MacStadium

User Guide

How to Leverage MacStadium's VMware Private Cloud for iOS and macOS Continuous Integration Projects

macstadium.com
hello@macstadium.com
+1.877.250.3497

Contents

In this guide...

How to Leverage MacStadium's VMware Private Cloud for iOS and macOS Continuous Integration Projects

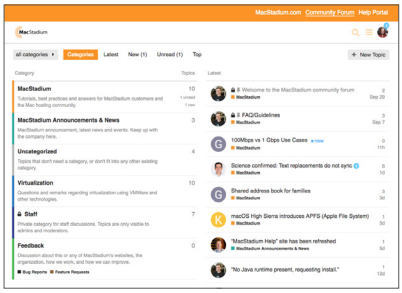
In this guide, we'll describe setting up a MacStadium VMware private cloud to run builds for macOS or iOS continuous integration projects on dedicated Mac hardware.

Continuous integration (CI) is a modern development term for ensuring that your code passes a series of tests (integration tests, historically) on every code check-in. This means having a central point where your code can be checked and have the results reported back to you. This allows teams to iterate rapidly and collaborate on their source code management (SCM) platform of choice (e.g. Github, Gitlab or Bitbucket).

There are special considerations that can make iOS and macOS CI projects harder to implement than in the Linux world. For one, Apple requires that iOS and macOS apps be developed on authentic Apple hardware, which means you can't simply virtualize an iOS or macOS environment to run on non-Apple hardware. In Linux, there are tools like Docker that can provide isolated runtime environments for repeatable tests in those environments.

While in the Mac world there aren't native containerization tools, there are virtualization tools like VMware (see What's the difference between containers and virtual machines) that can be used for similar purposes. This means that on top of MacStadium's physical Mac hosts, we can create several virtual macOS machines and use them as targets for running our tests.

Have questions? Visit the MacStadium Community Forum.



Choosing a Plan

Virtualization allows you to run more than one virtual host per physical machine. Since you are dividing the resources of a single host into several virtual ones, you will need to choose the number of physical machines you need based on how many concurrent builds you want to run. The more concurrent builds you do, the less time developers are waiting for builds. Beyond concurrency, you will need to run a virtual machine for each version of macOS that you are targeting.

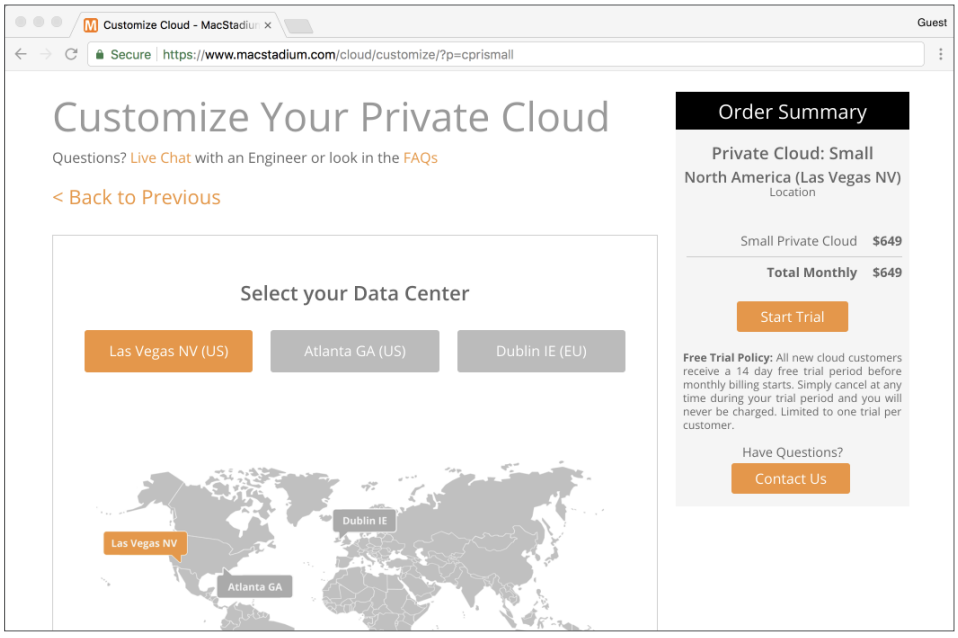
Tip!

