

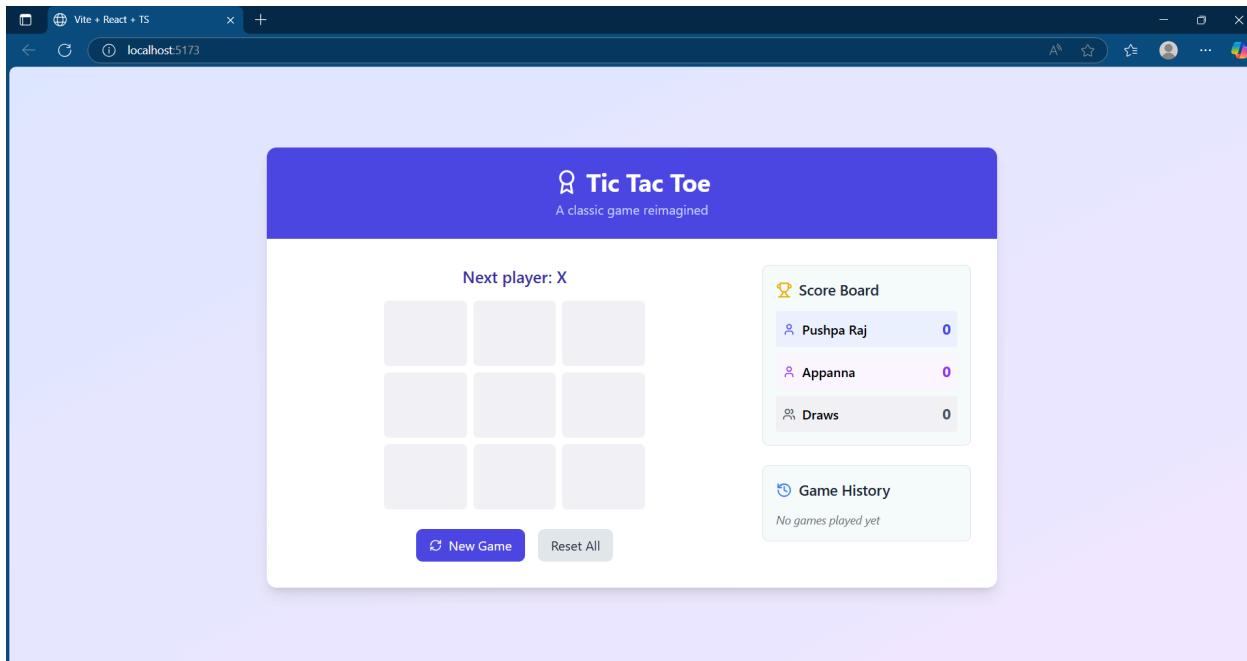
```
lokesh@Lokesh-K:~/devseco ~ + √
lokesh@Lokesh-K:~$ git clone https://github.com/iam-veeramalla/devsecops-demo.git
Cloning into 'devsecops-demo'...
remote: Enumerating objects: 103, done.
remote: Total 103 (delta 0), reused 0 (delta 0), pack-reused 103 (from 1)
Receiving objects: 100% (103/103), 65.34 KiB | 471.00 KiB/s, done.
Resolving deltas: 100% (43/43), done.
lokesh@Lokesh-K:~$ ls
devsecops-demo
lokesh@Lokesh-K:~$ cd devsecops-demo
lokesh@Lokesh-K:~/devsecops-demo$ ls
Dockerfile      index.html      package.json    tailwind.config.js  tsconfig.node.json
README.md       kubernetes     postcss.config.js tsconfig.app.json   vite.config.ts
eslint.config.js package-lock.json  src          tsconfig.json
lokesh@Lokesh-K:~/devsecops-demo$ npm install
added 329 packages, and audited 330 packages in 35s
85 packages are looking for funding
  run 'npm fund' for details
7 vulnerabilities (2 low, 4 moderate, 1 high)
To address all issues, run:
  npm audit fix
Run 'npm audit' for details.
lokesh@Lokesh-K:~/devsecops-demo$
```

- 1.Cloned the project repository from GitHub
- 2.Navigated into the cloned project directory
- 3.Installed all dependencies using - npm install

```
lokesh@Lokesh-K:~/devsecops-demo ~ + √
lokesh@Lokesh-K:~/devsecops-demo$ npm run dev
> vite-react-typescript-starter@0.0.0 dev
> vite

VITE v5.4.8  ready in 617 ms
→ Local:  http://localhost:5173/
→ Network: use --host to expose
→ press h + enter to show help
Browserslist: caniuse-lite is outdated. Please run:
  npx update-browserslist-db@latest
  Why you should do it regularly: https://github.com/browserslist/update-db#readme
```

- 4.Ran the development server using **npm run dev**



```

lokesh@Lokesh-K:~/devsecops-demo$ ls
Dockerfile      index.html    package-lock.json  src          tsconfig.json
README.md       kubernetes   package.json      tailwind.config.js  tsconfig.node.json
eslint.config.js node_modules postcss.config.js  tsconfig.app.json vite.config.ts
lokesh@Lokesh-K:~/devsecops-demo$ docker build -t tictac:v1 .
Command 'docker' not found, but can be installed with:
sudo apt install docker.io  # version 26.1.3-0ubuntu1-24.04.1, or
sudo apt install podman-docker # version 4.9.3+ds1-1ubuntu0.2
lokesh@Lokesh-K:~/devsecops-demo$ sudo apt install docker.io -y
[sudo] password for lokesh:
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following additional packages will be installed:
  bridge-utils containerd dns-root-data dnsmasq-base iptables libip4tc2 libip6tc2 libnetfilter-conntrack3
  libnftnetlink0 libnftables1 libnftnl11 nftables pigz runc ubuntu-fan
Suggested packages:
  ifupdown aufs-tools btrfs-progs cgroupfs-mount | cgroup-lite debootstrap docker-buildx docker-compose-v2 docker-doc
  rinse zfs-fuse | zfsutils firewalld
The following NEW packages will be installed:
  bridge-utils containerd dns-root-data dnsmasq-base docker.io iptables libip4tc2 libip6tc2 libnetfilter-conntrack3
  libnftnetlink0 libnftables1 libnftnl11 nftables pigz runc ubuntu-fan
0 upgraded, 16 newly installed, 0 to remove and 173 not upgraded.
Need to get 80.2 MB of archives.
After this operation, 304 MB of additional disk space will be used.
Get:1 http://archive.ubuntu.com/ubuntu noble/universe amd64 pigz amd64 2.8-1 [65.6 kB]
Get:2 http://archive.ubuntu.com/ubuntu noble/main amd64 libip4tc2 amd64 1.8.10-3ubuntu2 [23.3 kB]
Get:3 http://archive.ubuntu.com/ubuntu noble/main amd64 libip6tc2 amd64 1.8.10-3ubuntu2 [23.7 kB]
Get:4 http://archive.ubuntu.com/ubuntu noble/main amd64 libnftnetlink0 amd64 1.0.2-2build1 [14.8 kB]
Get:5 http://archive.ubuntu.com/ubuntu noble/main amd64 libnetfilter-conntrack3 amd64 1.0.9-6build1 [45.2 kB]

```

## Running the App and Preparing Docker Build :

### 1.Verified App Running Locally:

- Accessed the running app on <http://localhost:5173>
- Confirmed the **Tic Tac Toe** game UI loads properly with score board, game board, and control buttons.

