oVirt

Kubernetes - Load Balancing For Virtual Machines (Pods)

4th of Feb 2018

Yanir Quinn
Senior Software Engineer
Red Hat

About me and oVirt/Kubevirt

Yanir Quinn

Working for Red Hat since 2016

oVirt SLA team / KubeVirt

oVirt

- oVirt | Red Hat Virtualization
- KubeVirt

Agenda

A quick intro to Kubevirt

What is Kubernetes + Kubevirt in a nutshell

Scheduling in Kubernetes

Basic characteristics of scheduling in kubernetes

Why load balancing ?

What is missing in current Kubernetes release
What work can be done or in progress

Dscheduler

Kubernetes incubator project

Dry run functionality for Kubernetes

What is scheduling simulation

Dry run scheduling use cases

VM migration

What is VM migration ?
How can it be reflected on pods ?

Load balancing algorithm

An example for a load balancing algorithm that can
Make use of dry run scheduling

Kubernetes

- Kubernetes is about managing tasks and applications at cloud scale

- By using container runtimes It allows managing containers on a cluster-level

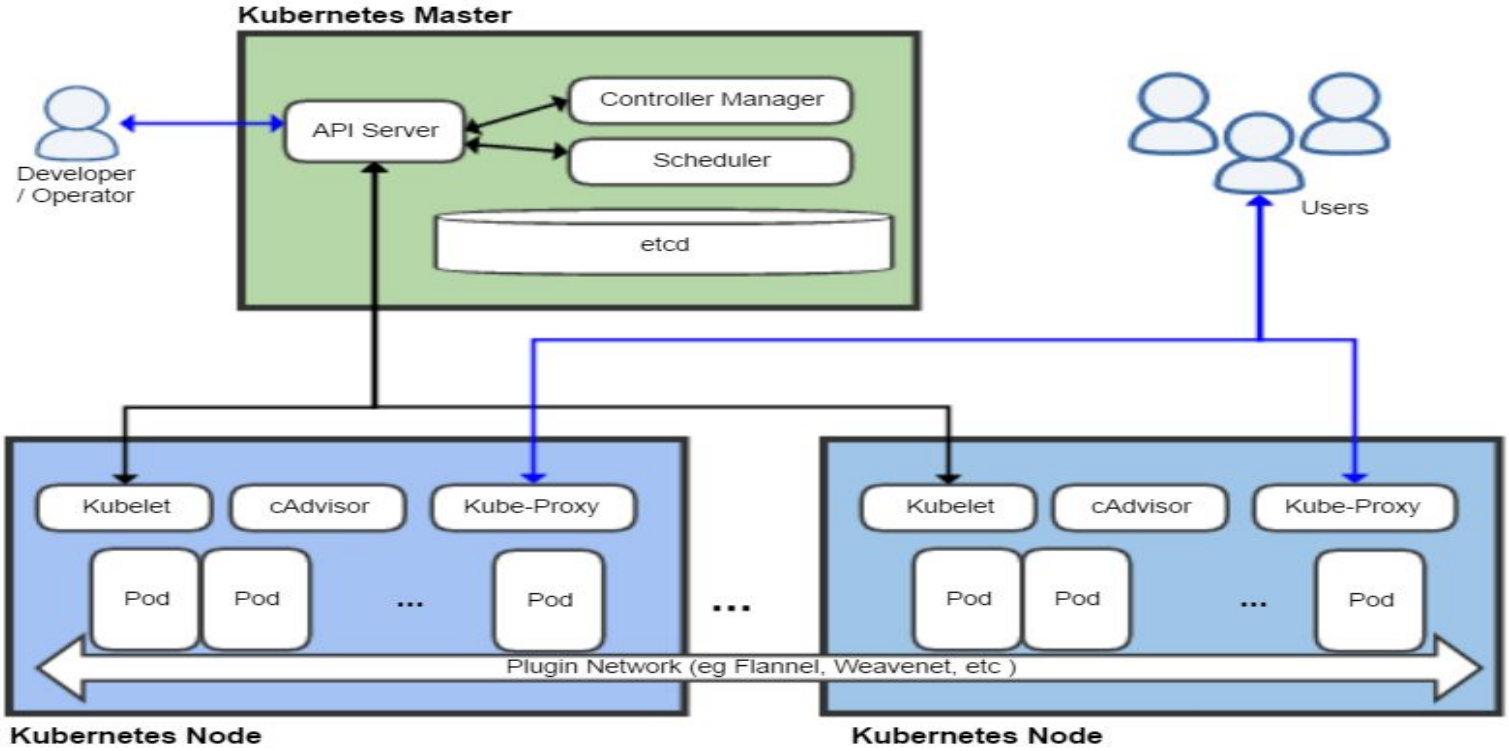
With Kubernetes You Can

- Orchestrate containers across multiple hosts
- Make better use of hardware to maximize resources needed to run your enterprise apps
- Control and automate application deployments and updates
- Mount and add storage to run stateful apps

