

Photon Controller Getting Started Guide

End User Workflows

Photon Controller is a distributed multi-tenant control plane that, in a production environment, would be deployed across several hosts to provide high availability and scale. This environment provides a minimal functional deployment on a local Fusion or Workstation hypervisor. The goal is to provide a working environment to allow testing of basic controller functionality, but will be severely limited in terms of the number of Clusters and VMs that can be created. The installation itself is very opinionated. **We have seeded meta-data in the Controller to minimize the steps required for installation, so you must use names and network configurations that we have pre-defined.**

Requirements:

Windows or Mac Desktop with 16 GB RAM and 50GB available Disk space

Note: if your machine has less than 16GB RAM, you may still be able to run this by enabling memory overcommit for Fusion or workstation. This configuration has not been completely test. To enable overcommit, add the following line to your config file:

```
prefvmx.minVmMemPct = 25
```

The config file is located as follows:

C:\ProgramData\VMware\VMware Workstation\config.ini for Workstation
/Library/Preferences/VMware\Fusion/config for Fusion

Note: This file may not exist by default and you will have to create it

VMWare Fusion 8

VMware Workstation 12

ESXi 6.0 installed in local Workstation or Fusion VM

ESXi VM must be configured with IP 192.168.209.31

ESXi VM must be configured with datastore named datastore1

ESXi VM must have root password that meets esxi v6 complexity rules. Make sure it has at least 8 characters not counting an uppercase first character, at least 1 upper case, at least 1 lowercase and at least one special character.

Note: Other versions of the required software may work, but these are the versions we have tested.

Note2: Some of these commands are long and unfortunately do not copy/paste well from a PDF. If you do not want to type them, you might try copy/paste into word and then copy/paste from there to your command line.

Installation workflow

- 1) Install VMware Fusion or Workstation on your laptop

[Download Evaluation of VMware Fusion](#)

[Download Evaluation of VMware Workstation](#)

Note: A Free Trial License is Available at this link.

Our deployment will use a private network for your VMs and will assume that the default Fusion/Workstation private network is being used in your deployment. That means NAT'd VMs should be deployed on the 192.168.209.x network. If that is not your default NAT network, you must change it.

Instruction are in step 2) below. IPs between 192.168.209.3-128 are Static and 192.168.209.129-254 are in the defined DHCP scope.

2) Create a VM and install ESXi 6.0.

[Download Evaluation of ESXi 6.0.0](#)

- a. Click on Download esxi 6.0, you will be taken to the MyVMware login screen. Login and download ESXi 6.0.

Note: If you do not already have a MyVMware account, click on "Create An Account" to acquire one.

Fusion 8 ESXi VM Creation Steps

- From Fusion, select File → New
- Select Create a Custom Virtual Machine → VMware ESX → ESXi 6
- Select Create New Virtual disk
- Select Custom Settings → Save
- You should now see the Settings Menu. Select CD/DVD (IDE)
- Click Show All → Processors & Memory → set to 8192 MB
- Select the dropbox and choose your downloaded ISO file

Note: If you have 16GB or greater, or turned on memory overcommit, you can make this vm larger. A larger esxi host will allow the creation of more/larger clusters. 8GB is a minimal configuration. The default disk size is 40 GB, but you may be able to lower this if you are short on space.

- Close Settings and Start the VM. ESXi will now begin installing. You will need to provide your OS Admin password.
- If you are not using Fusion8 (or are on Fusion8 through upgrade), the default NAT address space is different from what is required for this deployment and must be changed. Depending on your version of Fusion, you may be able to add a second NAT network so that existing NATd VMs can keep their same connectivity. Setting up a second NAT network is beyond the scope of this document, but see the FAQ at the end of this document for the general steps. To change your existing NAT network:
 - Quit Fusion
 - From a terminal shell
cd /Library/Preferences/VMware\Fusion
"Yes the \ and blank space are correct"
 - vi networking

- Modify the Netmask and subnet settings for the VNET_8 adapter as follows:
subnet 192.168.209.0
netmask = 255.255.255.0
- Restart Fusion