## **2.Prepared for Docker Build:**

- Listed project files using `ls` to ensure the presence of a `Dockerfile`.
  - Attempted to build a Docker image named `tictac:v1` using:

## Error Faced:

- **docker command not found** – The system did not have Docker installed.

So i Installed Docker using **apt** :

```
sudo apt install docker.io -y
```

This installed the necessary Docker packages, including `docker-compose`, `buildx`, and other dependencies.

After installation, Docker was ready to build the image.

```
lokesh@Lokesh-K:~/devseco x + -   
Setting up dnsmasq-base (2.90-2ubuntu0.1) ...  
Setting up ubuntu-fan (0.12.16) ...  
Created symlink /etc/systemd/system/multi-user.target.wants/ubuntu-fan.service → /usr/lib/systemd/system/ubuntu-fan.service.  
Processing triggers for man-db (2.12.0-4ubuntu0.1) ...  
Processing triggers for dbus (1.14.10-4ubuntu4.1) ...  
Processing triggers for libc-bin (2.39-0ubuntu8.4) ...  
lokesh@Lokesh-K:~/devsecops-demo$ docker build -t tictac:v1 .  
DEPRECATED: The legacy builder is deprecated and will be removed in a future release.  
    Install the buildx component to build images with BuildKit:  
    https://docs.docker.com/go/buildx/  
  
permission denied while trying to connect to the Docker daemon socket at unix:///var/run/docker.sock: Post "http://%2Fvar%2Frun%2Fdocker.sock/v1.47/build?buildargs=%7B%7D&cachefrom=%5B%5D&cgroupparent=&cpuperiod=0&cpuquota=0&cpusetcpus=&cpusetmems=&cpushares=0&dockerfile=Dockerfile&labels=%7B%7D&memory=0&memswap=0&networkmode=default&rm=1&shmsize=0&t=tictac%3Av1&target=&ulimits=%5B%5D&version=1": dial unix /var/run/docker.sock: connect: permission denied  
lokesh@Lokesh-K:~/devsecops-demo$ sudo usermod -aG docker lokesh  
lokesh@Lokesh-K:~/devsecops-demo$ docker build -t tictac:v1 .  
DEPRECATED: The legacy builder is deprecated and will be removed in a future release.  
    Install the buildx component to build images with BuildKit:  
    https://docs.docker.com/go/buildx/  
  
permission denied while trying to connect to the Docker daemon socket at unix:///var/run/docker.sock: Post "http://%2Fvar%2Frun%2Fdocker.sock/v1.47/build?buildargs=%7B%7D&cachefrom=%5B%5D&cgroupparent=&cpuperiod=0&cpuquota=0&cpusetcpus=&cpusetmems=&cpushares=0&dockerfile=Dockerfile&labels=%7B%7D&memory=0&memswap=0&networkmode=default&rm=1&shmsize=0&t=tictac%3Av1&target=&ulimits=%5B%5D&version=1": dial unix /var/run/docker.sock: connect: permission denied  
time="2025-07-11T18:14:51Z" level=error msg="Can't add file /home/lokesh/devsecops-demo/package-lock.json to tar: io: read/write on closed pipe"  
time="2025-07-11T18:14:51Z" level=error msg="Can't close tar writer: io: read/write on closed pipe"  
lokesh@Lokesh-K:~/devsecops-demo$ |
```

```
Command Prompt      x  lokesh@Lokesh-K: ~/devsecops      +  v
lokesh@Lokesh-K:~$ ls
devsecops-demo
lokesh@Lokesh-K:~$ cd devsecops-demo
lokesh@Lokesh-K:~/devsecops-demo$ ls
Dockerfile          index.html    package-lock.json   src          tsconfig.json
README.md           kubernetes   package.json       tailwind.config.js  tsconfig.node.json
eslint.config.js   node_modules  postcss.config.js  tsconfig.app.json vite.config.ts
lokesh@Lokesh-K:~/devsecops-demo$ docker build -t tictac:v1 .
DEPRECATED: The legacy builder is deprecated and will be removed in a future release.
            Install the buildx component to build images with BuildKit:
            https://docs.docker.com/go/buildx/
Sending build context to Docker daemon 205.3kB
Step 1/10 : FROM node:20-alpine AS build
20-alpine: Pulling from library/node
fe07684b16b8: Pull complete
0c54b794b004: Pull complete
a54eaddfc49c: Pull complete
b7f182da327e: Pull complete
Digest: sha256:fa316946c0cb1f041fe46dda150f3085b71168555e5706ec0c7466a5bae12244
Status: Downloaded newer image for node:20-alpine
--> da9722576d07
Step 2/10 : WORKDIR /app
--> Running in 5674055f88e1
--> Removed intermediate container 5674055f88e1
--> b1eb32e1c66f
Step 3/10 : COPY package*.json ./
```

Executed Docker build command:

**docker build -t tictac:v1 .**

```
Command Prompt      x  lokesh@Lokesh-K: ~/devsecops      +  v
--> 2def1893f905
Step 7/10 : FROM nginx:alpine
alpine: Pulling from library/nginx
fe07684b16b8: Already exists
3b7062d09e02: Pull complete
fb746e72516f: Pull complete
a9ff9baf1741: Pull complete
2c127093dfc7: Pull complete
63ddaa2adf85b: Pull complete
b55ed7d7b2de: Pull complete
92971aeb101e: Pull complete
Digest: sha256:b2e814d28359e77bd0aa5fed1939620075e4ffa0eb20423cc557b375bd5c14ad
Status: Downloaded newer image for nginx:alpine
--> 77656422f700
Step 8/10 : COPY --from=build /app/dist /usr/share/nginx/html
--> bded74c0f932
Step 9/10 : EXPOSE 80
--> Running in 6fbcd096ef4b
--> Removed intermediate container 6fbcd096ef4b
--> 08741dbdfd60
Step 10/10 : CMD ["nginx", "-g", "daemon off;"]
--> Running in 485ae65b20d3
--> Removed intermediate container 485ae65b20d3
--> 45d036fddedd
Successfully built 45d036fddedd
Successfully tagged tictac:v1
lokesh@Lokesh-K:~/devsecops-demo$
```

```
lokesh@Lokesh-K:~/devsecops-demo$ git add .
lokesh@Lokesh-K:~/devsecops-demo$ git commit -m "Initial commit"
[master (root-commit) e8869b9] Initial commit
 30 files changed, 5835 insertions(+)
 create mode 100644 .dockerignore
 create mode 100644 .github/workflows/README.md
 create mode 100644 .github/workflows/ci-cd.yml
 create mode 100644 .gitignore
 create mode 100644 Dockerfile
 create mode 100644 README.md
 create mode 100644 eslint.config.js
 create mode 100644 index.html
 create mode 100644 kubernetes/README.md
 create mode 100644 kubernetes/deployment.yaml
 create mode 100644 kubernetes/ingress.yaml
 create mode 100644 kubernetes/service.yaml
 create mode 100644 package-lock.json
 create mode 100644 package.json
 create mode 100644 postcss.config.js
 create mode 100644 src/App.tsx
 create mode 100644 src/_tests__/_gameLogic.test.ts
 create mode 100644 src/components/Board.tsx
 create mode 100644 src/components/GameHistory.tsx
 create mode 100644 src/components/ScoreBoard.tsx
 create mode 100644 src/components/Square.tsx
 create mode 100644 src/index.css
 create mode 100644 src/main.tsx
```