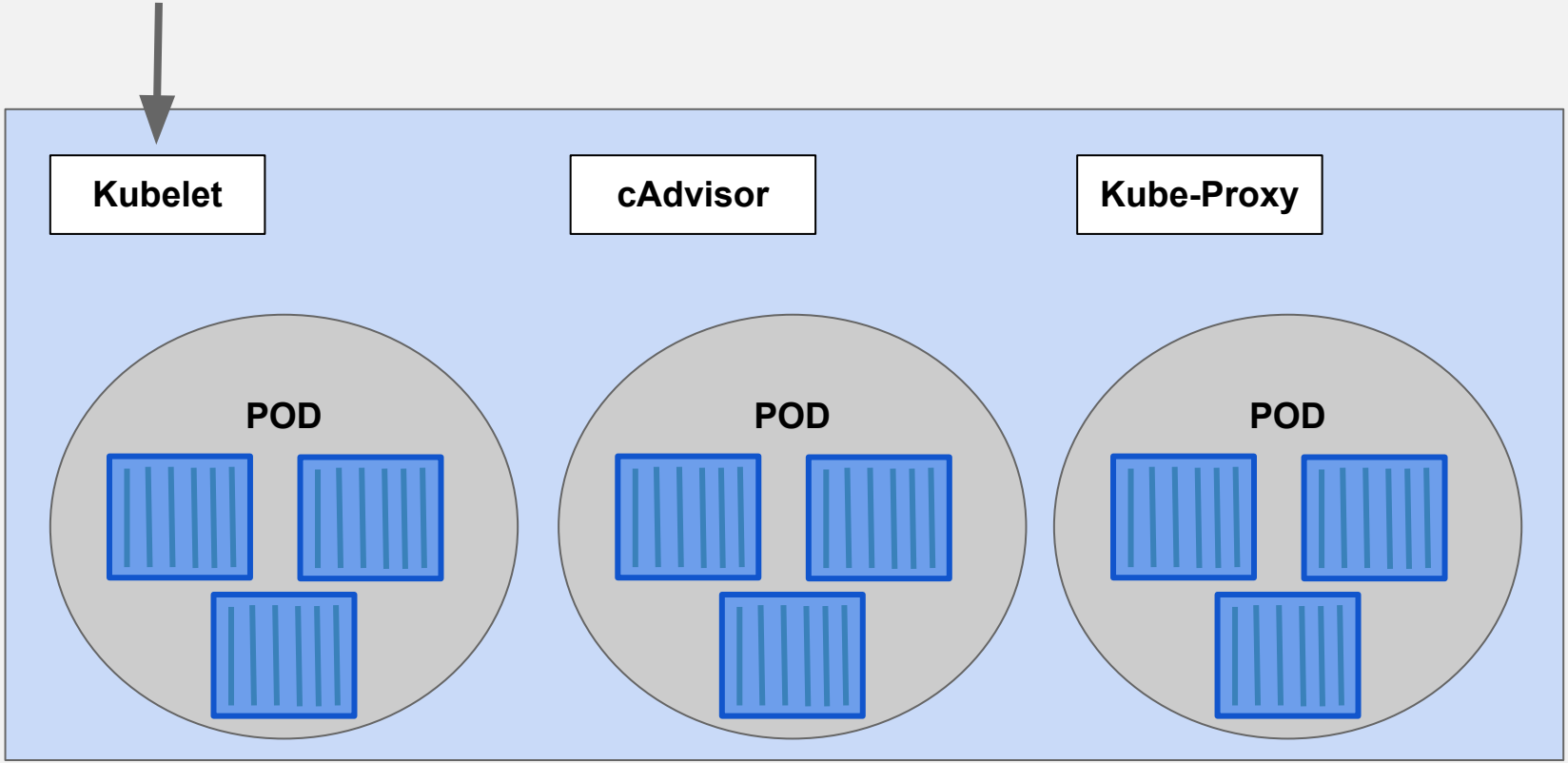
With Kubernetes You Can

- Scale containerized applications and their resources on the fly
- Declaratively manage services, which guarantees the deployed applications are always running how you deployed them
- Health-check and self-heal your apps with autoplacement, autorestart, autoreplication, and autoscaling