Workstation 12 ESXi VM Creation Steps

- Adjust the NAT Network IP Select → Edit → Virtual Network Editor
 - Select VMnet8 → change subnet IP to 192.168.209.0 → OK
 - From Workstation, Select File → New Virtual Machine
 - Select Typical Install → Installer Disc Image and select your downloaded ESXi ISO.
 - Select Defaults until “Customize Hardware” and set VM Memory to 8192MB
 - At “Specify Disk Capacity” → Store virtual disk as a single file.
 - Install will start Automatically in a few seconds
- b. Now Install is Complete, Accept the Prompts with Enter
- c. Accept Eula
Fusion: send F11 Key from Virtual Machine → Send Key Menu → F11
Workstation: If F11 doesn't work, try CTRL-F11 or fn-CTRL-F11
- d. Accept Default Disk and Language
- e. Create ESXi Root Password
- f. Confirm Install by again sending the F11 key. Note: you may have to hit the Control and Command keys together to get the cursor out of the VM console Window.
- g. After Install Completes, disconnect the CD and hit Enter to Reboot.
- h. The ESXi Host VM must be configured with the following Static Network:
IP: 192.168.209.31
Gateway: 192.168.209.2
DNS: 192.168.209.2
Netmask: 255.255.255.0
Select Configure Management Network → Enter → IPv4 Configuration
- i. Select Set static IPv4 Address and network configuration using the space bar and change IP address to 192.168.209.31 → Enter
- j. Select <Esc> and Yes to restart Management Network
- k. Select Test management Network → Enter to verify connectivity.
- l. Enable SSH, Select Troubleshooting Options → Enable SSH → Esc

Note: This VM must be configured with the following datastore name: datastore1 This is the default, but it should not be changed.
Also, A licensed Standard, Enterprise or Enterprise+ ESXi VM is

required for Photon Controller. As of this moment, the Free esxi hypervisor is not supported. If you used the link above to download Esxi, you have a 60 day free trial license. After the evaluation period, you must register for a VMware product license.

3) Install New ESXi Embedded Host Client onto your ESXi host. (Optional)

This is not required to run Photon Controller, but If you haven't seen this, it rocks!! It's a new web client tech preview that runs against an individual host and does not require vCenter. BTW, for the Workstation version, if you have it, you can run the vSphere C# client, but the standard web client requires vCenter. [Download Technology Preview of ESXi Host Client](#)

- a. Click Download
- b. Installation instructions are on the Instructions Tab. That is the source of truth for installation, but as of this writing, the steps were as below.
- c. `scp "filename".vib root@192.168.209.31:/tmp`
- d. `ssh root@192.168.209.31 esxcli software vib install -v /tmp/"filename".vib --force`
- e. From Browser: <http://192.168.209.31/ui> root/YourESXiVMPassword

4) Download and Deploy the Photon Controller CLI (GO CLI)

Fusion 8 CLI install :

[Photon Controller CLI for Mac](#)

After downloading: `chmod +x photon`
`mv photon /usr/local/bin`

Workstation 12 CLI install:

[Photon Controller CLI for Windows](#)

After downloading, add .exe to the PATH -
`PATH %PATH%;<path to the photon directory>`

5) Download and Deploy the Photon Controller Control Plane OVA

This VM contains all of the Photon Controller Services running in Docker containers within the Photon OS.

- a. Download OVA File
[Photon Controller Devbox Control Plane OVA](#)

Fusion 8 Photon Controller VM Deployment Steps

- Import the ova file into Fusion: File → Import → Choose File
- Select Open → Continue → Save → Finish
- Verify that the API server is available. From your browser: <https://192.168.209.29:443/api> You should see a webpage with a green title bar that reads “apife” and a browseable list of URLs exposed by the API, generated by swagger. If you get a browser timeout or see a “503 service unavailable” error, this is normal. Wait a few minutes, refreshing the browser periodically. If after a few minutes the page does not come up, restart the VM, and try again.

Workstation 12 Photon Controller VM Deployment Steps

- Click on “open a new virtual machine” in workstation window.
- In the pop window, give the path of the ova file downloaded. workstation will create the VM.
- Verify that the API server is available. From your browser: <https://192.168.209.29:443/api> You should see the swagger interface. If you do not see this in a minute or so, then restart the VM.

This VM is configured with the following Static Network:

IP: 192.168.209.29

Gateway: 192.168.209.2

DNS: 192.168.209.2

Netmask: 255.255.255.0

The root password is vmware

6) Add Host To your Photon Controller Deployment. You have created the esxi VM, but the Photon Controller does not know about it. We are telling Photon Controller that this host can be used to place “CLOUD” resources. Think of a Cloud host as the place for Tenant/User VMs, disks or Clusters. The first command is to point your CLI to the Photon Controller Endpoint.