The screenshot shows a GitHub DevSecOps pipeline interface. At the top, there's a navigation bar with links for Code, Issues, Pull requests, Actions, Projects, Wiki, Security, Insights, and Settings. A search bar is also present. Below the navigation, a link to a CI/CD Pipeline is shown, followed by a button to Update Username #2. On the left, a sidebar displays sections for Summary, Jobs, and Workflow file. Under Jobs, 'Unit Testing' is selected, and under Workflow file, 'Static Code Analysis' is listed. The main area shows a 'Unit Testing' workflow step that succeeded 1 minute ago. It details the following steps:

- Set up job (1s)
- Checkout code (0s)
- Setup Node.js (3s)
- Install dependencies (4s):
  - Run npm ci (1s)
  - added 329 packages, and audited 330 packages in 5s (4s)
  - 85 packages are looking for funding (6s)
  - run 'npm fund' for details (7s)
  - 7 vulnerabilities (2 low, 4 moderate, 1 high) (8s)
  - To address all issues, run: (9s)
  - npm audit fix (10s)
  - Run 'npm audit' for details. (11s)
- Run tests (1s)

A 'Search logs' input field is located at the bottom right of the workflow details.

```

lokesh@Lokesh-K:~/devsecops-demo$ docker run -d -p 80:80 ghcr.io/lokeshhhk/devsecopss:sha-e2135d2e3c6bb175e5d0d0f7a861e197351ffcc1
Unable to find image 'ghcr.io/lokeshhhk/devsecopss:sha-e2135d2e3c6bb175e5d0d0f7a861e197351ffcc1' locally
docker: Error response from daemon: denied.
See 'docker run --help'.
lokesh@Lokesh-K:~/devsecops-demo$ docker logout
Removing login credentials for https://index.docker.io/v1/
lokesh@Lokesh-K:~/devsecops-demo$ docker login

USING WEB-BASED LOGIN
To sign in with credentials on the command line, use 'docker login -u <username>'

Your one-time device confirmation code is: GQTH-PPCB
Press ENTER to open your browser or submit your device code here: https://login.docker.com/activate

Waiting for authentication in the browser...
^CLogin canceled
lokesh@Lokesh-K:~/devsecops-demo$ docker login ghcr.io
Authenticating with existing credentials...
WARNING! Your password will be stored unencrypted in /home/lokesh/.docker/config.json.
Configure a credential helper to remove this warning. See
https://docs.docker.com/engine/reference/commandline/login/#credential-stores

Login Succeeded
lokesh@Lokesh-K:~/devsecops-demo$ cd /home/lokesh/.docker/config.json.
-bash: cd: /home/lokesh/.docker/config.json.: No such file or directory
lokesh@Lokesh-K:~/devsecops-demo$ docker logout ghcr.io
Removing Login credentials for ghcr.io
lokesh@Lokesh-K:~/devsecops-demo$ docker login ghcr.io
Username: lokeshhhk
Password:
WARNING! Your password will be stored unencrypted in /home/lokesh/.docker/config.json.
Configure a credential helper to remove this warning. See
https://docs.docker.com/engine/reference/commandline/login/#credential-stores

Login Succeeded
lokesh@Lokesh-K:~/devsecops-demo$ docker run -d -p 80:80 ghcr.io/lokeshhhk/devsecopss:sha-e2135d2e3c6bb175e5d0d0f7a861e197351ffcc1
Unable to find image 'ghcr.io/lokeshhhk/devsecopss:sha-e2135d2e3c6bb175e5d0d0f7a861e197351ffcc1' locally

```

This error occurred because if the image does not present locally it has to pulled from docker registry here in this case we used a Github container registry.

Then why the error occurred means when we try to login to docker registry it asks credentials Username and for password we need to give access token

I first logged in with docker with access token that does not have the ability of read packages So it wasn't able to fetch from that.

Debug : Login with the credentials with username and access token with read write packages enabled.

```

Command Prompt      lokesh@Lokesh-K:~/devsecops-demo$ docker logout ghcr.io
Removing login credentials for ghcr.io
lokesh@Lokesh-K:~/devsecops-demo$ docker login ghcr.io
Username: lokeshhhk
Password:
WARNING! Your password will be stored unencrypted in /home/lokesh/.docker/config.json.
Configure a credential helper to remove this warning. See
https://docs.docker.com/engine/reference/commandline/login/#credential-stores

Login Succeeded
lokesh@Lokesh-K:~/devsecops-demo$ docker run -d -p 80:80 ghcr.io/lokeshhhk/devsecopss:sha-e2135d2e3c6bb175e5d0d0f7a861e197351ffcc1
Unable to find image 'ghcr.io/lokeshhhk/devsecopss:sha-e2135d2e3c6bb175e5d0d0f7a861e197351ffcc1' locally
sha-e2135d2e3c6bb175e5d0d0f7a861e197351ffcc1: Pulling from lokeshhhk/devsecopss
fe07684b16b8: Already exists
3b7062d69e02: Already exists
fb746e72516f: Already exists
a9ff9ba1741: Already exists
2c127093dfc7: Already exists
63d0a2adfb58: Already exists
b58ed7d7b2de: Already exists
9297laeb101e: Already exists
c66e0b51c290: Pull complete
Digest: sha256:08ef58757539896a24bf7adc88d62bd8bee508cd328134c3fc857be2dalaef9
Status: Downloaded newer image for ghcr.io/lokeshhhk/devsecopss:sha-e2135d2e3c6bb175e5d0d0f7a861e197351ffcc1
636c8c4901ce4a5ce4afe90e07e931aa973f56ac7267041f1041e328873c69d
lokesh@Lokesh-K:~/devsecops-demo$ 

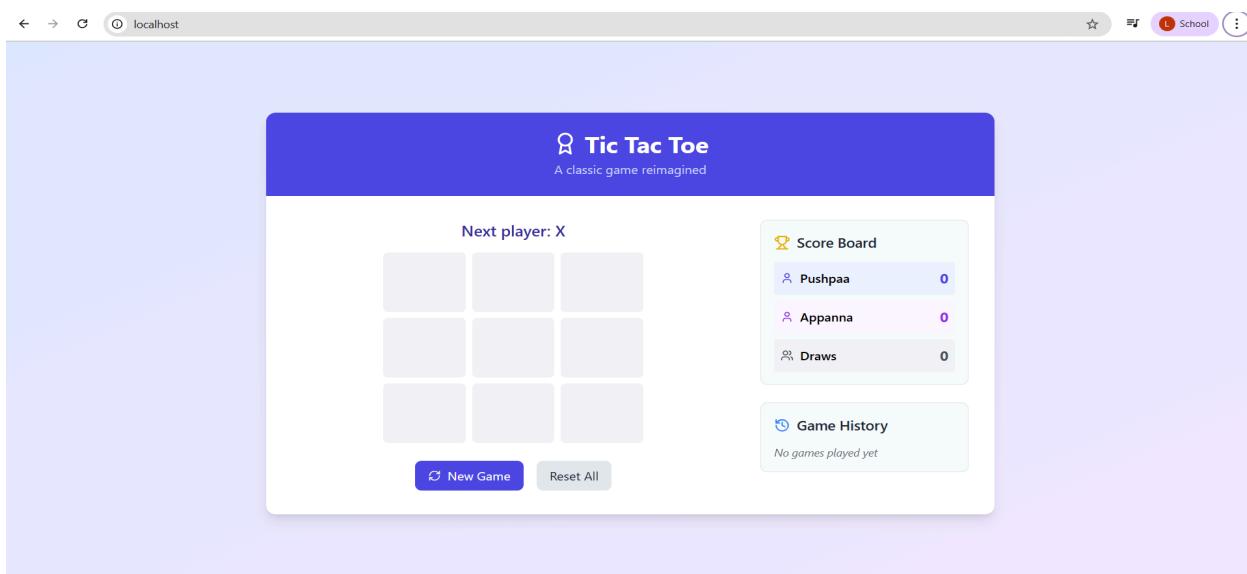
```

