Cloudbursting with Kubernetes

Using Kubernetes Federation and Cluster Autoscaler

Irfan UR Rehman
@irfanurrehman

Quinton Hoole
@quinton-hoole

Agenda

- Intro
- Possible cloudbursting setups with kubernetes
- Quick look at k8s federation
- Quick look at k8s cluster autoscaler
- Reference implementation
- Demo

Audience

- Needs to be well versed with kubernetes
- Needs to have some idea about kubernetes federation project
- Needs to have some idea about kubernetes cluster autoscaler

Cloud Burst

- https://azure.microsoft.com/en-in/overview/what-is-cloud-bursting/
  - In cloud computing, cloud bursting is a configuration which is set up between a private cloud and a public cloud to deal with peaks in IT demand.
- The idea applicable to any hybrid cloud configuration, involving a low cost cloud with limited resources and and a higher cost cloud with much larger set of available resources on demand.

- We focus this talk to achieving something similar with kubernetes for kubernetes styled containerized workloads.

Some Background...

Kubernetes Federation - Origins (v1)

- Idea incubation - somewhere in 2015.
- Work incubation 2015 Dec as **Ubernetes Lite.**
- Name upgraded to **Kubernetes Federation** around 2016 Oct with more serious participation from multiple organisations.
- Initial functionality came in as part of k8s release 1.5.
- Moved outside core into its own repo around 2017 Sep.

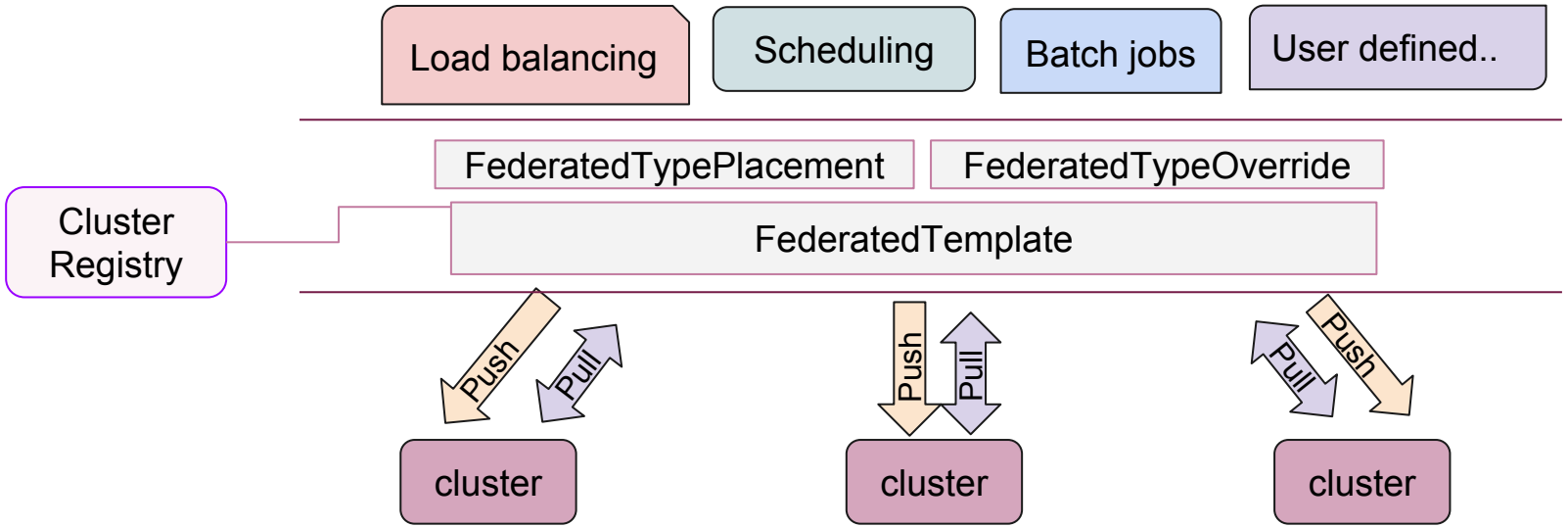
Kubernetes Federation - Goals

- **Capacity Overflow (aka cloudbursting)**
  - What happens if I run out of capacity in my cluster
- **Sensitive Workloads**
  - I have multiple clusters but want to run sensitive workloads only in specific clusters
- **Vendor lock-in avoidance**
  - Run workloads in multiple service providers clusters
- **HA**