A good rule of thumb is 2-3 virtual machines per physical host.

To get started:

1. Pick the VMware cloud plan that suits you: https://www.macstadium.com/cloud/
2. Choose the data center closest to you
3. Choose “Start Trial” and create an account

MacStadium will provision your new environment and email you at the account you nominated. Typically this takes 2-3 working days.



Connecting to Your Cloud

The provisioning team at MacStadium will create a ticket you can access via the MacStadium portal with your IP Plan document (by default in Excel format, but Google Docs can import it if necessary).

To access your environment, you will first need to connect via a VPN:

- On Windows, see Configuration Cisco IPSEC VPN in Windows using Shrew VPN Client
- On macOS, see How to setup a MacStadium Cisco IPSEC VPN connection

Connecting to the vSphere Web Client

Once you're connected to the VPN, you can connect to your vCenter instance. We will use the web interface for this example, although there are native options for Windows and macOS.

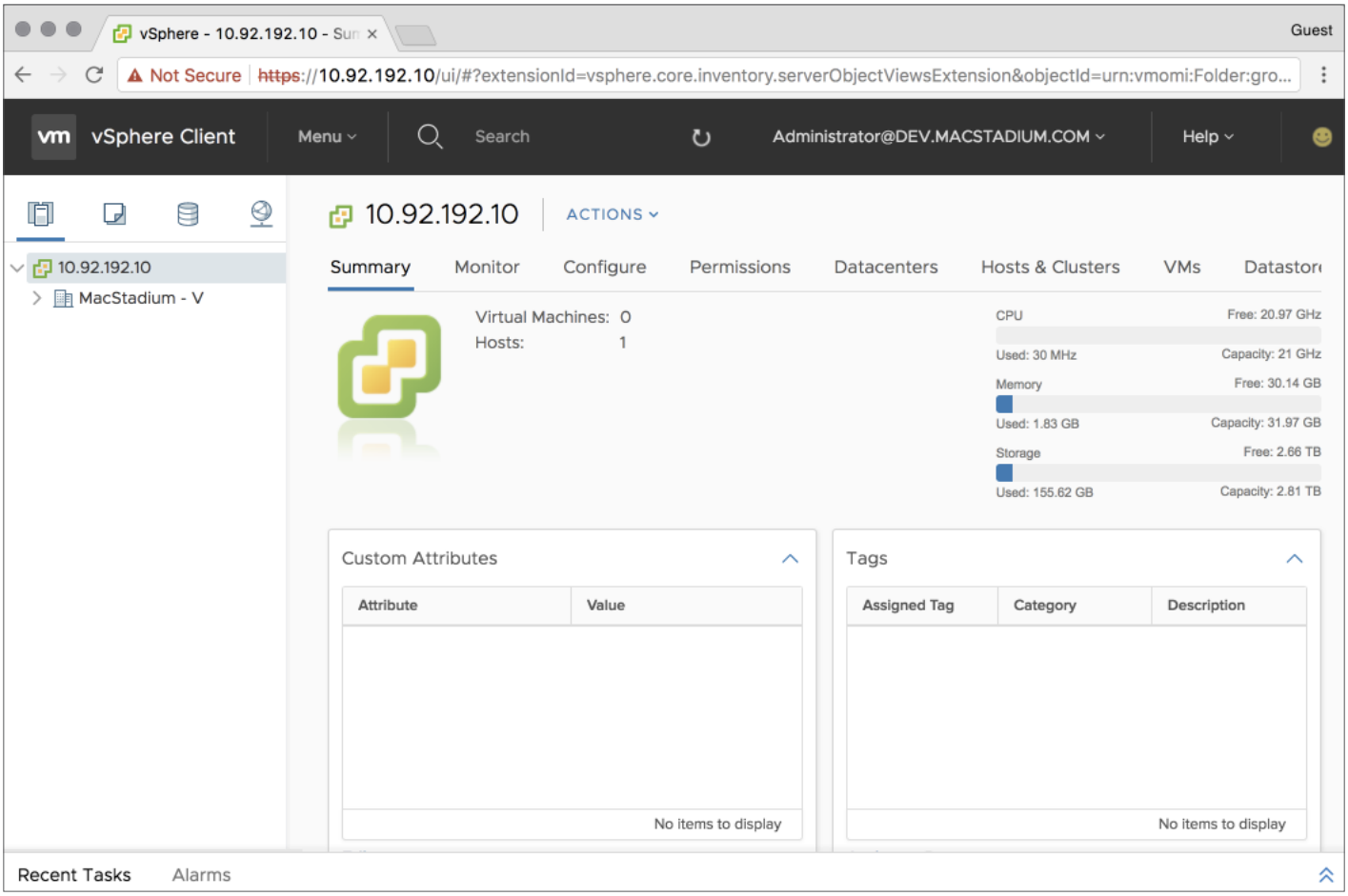
1. Click the "vCenter Web Client" URL in "Step 2: vCenter Login" in your IP Plan.