Logged In to Docker with GHCR Using Personal Access Token (PAT):

**docker login ghcr.io**

Username: lokeshhhh

Password: Used a GitHub Personal Access Token (PAT).



Pulled Image from GHCR:

**docker run -d -p 80:80 ghcr.io/lokeshhhh/devsecopss:<sha-tag>**

Pulled a specific image using its SHA digest tag from the GitHub Container Registry. Image layers already existed locally, so Docker reused them.

Confirmed That Application Was Running:

- Opened browser at <http://localhost:80>
- Verified that the Tic Tac Toe application loaded successfully.

## Error Faced:

Initially got:

```
Unable to find image 'ghcr.io/lokeshhhh/devsecopss:sha=...' locally
```

- This was expected, as Docker tried to pull it from GHCR.

## Debugging :

- Ensured the correct image URL was used, with valid SHA digest.
- Verified Personal Access Token used during login had `read:packages` permission for GHCR.
- Once image was successfully pulled, the container started and the app was accessible via the browser.

```
login Succeeded
lokesh@Lokesh-K:~/devsecops-demo$ docker run -d -p 80:80 ghcr.io/lokeshhhh/devsecopss:sha-e2135d2e3c6bb175e5d0d0f7a861e197351ffcc1
unable to find image 'ghcr.io/lokeshhhh/devsecopss:sha-e2135d2e3c6bb175e5d0d0f7a861e197351ffcc1' locally
ha-e2135d2e3c6bb175e5d0d0f7a861e197351ffcc1: Pulling from lokeshhhh/devsecopss
e07684b16b8: Already exists
b7062d09e02: Already exists
b746e72516f: Already exists
9ff9ba1741: Already exists
c127093dfc7: Already exists
3ddaa2adfb85b: Already exists
55ed7d7b2de: Already exists
2971aeb101e: Already exists
66e0b5ic290: Pull complete
Digest: sha256:08ef587575339896a24bf7edc88d62bd8bee508cd328134c3fc857be2dalaef9
Status: Downloaded newer image for ghcr.io/lokeshhhh/devsecopss:sha-e2135d2e3c6bb175e5d0d0f7a861e197351ffcc1
36c8c4901cce4a5c4afe96e07e931aa973f56ac7267041f1041e328873c69d
lokesh@Lokesh-K:~/devsecops-demo$ docker run -d -p 8080:80 ghcr.io/lokeshhhh/devsecopss:sha-e2135d2e3c6bb175e5d0d0f7a861e197351ffcc1
a73fd32b635bf3878224f15d2aacbbc23ec2667daab8312ee151f59d97f0cb0
lokesh@Lokesh-K:~/devsecops-demo$ curl localhost:80
!doctype html
html lang="en">
<head>
  <meta charset="UTF-8" />
  <link rel="icon" type="image/svg+xml" href="/vite.svg" />
  <meta name="viewport" content="width=device-width, initial-scale=1.0" />
  <title>Vite + React + TS</title>
  <script type="module" crossorigin src="/assets/index-DKcG-qpk.js"></script>
  <link rel="stylesheet" crossorigin href="/assets/index-CFyCZ5rs.css">
</head>
<body>
  <div id="root"></div>
</body>
</html>
lokesh@Lokesh-K:~/devsecops-demo$ curl localhost:8080
!doctype html
html lang="en">
<head>
```

## Pulled and Ran Docker Image from GitHub Container Registry:

```
docker run -d -p 8080:80 ghcr.io/lokeshhhh/devsecopss:<sha>
```

- This launched the Docker container on port `8080` (host) mapped to port `80` (container).
- The image was successfully pulled and container started.

```

lokesh@Lokesh-K:~/devsecops-demo$ curl localhost:80
<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="UTF-8" />
    <link rel="icon" type="image/svg+xml" href="/vite.svg" />
    <meta name="viewport" content="width=device-width, initial-scale=1.0" />
    <title>Vite + React + TS</title>
    <script type="module" crossorigin src="/assets/index-DKcG-qpk.js"></script>
    <link rel="stylesheet" crossorigin href="/assets/index-CFyCZ5rs.css">
  </head>
  <body>
    <div id="root"></div>
  </body>
</html>
lokesh@Lokesh-K:~/devsecops-demo$ curl localhost:8080
<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="UTF-8" />
    <link rel="icon" type="image/svg+xml" href="/vite.svg" />
    <meta name="viewport" content="width=device-width, initial-scale=1.0" />
    <title>Vite + React + TS</title>
    <script type="module" crossorigin src="/assets/index-DKcG-qpk.js"></script>
    <link rel="stylesheet" crossorigin href="/assets/index-CFyCZ5rs.css">
  </head>
  <body>
    <div id="root"></div>
  </body>
</html>
lokesh@Lokesh-K:~/devsecops-demo$ docker ps
CONTAINER ID IMAGE COMMAND CREATED STATUS
PORTS NAMES
0a73fd32b635 ghcr.io/lokeshhhk/devsecopss:sha-e2135d2e3c6bb175e5d0d0f7a861e197351ffcc1 "/docker-entrypoint..." 7 minutes ago Up 7 minute
s 0.0.0.0:8080->80/tcp, [::]:8080->80/tcp mystifying_villani
636c8c4901cc ghcr.io/lokeshhhk/devsecopss:sha-e2135d2e3c6bb175e5d0d0f7a861e197351ffcc1 "/docker-entrypoint..." 38 minutes ago Up 38 minut
es 0.0.0.0:80->80/tcp, :::80->80/tcp youthful_carson
lokesh@Lokesh-K:~/devsecops-demo$ 