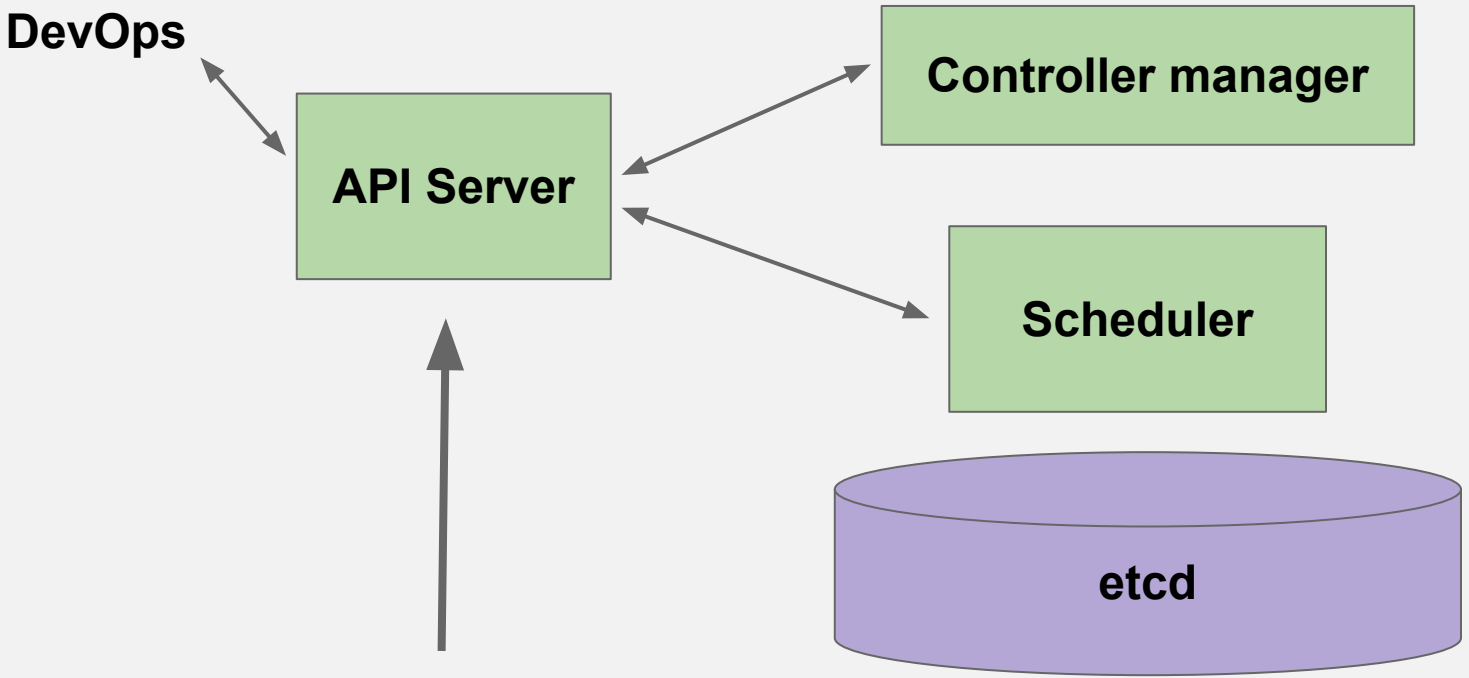
Kubernetes


Kubernetes Master
Developer
/ Operator
API Server
Controller Manager
Scheduler
etcd
Users
Kubelet
cAdvisor
Kube-Proxy
Pod
Pod
...
Pod
Kubelet
cAdvisor
Kube-Proxy
Pod
Pod
...
Pod
Plugin Network (eg Flannel, Weavenet, etc)
Kubernetes Node
Kubernetes Node

Kubernetes Node



Kubernetes Master



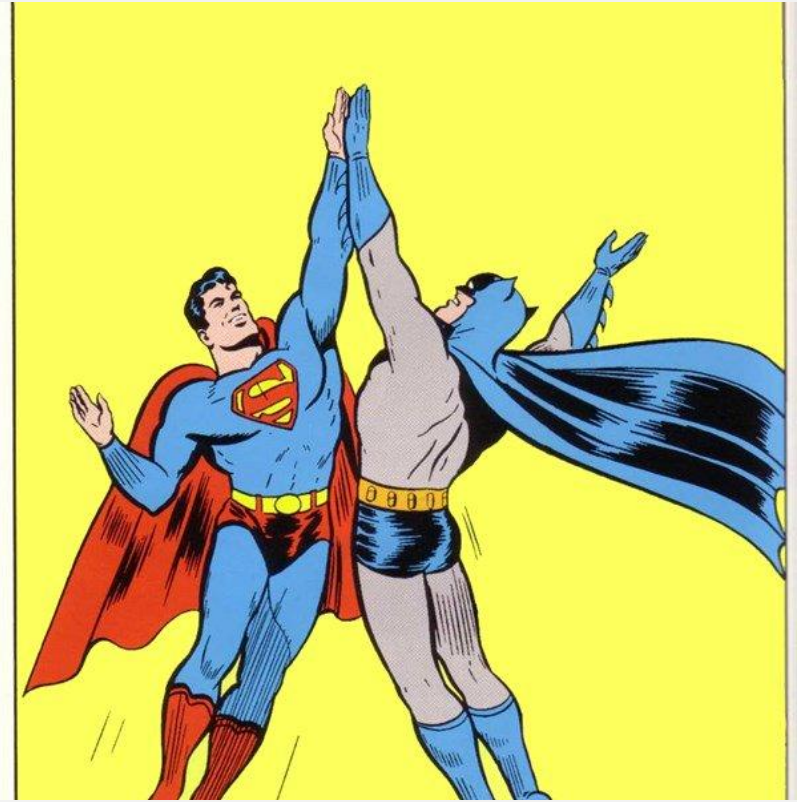
KubeVirt

Tries to close the gap between containers and VMs by enabling VM management on top of Kubernetes and integrating the VM into Kubernetes infrastructure where possible

