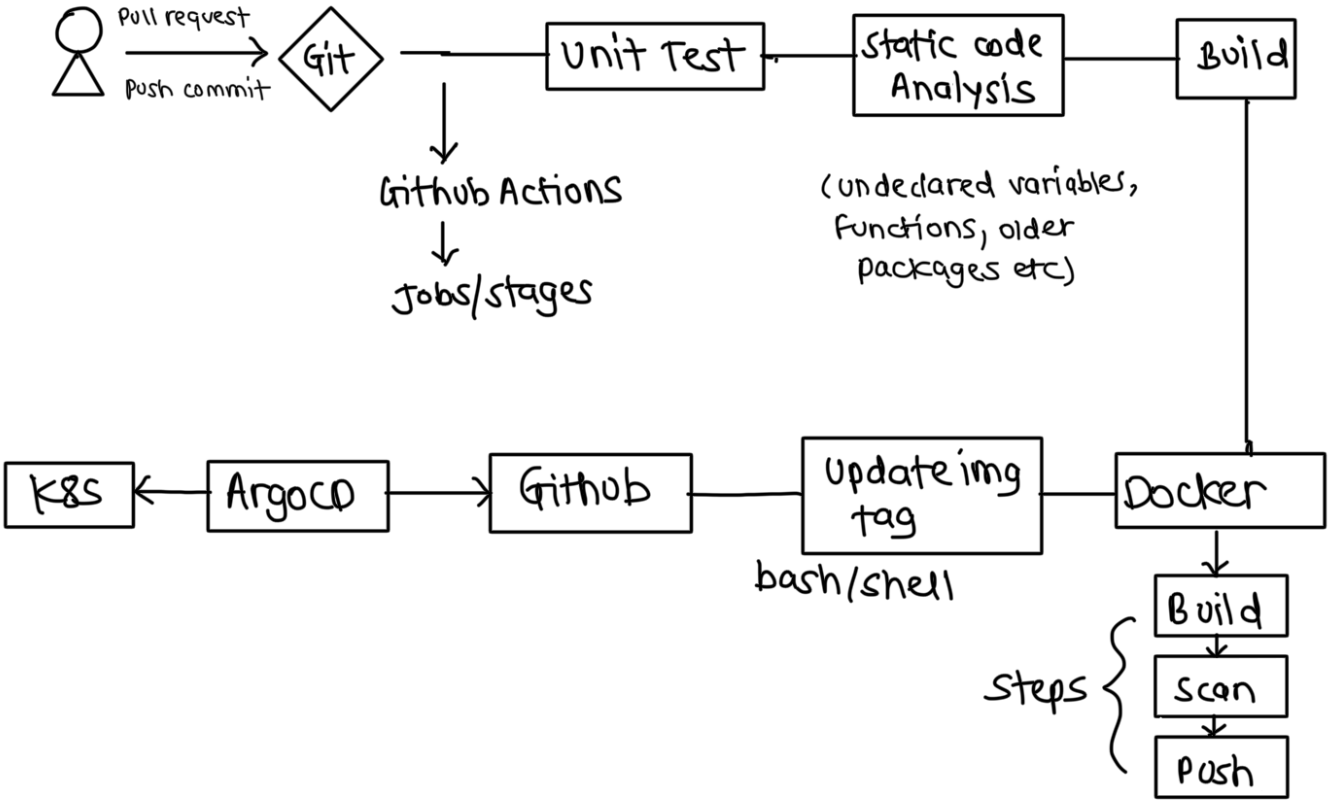
Devsecops CI/CD Pipeline Implementation (TictacToe game)

Devsecops is devops with security mindset

Why we use devsecops ?

1) Most of codes are written with help of AI Assiantance We can not say that code is secure or not because if it is hardcoding the secrets or using older packages or using packages that has critical vulnarabilities.

Project Description:



When devlopers triggers a pull request or push commit to repo Github workflows in this case we are using github actions get triggrs and with within github

workflows there are multiple jobs.(stages) such as unit Test, static code Analysis in static code analysis it checks if there any unused variables, functions, older packages etc When unit testing and SCA done then only it build the application. Onces its sussessful go to next stage i.e. docker within docker there multiple steps such as building the docker image, scanning the image by using trivy and pushing to container registry here we use github container registry we can use docker container registry but gcr is more secure.

