Vendor: EC-Council

Exam Code: 312-97

Exam Name: EC-Council Certified DevSecOps Engineer (ECDE)

Version: DEMO

QUESTION 1
Sandra Oliver joined SinClare Soft Pvt. Ltd. as a DevSecOps engineer in January of 2010. Her organization develops software and web applications related to the healthcare industry. Using IAST runtime security testing technology, she is detecting and diagnosing security issues in applications and APIs. The IAST solution used by Sandra encompasses a web scanner with an agent that works inside the server that hosts the application to provide additional analysis details such as the location of the vulnerability in the application code. Based on the given information, which of the following IAST solutions is Sandra using?

A. Active IAST.
B. Semi-passive IAST.
C. Semi-active IAST.
D. Passive IAST.

Answer: C
Explanation:
Interactive Application Security Testing (IAST) solutions are classified based on how they interact with the application and runtime environment. In this scenario, the solution uses a web scanner to actively send requests to the application while also deploying an agent inside the application server to observe runtime behavior and map vulnerabilities directly to source code locations. This combined approach is known as semi-active IAST. It is considered "semi-active" because it actively drives traffic through the application using a scanner, while the agent passively observes execution paths, data flows, and method calls. Passive IAST solutions rely only on observing existing traffic and do not use scanners, while active IAST solutions do not typically rely on deep runtime agents in the same manner. Semi-active IAST significantly reduces false positives and provides precise remediation details, making it highly effective during the Build and Test stage, where applications are actively exercised and security issues can be identified and fixed before release.

QUESTION 2
Judi Dench has recently joined an IT company as a DevSecOps engineer. Her organization develops software products and web applications related to electrical engineering. Judi would like to use Anchore tool for container vulnerability scanning and Software Bill of Materials (SBOM) generation. Using Anchore grype, she would like to scan the container images and file systems for known vulnerabilities, and would like to find vulnerabilities in major operating system packages such as Alpine, CentOS, Ubuntu, etc. as well as language specific packages such as Ruby, Java, etc. Which of the following commands should Judi run to scan for vulnerabilities in the image using grype?

A. grype < image > --scope all-layers.
B. grype packages < image > --scope all-layers.
C. grype packages < image >.
D. grype < image >.

Answer: A
Explanation:
Grype is a vulnerability scanning tool used to analyze container images and file systems for known vulnerabilities across operating system and application dependencies. The most effective way to perform a comprehensive scan is by running the grype <image> --scope all-layers command. This ensures that vulnerabilities are detected across all layers of the container image, not just the final runtime layer. Containers often inherit vulnerabilities from base images or intermediate layers, making full-layer scanning essential. The packages subcommand is used for listing detected packages rather than performing vulnerability analysis. Running Grype during the Build and Test stage allows DevSecOps teams to identify vulnerable base images and

dependencies early, reducing the risk of deploying insecure containers into production and supporting secure container lifecycle management.

QUESTION 3
Jeremy Renner has been working as a senior DevSecOps engineer at an IT company that develops customized software to various customers stretched across the globe. His organization is using Microsoft Azure DevOps Services. Using an IaC tool, Jeremey deployed the infrastructure in Azure. He would like to integrate Chef InSpec with Azure to ensure that the deployed infrastructure is in accordance with the architecture and industrial standards and the security policies are appropriately implemented. Therefore, he downloaded and installed Chef InSpec. He used Azure CLI command for creating an Azure Service Principal with reader permission to the Azure resources, then he exported the generated credentials. After installation and configuration of Chef InSpec, he would like to create the structure and profile. Which of the following commands should Jeremy use to create a new folder jyren-azureTests with all the required artifacts for InSpec tests?

A. inspec init prof jyren-azureTests.
B. inspec init profile jyren-azureTests.
C. chef inspec init profile jyren-azureTests.
D. chef inspec init profile jyren-azureTests.