Kubernetes Federation - Evolution (v2)

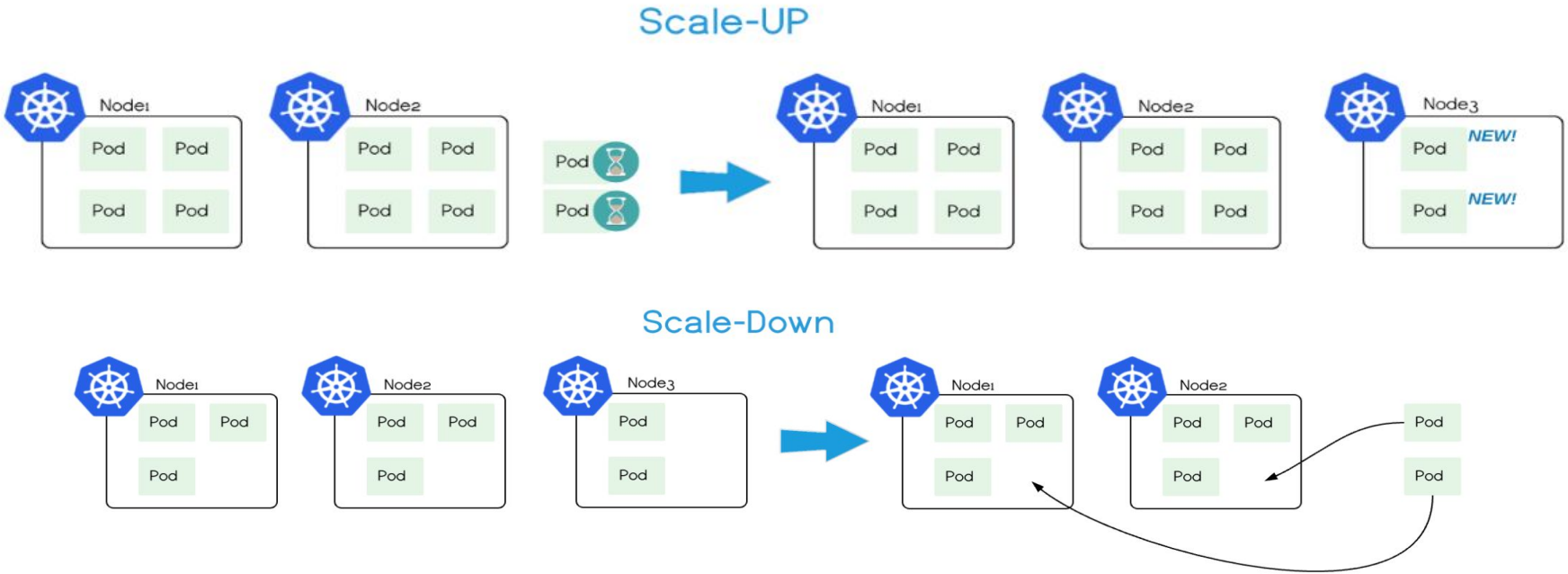
- Federated information which lived in annotations in v1 gets its own API resources.
- Primary federated resource templatises and wraps corresponding k8s resource (akin to pod template in replicaset). For example:
  - FederatedSecret has k8s v1/secret.
- Further broken into multiple API resources:
  - Modularity
  - Composability
  - Enables multiple flavours of controllers implementations

Kubernetes Federation - Evolution (v2)



Coming back to need based workload scaling...

