Using Project Photon on Google Compute Engine

What is 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 outlines how to get started running Project Photon on Google Compute Engine and will provide an overview of the following:

- Prerequisites for running Project Photon on GCE.
- Creating a GCE image from Project Photon OS installed on VMware Workstation or Fusion
- Adding Project Photon image to GCE
- Launching a Project Photon instance on GCE

Prerequisites

The following list is required for preparing and launching Photon on Google Compute Engine. Note that Google Compute Engine provides a public Project Photon image.   If you choose to create custom private image then please follow the optional instructions for creating and loading an image to Google Compute Engine.

Required:

- An account on Google Compute Engine
- Google Cloud SDK
- VMware Fusion
- Linux or Mac OS X*.   *If using Mac OS X GNU Tar must be installed.
- Default of 15GB free space for Photon VM and resulting GCE image. Size may increase or decrease based on your needs.
- QEMU   *See Mac OS X notes below.
- Compile Photon from source. Note: If compiling Photon from source, please utilize the link provided in the section "Building the ISO". Once an ISO is produced you may use this document to customize the installed ISO for GCE.

Mac OS X notes:

> To run GNU tar (gtar) and QEMU on OS X you must install a package manager such as Home Brew. Installing Home Brew on OS X is beyond the scope of this document.

Links:

- Google Cloud SDK: https://cloud.google.com/sdk/#Quick_Start
- Home Brew for OSX : http://brew.sh/
- QEMU : http://wiki.qemu.org/

Create a custom private Photon image on GCE

The following section provides instructions for creating a custom private image on GCE.

Downloading Project Photon

To download the Project Photon ISO either download the latest ISO image from here or clone the GitHub Photon repository and build the ISO.

Preparing Photon for Google Compute Engine (GCE)

The steps for preparing Photon for Google Compute Engine assume that you have already installed Photon into a VM running on VMware Fusion or Workstation. For instructions for installing Photon in VMware Fusion or Workstation, please see the Using Project Photon on Fusion/Workstation Getting Started guide available here.

Use SSH to prepare the image. There are several commands that must be run to prepare the image for GCE. Some users may find it easier to use SSH and copy/paste the commands into the Photon OS.

1. Add an unprivileged user as remote login as remote is disabled.
   a. Root [~]# useradd –mU photon
   b. Root [~]# password photon
2. From your workstation ssh into the VM. You can find the IP of Photon by using the ifconfig –a command.

Prepare Photon Disk

To run on Google Compute engine Photon must change it's partition table from GPT to MBR. The following commands will install dependencies from the Photon ISO.

```
mount /dev/cdrom /media/cdrom
tdnf install yum
tdnf install python2-libs
yum install ntp sudo wget tar which gptfdisk sed findutils grep gzip --nogpgcheck -y
```

The following commands will convert the partition table from GPT to MBR and update GRUB to be compatible with GCE.

```
sgdisk -m 1:2 /dev/sda
grub-install /dev/sda
# Enable serial console on grub for GCE.
cat << EOF >> /etc/default/grub
GRUB_CMDLINE_LINUX="console=ttyS0,38400n8"
GRUB_TERMINAL=serial
GRUB_SERIAL_COMMAND="serial --speed=38400 --unit=0 --word=8 --parity=no --stop=1"
EOF
```

vmware®

```
# Create new grub.cfg based on the settings in /etc/default/grub
grub-mkconfig -o /boot/grub/grub.cfg
```

Install Google Cloud SDK and GCE packages required for Google Compute.

```
yum install google-daemon google-startup-scripts --nogpgcheck -y

cp /usr/lib/systemd/system/google* /lib/systemd/system/
cd /lib/systemd/system/multi-user.target.wants/

# Create links in multi-user.target to auto-start these scripts and services.
for i in ../google*; do  ln -s $i `basename $i`; done

cd /tmp/; wget https://dl.google.com/dl/cloudsdk/release/google-cloud-sdk.tar.gz --no-check-
certificate
tar -xf google-cloud-sdk.tar.gz
cd google-cloud-sdk
./install.sh
```

Update the /etc/hosts file for the GCE metadata host

```
echo "169.254.169.254 metadata.google.internal metadata" >> /etc/hosts
```