Answer: B
Explanation:
Chef InSpec provides a command-line interface for creating and executing compliance profiles. To initialize a new profile with the required directory structure, metadata file, and example controls, the correct command is inspec init profile <profile-name>. In Jeremy's case, running inspec init profile jyren-azureTests creates a new folder with all required artifacts needed to write and run Azure compliance tests. Options using prof are invalid abbreviations, and prefixing the command with chef is incorrect when using the InSpec CLI directly. Creating a structured InSpec profile during the Build and Test stage enables automated validation of infrastructure against architectural standards and security policies, supporting Infrastructure as Code security and continuous compliance practices.

QUESTION 4
Brady Coleman is a senior DevSecOps engineer at CloudVac Security Private Ltd. He has created a new container named "eccbrad" from the centos:7 image using the command docker run -i -t --name geeklab centos:7 /bin/bash. Now, Brady wants to install the httpd package inside the eccbrad container. Which of the following commands should Brady use to install the httpd package inside the container?

A. sudo install-httpd.
B. sudo install httpd.
C. yum install-httpd.
D. yum install httpd.

Answer: D
Explanation:
The CentOS operating system uses the yum package manager to install software packages. Inside a CentOS-based Docker container, the correct way to install the Apache HTTP Server is by running yum install httpd. Containers typically run as the root user by default, making the use of sudo unnecessary. Commands such as install-httpd are invalid because yum requires the install keyword followed by the package name separated by a space. Installing required packages inside containers should be done carefully to avoid bloating images and increasing the attack

surface. During the Operate and Monitor stage, DevSecOps teams must balance functionality with container hardening best practices by installing only necessary components.

QUESTION 5
Elizabeth Moss has been working as a DevSecOps engineer in an IT company located in San Diego, California. Due to the robust security and cost-effective service provided by AWS, her organization transferred all the workloads from on-prem to AWS cloud in 2017. Elizabeth would like to prevent committing AWS keys into repositories; therefore, she created a global git-templates directory using command line. Then, she created another directory, named it as hooks, wherein she created a file named pre- commit. In the pre-commit file, Elizabeth pasted the script that would prevent committing AWS keys into the repositories. She would like to ensure that the hook is executable. Which of the following command should Elizabeth run to make sure that the pre-commit hook is executable?

A. chmod a+x ~/.git-templates/hooks/pre-commit.
B. chmod a+e ~/.git-templates/hooks/pre-commit.
C. chmod a+x ~/.hooks/git-templates/pre-commit.
D. chmod a+e ~/.hooks/git-templates/pre-commit.

Answer: A
Explanation:
Git hooks must have executable permissions to run automatically during Git operations such as commits. The standard way to make a file executable on Unix-like systems is by using the chmod command with the +x flag. In Elizabeth's setup, the pre-commit hook is located in the ~/.git-templates/hooks/ directory, so the correct command is chmod a+x ~/.git-templates/hooks/pre-commit. The a+x option grants execute permission to all users, ensuring that the hook runs regardless of the user context. Options using +e are invalid because e is not a recognized permission flag. Ensuring that the hook is executable during the Code stage helps prevent accidental exposure of AWS credentials by enforcing security checks before commits are finalized.

QUESTION 6
Christopher Brown has been working as a DevSecOps engineer in an IT company that develops software and web applications for an ecommerce company. To automatically detect common security issues and coding error in the C++ code, she performed code scanning using CodeQL in GitHub. Which of the following entries will Christopher find for CodeQL analysis of C++ code?

A. CodeQL/Analyze (cp) (pull-request).
B. CodeQL/Analyze (cp) (push-request).
C. CodeQL/Analyze (cpp) (push-request).
D. CodeQL/Analyze (cpp) (pull-request).

