

DevOps-21IS7PEDVR

DevOps Course Code: 21IS7PEDVR

Updated: Sept 2024

Hardcopy Page



Table of Contents

DevOps-21IS7PEDVR	1
Course Summary	4
Course Approach	4
Introduction	5
What is DevOps	5
Why Learn DevOps	6
Quick Gyan	6
DevSecOps Tools	8
Installations	9
Source Code Reference	11
Lab-1: Dockers	12
Exercise-1: Run nginx web-server inside a docker and connect to it.	12
Exercise-2: Create a python based web server and run as a container	17
Introduction to Kubernetes	20
What is Kubernetes	20
Key Features of Kubernetes	20
MiniKube	20
Lab-2: Kubernetes with minikube	21
Exercise-2.1: Installation of miniKube	21
Exercise-2.2: Create a minikube cluster	21
Jenkins	22
Start Jenkins	22
Stop Jenkins	22
Restart Jenkins	22
Status Check	22
Unlocking Jenkins	22
Session 3	23
Preparation Steps	23
Install Jenkins	23
Install Git	23
Create App “Dewdrop” - User Registration Web Service	23
Write Code	23
Create Repo in Github	24

Clone Repo	24
Create source code file	25
Add code to repo	25
Check status of git	25
Commit to Repo	25
Push to Repo	25
Create first Jenkins job	25
Nexus Repository	27
Access Nexus UI	27
Kubernetes using Minikube	29
What is Kubernetes	29
Python Web Server	31
Lesson Plan	31
Lesson Progress	32

Course Summary

Course Title	DEVOPS				
Course Code	21IS7PEDVR	Credits	2	L-T-P	0-0-2
CIE	50 Marks	SEE	100 Marks (50% Weightage)		
Contact Hours / Week	4	Total Lecture Hours		48	

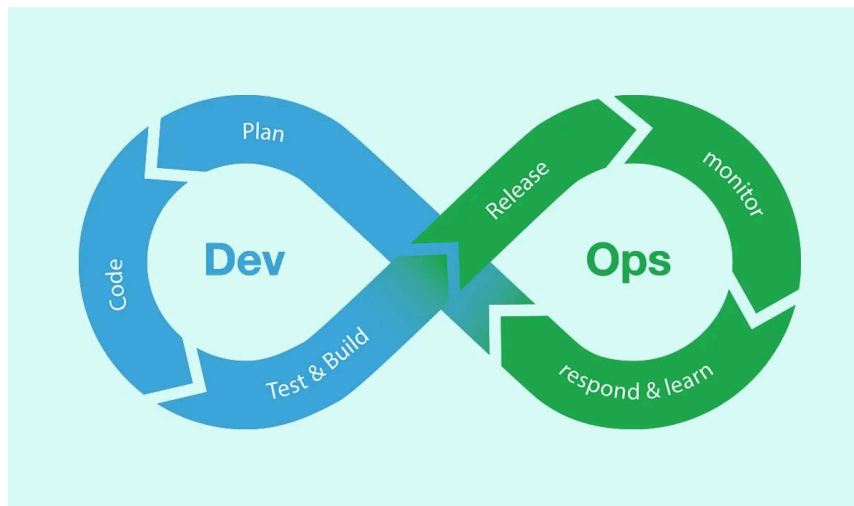
Course Approach

DevOps is a lab only course intended to impart practical skills of DevOps processes and development. This course involves creation of a software microserver application using Golang and MongoDB as backend techstack and introduces various DevOps tools to create a continuous integration and continuous deployment (CI/CD) pipeline.

Introduction

What is DevOps

DevOps refers to integration of people and processes combining aspects of software development and its deployment operations. It also refers to integration of the development (Dev) and IT operations (Ops) team to ensure rapid pace of software development and delivery to meet business requirements.



DevOps

Development (Dev)	Operations (Ops)	DevOps (Dev + Ops)	DevSecOps (Dev+Security+Ops)
Development of a software / app <ul style="list-style-type: none">• Requirements• Design• Coding• Build• Test• Commit	Deployment of a software / app : <ul style="list-style-type: none">• Prepare infrastructure - Data Center, Cloud• VM, Servers, Docker, Kubernetes• Networking• Release• Monitoring	Process to combine and automate Development and Operations For continuous Integration and Continuous Delivery of software or applications	Including Security controls in development and deployment of software <ul style="list-style-type: none">• Firewalls• API Security• Authentication• Authorization• Encryption• Vulnerability Assessment

--	--	--	--

Why Learn DevOps

In the era of “**Super Apps**” and Technology enabled service delivery, key business viability is faster application delivery that meets the needs of its consumers. This requires core IT functions such system design, software development, infrastructure provisioning, security and monitoring to be put in place in a way that is scalable and automated.

