU 0: free memory 12000 MiB / 12000 MiB
|- GPU 1: free memory 12000 MiB / 12000 MiB
|- GPU 2: free memory 12000 MiB / 12000 MiB
|- GPU 3: free memory 12000 MiB / 12000 MiB
```

```
- server 1 [172.31.51.26:56003]: running 0 tasks
|- GPU 0: free memory 12000 MiB / 12000 MiB
|- GPU 1: free memory 12000 MiB / 12000 MiB
- server 2 [172.31.51.42:56003]: running 0 tasks
|- GPU 0: free memory 12000 MiB / 12000 MiB
|- GPU 1: free memory 12000 MiB / 12000 MiB
```

Run a Health Check

You can access the health check from the command line.

- To check the health of all vSphere Bitfusion servers and the Bitfusion client, run `bitfusion health`.
- To check the health of a single vSphere Bitfusion client or server, run `bitfusion localhealth`.

Check vSphere Bitfusion Version

To check the version of vSphere Bitfusion that is installed, run the `version` command.

```
Bitfusion version: 2.5.0 release
```

Display GPU Information

To display GPU information, run the `smi` command. Alternatively, to receive a similar output, you can start the `nvidia-smi` application with the `run` command.

```
+-----+
| 172.16.31.243:56001 | Driver Version: 440.64.00 |
+-----+
| GPU  Name          Persistence-M | Virt Mem    Alloc / All | BusId  Vol Uncorr ECC |
| Fan  Temp  Perf      Pwr:Usage/Cap | Phy Mem    Used  / All | GPU-Util  Compute M. |
|=====+=====+=====+=====+=====+=====+=====+=====+=====+
| 0    Tesla T4      Disabled | 0          MB / 15109  MB | 00000000:13:00.0  0 |
| 0 %   36C   P8      10W /  70W | 11         MB / 15109  MB |    0%           Default |
+-----+
+-----+
| 172.16.31.241:56001 | |
+-----+
```

Test the Bandwidth

To test the bandwidth and latency between the vSphere Bitfusion client and servers, run the `net_perf` command.

Single network interface

```
Displayed results are calculated from round-trip measurements
BW(1MB) = 1000/(LAT(1MB) - LAT(1B))

[ <client>] ens160 => [10.202.8.169] net1 ( tcp) Single packet lat = 51 us, bw(1MB) = 1.71
```

```

GB/s
[ <client>] ens160 => [10.202.8.185] net1 ( tcp) Single packet lat = 48 us, bw(1MB) = 1.09
GB/s
[ <client>] ens160 => [10.202.8.233] net1 ( tcp) Single packet lat = 50 us, bw(1MB) = 0.87
GB/s

```

Multiple network interfaces

```

Displayed results are calculated from round-trip measurements
BW(1MB) = 1000/(LAT(1MB) - LAT(1B))

[ <client>] ens160 => [10.202.8.169] net1 ( tcp) Single packet lat = 51 us, bw(1MB) = 1.71
GB/s
[ <client>] ens160 => [10.202.8.185] net1 ( tcp) Single packet lat = 48 us, bw(1MB) = 1.09
GB/s
[ <client>] ens160 => [10.202.8.233] net1 ( tcp) Single packet lat = 50 us, bw(1MB) = 0.87
GB/s
[ <client>] ens192f0 => [10.202.8.169] net2 ( tcp) Single packet lat = 47 us, bw(1MB) = 2.14
GB/s
[ <client>] ens192f0 => [10.202.8.185] net2 ( tcp) Single packet lat = 49 us, bw(1MB) = 1.11
GB/s
[ <client>] ens192f0 => [10.202.8.233] net2 ( tcp) Single packet lat = 50 us, bw(1MB) = 1.15
GB/s
[ <client>] vmw_pvrDMA0 => [10.202.8.169] vmw_pvrDMA0 (infiniband) Single packet lat = 19 us,
bw(1MB) = 3.66 GB/s Single packet Write lat = 8 us, bw = 10.101 GB/s
[ <client>] vmw_pvrDMA0 => [10.202.8.185] vmw_pvrDMA0 (infiniband) Single packet lat = 21 us,
bw(1MB) = 3.45 GB/s Single packet Write lat = 8 us, bw = 10.5263 GB/s
[ <client>] vmw_pvrDMA0 => [10.202.8.233] vmw_pvrDMA0 (infiniband) Single packet lat = 21 us,
bw(1MB) = 3.46 GB/s Single packet Write lat = 8 us, bw = 10.4167 GB/s

```

Request Help

To get the full list of vSphere Bitfusion CLI commands or more information about a specific command, run the `help` command.

