

Using Project Photon on VMware vSphere 5.5 & 6.0

What is Project Photon?

Project Photon is a tech preview of an open source, Linux container host runtime optimized for vSphere. Photon is extensible, lightweight, and supports the most common container formats including Docker, Rocket (rkt) and Garden.

Project Photon includes a small footprint, yum-compatible, package-based lifecycle management system, and will support an rpm-ostree image-based system versioning.

When used with development tools and environments such as VMware Fusion, VMware Workstation, HashiCorp (Vagrant and Atlas) and production runtime environment (vSphere, vCloud Air), Photon allows seamless migration of container based Apps from development to production.

Introduction

This document explains how to get started using Project Photon as a runtime environment for Linux containers by running Project Photon as a virtual machine on VMware vSphere. This guide will provide step-by-step instructions on how to download Project Photon, provide details of the various install options and provide a walkthrough of installing the full Project Photon distribution.

Once Project Photon is installed, this guide will also provide instructions on how to demonstrate how simple it can be to deploy a containerized application with Docker and will highlight the installation of a web server simply by running one command!.

About the Author

Josh Gray is a Solution Engineer and a member of Global Technical and Professional Services - @jasper9



Photon Install - Prerequisites

In order to install and start using Project Photon with VMware vSphere the following pre-requisites must be satisfied:

Either download the latest Photon ISO from [here](#) or clone the [GitHub Photon repository](#) and build the ISO using the instructions found on the GitHub repo.

This guide is based on the VMware vSphere 6. Our recommendation is to always use the latest version; although, vSphere 5.5 onwards should work as advertised. Keep in mind, that Photon is being released, initially, as a tech preview – as such should you encounter any issues, please let us know in the Photon Community Forum, found [here](#).

Also, follow us @VMwarePhoton for news and updates.

Downloading Photon

As per the pre-requisites, either download the latest ISO image from [here](#) or clone the [GitHub Photon repository](#) and build the ISO.




Installing Photon from an ISO Image

Once the ISO image is uploaded to a datastore attached to the host on which you'll create the Photon VM, create a new virtual machine with the following settings.

Select a compatibility level for your VM, as shown below. Photon doesn't require any particular compatibility, but VMware recommends that you choose the newest available for your release of vSphere.

Select compatibility
Select compatibility for this virtual machine depending on the hosts in your environment

The host or cluster supports more than one VMware virtual machine version. Select a compatibility for the virtual machine.

Compatible with: 

This virtual machine uses hardware version 11 and provides the best performance and latest features available in ESXi 6.0.

For Guest OS Family choose "Linux", for Guest OS Version choose "Other 3.x Linux (64-bit)".

Select a guest OS
Choose the guest OS that will be installed on the virtual machine

Identifying the guest operating system here allows the wizard to provide the appropriate defaults for the operating system installation.

Guest OS Family:

Guest OS Version:



The recommended virtual hardware settings for your Photon VM are heavily dependent upon container load you intend to run within Photon. To start, we recommend 2 vCPU, 384MB memory, 20GB hard disk. Any unwanted devices should be removed. Be sure to mount the Photon ISO on the CD/DVD Drive and click Connect At Power On.

Customize hardware

Configure the virtual machine hardware

Virtual Hardware
VM Options
SDRS Rules

*CPU	2			
*Memory	384		MB	
New Hard disk	20		GB	
New SCSI controller	VMware Paravirtual			
*New Network	VM Network		<input checked="" type="checkbox"/> Connect...	
*New CD/DVD Drive	Datastore ISO File			
Status	<input checked="" type="checkbox"/> Connect At Power On			
CD/DVD Media	[LUN1]/photon.iso		Browse...	
Device Mode	Passthrough CD-ROM			
Virtual Device Node	IDE(1:0) New CD/DVD Drive			
New Floppy drive	Client Device		<input type="checkbox"/> Connect...	
Video card	Specify custom settings			
VMCI device				
Other Devices				



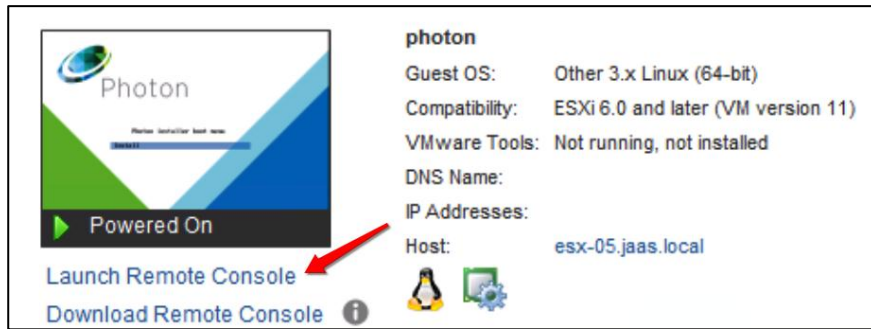
To summarize, these are the settings we recommend as a starting point for your Project Photon container runtime host:

Thin provisioned, hardware compatibility: ESXi 6.0 and later (VM version 11)

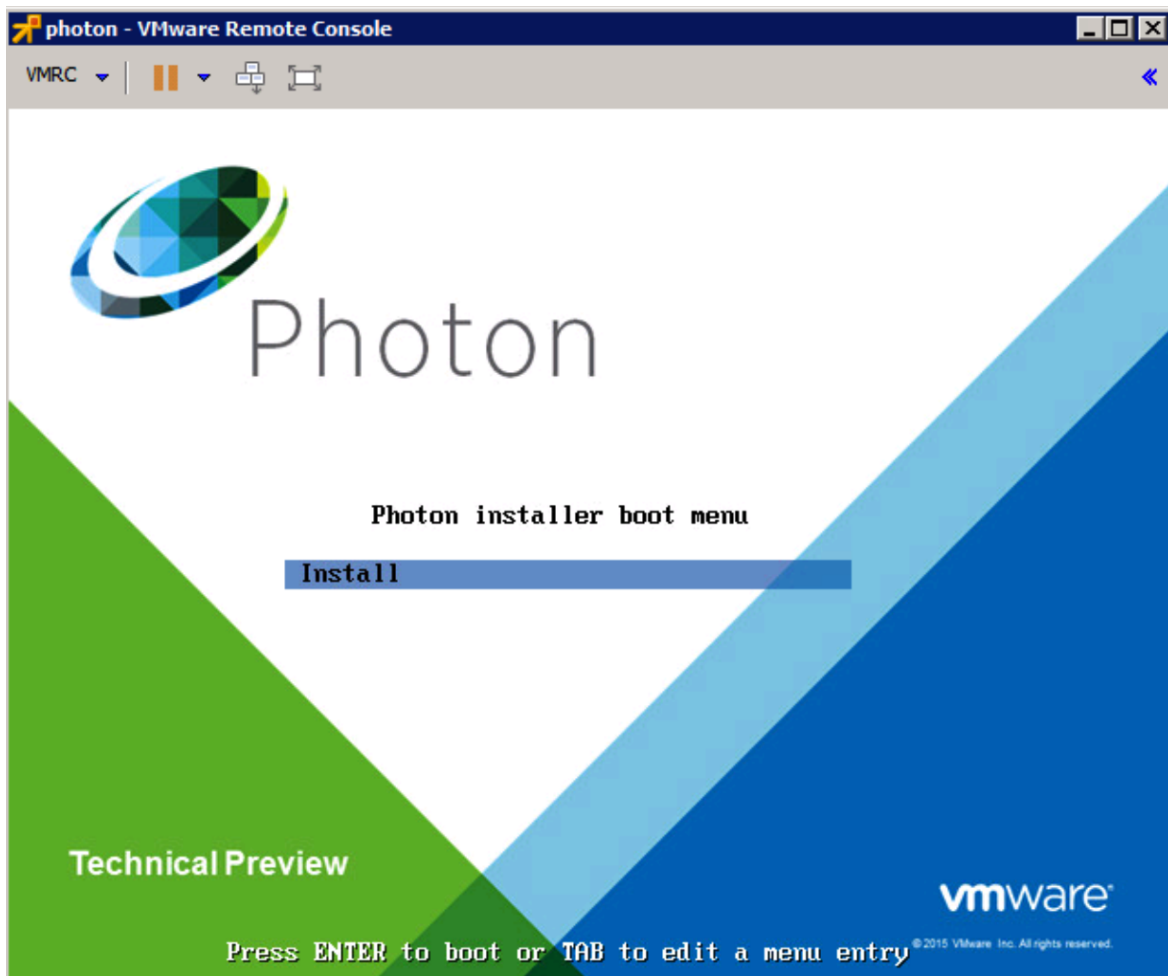
Provisioning type:	Create a new virtual machine
Virtual machine name:	photon
Folder:	Datacenter
Host:	esx-06.jaas.local
Datastore:	vsanDatastore
Guest OS name:	Other 3.x Linux (64-bit)
CPUs:	2
Memory:	384 MB
NICs:	1
NIC 1 network:	VM Network
NIC 1 type:	VMXNET 3
SCSI controller 1:	VMware Paravirtual
Create hard disk 1:	New virtual disk
Capacity:	20.00 GB
Datastore:	vsanDatastore
Virtual device node:	SCSI(0:0)
Mode:	Dependent