To sustain the rapid pace of application development and deployment, a number of core IT functions such as -

- application or service requirements
- system design
- software development
- testing
- infrastructure provisioning (*platform, compute, storage, databases*)
- Networking
- Security
- Monitoring

These processes have to be put together to enable faster application design, development, deployment and delivery, involving integration of many IT processes

With the explosion of digitization, technology enabled services and cloud native applications, know-how of DevOps processes, skills and tools are in high demand to meet rapid digitization and service delivery to consumers.

Quick Gyan

Super App - A mobile or front-end application designed to provide multiple services such such as e-commerce, transport, payment, marketplace, food, delivery to consumers. The services are often arranged as multiple ‘tiles’ within a single mobile application or provided as a cluster of specific purpose applications.

DevOps vs DevSecOps -- DevSecOps is an extension of DevOps to integrate the needs of Security requirements of application and cybersecurity including - secure coding design, secure network and deployment to ensure application is secure against cyberattacks.

CI/CD - Continuous Integration/Continuous Delivery - Continuous Integration refers to

the process of merging and integrating code created by many developers to a central repository to create a buildable and testable code base. Continuous Delivery refers to the process of delivery of software code releases that is tested, verified for use by customers. CI/CD in general refers to the end-to-end automation of software engineering process from coding, commit, building testing, deployment and release.

DevSecOps Tools

Code Repository	Git, GitHub, GitLab, BitBucket	
Build	Maven, Gradle, Custom scripts	
Image Repository	DockerHub, Nexus, JFrog Artifactory	
Create Dev Pipeline	Jenkins	
Infrastructure	Data Center - Cisco, Vmware, Dell, HP Cloud - AWS, GCP, Azure, Oracle	
Deployment Automation	Virtual Machine, Docker, Kubernetes	
Monitoring	Nagios, Prometheus, Datadog	
Infrastructure Provisioning	Terraform	
Configuration Management	Ansible, Puppet, Chef	
Scripting Language	Python	

Installations

	Virtualbox	Virtualbox	https://www.virtualbox.org/
1	Go	Go Language	https://golang.org/doc/install
2	IDE	Visual Studio	https://code.visualstudio.com/
3	Echo	Echo Web Framework for Go	https://echo.labstack.com/
4	GitHub	Code Repository and Version Control	https://github.com/git-guides/install-git Linux: sudo apt install git
5	Jenkins	Automation tool for software code building, deployment and release	https://www.jenkins.io/doc/book/installing/ Use this script for Linux installation: https://github.com/sandeepacademe/devops/blob/main/scripts/jenkins-installation-script.sh Jenkins will be installed in: /var/lib/jenkins
6	Nexus	Store artifacts (compiled code) code created by build pipeline	Download Linux or Windows binary https://help.sonatype.com/repomanager3/product-information/download After download, refer here for installation details https://help.sonatype.com/repomanager3/installation-and-upgrades/installation-methods

DevOps Lab Exam Prep

Prerequisites for DevOps Lab Exam:

1. Students: Sign Up for GitHub account (<https://github.com/>) and login in with your GitHub credentials
2. Students: Sign Up for DockerHub account (<https://hub.docker.com/>) and login with your DockerHub credentials
3. Software- Install the following software installations on your desktop/laptop. **Check if they are already installed, if not, Refer to the following links and also to Internet/Youtube videos for help on installations.**
 - a. Java JDK - https://download.oracle.com/java/17/latest/jdk-17_windows-x64_bin.msi
 - b. Git - <https://github.com/git-for-windows/git/releases/download/v2.39.1.windows.1/Git-2.39.1-64-bit.exe>
 - c. Python - <https://www.python.org/ftp/python/3.10.9/python-3.10.9-amd64.exe>
 - d. VirtualBox -
 - i. Note there is dependency of C++ package for VirtualBox. Install this first from this link: [:Latest supported Visual C++ Redistributable downloads | Microsoft Learn](#)
 - ii. Install VirtualBox after completing previous step
<https://download.virtualbox.org/virtualbox/7.0.6/VirtualBox-7.0.6-155176-Win.exe>
 - e. WSL2 for Windows - <https://learn.microsoft.com/en-us/windows/wsl/install-manual>
 - f. Dockers - <https://desktop.docker.com/win/main/amd64/Docker%20Desktop%20Installer.exe>
 - g. Jenkins - <https://www.jenkins.io/download/thank-you-downloading-windows-installer-stable>
 - h. MiniKube for Windows - <https://storage.googleapis.com/minikube/releases/latest/minikube-installer.exe>
 - i. Note: While installing minikube for Windows, you may get a warning from Windows Defender that it is an unsafe application. This is a known issue, see thread: [windows defender blocks minikube-installer.exe · Issue #7979 · kubernetes/minikube · GitHub](#), for now it is safe to override and install it.
4. Do restart your system post installations or restart any command prompt terminals to ensure Path and other environment variables are set.

