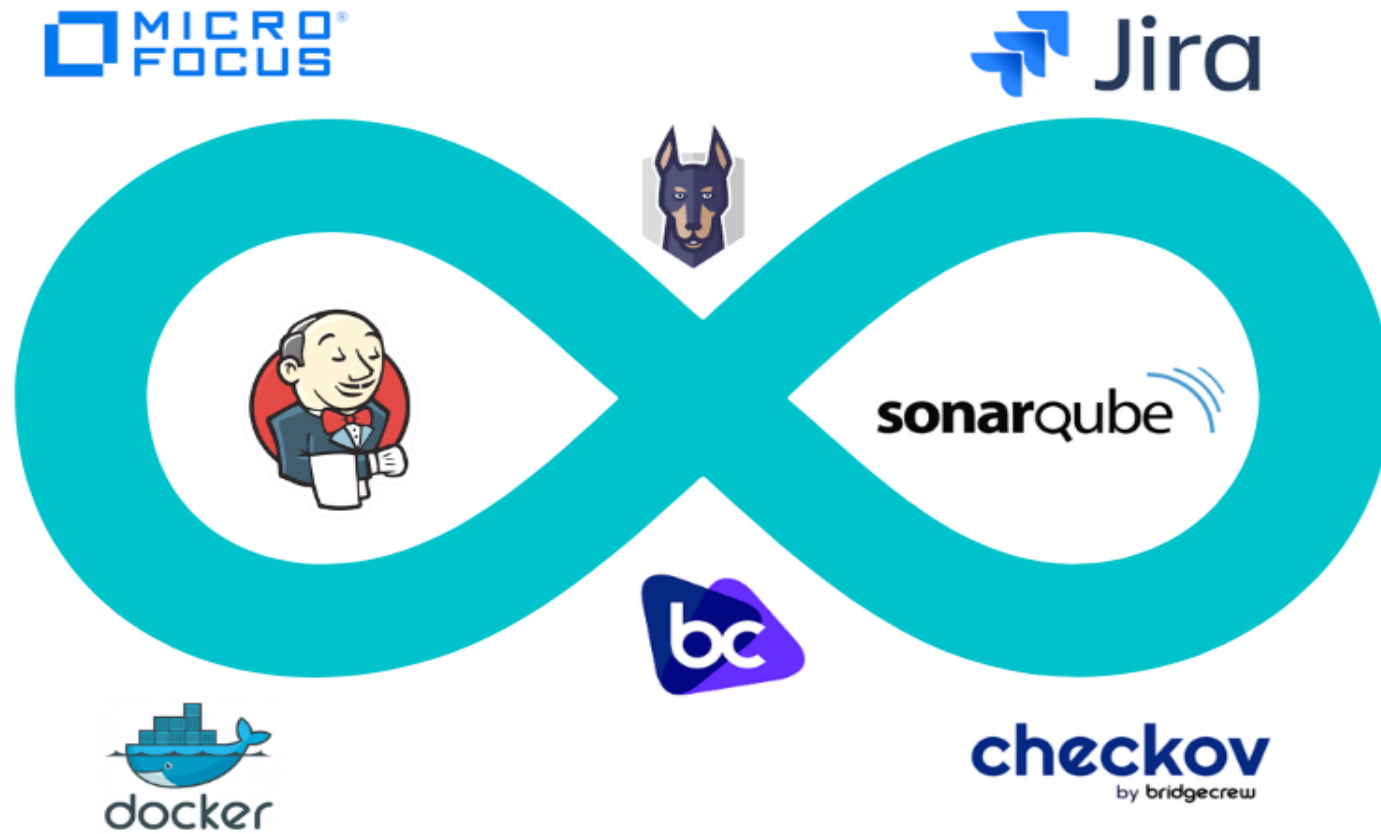
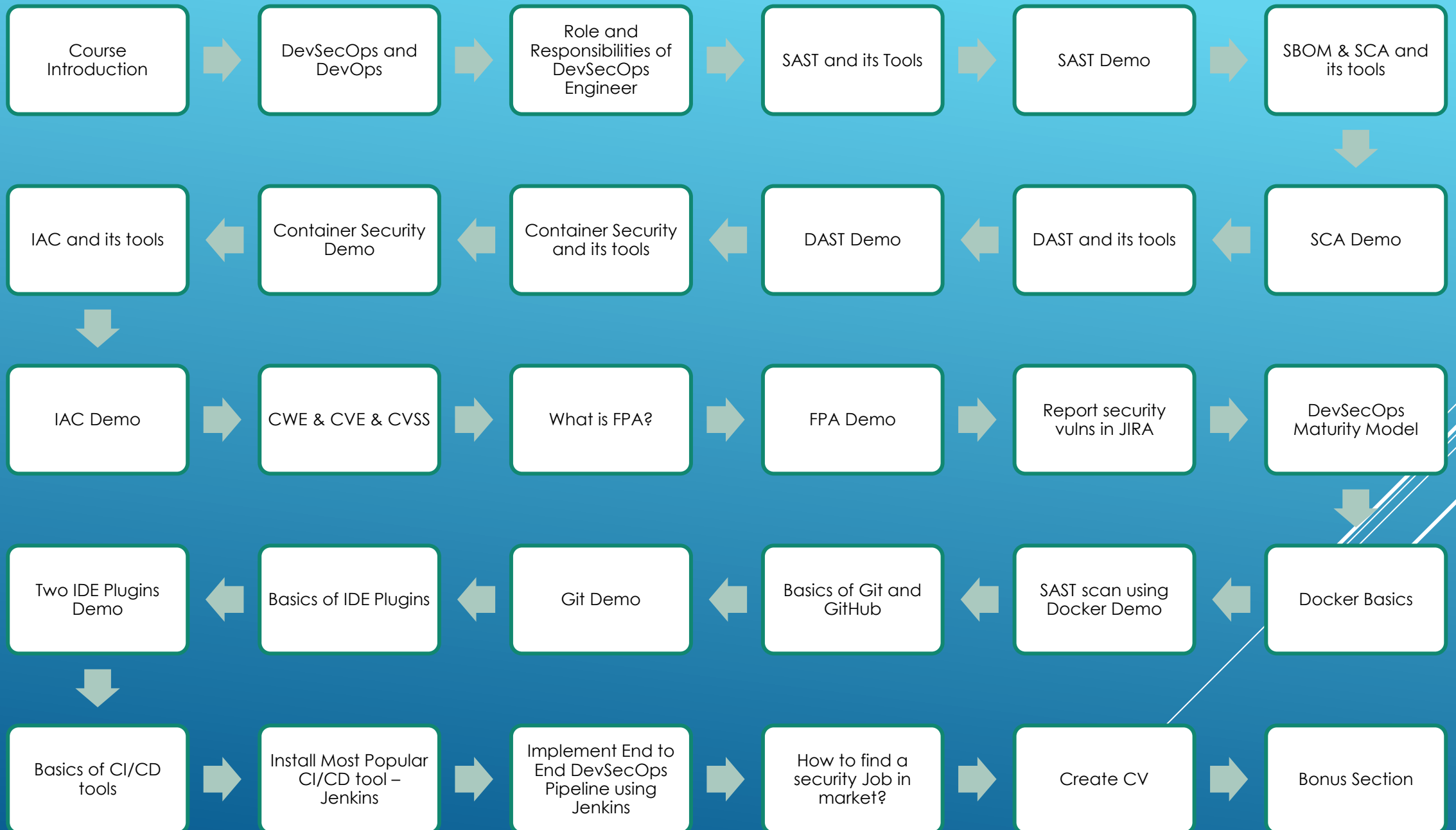


