# Techstrong Deepfactor SCA 2.0 Workshop

## Introduction

This workshop is designed to showcase Deepfactor's runtime SCA capability and how it can be used to prioritize SCA vulnerabilities. In this workshop we will scan and run a spring boot container image and experience how we can use the SCA 2.0 framework to prioritize the true risk rather than relying on CVSS score alone.

## Goal

Experience the power of Deepfactor's SCA 2.0 framework!

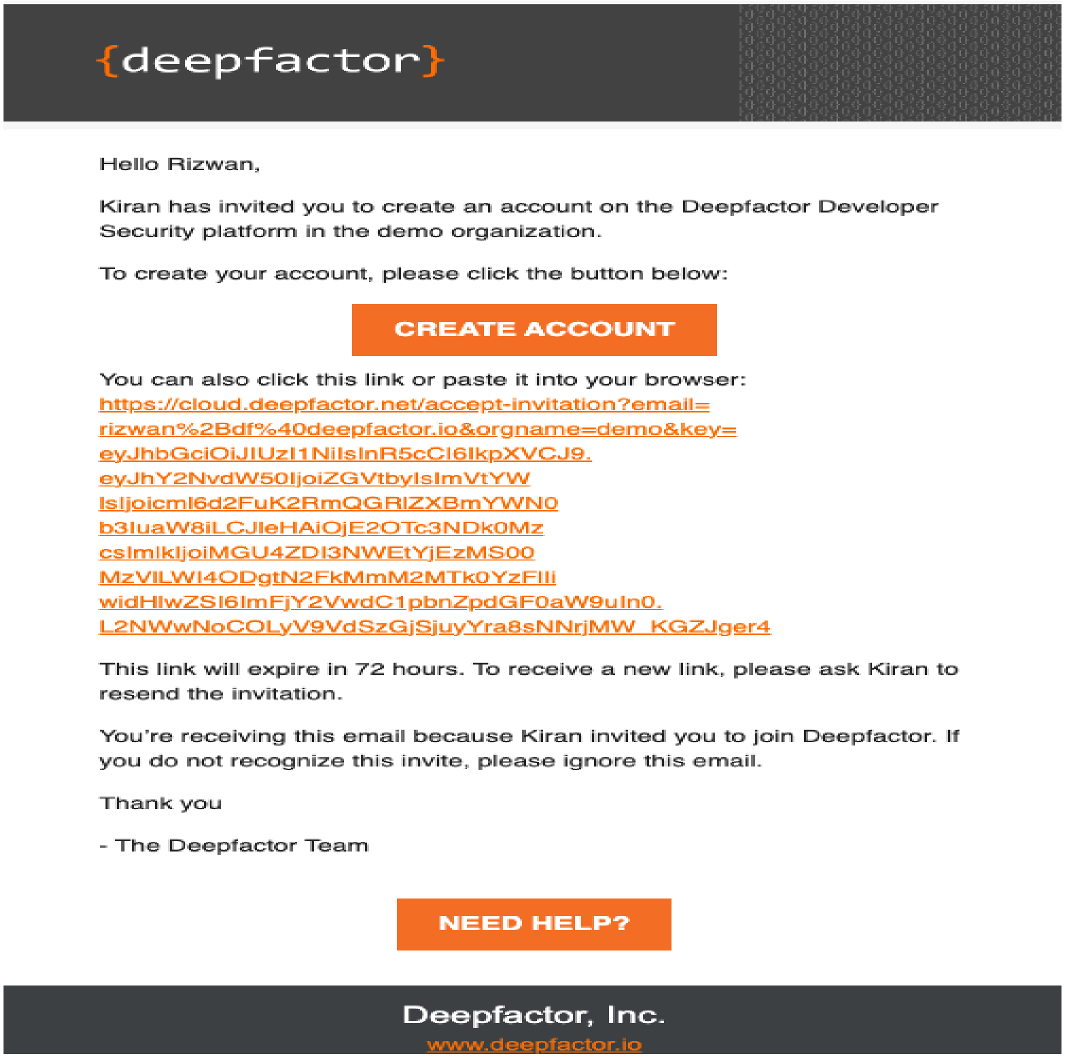
# Workshop Logistics

## Activate Deepfactor Account

Please check for an email with the subject "*Deepfactor Workshop has invited you to create a Deepfactor account*" sent to your email address used to register for the workshop.

