

What's new in Kubernetes v1.19?

Thomas Hartland

Kubernetes v1.19

- v1.19.0 released on 26th August
- v1.19.1 bugfix released on 9th September
- v1.19.2 bugfix released on 16th September

- Our template released soon

Release notes

- The full release notes are available here
<https://kubernetes.io/docs/setup/release/notes/>

The biggest changes

- When the cluster autoscaler is deployed it now uses node group auto-discovery for v1.19 clusters¹.
- In v1.19 the kube-scheduler can be configured with multiple scheduling policies.

¹<https://github.com/kubernetes/autoscaler/releases/tag/cluster-autoscaler-1.19.0>

Cluster autoscaler

- Create a cluster

```
$ openstack coe cluster create \  
  --cluster-template kubernetes-1.19.2-1 \  
  --merge-labels \  
  --labels auto_scaling_enabled=true \  
  my-cluster
```

Cluster autoscaler

- The default-worker node group will not be autoscaled until you set a maximum size for it.

```
$ openstack coe nodegroup update  
    my-cluster default-worker replace /max_node_count=5
```

Cluster autoscaler

- Create a node group

```
$ openstack coe nodegroup create \  
  my-cluster \  
  --nodes 3 \  
  --min-nodes 1 \  
  --max-nodes 5 \  
  --role autoscaling \  
  my-node-group
```

Cluster autoscaler

- Any other roles can be autoscaled by editing the cluster-autoscaler deployment.

```
--node-group-auto-discovery=magnum:role=worker,autoscaling,other-role
```


Cluster autoscaler

- A node group must match one of the configured roles and have a maximum node count set to be able to be autoscaled.

```
$ openstack coe nodegroup update  
    my-cluster default-worker replace /max_node_count=5
```

- To disable autoscaling for a group, remove the max node count.

```
$ openstack coe nodegroup update  
    my-cluster default-worker remove /max_node_count
```

Scheduling policies

- k8s v1.19 allows the kube-scheduler to be configured with multiple scheduling policies.
- We are using this to define a bin-packing policy in all v1.19 clusters.

Scheduling policies

- To use another scheduling policy, set `schedulerName` in the pod spec:

```
kind: Deployment
```

```
spec:
```

```
  template:
```

```
    spec:
```

```
      schedulerName: bin-packing-scheduler
```

- The scheduler will use the bin packing policy for pods in this deployment.

Scheduling policies

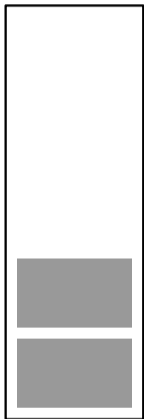
```
apiVersion: kubescheduler.config.k8s.io/v1beta1
kind: KubeSchedulerConfiguration
profiles:
  - schedulerName: default-scheduler
  - schedulerName: bin-packing-scheduler
plugins:
  score:
    disabled:
      - name: NodeResourcesLeastAllocated
    enabled:
      - name: NodeResourcesMostAllocated
      weight: 100
```

Scheduling policies

- Other policies can be created by enabling/disabling the available plugins², or changing their weightings.