```

## Verified Application Using curl:

`curl localhost:80`

`curl localhost:8080`

- Both commands returned raw HTML source of the React app, indicating that the container is serving content properly.

```

lokesh@Lokesh-K:~/devsecops-demo$ docker ps
CONTAINER ID IMAGE COMMAND CREATED STATUS
PORTS NAMES
0a73fd32b635 ghcr.io/lokeshhhk/devsecopss:sha-e2135d2e3c6bb175e5d0d0f7a861e197351ffcc1 "/docker-entrypoint..." 7 minutes ago Up 7 minute
s 0.0.0.0:8080->80/tcp, [::]:8080->80/tcp mystifying_villani
636c8c4901cc ghcr.io/lokeshhhk/devsecopss:sha-e2135d2e3c6bb175e5d0d0f7a861e197351ffcc1 "/docker-entrypoint..." 38 minutes ago Up 38 minut
es 0.0.0.0:80->80/tcp, :::80->80/tcp youthful_carson
lokesh@Lokesh-K:~/devsecops-demo$ docker stop 636c8c4901cc
636c8c4901cc
lokesh@Lokesh-K:~/devsecops-demo$ curl localhost:80
curl: (7) Failed to connect to localhost port 80 after 1 ms: Couldn't connect to server
lokesh@Lokesh-K:~/devsecops-demo$ curl localhost:8080
<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="UTF-8" />
    <link rel="icon" type="image/svg+xml" href="/vite.svg" />
    <meta name="viewport" content="width=device-width, initial-scale=1.0" />
    <title>Vite + React + TS</title>
    <script type="module" crossorigin src="/assets/index-DKcG-qpk.js"></script>
    <link rel="stylesheet" crossorigin href="/assets/index-CFyCZ5rs.css">
  </head>
  <body>
    <div id="root"></div>
  </body>
</html>
lokesh@Lokesh-K:~/devsecops-demo$ 

```

## 1. Verified HTML Output Using `curl`:

Ran both:

```
curl localhost:80  
curl localhost:8080
```

- Received correct HTML content from both ports, confirming both containers were serving the React app.

## 2. Checked Running Docker Containers:

Used:

```
docker ps
```

Confirmed two containers running the **same image**:

- One exposed on port `8080:80`
- One on port `80:80`

## 3. Stopped One Container to Test Port Behavior:

Stopped the container mapped to port 80:

```
docker stop <container_id>
```

Verified the result using:

```
curl localhost:80  
curl localhost :8080
```

Got error:

```
curl: (7) Failed to connect to localhost port 80: Couldn't connect to server
```

- But `curl localhost:8080` continued to work, confirming only the `80`-mapped container was stopped.

## ✖ Error Faced:

- After stopping the container listening on port 80, a curl request to `localhost:80` failed — **as expected**.

## 🛠 Debugging:

- Used `docker ps` to identify port mappings and container statuses.
- Used `docker stop` to isolate and test container access behavior.
- Used `curl` on both ports to verify which port was still serving the app.



This shows how **Docker port mappings work**. If a container is mapped to `8080:80`, only `localhost:8080` will work, and not `localhost:80`, unless another container explicitly exposes that.

The image displays two screenshots of a web browser window. Both screenshots show a Tic Tac Toe game interface.

**Screenshot 1 (Top):** The address bar shows `localhost`. The page content is blank, indicating that the application is not yet running or is not correctly mapped to the port.

**Screenshot 2 (Bottom):** The address bar shows `localhost:8080`. The page displays a Tic Tac Toe board with the text "Next player: X". To the right of the board is a "Score Board" section showing three entries: "Pushpaa" with 0 points, "Appanna" with 0 points, and "Draws" with 0 points. Below the board is a "Game History" section stating "No games played yet". At the bottom are "New Game" and "Reset All" buttons.

```

</html>
lokesh@LoKesh-K:~/devsecops-demo$ # For AMD64 / x86_64
[ $uname -m ] = x86_64 ] && curl -Lo ./kind https://kind.sigs.k8s.io/dl/v0.29.0/kind-linux-amd64
# For ARM64
[ $uname -m ] = aarch64 ] && curl -Lo ./kind https://kind.sigs.k8s.io/dl/v0.29.0/kind-linux-arm64
chmod +x ./kind
sudo mv ./kind /usr/local/bin/kind
  % Total    % Received % Xferd  Average Speed   Time     Time  Current
          Dload  Upload Total Spent   Left Speed
100  97 100  97  0    0   78      0  0:00:01  0:00:01 --:--:--  78
  0    0    0    0    0    0    0      0  0:00:02  0:00:02 --:--:--    0
100 10.5M 100 10.5M  0    0  717k      0  0:00:15  0:00:15 --:--:-- 779k
[sudo] password for lokesh:
lokesh@LoKesh-K:~/devsecops-demo$ kind
kind creates and manages local Kubernetes clusters using Docker container 'nodes'

Usage:
  kind [command]

Available Commands:
  build      Build one of [node-image]
  completion Output shell completion code for the specified shell (bash, zsh or fish)
  create     Creates one of [cluster]
  delete     Deletes one of [cluster]
  export     Exports one of [kubeconfig, logs]
  get        Gets one of [clusters, nodes, kubeconfig]
  help       Help about any command
  load       Loads images into nodes
  version    Prints the kind CLI version

Flags:
  -h, --help            help for kind
  -q, --quiet           silence all stderr output
  -v, --verbosity int32 info log verbosity, higher value produces more output
  --version             version for kind

Use "kind [command] --help" for more information about a command.
lokesh@LoKesh-K:~/devsecops-demo$ clear

