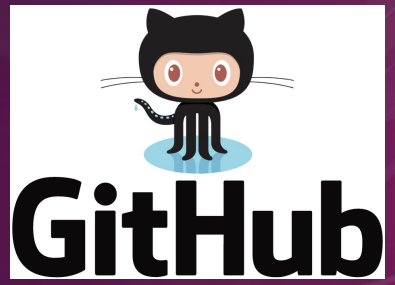
DevSecOps - Background, Status and Future Challenges

F. Ozgur Catak

University of Stavanger

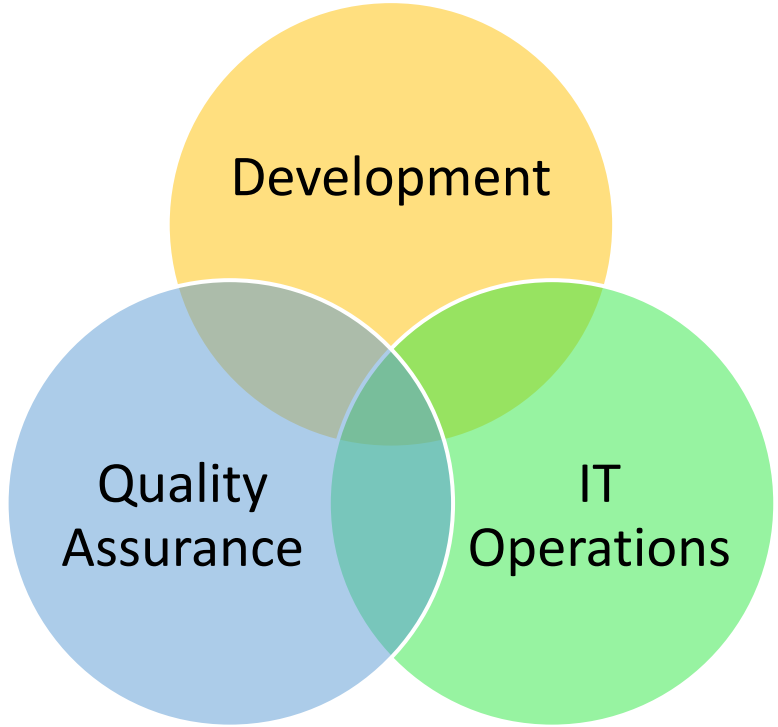
https://github.com/ocatak/devsecops-tutorial



DevOps

A philosophy, aims at building up a culture of collaboration between originally isolated teams.

Departments
- Software development
- deployment operations



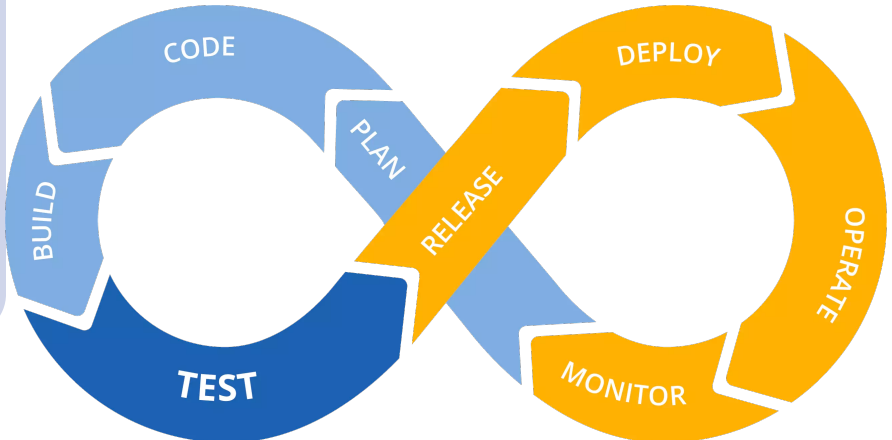
Improving the efficiency by eliminating the boundaries between these two phases of development.

Includes

Continuous integration, where the coding, building, integration, and testing processes are carried out.

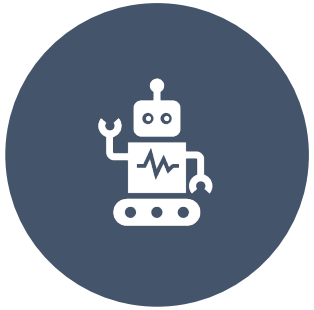
Continuous delivery, which includes continuous integration but mainly focuses on product delivery.

Continuous deployment, which aims at automating project deliveries.



Defining DevOps: C-A-M-S

CULTURE



AUTOMATION

MEASUREMENT

SHARING

Culture

- Primary charactheristics of DevOps culture: Increased collaborations
- Dev-Ops interact for a single target: to break down the barries

INTERACTION

TARGET

TEAMS

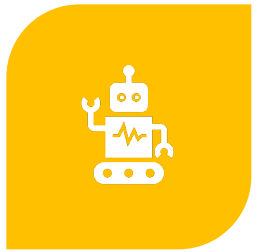
DEVELOPMENT - OPERATION

Automation

- Keyword: automation
- Speed-up feedback between teams
- The app will be delivered to the customers easily and successfully.
- DevOps pipeline: flow of information
- New functionality in the app.
  - New library, New attack surface, SQLi?

FEEDBACK

PROCESS



TOOLS FOR COLLABORAION

Measurement

- Track improvement
- See bottlenecks/problems easily
- Performance of the production system
  - Production system didn’t degraded anything like response time,

PERFORMANCE

MEASUREMENT

BOTTLENECKS

Sharing

- Sharing ideas, problems, and solutions between various teams
- By transfering the knowledge, the collective intelligence increases and benefits for everyone.

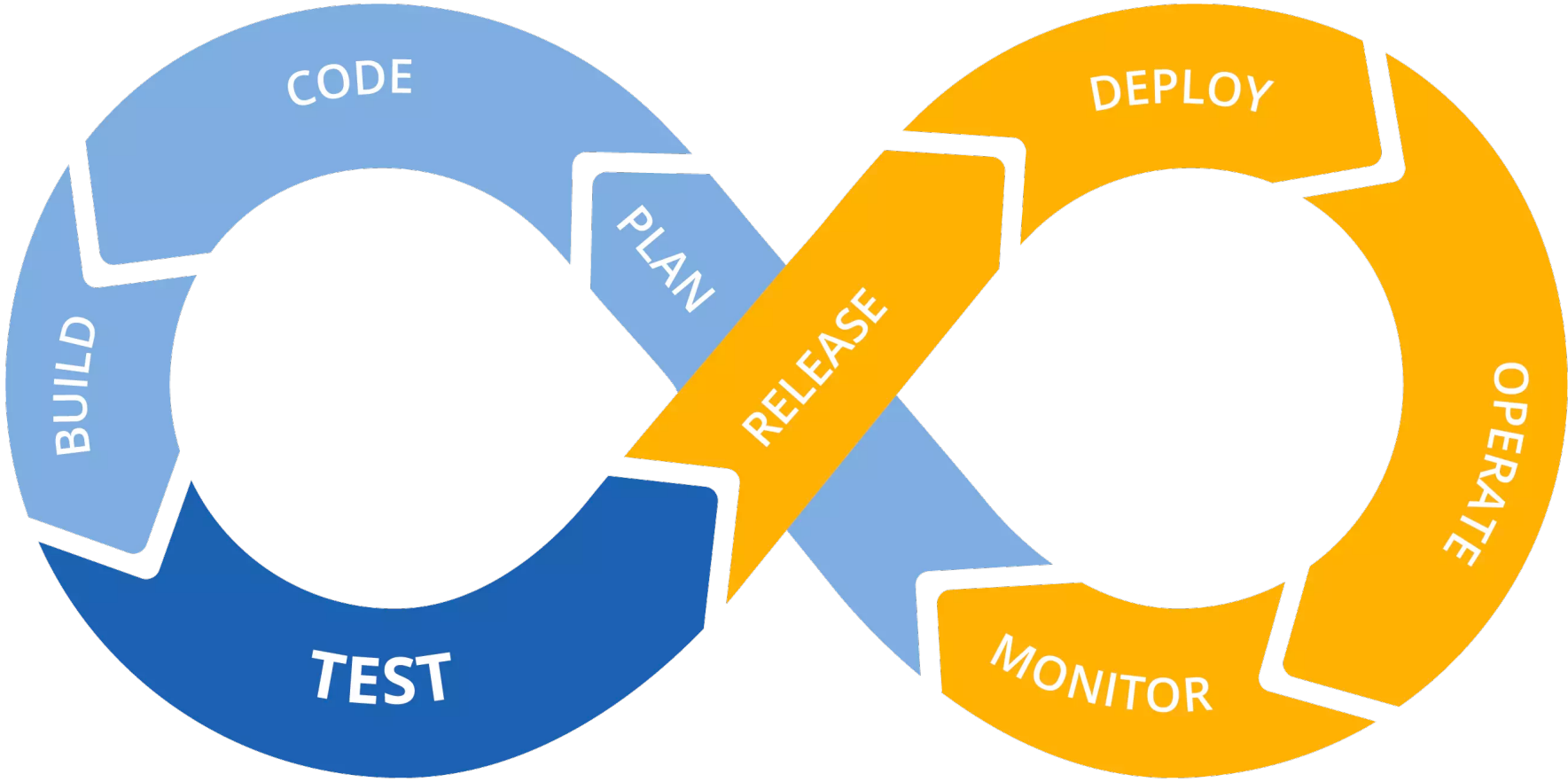
IDEA

SHARING

TALK

DevOps Lifecycle

- 8 stages in the
  - Planing
  - Development
  - Operation



DevOps Tools and Technologies

Version control systems: Git, SVN, Mercurial, etc.

Continuous integration (CI): Jenkins and TeamCity

Containers: Docker or Vagrant.

Framework automation tools: Python, Shell, or Bash.

Cloud services: Microsoft Azure, Google Cloud, Amazon Web Services

Traditional Security Perspective: Penetration testing

WHAT IS A **PENETRATION TEST?**

An authorized attack on a computer system, network, or application to identify security vulnerabilities bad actors might exploit.