More References:

- How to create a Github account - <https://www.youtube.com/watch?v=QUtk-Uuq9nE>
- How to create a Docker Hub account - https://www.youtube.com/watch?v=h2Lwo7SD_pk
- Java JDK Installation (ver 11 or above) - <https://www.youtube.com/watch?v=DTZAz9rj0kU>
- Python Installation (ver 3.9 or above) - <https://www.youtube.com/watch?v=Kn1HF3oD19c>
- Docker Installation - <https://www.youtube.com/watch?v=5nX8U8Fz5S0>
- Jenkins Installation - <https://www.youtube.com/watch?v=MlvNXFdPhB8>
- MiniKube Installation - <https://www.youtube.com/watch?v=TAM-DLPX9XA>
- Jenkins file Tutorial - <https://www.youtube.com/watch?v=RsD2nzPY0is>
- Jenkins <https://youtu.be/pMO26j2OUME>

Source Code Reference

<https://github.com/sandeepacademe/dewdrop>

Lab-1: Dockers

Exercise-1: Run nginx web-server inside a docker and connect to it.

(1) Use docker pull to download images from docker hub - **docker pull nginx**

docker pull nginx

```
Using default tag: latest
latest: Pulling from library/nginx
a603fa5e3b41: Pull complete
c39e1cda007e: Pull complete
90cfefba34d7: Pull complete
a38226fb7aba: Pull complete
62583498bae6: Pull complete
9802a2cfdb8d: Pull complete
Digest:
sha256:e209ac2f37c70c1e0e9873a5f7231e91dcd83fdf1178d8ed36c2ec09974210ba
Status: Downloaded newer image for nginx:latest
```

(2) List docker images - **docker image ls**

docker images ls

REPOSITORY	TAG	IMAGE ID	CREATED	SIZE
REPOSITORY				
CREATED	SIZE			
nginx	latest	88736fe82739	2 weeks ago	142MB

(3) Use docker run to run nginx : Parameters: name: alpha-nginx, host-port:80, container-port: 8080, use detach mode

docker run --add-host=alpha-local:<ipaddress> -p 80:80 nginx

```
a645caf0857f8fd8f16cedf8f4cd1acd9bb2d77fe1e023e90ed6a8ef3273b75e
```

Note: Use ipconfig (Windows) and ifconfig -a (linux) to get your system's ip-address

(4) Check the running instance of the container alpha-nginx

docker container ls

CONTAINER ID	IMAGE	COMMAND	CREATED
a645caf0857f	nginx	"/docker-entrypoint...."	48 seconds ago
STATUS	PORTS		Up
NAMES			
46 seconds	80/tcp, 0.0.0.0:80->8080/tcp		
alpha-nginx			

(5) Login to alpha-nginx container and check status of nginx process inside the container