```

## Created a Kubernetes Cluster Using KIND:

```
kind create cluster --name=devsecops-cluster
```

```

lokesh@LoKesh-K:~/devsecops-demo$ kind create cluster --name=devsecops-cluster
Creating cluster "devsecops-cluster" ...
  Ensuring node image (kindest/node:v1.33.1) 
  Ensuring node image (kindest/node:v1.33.1) 
lokesh@LoKesh-K:~/devsecops-demo$ kind create cluster --name=devsecops-cluster
Creating cluster "devsecops-cluster" ...
  Ensuring node image (kindest/node:v1.33.1) 
  Preparing nodes 
  Writing configuration 
  Starting control-plane 
  Installing CNI 
  Installing StorageClass 
Set kubectl context to "kind-devsecops-cluster"
You can now use your cluster with:

kubectl cluster-info --context kind-devsecops-cluster

Thanks for using kind! 😊
lokesh@LoKesh-K:~/devsecops-demo$ curl -LO "https://dl.k8s.io/release/$(curl -L -s https://dl.k8s.io/release/stable.txt)/bin/linux/amd64/kubectl"
% Total    % Received % Xferd  Average Speed   Time     Time  Current
          Dload  Upload Total Spent   Left Speed
100 138 100 138  0    0   303      0  --:--:-- --:--:-- 303
100 57.3M 100 57.3M  0    0  3071k      0  0:00:19  0:00:19 --:--:-- 3775k
lokesh@LoKesh-K:~/devsecops-demo$ sudo install -o root -g root -m 0755 kubectl /usr/local/bin/kubectl
lokesh@LoKesh-K:~/devsecops-demo$ kubectl config current-context
kind-devsecops-cluster
lokesh@LoKesh-K:~/devsecops-demo$ kubectl get nodes
NAME           STATUS   ROLES      AGE   VERSION
devsecops-cluster-control-plane   Ready   control-plane   3m21s   v1.33.1
lokesh@LoKesh-K:~/devsecops-demo$ 

```

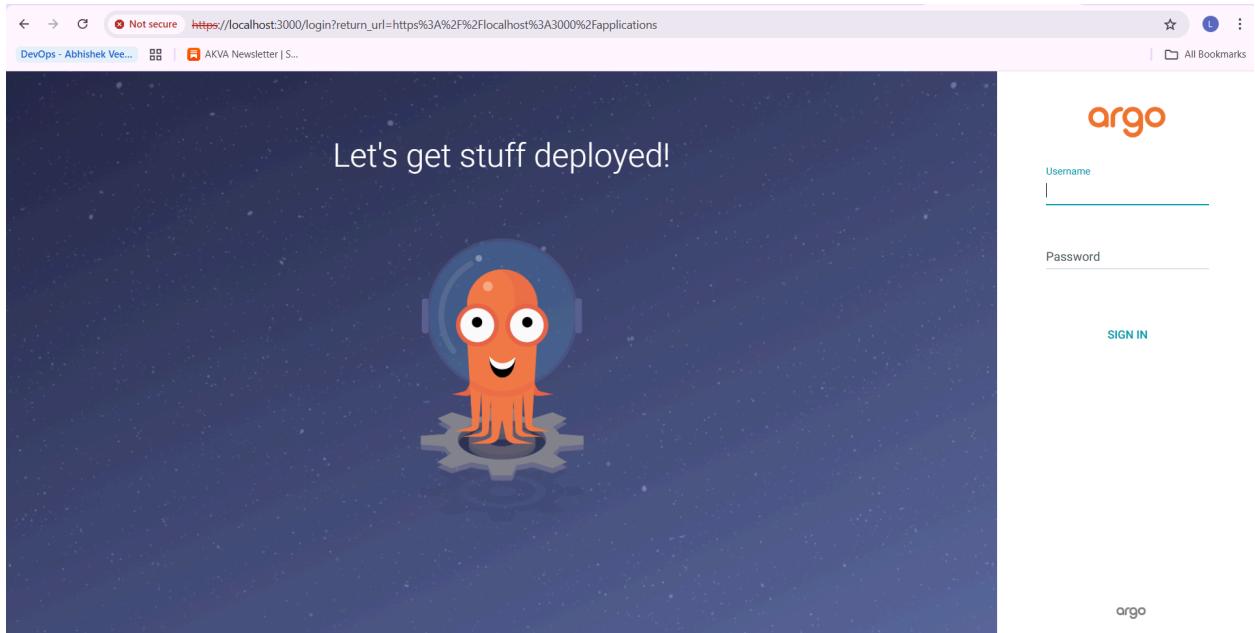
```

lokesh@Lokesh-K:~/devsecops-demo$ kubectl create namespace argocd
namespace/argocd created

kubectl apply -n argocd -f https://raw.githubusercontent.com/argoproj/argo-cd/stable/manifests/install.yaml
customresourcedefinition.apiextensions.k8s.io/applications.argoproj.io created
customresourcedefinition.apiextensions.k8s.io/applicationsets.argoproj.io created
customresourcedefinition.apiextensions.k8s.io/approjects.argoproj.io created
serviceaccount/argocd-application-controller created
serviceaccount/argocd-applicationset-controller created
serviceaccount/argocd-dex-server created
serviceaccount/argocd-notifications-controller created
serviceaccount/argocd-redis created
serviceaccount/argocd-repo-server created
serviceaccount/argocd-server created
role.rbac.authorization.k8s.io/argocd-application-controller created
role.rbac.authorization.k8s.io/argocd-applicationset-controller created
role.rbac.authorization.k8s.io/argocd-dex-server created
role.rbac.authorization.k8s.io/argocd-notifications-controller created
role.rbac.authorization.k8s.io/argocd-redis created
role.rbac.authorization.k8s.io/argocd-server created
clusterrole.rbac.authorization.k8s.io/argocd-application-controller created
clusterrole.rbac.authorization.k8s.io/argocd-applicationset-controller created
clusterrole.rbac.authorization.k8s.io/argocd-server created
rolebinding.rbac.authorization.k8s.io/argocd-application-controller created
rolebinding.rbac.authorization.k8s.io/argocd-applicationset-controller created
rolebinding.rbac.authorization.k8s.io/argocd-dex-server created
rolebinding.rbac.authorization.k8s.io/argocd-notifications-controller created
rolebinding.rbac.authorization.k8s.io/argocd-redis created
rolebinding.rbac.authorization.k8s.io/argocd-server created
clusterrolebinding.rbac.authorization.k8s.io/argocd-application-controller created
clusterrolebinding.rbac.authorization.k8s.io/argocd-applicationset-controller created
clusterrolebinding.rbac.authorization.k8s.io/argocd-server created
configmap/argocd-cm created
configmap/argocd-cmd-params-cm created
configmap/argocd-gpg-keys-cm created
configmap/argocd-notifications-cm created
configmap/argocd-rbac-cm created
configmap/argocd-ssh-known-hosts-cm created
configmap/argocd-tls-certs-cm created

networkpolicy.networking.k8s.io/argocd-server-network-policy created
lokesh@Lokesh-K:~/devsecops-demo$ kubectl get pods -n argocd
NAME                               READY   STATUS    RESTARTS   AGE
argocd-application-controller-0   0/1     ContainerCreating   0   65s
argocd-applicationset-controller-655cc58ff8-786ln 1/1     Running   0   67s
argocd-dex-server-7d9dfb4fb8-bt2qf 0/1     PodInitializing   0   66s
argocd-notifications-controller-6c6848bc4c-7mc85 0/1     ContainerCreating   0   66s
argocd-redis-656c79549c-z476l 0/1     PodInitializing   0   66s
argocd-repo-server-856b768fd9-lfnp5 0/1     Init:0/1      0   66s
argocd-server-99c485944-zf6vn 0/1     ContainerCreating   0   65s
lokesh@Lokesh-K:~/devsecops-demo$ kubectl get pods -n argocd -w
NAME                               READY   STATUS    RESTARTS   AGE
argocd-application-controller-0   0/1     ContainerCreating   0   73s
argocd-applicationset-controller-655cc58ff8-786ln 1/1     Running   0   75s
argocd-dex-server-7d9dfb4fb8-bt2qf 0/1     PodInitializing   0   74s
argocd-notifications-controller-6c6848bc4c-7mc85 0/1     Running   0   74s
argocd-redis-656c79549c-z476l 0/1     PodInitializing   0   74s
argocd-repo-server-856b768fd9-lfnp5 0/1     Init:0/1      0   74s
argocd-server-99c485944-zf6vn 0/1     Running   0   73s
argocd-repo-server-856b768fd9-lfnp5 0/1     Init:0/1      0   74s
argocd-repo-server-856b768fd9-lfnp5 0/1     PodInitializing   0   75s
argocd-application-controller-0   0/1     Running   0   74s
argocd-application-controller-0   1/1     Running   0   86s
argocd-dex-server-7d9dfb4fb8-bt2qf 1/1     Running   0   92s
argocd-server-99c485944-zf6vn 1/1     Running   0   100s
argocd-redis-656c79549c-z476l 1/1     Running   0   103s
argocd-repo-server-856b768fd9-lfnp5 0/1     Running   0   104s
argocd-repo-server-856b768fd9-lfnp5 1/1     Running   0   114s
^C
lokesh@Lokesh-K:~/devsecops-demo$ kubectl get pods -n argocd
NAME                               READY   STATUS    RESTARTS   AGE
argocd-application-controller-0   1/1     Running   0   9m40s
argocd-applicationset-controller-655cc58ff8-786ln 1/1     Running   0   9m42s
argocd-dex-server-7d9dfb4fb8-bt2qf 1/1     Running   0   9m41s
argocd-notifications-controller-6c6848bc4c-7mc85 1/1     Running   0   9m41s
argocd-redis-656c79549c-z476l 1/1     Running   0   9m41s
argocd-repo-server-856b768fd9-lfnp5 1/1     Running   0   9m41s
argocd-server-99c485944-zf6vn 1/1     Running   0   9m40s

