1. Az download

curl -sL https://aka.ms/InstallAzureCLIDeb | sudo bash

1. az login
2. az –version
3. az aks create --resource-group rgDivya --name myAKSCluster --node-count 2 --enable-addons monitoring --generate-ssh-keys

location of public and private keys

/home/divya1501/.ssh/id\_rsa' and '/home/divya1501/.ssh/id\_rsa.pub'

We need to mention –generate-ssh-keys without it it will not run

Error :: An RSA key file or key value must be supplied to SSH Key Value. You can use --generate-ssh-keys to let CLI generate one for you

The command completes and returns JSON-formatted information about the cluster.

It takes time to create a cluster.

We create cluster via GUI also.

Till now your kubeconfig file has not been generated.

Ow we need to use kubectl to connect to cluster. And we will download it once we download then we will get the kubeconfig file.

---az aks install-cli

---az aks get-credentials --resource-group myResourceGroup --name myAKSCluster

This command downloads credentials and configures the Kubernetes CLI to use them.

---kubectl get nodes

1. we can create as many cluster we want to create in one VM.

.kube/config will have information of all clusters and will refer to current cluster that we point.

--- az aks get-credentials --resource-group myResourceGroup --name myDivyaCluster

Now kubectl get nodes will refer to the nodes and resources that we are referring.

<https://devopscube.com/setup-jenkins-on-kubernetes-cluster/> -🡪 LOHIT

1. Created pod file for Jenkins
2. No we wish to add slaves to this Jenkins for which we will use kubernetes plugin.and add the slave to our Jenkins

Jenkins -> configure system -> add cloud -> add kubernetes to it

<https://www.oracle.com/webfolder/technetwork/tutorials/obe/oci/configure_jenkins_kubernetes_plugin/configurekubernetesplugin.html>

The Jenkins Kubernetes plug-in uses a Kubernetes cluster to dynamically provision a Jenkins agent using the Kubernetes scheduling mechanisms to optimize the loads, run a single build, and then tear-down that agent.

Pod template :

Image of the container should be a jnlp(Java Network Launching Protocol) image .

${computer.jnlpmac} ${computer.name} mandatory argument to pass in container template.

We have created service for tunnel for port 50000.(Mandatory to have communication b/w master node and kubernetes slave)

JNLP dynamically create pod use it and kill it.

<https://github.com/jenkinsci/kubernetes-plugin>

1. Now we will create our jenkinsfile
   1. Tools section will use when
   2. Add THE GIT Credentials
   3. Trying to create custom workspace

23rd July 2019

Sonarqube into the architecture

We are now wanting to get sonarqube up and running on a pod in k8s cluster.

Sonarqube needs a postgre database .

<https://github.com/coderise-lab/k8s-sonarqube>

first the postgres-deployment must be up and running and see it should be compatible with the sonarqube running.

Changed the image of postgres to “postgres:9.4” else it was not compatible with our sonarqube and also giving an error of

elasticsearch:5.0.0 max virtual memory areas vm.max\_map\_count [65530] likely too low, increase to at least [262144]

solution 1 : sudo sysctl -w vm.max\_map\_count=262144

will apply only for the current session

solution 2: in this file /etc/sysctl.conf

add this line vm.max\_map\_count=262144

and restart ur machine for sure.

Changed the image of sonarqube also since it was giving crashloopback error.

Sonarqube to “sonarqube:7.6-community”

Created a loadbalancer kind of service for sonarqube to access it form outside .

Now since the sonarqube-deployment file had -Dsonar.web.context=/sonar the sonarqube runs

Therefore u will be able to access to sonarqube service as

http://<external ip of service>:<container port>/sonar

remember opening the port on the node to which ur pod has been assigned.

Options to be used with curl :

-o : to download something and save with a different name

-O : to download and save with the same name.

Curl –L

Curl –v

Curl –k

**WEBPAGE STATUS, AND ERROR CODES**

* 100 ranged status codes are informational
* 200 ranged status codes are successful
* 300 ranged status codes represent redirection
* 400 ranged status codes are client errors
* 500 ranged status codes are user errors

A 302 is a temporary redirect – sister to the [301 permanent redirect](https://www.infront.com/blog/the-infront-blog/301-redirects---save-your-search-placment).

A 404 status code unfortunately is somewhat common – it’s a “page not found” error. These appear commonly when a page has been deleted but not redirected.

 Another reason a user may receive a 404 error is a miss type, if the actual URL is Example.com/A and the user types Example.com/B - /B does not exist and therefore cannot be found so the user would receive a 404.

This was the error we were facing

A 403 status code means access is forbidden. It means the request was valid (the webpage exists) but you cannot access it, typically due to permissions.

A 408 status code represents a “time out”. It means the server is taking too long to return a result.

A 500 code represents an internal server issue – this one is BIG! It means your server cannot be reached and therefore your site cannot be accessed.

Intergrating sonarqube with Jenkins and maven

Jenkins->configure system->add ur sonarqube here give a name to it and the server url a server authentication token for the token go to sonarqube .(secret text)

9ec7d8590f22401dbcfd7b5c0ba957ff080060ce

-Dproject.settings=sonar.properties

sh "mvn sonar:sonar -Dsonar.host.url=http://52.172.158.204:9000/sonar/ -Dsonar.projectName=My\_project -Dsonar.profile=newprofile"

we can add various flags to sonar this way

for editing the projectname that would come up on the sonarqube dashboard or the quality profile.

<https://www.devopsschool.com/tutorial/sonarqube/sonarqube-properties.html>

25 July 2019

SONAR SCANNER

<https://docs.sonarqube.org/latest/analysis/scan/sonarscanner/>

<https://community.sonarsource.com/t/sonarqube-in-jenkins-pipeline/1495>

This step is mandatory if you want to trigger any of your SonarQube analyses with the SonarScanner INSTEAD OF MAVEN

<https://docs.sonarqube.org/latest/analysis/scan/sonarscanner-for-jenkins/#AnalyzingwithSonarQubeScannerforJenkins-AnalyzinginaJenkinspipeline>

added sonar scanner in global tool configuration automatic install

JUNIT

JUnit is a unit testing framework for Java programming language.

Testing is the process of checking the functionality of an application to ensure it runs as per requirements. Unit testing comes into picture at the developers’ level; it is the testing of single entity (class or method). Unit testing plays a critical role in helping a software company deliver quality products to its customers.

Mvn test will take all ur files in the src/test folder.

junit 'target/surefire-reports/\*\*/\*.xml'

this will display the reports on the Jenkins UI.

The surefire plugin works with junit to publish reports in html format.

If the maven build succeeded, archive the JUnit test reports for display in the Jenkins web UI.

**Maven Surefire Report Plugin**

The Surefire Report Plugin parses the generated TEST-\*.xml files under ${basedir}/target/surefire-reports and renders them using DOXIA, which creates the web interface version of the test results.

JFROG ARTIFACTORY

JFrog Artifactory is the only Universal Repository Manager supporting all major packaging formats, build tools and CI servers.

Setup for HA jfrog

<https://www.jfrog.com/confluence/display/RTF/HA+Installation+and+Setup>

LOADBALANCING