docker exec -ti alpha-nginx bash

```
root@a645caf0857f:/# ps
bash: ps: command not found

root@a645caf0857f:/# apt-get update
Get:1 http://deb.debian.org/debian bullseye InRelease [116 kB]
Get:2 http://deb.debian.org/debian-security bullseye-security InRelease [48.4 kB]
Get:3 http://deb.debian.org/debian bullseye-updates InRelease [44.1 kB]
Get:4 http://deb.debian.org/debian bullseye/main amd64 Packages [8184 kB]
Get:5 http://deb.debian.org/debian-security bullseye-security/main amd64 Packages [208 kB]
Get:6 http://deb.debian.org/debian bullseye-updates/main amd64 Packages [14.6 kB]
Fetched 8615 kB in 3s (2953 kB/s)
Reading package lists... Done

root@a645caf0857f:/# apt-get install procps
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following additional packages will be installed:
  libgpm2 libncurses6 libncursesw6 libprocps8 psmisc
Suggested packages:
  gpm
The following NEW packages will be installed:
  libgpm2 libncurses6 libncursesw6 libprocps8 procps psmisc
0 upgraded, 6 newly installed, 0 to remove and 4 not upgraded.
Need to get 1034 kB of archives.
After this operation, 3474 kB of additional disk space will be used.
Do you want to continue? [Y/n] y
Get:1 http://deb.debian.org/debian bullseye/main amd64 libncurses6 amd64 6.2+20201114-2 [102 kB]
Get:2 http://deb.debian.org/debian bullseye/main amd64 libncursesw6 amd64 6.2+20201114-2 [132 kB]
```

```

Get:3 http://deb.debian.org/debian bullseye/main amd64 libprocps8 amd64
2:3.3.17-5 [63.9 kB]
Get:4 http://deb.debian.org/debian bullseye/main amd64 procps amd64
2:3.3.17-5 [502 kB]
Get:5 http://deb.debian.org/debian bullseye/main amd64 libgpm2 amd64 1.20.7-8
[35.6 kB]
Get:6 http://deb.debian.org/debian bullseye/main amd64 psmisc amd64 23.4-2
[198 kB]
Fetched 1034 kB in 0s (3238 kB/s)
debconf: delaying package configuration, since apt-utils is not installed
Selecting previously unselected package libncurses6:amd64.
(Reading database ... 7823 files and directories currently installed.)
Preparing to unpack .../0-libncurses6_6.2+20201114-2_amd64.deb ...
Unpacking libncurses6:amd64 (6.2+20201114-2) ...
Selecting previously unselected package libncursesw6:amd64.
Preparing to unpack .../1-libncursesw6_6.2+20201114-2_amd64.deb ...
Unpacking libncursesw6:amd64 (6.2+20201114-2) ...
Selecting previously unselected package libprocps8:amd64.
Preparing to unpack .../2-libprocps8_2%3a3.3.17-5_amd64.deb ...
Unpacking libprocps8:amd64 (2:3.3.17-5) ...
Selecting previously unselected package procps.
Preparing to unpack .../3-procps_2%3a3.3.17-5_amd64.deb ...
Unpacking procps (2:3.3.17-5) ...
Selecting previously unselected package libgpm2:amd64.
Preparing to unpack .../4-libgpm2_1.20.7-8_amd64.deb ...
Unpacking libgpm2:amd64 (1.20.7-8) ...
Selecting previously unselected package psmisc.
Preparing to unpack .../5-psmisc_23.4-2_amd64.deb ...
Unpacking psmisc (23.4-2) ...
Setting up libgpm2:amd64 (1.20.7-8) ...
Setting up psmisc (23.4-2) ...
Setting up libncurses6:amd64 (6.2+20201114-2) ...
Setting up libncursesw6:amd64 (6.2+20201114-2) ...
Setting up libprocps8:amd64 (2:3.3.17-5) ...
Setting up procps (2:3.3.17-5) ...
Processing triggers for libc-bin (2.31-13+deb11u5) ...

root@a645caf0857f:/# ps
  PID TTY          TIME CMD
   34 pts/1        00:00:00 bash
  386 pts/1        00:00:00 ps

root@a645caf0857f:/# ps -eaf
UID          PID    PPID  C STIME TTY          TIME CMD
root          1        0  0 17:58 pts/0        00:00:00 nginx: master process nginx
-g daemon off;
nginx        30        1  0 17:58 pts/0        00:00:00 nginx: worker process
nginx        31        1  0 17:58 pts/0        00:00:00 nginx: worker process
nginx        32        1  0 17:58 pts/0        00:00:00 nginx: worker process

```

nginx	33	1	0	17:58	pts/0	00:00:00	nginx: worker process
root	34	0	0	18:02	pts/1	00:00:00	bash
root	387	34	0	18:04	pts/1	00:00:00	ps -eaf