DEVSECOPS FOR THE ABSOLUTE BEGINNERS

BEGINNER TO ADVANCED



AGENDA



ABOUT THIS COURSE

This course is designed to introduce freshers, DevOps Engineers, QA professionals to the field of Application Security. As Application Security is a niche domain, there is a lot of demand in this field and very high paying jobs remain vacant because of unavailability of right talent.

INTRODUCTION TO DEVSECOPS

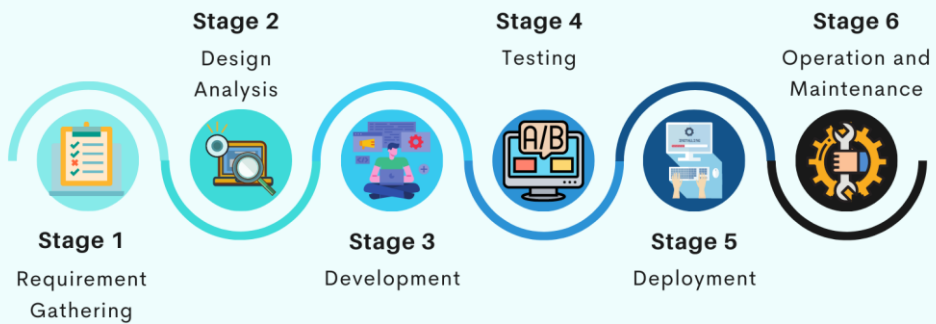
- ▶ What is DevSecOps?
- ▶ How its different from DevOps?

WHAT IS DEVSECOPS AND HOW ITS DIFFERENT FROM DEVOPS?

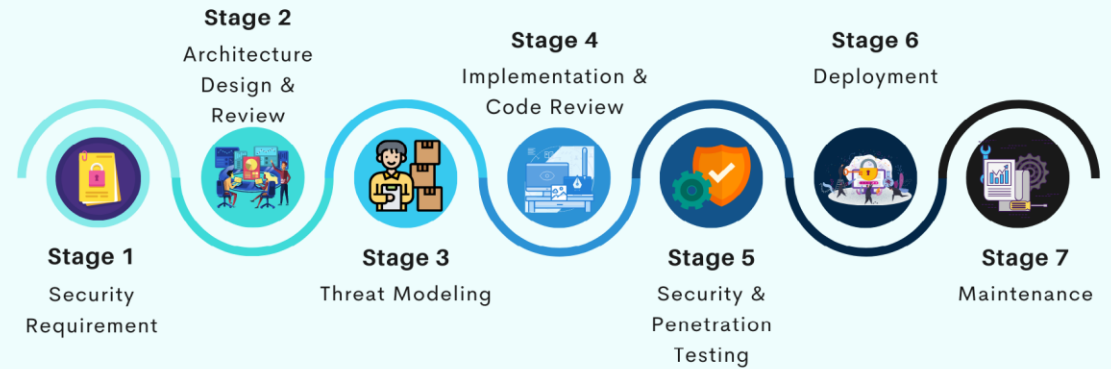
- ▶ **DevSecOps** is an approach to software development that focuses on the integration of security practices in the DevOps process. **DevOps** is a set of practices that aims to improve collaboration and communication between development and operations teams, with the goal of delivering software updates and applications more quickly and efficiently.
- ▶ While DevOps focuses on improving the speed and efficiency of software development and deployment, DevSecOps adds a focus on security. It involves incorporating security practices and tools into the software development process from the beginning, rather than treating security as an afterthought. This can include activities such as performing security testing, scanning for vulnerabilities, and implementing security controls as part of the continuous integration and delivery process.
- ▶ The goal of DevSecOps is to ensure that security is considered at every stage of the software development lifecycle, rather than being addressed as a separate concern. By integrating security into the DevOps process, organizations can improve the security of their software and reduce the risk of security breaches. DevSecOps helps organizations to more effectively balance the competing goals of fast delivery and strong security.