https://github.com/kubevirt

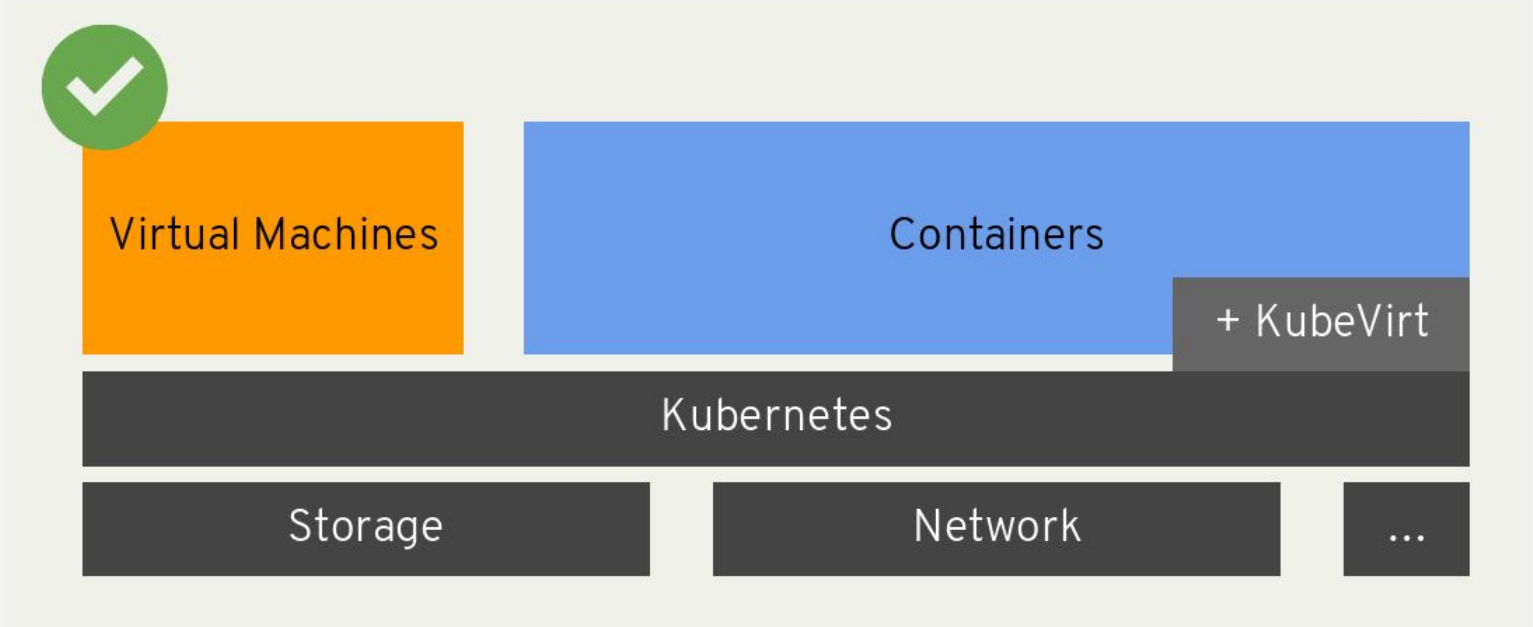
Converged Infrastructure

Virtual Machines



Containers

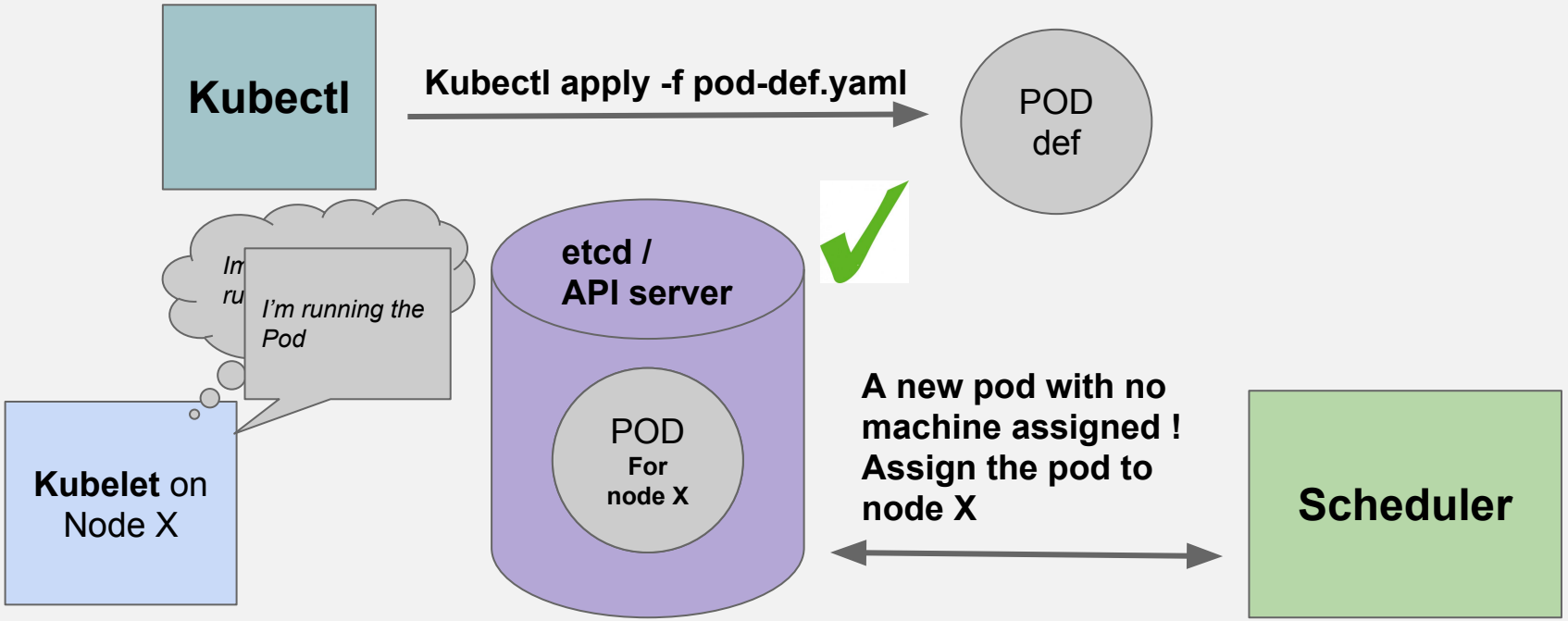
KubeVirt


Virtual Machines
Containers
+ KubeVirt
Kubernetes
Storage
Network
...

KubeVirt

- Drops directly into existing Kubernetes Clusters
  - No additional host setup
  - Simple install
  - Extends Kubernetes so VMs can be scheduled alongside Containers
- Ties VMs into Pod Network