Click on the CREATE ACCOUNT link to activate your account. Once the account is activated, you may proceed to login step # 1 from the workshop section.

Following is screen capture of the sample email for reference.



Hello Rizwan,

Kiran has invited you to create an account on the Deepfactor Developer Security platform in the demo organization.

To create your account, please click the button below:

You can also click this link or paste it into your browser:
https://cloud.deepfactor.net/accept-invitation?email=rizwan%2Bdf%40deepfactor.io&orgname=demo&key=eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJhY2NvdW50IjoiZGVtbyIsImVtYWIsIjoicml6d2FuK2RmQGRlZXBmYWN0b3IuaW8iLCJleHAiOjE2OTc3NDk0MzcsImlkIjoiMGU4ZDI3NWEtYjEzMS00MzViLWI4ODgtN2FkMmM2MTk0YzFlIiwidHlwZSI6ImFjY2VwdC1pbnZpdGF0aW9uIn0.L2NWwNoCOLyV9VdSzGjSjuyYra8sNNrjMW_KGZJger4

This link will expire in 72 hours. To receive a new link, please ask Kiran to resend the invitation.

You're receiving this email because Kiran invited you to join Deepfactor. If you do not recognize this invite, please ignore this email.

Thank you

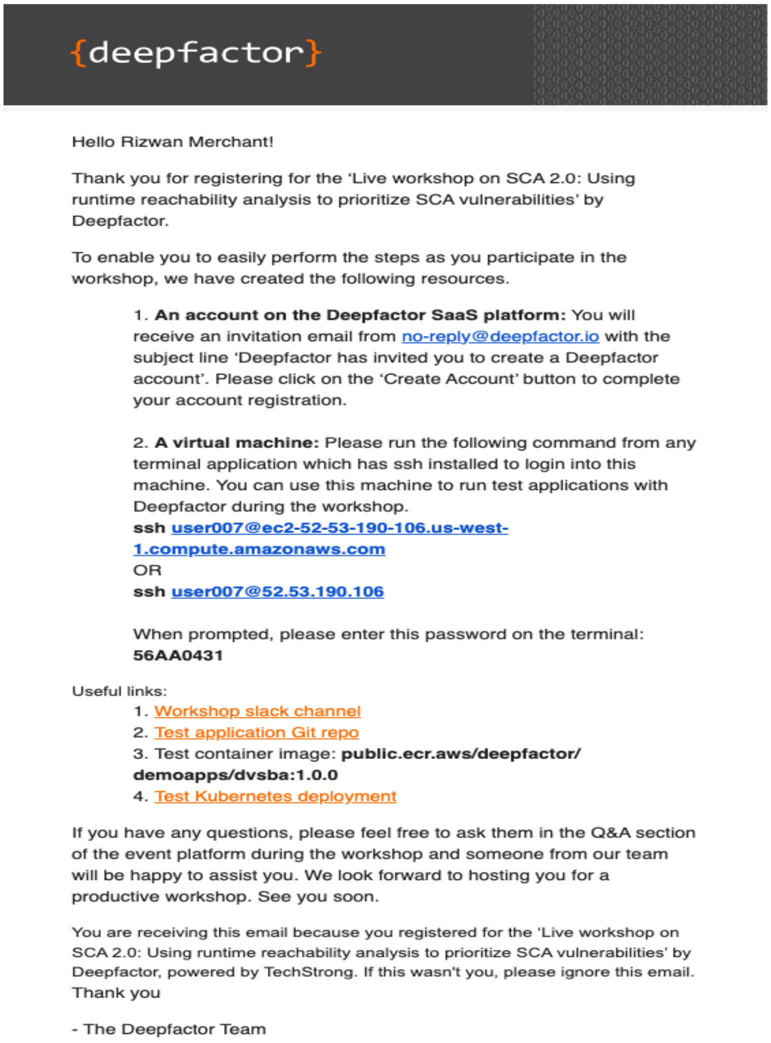
- The Deepfactor Team

Deepfactor, Inc.
www.deepfactor.io

## Additional resources

A few additional resources are shared with you over an email. These include the following:

1. Login to the test VM using command from item # 2 on virtual machine.You need ssh to login to this VM instance
2. Join Deepfactor Slack channel using link in items # 1 from Useful links section
3. GitHub repository for the application used in this workshop



{deepfactor}

Hello Rizwan Merchant!