TRADITIONAL SDLC STAGES



SECURE SDLC STAGES



Focus on Security

INTRODUCTION TO DEVSECOPS

- ▶ Roles and Responsibilities of DevSecOps Engineer

ROLES AND RESPONSIBILITIES OF DEVSECOPS ENGINEER

- ▶ To Perform Static Application Security Testing (SAST) using IDE plugins e.g. *SonarLint*
- ▶ To implement credential scanning tools like *Git Guardian* in CI/CD pipeline or at commit level e.g. *GIT Secrets*
- ▶ To integrate SAST tools like *SonarQube*, *SonarCloud*, *Fortify Source Code Analysis*, *Veracode*, *Checkmarx* tools in CI/CD Pipeline
- ▶ To integrate SCA tools like *OWASP Dependency Check*, *Snyk*, *Blackduck* in CI/CD Pipeline
- ▶ To integrate DAST tools like *OWASP ZAP*, *WebInspect*, *Veracode DAST*, *Burp* in CI/CD Pipeline
- ▶ To integrate Container Scanning tools like *Trivy*, *AQUA* in CI/CD Pipeline
- ▶ To integrate IAC scanning tools like *Snyk*, *Bridgecrew* in CI/CD Pipeline
- ▶ To perform *SAST*, *SCA and DAST* security scanning and perform False positive Analysis on security vulnerabilities
- ▶ To report security vulnerabilities in ticketing tool like JIRA and assign tickets to Development team to fix these security vulnerabilities
- ▶ To write script in YAML and other scripting languages for integrating security tools in CI/CD Pipelines. This is also called **Shift Left Approach**
- ▶ To improve DevSecOps Maturity levels of overall Development processes
- ▶ To implement Cloud Security controls and recommendations for compliance

INTRODUCTION TO DEVSECOPS

- ▶ What is Static Application Security Testing (SAST)?
- ▶ What are the commercial and open source tools used for SAST?

SAST AND ITS TOOLS

- ▶ **SAST or Static Application Security Testing** is a type of security testing that involves analyzing the source code of a software application to identify security vulnerabilities. SAST is typically performed during the development phase of an application, before it is deployed.
- ▶ SAST tools are designed to analyze source code for known vulnerabilities and security weaknesses. They can also identify areas of the code that may be prone to security issues, such as code that is difficult to maintain or understand.
- ▶ SAST tools typically operate by scanning the source code of an application and looking for patterns or specific vulnerabilities. The results of a SAST scan can be used to identify and fix vulnerabilities before an application is deployed, which can help to improve the overall security of the application.
- ▶ Commercial SAST tools are *Checkmarx, Microfocus Fortify Source Code Analysis, Microfocus Fortify On Demand, Veracode, SonarQube, SonarCloud*. All these tools can be integrated in CI/CD platforms.
- ▶ Some SAST Commercial IDE plugins are provided by *Checkmarx, Veracode, Microfocus organizations*
- ▶ Free or Open source SAST tools are *SonarQube, SonarCloud, Snyc*
- ▶ Some SAST Free or OpenSource IDE plugins – *SonarLint, Snyc*

SAST SCAN OUTPUT SAMPLE

SAST DEMO WITH FORTIFY ON DEMAND

INTRODUCTION TO DEVSECOPS

- ▶ What is Software Bill of Materials (SBOM) and how its different from Software Composition Analysis (SCA) scan?
- ▶ What are the commercial and open source tools used for SCA?

SCA AND ITS TOOLS

- ▶ **SBOM stands for "Software Bill of Materials"** is a list of all the software components and their versions that are included in a particular software application or system. The SBOM includes both the first-party components that were developed by the software vendor and the third-party components that were used to build the application. The purpose of an SBOM is to provide a complete and accurate inventory of all the components that make up a software application or system. SBOMs are an important part of software composition analysis (SCA) and are increasingly being used in the software development and maintenance process to ensure the security and compliance of applications.
- ▶ **Software Composition Analysis(SCA)** scan is a process of identifying and analyzing the third-party libraries, frameworks, and other components that are used in a software application. The goal of SCA is to identify potential security vulnerabilities, licensing issues, and other risks associated with the use of third-party components in an application.
- ▶ It involves analyzing the source code of an application to identify the third-party components that it uses, and then evaluating these components for security vulnerabilities, licensing issues, and other risks. The results of the SCA process can be used to identify and fix potential problems before the application is released, or to assess the security and compliance of an existing application.
- ▶ SCA is an important aspect of software development and maintenance, as it helps to ensure that applications are secure and compliant with industry standards and regulations. It is especially important in large organizations where multiple developers may be working on different parts of the same application, or where applications are built using a variety of third-party components.

SCA AND ITS TOOLS

- ▶ Commercial SCA tools are *Snyk*, *Veracode SCA*, *BlackDuck*. All these tools can be integrated in CI/CD platforms.
- ▶ Free or Open source SCA tools are *OWASP Dependency Check*

SCA SCAN OUTPUT SAMPLE

SEVERITY

☐ Critical

1

☐ High

13

☐ Medium

9

☐ Low

4

PRIORITY SCORE

Scored between 0 - 1000

FIXABILITY

☐ Fixable

8

☐ Partially fixable

0

☐ No fix available

19

EXPLOIT MATURITY

☐ Mature

0

☐ Proof of concept

12

☐ No known exploit

15

☐ No data

0

STATUS

☒ Open

27

Search...

27 of 27 issues

Sort by highest priority score

H

qs - Prototype Poisoning

VULNERABILITY

CWE-1321

CVE-2022-24999

CVSS 7.5

HIGH

SNYK-JS-QS-3153490

SCORE

696

Introduced through

express@4.16.3

Exploit maturity

PROOF OF CONCEPT

Fixed in

qs@6.2.4, @6.3.3, @6.4.1, @6.5.3, @6.6.1, @6.7.3, @6.8.3, @6.9.7, @6.10.3

Show more detail

NEW

Learn about this type of vulnerability

Ignore

M

morgan - Arbitrary Code Injection

VULNERABILITY

CWE-94

CVE-2019-5413

CVSS 6.8

MEDIUM

SNYK-JS-MORGAN-72579

SCORE

661

Introduced through

morgan@1.9.0

Exploit maturity

PROOF OF CONCEPT

Fixed in

morgan@1.9.1

Show more detail

NEW

Learn about this type of vulnerability

Ignore

SCA DEMO WITH SNYK

INTRODUCTION TO DEVSECOPS

- ▶ What is Dynamic Application Security Testing (DAST) scan?
- ▶ What are the commercial and open source tools used for DAST?