The Process

1. INFORMATION GATHERING
2. THREAT MODELING
3. VULNERABILITY ANALYSIS
4. EXPLOITATION
5. POST-EXPLOITATION
6. REPORTING

Types of Penetration Tests

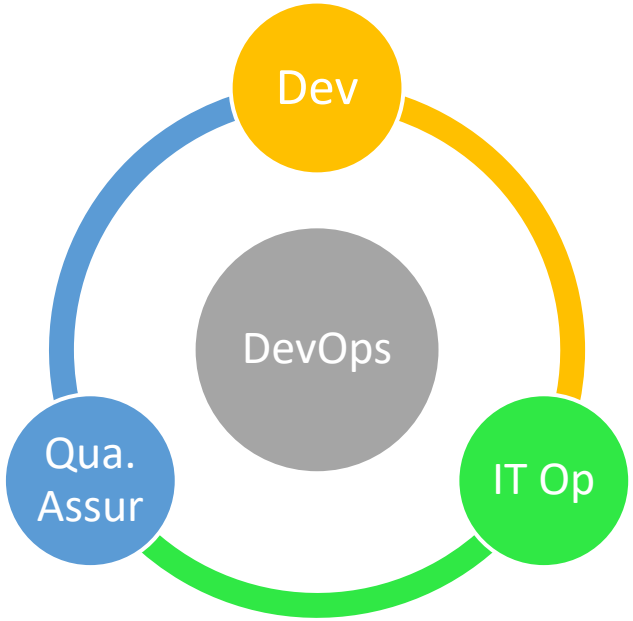
- Application
- Network
- Physical
- IoT/Mobile

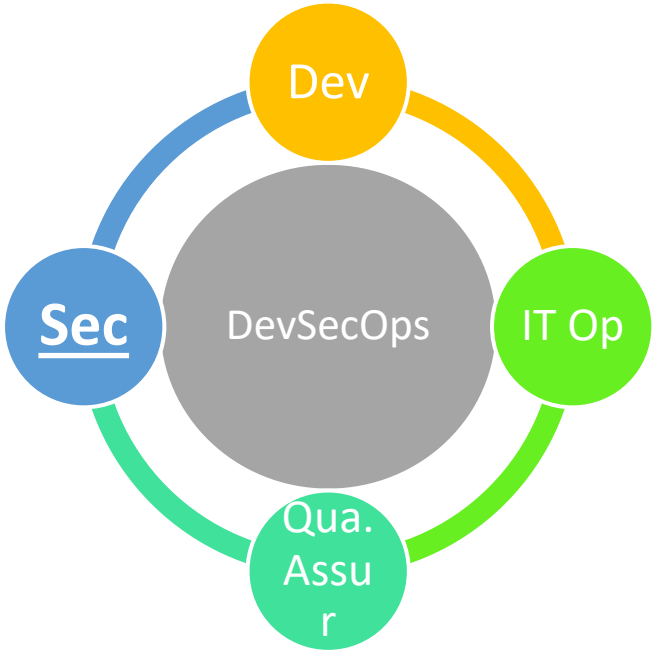
Why Conduct a Penetration Test?

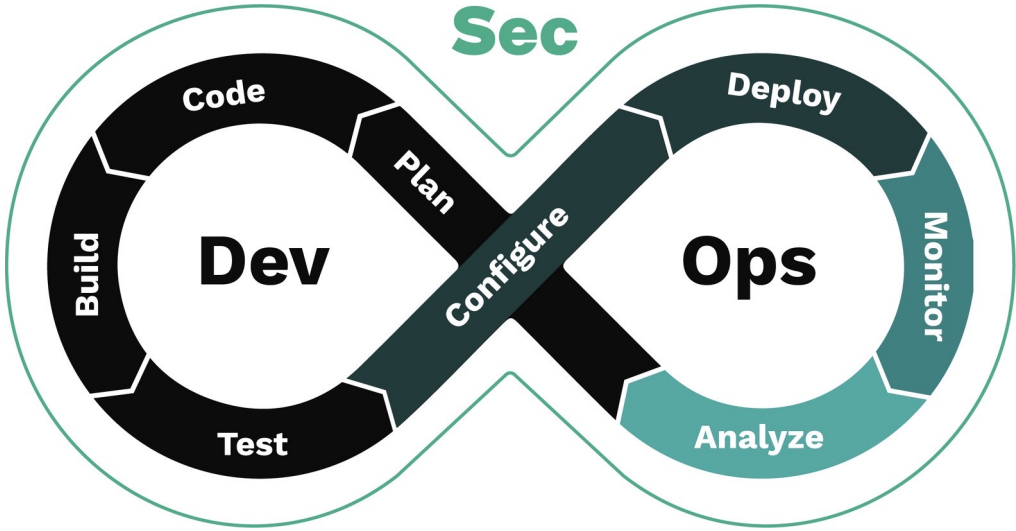
- Identify security vulnerabilities
- Validate compliance with policies
- Evaluate effectiveness of defenses

DevSecOps: what's and differences from DevOps?

- It is a philosophy of **integrating security methods** into a DevOps process.
- From the very start of a **SDLC**, DevSecOps works to make the application secure by introducing a variety of security techniques.
- DevOps heavily relies on automation. The same is true for DevSecOps, which aims at automating every aspect, including security audit.







Why DevSecOps

Companies implementing DevSecOps address vulnerabilities faster than others

Higher speed and agility: security teams

Quality and compliance assurance

Implement, Educate & Follow best practices for secure coding and infrastructure provisioning/ deployment / configuration.

DevSecOps: Securing CI/CD

Integrating security in the CI/CD pipeline which helps minimizing threats/vulnerabilities with every code check in.

Automation of security part.

- Minimize the manual Pentest effort.

To bridge the gap between Dev. Sec and Ops teams

DevSecOps: Upcoming hot buzz

As DevOps & agile enables speed, it comes with risk as well.

Frequent code changes, expanding attack surfaces, day by day growing number of hacks is building the market for DevSecOps

80% of threats & vulnerabilities can be tackled if we enable DevSecOps (20% effort). Famous 80/20 rules or “Pareto principle”

Cloud Security Alliance: The Six Pillars of DevSecOps: Automation

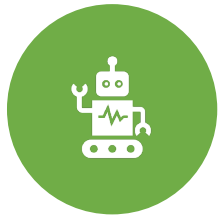
The **Cloud Security Alliance** (CSA) is the world's leading organization dedicated to defining and raising awareness of best practices to help ensure a **secure cloud** computing environment.

COLLECTIVE RESPONSIBILITY

COLLABORATION AND INTEGRATION

PRAGMATIC IMPLEMENTATION

BRIDGING COMPLIANCE AND DEVELOPMENT



AUTOMATION

MEASURE, MONITOR, REPORT AND ACTION

https://cloudsecurityalliance.org/press-releases/2020/07/07/cloud-security-alliance-publishes-new-paper-the-six-pillars-of-devsecops-automation/

Why do we need DevSecOps

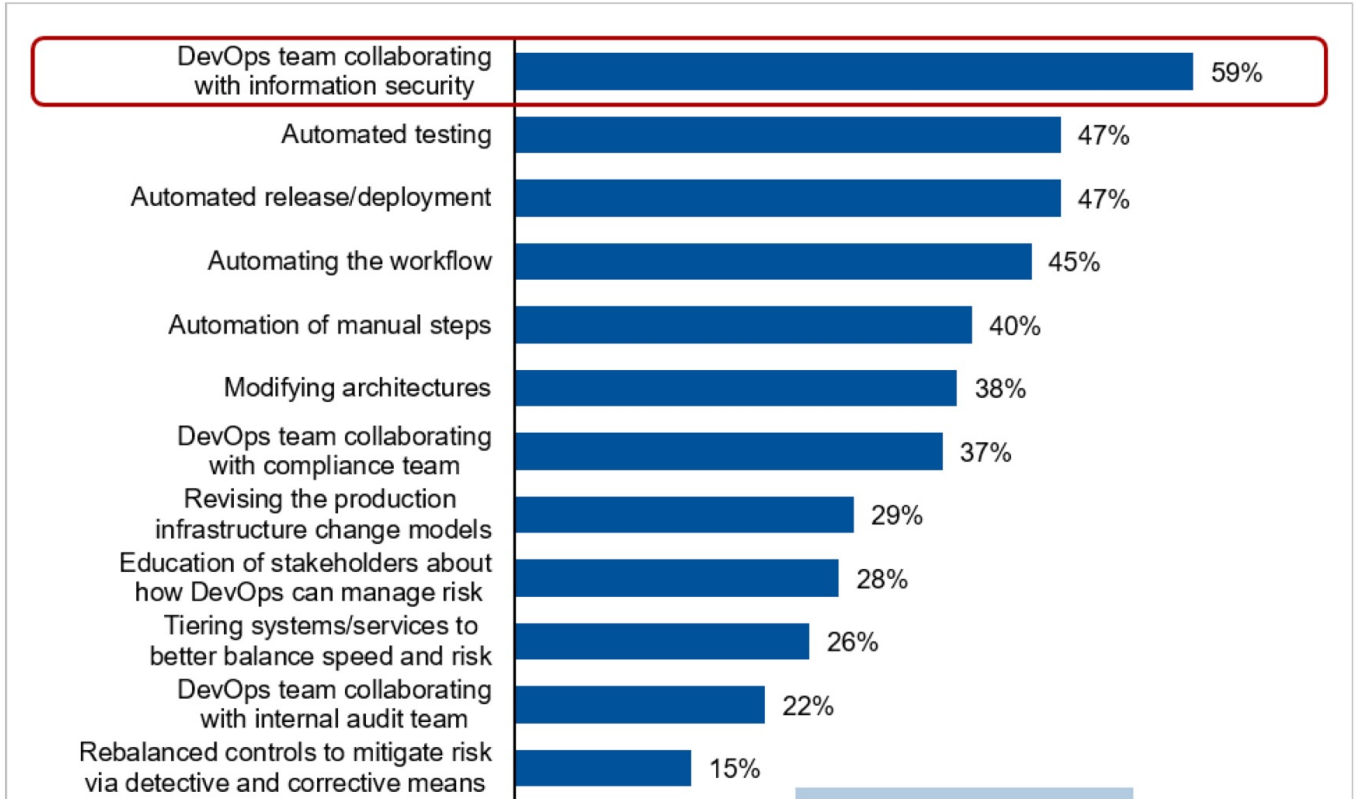
- implement and maintain the compliance reqirements
- minimize security bottlenecks also automate the security checks
- collaborate with Information security team in the upcoming future by 2021

