# Developing locally using VM for docker

Setting Up VirtualBox and Docker : refer to Sumit’s file

**Shared Folder**

Create a shared folder for virtualbox, and share it according to other documentation

C:\myworkspace 🡨 where my projects will be

**Code and Build locally on your PC**

1. Your favorite ID
2. I used OpenJDK8 as I wasn’t able to get Java SDK
3. Installed Maven for building.
4. Build the artifact:

>> c:\myworkspace\some-project\build\target\hello.jar

**In your VM:**

1. Login to harbor:

docker login docker.mspx.spratingsvpc.com

1. Build & Push
2. cd to your /media/sf\_myworkspace/some-project 🡨 this is my share folder
3. build the docker image, args passed in b/c my docker file asks for JAR\_FILE as an ARG

docker build . -t docker.mspx.spratingsvpc.com/isg/hello:v1 --build-arg JAR\_FILE=build/target/hello.jar

1. Optional: test your container in the vm

docker run –t docker.mspx.spratingsvpc.com/isg/hello:v1 –p 8080:8080

1. push the image to our private repo

*docker push docker.mspx.spratingsvpc.com/isg/hello:v1*

**In your VM or back in your PC:**

* whichever env it is , makes sure you setup to access kubernetes

get kubectl for your env:

<https://docs.aws.amazon.com/eks/latest/userguide/install-kubectl.html>

get aws eks authenticator :: this helps kubectl authenticate to eks

<https://docs.aws.amazon.com/eks/latest/userguide/install-aws-iam-authenticator.html>

Assume you’ve installed python and installed necessary libs like awscli, boto3, lxml etc

* login to aws

python STS-TokenGen.py

* configure kubectl (do this only once)

aws eks --region us-east-1 update-kubeconfig --name isg-eks-poc-cluster

Now once you’re logged in you can deploy to knative. This is assuming you’ve already ushed the image to harbor:

1. make sure you have a yaml to describe your knative deployment:

apiVersion: serving.knative.dev/v1

kind: Service

metadata:

name: hello

namespace: default

spec:

template:

spec:

containers:

- image: docker.msbx.spratingsvpc.com/isg/hello:v1

env:

- name: TARGET

value: "My Hello Svc"

1. run the deploy command

kubectl apply –filename service.yaml

1. now check with these commands:

kubectl get ksvc 🡨 this will list all knative services

kubectl get pods 🡨 see if your pods are ready

kubectl get ksvc hello 🡨 this will print out the url to your svc.

All done.