- Set the target: *photon target set* <https://192.168.209.29:443>
- photon host create --username root --password*

YourESXiVmPassword --tag CLOUD --address 192.168.209.31

Hit Enter when prompted for MetaData

This command updates the local cloudstore with the Host metadata and installs the Photon Controller Agent VIB on your host.

- Very Host has been added is in listed as “READY”:
photon host list

7) We have pre-created a Photon Tenant, Resource Ticket, Project and uploaded a set of VM and Disk flavors. You must now point your CLI to your tenant and your project.

Note: We have included a brief discussion of each of these at the bottom of this

document. See the Photon Controller Getting Started Guide for a discussion of these objects.

- a. Set the Tenant: *photon tenant set demo*
- b. Set the Project: *photon project set dev-project*

8) Upload base images into Photon Controller shared image datastore.

In a production environment you would have a set of images that would be made available to users for VM and Persistent Disk creation. These images might be replicated across many hosts to facilitate rapid local object creation on that host. In our environment, we are using a single local datastore attached to our esxi host and we need to upload our images there.

- a. Download Kubernetes Image [Photon Controller Kubernetes Image](#)
- b. Download Mesos Image [Photon Controller Mesos Image](#)
- c. Download Swarm image [Photon Controller Swarm Image](#)
- d. Upload each of these images to Photon controller:
*photon image create KubernetesImagefilename
-n photon-kubernetes-vm.vmdk -i EAGER
photon image create MesosImagefilename -n photon-mesos-vm.vmdk
-i EAGER
photon image create SwarmImagefilename -n photon-swarm-vm.vmdk
-i EAGER*

Note: The images must have the names listed here in the -n clause. Also, only upload the images for the cluster types you want to deploy. It takes a little time for the upload and consumes space on your datastore.

- e. Verify that the images uploaded successfully
photon image list

Kubernetes Cluster Deployment

10) Create a 2 Node Kubernetes Cluster and Deploy the GuestBook App

```
photon cluster create -n Kube2 -k KUBERNETES --dns 192.168.209.2  
--gateway 192.168.209.2 --netmask 255.255.255.0 --master-ip  
192.168.209.35 --container-network 10.2.0.0/16 --etcd1 192.168.209.36 -s 2
```

Note: --Container-network should be in CIDR format. Ex. 10.2.0.0/16
Also, when prompted for etcd2, hit return. We are only creating 1 because of the size of our environment.

Note2: We can create these clusters with very small VMs, allowing you to create larger or more clusters. See the Best Practices and Troubleshooting section at the end of this document for details

Note3: If you want to see the VMs that were created on your esxi host VM, connect to the host client you may have installed in Optional step 3) above. From your browser: <http://192.168.209.31/ui> esxi vm username and password. In the navigation pane on the left side, click on VMs to see the list.

- a. List Cluster
photon cluster list
- b. View Cluster Details
photon cluster show "uuid of cluster"
- c. Show Cluster VMs
photon cluster list_vms "uuid of cluster"
- d. After you deploy your app on this cluster, You may need to delete it in order to create additional clusters in this small environment. When you are ready to delete it, run the following command.
photon cluster delete "uuid of cluster"

11) Deploy Basic Web Server Application onto this Kubernetes cluster.

- a. Download Kubectl utility

[Download Kubectl for Linux or MacOS](#)

Make it executable and put it in your path

```
chmod +x kubectl
```

```
mv kubectl /usr/local/bin (Mac Only)
```

[Download kubectl Tar File for Windows](#)

Unzip and untar the downloaded zip file and use kubectl.exe from the source tree platforms/windows/amd64/kubectl.exe

- b. Download Replication Controller and Service files

[Apache Tomcat Replication Controller yml](#)

[Apache Tomcat Service yml](#)

- c. Create the application by deploying a replication controller and corresponding service. Note: Some users have noted 7-10 min image download time on first deployment. This has not been the case in all instances.

```
kubectl -s 192.168.209.35:8080 create -f  
PathToPhoton-Controller-Tomcat-rc.yml
```

```
kubectl -s 192.168.209.35:8080 create -f  
PathToPhoton-Controller-Tomcat-service.yml
```

- d. Check if the pods are running.

```
kubectl -s 192.168.209.35:8080 get pods
```

Look for a status of "Running". You now have a Tomcat Server running on your cluster.