²<https://kubernetes.io/docs/reference/scheduling/config/#scheduling-plugins-1>

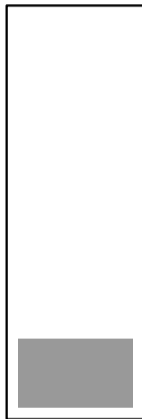
Node 1

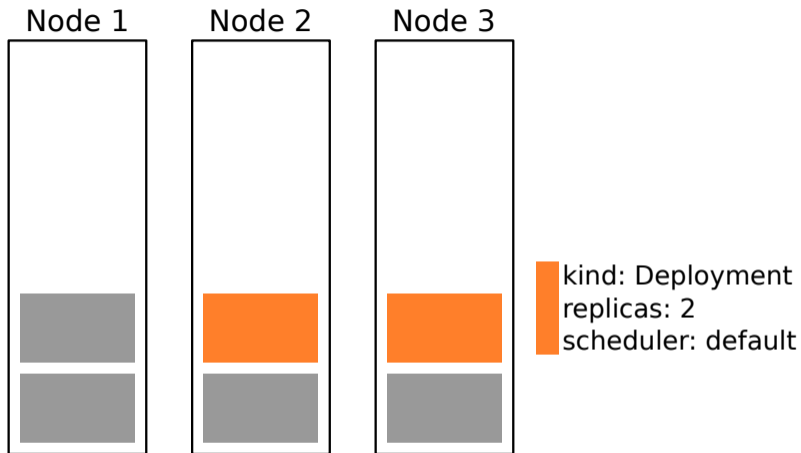


Node 2

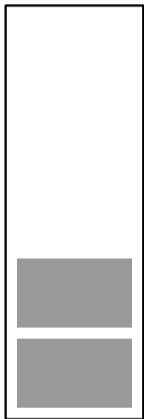


Node 3

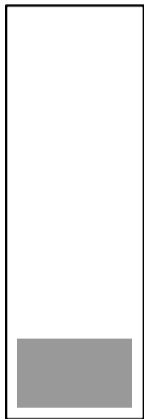




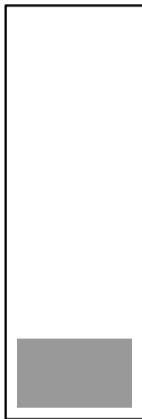
Node 1

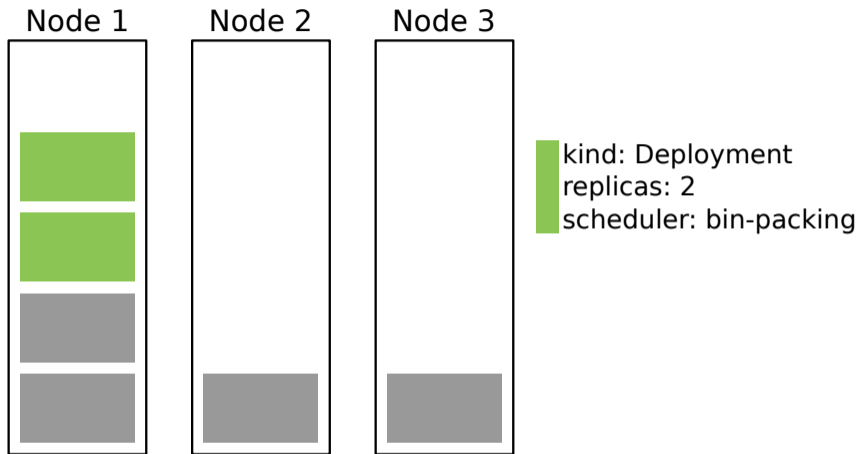


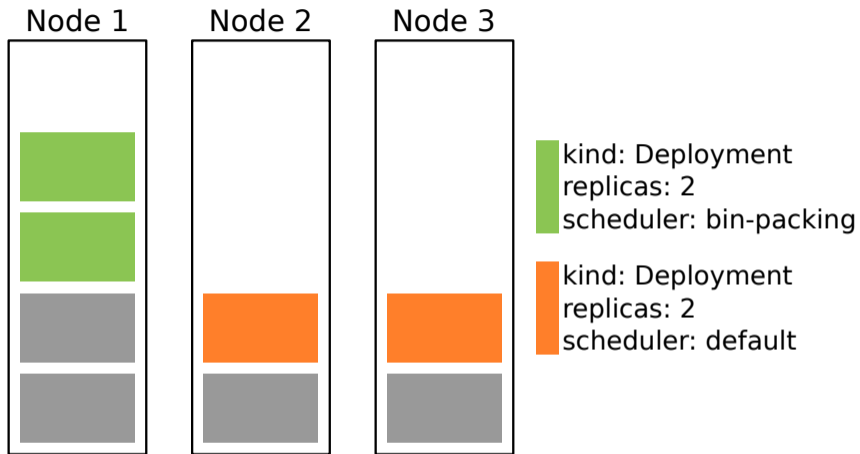
Node 2

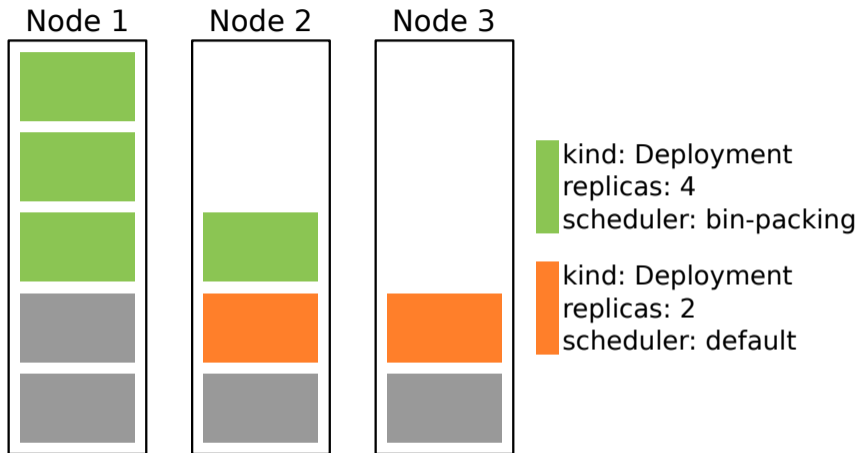


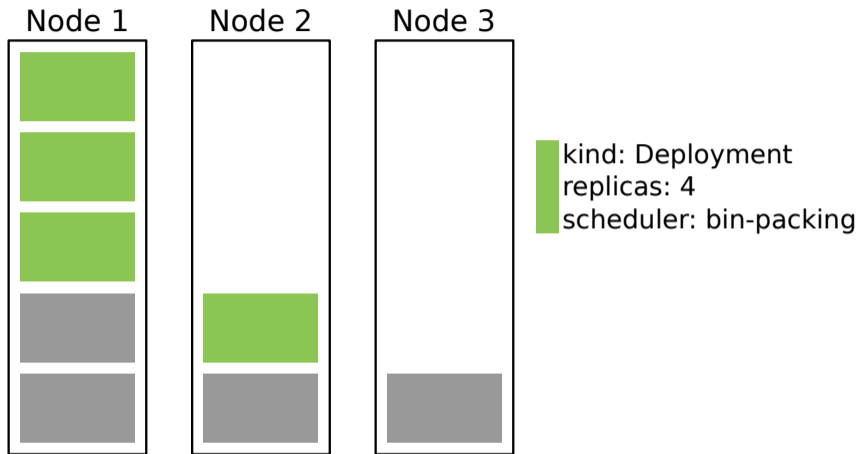
Node 3

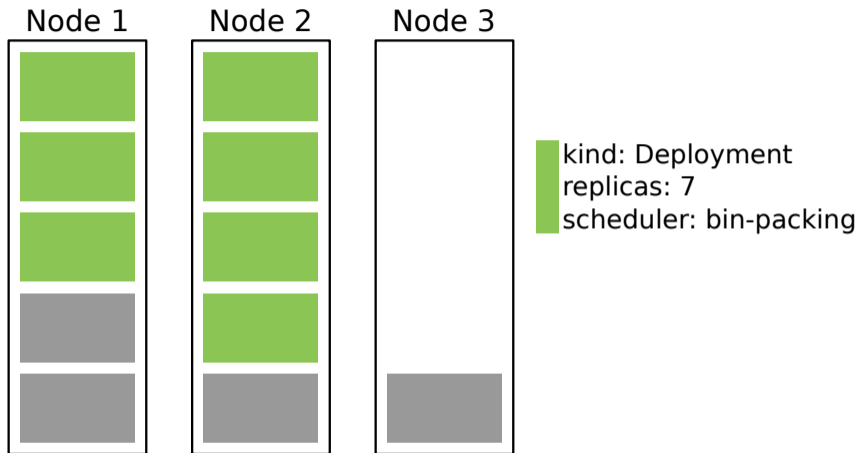


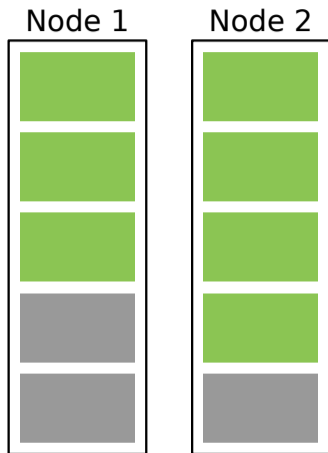












kind: Deployment
replicas: 7
scheduler: bin-packing

Ingress moves from beta

```
apiVersion: networking.k8s.io/v1beta1  
kind: Ingress
```



```
apiVersion: networking.k8s.io/v1  
kind: Ingress
```

Ingress moves from beta

In beta:

```
spec:  
  rules:  
  - http:  
    paths:  
    - path: /  
      backend:  
        serviceName: test  
        servicePort: 80
```


Ingress moves from beta

In v1:

```
spec:
```

```
  rules:
```

```
  - http:
```

```
    paths:
```

```
    - path: /
```

```
      backend:
```

```
        service:
```

```
          name: test
```

```
          port:
```

```
            number: 80
```

Immutable secrets/configmaps

- Adding `immutable: true` to the definition of a secret³ or configmap⁴ marks it as read-only.
- The only way to change the content of the secret/configmap is to delete and recreate it.

³<https://kubernetes.io/docs/concepts/configuration/secret/#secret-immutable>

⁴<https://kubernetes.io/docs/concepts/configuration/configmap/#configmap-immutable>

kubectl debug still in alpha

- Accessible under `kubectl alpha debug`.
- Can now debug nodes.

```
$ kubectl alpha debug node/k8s-3m3nhfsvbrix-node-0  
-it --image=ubuntu:focal
```

- Can debug a pod by creating a duplicate of it.

```
$ kubectl alpha debug pod-6856796c4f-227fq  
--copy-to="pod-copy" -it --image=ubuntu:focal
```

End

That's the highlights!