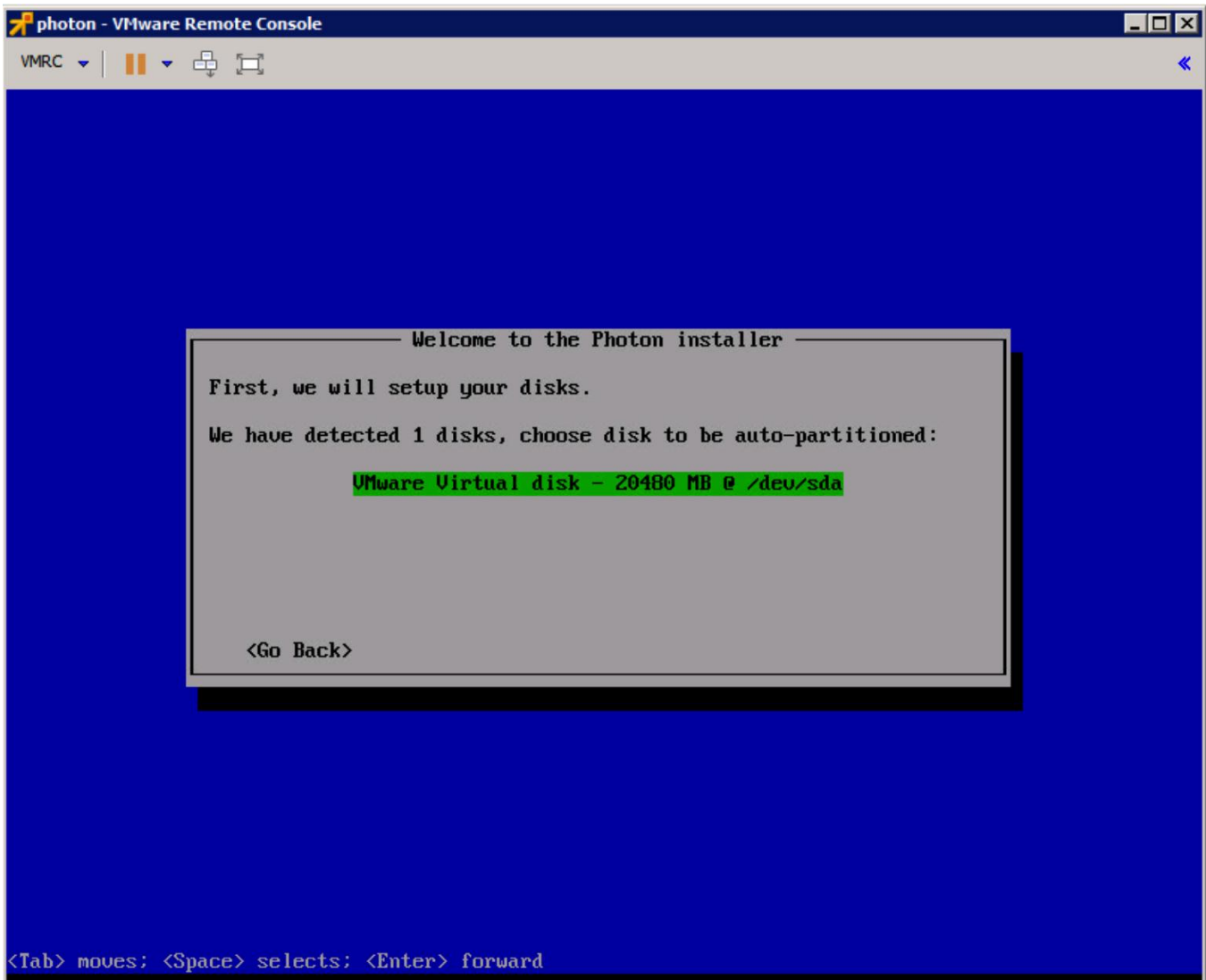
Power on the Virtual Machine and, within a few seconds, the Project Photon Installer Boot Menu will appear. Download and install the Remote Console if you do not have it already, otherwise click Launch Remote Console to interact with the installer.