- e. View the application page in your browser

<http://192.168.209.35:30001>

- f. Scale out to multiple Replication Controllers

```
kubectl -s 192.168.209.35:8080 scale --replicas=2 rc tomcat-server
```

- g. Verify that you now have two copies of the Tomcat Server Pod.

```
kubectl -s 192.168.209.35:8080 get pods
```

Mesos Cluster Deployment

- 12) We are going to deploy a 2 Node Mesos Cluster and create a simple app on the cluster:

Note: This will create 7 VMs. Master VMs will have 512MB of RAM and Workers will have 256MB. So, depending on the amount of RAM you gave your ESXi VM, you may need to delete other clusters in order for this to create successfully and remove this cluster in order to create a Kubernetes or Swarm cluster.

Execute:

```
photon cluster create -n Mesos2 -k MESOS --dns 192.168.209.2 --gateway  
192.168.209.2 --netmask 255.255.255.0 --zookeeper1 192.168.209.51 -s 2
```

Note: when prompted for zookeeper2, hit return. We are only creating 1 because of the size of our environment.

Note2: If you want to see the VMs that were created on your esxi host VM, connect to the host client you may have installed in Optional step 3) above. From your browser: <http://192.168.209.31/ui> esxi vm username and password. In the navigation pane on the left side, click on VMs to see the list.

- a. List Cluster to verify the create was successful
photon cluster list
- b. View Cluster Details and Find both the Master and Marathon VM IPs. Marathon is the Mesos init system and you will use it in a later step to deploy an app. Look for the vm with the name marathon-uuid and note its IP. For the Master VM, you can choose any of the 3 VMs with the name master-uuid. One of them will be the leader and you will be re-directed to that when automatically.
photon cluster show "uuid of cluster"
- c. Show Cluster VMs
photon cluster list_vms "uuid of cluster"
- d. After you deploy your app on this cluster, You may need to delete it in order to create additional clusters in this small environment. When you are ready to delete it, run the following command.
photon cluster delete "uuid of cluster"

13) Deploy a simple App on your Mesos cluster Using Marathon

- a. Connect to Marathon Server from browser: <http://marathonIP:8080>
You should have noted this IP from the previous section
- b. Click New App Button
- c. Enter "photon" in the ID field
- d. Use defaults for CPU, Mem and Disk
- e. Enter 5 for number of Instances
- f. In the Command Section, enter:
while [true]; do echo 'Photon Controller Rocks!!!' ;sleep 5; done

- This command will be run in 5 separate containers across the cluster
- g. Click create button
 - h. You will shortly see instances started one at a time on the worker node.
 - i. Expand the photon ID to see the running containers
 - j. Try the scale button to increase to 7 Instances
 - k. Connect to the Master Server from the Browser `http://MasterIP:5050`
You should have noted this IP from the previous section
 - l. See each of the active tasks
- Note: In this NAT deployment, the VMs on the private network cannot access each other via DNS, so you cannot Drill into the Sandboxes of these VMs. This would be possible if you set Static DNS records or connected a DNS server to your private network.

Swarm Cluster Deployment

14) We are going to deploy a 2 Node Swarm Cluster:

Execute:

```
photon cluster create -n Swarm2 -k Swarm --dns 192.168.209.2 --gateway  
192.168.209.2 --netmask 255.255.255.0 --etcd1 192.168.209.55 -s 2
```

Note: when prompted for etcd2, hit return. We are only creating 1 because of the size of our environment.

Note2: If you want to see the VMs that were created on your esxi host VM, connect to the host client you may have installed in Optional step 3) above. From your browser: <http://192.168.209.31/ui> esxi vm username and password. In the navigation pane on the left side, click on VMs to see the list.

*b. List Cluster
photon cluster list*

b. View Cluster Details and Find Master VM IP
photon cluster show "uuid of cluster"

15) Deploy a simple application on your Swarm Cluster

a. If you do not have a Docker Client, get it here:

<https://www.docker.com/docker-toolbox>

- b. Once you have found the Swarm Master VM IP you can set your docker CLI to point to it by setting the DOCKER_HOST variable.

```
export DOCKER_HOST="tcp://<Master VM IP>:8333
```

Now you can use your docker CLI against the Swarm cluster.