Configure the NTP service to start at boot and use Google's NTP server.

```
sed -i -e "/server/d" /etc/ntp.conf
cat /etc/ntp.conf
echo "server 169.254.169.254" >> /etc/ntp.conf
# Create ntpd.service to auto starting ntp server.
cat << EOF >> /lib/systemd/system/ntpd.service
[Unit]
Description=Network Time Service
After=network.target nss-lookup.target

[Service]
Type=forking
PrivateTmp=true
ExecStart=/usr/sbin/ntpd -g -u ntp:ntp
Restart=always

[Install]
WantedBy=multi-user.target
```

```
  EOF    #

 Add link in multi-user.target.wants to auto start this service.
 cd /lib/systemd/system/multi-user.target.wants/
 ln -s ../ntpd.service ntpd.service
```

Set the time zone to UTC

```
  ln -sf /usr/share/zoneinfo/UTC /etc/localtime
```

Update the DNS resolver to use Google DNS

```
  echo "nameserver 8.8.8.8" >> /etc/resolv.conf
```

Remove ssh host keys and add script to generate new keys at boot

```
 rm /etc/ssh/ssh_host_*

 # Depending on the installation, you may need to purge the following keys
 rm /etc/ssh/ssh_host_rsa_key*
 rm /etc/ssh/ssh_host_dsa_key*
 rm /etc/ssh/ssh_host_ecdsa_key*
 sed -i -e "/exit 0/d" /etc/rc.local
 echo "[ -f /etc/ssh/ssh_host_key ] && echo 'Keys found.' || ssh-keygen -A" >> /etc/rc.local
 echo "exit 0" >> /etc/rc.local    printf "GOOGLE\n" > /etc/ssh/sshd_not_to_be_run
```

Update SSHD configuration per Google Compute Engine recommendations

Note that SSHD configuration located at /etc/ssh/sshd_config should be updated per the recommendations for Google Compute Engine. The recommendations for SSHD configurations on Google Compute Engine can be found at the following link: https://cloud.google.com/compute/docs/tutorials/building-images#ssh

Configure the MTU on the network interface to 1460

```
  # Create a startup service in systemd that will change MTU and exits
  cat << EOF >> /lib/systemd/system/eth0.service
  [Unit]
  Description=Network interface initialization
  After=local-fs.target network-online.target network.target
  Wants=local-fs.target network-online.target network.target

  [Service]
  ExecStart=/bin/ifconfig eth0 mtu 1460 up
  Type=oneshot

  [Install]
  WantedBy=multi-user.target
  EOF
```
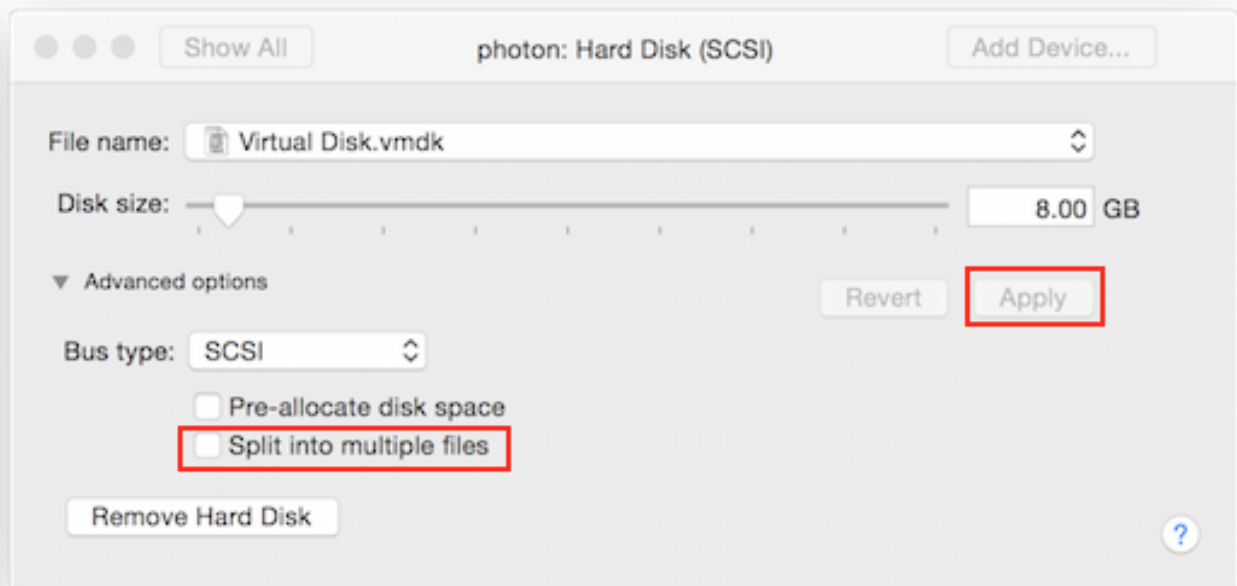
vmware®

```
# Make this service auto-start at boot.
cd /lib/systemd/system/multi-user.target.wants/
ln -s ../eth0.service eth0.service
```

Create GCE image
At this point the Photon OS is compatible with GCE. However, the VM must now be converted to a raw disk and then packaged for GCE.

