### Kubernetes Cluster Performance, Resource Management, and Cost Impact.

**YOUR HOSTS** 





Elijah Oyekunle

Hasham Haider



### Replex

#### **Cloud Native Governance and Cost Control**







40 % of instances are one to two sizes bigger than needed for their workloads

# 50-75 % of the spend on these instances is wasted



### But we are using K8s and containers!

K8s introduces even more complexity

#### Nodes

- Public cloud provider instances or physical machines in data centers
- Nodes offer resources
- Are nodes being utilized optimally?

#### Pods

- Pods consume resources
- Resource requests and limits
- Are pods/containers using allocated resources efficiently?





\$14 billion wasted in public cloud on idle or unused resources and overprovisioning





### **Cluster Performance?**

#### **Cluster Utilization**

- How efficiently is the underlying cluster infrastructure being used?
- Utilization of Nodes/instances
- Underprovisioned/overprovisioned?

#### Workload efficiency

- Are containers/pods using requested resources efficiently?
- Underprovisioned/overprovisioned?

#### **Idle resources**

- Are idle resources being monitored to minimize wasted spend?

#### Monitor Cluster Performance

- Prometheus
- Grafana







Pod 4 starts to exceed CPU allocation

#### Pod 1

Requests: 1 CPU Limits: 2 CPU

## Pod 2

Requests: 1 CPU Limits: 2 CPU

#### Pod 3

Requests: 1 CPU Limits: 2 CPU

#### Pod 4

Requests: 1 CPU Limits: 2 CPU

### NODE 1 4 vCPUs

16 GiB









Pod 5

Requests: 1 CPU Limits: 2 CPU

> NODE 2 2 vCPUs

4 GiB

#### Node 2 utilization: 50%



Pod 1 Requests: 1	Pod 2 Requests: 1	Pod 3 Requests: 1
Usage: 0.2	Usage: 0.2	Usage: 0.2
Pod 4	Pod 5	Pod 6



Pod 1	Pod 2	Pod 3
Requests: 1 Limits: 1 Usage: 0.2	Requests: 1 Limits: 1 Usage: 0.2	Requests: 1 Limits: 1 Usage: 0.2
De d 4		
P00 4	Pod 5	Pod 6

Pod CPU utilization: 20%



Pod 1 Requests: 0.3 Limits: 1 Usage: 0.2	Pod 2 Requests: 0.3 Limits: 1 Usage: 0.2	Pod 3 Requests: 0.3 Limits: 1 Usage: 0.2
Pod 4 Requests: 0.3 Limits: 1 Usage: 0.2	Pod 5 Requests: 0.3 Limits: 1 Usage: 0.2	Pod 6 Requests: 0.3 Limits: 1 Usage: 0.2

#### Pod CPU utilization: 75%



Pod 1	Pod 2	Pod 3
Requests: 0.3 Limits: 1 Usage: 0.2	Requests: 0.3 Limits: 1 Usage: 0.2	Requests: 0.3 Limits: 1 Usage: 0.2
Pod 4	Pod 5	Pod 6

Node CPU utilization: 15% Pod CPU utilization: 75%



Pod 1 Requests: 0.3 Limits: 1 Usage: 0.2	Pod 2 Requests: 0.3 Limits: 1 Usage: 0.2	Pod 3 Requests: 0.3 Limits: 1 Usage: 0.2	Node CPU util Pod CPU utiliz
Pod 4	Pod 5	Pod 6	
Requests: 0.3	Requests: 0.3	Requests: 0.3	
Limits: 1	Limits: 1	Limits: 1	
Usage: 0.2	Usage: 0.2	Usage: 0.2	

#### Node1: 8 vCPUs, 32 GiB



zation: 15

Pod 1	Pod 2	Pod 3
Requests: 0.3 Limits: 1 Usage: 0.2	Requests: 0.3 Limits: 1 Usage: 0.2	Requests: 0.3 Limits: 1 Usage: 0.2
Pod 4	Pod 5	Pod 6
1001	1005	rouo

Node CPU utilization: 60% Pod CPU utilization: 75% Savings??