MAINTAIN COMPLIANCE / REGULATORY REQUIREMENT

SECURITY AUTOMATION / MINIMIZE SECURITY BOTTLENECKS

Gartner Report: 10 Things to Get Right for Successful DevSecOps

Figure 1. Strategies to Overcome Hurdles to Using DevOps in Regulated Situations



Practical Transitioning from DevOps to DevSecOps

Static Application Security Testing (SAST)

- White-box security testing
  - Access to the underlying framework, design, and implementation
  - The app is tested from the inside out
  - It is **developer approach**
- Finds vulnerabilities earlier in the SDLC
- Less expensive to fix vulnerabilities
- Can’t discover run-time and environment related issues (like configuration vulnerabilities)

Dynamic Application Security Testing (DAST)

- Black-box security testing
  - The tester doesn’t know the technologies or frameworks of the app
  - The app is tested from the outside in
  - It is **hacker approach**
- Finds vulnerabilities toward the end of the SDLC
- More expensive to fix vulnerabilities
- Can discover run-time and environment related issues (like configuration vulnerabilities)

DevOps vs DevSecOps

- Traditional Build Phases
- Future Build Phases

devsecops-tutorial

DevSecOps best practices

View the Project on GitHub
ocatak/devsecops-tutorial

DevSecOps Practical Implementation

https://github.com/ocatak/devsecops-tutorial

http://www.ozgurcatak.org/devsecops-tutorial/

This project is maintained by ocatak

Hosted on GitHub Pages — Theme by orderedlist

DevSecOps - Background, Status and Future Challenges

- F. Ozgur Catak - f.ozgur.catak@gmail.com
- GitHub repository for the lecture at **University of Stavanger**
- June 14th, 2021

Jenkins Installation

we can run Jenkins as Docker image

```
docker pull jenkins
```

Contributing

Pull requests are welcome. For major changes, please open an issue first to discuss what you would like to change.

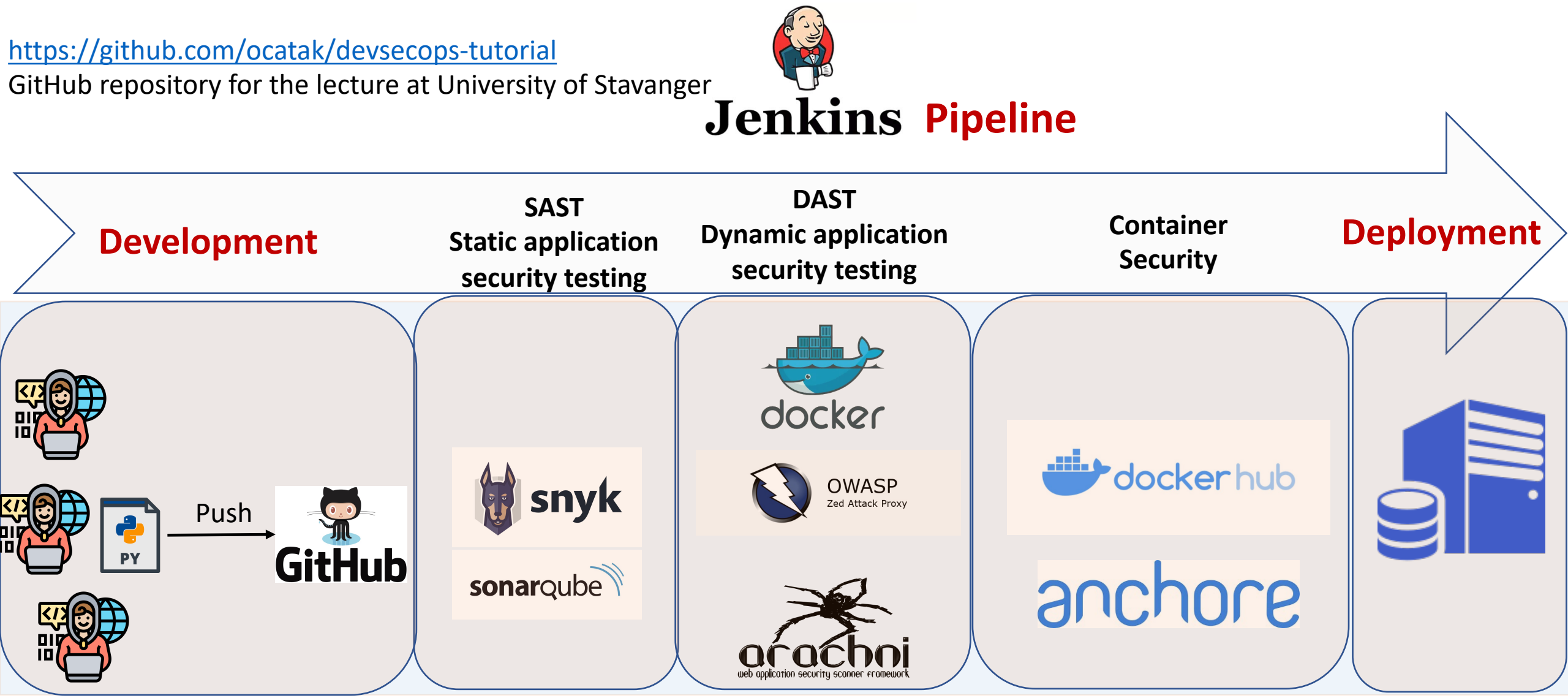
Please make sure to update tests as appropriate.

License

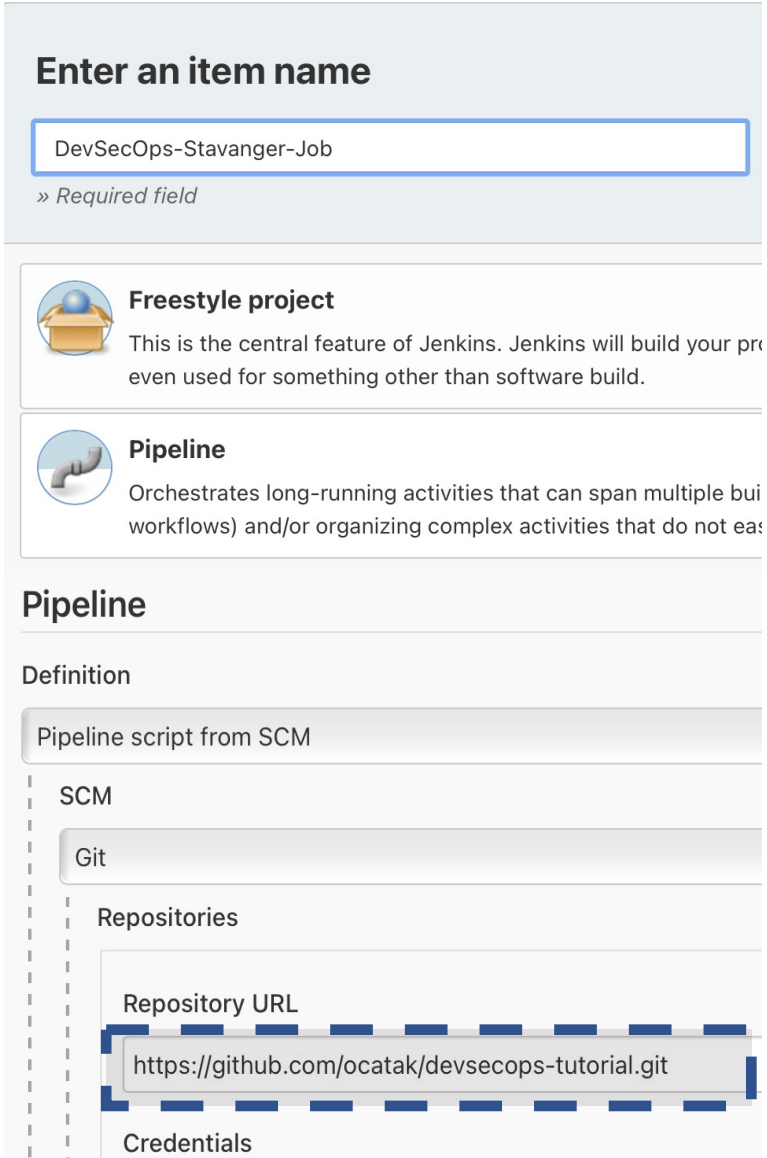
MIT

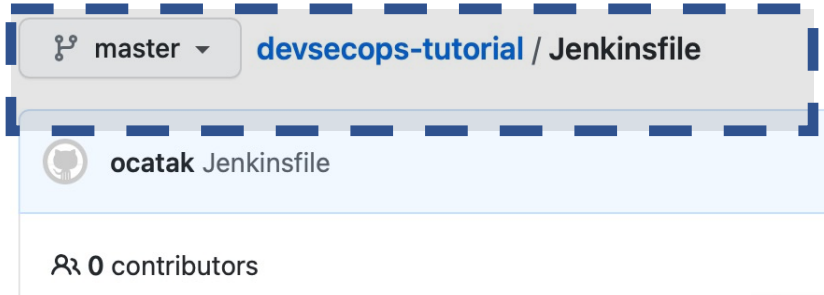
Example Web Application: DevSecOps

https://github.com/ocatak/devsecops-tutorial
GitHub repository for the lecture at University of Stavanger



