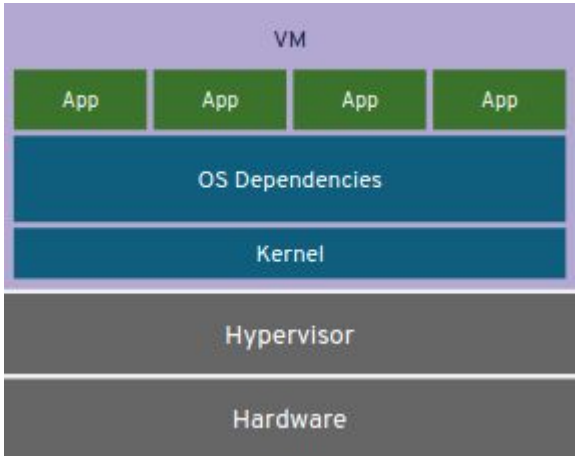
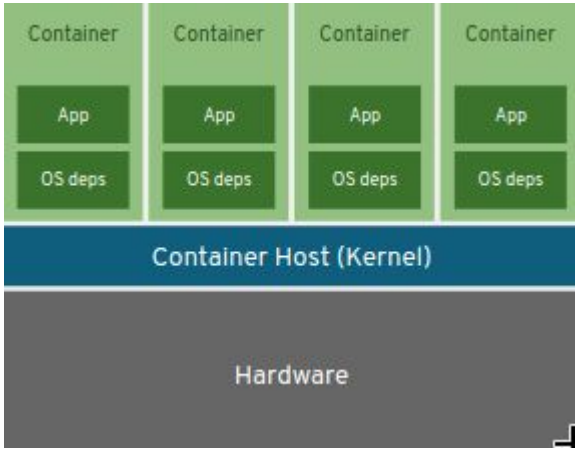

Kubevirt
Why What and How

Who is this person?

- John Griffith
- Developer at RedHat working on Kubernetes/Kubevirt
- You may remember me from past hits like SolidFire, OpenStack and even NetApp
- https://github.com/j-griffith
- http://j-griffith.github.io/
- @jdg_8

Quick Refresh, VM vs Container





Containers are the new “cool kids”

Automation and Declarative Systems RULE!

Nobody (or almost nobody) thinks:

> *“Hey, I should do a really cool presentation on how I maintain and run a custom production floor app built on Windows 2000 Server that can’t be replaced or run on anything but our customized version of Windows 2000!”*

Well, ok, that “might” be interesting… but not the type of thing you line out the door to see probably.

Fact is VMs aren’t really going away anytime soon

We still need VMs

Still a number of things you can’t do with Containers

- Custom Kernels
- Security/Isolation
- Monolithic Applications
- Redesign/Rewrite that old application (ummm… yeah sure)
- Old legacy apps, may not have licensing, tool chains etc
- Some things just don’t (or can’t) fit in a Container
- Hardware abstractions are really useful in some cases
- I just like my VM Pets, they’re part of the family

Just “stuffing” it in a Container might not go well

Run two environments, problem solved!

Legacy VM Env

- VMware
- OpenStack
- CloudStack
- ...

Shiny new/cool stuff

- Swarm
- Kubernetes
- ….

I can have it all!!! VM's when I need them, and Containers/K8s for everything else! I'm a "Happy Puppy"!

Run two environments, now I have a new problem!

Double Trouble

- Two environments to maintain
- Two WorkFlows
- Two platforms to manage

Managing infrastructure and the platforms on top of them can be a challenge, adding more of these can make for a “sad puppy”

