

Labs Workbench on Azure

Notes from NDS-765.

Create Trial Account

Azure offers a \$200 free trial: portal.azure.com

Provision Kubernetes Container Service

Followed basic instructions provided in [Get started with a Kubernetes cluster in Container Service](#):

Download the [Azure CLI](#)

```
curl -L https://aka.ms/InstallAzureCli | bash
```

Login via Azure CLI

```
az login
```

Create resource group

```
az group create -n acsrg1 -l "westus"
```

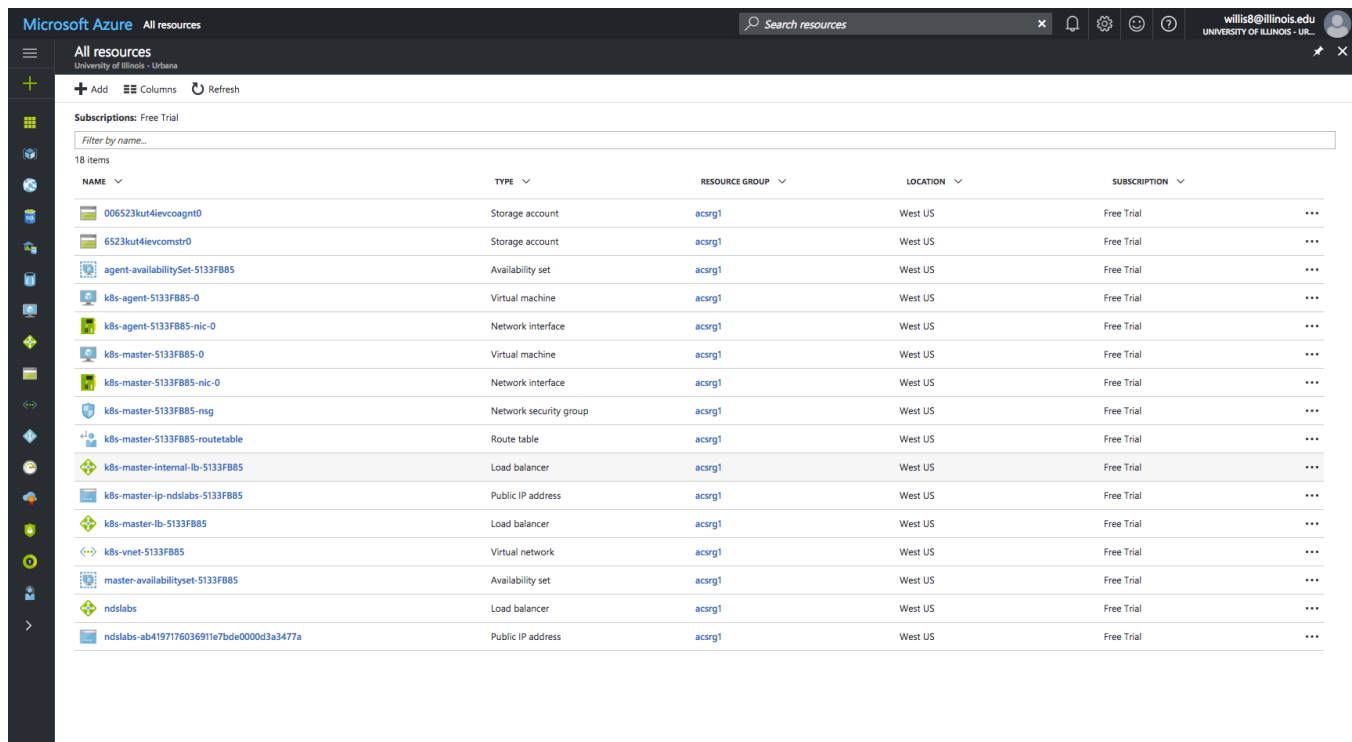
Create Kubernetes cluster. You're only given 4 vcpus in the trial account, so will need to create only 1 master and 1 agent and change the VM size:

```
az aks create --orchestrator-type=kubernetes --resource-group acsrg1 --name=ndslabs --dns-prefix=ndslabs --agent-vm-size=Standard_D1_v2 --master-count=1 --agent-count=1
```

Install the Kubernetes client (creates kube-config profile)

```
az aks kubernetes install-cli
```

At this point you have a running Kubernetes cluster with single master, single compute/agent. Azure provisions a number of resources (storage, network, loadbalancer, etc).



