

# Cloud container orchestration with Azure Kubernetes Service (AKS) – MySQL + WordPress

## Kubernetes: An Overview

Also known as K8s or kube, Kubernetes is an open source project written in the Go language and originally developed by Google. Currently maintained by the Cloud Native Computing Foundation, Kubernetes has management capabilities that enable Docker container orchestration, self-healing, restarting, replication, and scaling.

As with other popular technologies today, Kubernetes also has a command-line tool: kubectl, a utility that allows instructions for managing clusters and containers to be executed.

The different structures controlled via Kubernetes will be created from YAML format files and by executing commands via kubectl. A testing environment can be made available with the installation of Minikube (software that enables the creation of a local cluster for use by Kubernetes).

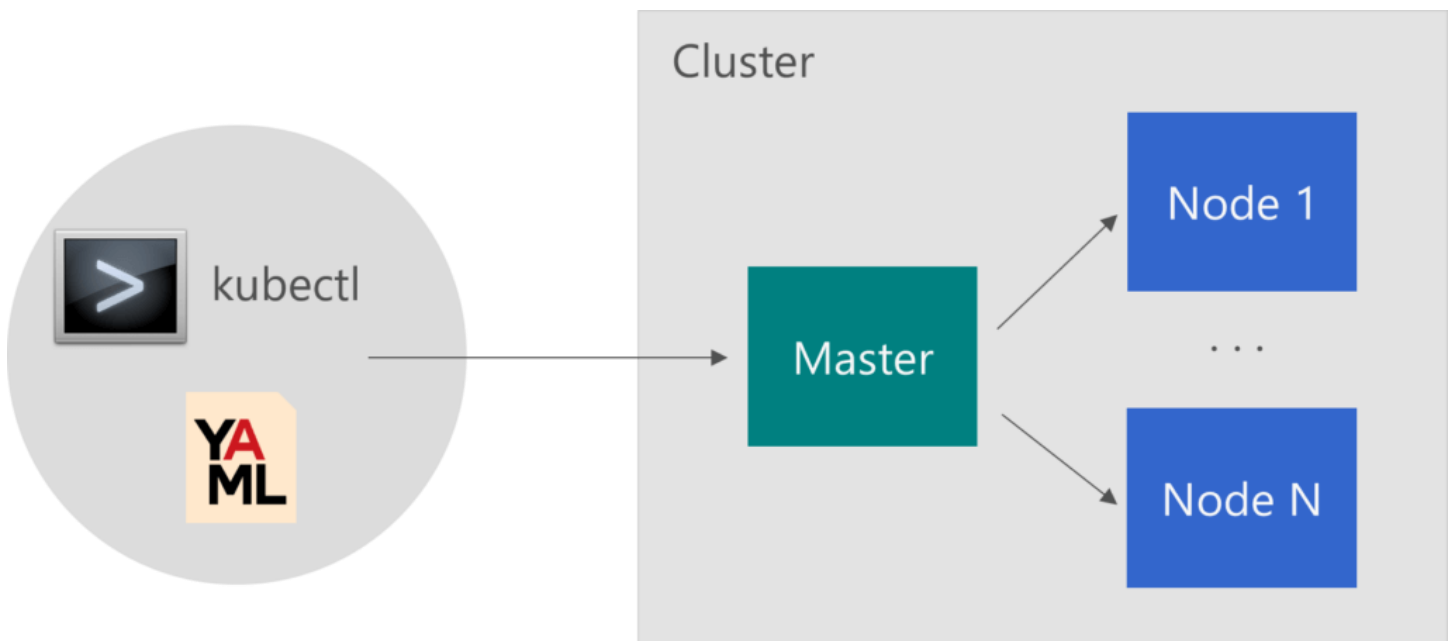
Within the architecture adopted by Kubernetes the following elements should be highlighted:

**Master;**  
**Nodes;**  
**Pod;**  
**Deployment;**  
**Service;**  
**Replication Controller;**  
**Kubelet**

A cluster in Kubernetes is divided into:

**Master**, machine that controls and is responsible for assigning tasks to different Nodes;

**Nodes** (nodes), one or more machines that perform the tasks assigned by the Master.



Hi Tim you gave me free will to build some so for that infrastructure I will use **Azure CLI (Bash)** and build everything through command line. My goal is show some start to have a high demand in the market and imagination is the limit here. \* I wish the task could be done in raspberry Pi so we could use Debian and try put 150 containers on it running nginx, that will definaly blow some minds.

My commands and YAML files are in my github repository

<https://github.com/Alexrubiolv/Wordpress-MySQL-Kubernetes>

With the az login statement, an Azure account will be logged in to allow Bash cloud interaction (also possible through PowerShell), with this procedure involving opening a browser window and providing credentials of access.

