Docker desktop download

<https://docs.docker.com/desktop/install/windows-install/>

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

A computer screen shot of a diagram

Description automatically generated

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

Configure kuberntes for window

Step 1 : download minikube from https://kubernetes.io/docs/tasks/tools/

Step 2 : download kubectl

Step 3 : Keep both in c driver any folder and set path to environment variable

Step 4 :run below command

1. Minikube version
   1. Minikube start –driver=docker
   2. Minikube status
   3. kubectl cluster-info
   4. kubectl get node
   5. kubectl docker-env
   6. Copy the last line start with @FOR and run
   7. Docker images
   8. Go to your spring boot project folder and create docker image as below
   9. C:\Users\RajeshUpadhyay\Desktop\jenkinsgithub>docker build -t springboot-k8-demo:1.0 .
   10. docker images
   11. kubectl create deployment spring-boot-k8s --image=springboot-k8-demo:1.0 --port=8080
   12. kubectl get deployment
   13. kubectl describe deployment spring-boot-k8s
   14. kubectl get pod
   15. kubectl logs spring-boot-k8s-5884c78774-w9q5f
   16. kubectl get deployment
   17. create service using

kubectl expose deployment spring-boot-k8s --type=NodePort

* 1. kubectl get service
  2. minikube service spring-boot-k8s –url
  3. minikube dashboard
  4. kubectl delete deployment spring-boot-k8s
  5. kubectl delete service spring-boot-k8s
  6. minikube stop

**Deploying Spring Boot on AWS EKS (Elastic Kubernetes Service )**

A diagram of a container

Description automatically generated

A screenshot of a computer

Description automatically generated

Step 1 : Create spring boot project, create Dockerfile, define finalname in pom.xml, build project using mvn clean install then create image as

Step 2 : Docker build -t springbooteks:1.0 .

Step 3 : Install aws client download software using link

Open power shell to admin mode and run below command

Install-Module -Name AWS.Tools.Common

Install-Module -Name AWS.Tools.ECR

Step 4 : Create secret key id and secret key from aws account

And store it in credential file of aws in your machine

C:\Users\RajeshUpadhyay\.aws\credentials

[default]

aws\_access\_key\_id = AKIAXYKJWPELRXNKGQBG

aws\_secret\_access\_key = d5VtnXVg6RoogG+u8cxgxyCCd3pE0f5wjAQc1aWm

step 5 : install aws client on your machine using https://docs.aws.amazon.com/cli/v1/userguide/install-windows.html

step 6 : Goto aws console and search for ECR and create repository

A screenshot of a computer

Description automatically generated

step 7 : open repository and click on view push command

A screenshot of a computer

Description automatically generated

Step 8 : copy all 4 command one by one and run on powershell command from your project as below

PS C:\WINDOWS\system32> (Get-ECRLoginCommand).Password | docker login --username AWS --password-stdin 533267380503.dkr.ecr.us-east-1.amazonaws.com

Login Succeeded

PS C:\WINDOWS\system32> cd\

PS C:\> cd .\Users\

PS C:\Users> cd .\RajeshUpadhyay\

PS C:\Users\RajeshUpadhyay> cd .\Downloads\

PS C:\Users\RajeshUpadhyay\Downloads> cd .\springbooteks\

PS C:\Users\RajeshUpadhyay\Downloads\springbooteks> cd .\springbooteks\

PS C:\Users\RajeshUpadhyay\Downloads\springbooteks\springbooteks> docker build -t rajeshdockerkub .

[+] Building 3.4s (7/7) FINISHED docker:desktop-linux

=> [internal] load build definition from Dockerfile 0.1s

=> => transferring dockerfile: 161B 0.0s

=> [internal] load metadata for docker.io/library/openjdk:8 2.2s

=> [internal] load .dockerignore 0.0s

=> => transferring context: 2B 0.0s

=> [internal] load build context 0.8s

=> => transferring context: 17.62MB 0.8s

=> [1/2] FROM docker.io/library/openjdk:8@sha256:86e863cc57215cfb181bd319736d0baf625fe8f150577f9eb58bd937f5452cb 0.0s

=> CACHED [2/2] ADD target/springbooteks.jar springbooteks.jar 0.0s

=> exporting to image 0.1s

=> => exporting layers 0.0s

=> => writing image sha256:7e7d4d2fb7153d1fd6a81c0174426b2e70f66d78d6f6b058c1774af05fa01242 0.0s

=> => naming to docker.io/library/rajeshdockerkub 0.0s

View build details: docker-desktop://dashboard/build/desktop-linux/desktop-linux/larzln0kg96t69pqsgnv36gaq

What's next:

View a summary of image vulnerabilities and recommendations → docker scout quickview

PS C:\Users\RajeshUpadhyay\Downloads\springbooteks\springbooteks> docker tag rajeshdockerkub:latest 533267380503.dkr.ecr.us-east-1.amazonaws.com/rajeshdockerkub:latest

PS C:\Users\RajeshUpadhyay\Downloads\springbooteks\springbooteks> docker push 533267380503.dkr.ecr.us-east-1.amazonaws.com/rajeshdockerkub:latest

The push refers to repository [533267380503.dkr.ecr.us-east-1.amazonaws.com/rajeshdockerkub]

681c0106f2d4: Pushed

6b5aaff44254: Pushed

53a0b163e995: Pushed

b626401ef603: Pushed

9b55156abf26: Pushed

293d5db30c9f: Pushed

03127cdb479b: Pushed

9c742cd6c7a5: Pushed

latest: digest: sha256:261e86723a9967bcf9a45dbfab06a75948c34d3c83ace41b9c7e114c2a6c0e57 size: 2007

PS C:\Users\RajeshUpadhyay\Downloads\springbooteks\springbooteks>