Kubernetes Cluster Autoscaler



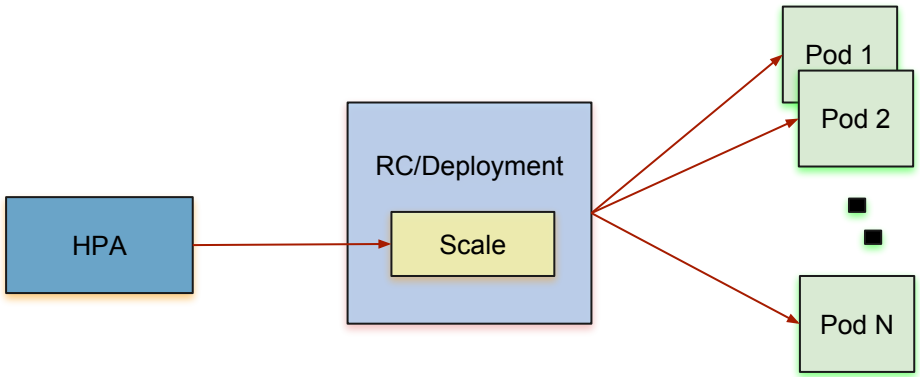
** image source - http://blog.spotinst.com/2017/06/14/k8-autoscaler-support/

Kubernetes Cluster Autoscaler

- Tool that automatically adjusts the size of a kubernetes cluster.
- The current design works on minimal disruption:
  - Scale the cluster up by adding nodes, if there are pods which fail to run due to insufficient resources
  - Scale the cluster down if some nodes in the cluster are so underutilized, for an extended period of time, that they can be deleted and their pods will be easily placed on some other, existing nodes
- Available for GCE, GKE, AWS and Azure.
- Already marked as GA.

Kubernetes Pod Autoscaler

- Scales Deployments/Replicasets using the scale interface.
- Scales the number of pods to meet the match the observed average CPU utilization to the target specified by user.



Cloud bursting using k8s clusters?

Single cluster ?

- A k8s cluster is typically homogeneous set of nodes (a single cloud provider/onprem), but these aren't hard requirements for k8s.
- A single k8s cluster can span on prem infra and a public cloud.
- Limited set of local node pool and a mechanism/toolset which provisions and joins nodes from a public cloud when needed.

Problems

- Security
- Networking
- Inter cluster network delays

Not viable/recommended

Multiple clusters in different clouds

- A local k8s cluster.
- One or more remote clusters.
- A mechanism/toolset which can monitor the application metrics, provision the cluster(s) in public cloud(s) and replicate, scale and/or migrate the application when needed.

