OWASP Juice Shop Deployment on Kubernetes

Introduction

Project Goal: To successfully deploy the OWASP Juice Shop—a modern, intentionally vulnerable e-commerce application used for security training—onto a Kubernetes cluster.

Expected Outcome: By the end of this project, we will have a persistent, externally accessible instance of the Juice Shop running, demonstrating core container orchestration principles and validating access to the application via an assigned cluster port. This serves as a foundational exercise in managing containerized applications at scale.

What is OWASP?

OWASP stands for the Open Worldwide Application Security Project. It is a global, non-profit organization dedicated to improving software security.

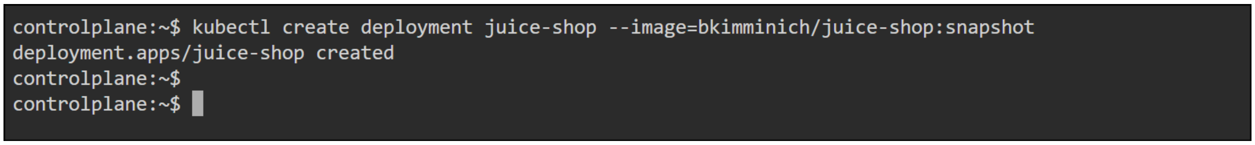
- Mission: OWASP provides freely available resources, tools, documentation, and best practices to help organizations and individuals design, develop, acquire, and maintain secure software.
- Key Contribution (The OWASP Top 10): OWASP is most famous for its regularly updated Top 10 report, which outlines the ten most critical security risks to web applications. This report serves as an essential awareness document and guide for developers and security teams globally.
- Community Model: It operates as an open community, meaning anyone can participate in and contribute to its projects, such as the Juice Shop.

Kubernetes Deployment Steps

This section details the steps to deploy the bkimminich/juice-shop Docker image using Kubernetes.

Step1: Create Deployment

kubectl create deployment juice-shop –image=bkimminich/juice-shop:snapshot

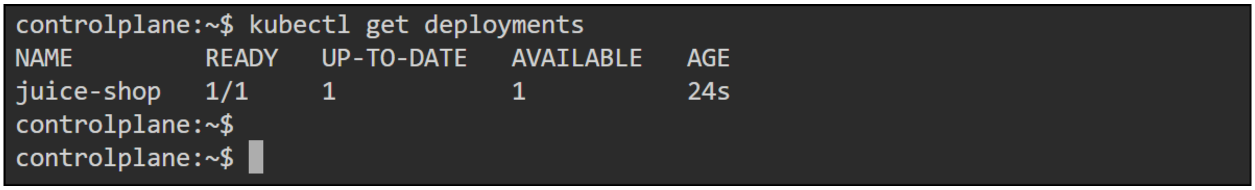


```
controlplane:~$ kubectl create deployment juice-shop --image=bkimminich/juice-shop:snapshot
deployment.apps/juice-shop created
controlplane:~$
controlplane:~$ 
```

Create the deployment object directly from the command line, specifying the name and image.

Step2: Checking Deployment Status

kubectl get deployments

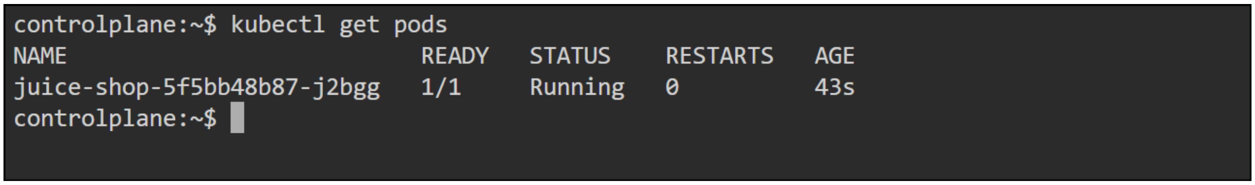


```
controlplane:~$ kubectl get deployments
NAME         READY   UP-TO-DATE   AVAILABLE   AGE
juice-shop   1/1     1            1           24s
controlplane:~$
controlplane:~$ 
```

checks the status of the deployment object

Step3: Checking Container Status

kubectl get pods

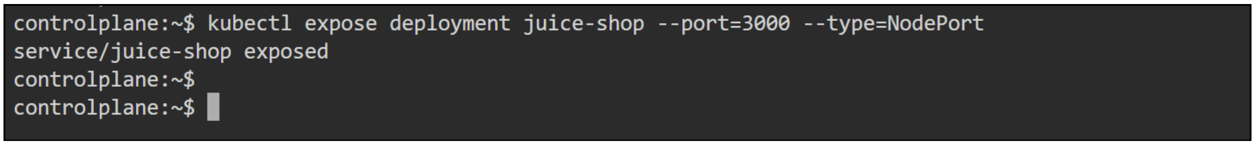


```
controlplane:~$ kubectl get pods
NAME                          READY   STATUS    RESTARTS   AGE
juice-shop-5f5bb48b87-j2bgg   1/1     Running   0          43s
controlplane:~$ 
```

Lists individual pods managed by the Deployments. The output confirms that the container is downloaded and running.