```

NAME:
    bitfusion - Run application with VMware Bitfusion

USAGE:
    bitfusion <command> <options> "application"
    bitfusion <command> <options> -- [application]
    bitfusion help [command]

    For more information, system requirements, and advanced usage please visit
    docs.bitfusion.io

COMMANDS:
    tls-certs, TC    Manage TLS certificates used by bitfusion server.  Requires root
privileges.
    version, v       Display full Bitfusion version
    localhealth, LH  Run health check on current node only
    dealloc          Deallocate license certificate.  Requires root privileges.
    crashreport      Send crash report to bitfusion
    license          Check license status

```



```

list_gpus      List the available GPUs in a shared pool
initdb        Init database setup
token         Fetch and manipulate tokens
register       Register remote server as the plugin
unregister     Unregister remote plugin
removenode    Remove unavailable nodes
user          Manage bitfusion users
help, h       Shows a list of commands or help for one command

Client Commands:
client, c     Run application
health, H    Run health check on all specified servers and current node
request_gpus Request GPUs from a shared pool
release_gpus Release GPUs back into a shared pool. Options must match a previous
request_gpus command
run          Request GPUs from a shared pool, run a client command, then release the
GPUs
stats        Gather stats from all servers.
smi          Display smi-like info for all servers.
local        Run a CUDA application locally
net_perf     Gather network performance data from all SRS servers.

Server Commands:
server, s     Run dispatcher service - listens for 'bitfusion client'
commands
resource_scheduler, srs Run Bitfusion resource scheduler (SRS) on GPU server
analytics     Run Bitfusion analytics server
manager       Run Bitfusion manager server

EXAMPLES:
$ sudo bitfusion init -l <license_key>

$ bitfusion resource_scheduler --srs_port 50001

$ bitfusion run -n 4 -- <application>

```

Monitoring vSphere Bitfusion in the vSphere Bitfusion Plug-In

You can view IP addresses, host names, GPU allocation, memory use, and other data of your vSphere Bitfusion cluster, servers, and clients in the vSphere Bitfusion Plug-in.

Monitoring vSphere Bitfusion Cluster

You can use the vSphere Bitfusion Plug-in to view the following data for your cluster.

- The IP address of the primary vSphere Bitfusion server. The vSphere Bitfusion Plug-in uses the IP for communication.
- The allocation history of GPUs, shown in the Cluster GPU Allocation chart. The chart covers a range from the last 5 minutes to the last 30 days, the number of GPUs populating the cluster, and the number of GPUs allocated from all vSphere Bitfusion servers.

- All vSphere Bitfusion servers in the vSphere Bitfusion cluster, including servers that have been disabled or powered off, shown in the Servers table. Each entry displays a host name, IP address, and the number of the allocated GPUs.
- All vSphere Bitfusion clients that have run applications on the vSphere Bitfusion servers, shown in the Clients table. Each entry lists a host name, ID, and the number of GPUs currently allocated to the client.

Monitoring vSphere Bitfusion Servers

You can use the vSphere Bitfusion Plug-in to view the following data for your servers.

- All vSphere Bitfusion servers in the vSphere Bitfusion cluster, shown in the Servers table. You can select any server to display the server details. The table displays each server's host name, IP address, current GPU allocation, and the current health state.
- A heat map with an entry for each GPU on the server, shown in the Allocation chart. Each cell displays by intensity of color how engaged the GPU is during the selected time interval . The level of engagement is a weighted sum of memory allocation and CUDA cell use.
- Memory and core use charts, one pair for each GPU. The Memory charts also show the memory capacity.
- The outgoing and incoming traffic for each network interface.

Monitoring vSphere Bitfusion Clients

You can use the vSphere Bitfusion Plug-in to view the following data for your clients.

- All vSphere Bitfusion clients in the vSphere Bitfusion cluster, shown in the Clients table. A new entry appears on the list after a new client runs a vSphere Bitfusion command that requires a server connection for the first time. You can select a client to display the client details. The table displays each client's host name, ID, current GPU allocation, and version.
- The GPUs that are allocated to a client, shown in the GPU Assignment chart. A client can run multiple applications, each allocating separate GPUs, but they are displayed together. Allocations of partial GPUs add the fractional value to the sum.

Monitoring vSphere Bitfusion in the CLI

By using CLI commands, you can check the shadow memory of a vSphere Bitfusion client, the MTU size of your network, and the network interfaces for error statistics and dropped packet counts.

Shadow Memory Check

The vSphere Bitfusion client uses a part of its memory space as a shadow memory of the allocated remote GPU memory. The precise amount of memory required on the client host varies between applications. The shadow memory check determines if the host's memory is as large as the GPU memory. For more information about memory requirements, see the *System Requirements for vSphere Bitfusion* topic in the *VMware vSphere Bitfusion Installation Guide*.

You can see the amount of memory on your client from the `MemTotal` line of the pseudo file `/proc/meminfo`. To calculate the GPU memory, from a GPU server, you can run the `bitfusion smi` or `nvidia-smi` command, and add up the memory sizes of all GPUs.

You can add more memory to the vSphere Bitfusion client to meet the requirement. Alternatively, when you run applications, do not allocate more GPUs than you can shadow in the memory of the vSphere Bitfusion client.

MTU Size Check

The vSphere Bitfusion performance relies on a healthy, low-latency, and high-speed network. Applications perform better when they send a few large packets instead of many small packets. The maximum transfer unit (MTU) check determines whether you have a large (≥4K) MTU setting for all high-speed (≥10 Gbps) interfaces. Ignore this check for interfaces you do not use with vSphere Bitfusion.

Note For best performance of applications running under vSphere Bitfusion, set the MTU to 4096 or higher and set vSphere Bitfusion clients to match the MTU size of the deployed vSphere Bitfusion servers. If the MTU is above 1500, enable jumbo frames in the network switches.

To obtain and set the MTU size, see the following examples.

- To check the MTU size, you can run the `ifconfig` command.
- To change the MTU size on network interface `enp175s` to 4096 bytes, you can run `ifconfig enp175s mtu 4096`.

For more information on MTUs, see [Determine maximum MTU](#).

Network Errors Check

You can check the network interfaces for error statistics and dropped packet counts. The files are in the following locations.