NAME	TYPE	RESOURCE GROUP	LOCATION	SUBSCRIPTION
006523kut4ievcognt0	Storage account	acsrg1	West US	Free Trial
6523kut4ievcognt0	Storage account	acsrg1	West US	Free Trial
agent-availabilitySet-5133FB85	Availability set	acsrg1	West US	Free Trial
k8s-agent-5133FB85-0	Virtual machine	acsrg1	West US	Free Trial
k8s-agent-5133FB85-nic-0	Network interface	acsrg1	West US	Free Trial
k8s-master-5133FB85-0	Virtual machine	acsrg1	West US	Free Trial
k8s-master-5133FB85-nic-0	Network interface	acsrg1	West US	Free Trial
k8s-master-5133FB85-nsg	Network security group	acsrg1	West US	Free Trial
k8s-master-5133FB85-routetable	Route table	acsrg1	West US	Free Trial
k8s-master-internal-lb-5133FB85	Load balancer	acsrg1	West US	Free Trial
k8s-master-ip-ndslabs-5133FB85	Public IP address	acsrg1	West US	Free Trial
k8s-master-lb-5133FB85	Load balancer	acsrg1	West US	Free Trial
k8s-vnet-5133FB85	Virtual network	acsrg1	West US	Free Trial
master-availabilitySet-5133FB85	Availability set	acsrg1	West US	Free Trial
ndslabs	Load balancer	acsrg1	West US	Free Trial
ndslabs-ab4197176036911e7bde000d3a3477a	Public IP address	acsrg1	West US	Free Trial

I was able to make minor modifications to the ndslabs-startup templates to get a basic Labs Workbench running almost immediately. I've captured the hardcoded configuration in my [ndslabs-repo on the azure branch](#).

- Privileged pods:

- works as expected
- Ingress:
 - works as expected
- Wildcard DNS/TLS
 - Wildcard DNS worked fine via Google domains
 - *.azure.ndslabs.org > public IP if ingress ilb
 - TLS should work (disabled during testing)
- Access to Kubernetes API
 - apiserver assumes token path. Added TOKEN_PATH configuration to entrypoint.sh
- Access to etcd:
 - Requires dedicated etcd (etcd.yaml)
- SMTP server
 - Problem: [Azure requires SMTP mail relay](#). This will require code changes
 - Workaround – create accounts via ndslabsctl
- Node labeling:
 - works as expected
- Loadbalancer/Public IP
 - works as expected (via kubectl expose)
- Storage:
 - Broken by default
 - Need to use AzureFile or GlusterFS
 - AzureFile
 - Need to test how/if this works
 - May support quotas, but only through direct API integration
 - GlusterFS requires deploying Gluster cluster
 - • <https://github.com/Azure/azure-quickstart-templates/tree/master/gluster-file-system>
- Addons
 - Access to Kubedash via kubectl proxy
 - Monitoring
 - Heapster installed, but no Grafana by default
 - Logging
 - No Kibana/Fluentd or cluster-wide logging services
 - DNS
 - yes
 - Container registry as separate service
 - <https://azure.microsoft.com/en-us/services/container-registry/>

Creating an [azureFile](#) volume:

- Create secret with azurestorageaccountkey, azurestorageaccountname
- Create PersistentVolume of type azureFile referencing secret
- Attach volume to pod

Activity log:

```
kubectl create -f loadbalancer.yaml
kubectl expose rc nginx-ilb-rc --port=80 --type=LoadBalancer
# Wait for endpoint IP to be assigned
kubectl get sv
kubectl create -f default-ingress.yaml
kubectl create -f default-backend.yaml
kubectl create -f etcd.yaml
kubectl create -f apiserver.yaml
kubectl create -f webui.yaml
kubectl logs ndslabs-apiserver-bmc8h
kubectl delete -f apiserver.yaml
kubectl label nodes k8s-agent-5133fb85-0 ndslabs-node-role=compute
kubectl logs ndslabs-apiserver-qkcgj
kubectl proxy
kubectl get pods --all-namespaces
```

```
willis8-9f6b:templates willis8$ kubectl get pods --all-namespaces
```

NAMESPACE	NAME	READY	STATUS	RESTARTS	AGE
default	default-http-backend-q4s7n	1/1	Running	0	1h
default	ndslabs-apiserver-qkcgj	1/1	Running	1	15m
default	ndslabs-etcd	1/1	Running	0	1h
default	ndslabs-webui-gd0d2	1/1	Running	0	45m
default	nginx-ilb-rc-mr1cb	1/1	Running	4	1h
kube-system	heapster-v1.2.0-1448994189-kx6hs	2/2	Running	0	2h
kube-system	kube-addon-manager-k8s-master-5133fb85-0	1/1	Running	0	2h
kube-system	kube-apiserver-k8s-master-5133fb85-0	1/1	Running	0	2h
kube-system	kube-controller-manager-k8s-master-5133fb85-0	1/1	Running	0	2h
kube-system	kube-dns-v19-7f4b0	3/3	Running	0	2h
kube-system	kube-dns-v19-9bt80	3/3	Running	0	2h
kube-system	kube-proxy-4l6db	1/1	Running	0	2h
kube-system	kube-proxy-zshzj	1/1	Running	0	2h
kube-system	kube-scheduler-k8s-master-5133fb85-0	1/1	Running	0	2h
kube-system	kubernetes-dashboard-696481038-s7hgz	1/1	Running	0	2h
willis8	srf43z-jupytermn-knvw1	1/1	Running	0	10m

```
willis8-9f6b:templates willis8$
```