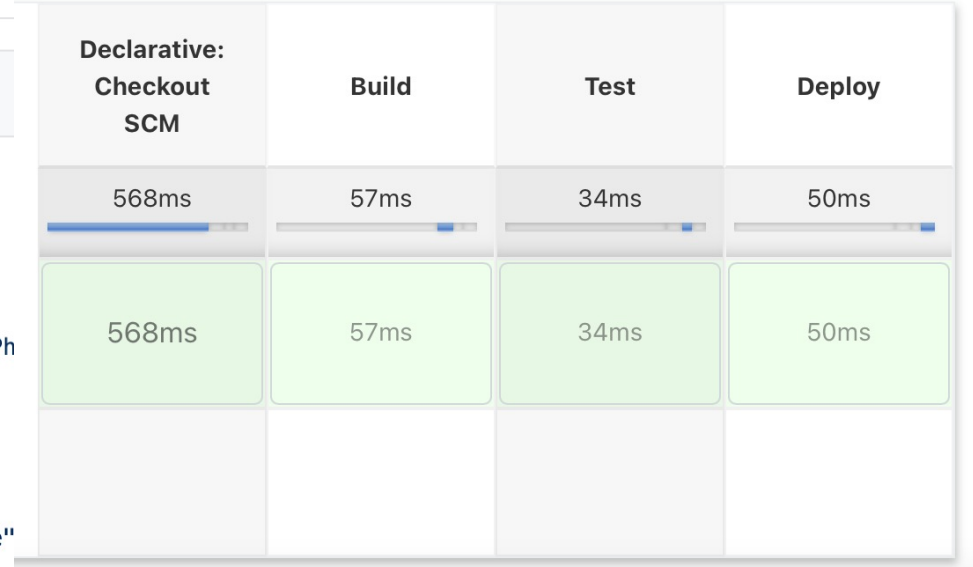
Jenkins: Pipeline Creation





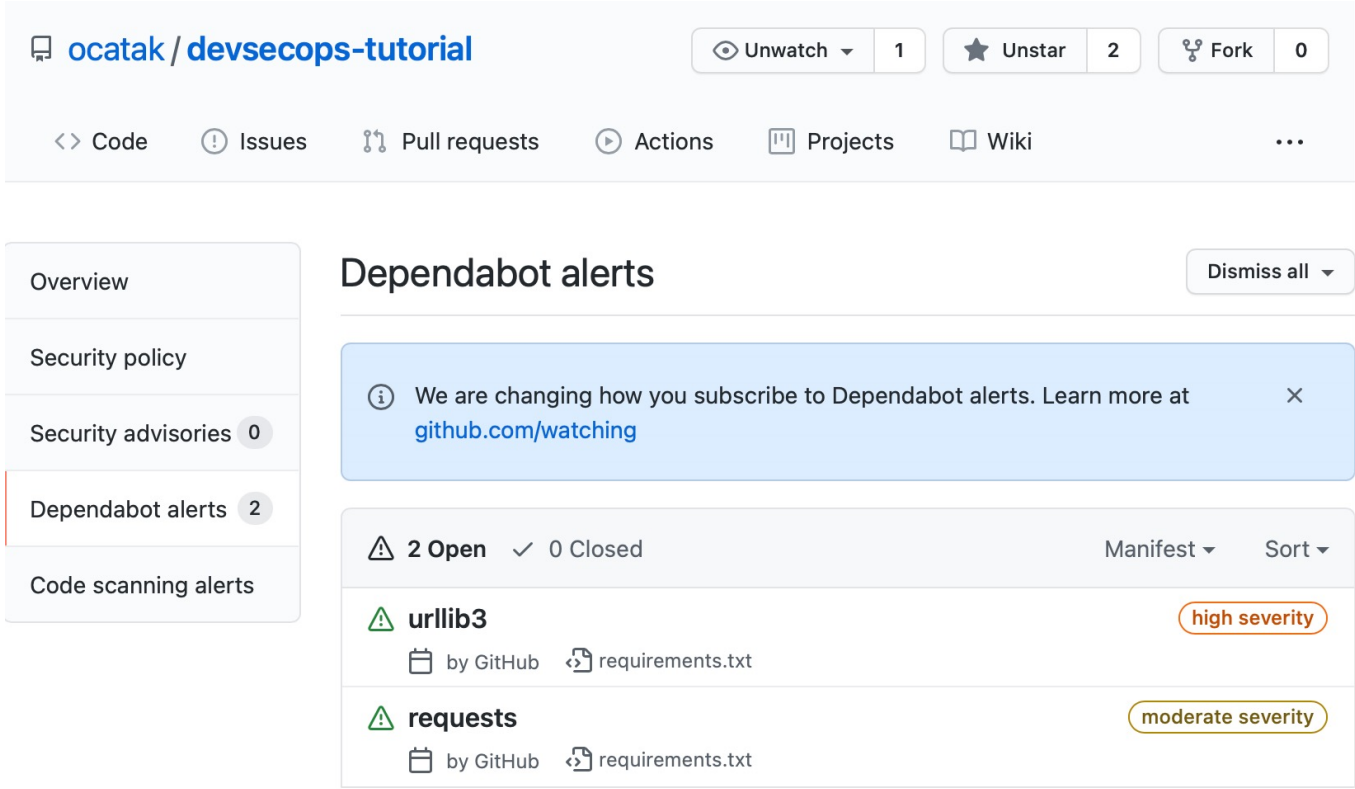
21 lines (20 sloc) | 267 Bytes

```
1  pipeline{
2    agent any
3    stages {
4          stage ('Build') {
5                  steps {
6                          echo "Build Ph
7                  }
8          }
9          stage ('Test') {
10             steps{
11                     echo "Test Phase"
12                 }
13         }
14         stage ('Deploy') {
15               steps{
16                       echo "Deploy Phase"
17                   }
18           }
```



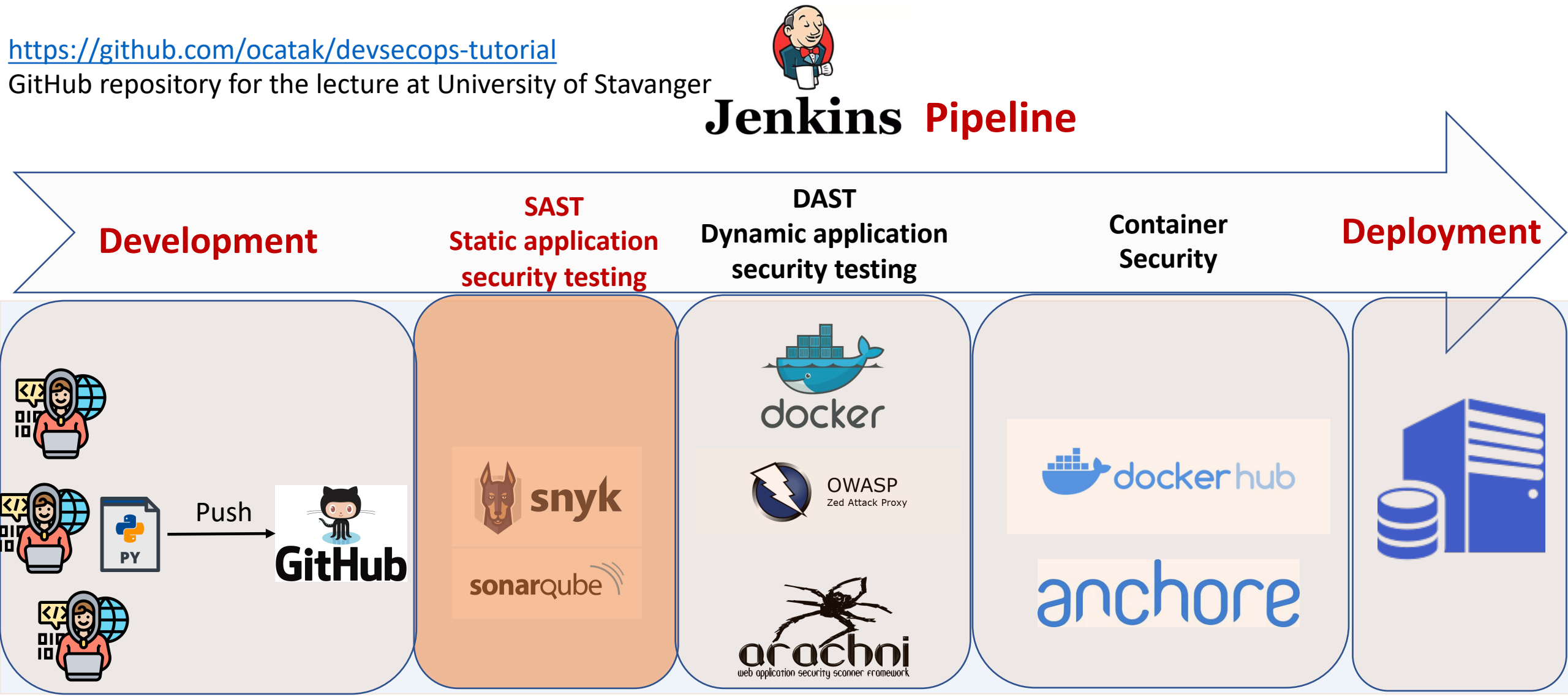
Source Code Security (GitHub)

- Browsing security vulnerability in the *GitHub Advisory DB*
  - *https://github.com/advisories* : It allows you to browse or search for vulnerabilities which affect open source projects on GitHub
  - Below sources are used for vulnerabilities by GitHub
    - The National Vulnerability Database
    - A combination of machine learning and human review to detect vulnerabilities in public commits on GitHub
    - Security advisories on GitHub
    - FriendsOfPHP
- Security alerts for vulnerable dependencies
- Automated security updates
- Update vulnerable dependencies in repository
- Managing alerts for vulnerable dependencies in your organization



Example Web Application: DevSecOps

https://github.com/ocatak/devsecops-tutorial
GitHub repository for the lecture at University of Stavanger



DevSecOps: SAST - SonarQube

It is an open-source tool used for continuous inspection of code quality, static analysis of code to detect bugs, code smells and security vulnerabilities on 20+ programming languages.

It provides reports on duplicated lines of code, coding standards, unit tests, coverage, code complexity, comments, bugs etc.

Benefits of SonarQube

Reduces risks in software development within a short span of time, detects bugs in the code and automatically alerts developers.

Receives all files as input , stores them in a database, analyses them with barriers and displays it in a dashboard.

It increases the productivity of Dev team by detecting duplicated & redundant code.

Dev Team receive regular feedback on quality issues and it helps them increase their programming skills.

