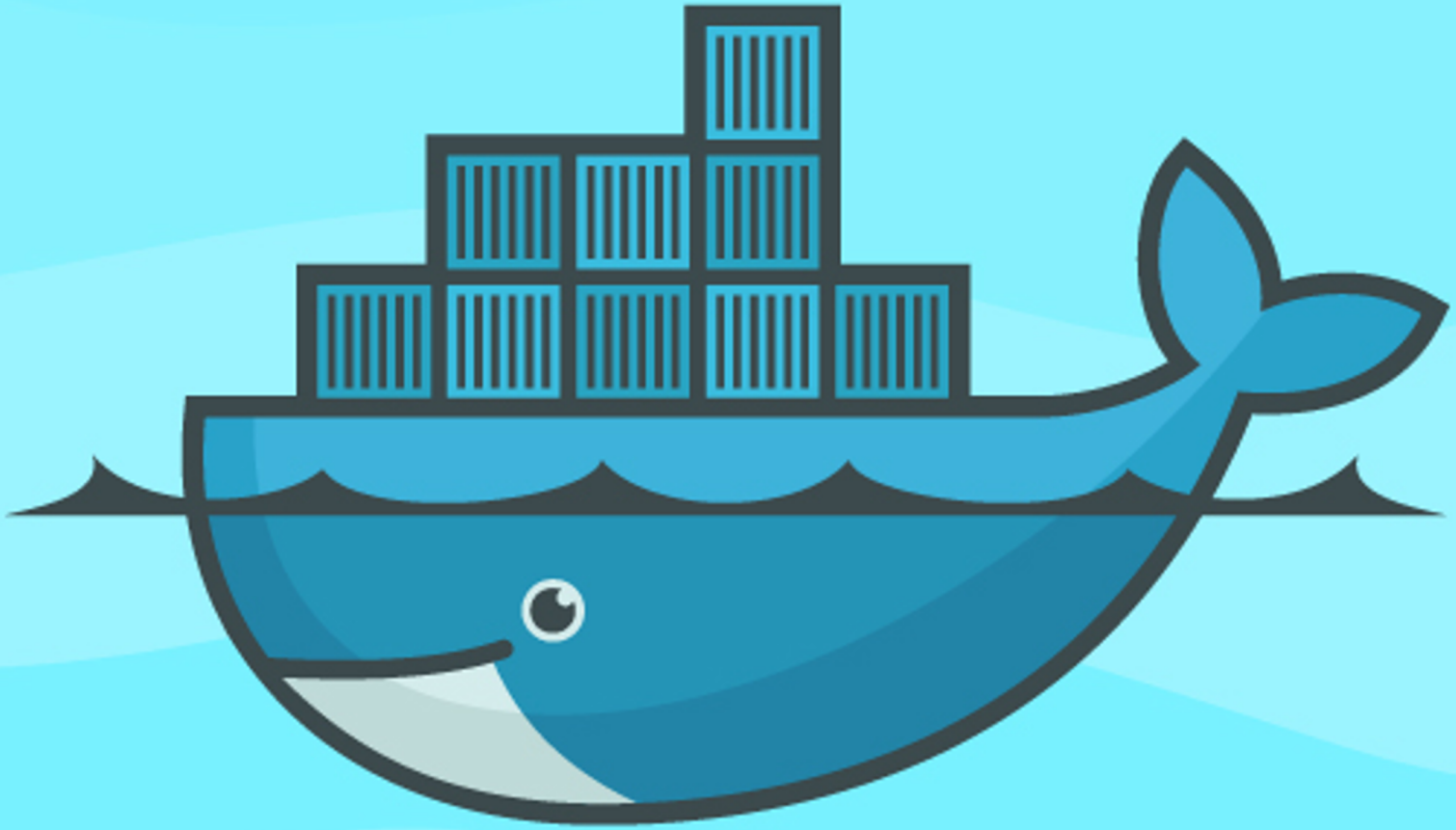
EGT DEVOPS TASK



Solution
by
Grigor Ivanov

1. Do a presentation on building CI and CD solution for the application, note the fact that we are deploying on Bare-Metal, thus the deployment should happen with Ansible or BASH

Performed Steps:

- Created Dockerfiles
  - SpringBootApi :”api/api.Dockerfile”
  - NodeJS:"web/web.Dockerfile”
- Created Docker-Compose YML
- Created CI/CD using Jenkins Pipeline.

Please refer to the “Jenkinsfile" located in the main github folder.

On each build we achieve:

- Containerization of all components
- Creating MySQL db container
- Building the backend using maven
- Building the web part using npm
- Building docker images from the created dockerfiles
- Publishing the docker images to GitHub
- Deploying the application

Resources:

- GitHub Repo:
https://github.com/mortosss/egt_devops_task
- Jenkins Job:
http://212.5.153.24:9898/job/EGT_DevOps_CICD/
- The Deployed App:
http://212.5.153.24:3000
- DockerHub Builded Images:
  - NodeJS:
  https://hub.docker.com/repository/docker/mortos/nodejsweb
  - SpringBoot API:
  https://hub.docker.com/repository/docker/mortos/springbootapi

2. Add in the presentation some info on implementing monitoring solution to servers, where the application will be deployed

Concept Proposal:

- Using OpsGenie:
  - Setup HeartBeat for receiving curl requests
  - Setup CI/CD pipeline or CronJob for sending the curl requests

- The Goal
  - While there is a succesful curl request to the api, the heartbeat we will not raise an alarm, otherwise an alarm is going to be raised in OpsGenie

3. Prepare a running Docker-Compose recipe for local development environment

Prerequisites:

- Installed Docker
- Installed Docker-Compose

Start a Local deployment:

- Clone the github repo
- Perform "*docker-compose up –d*" in the main repo folder
- Check the deployed application in http://localhost:3000

4. Prepare a sample workflow with Jenkins, using groovy on building the application and storing its artifacts to either Nexus NPM and Maven repositories

Sample WorkFlow for publishing Maven:

- Prerequsites
  - Install Maven Plugin
  - Install NPM-Cli-Login Plugin
- Stages in Jenkins Pipeline:
  - Build
  - Unit Tests
  - Deploy

Please refer to ‘cicd’ folder in the github repo for checking example pipelines

Docker-Compose Guide for Dev/QAs

- Prerequisites:
  - Installed Docker
  - Installed Docker-Compose

- Start a Local deployment:
  - Clone the github repo
  - Perform "*docker-compose up –d*" in the main repo folder
  - Check the deployed application in http://localhost:3000