You have seen how easy it is to deploy container based applications into your environment using the application framework of your choice. To gain a more complete understanding of the scope of Photon Controller, please see the Photon Controller Getting Started Guide available here:

FAQ and Issues with the Document/Implementation

- 1) What if my host did not add successfully?
 - A. verify that you can ssh root@192.168.209.31 with password "VMware1!"
- 2) What if my Image did not upload successfully?
 - A. Verify the filepath you entered.
- 3) What if my Mesos Cluster create failed?
 - A. More than likely, you have run out of available resource. In this small environment, 1 Mesos cluster will take up most of the available RAM. Each VM in the cluster will require 1GB of RAM. Verify that you have at least 7GB of RAM available on your esxi host VM.
 - B. Verify that the static IPs you assigned for Zookeeper nodes are in the range of 192.168.209.3-127, are not in use, and that the gateway, dns and netmask are correct.
 - C. Verify that you have uploaded the Mesos image and the name starts with photon-mesos-vm
- 4) What is a Tenant?
 - A. Photon controller is a multi user system that abstracts physical resources and allocates them across users of the system. The tenant is allocated a pool of resource that can be consumed by users with access to this tenant. This pool is created through resource tickets. The process for providing resources to users to to create a tenant, create resource tickets with a fixed amount of CPU and RAM, then create projects that are allocated resource from the pool of tenant resources
- 5) What is a Project?
 - A. A project is a way to allocate the pool of resource assigned to a Tenant through Resource tickets. I Tenant can have many projects.
- 6) What is a Resource Ticket?

- A. A resource ticket is the mechanism for allocated specific pools of resource (ie. CPU, RAM) to Tenants and projects.
- 7) I don't want to change my existing vmnet8 network. Can I add a new network for this application?
- A. Depending on the version of Fusion or Workstation you are running, you may be able to add a second NAT network. Please check your Workstation or Fusion documentation for details.
- For Fusion execute the following:
- ```
sudo vmnet-cfgcli vnetcfgadd VNET_7_DHCP yes
sudo vmnet-cfgcli vnetcfgadd VNET_7_HOSTONLY_NETMASK 255.255.255.0
sudo vmnet-cfgcli vnetcfgadd VNET_7_HOSTONLY_SUBNET 192.168.209.0
sudo vmnet-cfgcli vnetcfgadd VNET_7_VIRTUAL_ADAPTER yes
sudo vmnet-cfgcli vnetcfgadd VNET_7_NAT yes
sudo vmnet-cli --configure
sudo vmnet-cli --stop
sudo vmnet-cli --start
```

### Best Practices and Troubleshooting

- 1) NAT network must use the 192.168.209.0 network. This is because we are not doing a normal Photon Controller deployment. We have already registered the various services in Zookeeper with this network and the expectation that the Control plane will on 192.168.209.29
- 2) When a host is made available to be used to place CLOUD vms, through the photon create host command, Photon Controller installs a VIB on the esxi host. This VIB is Controller agent and will immediately try to register itself on the IP and port of the Control plane (192.168.209.29). If you remove your Control Plane VM and deploy it, you must remove the VIB from your host. If you are seeing random internal errors, this may be the problem. The steps to remove the VIB and reset your environment are:
  - a) `ssh root@192.168.209.31` Enter your esxi VM password
  - b) `esxcli software vib list` and look for photon-controller-agent
  - c) `esxcli software vib remove -n photon-controller-agent`
  - d) restart the VM
  - e) Delete the ControlPlane VM and re-import. You will need to follow all of the steps from the Download the Photon Controller Control Plane OVA section above.
- 3) If you attempt to create a cluster and see messages containing a failure in "Allocate Resources" you may have used all the resources available on your esxi VM. You can

check this by logging in to the Host Client at 192.168.209.31. Click on host in the navigator and look at the CPU, Memory and Storage consumed. Memory will probably be the most constrained. Each VM is configured with a flavor that allocates 1GB of memory, so if you used the 8GB esxi recommendation in this guide, you can see how many VMs you might be able to create.

- 4) Where are the log files?
  - a) The logs are in /vagrant/log directory in PhotonControllerDevbox VM.
  - b) ssh root@192.168.209.29 password is vmware
  - c) The most useful logs to you will be management-api.log and deployer.log
    - i) Management-api.log captures the results of API calls and is where you can look for errors that come back to your command line
    - ii) Deployer.log, among other things, handles the logging for the cluster commands. So if you have an error in a cluster create for instance, you might start here.
  