Select Install to proceed



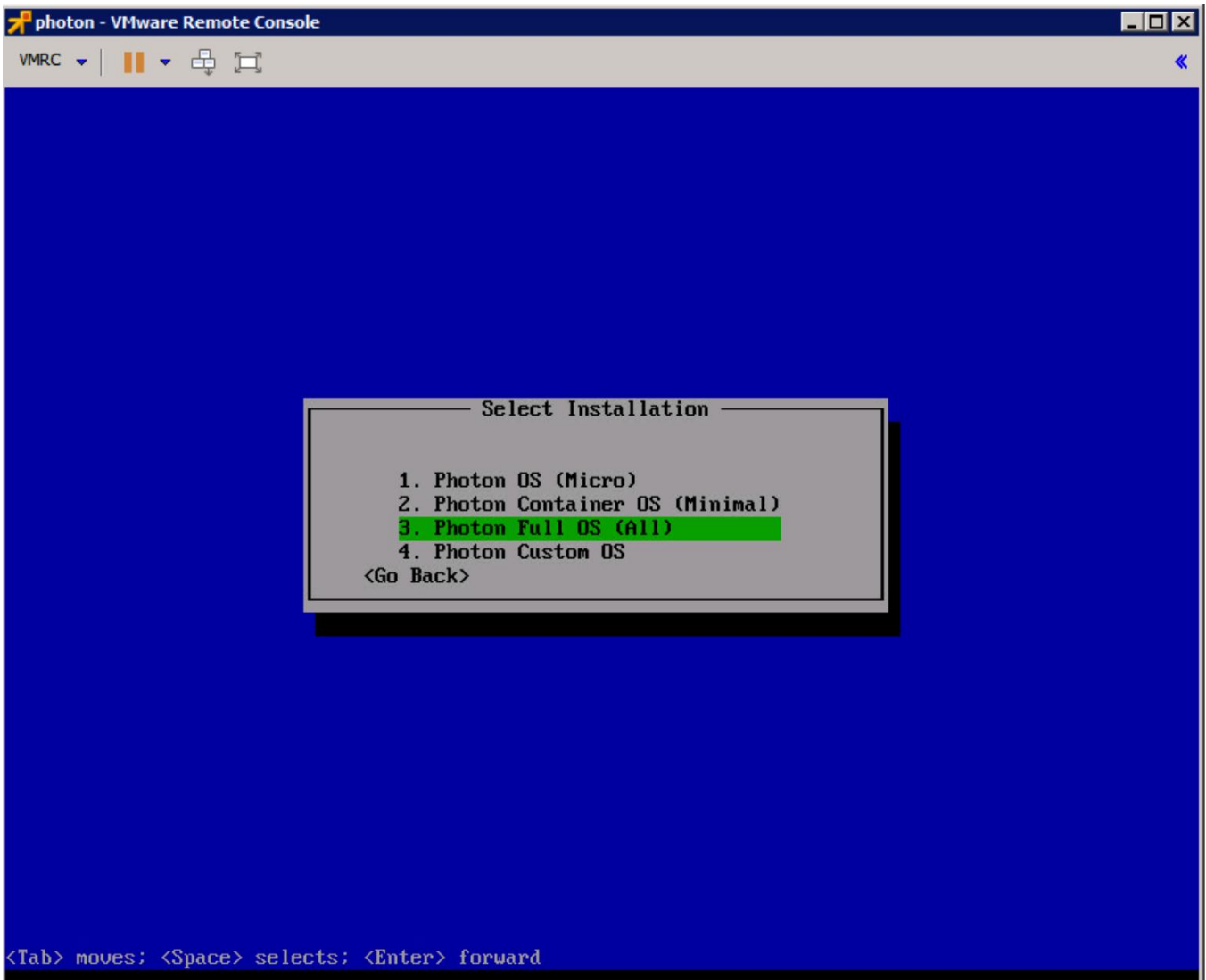
Select – “Install” to proceed.



The Installer will detect 1 x Disk which the 20GB disk configured as part of the virtual hardware setup, select the disk and press enter. You will be prompted to confirm it is okay to erase the entire disk; select “yes” to accept.