DAST AND ITS TOOLS

- ▶ **DAST stands for "Dynamic Application Security testing"** and is a method of testing the security of an application or system by actively interacting with it and attempting to exploit vulnerabilities.
- ▶ DAST is typically performed after an application has been developed and deployed, and is designed to identify vulnerabilities that may not have been detected during the design or development phases. It involves sending various inputs to the application or system and observing the responses, looking for indications of vulnerabilities such as unhandled exceptions, error messages, or other unusual behavior.
- ▶ DAST can be an effective way to identify and address vulnerabilities in an application or system, but it is important to note that it is not a substitute for other types of security testing such as static analysis or penetration testing. It is often used in combination with these other techniques as part of a comprehensive security testing strategy.
- ▶ Commercial DAST tools are *WebInspect*, *Veracode DAST*, *Burp Professional*. These tools can be integrated in CI/CD platforms.
- ▶ Free or Open source DAST tools are *OWASP ZAP*, *Burp Community Edition*.

DAST SCAN OUTPUT SAMPLE


ZAP Scanning Report - Mozilla Firefox

FileEditViewHistoryBookmarksToolsHelp

ZAP Scanning Report

file:///root/Documents/intro-to-owasp-zap.html90%Search

Offensive SecurityKali LinuxKali DocsKali ToolsExploit-DBAircrack-ngKali ForumsNetHunterMost VisitedOffensive SecurityKali Linux

ZAP Scanning Report

Summary of Alerts

Risk Level	Number of Alerts
High	2
Medium	2
Low	5
Informational	0

Alert Detail

High (Medium)	Cross Site Scripting (Reflected)
Description	<p>Cross-site Scripting (XSS) is an attack technique that involves echoing attacker-supplied code into a user's browser instance. A browser instance can be a standard web browser client, or a browser object embedded in a software product such as the browser within WinAmp, an RSS reader, or an email client. The code itself is usually written in HTML/JavaScript, but may also extend to VBScript, ActiveX, Java, Flash, or any other browser-supported technology.</p> <p>When an attacker gets a user's browser to execute his/her code, the code will run within the security context (or zone) of the hosting web site. With this level of privilege, the code has the ability to read, modify and transmit any sensitive data accessible by the browser. A Cross-site Scripted user could have his/her account hijacked (cookie theft), their browser redirected to another location, or possibly shown fraudulent content delivered by the web site they are visiting. Cross-site Scripting attacks essentially compromise the trust relationship between a user and the web site. Applications utilizing browser object instances which load content from the file system may execute code under the local machine zone allowing for system compromise.</p> <p>There are three types of Cross-site Scripting attacks: non-persistent, persistent and DOM-based.</p> <p>Non-persistent attacks and DOM-based attacks require a user to either visit a specially crafted link laced with malicious code, or visit a malicious web page containing a web form, which when posted to the vulnerable site, will mount the attack. Using a malicious form will oftentimes take place when the vulnerable resource only accepts HTTP POST requests. In such a case, the form can be submitted automatically, without the victim's knowledge (e.g. by using JavaScript). Upon clicking on the malicious link or submitting the malicious form, the XSS payload will get echoed back and will get interpreted by the user's browser and execute. Another technique to send almost arbitrary requests (GET and POST) is by using an embedded client, such as Adobe Flash.</p> <p>Persistent attacks occur when the malicious code is submitted to a web site where it's stored for a period of time. Examples of an attacker's favorite targets often include message board posts, web mail messages, and web chat software. The unsuspecting user is not required to interact with any additional site/link (e.g. an attacker site or a malicious link sent via email), just simply view the web page containing the code.</p>
URL	http://webscantest.com/info/db/search_by_name.php
Method	POST
Parameter	fname
Attack	</table><script>alert(1);</script><table>
Evidence	</table><script>alert(1);</script><table>

DAST DEMO WITH OWASP ZAP HOSTED SCAN

INTRODUCTION TO DEVSECOPS

- ▶ What is Container Security scan?
- ▶ What are the commercial and open source tools used for Container Security scan?

CONTAINER SECURITY AND ITS TOOLS

- ▶ **Containers** are a way of packaging and deploying applications in a lightweight, portable format. They allow developers to package an application and its dependencies together in a single package, making it easy to deploy and run the application on any machine that is compatible with the container runtime.
- ▶ Like any other type of software, containers can be vulnerable to security issues if they are not properly configured or managed. To ensure the security of container-based applications, container security scans are performed.
- ▶ Commercial Container Security tools are *AQUA*, *Prisma Cloud*, *Snyk*. These tools can be integrated in CI/CD platforms.
- ▶ Free or Open source DAST tools are *Trivy*, *Snyk Community Edition*.

CONTAINER SECURITY SCAN OUTPUT SAMPLE

```
root@controlplane:~# trivy image nginx:alpine
2021-10-21T16:35:58.005Z      INFO    Detected OS: alpine
2021-10-21T16:35:58.005Z      INFO    Detecting Alpine vulnerabilities...
2021-10-21T16:35:58.009Z      INFO    Number of language-specific files: 0
```

```
nginx:alpine (alpine 3.14.2)
```

```
=====  
Total: 6 (UNKNOWN: 0, LOW: 0, MEDIUM: 2, HIGH: 2, CRITICAL: 2)
```

LIBRARY	VULNERABILITY ID	SEVERITY	INSTALLED VERSION	FIXED VERSION	TITLE
curl	CVE-2021-22945	CRITICAL	7.78.0-r0	7.79.0-r0	curl: use-after-free and double-free in MQTT sending -->avd.aquasec.com/nvd/cve-2021-22945
	CVE-2021-22946	HIGH			curl: Requirement to use TLS not properly enforced for IMAP, POP3, and... -->avd.aquasec.com/nvd/cve-2021-22946
	CVE-2021-22947	MEDIUM			curl: Server responses received before STARTTLS processed after TLS handshake -->avd.aquasec.com/nvd/cve-2021-22947
libcurl	CVE-2021-22945	CRITICAL			curl: use-after-free and double-free in MQTT sending -->avd.aquasec.com/nvd/cve-2021-22945
	CVE-2021-22946	HIGH			curl: Requirement to use TLS not properly enforced for IMAP, POP3, and... -->avd.aquasec.com/nvd/cve-2021-22946
	CVE-2021-22947	MEDIUM			curl: Server responses received before STARTTLS processed after TLS handshake -->avd.aquasec.com/nvd/cve-2021-22947

```
root@controlplane:~# █
```



CONTAINER SECURITY DEMO WITH DOCKER SCAN



INTRODUCTION TO DEVSECOPS

- ▶ What is IAC (Infrastructure As Code) scan?
- ▶ What are the commercial and open source tools used for IAC scan?