replex

Node1: 2 vCPUs, 4 GiB

Туре	vCPUs	RAM	Monthly Cost (eu-central-1)
c5.4xlarge	16	32 GB	\$ 568
m5.4xlarge	16	64 GB	\$ 673
r5.2xlarge	8	64 GB	\$ 445





Development environments account for 44% of compute spend

Non-production environments do not need to run 24/7



### **Isolate Development Environments**

Create separate namespace for Dev environment

```
{
"apiVersion": "v1",
"kind": "Namespace",
"metadata": {
"name": "development",
"labels": {
"name": "development"
}
}
```



### **Isolate Dev Teams**

Create separate namespaces for individual Dev teams

```
"apiVersion": "v1",
"kind": "Namespace",
"metadata": {
"name": "dev_team_1",
"labels": {
"name": "dev_team_1"
}
}
```



### **Default CPU requests and limits**

#### Default



#### Default.cpu.request

- Default CPU requests for all pods

#### Default.cpu.limit

- Default CPU limits for all pods

apiVersion: v1 kind: LimitRange metadata: name: cpu-limit-range spec: limits: - default: cpu: 1 defaultRequest: cpu: 0.5 type: Container



### **Default Memory requests and limits**

#### Default



#### Default.memory.request

- Default memory requests for all pods

#### Default.memory.limit

- Default memory limits for all pods

apiVersion: v1 kind: LimitRange metadata: name: mem-limit-range spec: limits: - default: memory: 512Mi defaultRequest: memory: 256Mi type: Container



### Max/Min CPU constraints

#### **Limit Ranges**



#### Max.cpu

- Max CPU that can be allocated to individual containers

#### Min.cpu

- Min CPU that can be allocated to individual containers

apiVersion: v1 kind: LimitRange metadata: name: cpu-min-max-demo-lr spec: limits: - max: cpu: "800m" min: cpu: "200m" type: Container



### Max/Min Memory constraints

#### **Limit Ranges**



#### Max.memory

- Max mem that can be allocated to individual containers

#### **Min.memory**

- Min mem that can be allocated to individual containers

apiVersion: v1 kind: LimitRange metadata: name: mem-min-max-demo-lr spec: limits: - max: memory: 1Gi min: memory: 500Mi type: Container



### Restrict total resource consumption of Dev Namespace

#### **Resource Quotas**



#### Limits.cpu

Sum of CPU limits for all pods must not exceed this amount

#### Requests.cpu

- Sum of CPU requests for all pods must not exceed this amount



#### Limits.memory

- Sum of mem limits for all pods must not exceed this amount

#### **Requests.memory**

Sum of mem requests for all pods must not exceed this amount

apiVersion: v1 kind: ResourceQuota metadata: name: mem-cpu-demo spec: hard: requests.cpu: "1" requests.memory: 1Gi limits.cpu: "2" limits.memory: 2Gi



### Restrict total resource consumption of Dev Namespace

#### **Resource Quotas**



#### requests.ephemeral-storage

- Across all pods in the namespace, the sum of local ephemeral storage requests cannot exceed this value.

#### limits.ephemeral-storage



 Across all pods in the namespace, the sum of local ephemeral storage limits cannot exceed this value. apiVersion: v1 kind: ResourceQuota metadata: name: mem-cpu-demo spec: hard: requests.cpu: "1" requests.memory: 1Gi limits.cpu: "2" limits.memory: 2Gi



### **Restrict number of API objects in the Dev Namespace**

#### Pod.quota

- max number of pods that can run in a namespace

#### Service.quota

- max number of services that can run in a namespace

#### persistenvolumeclaims.quota

 max number of PVCs that can run in a namespace apiVersion: v1 kind: ResourceQuota metadata: name: object-quota-demo spec: hard: persistentvolumeclaims: "1" services: "2" pod: "150"



### **Restrict number of API objects in the Dev Namespace**

#### loadbalancers.quota

- max number of LBs that can run in a namespace

#### nodeports.quota

- max number of node ports that can run in a namespace

#### replicationcontrollers.quota

- The total number of replication controllers that can exist in the namespace.