```
/sys/class/net/<interface>/statistics/*errors
/sys/class/net/<interface>/statistics/*dropped
```

If your network is healthy, the error count between the checks does not increase, new error messages do not occur, and no packets are dropped. The files are zeroed out only after a reboot.

Troubleshooting vSphere Bitfusion

7

The vSphere Bitfusion troubleshooting topics provide solutions to problems that you might encounter when performing tasks by using the vSphere Bitfusion plug-in and the command-line interface (CLI).

This chapter includes the following topics:

- [vSphere Bitfusion Client ID Changes](#)
- [Deleted vSphere Bitfusion Clients Can Request GPUs](#)
- [vSphere Bitfusion Client Cannot Connect to the vSphere Bitfusion Servers](#)
- [vSphere Bitfusion Server Cannot Start](#)

vSphere Bitfusion Client ID Changes

After you request a GPU for the first time, the ID of a vSphere Bitfusion client with version 2.0.2 and earlier changes.

Cause

When a virtual machine of a vSphere Bitfusion client with version 2.0.2 and earlier is enabled, the client ID appears in the vSphere Bitfusion plug-in. After the client requests GPUs for the first time, this ID changes.

Solution

Upgrade your vSphere Bitfusion client and servers to the latest version. For more information, see *Upgrading vSphere Bitfusion* in the *VMware vSphere Bitfusion Installation Guide*.

Deleted vSphere Bitfusion Clients Can Request GPUs

vSphere Bitfusion clients with version 2.0.2 and earlier can still request GPUs after being deleted from the cluster.

Cause

After you delete a vSphere Bitfusion client version 2.0.2 and earlier by using the vSphere Bitfusion plug-in, the client can continue requesting GPUs from the vSphere Bitfusion servers.

Solution

- If you enabled the client by using the vSphere Bitfusion plug-in, in the virtual machine terminal of the vSphere Bitfusion client, run the following commands.