- Integrates with other Kubernetes concepts
- Manage VMs like Pods

Scheduling in Kubernetes

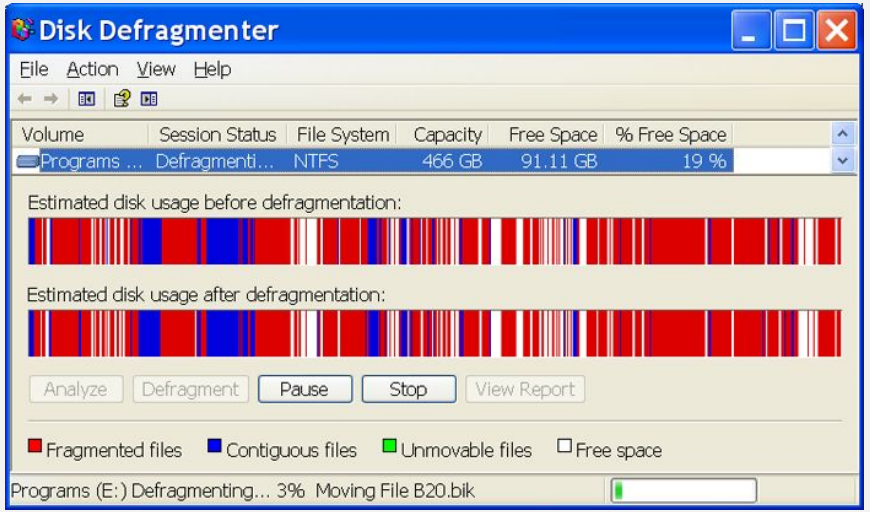


Why Load Balancing? (Rescheduling)

Free "Holes"

Scheduling decisions are only made at the time Pods are created

Optimize cluster layout



Current Work Related to Rescheduling

- Autoscaling
- Eviction Policy
- Affinity + Taints and Tolerations
- Pod Priority and Preemption
  - Critical add-ons