2. Copy and paste the username and password, being careful not to include white space at the end of the fields.

If you have vSphere 6.5+ then you can use the HTML5 version by changing your login url to `/ui`, for example https://10.92.192.10/ui.

Otherwise, if you are using the Flash version and you receive a white screen, you might need to whitelist the site in your browser. In Chrome, this is under Content Settings > Flash. You will need to add an exception to allow for the IP address of your vCenter host (e.g 10.92.192.10). Then, refresh the original page and it should load.

You should now be able to see your private cloud!



Creating a Virtual Machine

Your plan selection will determine the number of physical hosts represented as part of your cluster in your data center in vSphere. This ultimately lets you manage multiple networks or groups of hosts within the same interface as you grow. For now, we will be dealing with the cluster that represents your private cloud, for this example named “MacPro_ Cluster.”

To create a single virtual machine running macOS:

1. In the Navigator, right-click on the Cluster and select “New Virtual Machine.”

2. Select “Create a new virtual machine” and then select Next.

3. Choose a name, e.g “macOS-10.12” and then select Next.

4. In “Select a compute resource,” click Next. If prompted, select a specific host to deploy to if your cloud doesn’t have [DRS Mode][] turned on. This just means that your image will run on that specific host vs. VMware determining a host for you.

5. Select the datastore for your VM’s files (generally you will only have one of these). Select Next.

6. Select the default presented for Compatibility and then Next.

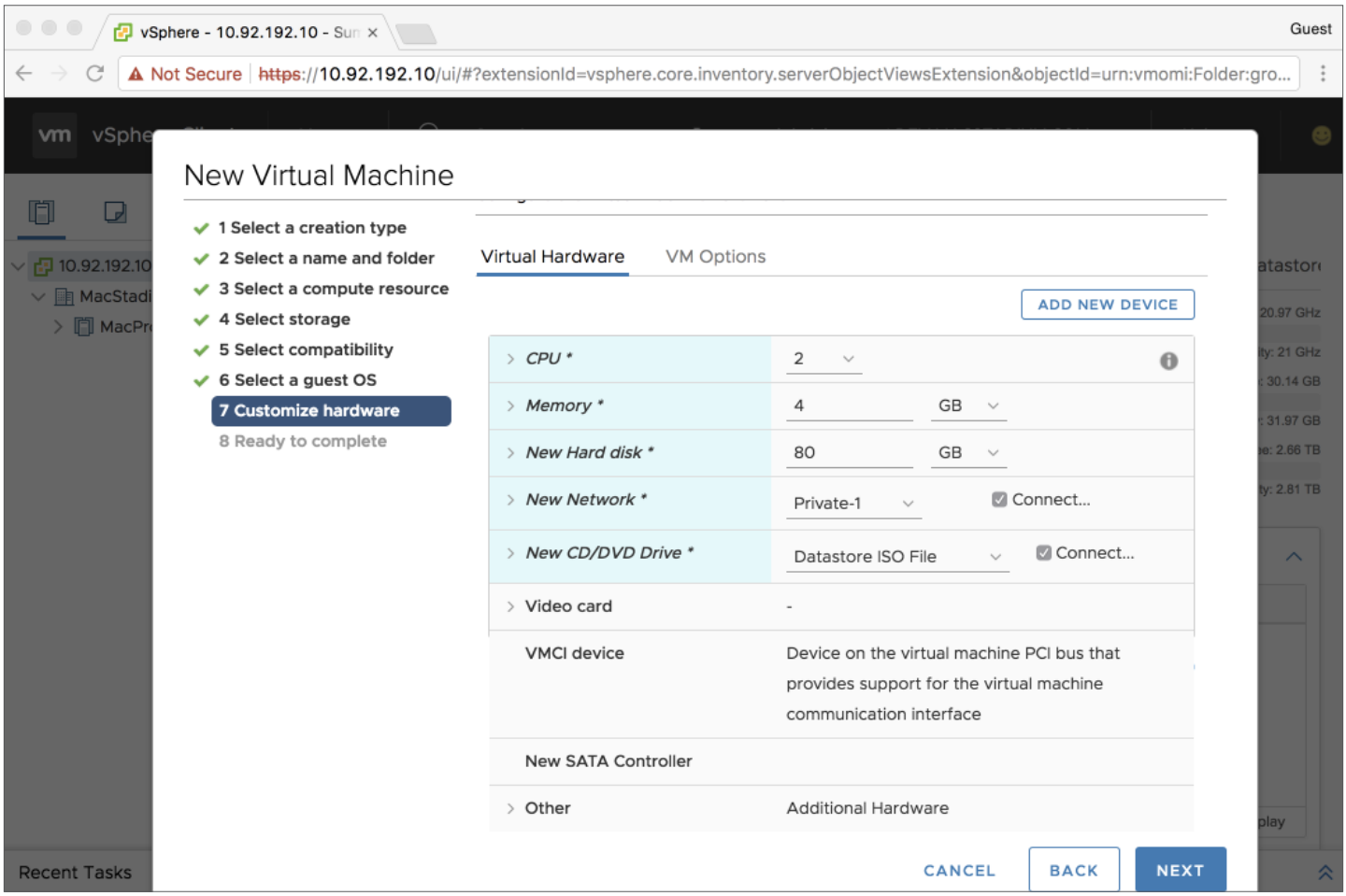
7. Select Guest OS Family `Other` and Guest OS Version `Apple Mac OS X 10.12 (64-bit)`. Don’t worry that this won’t match the actual macOS version; we only need one that is close.

8. Select the follwing virtual hardware settings before selecting Next:
   - CPU: `2`
   - Memory: `4GB`
   - New Hard Disk: `80GB`
   - New Network: `Browse > Private-1` (The name of your private range in your IP Plan)

Since the VM settings shown here are only an example, your resource allocation strategy will ultimately depend on your own specific needs. Keep in mind that resources are finite, and over-allocating resources can cause unexpected results and bad behavior. Our recommendation would be to test, gather feedback, and then iterate. For example, start with 2 CPU, 4GB RAM, 80GB hard disk, etc., and then increase/decrease as needed.