(6) Check if nginx web-service is accessible using browser or curl pointing to http://<localhost>:80

```

curl 192.168.96.1:80
<!DOCTYPE html>
<html>
<head>
<title>Welcome to nginx!</title>
<style>
html { color-scheme: light dark; }
body { width: 35em; margin: 0 auto;
font-family: Tahoma, Verdana, Arial, sans-serif; }
</style>
</head>
<body>
<h1>Welcome to nginx!</h1>
<p>If you see this page, the nginx web server is successfully installed and
working. Further configuration is required.</p>

<p>For online documentation and support please refer to
<a href="http://nginx.org/">nginx.org</a>.<br/>
Commercial support is available at
<a href="http://nginx.com/">nginx.com</a>.</p>

<p><em>Thank you for using nginx.</em></p>
</body>
</html>

```


docker container ls

CONTAINER ID	IMAGE	COMMAND	CREATED	STATUS	PORTS	NAMES
6f88be2ee378	nginx	"/docker-entrypoint...."	5 minutes ago	Up 5 minutes	0.0.0.0:80->80/tcp	beautiful_bohr

docker container stop 6f88be2ee378

6f88be2ee378

docker container ls

CONTAINER ID	IMAGE	COMMAND	CREATED	STATUS	PORTS	NAMES
--------------	-------	---------	---------	--------	-------	-------

Exercise-2: Create a python based web server and run as a container

(1) Download the python code from:

<https://github.com/sandeepacademe/devops/blob/main/py-webserver/py-webserver.py>

Note : this requires python and it's dependencies flask, jsonify, requests to be installed

(2) Create docker file called **my-docker-file**

```
FROM python:3
ADD py-webserver.py /
RUN pip install flask
RUN pip install requests
RUN pip install jsonify
CMD ["python", "./dewdrop.py"]
```

(3) Build docker image

docker build -f .\my-docker-file -t myserver:1 .

```
[+] Building 2.9s (11/11) FINISHED
=> [internal] load build definition from dewdrop.dockerfile
0.1s
=> => transferring dockerfile: 40B
0.0s
=> [internal] load .dockerignore
0.0s
=> => transferring context: 2B
0.0s
=> [internal] load metadata for docker.io/library/python:3
2.6s
=> [auth] library/python:pull token for registry-1.docker.io
0.0s
=> [internal] load build context
0.0s
=> => transferring context: 32B
0.0s
=> [1/5] FROM
docker.io/library/python:3@sha256:10fc14aa6ae69f69e4c953c9fd9b0964843d8c16395
0491d2138af891377bc1d 0.0s
=> CACHED [2/5] ADD dewdrop.py /
0.0s
=> CACHED [3/5] RUN pip install flask
0.0s
=> CACHED [4/5] RUN pip install requests
0.0s
=> CACHED [5/5] RUN pip install jsonify
0.0s
```

```
=> exporting to image
0.1s
=> => exporting layers
0.0s
=> => writing image
sha256:c44489e985849c03311529e05f0b6f7ce9368949a6fc829b0f06efdd574dadee
0.0s
=> => naming to docker.io/library/myserver:1
0.0s
```

Use 'docker scan' to run Snyk tests against images to find vulnerabilities and learn how to fix them

(4) Check if container is built - run docker images to list the built image

```
docker images
REPOSITORY          TAG          IMAGE ID
CREATED             SIZE
myserver            1            c44489e98584    13
days ago           954MB
```

(5) Deploy the container, using docker run

```
docker run --add-host=alpha-local:<ipaddress> -p 80:80 myserver
```

```
docker run -p 127.0.0.1:80:8080/tcp --name myserver myserver:1
* Serving Flask app 'dewdrop'
* Debug mode: on
WARNING: This is a development server. Do not use it in a production
deployment. Use a production WSGI server instead.
* Running on http://127.0.0.1:5000
Press CTRL+C to quit
* Restarting with stat
* Debugger is active!
* Debugger PIN: 291-674-522
```

(6) Check server is running

<https://github.com/sandeepacademe/devops/blob/main/py-webserver/py-websever.py>

Introduction to Kubernetes (K8s)

Refer: <https://www.youtube.com/watch?v=X48VuDVv0do>

What is Kubernetes (abbreviated as “K8s”)

- It's a Container Management (Orchestration) system
- Developed and open sourced by Google (July 2015)
- With increased use of Containers (e.g. LXC, Dockers) for deploying applications, there is a need for managing a large number of Containerized applications.
- Kubernetes is a management tool for deploying, automating, and scaling Containerized applications.
- Kubernetes enables High Availability -it enables Containerized applications to be always available by providing replication and redundancy
- Helps applications to control scaling - Scale Up or Scale Down based on load on the application

Key Features of Kubernetes

- Pods - smallest unit of compute containing **one or more** containers
- Replication Controller
- Storage Management
- Resource Monitoring

Minikube

Minikube is a tool that sets up a light weight Kubernetes environment on a local PC or laptop.