Answer: D
Explanation:
When GitHub Code Scanning is enabled using CodeQL, each supported programming language is identified by a specific language key. For C++ code, CodeQL uses the identifier cpp, not "cp." CodeQL workflows are commonly configured to run during pull request events so that security issues and coding errors can be detected and reviewed before code is merged into the main branch. As a result, the CodeQL analysis entry displayed in GitHub Actions and the Security tab for C++ pull request analysis appears as CodeQL/Analyze (cpp) (pull-request). Options A and B are incorrect because "cp" is not a valid CodeQL language identifier. Option C uses the correct language identifier but references an incorrect event format. Identifying the correct CodeQL analysis entry helps DevSecOps engineers confirm that scans are executing correctly for the

intended language during the Code stage and that security feedback is available early in the development lifecycle.

QUESTION 7
Evan Peters has been working as a DevSecOps engineer in an IT company located in Denver, Colorado. His organization has deployed various applications on Docker containers. Evan has been running SSH service inside the containers, and handling of SSH keys and access policies is a major security concern for him. What will be the solution for Evan security concern?

A. Run SSH on the registry and utilize docker exec for interacting with the container.
B. Run SSH on the docker build and utilize docker exec for interacting with the container.
C. Run SSH on the client and utilize docker exec for interacting with the container.
D. Run SSH on the host and utilize docker exec for interacting with the container.

Answer: D
Explanation:
Running an SSH service inside Docker containers is considered a security anti-pattern because it increases the attack surface and complicates key and access management. Containers are designed to run a single primary process and be managed externally rather than accessed via SSH. The recommended solution is to run SSH on the host system and use docker exec to interact with containers when administrative access is required. This approach eliminates the need to manage SSH keys inside containers, reduces exposure to brute-force attacks, and simplifies access control. The other options incorrectly suggest running SSH in inappropriate locations such as the registry, client, or build process, which do not address the core security concern. During the Operate and Monitor stage, minimizing unnecessary services within containers is critical to enforcing least privilege and maintaining a secure runtime environment.

QUESTION 8
Craig Kelly has been working as a software development team leader in an IT company over the past 8 years. His team is working on the development of an Android application product. Sandra Oliver, a DevSecOps engineer, used DAST tools and fuzz testing to perform advanced checks on the Android application product and detected critical and high severity issues. She provided the information about the security issues and the recommendations to mitigate them to Craig's team. Which type of security checks performed by Sandra involve detection of critical and high severity issues using DAST tools and fuzz testing?

A. Commit-time checks.
B. Build-time checks.
C. Deploy-time checks.
D. Test-time checks.

Answer: D
Explanation:
Dynamic Application Security Testing (DAST) and fuzz testing require a running application in order to actively probe for vulnerabilities such as injection flaws, authentication bypasses, and improper input handling. These techniques are therefore performed after the application has been built and deployed to a testing environment, categorizing them as test-time checks. Commit-time and build- time checks rely primarily on static analysis and dependency scanning and do not exercise application behavior at runtime. Deploy-time checks focus on configuration validation rather than aggressive attack simulation. Test-time checks are specifically designed to uncover critical and high- severity vulnerabilities by mimicking real-world attack scenarios. Performing DAST and fuzz testing during this stage allows teams to detect exploitable flaws before production release, significantly strengthening application security.

QUESTION 9
SinCaire is a software development company that develops web applications for various clients. To measure the successful implementation of DevSecOps, the organization enforced U.S. General Service Administrator (GSA) high-value DevSecOps metrics. Which of the following metrics implemented by SinCaire can measure the time between the code commit and production, and tracks the bug fix and new features throughout the development, testing, and production phases?

A. Mean time to recovery (for applications).
B. Change volume (for application).
C. Time to value.
D. Change lead time (for application).

Answer: D
Explanation:
Change lead time measures the duration between a code commit and its successful deployment into production. This metric tracks how efficiently new features, bug fixes, and changes move through development, testing, and release stages. It is a key DevSecOps performance indicator used to assess pipeline efficiency and the effectiveness of automation and security integration. Mean time to recovery focuses on restoring service after incidents, change volume measures the number of changes rather than delivery speed, and time to value is a broader business metric. Change lead time directly reflects how well DevSecOps practices enable rapid yet secure delivery, making it the correct metric for measuring commit-to-production flow across all phases.