IAC SECURITY AND ITS TOOLS

- ▶ **Infrastructure as Code (IAC)** refers to the practice of using code to define and manage infrastructure resources such as servers, networking components, and other IT resources. IAC allows organizations to automate the provisioning, configuration, and management of their infrastructure, which can improve efficiency and reduce the risk of errors.
- ▶ An IAC security scan is a process of analyzing IAC code and configurations to identify potential security vulnerabilities or misconfigurations. This can be done manually or using automated tools that are designed to scan IAC code for known vulnerabilities or patterns that may indicate potential security issues.
- ▶ Commercial IAC Security tools are *Checkov, Snyk, CloudSploit*. These tools can be integrated in CI/CD platforms.
- ▶ Free or Open source DAST tools are *Snyk Community Edition*.

IAC SECURITY SCAN OUTPUT SAMPLE

The screenshot displays the GitHub interface for the repository `bridgecrewio/terragoat`. The `Security` tab is active, showing 16 open alerts. The left sidebar contains navigation links: Overview, Security policy, Security advisories (0), Code scanning alerts (16), and bridgecrew. The main content area is titled `Code scanning` and includes a filter bar with the text `tool:bridgecrew is:open`. Below the filter, a table lists 16 alerts, all of which are open and detected 8 days ago. The alerts are categorized by branch (master) and severity (Warning). The alerts are as follows:

Alert	Branch	Severity	Detected
Ensure all data stored in the S3 bucket have versioning enabled	master	Warning	8 days ago
Ensure S3 bucket has MFA delete enabled	master	Warning	8 days ago
S3 Bucket has an ACL defined which allows public READ access.	master	Warning	8 days ago
Ensure all data stored in the S3 bucket is securely encrypted at rest	master	Warning	8 days ago
Ensure the S3 bucket has access logging enabled	master	Warning	8 days ago
Ensure all data stored in the S3 bucket have versioning enabled	master	Warning	8 days ago
Ensure S3 bucket has MFA delete enabled	master	Warning	8 days ago
Ensure all data stored in the S3 bucket is securely encrypted at rest	master	Warning	8 days ago
Ensure the S3 bucket has access logging enabled	master	Warning	8 days ago

IAC SECURITY DEMO WITH BRIDGECREW




INTRODUCTION TO DEVSECOPS

- ▶ What is CVE and CVS and how its related to DevSecOps?

CVE AND CVSS

- ▶ **CVE (Common Vulnerabilities and Exposures)** is a database of publicly disclosed cybersecurity vulnerabilities. It is maintained by the MITRE Corporation, a nonprofit organization that provides technical expertise to the U.S. government. Each CVE entry includes a unique identifier, a description of the vulnerability, and information about the affected software or hardware.
- ▶ **CVSS (Common Vulnerability Scoring System)** is a standardized method for evaluating the severity of vulnerabilities. It is designed to provide a consistent way of measuring the risk posed by a particular vulnerability, regardless of the specific software or hardware affected.
- ▶ CVSS scores are based on a number of factors including the potential impact of the vulnerability, the likelihood that it will be exploited, and the ease with which it can be exploited. Scores range from 0 to 10, with higher scores indicating greater severity.
- ▶ CVSS scores are often used to prioritize the response to vulnerabilities, with the most severe vulnerabilities being addressed first. They are also used by security analysts and others to compare the severity of different vulnerabilities and to understand the relative risk posed by different vulnerabilities.
- ▶ While performing False Positive Analysis, the severity of the incident needs be considered after reviewing these scores by the DevSecOps Engineer

CVE AND CVSS SAMPLE



[CVE List](#) [CNAS](#) [WGS](#) [Board](#) [About](#) [News & Blog](#)

[Search CVE List](#) [Downloads](#) [Data Feeds](#) [Update a CVE Record](#) [Request CVE IDs](#)

TOTAL CVE Records: **191743**

NOTICE: Transition to the all-new CVE website at WWW.CVE.ORG is underway and will last up to one year. ([detail](#))

NOTICE: Changes coming to [CVE Record Format JSON](#) and [CVE List Content Downloads](#) in 2022.

HOME > CVE > CVE-2021-44228

CVE-ID

CVE-2021-44228 [Learn more at National Vulnerability Database \(NVD\)](#)
• CVSS Severity Rating • Fix Information • Vulnerable Software Versions • SCAP Mappings • CPE Information

Description

Apache Log4j2 2.0-beta9 through 2.15.0 (excluding security releases 2.12.2, 2.12.3, and 2.3.1) JNDI features used in configuration, log messages, and parameters do not protect against an attacker who can control log messages or log message parameters can execute arbitrary code loaded from LDAP servers when message lookup substitution is enabled. From log4j 2.15.0, this behavior has been disabled by default. From version 2.16.0 (along with 2.12.2, 2.12.3, and 2.3.1), this functionality has been completely removed. Note that this vulnerability is specific to log4j-core and does not affect log4net, log4cxx, or other Apache Logging Services projects.

References

Note: [References](#) are provided for the convenience of the reader to help distinguish between vulnerabilities. The list is not intended to be complete.

CVE-2021-44228 Detail

UNDERGOING REANALYSIS

This vulnerability has been modified and is currently undergoing reanalysis. Please check back soon to view the updated vulnerability summary.

Description

Apache Log4j2 2.0-beta9 through 2.15.0 (excluding security releases 2.12.2, 2.12.3, and 2.3.1) JNDI features used in configuration, log messages, and parameters do not protect against attacker controlled LDAP and other JNDI related endpoints. An attacker who can control log messages or log message parameters can execute arbitrary code loaded from LDAP servers when message lookup substitution is enabled. From log4j 2.15.0, this behavior has been disabled by default. From version 2.16.0 (along with 2.12.2, 2.12.3, and 2.3.1), this functionality has been completely removed. Note that this vulnerability is specific to log4j-core and does not affect log4net, log4cxx, or other Apache Logging Services projects.