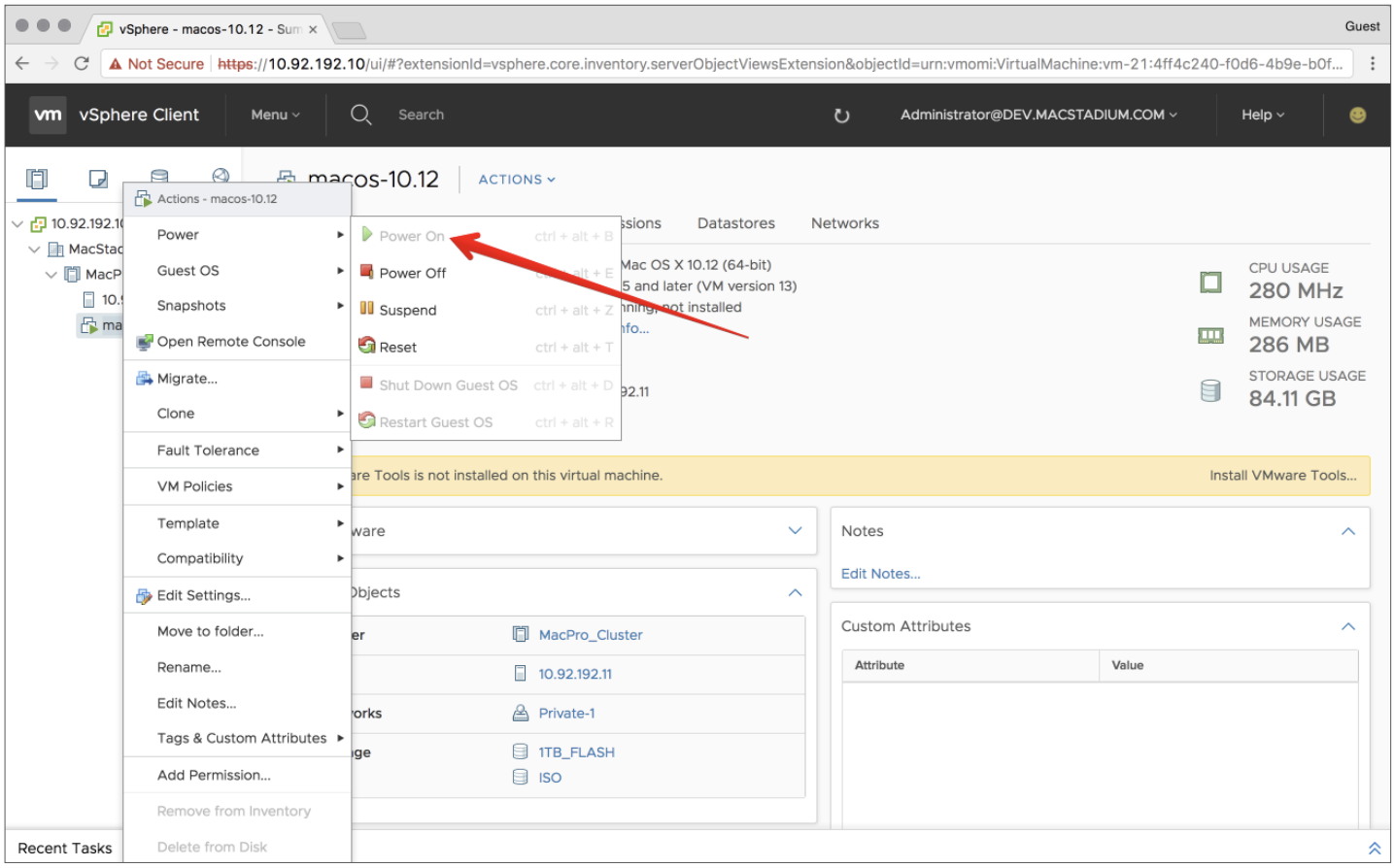
- Expand `CPU` and select “Expose hardware assisted virtualization to the guest OS” (in case you want a nested VM)
- New CD/DVD Drive: `Datastore ISO File`. Select `ISO > OSX > macOS_Sierra.iso`. Ensure that `Connect` is ticked.

9. Check to see that your summary looks correct, and then Finish.



Installing macOS

After roughly 30 minutes, you should now have a virtual machine listed within your cluster in the web interface.



To install macOS:

1. Right-click on the virtual machine and select `Power > Power On`.

2. Click on the virtual machine and then click on the preview image. You should now see the macOS installer initializing.

3. From the "Install macOS" screen, if you only see the DVD image, you will need to select `Utilities > Disk Utility` from the menu.
   - Select "VMware Virtual SATA Hard Drive Media"
   - Select "Erase" with defaults
   - Close Disk Utility and select "Untitled"

4. Wait for the installer to run.

5. When at “How Do You Connect,” select `Local network (Ethernet)`.

6. Select TCP/IP Connection Type `Manually` and enter the following details before Next:
   - IP Address: `10.254.50.2` (The first address from your IP Plan under the Private-1 range)
   - Subnet Mask: `255.255.255.0`
   - Router Address: `10.254.50.1`
   - DNS Servers: `8.8.8.8, 8.8.4.4` (Google’s DNS servers)

7. Skip Location Services and Apple ID.

8. Create an Account and note the account details. For this demo, we will use `buildkite` with a password of `buildkite`.