QUESTION 10
Dustin Hoffman has been working as a DevSecOps engineer in an IT company located in San Diego, California. For detecting new security vulnerabilities at the beginning of the source code development, he would like to integrate Checkmarx SCA tool with GitLab. The Checkmarx template has all the jobs defined for pipeline. Where should Dustin incorporate the Checkmarx template file `https://raw.githubusercontent.com/checkmarx-ltd/cx-flow/develop/templates/gitlab/v3/Checkmarx.gitlab-ci.yml'?

A. gitlab-cd.yml root directory.
B. gitlab-ci/cd.yml root directory.
C. gitlab.yml root directory.
D. gitlab-ci.yml root directory.

Answer: D
Explanation:
GitLab CI/CD pipelines are defined using a configuration file named gitlab-ci.yml, which must be placed in the root directory of the repository. This file controls pipeline stages, jobs, and template inclusions. To integrate Checkmarx SCA using a predefined template, the template reference must be included in the root-level gitlab-ci.yml file so GitLab can load and execute the defined jobs automatically. The other filenames listed in the options are not recognized by GitLab as valid pipeline configuration files. Integrating SCA at the Code stage allows early detection of vulnerable open-source dependencies, reducing remediation cost and preventing insecure components from progressing further in the DevSecOps pipeline.

QUESTION 11
Robin Tunney has been working as a DevSecOps engineer in an IT company located in Charleston, South Carolina. She would like to build a customized docker image using HashiCorp

Packer. Therefore, she installed Packer and created a file docker-ubuntu.pkr.hcl; she then added HCL block to it and saved the file. Which of the following commands should Robin execute to build the Docker image using Packer?

A. packer build docker-ubuntu.pkr.hcl.
B. packer -b docker-ubuntu.pkr.hcl.
C. packer b docker-ubuntu.pkr.hcl.
D. packer -build docker-ubuntu.pkr.hcl.

Answer: A
Explanation:
HashiCorp Packer is an image automation tool that uses the packer build command to create machine images from configuration files written in HCL or JSON. When Robin defines her Docker image configuration in the file docker-ubuntu.pkr.hcl, the correct way to initiate the build process is by running packer build docker-ubuntu.pkr.hcl. This command reads the configuration file, initializes required plugins, executes defined builders and provisioners, and produces the final Docker image. The other options are syntactically incorrect because Packer does not support abbreviated flags such as -b or alternative verbs like -build. Building container images during the Build and Test stage ensures that images are reproducible, standardized, and compliant with organizational security requirements before deployment. Using Packer also supports immutability and reduces configuration drift, which are key principles in secure DevSecOps pipelines.

QUESTION 12
Katie Holmes is working as a DevSecOps engineer at SeCSafe Anti-virus. The DevOps team of her organization has developed a distributed application with multiple microservices. Katie deployed all the microservices to the Kubernetes nodes successfully. The DevOps team approached Katie and informed her that the application is not working. Katie wants to check whether the Kubernetes cluster is working or not. Which of the following commands should Katie run step by step to verify that the Kubernetes is working?

A. kube-etcd version kube-etcd cluster-info.
B. kubernetes version kubebernetes cluster-info.
C. kube version kube cluster-info.
D. kubectl version kubectl cluster-info.

Answer: D
Explanation:
Kubernetes clusters are managed and inspected using the kubectl command-line tool. To verify whether a Kubernetes cluster is functioning correctly, administrators commonly run kubectl version to confirm that both the client and server components are reachable and operational. This is followed by kubectl cluster-info, which displays information about the cluster's control plane and core services. These commands together confirm API server availability, cluster connectivity, and basic health status. The other options list invalid command names such as kube, kubernetes, or kube- etcd, which are not used for standard cluster validation. Performing these checks during the Operate and Monitor stage helps quickly identify whether application issues stem from cluster-level problems or application-level misconfigurations. This supports faster troubleshooting and more reliable production operations.