You will now be presented with 4 installation options:-



Each install option provides a different runtime environment, depending on your requirements:

- **Photon OS (Micro):** Photon Micro is a completely stripped down version of Photon that can serve as an application container, but doesn't have sufficient packages for hosting containers. This version is only suited for running an application as a container. Due to the extremely limited set of packages installed, this might be considered the most secure version.
- **Photon Container OS (Minimum):** Photon Minimum is a very lightweight version of the container host runtime that is best suited for container management and hosting. There is sufficient packaging and functionality to allow most

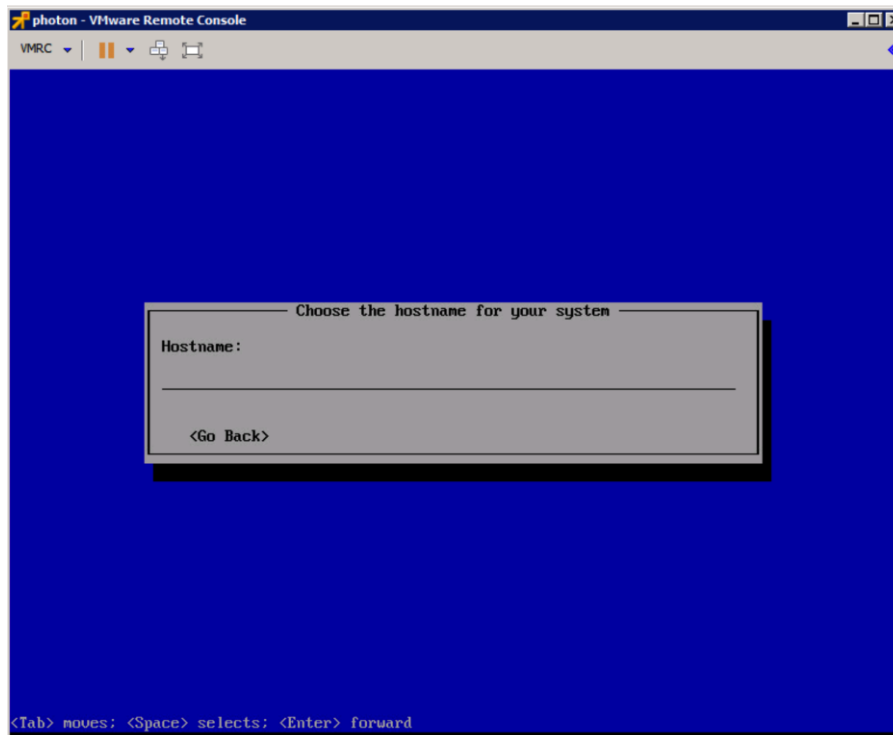


common operations around modifying existing containers, as well as being a highly performant and full-featured runtime.

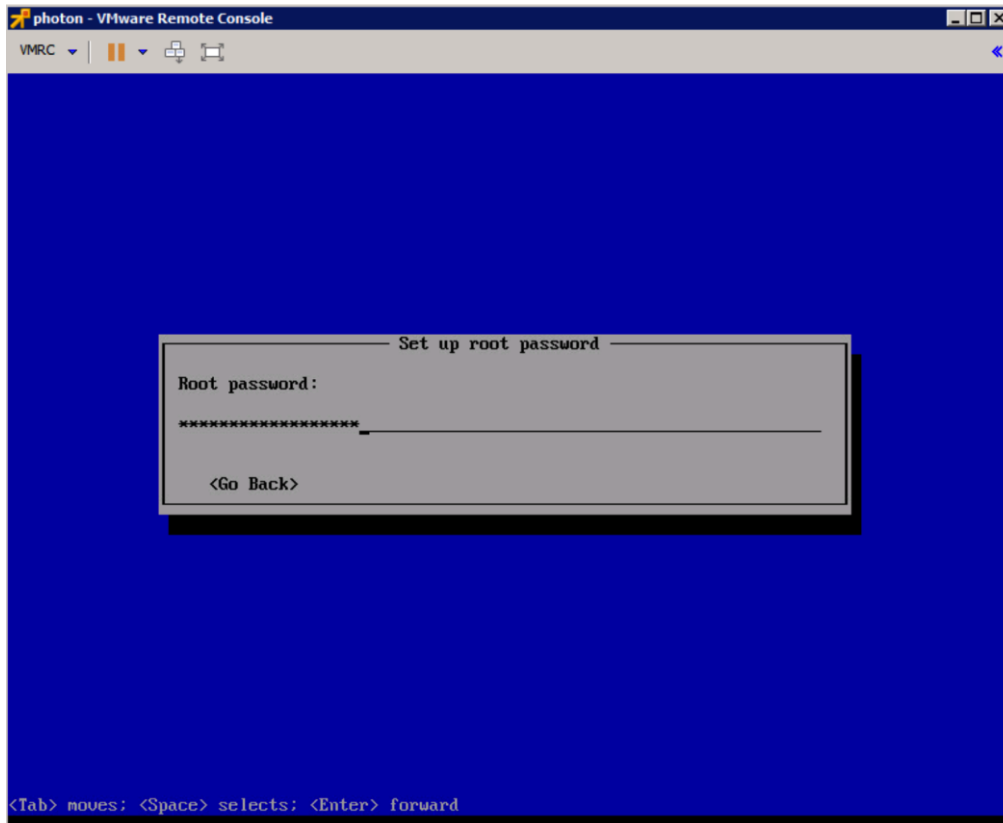
- **Photon Full OS (All):** Photon Full includes several additional packages to enhance the authoring and packaging of containerized applications and/or system customization. For simply running containers, Photon Full will be overkill. Use Photon Full for developing and packaging the application that will be run as a container, as well as authoring the container, itself. For testing and validation purposes, Photon Full will include all components necessary to run containers.
- **Photon Custom OS:** Photon Custom provides complete flexibility and control for how you want to create a specific container runtime environment. Use Photon Custom to create a specific environment that might add incremental & required functionality between the Micro and Minimum footprints or if there is specific framework that you would like installed.