When new image is created it then updated in kubernets manifest files like deployment.yml

Once image is updated it is push to github repo using shell-script. Now till this we complete CI part.

Now for CD we use ArgoCD tool what it do? ArgoCD detects whenever image tag is updated and then ArgoCD deploys new image to kubernets cluster.

Project steps:

1> Go to Git repo of Abhishek.veeramalla → Devsecops

2> Install nodejs, npm in your machine (t2.medium)

3) clone repo → switch to repo

4) npm install (It install all the dependency

that are written in package.json)

5) npm run build → it creates all the static files and stores in dist folder

6) Create dockerfile

```
# Build stage
FROM node:20-alpine AS build
WORKDIR /app
 COPY package*.json ./
 RUN npm ci
 COPY . .
 RUN npm run build

# Production stage
FROM nginx:alpine
copy --from=build /app/dist /usr/share/nginx/html
# Add nginx configuration if needed
# copy nginx.conf /etc/nginx/conf.d/default.conf
Expose 80
 CMD ["nginx", "-g", "daemon off;"]
```

7) docker build -t tiktactoe-demo:v1 .

8) docker run -d -p gogg:80 tiktactoe-demo:v1

9) Now we write github workflows like below:

10) Create github token → Github → settings → Developer settings → Personal Access token → classic → Generate new token → Give name → Select scopes as write: packages, read: packages → copy it save in notepad

11) Now we want Github container registry token
go repo settings → secrets and varriables (leftside) → Actions → New repository secret → give name as TOKEN in secreat paste personal Access token → Add secret

12) docker login ghcr.io | Username : github | password : PAT

13) Make some change in scoreBord.tsx as pushpa Raj to pushpa and then commit

14) Now automatically CICD pipeline gets triggred.

15) Now for checking Go to instance
docker login ghcr.io → paste required things

docker run -d -p 80:80 <image name>

we get image name from deployment.yml.

http://EC2-ip:80

16) Install kind → Go to browser → Kind install → Quick start → on linux

17) kind create cluster --name=devsecops-demo-cluster

18) kubectl install → Go to browser → kubectl install → on linux → curl -Lo command paste → sudo install -o root -g command paste

19) check using → kubectl config current-context

20) Now we have install ArgoCD

Go to ArgoDocs → Getting started → run both of command

21) check by → kubectl get pods -n argocd -w

All should be in running state

22) Delete the container that we forward on port 80

docker rm -f <container_id>

23) lsof -i:80

24) kubectl get svc -n argocd

25) kubectl port-forward svc/argocd-server 9000:80 -n argocd --address 0.0.0.0

26) Add port 9000 to SG

27) EC2-IP:9000 → Now we can access Argocd Application page

28) Now username is Admin for password go and

connect EC2 in another tab → kubectl get secrets -n argocd → kubectl edit secret argocd-initial-admin-secret -n argocd → copy password → it is base 64 encoded so →

echo <password> | base64 --decode → Now we get password → login to Argocd

29) Now we deploy the application

In ArgoCD → create app → Give Application name as devsecops-demo → select project name as default → SYNC policy as Automatic → in repo url paste ip of repo → path write kubernetes → cluster url be https://kubernetes.default.svc → Namespace default → create

30) Now we got an error of image pull back of error Why? because we use devsecops practise and in that we don't push the image in public dockerhub repo. Here ArgoCD can not take from github container registry it should have access to that.

How ArgoCD tage image from Github container registry

⇒ We should use imagepullsecrets in deployment.yml and it should have username and password

Now we have create imagepullsecret use below command:

(go repo → kubernets folder → readme.md → there is command)

kubectl create secret docker-registry github-container-registry\

--docker-server=ghcr.io \

--docker-username=Nilesh0203 \

--docker-password=<Personal Access token> \

--docker-email=test@gmail.com

31) Now delete the pod in Argocd

32) Now go scoreboard.tsx and change pusha to pusha raj and commit → CICD automatically trigger.
wait for some time

33) kubectl get pods

34) kubectl port-forward <any one pod name> 3700:80
--address 0.0.0.0

EC2-ip: 3700