Refer: [minikube start](#) | [minikube \(k8s.io\)](#)

Lab-2: Kubernetes with minikube

Exercise-2.1: Installation of miniKube

Install **minikube** on your laptop/desktop:

1. Install Hypervisor : VirtualBox
 - a. <https://www.virtualbox.org/manual/ch02.html>
2. Install Minikube
 - a. <https://minikube.sigs.k8s.io/docs/start/>
3. Verify Installation
 - a. Verify **kubectl** command to be working

Exercise-2.2: Create a minikube cluster

Refer: <https://minikube.sigs.k8s.io/docs/drivers/virtualbox/>

Commands:

```
minikube start --driver=virtualbox
```

```
minikube config set driver virtualbox
```

Jenkins

What is Jenkins

Jenkins is an open source automation server. It helps automate steps of software development from building code, testing and deploying enabling continuous integration and development.

What is Jenkins pipeline

Jenkins pipeline is a collection of events or jobs executed in a sequence. It is implemented as a combination of plugins to execute various build, test, deploy functions.

What is Jenkinsfile

What are the few reasons for using jenkinsfile?

Why should one use a pipeline?

Explain pipeline terms: pipeline, node, stage, step

explain declarative pipeline with syntax.

explain scripted pipeline with syntax

what is yaml file

explain 4 important key fields in yaml file

Pre-requisties:

JDK Installation version (11 v17 or 21) :

<https://www.oracle.com/java/technologies/downloads/#jdk21-windows>

Exercise 1 - Deploy Jenkins master and agent using Docker images

Create a Jenkins Master Dockerfile

Start Jenkins

```
sudo systemctl start jenkins
```

Stop Jenkins

```
sudo systemctl stop jenkins
```

Restart Jenkins

```
sudo systemctl restart jenkins
```

Status Check

```
sudo systemctl status jenkins
```

Unlocking Jenkins

Connect to : <http://localhost:8080>

To Unlock get password from : `sudo cat /var/lib/jenkins/secrets/initialAdminPassword`

write sample yaml file to create a nginx test pod
what is the purpose of the selector field in yaml file
using what field will you change the number of replicas in deployment yaml?
list 4 minikube kubectl commands along with the description

Session 3

Preparation Steps

Install Jenkins

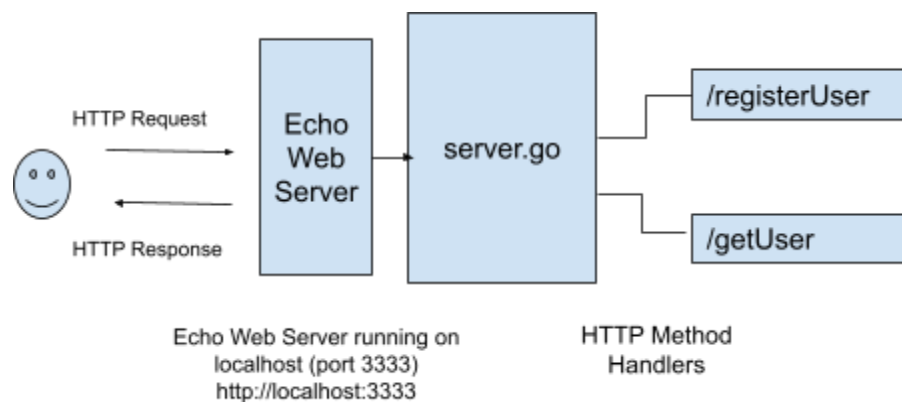
Refer to installation instructions here: <https://www.jenkins.io/doc/book/installing/>

Install Git

Refer to installation instructions here: <https://github.com/git-guides/install-git>

Create App “Dewdrop” - User Registration Web Service

Write Code



Software Architecture

- (a) Create a new director “dewdrop”
- (b) Download server.go from: <https://github.com/sandeepacademe/dewdrop/blob/main/server.go> into the directory
- (c) Run command: **go mod init dewdrop**
- (d) Run command: **go run server.go**
- (e) This results in the following error

```
go run v1-server.go
v1-server.go:21:2: no required module provides package
github.com/labstack/echo/v4; to add it: go get github.com/labstack/echo/v4
```

- (f) Run command to resolve dependencies: **go get github.com/labstack/echo/v4**
- (g) Finally, build and run server.go: **go run server.go**
- (h) This should bring basic Echo Service

2021/11/11 01:02:26 Initialize..