Thank you for registering for the 'Live workshop on SCA 2.0: Using runtime reachability analysis to prioritize SCA vulnerabilities' by Deepfactor.

To enable you to easily perform the steps as you participate in the workshop, we have created the following resources.

1. **An account on the Deepfactor SaaS platform:** You will receive an invitation email from no-reply@deepfactor.io with the subject line 'Deepfactor has invited you to create a Deepfactor account'. Please click on the 'Create Account' button to complete your account registration.

2. **A virtual machine:** Please run the following command from any terminal application which has ssh installed to login into this machine. You can use this machine to run test applications with Deepfactor during the workshop.
**ssh user007@ec2-52-53-190-106.us-west-1.compute.amazonaws.com**
OR
**ssh user007@52.53.190.106**

When prompted, please enter this password on the terminal:
**56AA0431**

Useful links:

1. Workshop slack channel
2. Test application Git repo
3. Test container image: **public.ecr.aws/deepfactor/demoapps/dvsba:1.0.0**
4. Test Kubernetes deployment

If you have any questions, please feel free to ask them in the Q&A section of the event platform during the workshop and someone from our team will be happy to assist you. We look forward to hosting you for a productive workshop. See you soon.

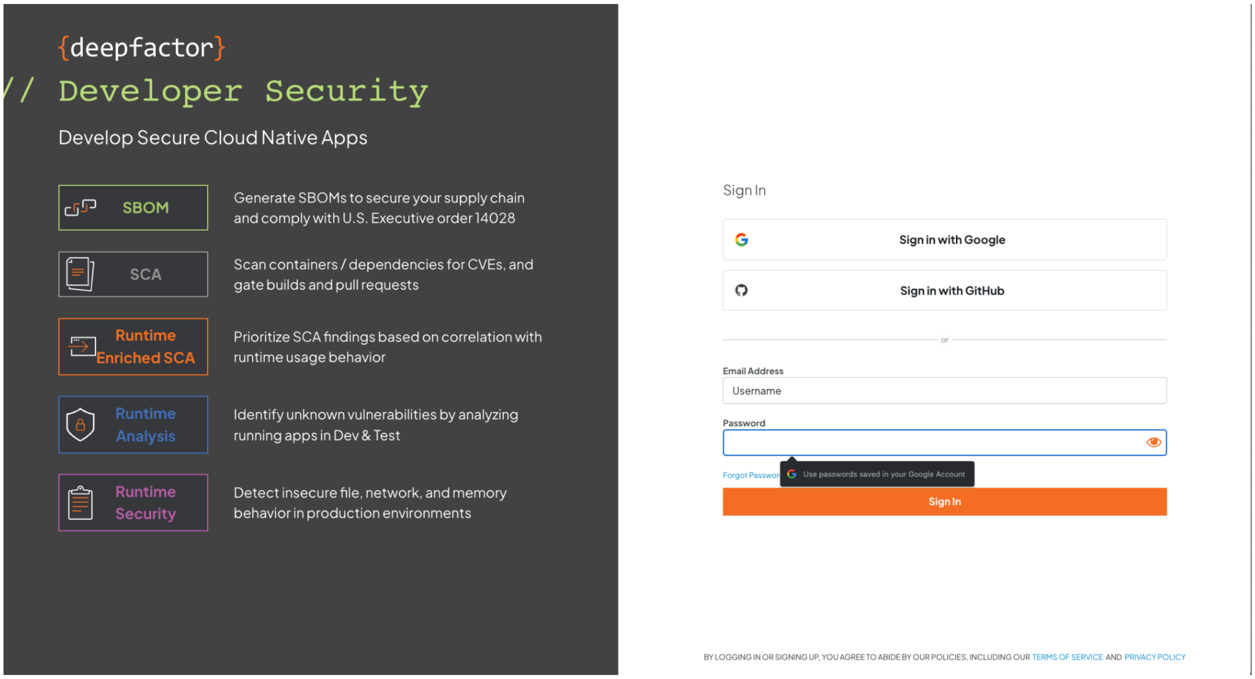
You are receiving this email because you registered for the 'Live workshop on SCA 2.0: Using runtime reachability analysis to prioritize SCA vulnerabilities' by Deepfactor, powered by TechStrong. If this wasn't you, please ignore this email.
Thank you

- The Deepfactor Team

# Workshop

## Step 1 - Login to Deepfactor Portal

Login to Deepfactor portal using credentials set during account activation. Following image is the login screenshot for reference