```



Here it will ask password for the argocd so we need to use like

**Kubectl get secrets -n argocd**

In that we need to view initial admin secret

**Kubectl edit secret argocd-initial-admin-secret -n argocd**

```
Lokesh@Lokesh-K: $ kubectl get secrets -n argocd
NAME                TYPE      DATA   AGE
argocd-initial-admin-secret  Opaque    1     20m
argocd-notifications-secret  Opaque    0     21m
argocd-redis            Opaque    1     20m
argocd-secret           Opaque    5     21m
Lokesh@Lokesh-K: $ kubectl edit secret argocd-initial-admin-secret -n argocd
Error from server (NotFound): secrets "argocd-initial-admin-secret" not found
Lokesh@Lokesh-K: $ kubectl edit secret argocd-initial-admin-secret -n argocd
Edit cancelled, no changes made.
Lokesh@Lokesh-K: $ echo RXN1aDZmbU5Bd1p1YzVnaQ== | base64 --decode
EshhGfmNAvZuc5giLokesh@Lokesh-K: $
```

Codepaces secrets - lokeshhhk | Not secure | Update Username - lokeshhhk/i | Trivy pipeline troubleshooting | lokeshhhk/devsecopss | cicdpipeline - Application Details

https://localhost:3000/applications/argocd/cicdpipeline?view=tree&resource=&node=%2FPod%2Fdefault%2Ftic-tac-toe-6ff6dc4f9-474rd%2F

AKVA Newsletter | ... All Bookmarks

**tic-tac-toe-6ff6dc4f9-474rd**

**SUMMARY** **EVENTS** (39) **LOGS**

KIND	Pod
NAME	tic-tac-toe-6ff6dc4f9-474rd
NAMESPACE	default
CREATED AT	07/12/2025 11:42:38 (7 minutes ago)
IMAGES	ghcr.io/lokeshhhk/devsecopss:sha-e2135d2e3c6bb175e5d0d0f7a861e197351ffcc1
STATE	ImagePullBackOff
STATE DETAILS	Back-off pulling image "ghcr.io/lokeshhhk/devsecopss:sha-e2135d2e3c6bb175e5d0d0f7a861e197351ffcc1": ErrImagePull: failed to pull and unpack image "ghcr.io/lokeshhhk/devsecopss:sha-e2135d2e3c6bb175e5d0d0f7a861e197351ffcc1": failed to resolve reference "ghcr.io/lokeshhhk/devsecopss:sha-e2135d2e3c6bb175e5d0d0f7a861e197351ffcc1": failed to authorize: failed to fetch anonymous token: unexpected status from GET request to https://ghcr.io/token?

Codepaces secrets - lokeshhhk | Not secure | Update Username - lokeshhhk/i | Trivy pipeline troubleshooting | lokeshhhk/devsecopss | cicdpipeline - Application Details

https://localhost:3000/applications/argocd/cicdpipeline?view=tree&resource=

AKVA Newsletter | ... All Bookmarks

**APPLICATION DETAILS TREE**

**DETAILS** **DIFF** **SYNC** **SYNC STATUS** **HISTORY AND ROLLBACK** **DELETE** **REFRESH** Log out

**Progressing**

**SYNC STATUS** Synced to HEAD (1f45098) **LAST SYNC** Sync OK to 1f45098

Auto sync is enabled.  
Author: GitHub Actions <actions@github.com>  
Comment: Update Kubernetes deployment with new image tag: ...  
Succeeded 13 minutes ago (Sat Jul 12 2025 11:42:38 GMT+0530)  
Author: GitHub Actions <actions@github.com>  
Comment: Update Kubernetes deployment with new image tag: ...

```

graph LR
    svc[tic-tac-toe svc] --> deploy[tic-tac-toe deploy]
    svc --> ingress[tic-tac-toe-ingress ing]
    deploy --> rs[tic-tac-toe-6ff6dc4f9 rs]
    ingress --> rs
    rs --> pod1[tic-tac-toe-6ff6dc4f9-6xd77 pod]
    rs --> pod2[tic-tac-toe-6ff6dc4f9-dvrdr pod]
    rs --> pod3[tic-tac-toe-6ff6dc4f9-h5wsf pod]
  