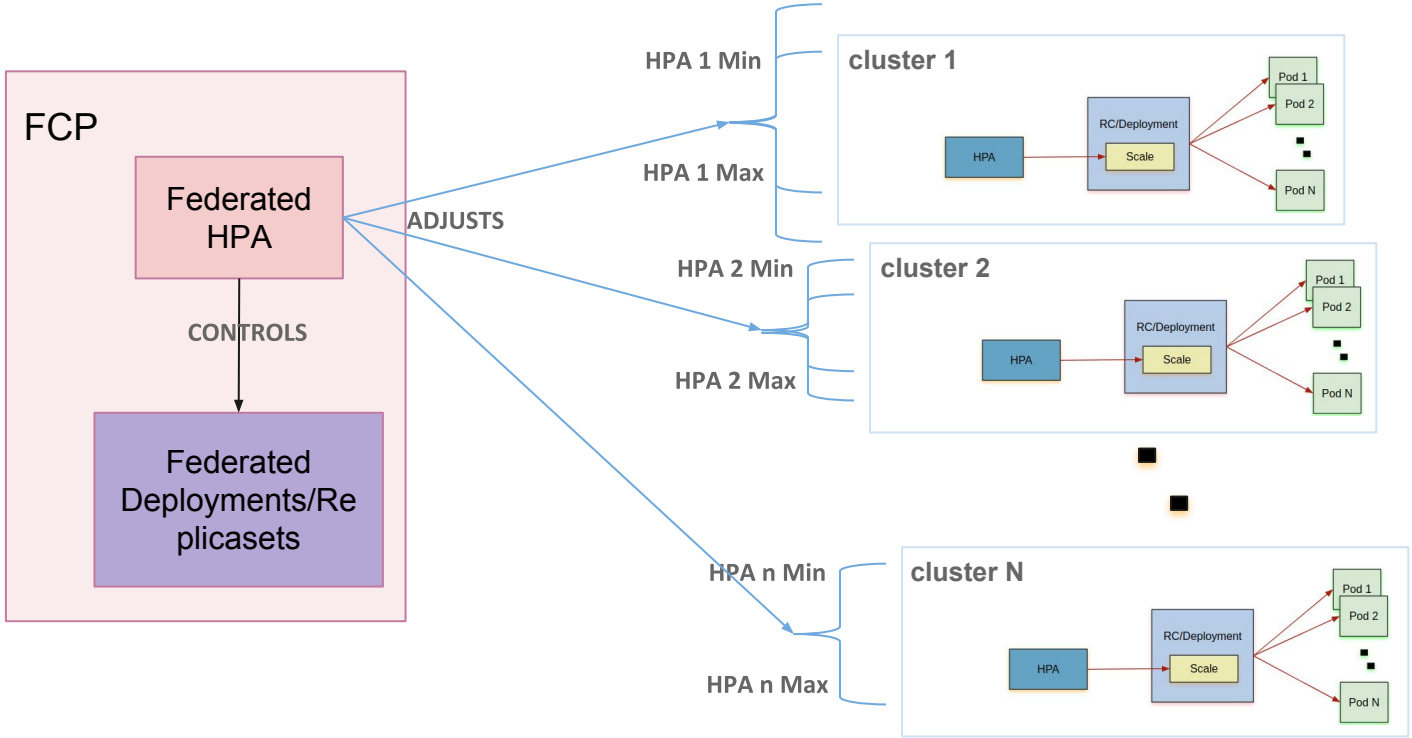
Problems

- Complex
- Unified application access (need LBs which can seamlessly migrate traffic also)

No available open src solution

Federated Pod Autoscaling with Cluster Autoscaling

Federated Pod Autoscaler



Federated Pod Autoscaler

Federation v1

HPA [exactly same as k8s]

```
apiVersion: autoscaling/v1
kind: HorizontalPodAutoscaler
metadata:
  name: php-apache
  namespace: default
spec:
  maxReplicas: 10
  minReplicas: 1
  scaleTargetRef:
    apiVersion: extensions/v1beta1
    kind: Deployment
    name: php-apache
  targetCPUUtilizationPercentage: 50
```

Preferences [proposed as annotations]

```
apiVersion: autoscaling/v1
kind: HorizontalPodAutoscaler
metadata:
  name: php-apache
  ....
  annotations:
    federation.kubernetes.io/hpa-preferences: |
      "us-east-a": {minReplica: 1, maxReplica: 4, weight: 1}
      "us-west-a": {minReplica: 1, maxReplica: 6, weight: 0}
---- OR ----
      "*": {minReplica: 1, maxReplica: 5}

spec:
  ....
```

Federated Pod Autoscaler

Federation v2 - Proposal

HPA [Federated wrapper]

```
apiVersion: federation.k8s.io/v1alpha1
kind: HorizontalPodAutoscalerTemplate
metadata:
  name: php-apache
  namespace: default
spec:
  template:
    apiVersion: autoscaling/v1
    kind: HorizontalPodAutoscaler
    metadata:
      name: php-apache
      namespace: default
    spec:
      maxReplicas: 10
      minReplicas: 1
    ....
    ....
```

Preferences [Proposed First class API]

```
apiVersion: federatedscheduling.k8s.io/v1alpha1
kind: HPASchedulingPreference
metadata:
  name: php-apache
  namespace: default
Spec:
  Targetref:
    apiVersion: federation.k8s.io/v1alpha1
    kind: DeploymentTemplate
    name: php-apache
  Clusters:
    “us-east-a”: {minReplica: 1, maxReplica: 4, weight: 1}
    “us-west-a”: {minReplica: 1, maxReplica: 6, weight: 0}
```

Out of the box

- One on prem cluster, with limited capacity (non-scaleable)
- One or more minimal (say 1 node) clusters in public clouds (scaleable)
- Pod autoscaling and cluster autoscaling enabled
- Federation

Problems

- A minimal cluster is needed in each cloud/infrastructure
- Unified application access

USEFUL - Yes

Out of the box...

However currently suitable for:

Workload Type	Communication between replicas	Suitability
Replication Workloads	Replicas don’t talk	Extremely suitable
Replication Workloads	Replicas talk	Cluster local access
Stateful Workloads	Replicas talk	No support yet

Demo

Extending

- Implement a cluster provisioner plugin with trigger interfaces mapped into federated HPA, if maintaining a minimal cluster is also a cost.
- Implement support for stateful workloads.
- Implement support for multiple metrics and custom metrics in federated HPA.

Q & A