For the purposes of this how-to guide, select “Option 3 – Photon Full OS (All)” this will install the most complete set of Photon packages. Once highlighted, press the Return key on your keyboard.

You will now be prompted for a hostname, as shown in the screenshot below: -



Enter a hostname of your choosing and press “Enter”. You will now be prompted for a root password: -

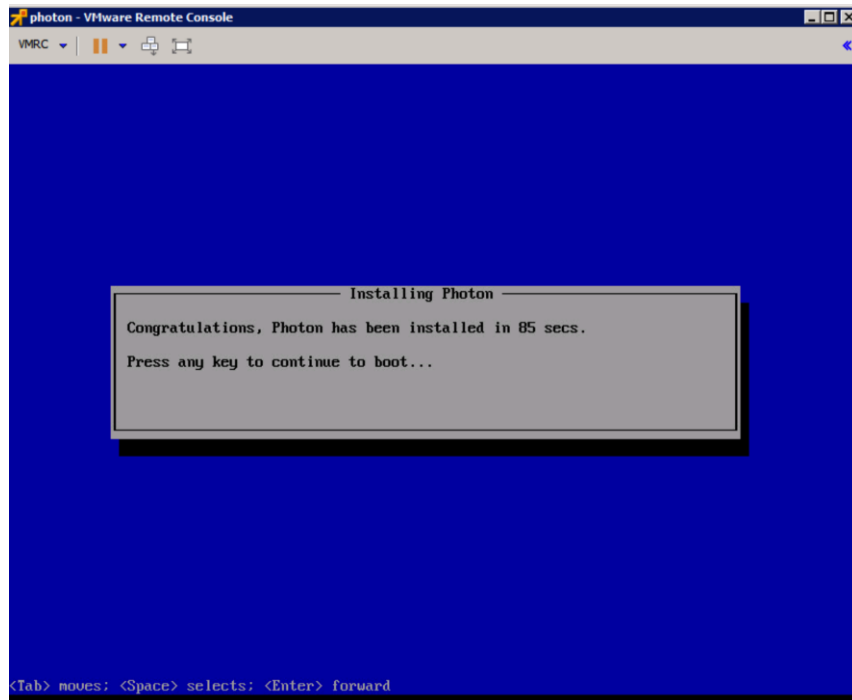
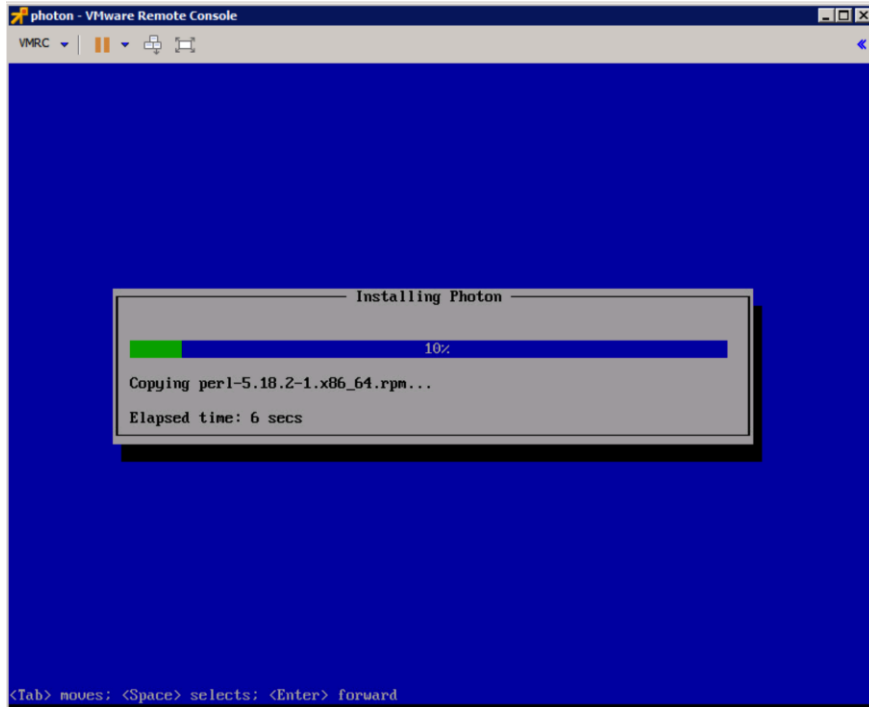