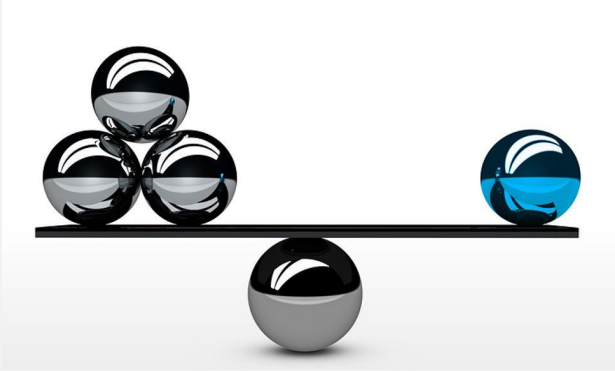
Current Work Related to Rescheduling

- Cluster Capacity
- Dscheduler

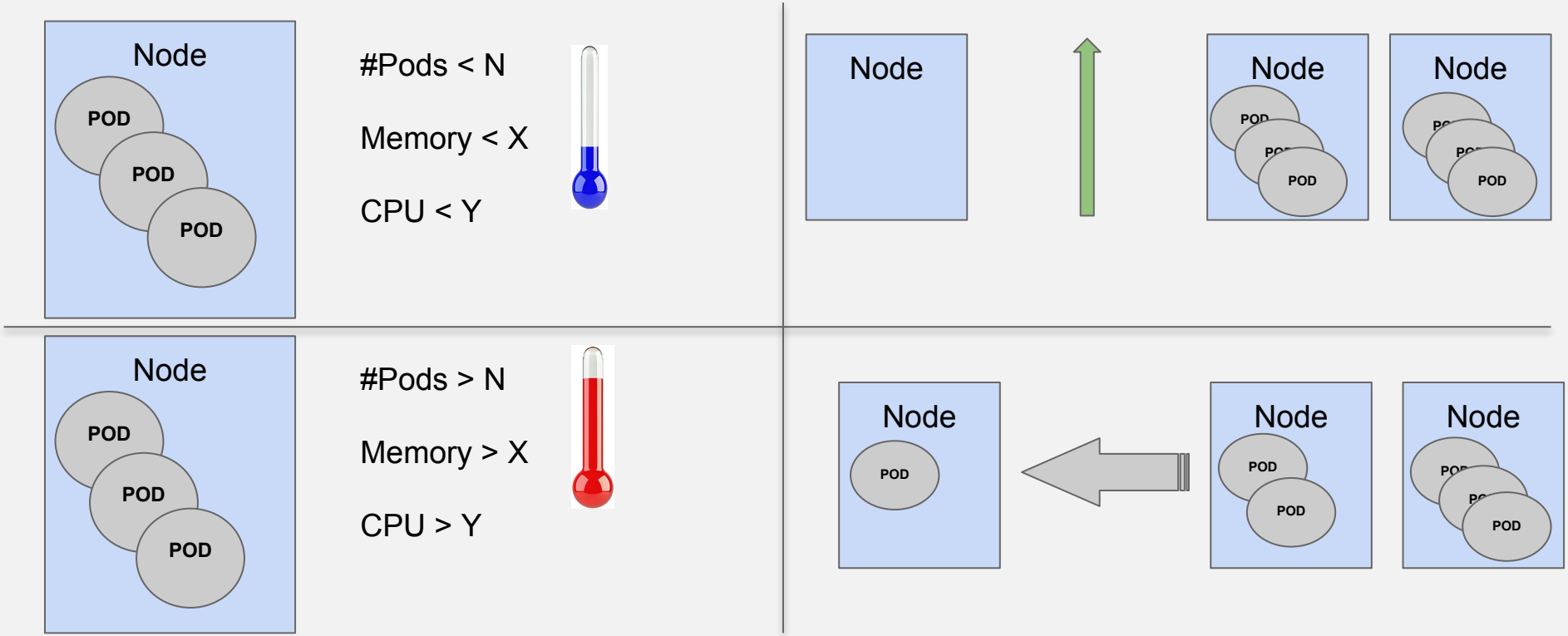
Kubernetes Incubator

Dscheduler

- Set a strategy for balancing the cluster
- Find pods that could be evicted
- Scheduler will reschedule the evicted pods
- Minimal to no disturbance on the cluster



Dscheduler - Policies and Strategies



Dscheduler - Strategies

```
apiVersion: "descheduler/v1alpha1"
kind: "DeschedulerPolicy"
strategies:
  "LowNodeUtilization":
     enabled: true
     params:
       nodeResourceUtilizationThresholds:
         thresholds:
           "cpu" : 20
           "memory": 20
           "pods": 20
         targetThresholds:
           "cpu" : 50
           "memory": 50
           "pods": 50
```

Nodes under that bound indicate Underutilization

Potential nodes from where pods Could be evicted

Dscheduler - Strategies

```
nodeResourceUtilizationThresholds:
  thresholds:
    "cpu" : 20
    "memory": 20
    "pods": 20
  targetThresholds:
    "cpu" : 50
    "memory": 50
    "pods": 50
```

Nodes under that bound indicate Underutilization

Potential nodes from where pods Could be evicted

Dscheduler - Eviction Policy

- Pods that cannot be evicted :
  - Critical
  - Not expected to terminate
  - Provide machine specific service
  - Have local storage

- Best efforts pods are evicted before Burstable and Guaranteed pods.

What Might Be Missing ?