{deepfactor}

// Developer Security

Develop Secure Cloud Native Apps

SBOM — Generate SBOMs to secure your supply chain and comply with U.S. Executive order 14028

SCA — Scan containers / dependencies for CVEs, and gate builds and pull requests

Runtime Enriched SCA — Prioritize SCA findings based on correlation with runtime usage behavior

Runtime Analysis — Identify unknown vulnerabilities by analyzing running apps in Dev & Test

Runtime Security — Detect insecure file, network, and memory behavior in production environments

Sign In

Email Address: Username

Password:

Use passwords saved in your Google Account

BY LOGGING IN OR SIGNING UP, YOU AGREE TO ABIDE BY OUR POLICIES, INCLUDING OUR TERMS OF SERVICE AND PRIVACY POLICY

Deepfactor Platform Login screen

## Step 2 - Login to the test VM

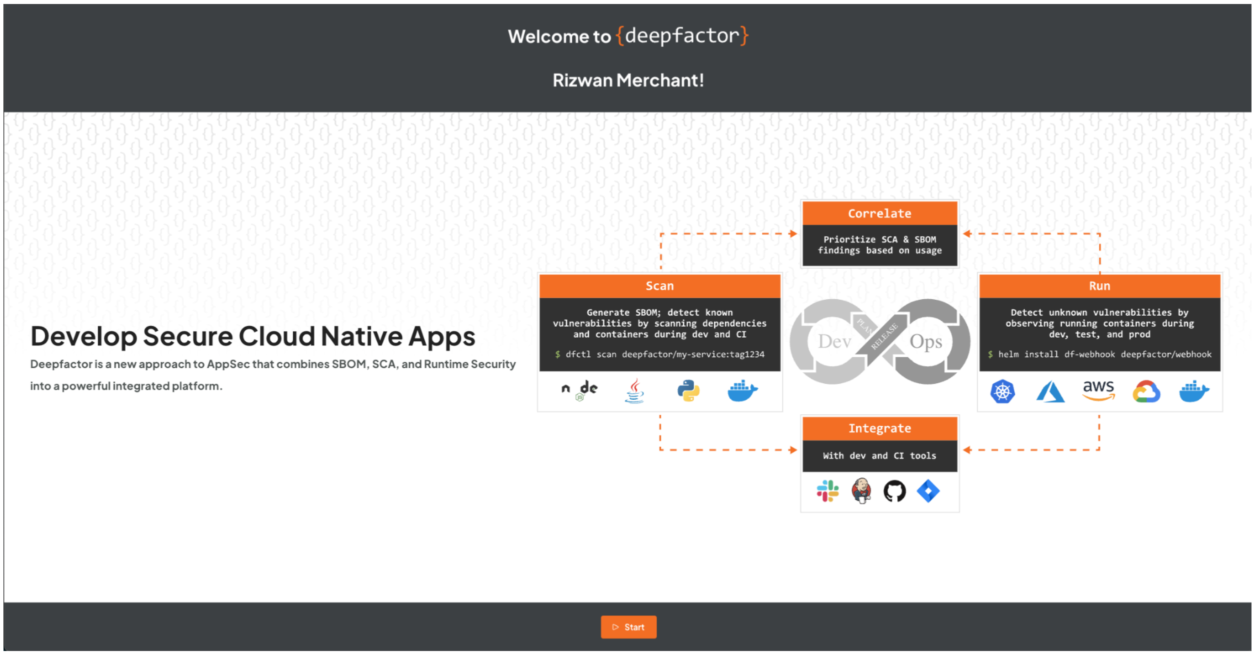
Login to the VM using the command and credentials from the registration email you received.

Following is a sample command

```
cmd #> ssh user007@ec2-52-53-190-106.us-west-1.compute.amazonaws.com
```

## Step 3 - Copy the Run Token

Step 3a - After successful login, click on the Start button at the bottom of the screen.



Welcome to {deepfactor}

Rizwan Merchant!

Develop Secure Cloud Native Apps

Deepfactor is a new approach to AppSec that combines SBOM, SCA, and Runtime Security into a powerful integrated platform.