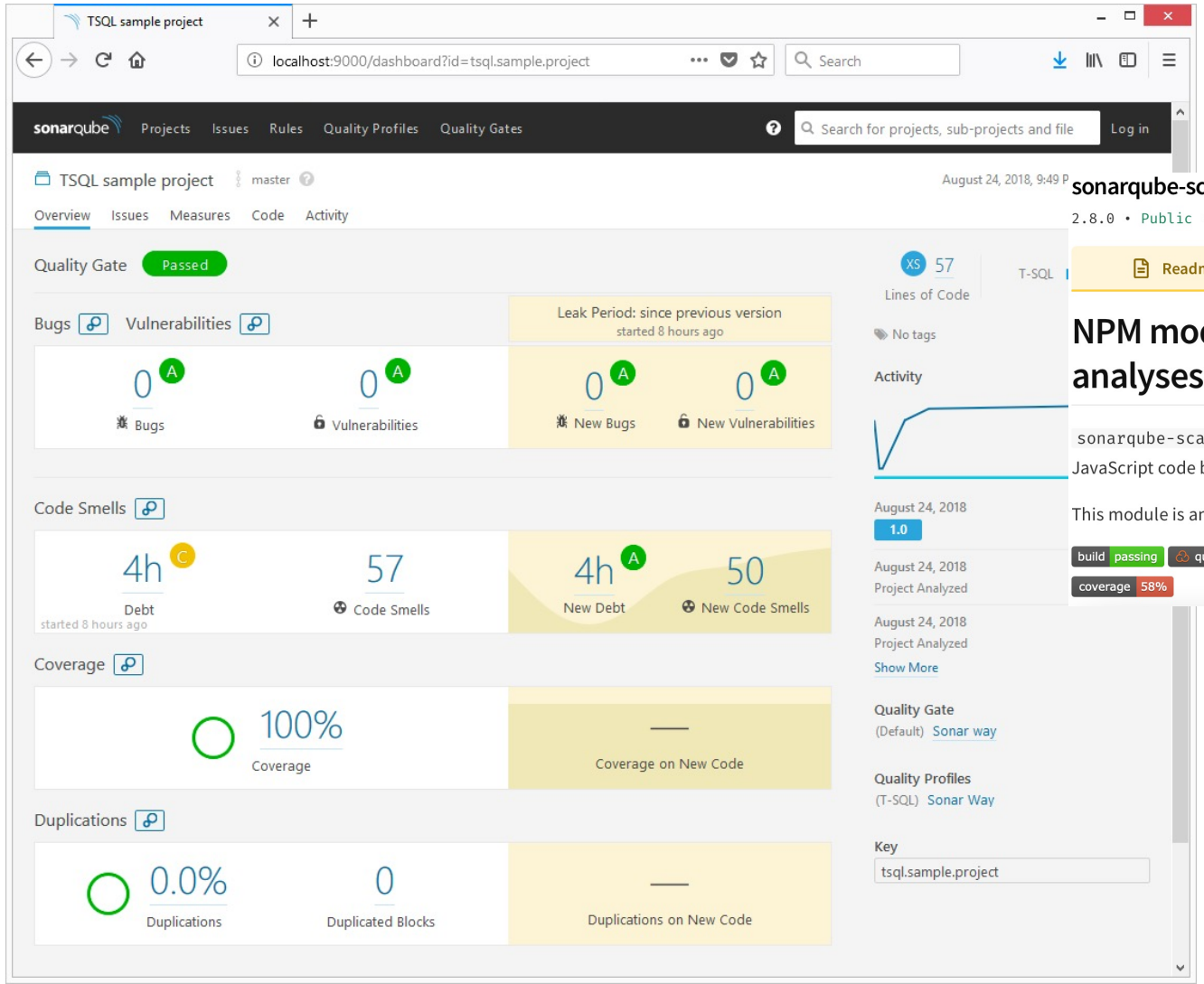
Code quality becomes part of CI/CD & Development team.

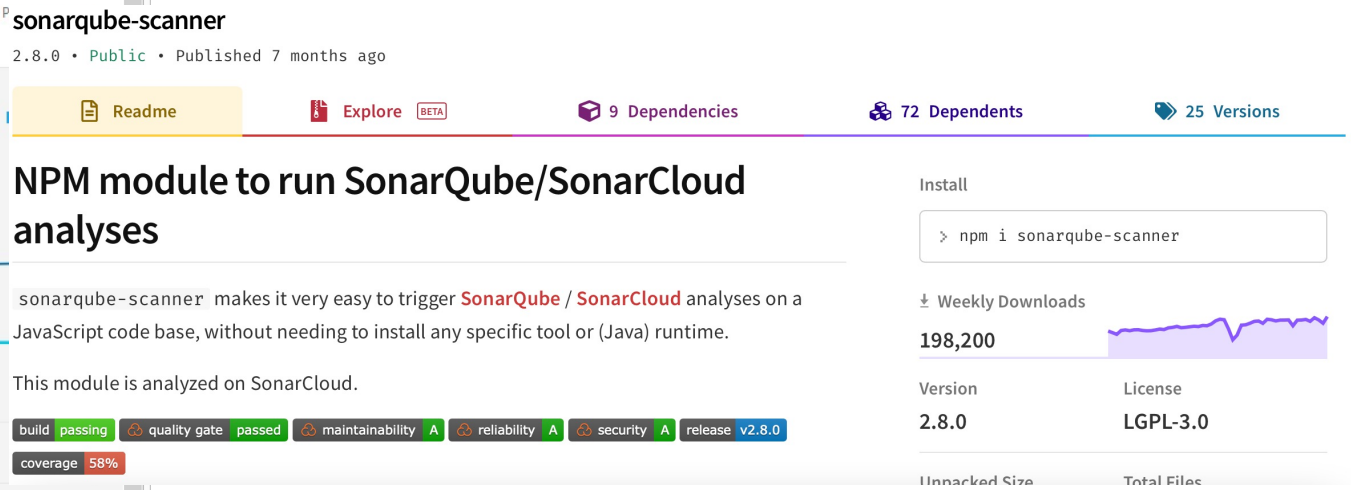
Security flaws detection & remediation.

DevSecOps: SAST - SonarQube

1. *git clone https://github.com/ocatak/SonarQube.git*
2. *sudo npm install sonarqube-scanner*
3. Run SonarQube Docker image: *docker-compose up –d*
4. *GitHub: sonar-project.js*

DevSecOps: SAST - SonarQube

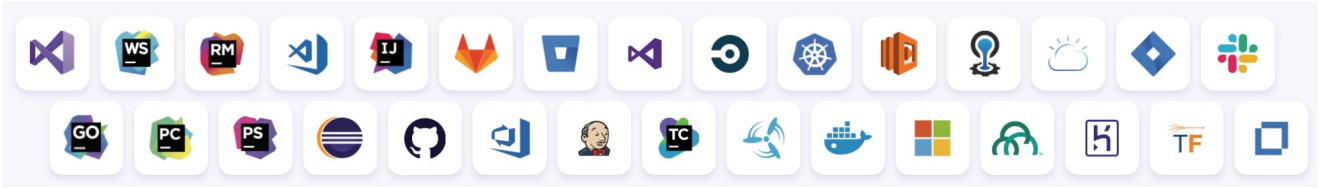




In Jenkins, add a new item “Execute Shell”
npm run sonar

DevSecOps: SAST - SNYK

- Lots of Multi-Ntional Companies are protected by SNYK
  - Google
  - Microsoft
  - Salesforce
- You can integrate it to your DevSecOps pipeline
  - Install snyk jenkins pluggin
  - Create an account in snyk
  - Create a token in snyk
  - create a synk job and input the git url for scanning,
  - add it to build step.
  - *npm install -g snyk*
  - *synk auth*



DevSecOps: SAST - SNYK

Execute shell

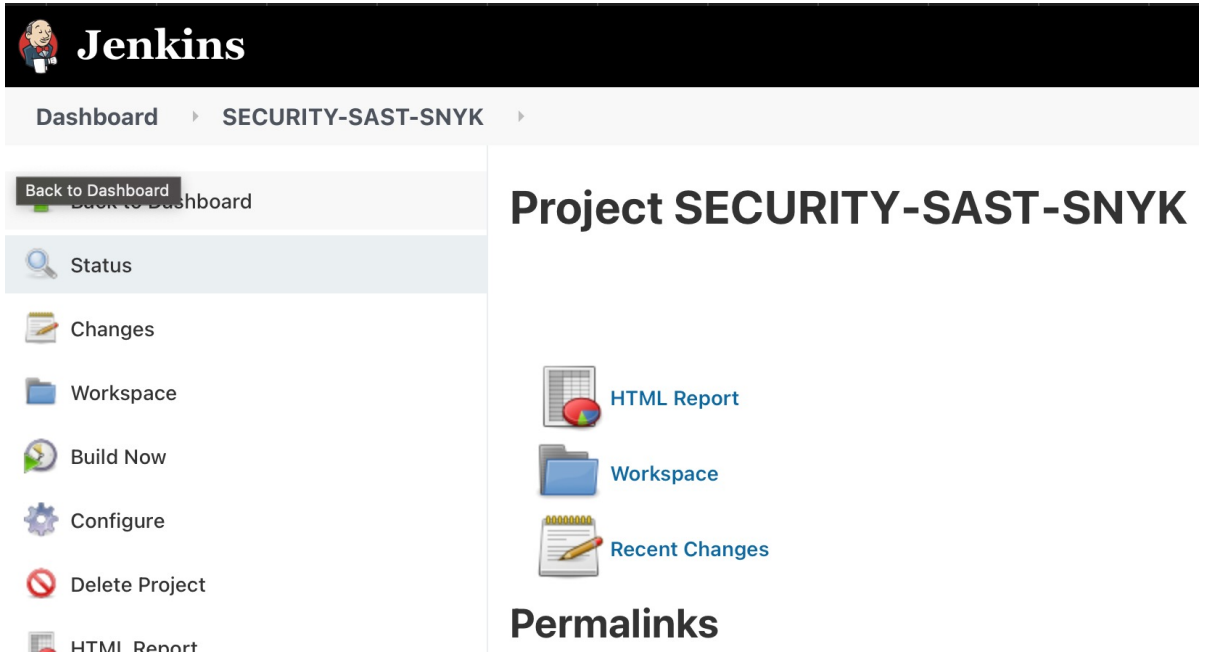
Command

```
cd ${WORKSPACE}
echo "snyk test && snyk monitor" > snyk.sh
chmod +x snyk.sh
/bin/bash snyk.sh || true
synk test --json | snyk-to-html -o results.html
```

See the list of available environment variables

26 lines (23 sloc) | 341 Bytes

```
1  pipeline{
2    agent any
3    stages {
4
5        stage('SAST') {
6           build 'SECURITY-SAST-SNYK'
7        }
8
9          stage ('Build') {
10                        steps {
11                                echo "Build Phase"
12                    }
13                }
14                stage ('Test') {
```