2021/11/11 01:02:26 Executing Initialization tasks

```

  ____  ____
 /  _/  _/  _/
/_  _/  _/  _/
/_  _/  _/  _/ v4.6.1
High performance, minimalist Go web framework
https://echo.labstack.com

_____O/_____
          O\
⇒ http server started on [::]:3333

```

Create Repo in Github

Now that we have a basic web service running, let's commit this code into a code repository. We use <https://github.com> as a code repository. Before we can "push" the code, we need a new repo in a github account.

Here some steps to follow:

- If you don't have a github.com account, goto <https://github.com> and "Sign up" for github account by choosing a username and password and then "Sign in" into github
- Once you are signed in to your github account, create a new repo to push the code. Let's call it "dewdrop"
- To create a new repo, In the upper right corner, click the + sign icon, then choose **New repository**. This will take you to a page where you can enter a repository name (here we give "dewdrop" as repository name), provide a description, and choose to initialize with a README typically used to describe the repo purpose.

Clone Repo

Create a new directory in your local machine and change directory to it.

Using git clone command, clone contents from your github to your local machine.

```
git clone https://github.com/<your-repo-name>/dewdrop.git
```

Create source code file

Using appropriate editor, create a source code file for example, **server.go** and put in the intended code,

Add code to repo

Use the 'git add' command to add individual code to github.

```
git add server.go
```

If there are more than one file, then use "git add -all" command to add all new files

Check status of git

Use the 'git status' command to know the changes performed in the local machine.

```
git add server.go
```

On branch main

Your branch is up to date with 'origin/main'.

Changes to be committed:

(use "git restore --staged <file>..." to unstage)

new file: server.go

Commit to Repo

Use the 'git commit' command to commit code changes to the github repository.

```
git config --global user.email "<your-email-id>"
```

```
git config --global user.name "<your name>"
```

Push to Repo

Use the "git push" command to finally push the change set into the github repository.

Note: You need "personal token" to be used as password while executing 'git push'.

```
git push
```

Create first Jenkins job

Nexus Repository

Access Nexus UI

Once Nexus is installed, it can be started using

```
// Linux
nexus run

// Windows
nexus.exe run
```

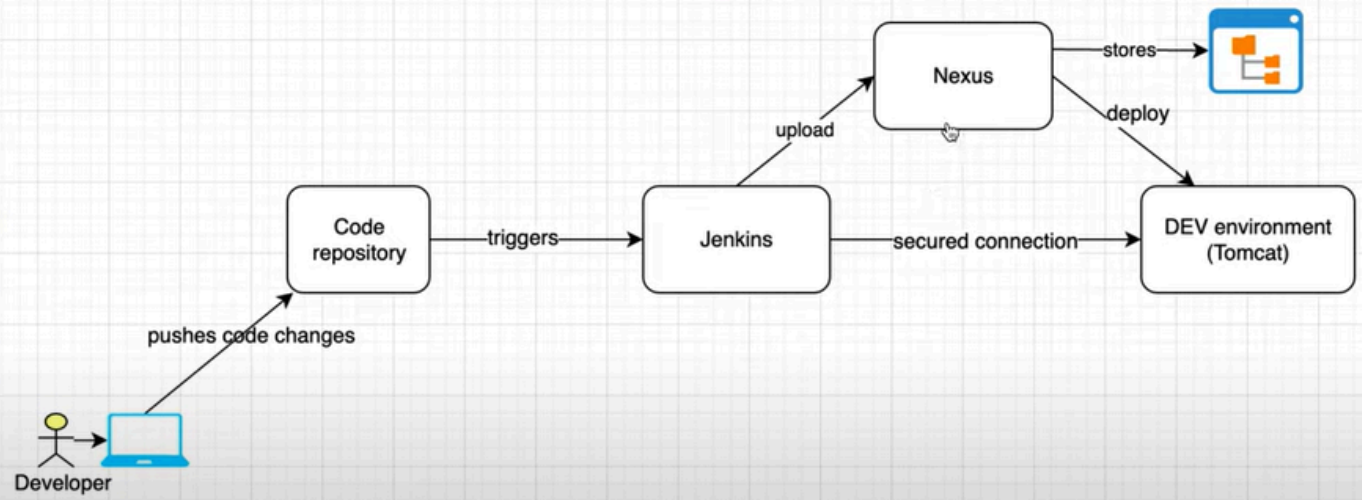
When Nexus starts successfully, it is notified as follows

```
2021-12-11 00:22:10,947+0530 INFO [jetty-main-1] *SYSTEM org.eclipse.jetty.serv
er.Server - Started @112478ms
2021-12-11 00:22:10,948+0530 INFO [jetty-main-1] *SYSTEM org.sonatype.nexus.boo
tstrap.jetty.JettyServer -
-----
Started Sonatype Nexus OSS 3.37.0-01
-----
2021-12-11 00:22:15,582+0530 INFO [qtp2078922547-85] *UNKNOWN org.apache.shiro.
session.mgt.AbstractValidatingSessionManager - Enabling session validation sched
uler...
2021-12-11 00:22:15,685+0530 INFO [qtp2078922547-85] *UNKNOWN org.sonatype.nexu
s.internal.security.anonymous.AnonymousManagerImpl - Using default configuration
: OrientAnonymousConfiguration{enabled=true, userId='anonymous', realmName='Nexu
sAuthorizingRealm'}
```