Correlate

Prioritize SCA & SBOM findings based on usage

Scan

Generate SBOM; detect known vulnerabilities by scanning dependencies and containers during dev and CI

```
$ dfctl scan deepfactor/my-service:tag1234
```

Run

Detect unknown vulnerabilities by observing running containers during dev, test, and prod

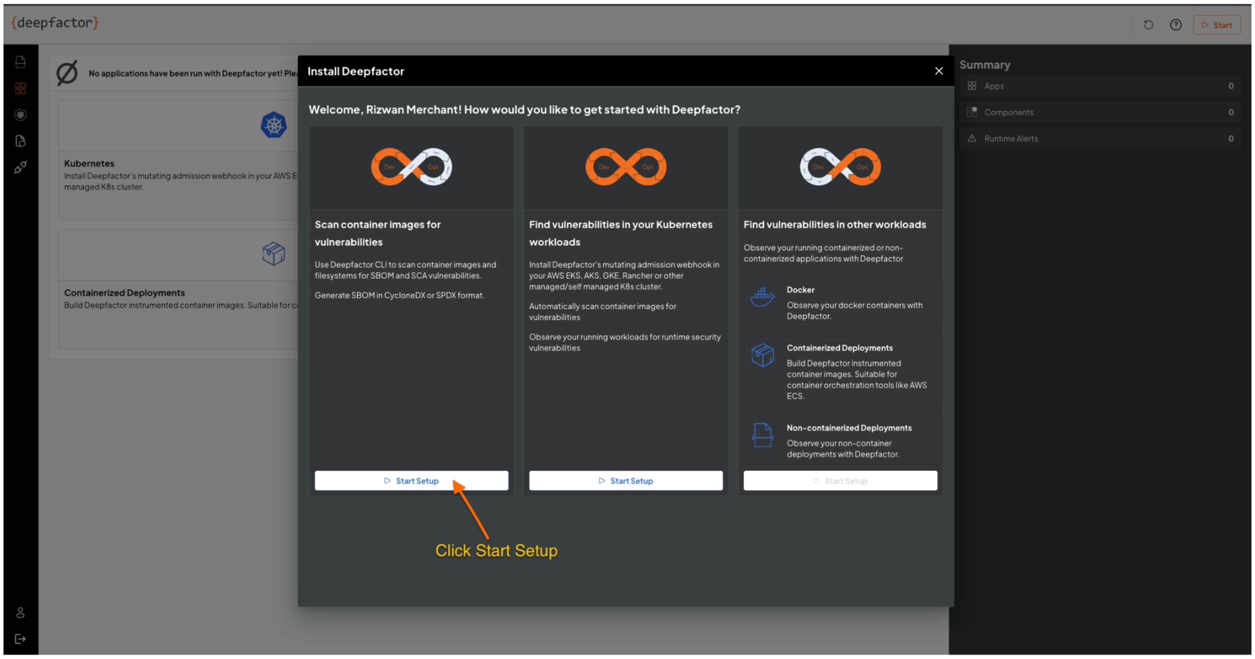
```
$ helm install df-webhook deepfactor/webhook
```

Integrate

With dev and CI tools

Step 3b

Click Start, will present three options. The first option is to Scan container images for vulnerabilities. Click Start Setup as shown below



No applications have been run with Deepfactor yet! Ple

Kubernetes

Install Deepfactor's mutating admission webhook in your AWS E managed K8s cluster.

Containerized Deployments

Build Deepfactor instrumented container images. Suitable for c

Install Deepfactor

Welcome, Rizwan Merchant! How would you like to get started with Deepfactor?

Scan container images for vulnerabilities

Use Deepfactor CLI to scan container images and filesystems for SBOM and SCA vulnerabilities.

Generate SBOM in CycloneDX or SPDX format.

▷ Start Setup

Find vulnerabilities in your Kubernetes workloads

Install Deepfactor's mutating admission webhook in your AWS EKS, AKS, GKE, Rancher or other managed/self managed K8s cluster.

Automatically scan container images for vulnerabilities

Observe your running workloads for runtime security vulnerabilities

▷ Start Setup

Find vulnerabilities in other workloads

Observe your running containerized or non-containerized applications with Deepfactor

Docker

Observe your docker containers with Deepfactor.

Containerized Deployments

Build Deepfactor instrumented container images. Suitable for container orchestration tools like AWS ECS.

Non-containerized Deployments

Observe your non-container deployments with Deepfactor.

