Running PowerShell Core using Docker

More and more of the Microsoft ecosystem is making its way into open source platforms. One of the very interesting products coming from the Microsoft camp lately is the PowerShell Core platform which is now ported to run on multiple underlying operating system environments.

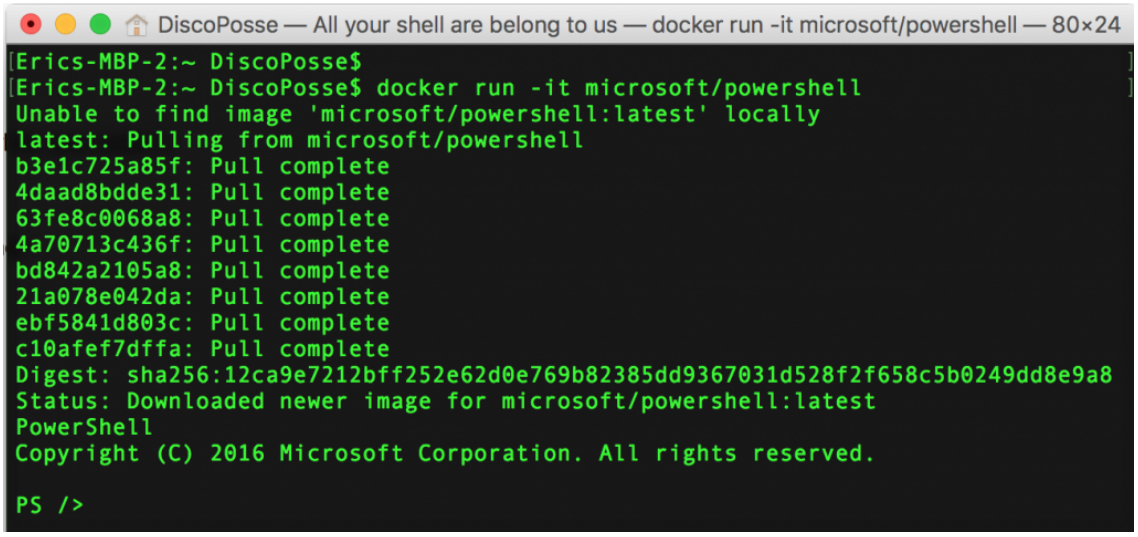
I covered the process to install the Mac OSX version which is very cool, but let's take the abstraction one level higher and look at running the PowerShell core inside a Docker candidate.

The first thing you'll want to do is head on over here to make sure you're running the nifty Docker Toolbox for your laptop or desktop environment if you haven't already got Docker available to use.

Running your first PowerShell Core container

The commands here may seem a little too easy, but that's by design. The containerized implementation makes the deployment and use of PowerShell core super easy!

Let's launch our first container with the `docker run -it microsoft/powershell` that will kick up a new container based on the image which is in the Docker public hub under the Microsoft organization. The `-it` means that we are launching in an interactive mode inside the container.



That gets you up and running to be able to run the PowerShell environment.

NOTE: There is still limited functionality compared to the full PowerShell on Microsoft environments. This is something that is changing with each release as the community and Microsoft themselves contribute towards more features.

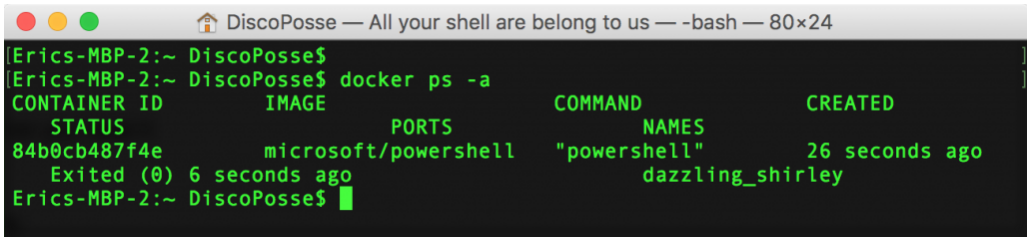
Exiting and re-entering the container

Getting out of the container is as easy as typing `exit` and the prompt. This will bring you back out to the local environment. That gives is an interesting situation where we have the container present, but it is stopped. If you run the same command as you did before, that actually launches an entirely new container.

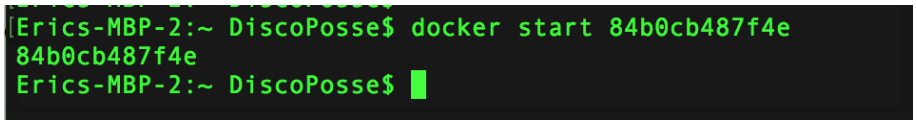
We need to do three things in order to get back in to the same container:

1. find the ID of the existing container
2. start the container using that ID
3. attach to the container

First, let's check the containers to find out the ID of the one we want using the `docker ps -a` command:



Use the `docker start [CONTAINER-ID]` command where `[CONTAINER-ID]` is the ID you see in your console:



Use the same ID and attach to the now active container with the `docker attach [CONTAINER-ID]` command:

That is all there is to it! Each time you exit, the container will automatically stop because we don't need to keep it running in the background. There are other ways to keep it running, but that is for another blog post

Removing the container is as simple as running the `docker rm [CONTAINER-ID]` where `[CONTAINER-ID]` is the ID we used before to attach to the existing container.