Step4: Creating the External Service

kubectl expose deployment juice-shop --port=3000 --type=NodePort

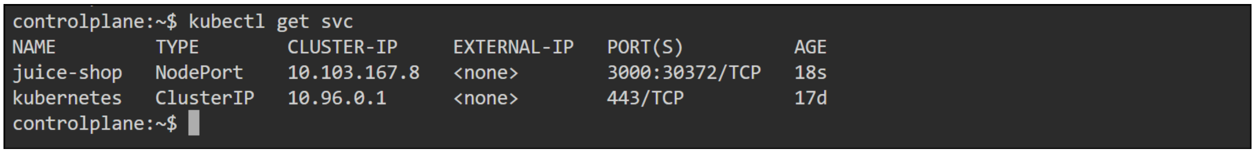


```
controlplane:~$ kubectl expose deployment juice-shop --port=3000 --type=NodePort
service/juice-shop exposed
controlplane:~$
controlplane:~$ 
```

Created a service named after the Deployment(juice-shop) setting the type to NodePort exposes internal port (3000) to the outside world on a high port (usually between 30000-32767) on every cluster node.

Step5: Retrieving the Service Port

kubectl get svc

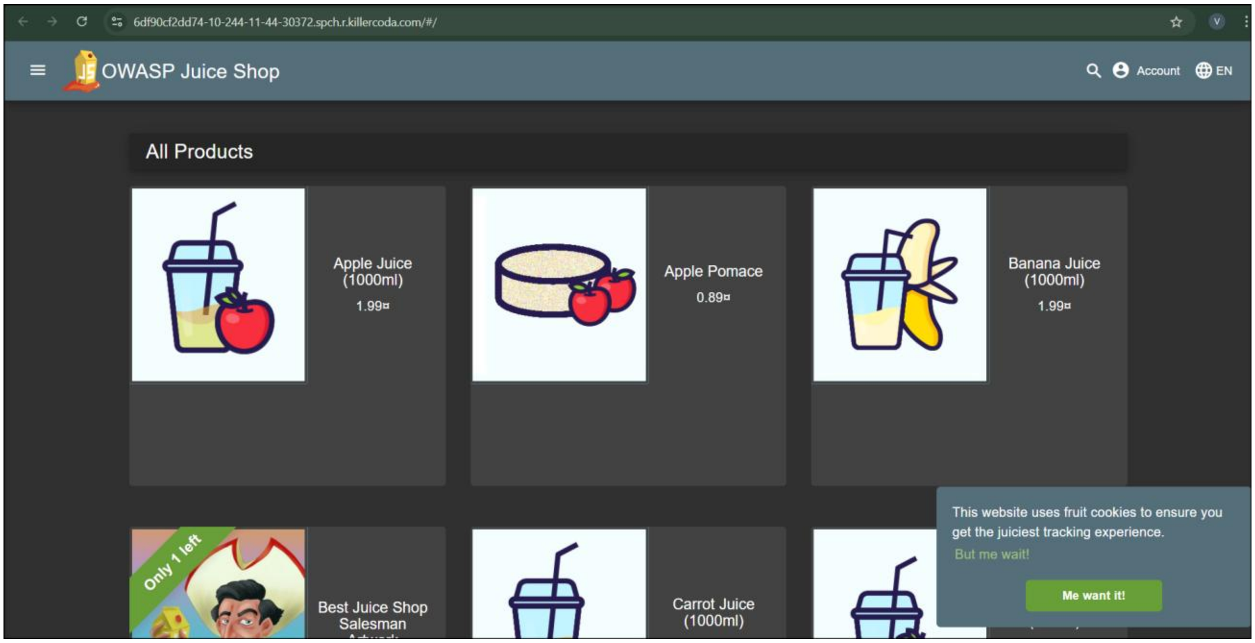


```
controlplane:~$ kubectl get svc
NAME         TYPE        CLUSTER-IP     EXTERNAL-IP   PORT(S)          AGE
juice-shop   NodePort    10.103.167.8   <none>        3000:30372/TCP   18s
kubernetes   ClusterIP   10.96.0.1      <none>        443/TCP          17d
controlplane:~$ 
```

Lists all active services, look at the juice-shop service’s PORTS column. You will see a mapping like 3000:30372/TCP. The 30372 number is the crucial NodePort needed to access the application externally.

Step6: Access the Application

<node-ip>:NodePort



6df90cf2dd74-10-244-11-44-30372.spch.r.killercoda.com/#/

This website uses fruit cookies to ensure you get the juiciest tracking experience.

Conclusion:

Successfully deployed the OWASP Juice Shop application on Kubernetes. A deployment ensured the container was running. And a NodePort Service provided external access, completing a basic, functional deployment.