apiVersion: v1 kind: ResourceQuota metadata: name: object-quota-demo spec: hard: replicationcontrollers: "1" services.loadbalancers: "2" services.nodeports: "0"



### **Performance Metrics to Monitor?**



#### **Total Resource Usage**

#### sum by (namespace, pod\_name, container) (rate (

container\_cpu\_usage\_seconds\_total{container!="",container!="POD",image!=""}[5m] ))

e query history				Load time: 81me
namespace_pod_name_container_name:container_cpu_usage_seconds_to	tal:sum_rate)			Resolution: 14s Total time series: 1
- insert metric at cursor - 🔻				
Console				
- 1h + Until >	Res. (s) O stacked			
	Tue, 08 Jan 2019 16:05:28 GMT			
Λ_	Tue, 08 Jan 2019 16:05:28 GMT value: 0.11618622282222214			
ں لیے				
		Π		
			no NEVY-	_ /
05 4				
			00	
16:00		16:15	16:30	16:45

#### **Memory Utilization**

# sum(node\_memory\_MemAvailable\_bytes) by (node) / sum(node\_memory\_MemTotal\_bytes) by (node)

Prometheus Alerts Graph Status + Help	
O Enable query history	
sum(namespace_pod_name_container_name:container_cpu_usage_seconds_total:sum_rate)	Load time: 81ms Resolution: 14s
Execute - insert metric at cursor -	Total and series. I
Graph Console	
- 1h	
0.115	
16:00 16:15 16:30	16:45 Remove Graph

plex

#### **CPU Utilisation**

#### (1 - avg(rate(node\_cpu\_seconds\_total{mode="idle"}[5m]))) \* 100

Prometheus Alerts Graph Status <del>-</del> Help		
Enable query history		
:node_cpu_utilisation:avg1m *100		Load time: 44ms Resolution: 14s
Execute - insert metric at cursor - 🔹		Iotai time series: 1
Graph Console		
Element	Value	
0	6.024999999844706	
		Remove Graph
Add Graph		l



### Pre-Built Grafana Dashboard to Monitor Cluster Performance





Kubernetes Optimization Report								
AvgMax MaxAvg MaxMax								
AWS 100% \$19,352.30 GCP 0% \$0 Azure 0% \$0		<sup>Costs</sup> \$19,352.30	Potential Costs \$7,212.96	Potential Savings \$12,139.34		2018 ~	November	
Details								
Cluster Name	Cloud	Existing			Replex Recor	nmended		
apse1a.raas.k8s.realmlab.net		Total Nodes:	Efficiency:	Cost:	Total Nodes:	Total Cost:	Saved:	
		U	50.00 %	\$2,002.90	-	\$1,434.40	\$1,540.50	
euc1a.raas.k8s.realmlab.net		Total Nodes:	Efficiency:	Cost:	Total Nodes:	Total Cost:	Saved:	
		8	19.41%	\$2,795.04	4	\$1,154.88	\$1,640.16	
usw2a raas k8s realmlah net		Total Nodes:	Efficiency:	Cost:	Total Nodes:	Total Cost:	Saved:	
		12	21.27%	\$3,877.20	6	\$1,696.32	\$2,180.88	

replex

< Finance R	leport							
Payment	• App •							
0	<ul> <li>AWS 74% \$7,645.32</li> <li>GCP 26% \$2,901.59</li> <li>Azure 0% \$0</li> </ul>	τα \$	otal Costs 1 <b>0.546.9</b> 1			Year: 2018	Month: ~ November	
Team Name	•	Pods ♦	EC2 Costs 🔶	GCP Costs 🔹	Azure Costs 🔹	% of Costs ◆	Download C	sv Trend
•	Team Saturn	8	\$5.70	\$3.10	\$0	\$0,08%	\$8.80	
•	Team Comet	9	\$1,534.18	\$0	\$0	\$14.55%	\$1,534.18	
• 🥥	Team Pluto	4	\$0.72	\$0.10	\$0	\$0.01%	\$0.82	

### Questions

Learn more: *replex.io/blog* 