HIGH SEVERITY

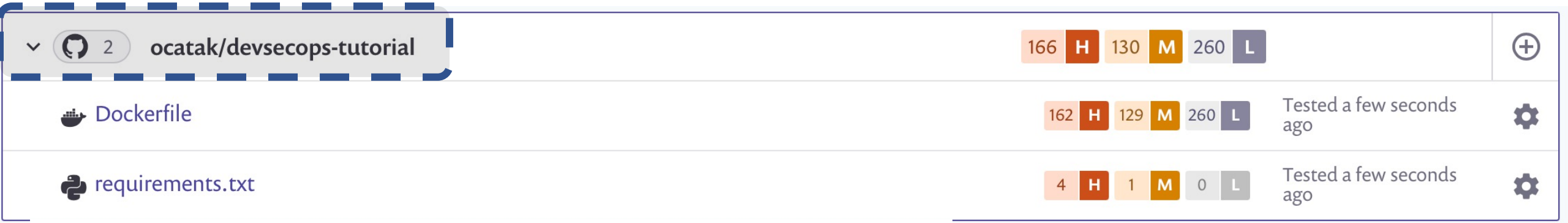
HTTP Header Injection

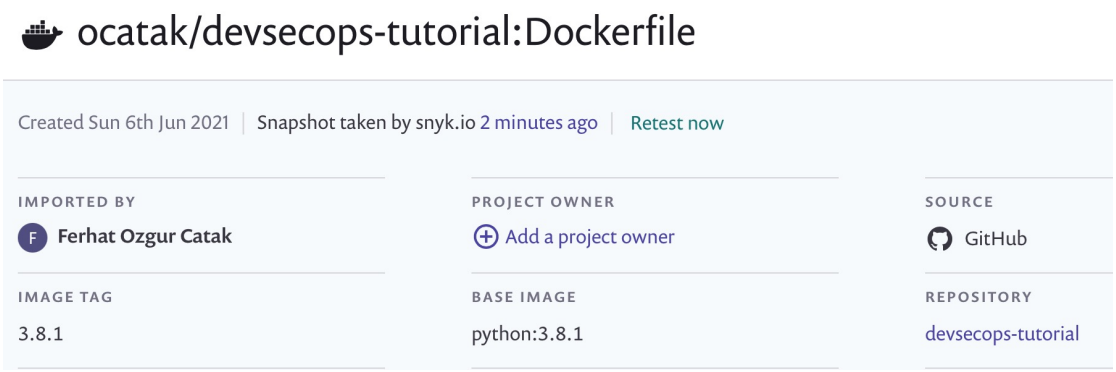
- Package Manager: pip
- Vulnerable module: urllib3
- Introduced through: devsecops-example@0.0.0 and urllib3@1.13.1

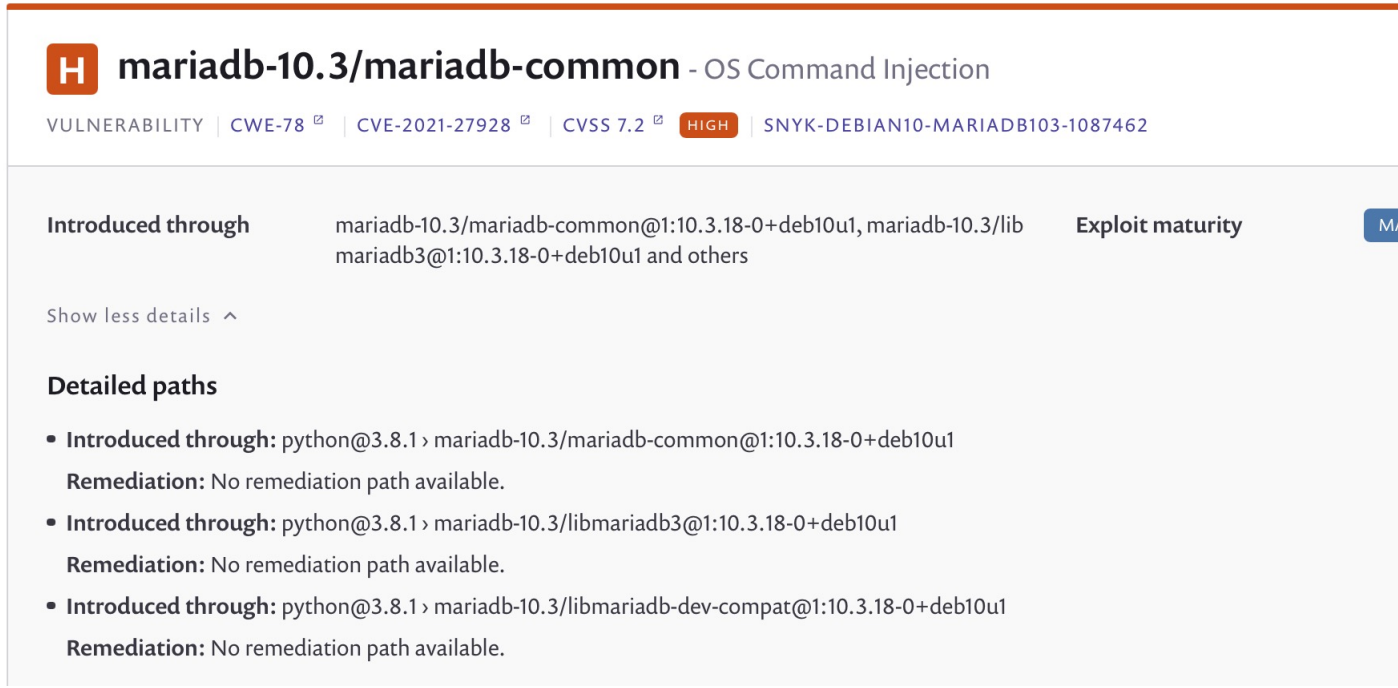
Detailed paths

- *Introduced through*: devsecops-example@0.0.0 › urllib3@1.13.1

DevSecOps: SAST - SNYK

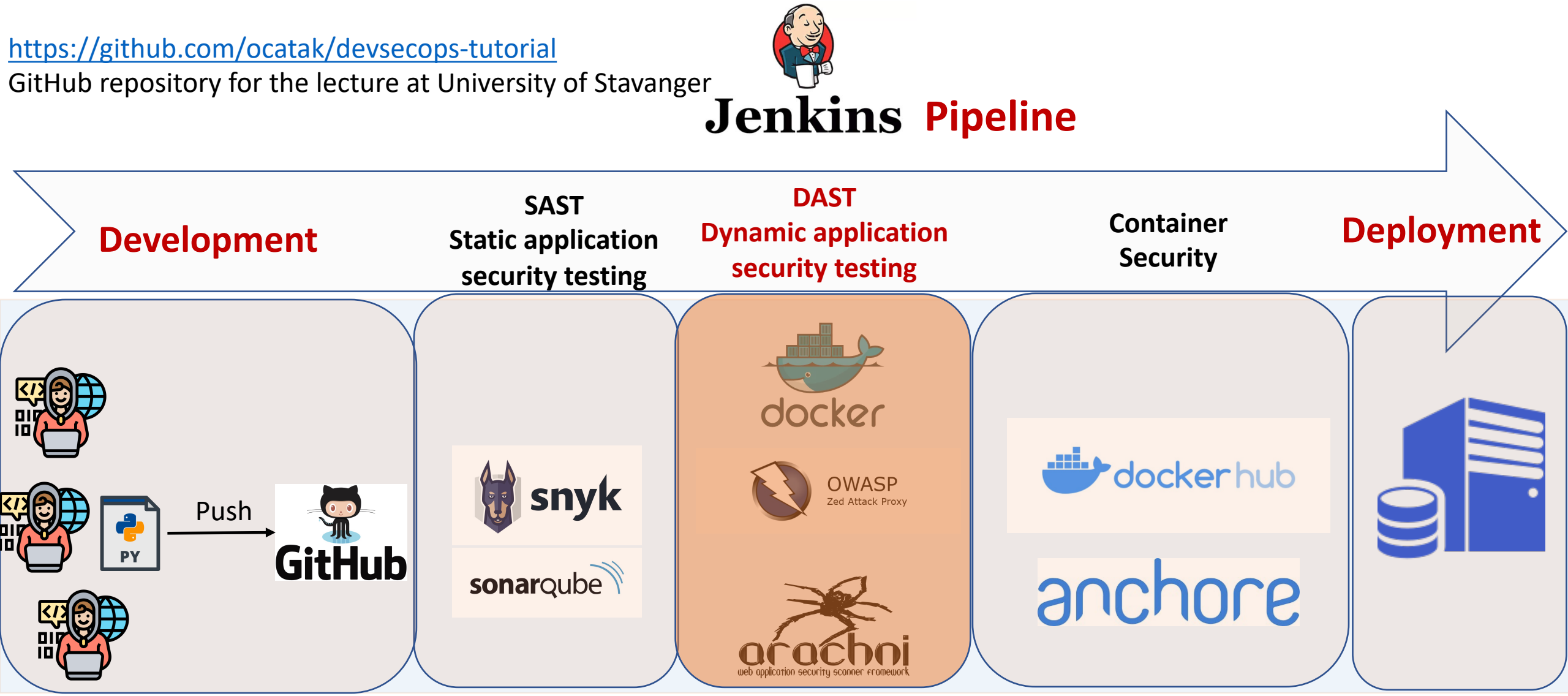






Example Web Application: DevSecOps

https://github.com/ocatak/devsecops-tutorial
GitHub repository for the lecture at University of Stavanger