```
a vmtoolsd --cmd 'info-set guestinfo.bitfusion.client.accesstoken'
```

```
b rm ~/.bitfusion/client.yaml
```

- If you enabled the client by generating an authorization token, use the vSphere Bitfusion plug-in to revoke the token of the client.
- Alternatively, you can upgrade your vSphere Bitfusion client and servers to the latest version. For more information, see *Upgrading vSphere Bitfusion* in the *VMware vSphere Bitfusion Installation Guide*.

vSphere Bitfusion Client Cannot Connect to the vSphere Bitfusion Servers

There are several scenarios when your vSphere Bitfusion client might be unable to connect to the vSphere Bitfusion servers in your cluster.

Problem

When trying to connect a vSphere Bitfusion client to the servers in your cluster, the connection fails. Typically, you see an error message that is similar to the following example.

```
Error querying server 10.115.27.120:56001:
Get https://10.115.27.120:56001/query: x509: certificate signed by unknown authority
Unable to contact any of the existing servers: 10.115.27.120:56001
```

Cause

- When you are moving or cloning a vSphere Bitfusion client to a new vCenter Server instance.
- When you install new vSphere Bitfusion servers, resulting in a new cluster creation, and you use your old vSphere Bitfusion clients.
- When you have an invalid vSphere Bitfusion server in your cluster.

Solution

- If you have an invalid vSphere Bitfusion server in your inventory, by using the vSphere Bitfusion plug-in, remove the server from the cluster.

For more information, see [Remove a vSphere Bitfusion Server](#).

If the server cannot be removed from the inventory by using the vSphere Bitfusion plug-in, in the terminal of a running vSphere Bitfusion server, run the `bitfusion removenode` command.

- Verify that all steps during the vSphere Bitfusion client enablement process are completed.

For example, verify that the virtual machine of the vSphere Bitfusion client is powered off during the enablement process.

For more information, see *Enabling the vSphere Bitfusion Client* in the *VMware vSphere Bitfusion Installation Guide*.

- In the virtual machine terminal of the vSphere Bitfusion client, run the following commands.

```
a  rm ~/.bitfusion/client.yaml
```

```
b  rm ~/.bitfusion/servers.conf
```

After the `client.yaml` and `servers.conf` are deleted, reenable the vSphere Bitfusion client.

For more information, see *Enabling the vSphere Bitfusion Client* in the *VMware vSphere Bitfusion Installation Guide*.

- If none of the previous steps resolve the issue, create logs from a vSphere Bitfusion server and send the support bundle to VMware support.

For more information, see [Create vSphere Bitfusion Server Logs](#).

vSphere Bitfusion Server Cannot Start

There are several scenarios when the virtual machine of your vSphere Bitfusion server cannot start due to GPU related issues.

Problem

When you power on the virtual machine of your vSphere Bitfusion server, the virtual machine cannot start.

Cause

Typically, the following scenarios are observed during the installation process of a new vSphere Bitfusion server.

- When you add multiple times the same GPU to a virtual machine of a vSphere Bitfusion server.
- When the total memory of the GPUs used on a vSphere Bitfusion server is more than 128 GB.
- When you use a GPU that is already assigned to a running vSphere Bitfusion server.

Solution

- If you add the same GPUs multiple times, vCenter Server adds the first GPU multiple times. You must manually update the ID of the PCI bus for the additional GPUs with a unique value.
 - a In the vSphere Client, right-click the virtual machine of the vSphere Bitfusion server and select **Edit Settings**.
 - b From each **PCI Device** drop-down menu, select a unique ID for the GPU.

- If the total memory of the GPUs used on a single vSphere Bitfusion server is more than 128 GB, you must change the value of the `pciPassthru.64bitMMIOSizeGB` property, which is the advanced virtual machine property for GPU passthrough.
 - a Calculate a correct value for the property. Count the number of PCI devices, such as GPUs and network cards, that a vSphere Bitfusion server virtual machine uses, multiply the number by the GPU size in GB, and round up the value to the next power of two. For example, to use GPU passthrough with two 16 GB GPU devices, round up the value to 64 ($2 * 16 = 32 * 2 = 64$). For a single 16 GB GPU, use a value of 32.
 - b Modify the virtual machine property.
 - 1 In the vSphere Client, select the virtual machine of the vSphere Bitfusion server, and power it off.
 - 2 With the virtual machine selected, select **Actions > Edit Settings > VM Options > Advanced > Edit Configuration**.
 - 3 Search for `pciPassthru.64bitMMIOSizeGB` and set a new value.
 - 4 Power on the virtual machine.
- If the GPU that you are assigning to a virtual machine of a vSphere Bitfusion server is already assigned to a running server, you must select a different GPU. You can pass through one GPU to one vSphere Bitfusion server.