Select a password of your choosing. Photon will not prompt to verify the password, so take care entering it. If you have trouble with unintentional repeated characters in the Remote Console follow KB 196 (<http://kb.vmware.com/kb/196>) for a setting to apply to the Virtual Machine. *note: Photon will not permit commonly used dictionary words to be set as a root password.*

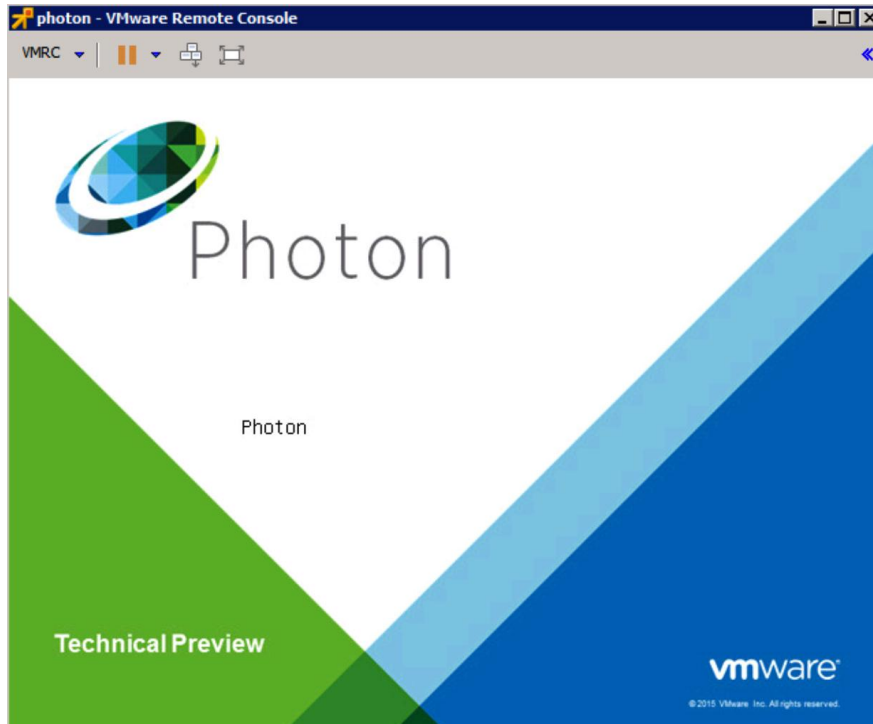
The installation process will now begin.

Typically, a “Full” install will take approximately 30-90 seconds to complete; a minimal will take only 9-12 seconds and a micro as little as 7 seconds. Once the install is complete, you will get a confirmation prompt on the screen stating “*Congratulations, Photon has been installed in xx secs, Press any key to continue to boot...*” - Press any key and the installation will proceed and complete, as shown in the following screenshots.





As the initial boot process begins, you will see this final loading screen before you are taken to a root login command prompt:



At the command prompt enter "root" as your user and you will be prompted for your root password.

You have now successfully setup Project Photon and are ready to use your container runtime environment.



Installing a Containerized Application to Help Demonstrate Capability

Now that you have your container runtime environment up and running, you may be wondering, “what can I do now?” A command prompt is not the most exciting! To help to demonstrate the ease in which you can deploy a containerized application, we will showcase how you can quickly get a Web Server up and running.

For this example, we will use the popular open source Web Server Nginx. The Nginx application has a customized VMware package and published as a dockerfile and can be downloaded, directly, through the Docker module from the Docker Hub.