Severity

[CVSS Version 3.x](#)[CVSS Version 2.0](#)

CVSS 3.x Severity and Metrics:



NIST: NVD

Base Score: **10.0 CRITICAL**

Vector: CVSS:3.1/AV:N/AC:L/PR:N/UI:N/S:C/C:H/I:H/A:H

INTRODUCTION TO DEVSECOPS

- ▶ What is False Positive Analysis and how its related to DevSecOps?

FALSE POSITIVE ANALYSIS (FPA)

- ▶ **False positive analysis** is the process of identifying and analyzing false positives in a system or process. A false positive, also known as a "false alarm," is an error that occurs when a system or process wrongly identifies something as a threat or problem when it is not.
- ▶ In the context of cybersecurity, false positives refer to instances where a security system or process wrongly identifies a code as being vulnerable. For example, If SonarQube says that CSRF token is not implemented in the system but the token is implemented with some other name would lead to False Positive event generated by SonarQube security tool.
- ▶ False positives can be a problem because they can waste resources, cause unnecessary disruptions, and lead to a lack of trust in the system or process. They can also lead to "alert fatigue," where users become desensitized to alerts and may ignore them even when a real threat is present.
- ▶ This involves improving the accuracy of the system by marking the incorrect security vulnerabilities in the system as False and then generating the report only with True Positives.

FALSE POSITIVE DEMO FOR SAST, SCA, DAST, IAC & CONTAINER SECURITY

INTRODUCTION TO DEVSECOPS

- ▶ Reporting True security vulnerabilities in ticketing tool like JIRA by DevSecOps Engineers

STEPS FOR REPORTING SECURITY VULNERABILITIES

- ▶ **Identify the vulnerability:** First, you need to identify the security vulnerability that you have discovered. This involves reviewing the code, scanning the system using security tools.
- ▶ **Document the vulnerability:** Next, you should document the vulnerability in as much detail as possible. This include information such as the name of the vulnerability, description, application name, how it can be exploited, and any potential impacts or consequences, any reference related to CVE or CVSS score and remediation recommendation.
- ▶ **Create a ticket:** Using the ticketing tool like JIRA, create a new ticket to report the security vulnerability and include all the above information discussed in previous step
- ▶ **Assign the ticket:** Assign the ticket to the appropriate Dev team or Product Owner for the application team responsible for taking decisions on addressing security vulnerabilities.
- ▶ **Follow up and ReTest:** Follow up on the ticket to ensure that it is being addressed in a timely manner. Depending on the severity of the vulnerability, it may be necessary to take additional steps to protect the system until the issue can be resolved.

REPORTING SECURITY VULN DEMO IN JIRA

INTRODUCTION TO DEVSECOPS (INTERMEDIATE LEVEL)

- ▶ DevSecOps Maturity Model

DEVSECOPS MATURITY MODEL

The DevSecOps Maturity Model is a framework that helps organizations understand and improve their DevSecOps practices. It defines a set of best practices and capabilities that organizations can strive to achieve in order to improve the security and quality of their software development and delivery processes.

There are several versions of the DevSecOps Maturity Model, but they generally include the following stages:

- ▶ **Ad Hoc:** In this initial stage, security is an afterthought and is not integrated into the software development process.
- ▶ **Repeatable:** In this stage, security practices are established, but they are still separate from the software development process.
- ▶ **Defined:** In this stage, security practices are integrated into the software development process and are defined in standard procedures and policies.
- ▶ **Managed:** In this stage, security practices are continuously monitored and optimized to improve efficiency and effectiveness.
- ▶ **Optimizing:** In this final stage, security practices are continuously improved and adapted to meet the evolving needs of the organization.

Each stage represents an increase in the level of integration and automation of security practices into the software development process. As organizations progress through the stages, they are able to deliver software more quickly and with higher security and quality.

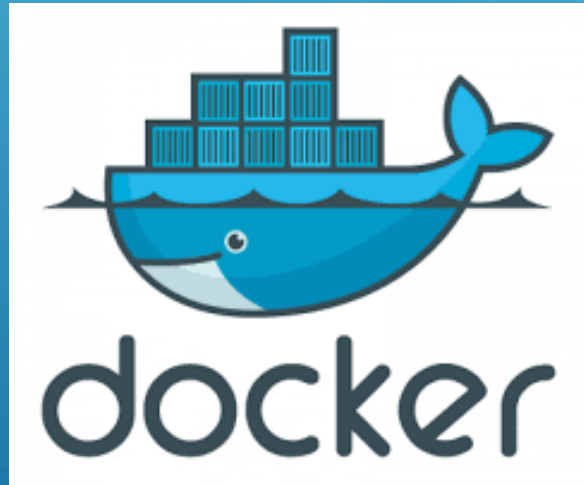


DEVSECOPS MATURITY MODEL BY OWASP



INTRODUCTION TO DEVSECOPS (INTERMEDIATE LEVEL)

- ▶ Docker and its benefits



DOCKER

Docker is a tool that simplifies the process of deploying and running applications by packaging them in lightweight, portable containers. Here are some basic concepts and terms that you should know when working with Docker:

- ▶ **Containers:** A container is a lightweight, standalone, and executable package that includes everything an application needs to run, such as code, libraries, and system tools. Containers allow you to package an application with all of its dependencies in a single package, making it easy to deploy and run on any platform.
- ▶ **Images:** An image is a lightweight, stand-alone, and executable package that contains everything an application needs to run. It is used to create containers.
- ▶ **Docker Engine:** The Docker Engine is the software that powers Docker. It consists of a daemon that runs on the host machine, a REST API that exposes the daemon's functionality, and a command-line interface (CLI) that allows you to interact with the daemon through your terminal.
- ▶ **Docker Hub:** Docker Hub is a cloud-based registry service that allows you to store and manage Docker images. You can use Docker Hub to find and share images with the Docker community, or you can use it to host your own images privately.
- ▶ **Dockerfile:** A Dockerfile is a text file that contains instructions for building a Docker image. It specifies the base image to use, the dependencies to install, and the commands to run when the container is started.