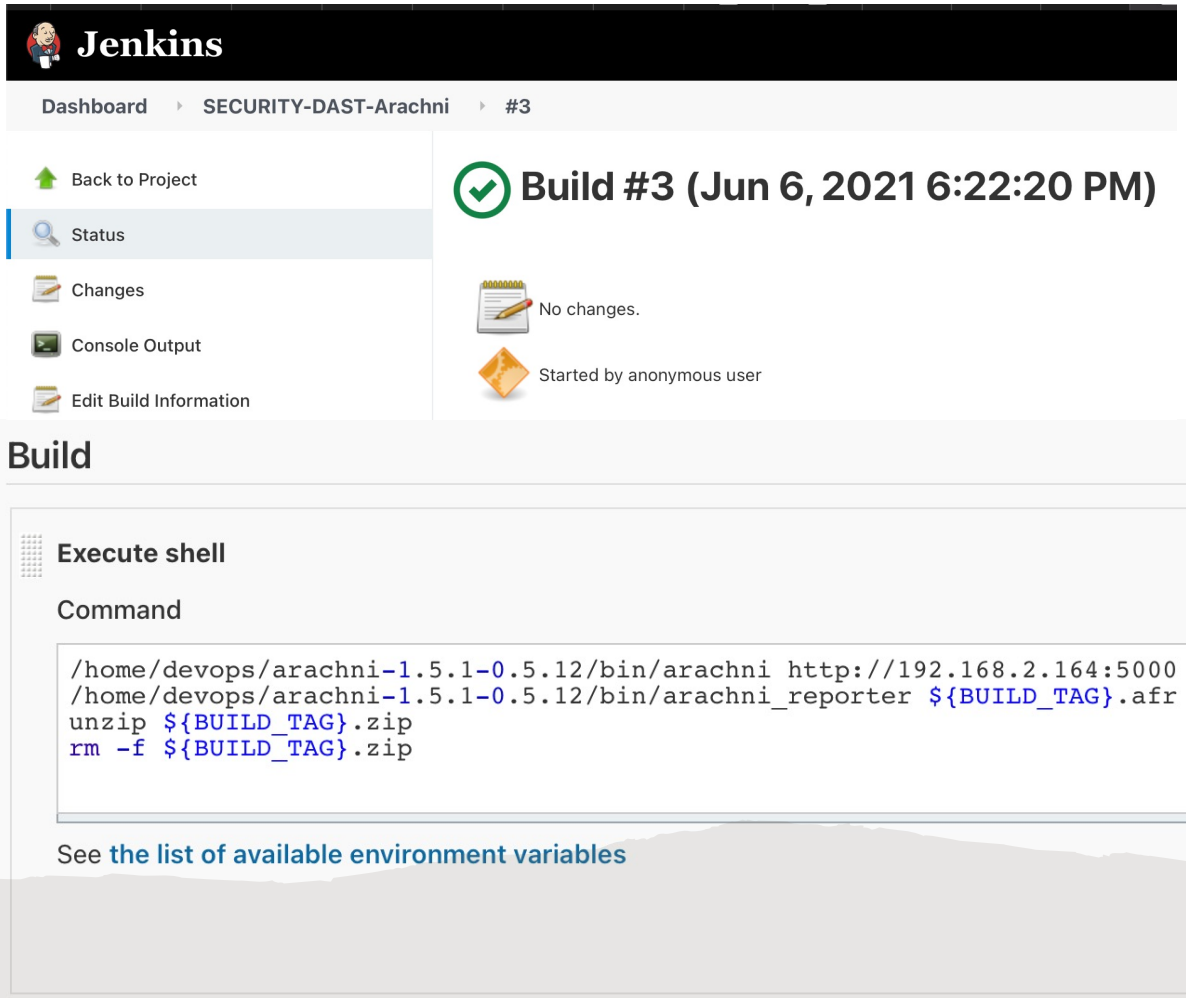


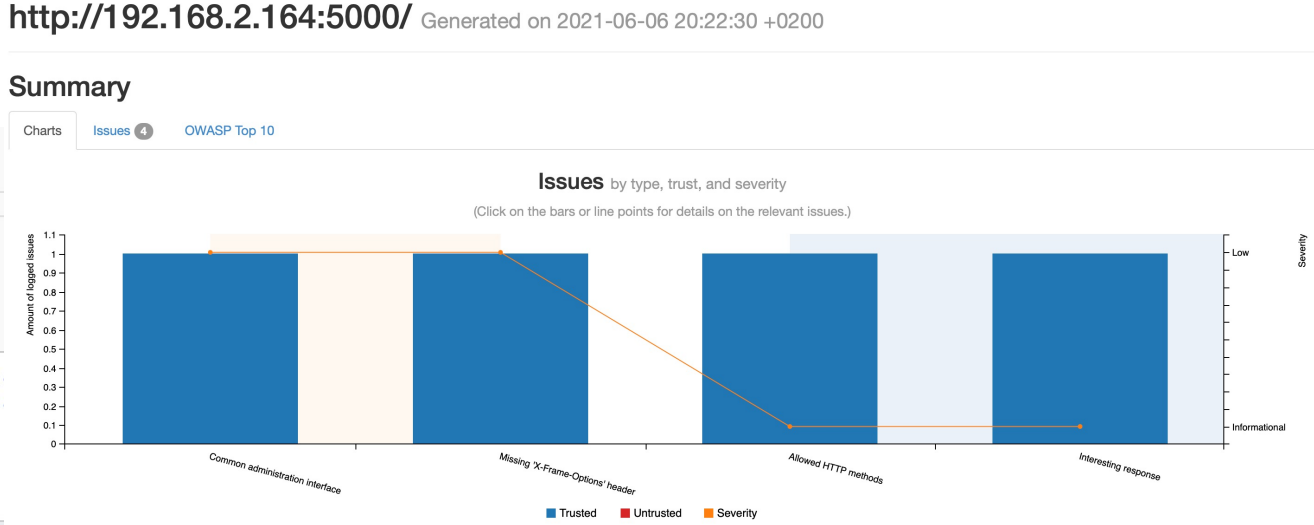
DevSecOps: DAST - ARACHNI

- Arachni is a feature-full, modular, high performance Ruby framework aimed towards helping PenTesters and administrators evaluate the security of modern web applications.
- It is free, with its source code public and available for review.
- https://github.com/Arachni/arachni

DevSecOps: DAST - ARACHNI

https://github.com/Arachni/arachni/releases/download/v1.5.1/arachni-1.5.1-0.5.12-linux-x86_64.tar.gz





DevSecOps: DAST - ARACHNI

Summary

Charts | Issues 4 | OWASP Top 10

Trusted 4

Low severity 2

Common administration interface 1

An administration interface was identified and should be reviewed.

	Vector type	HTTP method	Action
	`server`	`GET`	http://192.168.2.164:5000/console

Missing 'X-Frame-Options' header 1

Clickjacking (User Interface redress attack, UI redress attack, UI redressing) is a malicious technique of tricking a Web user into clicking on something different from what the user perceives they are clicking on, thus potentially revealing confidential information or taking control of their computer while clicking on seemingly innocuous web pages.

The server didnâ€™t return an `X-Frame-Options` header which means that this website could be at risk of a clickjacking attack.

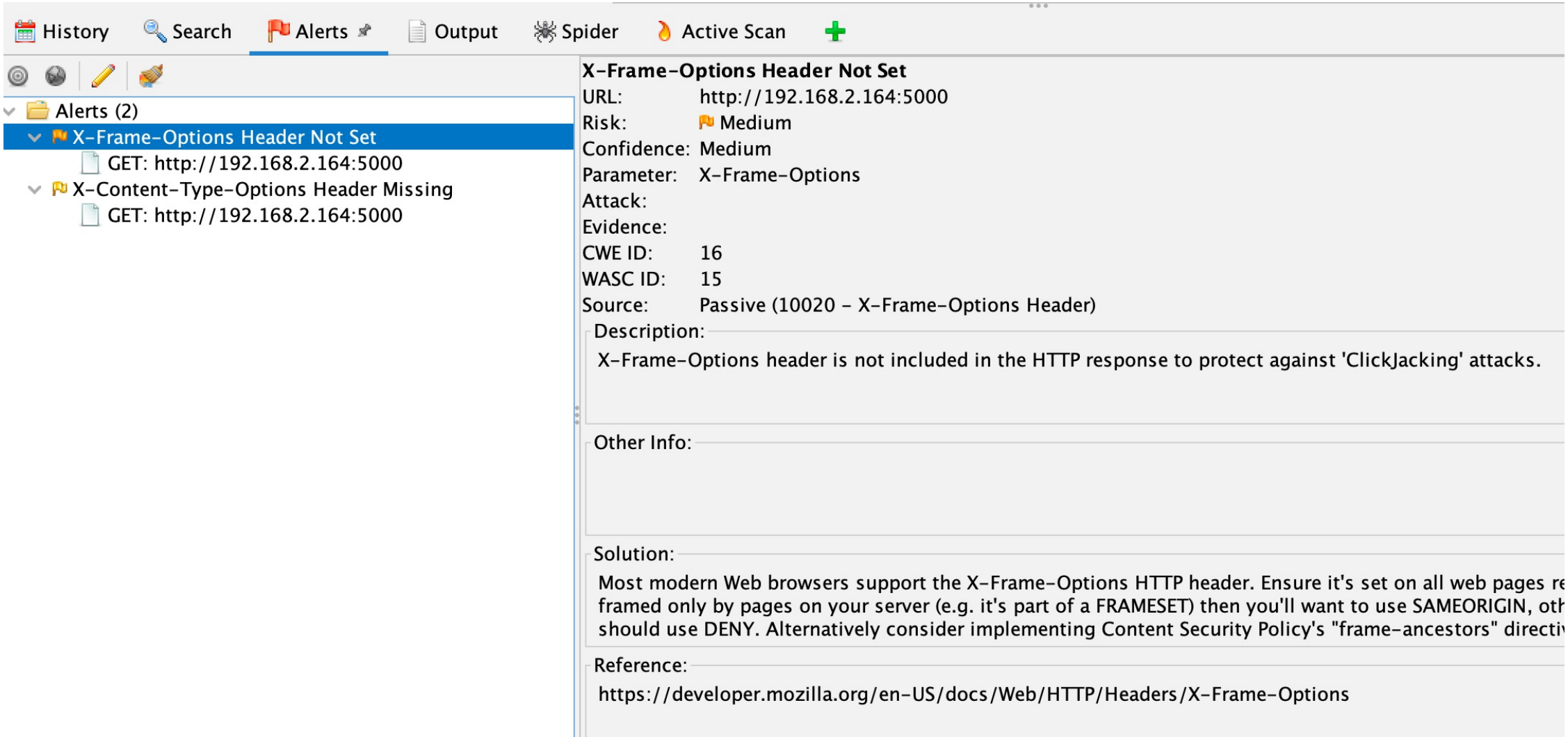
The `X-Frame-Options` HTTP response header can be used to indicate whether or not a browser should be allowed to render a page inside a frame or iframe. Sites can use this to avoid clickjacking attacks, by ensuring that their content is not embedded into other sites.