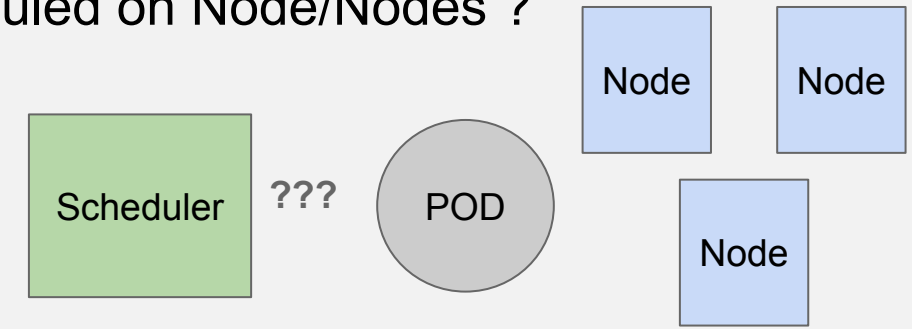
- Optionally verify if can schedule and where
- An accurate simulation of Pods reassignment
  - No duplicates of a Pod’s resource consumption
- Actual rescheduling of replacement pods to better nodes

Dry Run Functionality for Kubernetes

- Scheduler "dry run" endpoint that takes a pod and tells you which nodes it can schedule onto

- **Output**: A destination Node for a newly created Pod (without performing any real binding)

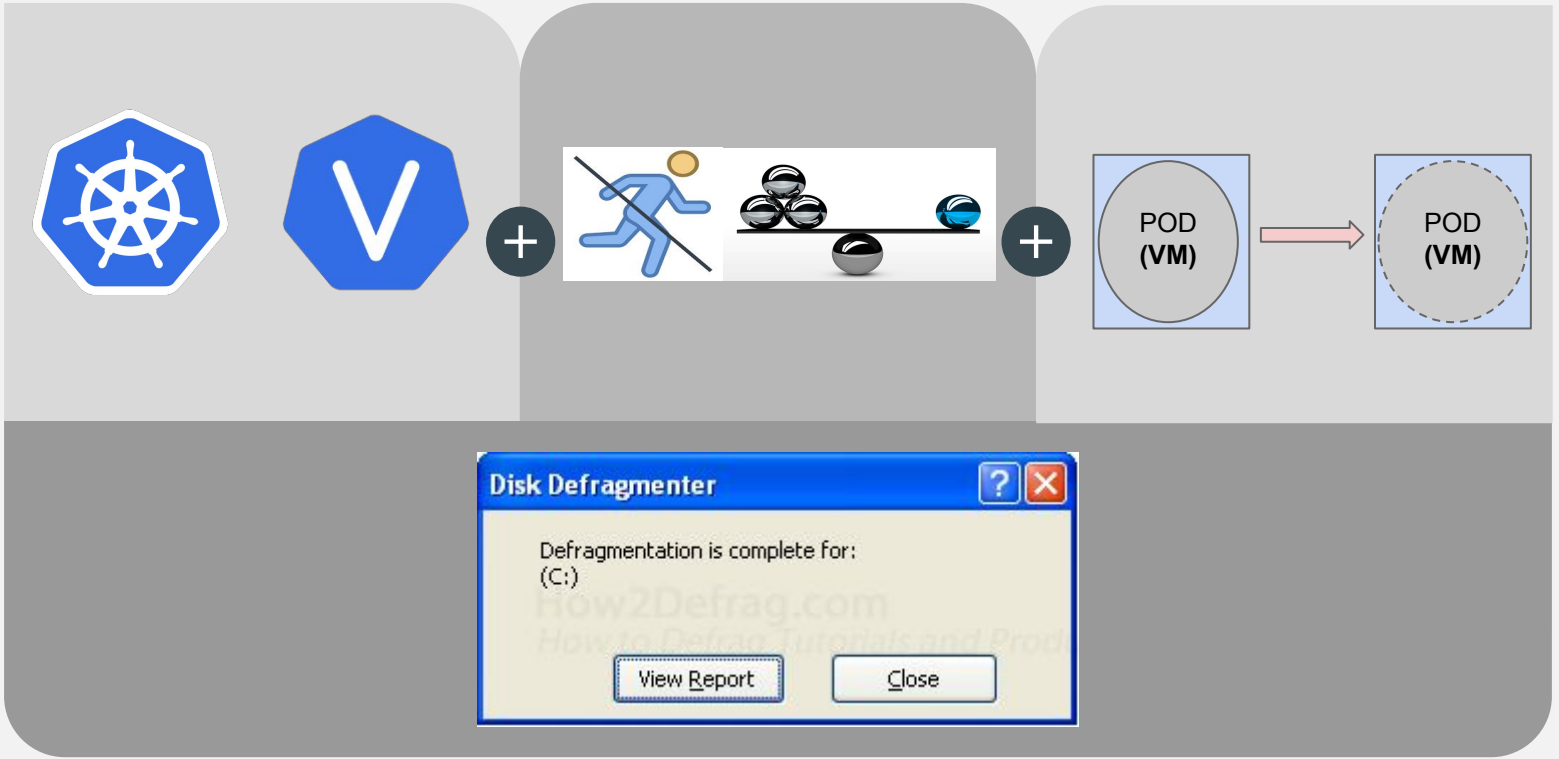
- **Output**: Can a Pod be scheduled on Node/Nodes ?