SIMPLE DOCKER FILE SAMPLE

```
FROM ubuntu:18.04  
RUN apt-get update && apt-get install -y nginx  
COPY index.html /var/www/html/index.html  
CMD ["nginx", "-g", "daemon off;"]
```

This Dockerfile specifies that it is based on the Ubuntu 18.04 image, installs the nginx web server, copies an index.html file into the container, and runs the nginx command when the container starts.

DOCKER INSTALLATION AND DEMO

INTRODUCTION TO DEVSECOPS (INTERMEDIATE LEVEL)

- ▶ Kubernetes and its benefits



KUBERNETES

Kubernetes is an open-source platform for automating the deployment, scaling, and management of containerized applications. It provides a set of tools and APIs that allow you to deploy and manage containerized applications in a distributed environment. Here are some basic concepts and terms that you should know when working with Kubernetes:

- ▶ **Cluster:** A Kubernetes cluster is a group of nodes (physical or virtual machines) that run containerized applications. A cluster consists of a set of worker nodes and a single control plane node.
- ▶ **Node:** A node is a physical or virtual machine that runs containerized applications. A node consists of a container runtime (such as Docker), a kubelet (a agent that communicates with the control plane), and a kube-proxy (a network proxy that runs on each node).
- ▶ **Pod:** A pod is the smallest deployable unit in Kubernetes. It is a logical host for one or more containers, which share the same network namespace, storage, and lifecycle.
- ▶ **Deployment:** A deployment is a resource in Kubernetes that manages a set of replicas of a pod. It ensures that the specified number of replicas of the pod are always running and automatically replaces any failed or terminated pods.
- ▶ **Service:** A service is a logical abstraction that defines a set of pods and a policy for accessing them. It enables you to access a group of pods using a single, stable IP address and DNS name.
- ▶ **Ingress:** An ingress is a resource that enables inbound traffic to access Kubernetes services. It acts as a reverse proxy, routing traffic from external sources to the appropriate service based on the hostname and path.

SOME KUBERNETES COMMANDS

INTRODUCTION TO DEVSECOPS (INTERMEDIATE LEVEL)

- ▶ Terraform and its benefits



TERRAFORM

Terraform is an open-source infrastructure as code (IaC) tool that allows you to define, create, and manage infrastructure resources using code. It supports a wide range of infrastructure providers, including cloud providers (such as AWS, Azure, and Google Cloud), as well as on-premises environments. Here are some basic concepts and terms that you should know when working with Terraform:

- ▶ **Infrastructure as code (IaC):** IaC is a practice that allows you to define and manage infrastructure resources using code, rather than manually creating and configuring them. This enables you to version control your infrastructure, automate deployments, and make it easier to collaborate with others.
- ▶ **Resources:** A resource is a component of your infrastructure that represents a physical or logical entity, such as a virtual machine, a database, or a networking component. In Terraform, you can use resources to create, configure, and manage these entities in your infrastructure.
- ▶ **Modules:** A module is a reusable unit of Terraform code that encapsulates a set of resources and their configurations. Modules allow you to abstract and reuse common infrastructure patterns, making it easier to manage and maintain your infrastructure.
- ▶ **State:** Terraform maintains a state file that stores the current state of your infrastructure. The state file tracks the resources that have been created and their configurations, and it is used to determine the necessary actions to reach the desired state defined in your code.
- ▶ **Providers:** A provider is a plugin that integrates Terraform with a specific infrastructure provider, such as AWS or Azure. It allows you to create and manage resources for that provider using Terraform.

SIMPLE TERRAFORM FILE SAMPLE

```
provider "aws" {  
  region = "us-west-2"  
}  
  
resource "aws_s3_bucket" "awesome-s3-bucket" {  
  bucket = "my-bucket"  
  acl    = "private"  
}
```

This configuration specifies the AWS provider and creates an S3 bucket with the name "awesome-s3-bucket" and a private access control list (ACL).

INTRODUCTION TO DEVSECOPS (INTERMEDIATE LEVEL)

- ▶ Git and GitHub Explained



GIT AND GITHUB

Git is a version control system that allows you to track changes to your code and collaborate with others. GitHub is a web-based platform that provides hosting for Git repositories and tools for collaboration, code review, and project management. Here are some basic concepts and terms that you should know when working with Git and GitHub:

- ▶ **Repository:** A repository (or "repo") is a collection of files and the history of changes to those files. In Git, a repository is a local copy of a project that you can modify and track changes to. On GitHub, a repository is a remote copy of a project that is hosted on the platform.
- ▶ **Commit:** A commit is a snapshot of the changes to your code at a particular point in time. In Git, you can commit your changes to your local repository and push them to a remote repository (such as on GitHub) to share them with others.
- ▶ **Branch:** A branch is a separate line of development in a repository. You can use branches to isolate new changes, experiment with new features, or work on fixes without affecting the main codebase.
- ▶ **Merge:** A merge is the process of integrating changes from one branch into another branch. In Git, you can use the merge command to combine the changes from multiple branches into a single branch.
- ▶ **Pull request:** A pull request is a request to merge changes from one branch into another branch. On GitHub, you can use pull requests to propose and review changes to a repository.

GIT AND GITHUB DEMO

INTRODUCTION TO DEVSECOPS (INTERMEDIATE LEVEL)