	Vector type	HTTP method	Action
	`server`	`GET`	http://192.168.2.164:5000/

DevSecOps: DAST – OWASP ZAP

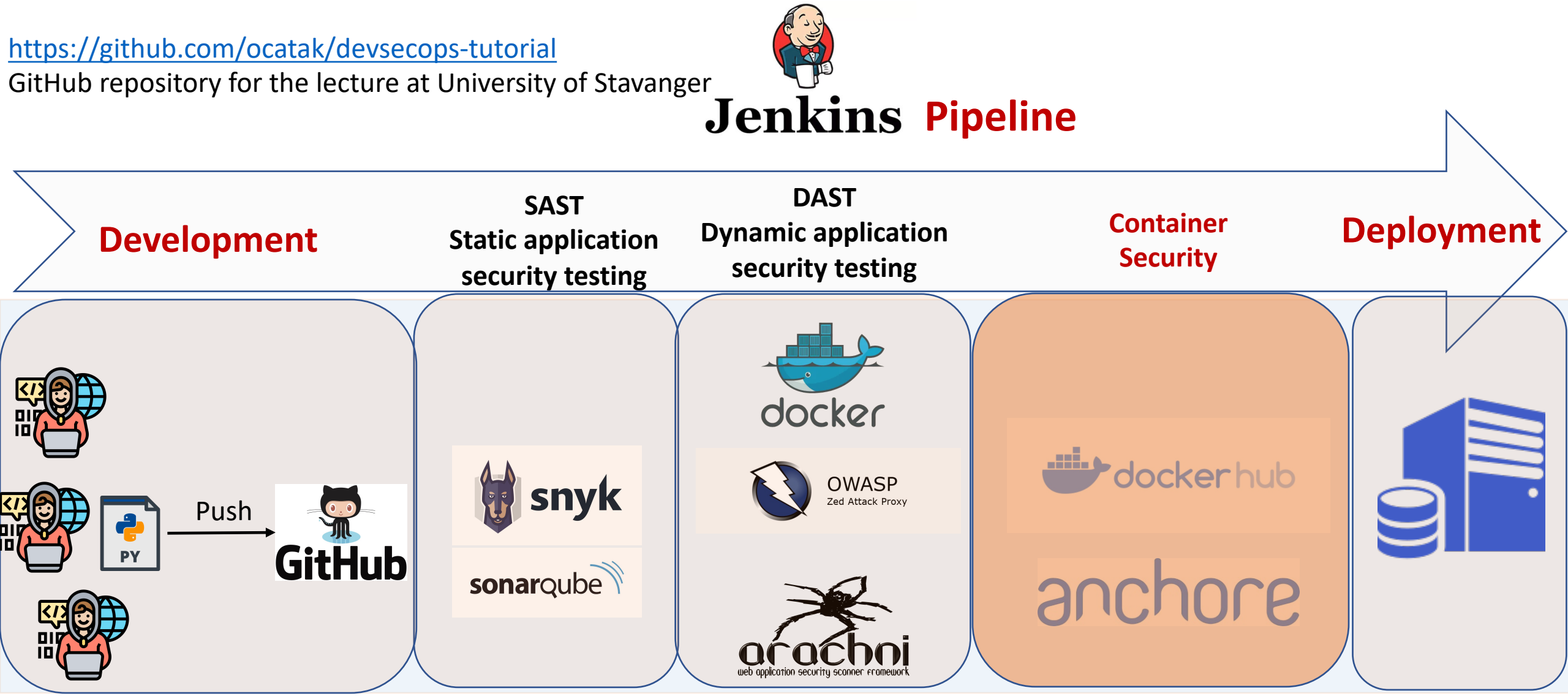
- Owasp-zap is one of the world's most popular free security tool and is actively maintained by hundreds of international volunteers.
- It helps to find vulnerabilities automatically with every ci/cd build.
- One of the best free tool for experienced PenTesters to be used for manual security testing.

DevSecOps: DAST – OWASP ZAP



Example Web Application: DevSecOps

https://github.com/ocatak/devsecops-tutorial
GitHub repository for the lecture at University of Stavanger



DevSecOps: Container Security - Anchore

- Anchore allows us to perform detailed analysis on container images, run queries, produce reports. Also we can define our own policies that can be used with CI/CD pipelines

DevSecOps: Container Security - Anchore

Execute shell

Command

```
echo "fcatak/devsecops_stavanger:new" ${WORKSPACE}/Dockerfile > anchore_images
cat anchore_images
```

See the list of available environment variables

Anchore Container Image Scanner

Anchore Build Options

Image list file

anchore_images

- [x] Fail build on policy evaluation FAIL result
- [x] Fail build on critical plugin error

Anchore Policy Evaluation Summary

Show 10 entries Search:

Repo Tag	Stop Actions	Warn Actions	Go Actions	Final Ac
docker.io/fcatak/devsecops_stavanger:new	27	77	0	STOP

Showing 1 to 1 of 1 entries Previous

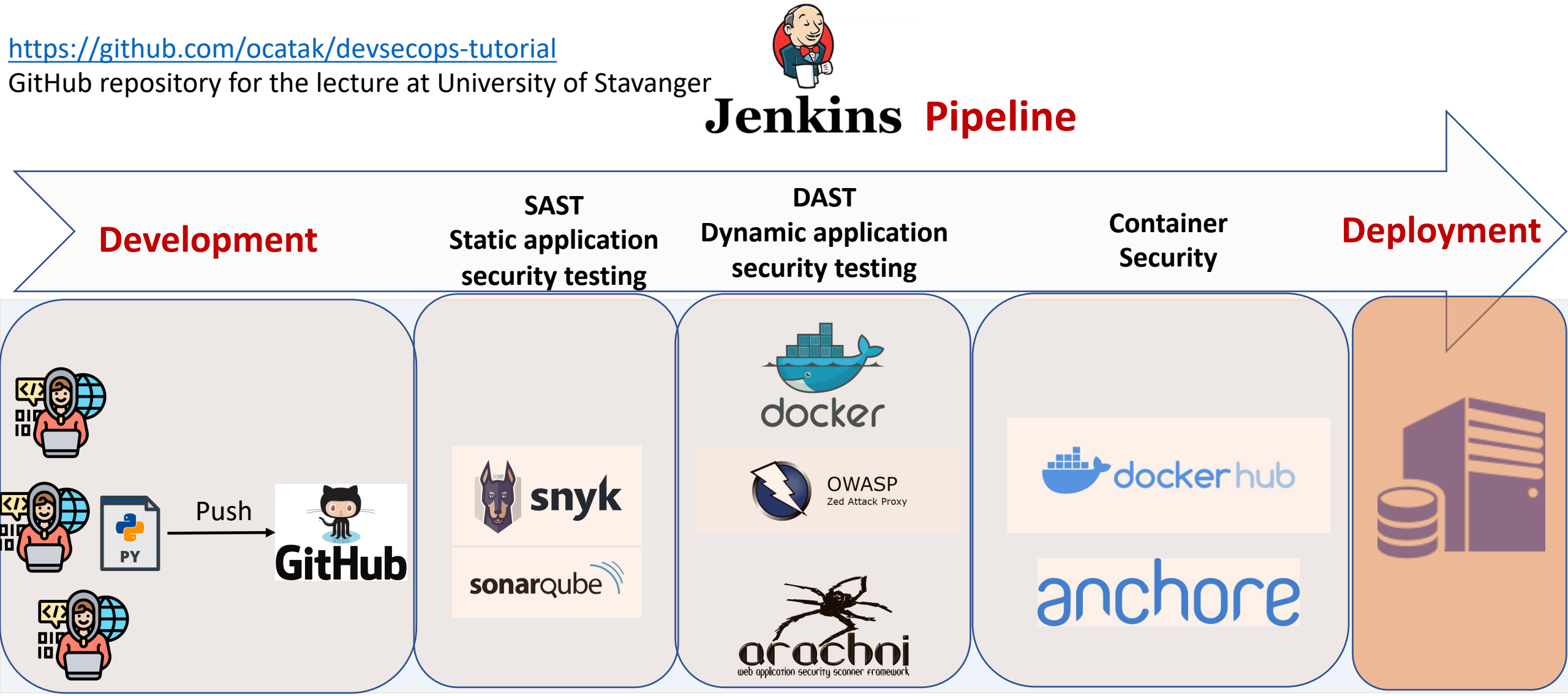
Anchore Policy Evaluation Report

Show 10 entries Search:

Image Id	Repo Tag	Trigger Id	Gate	Trigger	Check Output	Gate Action	Whitelisted
0d0b5c8bc74cc6156bb48fb1d8c6887ad8f1f34a3b44a1a64682dd819b19e32d	docker.io/fcatak/devsecops_stavanger:new	CVE-2019-25032+libunbound8	vulnerabilities	package	HIGH Vulnerability found in os package type (dpkg) - libunbound8 (CVE-2019-25032 - https://security-tracker.debian.org/tracker/CVE-2019-25032)	STOP	false
0d0b5c8bc74cc6156bb48fb1d8c6887ad8f1f34a3b44a1a64682dd819	docker.io/fcatak/devsecops_stavanger:new	CVE-2019-25033+libunbound8	vulnerabilities	package	HIGH Vulnerability found in os package type (dpkg) - libunbound8 (CVE-2019-25033 - https://security-tracker.debian.org/tracker/CVE-2019-	STOP	false

Example Web Application: DevSecOps

https://github.com/ocatak/devsecops-tutorial
GitHub repository for the lecture at University of Stavanger



What We Learned?

- Paradigm-shifting from time consuming Penetration testing Activities to DevSecOps
- PenTesters use same/similar tools (Kali, Sqlmap, Arachni, Nmap, Metasplot etc) to perform a security testing to find the vulnerabilities
- We can integrate them into our CI/CD Pipelines
- Speed-Up security checks

Future Challenges

- Complex Tool Integrations
  - come from various vendors.
  - Adding security tools makes things even more complex.
  - time-consuming, labor-intensive approach is incompatible with DevOps' high-speed,
- Shifting security left in the SDLC
  - Security principles apply to overall organization and not just to security engineering teams.
  - End-of-cycle security testing results in delays and vulnerabilities left in production

Search or jump to... / Pull requests Issues Marketplace Explore

ocatak / devsecops-tutorial

Unwatch 1 Unsta

Code Issues Pull requests Actions Projects Wiki Security 2 Insights Settings

We found potential security vulnerabilities in your dependencies.

Only the owner of this repository can see this message.

See Dependabot alerts

master 2 branches 0 tags Go to file Add file Code

ocatak SonarQube 512af74 5 days ago 23 commits

anchore-security	anchore-security	7 days ago
docker-jenkins	Jenkins	7 days ago
templates	README.md	7 days ago
Dockerfile	docker	7 days ago
Jenkinsfile	SNYK	5 days ago
README.md	README	6 days ago
arachni.sh	docker	7 days ago
docker-compose.yml	docker	7 days ago

Thank You

F. Ozgur Catak

f.ozgur.catak@gmail.com

https://www.ozgurcatak.org

About

DevSecOps best practices vulnerable Flask based we application

python docker jenkins devops course arachni devsecops

Readme

Releases

No releases published

Create a new release

Packages

No packages published

Publish your first package