From your workstation…

1. Shutdown the Photon VM.
2. Ensure that the Photon VM's is consolidated to one VMDK. By default, VMware Fusion splits a VMDK into multiple disks. To ensure compatibility with GCE the VMDK must be consolidated into a single disk.
    a. Go to the Photon VM settings on VMware Fusion and select Hard Disk (SCSI).
        i. Ensure the "Split into multiple files" option is unchecked.
        ii. If "Split into multiple files" was previously checked, then hit the apply button to consolidate into a single VMDK.

3. Change directory to the Photon Virtual Machine directory. Assuming the Virtual Machine is named Photon, the directory would be at the following location on Mac OS X.
    a. ~/Documents/Virtual Machines.localized/Photon.vmwarevm
4. Convert the VMDK to a raw image. Note: The raw image must be named disk.raw to be compatible with GCE.
    a. qemu-img convert –f vmdk –o raw Virtual\ Disk.vmdk disk.raw
5. Create a compressed tar file for GZ.
    a. On Mac OS X: gtar –Szcf photon.tar.gz disk.raw

b. On Linux: tar –Szcf photon.tar.gz disk.raw

Adding Photon to the your Google Compute Engine (GCE) images

At this point you now have a compressed raw image that can be uploaded to GCE. The following steps will create an image in GCE.

Prerequisite:

Prior to running the commands to add the image, ensure that you have the following.

1. The gsutil and gcloud tools installed.
2. An existing GCE storage bucket. In this example a storage bucket called "photon-bucket" has already been created.

a. Upload the file to a GCE storage bucket

gsutil cp photon.tar.gz gs://photon-bucket

b. Add the image to GCE

gcloud compute images create photon –source-uri gs://photon-bucket/photon.tar.gz

Launching a private Photon instance on Google Compute Engine (GCE)

You may launch the instance from GCE's web dashboard or SDK.

SDK:

gcloud compute instances create photon --image https://www.googleapis.com/compute/v1/projects/your-gce-project/global/images/photon

Using Photon to Run Containers

Once a Photon instance is provisioned on GCE a containerized application can be deployed.

As an example, NGNIX will be deployed into a Docker container. The Nginx application is packaged and published as a Dockerfile and will be downloaded from the Docker Hub.

From the command prompt on the Photon instance running in GCE.

Enable docker to run at boot

[~]$ sudo systemctl enable docker
Created symlink from /etc/systemd/system/multi-user.target.wants/docker.service to /lib/systemd/system/docker.service.

Start docker

[~]$ sudo systemctl start docker

Pull NGINX

Once docker is running, NGNIX and it's dependencies can now be installed. Run the following command to "pull" NGNIX and it's dependencies. Once the command completes, NGNIX will be installed and running.

[~]$ sudo *docker run -d -p 80:80 vmwarecna/nginx*
Unable to find image vmwarecna/nginx' locally
Pulling repository vmwarecna/nginx
9f4cd86c1aec: Download complete
511136ea3c5a: Download complete
f3c84ac3a053: Download complete
b7e01914d76cc010d52216ad7d7ccf0754f109f3a155a2d8bc4a9ab3323adfe0

Enable HTTP access to your Photon instance running in Google Compute Engine

In order to access the NGNIX instance from the internet, the Photon instance must be enabled for HTTP traffic in Google Compute Engine.

From the Google Developers Console, navigate to Compute, VM instances and select the Photon instance. Scroll down to the network section and enable "Allow HTTP traffic".

Network ─────────────────────────

default

☑ Allow HTTP traffic   ☐ Allow HTTPS traffic

vmware®

Verify that NGNIX is available from the Internet

Using your browser, enter the External IP of your Photon instance running in Google Compute engine. The external IP can be found on the "VM instances" page in the Google Developers Console. Your browser will display the following message.

Welcome to nginx on Photon!

If you see this page, the nginx web server is successfully installed and working. Further configuration is required.

For more Photon info such as Wiki's, guides and code itself - head to GitHub.

For online documentation and support for Nginx please refer to nginx.org.

Thank you for using trying nginx on Photon.