```

The diagram illustrates the deployment structure for the 'tic-tac-toe' application. It starts with a Service ('tic-tac-toe svc') which branches into a Deployment ('tic-tac-toe deploy') and an Ingress ('tic-tac-toe-ingress ing'). The Deployment ('tic-tac-toe-6ff6dc4f9') manages three pods: 'tic-tac-toe-6ff6dc4f9-6xd77', 'tic-tac-toe-6ff6dc4f9-dvrdr', and 'tic-tac-toe-6ff6dc4f9-h5wsf'. Each pod is currently running 1/1 instance.

```

template:
metadata:
  labels:
    app: tic-tac-toe
spec:
  containers:
    - name: tic-tac-toe
      image: ghcr.io/lokeshhhk/devsecopss:sha-e2135d2e3c6bb175e5d0d0f7a861e197351ffcc1
        # Image will be automatically updated by CI/CD pipeline
      imagePullPolicy: Always
    ports:
      - containerPort: 80
    resources:
      limits:
        cpu: "0.5"
        memory: "512Mi"
      requests:
        cpu: "0.2"
        memory: "256Mi"
    livenessProbe:
      httpGet:
        path: /
        port: 80
      initialDelaySeconds: 30
      periodSeconds: 10
    readinessProbe:
      httpGet:
        path: /
        port: 80

```

same image is update in the code also.

```

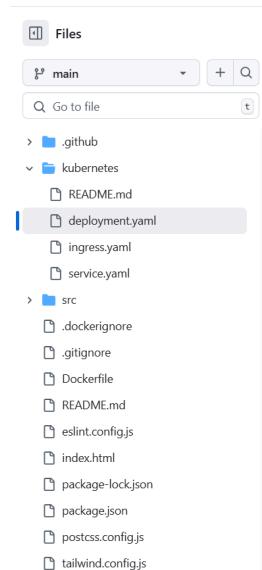
import { Trophy, User, Users } from 'lucide-react';

interface ScoreBoardProps {
  scores: {
    X: number;
    O: number;
    draw: number;
  };
}

const ScoreBoard: React.FC<ScoreBoardProps> = ({ scores }) => {
  return (
    <div className="bg-gray-50 p-4 rounded-lg border border-gray-200">
      <h2 className="text-lg font-semibold text-gray-800 mb-3 flex items-center gap-2">
        <Trophy className="h-5 w-5 text-yellow-500" />
        Score Board
      </h2>
      <div className="space-y-2">
        <div className="flex justify-between items-center p-2 bg-indigo-50 rounded">
          <div className="flex items-center gap-2">
            <User className="h-4 w-4 text-indigo-600" />
            <span className="font-medium">lokesh</span>
          </div>
          <span className="text-lg font-bold text-indigo-600">>{scores.X}</span>
        </div>
        <div className="flex justify-between items-center p-2 bg-purple-50 rounded">

```

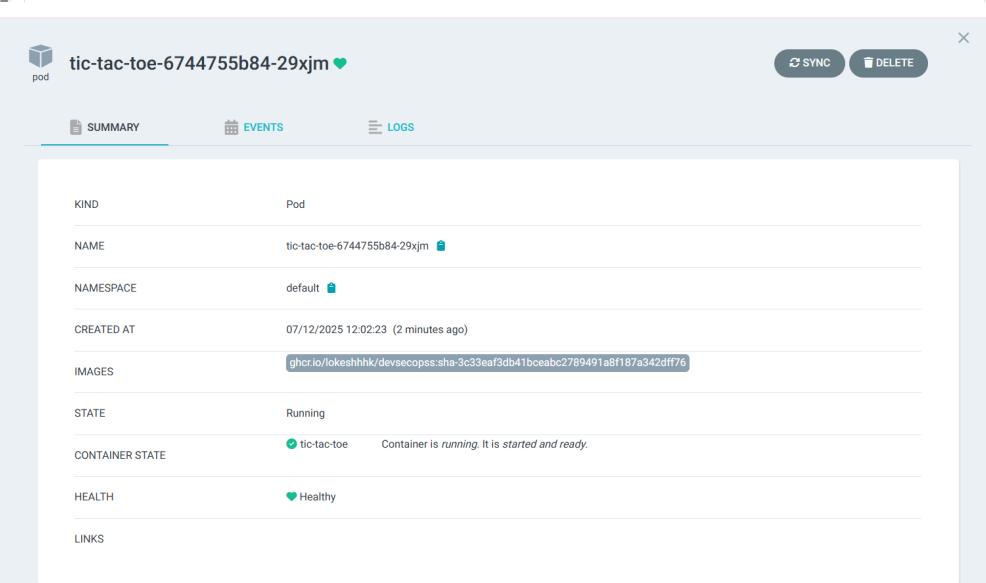
We make now change and lets see it is updated in application.



```

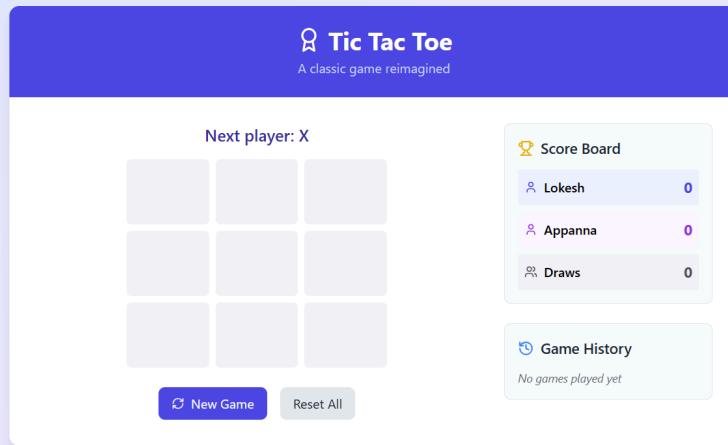
devsecopss / kubernetes / deployment.yaml
Code Blame 44 lines (44 loc) · 1 KB Code 55% faster with GitHub Copilot
main
deployment.yaml
  replicas: 3
  selector:
    matchLabels:
      app: tic-tac-toe
  template:
    metadata:
      labels:
        app: tic-tac-toe
    spec:
      containers:
        - name: tic-tac-toe
          image: ghr.io/lokeshhhk/devsecopss:sha-3c33eaef3db41bceabc2789491a8f187a342dff76
          # Image will be automatically updated by CI/CD pipeline
          imagePullPolicy: Always
          ports:
            - containerPort: 80
          resources:
            limits:
              cpu: "0.5"
              memory: "512Mi"
            requests:
              cpu: "0.2"
              memory: "256Mi"
          livenessProbe:
            httpGet:
              path: /
              port: 80
            initialDelaySeconds: 30
            periodSeconds: 10
          readinessProbe:

```



**tic-tac-toe-6744755b84-29xjm** ❤️

SUMMARY	EVENTS	LOGS
KIND	Pod	
NAME	tic-tac-toe-6744755b84-29xjm	
NAMESPACE	default	
CREATED AT	07/12/2025 12:02:23 (2 minutes ago)	
IMAGES	ghcr.io/lokeshhhk/devsecopss:sha-3c33eaef3db41bceabc2789491a8f187a342dff76	
STATE	Running	
CONTAINER STATE	tic-tac-toe	Container is running. It is started and ready.
HEALTH	Healthy	
LINKS		



---

Now the changes reflected in the application and the container image are also updated according to each commit.