Do you even stack, bro?

With the announcement of the Cloud Foundry Foundation initiative today, we have some really great things to look forward to in the cloud and virtualization industry.

The move was an indication in many ways about the direction in the landscape of infrastructure providers, and it is being driven by a new, agile, adapting consumer market. Plus, it is not just happening in Silicon Valley startups any more. This shift in the industry is starting to happen at the enterprise level, and is even seeping down into the small to medium (SMB) marketplace.

The Cloud Foundry architecture is a phenomenal example of adding an abstraction layer to the underlying infrastructure to allow for standardized delivery of application services.

Image source: http://www.slideshare.net/cdavisafc/cloud-foundry-technical-overview (slide 9 of 33)

You can already see that there are 3 specifically referenced virtualization platforms shown (VMware, OpenStack, Amazon Web Services) which was already a sign of how well adopted the platform has become.

Co-opetition is Good

In the past, there were many who frowned upon the idea of board members cross-pollinating, particularly when there was a commercial interest by some or all of those members. The success of the OpenStack Foundation in its governance model has opened the door for what Cloud Foundry is doing now with massive industry players IBM, EMC, VMware , and Rackspace each contributing 1.5 million dollars to the foundation.

Although there can be some challenges in the co-opetition space as the recent Piston-gate showed us, the ultimate winner of these enterprise players being focused on growth of an ecosystem will be the consumer. It is a profound change in our environment where we are seeing open source development gaining massive momentum, and both individual and corporate contributors are driving the growth.

Do you even stack, bro?

 I pulled that one from the classic “Do you even lift, bro?” meme. It speaks to the message being sent by this type of a foundation coming together, and how the “stack” is becoming very important. Stacking has long been tied to OpenStack given the name, but the reason that OpenStack was so named is because it is just that; it is an Open Stack.

The methodology of using stack infrastructure is the compartmentalized delivery of services which can be separated to “stacks” such as IaaS (Infrastructure-as-a-Service) and PaaS (Platform-as-a-Service).

Traditional datacenter models are slowly making an exit from the modern architect’s whiteboard. It is definitely a direction change for many, and it will be a slow adoption in the SMB space, but one way or another we will be seeing the shift towards stack-style infrastructure even in the <10 server market before too long in my opinion.

It wasn’t long ago that VMware became a household name and hypervisors have now permeated a

significant percentage of datacenters, even down to the SMB customers. In the same way I believe that cloud-style service delivery will become simpler, and more *Stack will become the standard installation. In light of that I'm voting for the change of vCloud to vStack…hey…I should trademark that

So what is the deal with Containers?

The move toward containers is not new. Linux has had containers available for quite some time using LXC and Libvirt, and the concept of sandbox virtualization of apps has existed in many environments. Not to mention the other vendor offerings like Ravello Systems and more that have give us template delivery of containerized virtual servers and virtual application environments.

And of course, there is Docker which has become the darling of the containerized virtualization space lately, and has spurred many others to add support for the format. Packer.io is another great tool that allows the creation of standardized images to be able to be used on multiple environments. We will see a lot of growth, and potentially fragmentation, in this area of development in the coming months. Ultimately, we will find that a widely adopted standard will rise to the top.

Are there too many "container" options?

One of the challenges in a burgeoning technology is that there can be a lot of players who enter the space and it can become crowded, especially in the absence of a "standard".

As the containerized ecosystem matures and gains more adoption, there will be key players that stand out among the many options which will inevitably guide us as we shift towards software abstracted, container delivery of applications and servers.

Join the movement!

With these great moves towards the adoption and development of open standards, we as architects, administrators and application designers will have to take a serious look at where we should be leaning when we look towards near-term and long-term deployment plans. If you do not have a cloud architecture already (private, public, or hybrid), you will most likely see one becoming a part of your IT portfolio in the coming months.

One way or another, this will enter into play in most environments. It may happen with cloud application adoption in the beginning, but once the comfort arises with that, the questions will come as to why our traditional virtual datacenters are not able to be consumed and provisioned in a similar fashion.

If you don't believe that, you can probably go 5-8 years back and the same statement will be said about virtualization platforms like VMware and Microsoft Hyper-V. Now is the time to get ahead of the shift before it is thrust upon you by your consumers which will leave you and your IT organization scrambling to catch up.

There are lots of options to get started with learning, and I encourage you to reach out to me directly on Twitter (@DiscoPosse) or even by email to eric – at – discoposse.com if you would like to chat on how to introduce stack infrastructure into your portfolio.

A little container fun

Just for those who remember it, I always think of this when I talk about containers. It’s for the old school Sesame Street fans