

HLF 1.4 on AKS

2019년 8월 30일 금요일 오후 1:12

1. Install Python 3.7
\$sudo apt-get update
\$sudo apt-get install -y build-essential libpq-dev libssl-dev openssl libffi-dev zlib1g-dev
\$sudo apt-get install -y python3.7
\$sudo apt-get install -y python3-pip python3-dev python3-venv
\$python3.7 -m pip install --upgrade pip
2. Make sure Python version is 3.7
3. Clone Nephos library from github
\$git clone <https://github.com/Dongbumlee/nephos.git>
\$cd nephos
\$python3 -m venv ./venv
4. Install requirements
\$cd nephos
\$sudo pip install -r requirements.txt
5. Install additional requirements by manually
\$sudo pip install PyYAML --ignore-installed
\$sudo pip install pyasn1-modules --ignore-installed
6. Install Hyperledger Fabric utility binaries (HLF version 1.4.3)
\$cd nephos
\$curl -o ./bootstrap.sh <https://raw.githubusercontent.com/hyperledger/fabric/master/scripts/bootstrap.sh>
\$sudo ./bootstrap.sh -d # -d : don't download docker files on my machine
7. Set PATH for Hyperledger Fabric utility binaries (\nephos\fabric-samples\bin)
\$sudo nano ~/.bashrc

Add PATH info like below and save
export PATH=(your nephos downloaded folder path)/nephos/fabric-samples/bin:\$PATH

Refresh Path
\$source ~/.bashrc

Check whether Path is working
\$cryptogen
usage: cryptogen [] [...]

Utility for generating Hyperledger Fabric key material

Flags:
--help Show context-sensitive help (also try --help-long and --help-man).

Commands:
help [...]
Show help.

generate []
Generate key material

showtemplate
Show the default configuration template

version
Show version information

extend []
Extend existing network
8. Create and set up AKS cluster on Azure
 - a. Create Resource group on your subscription
 - b. Create AKS in your resource group
 - c. Get Credential from K8S cluster
 - i. Install kubectl (<https://kubernetes.io/docs/tasks/tools/install-kubectl/>)
 - ii. Get K8s Credential(<https://docs.microsoft.com/en-us/cli/azure/aks?view=azure-cli-latest#az-aks-get-credentials>)
 - 1) az aks get-credentials --name (your aks cluster name) --resource-group (your resource group aks cluster deployed)
 - iii. Install Helm (<https://github.com/MicrosoftDocs/azure-docs/blob/master/articles/aks/kubernetes-helm.md>)
 - d. Install Ingress Controller on K8S cluster
 - i. Using Helm, install nginx ingress
\$helm install stable/nginx-ingress
 - ii. Check public IP for ingress controller
\$kubectl get svc -A

default	nginx-ingress-controller	LoadBalancer	10.0.102.9	52.231.24.75	80:30892/TCP,443:32690/TCP
default	nginx-ingress-default-backend	ClusterIP	10.0.92.45	<none>	80/TCP
 - iii. Add a A record to your DNS zone, point to the nginx public ip
 - 1) Go resource group for K8S cluster (MC_(resource group name)_(cluster name)_(region name)
 - 2) Select DNS zone
 - 3) Create A record for ingress-controller's public IP

Home > Resource groups > MC_HLF14_kubHLF_koreacentral > 53acc6c77fa94ae09b93.koreacentral.aksapp.io

53acc6c77fa94ae09b93.koreacentral.aksapp.io
DNS zone

Search (Ctrl+/) « + Record set → Move Delete zone Refresh

Overview
Activity log
Access control (IAM)
Tags
Diagnose and solve problems

Settings
Properties
Locks
Export template

Monitoring
Alerts
Metrics

Support + troubleshooting
New support request

Resource group (change) : mc_hlf14_kubhlf_koreacentral
Subscription (change) : Blockchain - Cryptlets EAP
Subscription ID

Tags (change) : Click here to add tags

Name server 1 : ns1-09.azure-dns.com.
Name server 2 : ns2-09.azure-dns.net.
Name server 3 : ns3-09.azure-dns.org.
Name server 4 : ns4-09.azure-dns.info.

You can search for record sets that have been loaded on this page. If you don't see what you're looking for, you can try scrolling to allow more record sets to load.

Search record sets

NAME	TYPE	TTL	VALUE	ALIAS
@	NS	172800	ns1-09.azure-dns.com. ns2-09.azure-dns.net. ns3-09.azure-dns.org. ns4-09.azure-dns.info.	
@	SOA	3600	Email: azuredns-hostmaster.microsoft... Host: ns1-09.azure-dns.com. Refresh: 3600 Retry: 300 Expire: 2419200 Minimum TTL: 300 Serial number: 1	
*	A	3600	52.231.24.75	

- e. Install Certificate Manager on K8S cluster(<https://hub.helm.sh/charts/jetstack/cert-manager>)
\$ kubectl apply -f <https://raw.githubusercontent.com/jetstack/cert-manager/release-0.10/deploy/manifests/00-crds.yaml>
\$ kubectl label namespace cert-manager certmanager.k8s.io/disable-validation="true"
\$ helm repo add jetstack <https://charts.jetstack.io>
\$ helm install --name cm --namespace cm jetstack/cert-manager

- f. Update clusterissuer information and create
 - i. Update email information in /nephos/examples/certManagerCI_production.yaml
spec:acme:email: **your email address**
 - ii. \$kubectl create -f ./examples/certManagerCI_production.yaml

9. Update configuration files in nephos
 - i. Update K8S cluster name in examples/prod/nephos_config.yaml
core:
Cluster: **(your K8S name)**
 - ii. Update ca ingress host information
Ingress:
hosts:
#ca.(your DNS zone name)
- ca.(53acc6c77fa94ae09b93.koreacentral.aksapp.io)
tls:
hosts:
#ca.(your DNS zone name)
- ca.(53acc6c77fa94ae09b93.koreacentral.aksapp.io)

10. Launch installation
\$ PYTHONPATH=. ./nephos/deploy.py --verbose -f ./examples/prod/nephos_config.yaml fabric 2>&1 | tee "hl_fabric_install_prod_\$(date +%Y-%m-%d_%l-%M-%p).log"

- For fresh re-installation :
\$rm -rf examples/prod/crypto/
\$rm -f examples/prod/config/fabric-ca-client-config.yaml
\$helm del --purge ca-pg
\$helm del --purge ca
\$helm del --purge cdb-peer1
\$helm del --purge cdb-peer2
\$helm del --purge kafka-hlf
\$helm del --purge ord1
\$helm del --purge ord2
\$helm del --purge peer1
\$helm del --purge peer2
\$kubectl delete namespace cas
\$kubectl delete namespace orderers
\$kubectl delete namespace peers
\$helm list