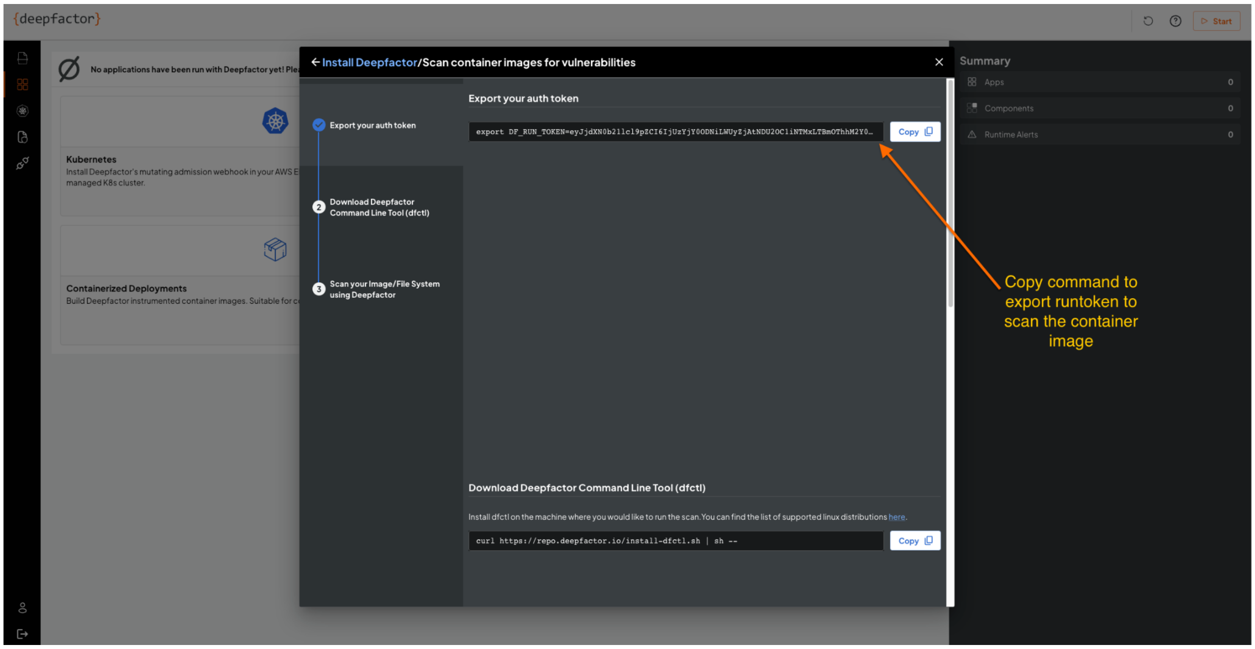
▷ Start Setup

Click Start Setup

| Summary | |
|---|---|
| Apps | 0 |
| Components | 0 |
| Runtime Alerts | 0 |

## Step 3c

## Copy command to export run token to scan container image



{deepfactor}
Start
No applications have been run with Deepfactor yet! Plea
Kubernetes
Install Deepfactor's mutating admission webhook in your AWS E
managed K8s cluster.
Containerized Deployments
Build Deepfactor instrumented container images. Suitable for c
Summary
Apps 0
Components 0
Runtime Alerts 0

← Install Deepfactor/Scan container images for vulnerabilities

✓ Export your auth token
2 Download Deepfactor Command Line Tool (dfctl)
3 Scan your Image/File System using Deepfactor

Export your auth token

```
export DF_RUN_TOKEN=eyJjdXN0b21lcl9pZCI6IjUzYjY0ODNiLWUyZjAtNDU2OC1iNTMxLTBmOThhM2Y0…
```

Download Deepfactor Command Line Tool (dfctl)

Install dfctl on the machine where you would like to run the scan. You can find the list of supported linux distributions here.

```
curl https://repo.deepfactor.io/install-dfctl.sh | sh --
```

Copy command to export runtoken to scan the container image

## Step 4 - Scan container image

After setting the run token using the export command below, you may scan container image using the `dfctl scan` command