The following statement uses Azure CLI to register the provider, which will allow AKS resource creation and management via the command line:

**az provider register -n Microsoft.ContainerService**

A resource group called TestKubernetes should be created, with it being linked to the West Europe:

**az group create --name TesteKubernetes --location westeurope**

The next command will generate a new AKS (ContagemService) feature with a cluster with two nodes, and will be linked to the group TestKubernetes:

**az aks create --resource-group TestKubernetes --nameService Count --node-count 2 --generate-ssh-keys**

After completing this first sequence of tasks, two new resource groups in the Azure portal:

[TestKubernetes](#) and [MC\\_TesteKubernetes\\_ContagemService\\_westeurope](#)

<input type="checkbox"/>	 MC_TesteKubernetes_ContagemService_westeurope	Free Trial	West Europe
<input type="checkbox"/>	 TesteKubernetes	Free Trial	West Europe

Microsoft Azure

Search resources, services, and docs (G+)

alexandrerrubio1976@g...  
DEFAULT DIRECTORY

Home > Resource groups > MC\_TestKubernetes\_ContagemService\_westeurope

### MC\_TestKubernetes\_ContagemService\_westeurope

Resource group

Search (Ctrl+)

+ Add Edit columns Delete resource group Refresh Move Export to CSV Assign tags More

Overview

Activity log

Access control (IAM)

Tags

Events

Settings

Quickstart

Deployments

Policies

Properties

Locks

Essentials

Filter by name... Type == all Location == all Add filter

Showing 1 to 6 of 6 records. Show hidden types

No grouping

Name	Type	Location
a5757c9f-06e6-473d-b4a9-cb27fe245ac	Public IP address	West Europe
aks-agentpool-30872158-nsg	Network security group	West Europe
aks-agentpool-30872158-routetable	Route table	West Europe
aks-nodepool1-30872158-vmss	Virtual machine scale set	West Europe
aks-vnet-30872158	Virtual network	West Europe
kubernetes	Load balancer	West Europe

Home > Resource groups > MC\_TestKubernetes\_ContagemService\_westeurope

### MC\_TestKubernetes\_ContagemService\_westeurope

Resource group

Search (Ctrl+)

+ Add Edit columns Delete resource group Refresh Move Export to CSV Assign tags More

Overview

Activity log

Access control (IAM)

Tags

Events

Settings

Quickstart

Deployments

Policies

Properties

Locks

Export template

Cost Management

Deleted resource group NetworkWatcherRG 12:45 AM  
Deleted resource group NetworkWatcherRG

Essentials

Filter by name... Type == all Location == all Add filter

Showing 1 to 10 of 10 records. Show hidden types

No grouping

Name	Type	Location
a5757c9f-06e6-473d-b4a9-cb27fe245ac	Public IP address	West Europe
aks-agentpool-30872158-nsg	Network security group	West Europe
aks-agentpool-30872158-routetable	Route table	West Europe
aks-nodepool1-30872158-vmss	Virtual machine scale set	West Europe
aks-vnet-30872158	Virtual network	West Europe
k8scontagem	Container registry	West Europe
kubernetes	Load balancer	West Europe
kubernetes-ab54e5167fa7511e999f85258b67750f	Public IP address	West Europe

< Previous Page 1 of 1 Next >

The screenshot shows the Azure portal interface for a Virtual Machine Scale Set (VMSS) named 'aks-nodepool1-30872158-vmss'. The breadcrumb navigation at the top indicates the path: Home > Resource groups > MC\_TestKubernetes\_ContagemService\_westeurope > aks-nodepool1-30872158-vmss. The left sidebar contains navigation links for Overview, Activity log, Access control (IAM), Tags, Diagnose and solve problems, Settings, Instances, Scaling, Storage, and Operating system. The main content area displays the following details:

Property	Value
Resource group (change)	MC_TestKubernetes_ContagemService_westeurope
Status	Succeeded
Location	West Europe
Subscription (change)	Free Trial
Subscription ID	9502de68-ef75-43ad-b8c0-b3fe014ab16e
Public IP address	40.119.145.62, 1 more
Virtual network/subnet	aks-vnet-30872158/aks-subnet
Operating system	Linux
Size	Standard_DS2_v2 (2 instances)
Ephemeral OS disk	Not applicable
Autoscaling	Off
Priority	Regular
Tags (change)	

## Deployment at Kubernetes

For creating objects in the cluster, kubectl, the command line utility for managing Kubernetes resources, will be employed.

In the next command, Azure CLI will be used to allow kubectl access to the AKS cluster:

**az aks get-credentials --resource-group TesteKubernetes --name ContagemService --overwrite-existing**

The definitions of the Deployment object that will be created will be in a YAML file, the contents of which are made available in the following listing:

- mysql-deployment.yaml
  - wordpress-deployment.yaml
- (Both on my github repo) <https://github.com/Alexrubiolv/Wordpress-MySQL-Kubernetes>

Now we will drop a series of commands to wrap everything

**#secret pass generator for MySQL**

```
cat <<EOF > ./kustomization.yaml
secretGenerator:
- name: mysql-pass
  literals:
  - password=Adminteste1976 (or your password)
EOF
```