To run Docker from the command prompt, enter the command below to initialize the docker engine: -

```
systemctl start docker
```

To ensure docker daemon service runs on every subsequent VM reboot, enter :

```
systemctl enable docker
```

Now the docker daemon service is running, it is a simple task to “pull” and start the Nginx Web Server container from Docker Hub. To do this, type the following command: -

```
docker run -d -p 80:80 vmwarecna/nginx
```



This will then pull the Nginx Web Server files and appropriate dependent containers to ensure this containerized application can run. You will see a screenshot similar to below, as the container and dependencies are downloaded and the container is prepared to run:

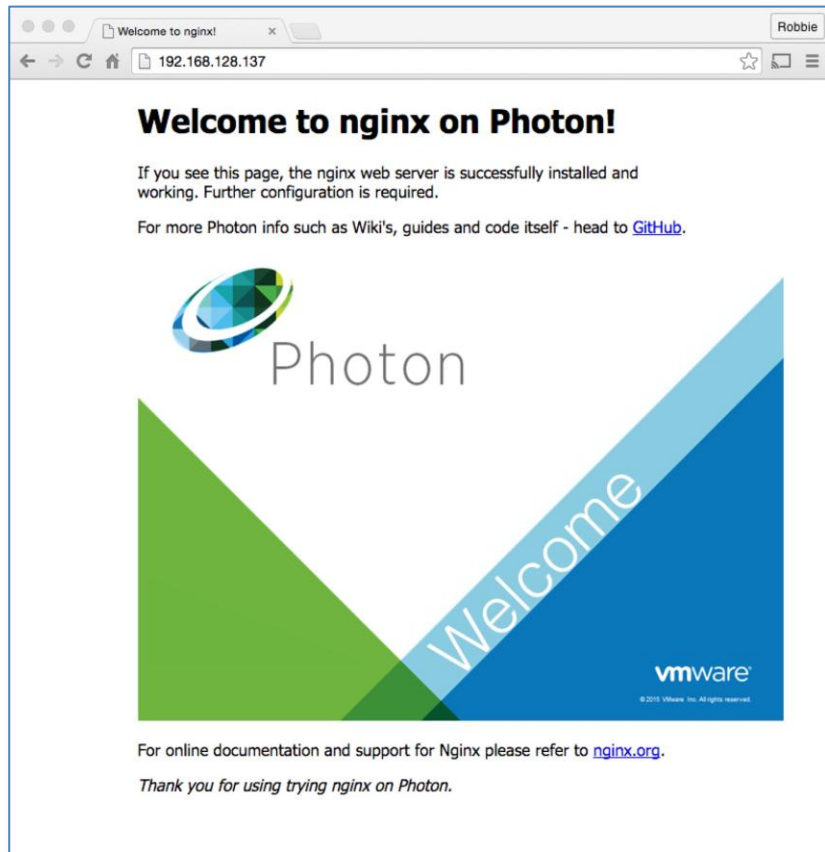
```
photon login: root
Password:
Last login: Tue Apr 14 11:51:14 +0000 2015 on /dev/tty1.
No mail.
root [ ~ ]# systemctl start docker
root [ ~ ]# systemctl enable docker
root [ ~ ]# docker run -d -p 80:80 umwarecna/nginx
Unable to find image 'umwarecna/nginx:latest' locally
Pulling repository umwarecna/nginx
6d36a12c92d5: Pulling image (latest) from umwarecna/nginx, endpoint: https://registry-1.docker.io/v
6d36a12c92d5: Download complete
511136ea3c5a: Download complete
e977d53b9210: Download complete
c9fa20ecce88: Download complete
184d60f5cc4f: Download complete
96d31e36bd8a: Download complete
fd2224b9a216: Download complete
7f3a2bd1e995: Download complete
4c4851e85e94: Download complete
560d74bc7ae1: Download complete
07f0aa77ae76: Download complete
126105e511b8: Download complete
775024331ffa: Download complete
db26ae25be15: Download complete
Status: Downloaded newer image for umwarecna/nginx:latest
2adef970435b9e5e6e63f7d76debc72984062ae8d1161b9f9209c5666cb8f28f
root [ ~ ]#
```

Once “docker run” process is completed, you will be returned to the command prompt. You now have a fully active website up and running through typing just a single command within Project Photon using containers.

To test that your Web Server is active, we need to get the IP address of the Project Photon Virtual Machine. To get the IP address, enter the following command `ifconfig`. This will now display a list of adapters connected to the virtual machine. Typically, the web server daemon will be bound on “eth0.”



Start a browser on your host machine and enter the IP address of your Project Photon Virtual Machine. The following screen will appear and that will show that your web server is active: -



You can now run any other containerized application from Docker Hub or your own containerized application within Project Discus.

We hope you enjoy using Photon as much as did creating it!