Run K8s on top of my Virtualization

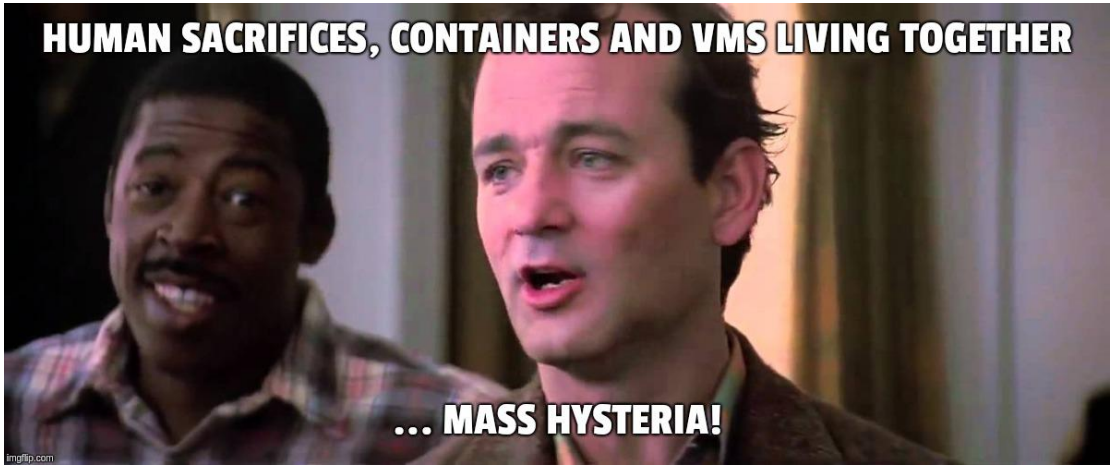
Not a bad approach

- Provides some flexibility
- Marks off the check boxes
- Fits the mental model of the Public Cloud Providers

Ummm… wait though

- I'm still managing two environments!
- I want bare metal performance
- I want K8s workflows for ALL MY THINGS
- K8s is what I want, and I want it NOW!

Running VMs AND Containers side by side in K8s



- Build, Modify and Deploy ALL THE THINGS in one way in one environment
- Single workflow for Devs/Ops
- Portability of VMs (wait.. whaaaat?)
- Only ONE environment to maintain!
- Migrate at your leisure (or don’t)

There’s a few approaches to this sort of thing

Mostly using new CRIs to control VMs (“*not that there’s anything wrong with that*”)

- RancherVM
- Kata-Containers
- Mirantis Virtlet
- Google gvisor

For now, *most* of these are focused on slightly different problems, mostly around isolation/security.

They’re cool too; you should check them out but let’s talk about a different approach...

Kubevirt

Not running Containers in VMs,instead let’s run and expose VMs in Containers!!

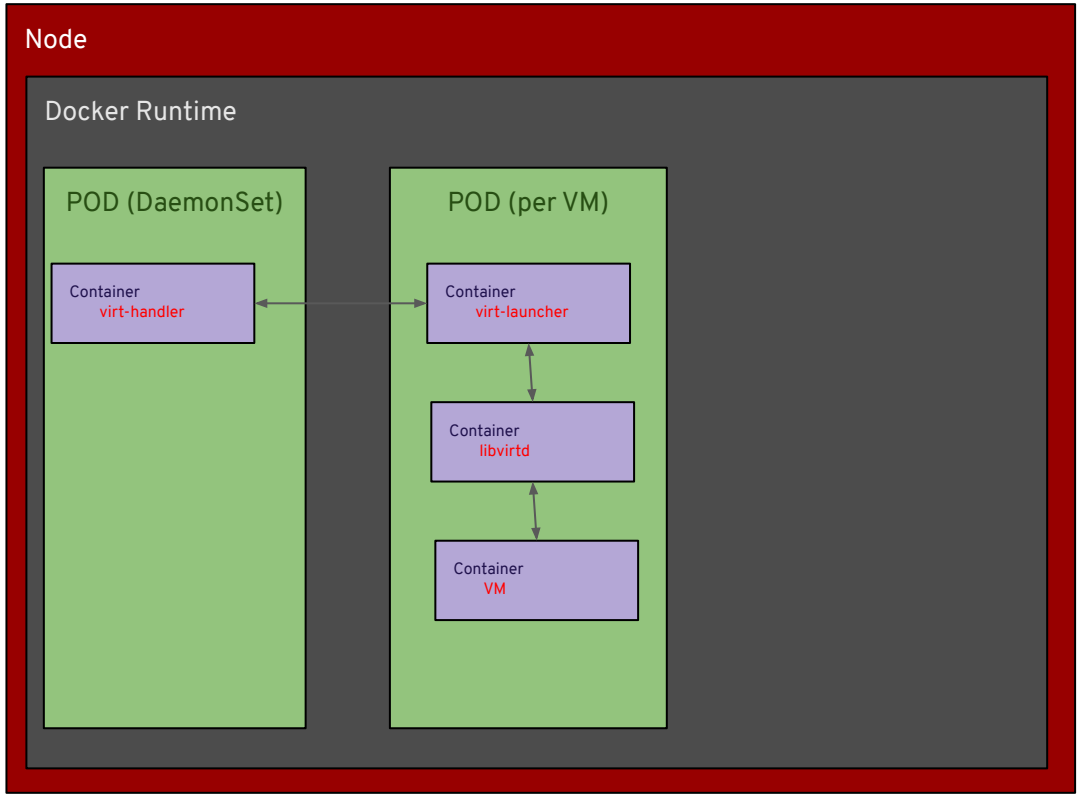
Then you’ll have:

- Unified platform to build, modify and deploy applications (Container or VM)
- BOTH VM workloads and Container workloads using the same automation, and K8s APIs
- Treat VMs just like any other K8's workload, while preserving its true “VMness”
- Bring your legacy VMs to a shiny new environment
- Working on leveraging virt-v2v to easily migrate existing VMs into the Cluster
- Main focus is enabling the migration of legacy apps/VMs in to a K8s world

A little bit of the “how”

- Extends an existing Kubernetes Cluster
- Just deploy it on your existing Cluster
  - Implemented as a Custom Resource Definition (CRD)
- Extends your K8s cluster to support VMs
- Sticks to K8s native approach as much as possible
  - Pod Networking
  - Storage supported in K8s works with your VMs
  - Just another Resource, same process using manifests etc

What does a Kubevirt VM look like?



Pretty Cool

“Hipster puppy”

Adding it to a running K8s Cluster

```
> export VERSION="v0.7.0"
> kubectl create -f https://github.com/kubevirt/kubevirt/releases/download/$VERSION/kubevirt.yaml
.....................
.....................
.....................
> curl -L -o virtctl \
    https://github.com/kubevirt/kubevirt/releases/download/$VERSION/virtctl-$VERSION-linux-amd64
.....................
.....................
.....................
> chmod +x virtctl
```

Notice we didn’t say anything about modifying your K8s Cluster or it’s Nodes, you don’t have to.

**In a production environment you would want to install KVM modules and enable nested virt*

Now it just works like any other K8s resource

```
> kubeclt create -f my-firstvm.yaml
virtualmachine.kubevirt.io "my-firstvm" created
virtualmachineinstancepreset.kubevirt.io "small" created
```

You can specify things in the manifest like you do with any other POD/Application, Volumes, Networking etc

Nice Extras

Kubevirt also has some extended projects like containerized-data-importer (CDI)

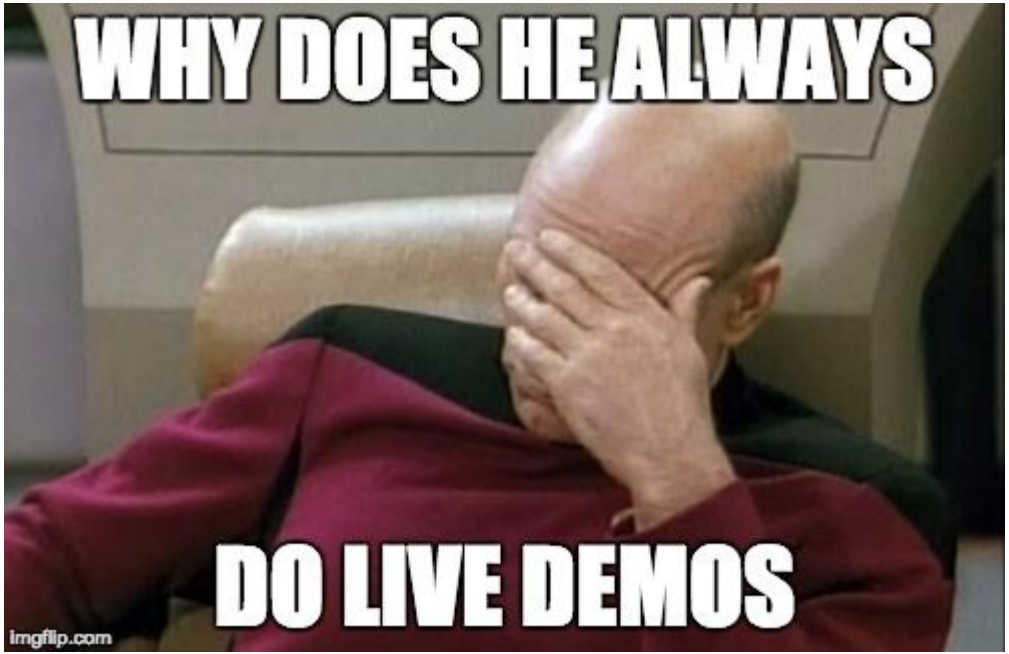
- Another Controller Add On for your Cluster
- Import existing Data or Images to PVCs in your Cluster
- Host assisted Cloning of PVs
- Basically “things to move data around more sensibly”

Check us out!

Lots of resources to easily try things out and get more details, find us:

- http://kubevirt.io/ (includes EASY tutorials)
- On Slack Kubernetes:virtualization
- @kubevirt (tweet-tweet)
- https://github.com/kubevirt
- At Kubecon in Seattle
- Weekly meetings?

Demo?



Because if things go wrong we'll all probably get something out of it (at least a good laugh).