9. Set the time zone to your local time.

You should now be on the desktop. You have a working Mac virtual machine!

Installing VMware Guest Tools

VMware has a set of tools that must be installed on macOS. These get mounted via a virtual CD drive.

ISO Download and Upload to vCenter

1. Download a recent version of VMware Tools (VMware Tools 10.0.1) and unzip it so that you have a `darwin.iso`.

2. Login into vCenter.

3. Use your storage tabs to navigate to your datastore.

4. Browse your datastore. You can upload the ISO to the root folder or create an ISO folder.

Install Directions

1. Shut down your machine from within macOS.
2. From the web interface, right-click on your virtual machine and select `Power > Power Off`.
3. Right-click on your virtual machine and select `Edit Settings`.
4. Change CD/DVD Drive 1 to `Datastore ISO File` and select the ISO you uploaded.
5. Right-click on your virtual machine and select `Power > Power On`.
6. After the reboot, go back into the desktop of the virtual machine and log in.
7. Install the VMware Tools when prompted.

Installing Build Tools

Now that you have a working machine, you will need to install the basic software needed to run builds. You can either do this via VNC or via the Web Console, or if you enable Remote Login in the macOS settings, you can SSH in. This guide explains the latter approach as it's easier to do repeatably and to automate as you scale up.

You can access your host via SSH from your desktop provided you have the VPN connected:

```
ssh buildkite@10.254.50.2
```

All further terminal commands in this guide will assume you are logged into the above.

Install Homebrew

Homebrew is a package manager for macOS and used for managing installation of the various tools needed for iOS and macOS CI efforts. This step will take about 15 minutes, and you will be required to enter your password at least once.

Read more about Homebrew at https://brew.sh/

Lots of tooling requires a modern Ruby installed (fastlane for instance), so we also update our system Ruby:

```
/usr/bin/ruby -e "$(curl -fsSL https://raw.githubusercontent.com/Homebrew/install/master/install)"
brew install rbenv ruby-build

# Add rbenv to bash so that it loads every time you open a terminal
echo 'if which rbenv > /dev/null; then eval "$(rbenv init -)"; fi' >> ~/.bash_profile
source ~/.bash_profile

# Install Ruby
rbenv install 2.4.2
rbenv global 2.4.2
ruby -v
```

Install Xcode

You can manually install Xcode via the App Store if required, but we recommend Xcode::Install, a tool for installing and managing multiple versions of Xcode. This allows you to use one machine for multiple versions and swap between them, similar to the approach we took in the previous step for Ruby:

```
gem install xcode-install
rbenv rehash
xcversion install 9.0.1
sudo xcode-select -s /Applications/Xcode.app
```

Next Steps

Now that you have a base operating system, you can create multiple virtual machines with the system variations you require. You can convert your VM to a template and then use that template to create multiple VMs that can easily be removed once they are no longer required.

1. Stop VM if running.
2. Right-click on VM > Select “Convert to Template.”

From that point you can create new VMs from that template.

FAQ

What’s the difference between containers and virtual machines?

From https://en.wikipedia.org/wiki/Virtual_machine:

> In computing, a virtual machine (VM) is an emulation of a computer system. Virtual machines are based on computer architectures and provide functionality of a physical computer. Their implementations may involve specialized hardware, software, or a combination.

VMWare provides virtual machines, which allows you to run different operating systems as the operating system interacts with virtualized hardware. Virtual Machines can be many gigabytes as they need to have the entire operating system and all the drivers and related software in the image and can take 10-20 seconds to boot.

Containers (otherwise known as Operating System Virtualization) is a more lightweight alternative that creates multiple isolated environment within an existing operating system and kernel. Ecosystems like Docker use copy-on-write filesystems to allow for completely ephemeral, isolated environments to be created to run a single process. The advantage to containers is that they start nearly instantly and the filesystem snapshots are much smaller as they don’t have to contain the entire operating system kernel. The downside is they are limited to the same operating system and kernel as the host environment.