Dry Run Functionality Use Cases

- Rescheduling / Load Balancing
- Simulation of the system's state after pods creation
  - Including assignment to nodes
  - Cluster capacity analysis, Autoscaler, DeamonSet Controller, etc.
- Using a replacement scheduler with stricter predicates
- Obstacles

Load Balancing Virtual Machines



Virtual Machines Migration

```
kind: Migration
metadata:
  generateName: my-migration
spec:
  nodeSelector:
    kubevirt.io/hostname: node1
  selector:
    name: testvm
status:
  phase: Succeeded
```

Backed by a controller:

- On object create, schedules a new Pod
- On successful Pod start, it triggers the migration
- At the end of the migration the object is moved to a final state
- Always **one** VirtualMachine object you reference

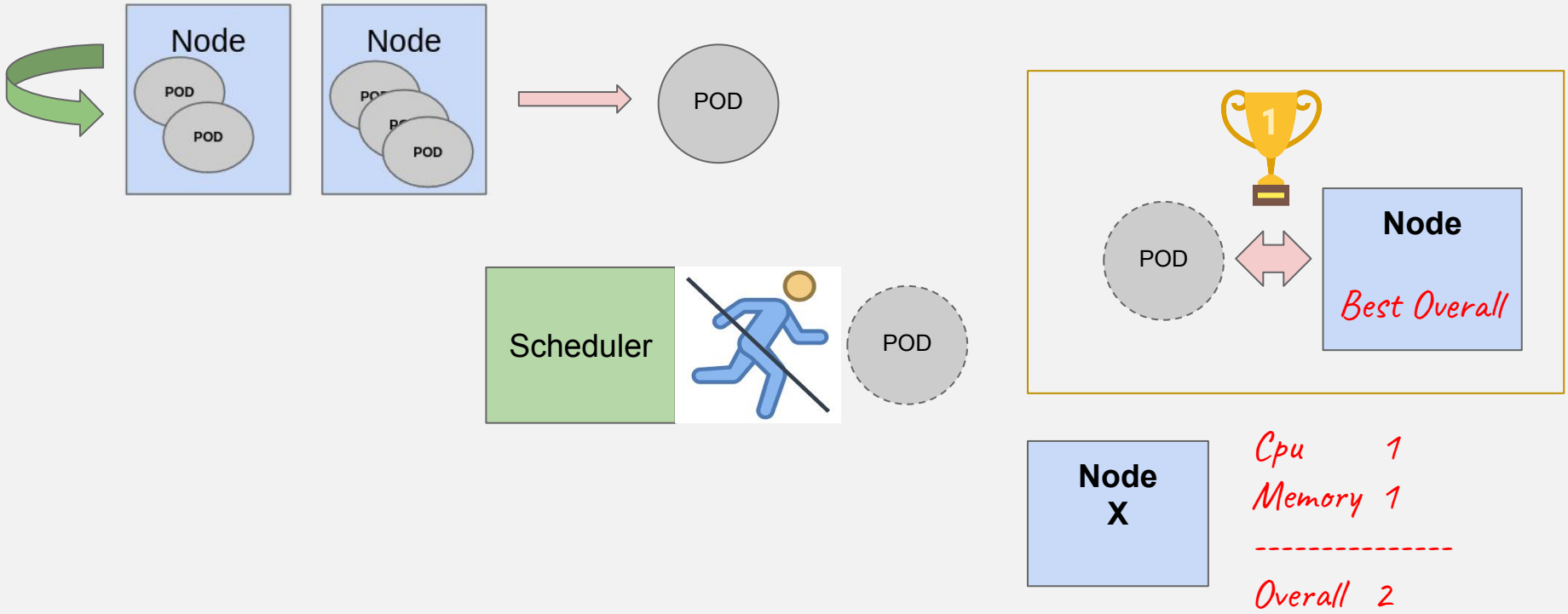
The objects **Migration** with **VirtualMachine** provide a consistent entry point to anything VirtualMachine related, like the Pod does for Kubernetes.

So How Can It Work On KubeVirt ?

- Integrate Dscheduler
  - Pod gets marked for deletion
  - Pod deletion blocked
  - Virtual machine is migrated behind the scenes

- Explicit load balancing based on the migration object

Load Balancing Algorithm



Summary

- **It's about balancing VMs**
- **Leveraging Kubernetes**
- **More to come**

oVirt

THANK YOU

http://www.ovirt.org

users@ovirt.org

yquinn@redhat.com