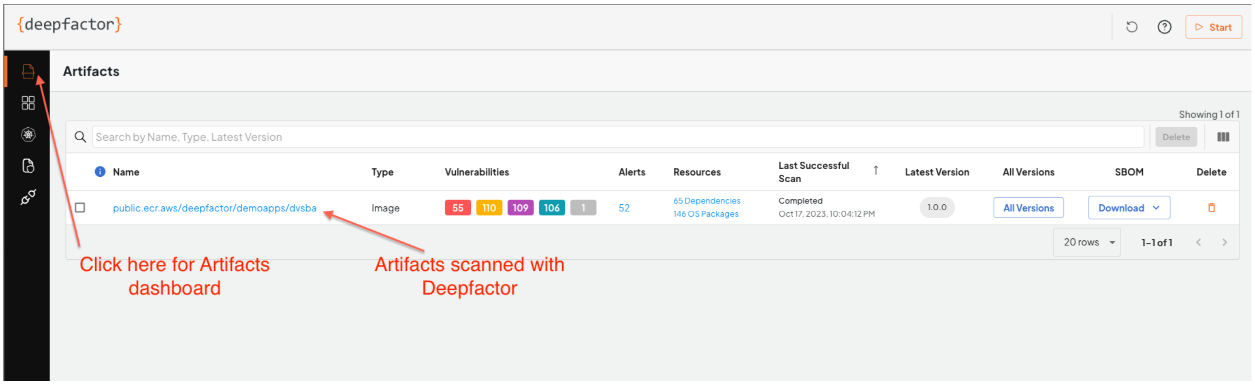
```
cmd #> export DF_RUN_TOKEN=<Run Token From Your Account>

cmd #> dfctl scan public.ecr.aws/deepfactor/demoapps/dvsba:1.0.0
```

Following is a sample output:

```
dfctl scan public.ecr.aws/deepfactor/demoapps/dvsba:1.0.0
Starting image scan
No match for registry type found
2023-10-18T03:42:33.882Z   info  successfully refreshed access token
2023-10-18T03:42:33.883Z   info  starting image scan...
2023-10-18T03:42:33.987Z   info  successfully registered scan agent
2023-10-18T03:42:33.988Z   info  artifact validation in progress...
2023-10-18T03:42:34.043Z   info  artifact validation done
2023-10-18T03:42:34.043Z   info  scan registration in progress...
2023-10-18T03:42:34.231Z   info  scan registration done
2023-10-18T03:42:34.231Z   info  scan in progress...
2023-10-18T03:42:34.292Z   info  scan complete
2023-10-18T03:42:34.301Z   info  Gathering exploit information
...
...
Deepfactor scan completed in 5 seconds.
```

After the scan completes you can check the Artifacts dashboard on Deepfactor portal for static SCA & SBOM of scanned artifacts. Following is sample screen capture of the dashboard



| | Name | Type | Vulnerabilities | Alerts | Resources | Last Successful Scan ↑ | Latest Version | All Versions | SBOM | Delete |
|---|---|---|---|---|---|---|---|---|---|---|
| □ | public.ecr.aws/deepfactor/demoapps/dvsba | Image | 55 110 109 106 1 | 52 | 65 Dependencies<br>146 OS Packages | Completed<br>Oct 17, 2023, 10:04:12 PM | 1.0.0 | All Versions | Download | 🗑 |

Click here for Artifacts dashboard

Artifacts scanned with Deepfactor

## Step 5 - Run the application

Run the application using the following command. Make sure your run token is set before you run your application

```
cmd #> dfctl run -a "vuln-spring-boot-app" -c "java" --docker-run -d -name
vuln-spring-boot-app --image public.ecr.aws/deepfactor/demoapps/dvsba:1.0.0
```

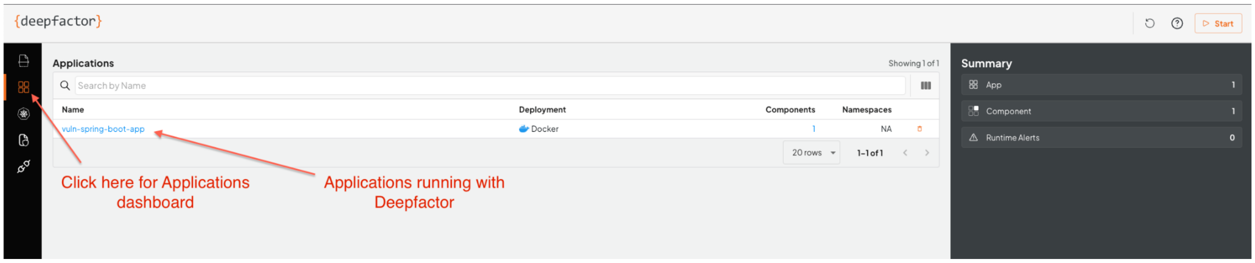
Following is sample output of this command

```
test: dfctl version:       "3.3.3-r2346" "6ef0f853418937e4d81ce89b88fbd9afb14f26f1"

test: dfctl: checking command line java

5ffe80a33767cccb0b920bcc0de49dd5f566e381b90b2aab246c11fedb2f5fe6
```