QUESTION 13
Thomas McInerney has been working as a senior DevSecOps engineer in an IT company that develops software products and web applications related to the healthcare sector. His organization deployed various applications in Docker containers. Thomas' team leader would like to prevent a container from gaining new privileges. Therefore, he asked Thomas to set

no_new_priv bit, which functions across clone, execve, and fork to prevent a container from gaining new privileges. Which of the following commands should Thomas use to list out security options for all the containers?

A. docker ps --quiet --all | xargs docker inspect --format ': SecurityOpt'.
B. docker ps -quiet -all | xargs docker inspect --format ': SecurityOpt='.
C. docker ps -quiet -all | xargs docker inspect --format ': SecurityOpt'.
D. docker ps --quiet --all | xargs docker inspect --format ': SecurityOpt='.

Answer: D
Explanation:
Docker allows inspection of container runtime configuration using the docker inspect command. To list security-related options such as no_new_privileges for all containers, the correct approach is to first retrieve all container IDs using docker ps --quiet --all and then pass them to docker inspect with a formatted output. The command docker ps --quiet --all | xargs docker inspect --format ': SecurityOpt=' correctly extracts the security options configured for each container. Options that use incorrect flags such as -quiet instead of --quiet, omit required parameters, or misformat the output string are invalid. Inspecting security options during the Operate and Monitor stage helps ensure that privilege escalation protections are enforced consistently, supporting container hardening and compliance with security benchmarks.

QUESTION 14
Alex Hales has been working as a DevSecOps in an IT company that develops software products and web applications for visualizing scientific data. He would like to trigger a Jenkins build job using Git post commit script or hooks that helps his team in saving time by automating commit. Therefore, before triggering the build job, Alex made changes and saved the code in the respective IDE under Git repository and added the changes in the master branch using git add command and ran the post commit script to check the status of the build. Then, he navigated to the Jenkins project and selected the "Trigger build remotely from Build triggers" radio button. It would automate the trigger every time a change gets committed to the project. Alex navigated back to Bash terminal to trigger the build job. Which of the following commands should Alex use in Bash terminal to trigger the build job?

A. git commit -m "commit from terminal".
B. github commit -b "commit from terminal".
C. github commit -m "commit from terminal".
D. git commit -b "commit from terminal".

Answer: A
Explanation:
Git post-commit hooks are executed automatically after a commit is successfully created. To trigger the Jenkins build job configured to respond to commits, Alex must create a valid Git commit using the correct Git command. The standard command to commit changes with a message is git commit - m "commit from terminal". Running this command records the changes in the repository and triggers the post-commit hook, which in turn initiates the Jenkins build. Commands using github commit are invalid because github is not a native Git command-line utility. The -b flag is also not used with git commit. Automating build triggers during the Code stage improves efficiency, reduces manual intervention, and ensures continuous integration is consistently enforced.

Thank You for Trying Our Product

Lead2pass Certification Exam Features:

★ More than **99,900** Satisfied Customers Worldwide.

★ Average **99.9%** Success Rate.

★ **Free Update** to match latest and real exam scenarios.

★ **Instant Download** Access! No Setup required.

★ Questions & Answers are downloadable in **PDF** format and **VCE** test engine format.

★ Multi-Platform capabilities - **Windows, Laptop, Mac, Android, iPhone, iPod, iPad**.

★ **100%** Guaranteed Success or **100%** Money Back Guarantee.

★ **Fast**, helpful support **24x7**.

View list of all certification exams: http://www.lead2pass.com/all-products.html

10% Discount Coupon Code: **ASTR14**