#YAML files for MySQL and Wordpress – Deployment – I will download both files

```
curl -LO https://k8s.io/examples/application/wordpress/mysql-deployment.yaml
curl -LO https://k8s.io/examples/application/wordpress/wordpress-deployment.yaml
```

or you can upload the files inside azure shell and use:

```
kubectl create -f /home/alexandre/mysql-deployment.yaml
kubectl create -f /home/alexandre/wordpress-deployment.yaml
```

### **#Deployment command**

```
cat <<EOF >> ./kustomization.yaml
resources:
- mysql-deployment.yaml
- wordpress-deployment.yaml
EOF
```

```
kubectl apply -k ./
```

```
Microsoft Azure Search resources, services, and docs (G+/) alexandrerrubio1976@g...
Bash Your cloud drive has been created in:
Subscription Id: 9502de68-ef75-43ad-b8c0-b3fe014ab16e
Resource group: cloud-shell-storage-westurope
Storage account: csb9502de68ef75x43adxb8c
File share: cs-alexandrerrubio1976-gmail-com-1003200076974d9f

Initializing your account for Cloud Shell...
Requesting a Cloud Shell.Succeeded.
Connecting terminal...

alexandre@Azure:~$
alexandre@Azure:~$ az provider register -n Microsoft.ContainerService
alexandre@Azure:~$ az group create --name TesteKubernetes --location westeurope
{
  "id": "/subscriptions/9502de68-ef75-43ad-b8c0-b3fe014ab16e/resourceGroups/TesteKubernetes",
  "location": "westeurope",
  "managedBy": null,
  "name": "TesteKubernetes",
  "properties": {
    "provisioningState": "Succeeded"
  },
  "tags": null,
  "type": "Microsoft.Resources/resourceGroups"
}
alexandre@Azure:~$ az aks create --resource-group TesteKubernetes --name ContagemService --node-count 2 --generate-ssh-keys
{
  "aadProfile": null,
}
alexandre@Azure:~$ az aks get-credentials --resource-group TesteKubernetes --name ContagemService --overwrite-existing
Merged "ContagemService" as current context in /home/alexandre/.kube/config
alexandre@Azure:~$ cat <<EOF >./kustomization.yaml
> secretGenerator:
> - name: mysql-pass
>   literals:
>     - password=Adminteste1976
> EOF
alexandre@Azure:~$ curl -LO https://k8s.io/examples/application/wordpress/mysql-deployment.yaml
% Total % Received % Xferd Average Speed Time Time Time Current
Dload Upload Total Spent Left Speed
100 185 100 185 0 0 584 0 --:--:-- --:--:-- --:--:-- 583
100 1238 100 1238 0 0 828 0 0:00:01 0:00:01 --:--:-- 6447
alexandre@Azure:~$ curl -LO https://k8s.io/examples/application/wordpress/wordpress-deployment.yaml
% Total % Received % Xferd Average Speed Time Time Time Current
Dload Upload Total Spent Left Speed
100 185 100 185 0 0 936 0 --:--:-- --:--:-- --:--:-- 939
100 1323 100 1323 0 0 1049 0 0:00:01 0:00:01 --:--:-- 7000
alexandre@Azure:~$ cat <<EOF >>./kustomization.yaml
> resources:
> - mysql-deployment.yaml
> - wordpress-deployment.yaml
> EOF
alexandre@Azure:~$ kubectl apply -k ./
```

To finish the task we will check everything

Kubectl get pods

Kubectl get pvc

Kubectl get service

Kubectl get service wordpress (external IP will be your WordPress webpage)

```
alexandre@Azure:~$ kubectl get pvc
```

NAME	STATUS	VOLUME	CAPACITY	ACCESS MODES	STORAGECLASS	AGE
mysql-pv-claim	Bound	pvc-ad5a573e-fa75-11e9-99f8-5258b67750ff	20Gi	RWO	default	5h8m
wp-pv-claim	Bound	pvc-b5521512-fa75-11e9-99f8-5258b67750ff	20Gi	RWO	default	5h8m

```
alexandre@Azure:~$ kubectl get service
```

NAME	TYPE	CLUSTER-IP	EXTERNAL-IP	PORT(S)	AGE
kubernetes	ClusterIP	10.0.0.1	<none>	443/TCP	6h58m
wordpress	LoadBalancer	10.0.243.38	51.105.165.105	80:32561/TCP	5h9m
wordpress-mysql	ClusterIP	None	<none>	3306/TCP	5h9m

```
alexandre@Azure:~$ kubectl get service wordpress
```

NAME	TYPE	CLUSTER-IP	EXTERNAL-IP	PORT(S)	AGE
wordpress	LoadBalancer	10.0.243.38	51.105.165.105	80:32561/TCP	5h9m

```
alexandre@Azure:~$
```

Last part:

## Scaling the application via command line

Kubectl will now be used to scale the application via the command line, with the following statement defining the use of any number of Pods to process requests sent to the test application:

kubectl scale deployment count-deployment --replicas = XX (number)

By executing the commands kubectl get deployment and kubectl get pods, you will now see the resulting Pods (covering the existing + new ones)