- ▶ IDE Plugins Explained with Demo



IDE SECURITY PLUGINS

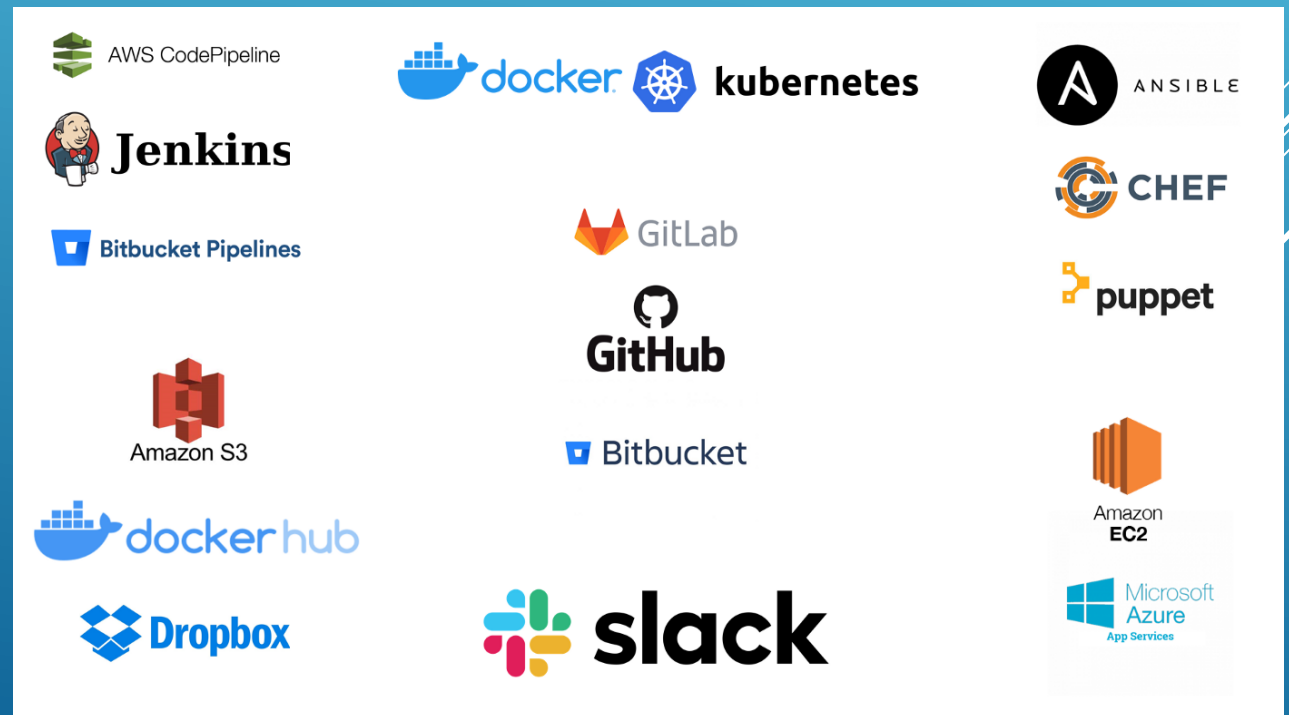
IDE security plugins are tools that can help you identify and fix security vulnerabilities in your code as you write it. They are commonly used in integrated development environments (IDEs), such as Eclipse, IntelliJ, and Visual Studio. Here are some basic concepts and terms that you should know when working with IDE security plugins:

- ▶ **Static code analysis:** Static code analysis is the process of analyzing code for potential vulnerabilities and security issues without executing it. IDE security plugins often include static code analysis tools that can scan your code and highlight potential vulnerabilities and security issues as you write it. Some Plugins are: Fortify, CheckMarx, Veracode IDE Plugins
- ▶ **Code review:** Code review is the process of reviewing code for quality, style, and security issues before it is merged into the main codebase. IDE security plugins may include features that allow you to request and perform code reviews, as well as tools that can help you identify potential security issues during the review process. For example: SonarLint IDE Plugin

IDE PLUGIN DEMO

INTRODUCTION TO DEVSECOPS (INTERMEDIATE LEVEL)

► CI/CD Tools – On Premise and Cloud



IDE SECURITY PLUGINS

CI/CD (Continuous Integration/Continuous Deployment) tools are software tools that enable developers to build, test, and deploy code changes more frequently and reliably. These tools are designed to automate and streamline the software development process, allowing developers to focus on writing code, rather than on manual tasks such as building and deploying code.

There are two main types of CI/CD tools: cloud-based and on-premise.

CD tools can be used to automate a wide range of tasks, including:

- Cloud-based CI/CD tools are hosted in the cloud, and are typically accessed through a web browser or API. These tools offer a number of advantages, including the ability to scale on demand, ease of use, and lower upfront costs. Some examples of cloud-based CI/CD tools include AWS Code Build, GCP Cloud Build, Azure DevOps, GitLab, Travis CI, and CircleCI.
- On-premise CI/CD tools are installed and run on a company's own servers or infrastructure. These tools offer more control and customization options, but also require a larger upfront investment and ongoing maintenance. Some examples of on-premise CI/CD tools include Jenkins, Bamboo, and TeamCity.

In general, CI/CD tools work by automating the process of building, testing, and deploying code changes. This typically involves integrating the tool with a version control system (such as Git), setting up a build pipeline (which defines the steps needed to build and test the code), and configuring deployment processes (which define how and where the code is deployed).

CI/CD tools can be used to automate a wide range of tasks, including:

- ▶ Building and compiling code
- ▶ Running automated tests
- ▶ Generating code coverage reports
- ▶ Deploying code to various environments (such as staging or production)
- ▶ Rolling back code changes in the event of failures or errors
- ▶ Sending notifications about build and deployment status



INTRODUCTION TO DEVSECOPS (INTERMEDIATE LEVEL)

► Cloud Essentials



CLOUD ESSENTIALS

The term "cloud" refers to a network of servers that are connected and accessible over the internet. These servers are used to store, process, and manage data and applications, and are typically owned and operated by a third-party service provider.

There are several different types of cloud services, including:

- ▶ **Infrastructure as a Service (IaaS):** This type of cloud service provides access to computing resources, such as storage, networking, and servers. Customers can use these resources to build and run their own applications and services. For example: AWS EC2 instance
- ▶ **Platform as a Service (PaaS):** This type of cloud service provides a development platform and tools for building, testing, and deploying applications. Customers can use these tools to build and run their own applications, without having to worry about managing the underlying infrastructure. For example: AWS RDS service
- ▶ **Software as a Service (SaaS):** This type of cloud service provides access to software applications that are hosted and managed by the service provider. Customers can use these applications over the internet, and typically pay a subscription fee or usage-based fee. For example: SonarCloud is SaaS service

Cloud services offer a number of advantages, including the ability to scale on demand, lower upfront costs, and improved reliability and security. They are often used to host applications, store data, and run other types of workloads.