- 5) The default configuration will generally support only a single cluster running at one time. Especially if its Mesos, because there are 3 Zookeeper, 1 Master and 1 Marathon VM in addition to the worker nodes you request. If you want to try different cluster types, then use the following commands to delete your cluster:
  - a) photon cluster list to get the uuid of the cluster
  - b) photon cluster delete uuid
  
- 6) Using Flavors to change the resources used by your clusters. First you will create a new flavor with a new CPU and Memory specification, next you will add that flavor to your Cluster Create command. Here is how to deploy a 2 worker node Mesos cluster with 2 CPU and 512 MB of RAM for each node.(Note: that when under 1GB, the amount of RAM must be a power of 2)
  - a) We will create the flavor using the CLI interactive mode:  
type: *photon flavor create* and hit Enter  
Flavor name: *small-mesos*  
Flavor kind (persistent-disk, ephemeral-disk, or vm): *vm*  
Limit 1 (Enter to finish)  
Key: *vm.cpu*  
Value: *2.0*  
Unit: COUNT  
Limit 2 (Enter to Finish)  
Key: *vm.memory*  
Value: *512*  
Unit: *MB*  
*Enter* and then *Enter*  
You now have a flavor called small\_mesos

- b) We will make a slight modification to the mesos cluster command used in the above Mesos section

```
photon cluster create -n Mesos2 -k MESOS --dns 192.168.209.2 --gateway 192.168.209.2 --netmask 255.255.255.0 --zookeeper1 192.168.209.51 -s 2 --vm_flavor small-mesos
```

Our default cluster creation flavor used 512MB for master VMs and 256MB for all other. With this new setup, each of our Mesos VMs will take up 512 MB of memory.

#### 7) Troubleshooting Reserve\_Resource Error:

If you see an error like this: 2015/11/10 01:08:12 esxcloud: Task '48b474d6-1773-4dca-a5cb-7a45a38e1904' is in error state. Examine task for full details. API Errors: [esxcloud: { HTTP status: '0', code: 'InternalError', message: 'Failed to rollout SwarmEtcd. Error: MultiException[java.lang.IllegalStateException: VmProvisionTaskService failed with error [Task "CREATE\_VM": step "RESERVE\_RESOURCE" failed with error code "InternalError", message "null"]]. /photon/clustermanager/vm-provision-tasks/24537ef5-5757-42ea-81ff-3b63cc5734b6}], data: 'map[]' }]

This means that the scheduler was unable to place the Vm on your host. There can be several causes. Your esxi host may be low on a resource, usually RAM in this environment. Or you have not correctly added the host to Photon Controller. It could also be because one or more of the Docker Containers in your Photon Controller have stopped. Do these things to troubleshoot:

- A. execute: Photon host list and verify that you have a single esxi host and it is in READY status
- B. login to 192.168.209.29 and execute docker ps. you should see 8 containers all with a status of Up. If one is not running, then reboot the VM and verify that they are up
- C. Connect to the Host Client from your browser and verify that there is available memory in your esxi host VM.
- D. The host client itself can use a lot of RAM. Verify from Monitor that your browser isn't using a tremendous amount of memory.

#### 8) Troubleshooting Cluster Failure Error

Error Msg: "Failed to rollout SwarmMaster"

Symptom: when creating cluster, the system hangs at step 4/5 for more than 10 minutes and then reports error.

Cause: DHCP service was not 'on' for VNET\_8.

Solution1: Set answer VNET\_8\_DHCP yes at /Library/Preferences/VMware\Fusion/networking

Set VNET\_8\_NAT yes as well

Solution2: Memory is very constrained in the devbox. Sometimes this causes the Photon Controller containers to be shut down by Docker when an out of memory condition occurs. Log in to 192.168.209.29 and execute `docker ps`. If any of those containers are not running, you can restart the the VM. If you have enough RAM, increase the memory on this VM beyond the default 3 GB. Check for Out of Memory errors by executing `dmesg`.

#### 9) Image Upload Fails

This can happen if you have previously run the Photon Host Create command and then deployed a new PhotonControllerDevbox VM without first removing the agent VIB from your esxi VM. If the agent process is running, it will attempt to register itself with your new PhotonControllerDevbox as soon as it comes up and you will now have old configuration in your new deployment. This can cause an error when trying to copy the image on upload. Follow the instructions in step 2 of Troubleshooting section to remove the agent VIB and then re-import your PhotonController Devbox VM.