Once the repository manager is started, access user interface using:

<http://localhost:8081>

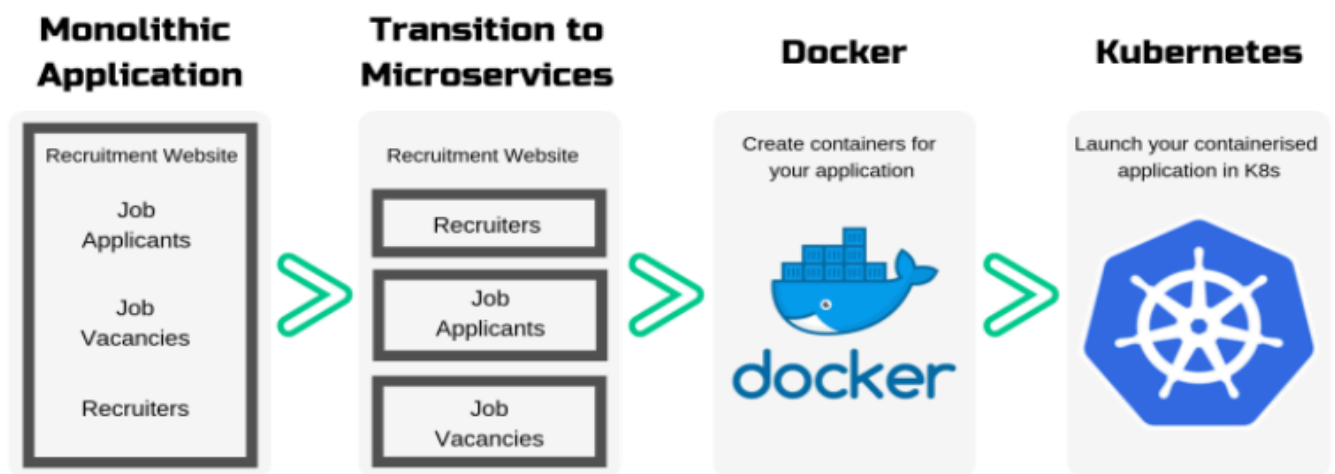
How Nexus Integrate with CI server (Jenkins)?



Kubernetes using Minikube

What is Kubernetes

- With increased use of Containers (e.g. LXC, Dockers) for deploying applications, there is a need for managing a large number of Containerized applications.
- Kubernetes is a management tool for deploying, automating, and scaling Containerized applications.
- It was developed by Google
- Kubernetes enables High Availability -it enables Containerized applications to be always available by providing replication and redundancy
- Helps applications to control scaling - Scale Up or Scale Down based on load on the application



Reference: <https://docs.bytemark.co.uk/article/kubernetes-terminology-glossary/>

Python Web Server

1. Install Python - <https://www.python.org/downloads/>
2. Install Flask, Requests, jsonify

```
pip install flask  
pip install requests  
pip install jsonify
```
3. Install Flask and Requests
4. Install Flask and Requests

Lesson Plan

Session	Date	Topics
1		
2		
3		
4		
5		

Lesson Progress

Se ssi on	Date	Activities Performed	Notes
1			
2			
3			
4			
5			
6			
7			
8			