After the application starts up you can check the Applications dashboard on Deepfactor portal for runtime insights and alerts. Following is sample screen capture of the dashboard



| Name | Deployment | Components | Namespaces |
|---|---|---|---|
| vuln-spring-boot-app | Docker | 1 | NA |

| Summary | |
|---|---|
| App | 1 |
| Component | 1 |
| Runtime Alerts | 0 |

Click here for Applications dashboard

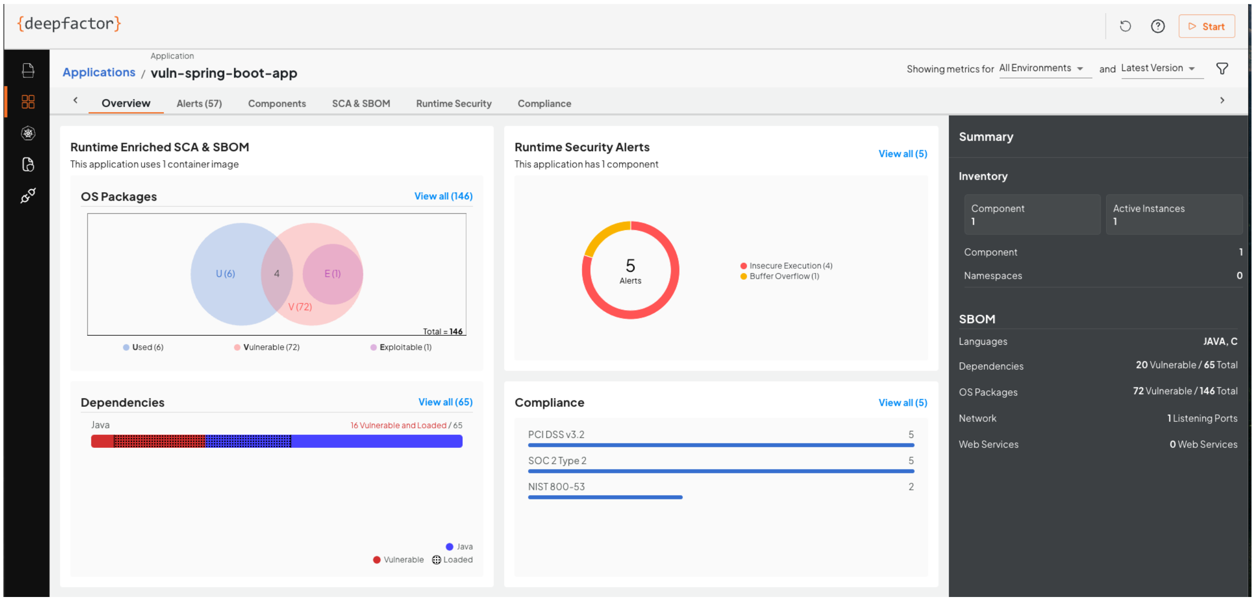
Applications running with Deepfactor

## Step 6 - Exercise your application

Run additional command on your running container

```
cmd #> docker exec -it vuln-spring-boot-app /bin/bash
root@xyz #> find /
```

Following is sample dashboard after the running the above command in the container running with Deepfactor



**Runtime Enriched SCA & SBOM**

This application uses 1 container image

**OS Packages** View all (146)

**Runtime Security Alerts** View all (5)

This application has 1 component

**Dependencies** View all (65)

Java 16 Vulnerable and Loaded / 65

**Compliance** View all (5)

| | |
|---|---|
| PCI DSS v3.2 | 5 |
| SOC 2 Type 2 | 5 |
| NIST 800-53 | 2 |

**Summary**

**Inventory**

| Component | Active Instances |
|---|---|
| 1 | 1 |

| | |
|---|---|
| Component | 1 |
| Namespaces | 0 |

**SBOM**

| | |
|---|---|
| Languages | JAVA, C |
| Dependencies | 20 Vulnerable / 65 Total |
| OS Packages | 72 Vulnerable / 146 Total |
| Network | 1 Listening Ports |
| Web Services | 0 Web Services |