312-97 Simulation Questions - Valid 312-97 Test Cost

Our 312-97 learning prep boosts the self-learning, self-evaluation, statistics report, timing and test stimulation functions and each function plays their own roles to help the clients learn comprehensively. The self-learning and self-evaluation functions of our 312-97 guide materials help the clients check the results of their learning of the 312-97 Study Materials. The timing function of our 312-97 training quiz helps the learners to adjust their speed to answer the questions and keep alert and our study materials have set the timer.

ECCouncil 312-97 Exam Syllabus Topics:

Topic	Details
Topic 1	• Understanding DevOps Culture: This module introduces DevOps principles, covering cultural and technical foundations that emphasize collaboration between development and operations teams. It addresses automation, CI • CD practices, continuous improvement, and the essential communication patterns needed for faster, reliable software delivery.
Topic 2	• DevSecOps Pipeline - Plan Stage: This module covers the planning phase, emphasizing security requirement identification and threat modeling. It highlights cross-functional collaboration between development, security, and operations teams to ensure alignment with security goals.
Topic 3	• DevSecOps Pipeline - Code Stage: This module discusses secure coding practices and security integration within the development process and IDE. Developers learn to write secure code using static code analysis tools and industry-standard secure coding guidelines.
Topic 4	• DevSecOps Pipeline - Build and Test Stage: This module explores integrating automated security testing into build and testing processes through CI pipelines. It covers SAST and DAST approaches to identify and address vulnerabilities early in development.

Topic 5	• Introduction to DevSecOps: This module covers foundational DevSecOps concepts, focusing on integrating security into the DevOps lifecycle through automated, collaborative approaches. It introduces key components, tools, and practices while discussing adoption benefits, implementation challenges, and strategies for establishing a security-first culture.

>> 312-97 Simulation Questions <<

Valid 312-97 Test Cost, 312-97 Exam Score

As candidates, the quality must be your first consideration when buying 312-97 learning materials. We have a professional team to collect the first-hand information for the exam. Our company have reliable channel for collecting 312-97 learning materials. We can ensure you that 312-97 exam materials you receiveare the latest version. We have strict requirements for the 312-97 Questions and answers, and the correctness of the answers can be guaranteed. In order to serve our customers better, we offer free update for you, so that you can get the latest version timely.

ECCouncil EC-Council Certified DevSecOps Engineer (ECDE) Sample Questions (Q31-Q36):

NEW QUESTION # 31
(Alex Hales recently joined TAVR Software Solution Pvt. Ltd. As a DevSecOps engineer. To automatically detect security loopholes in the web applications while building and testing them, he integrated OWASP ZAP DAST Plugin with Jenkins. How can Alex uniquely identify every build in the project?.)

- A. By specifying a file name followed by ${Profile_ID} in Post-build Actions tab.
- B. By specifying a file name followed by ${ZAPROXY_HOME} in Post-build Actions tab.
- C. By specifying a file name followed by ${Build_ID} in Post-build Actions tab.
- D. By specifying a file name followed by ${zap_scan} in Post-build Actions tab.

Answer: C

Explanation:
Jenkins automatically assigns a unique identifier to each build using the environment variableBUILD_ID.
When integrating OWASP ZAP with Jenkins, appending ${BUILD_ID} to output filenames or reports ensures that every scan result corresponds to a specific build execution. This avoids overwriting previous reports and allows traceability between build artifacts and security findings. Variables such as
${ZAPROXY_HOME} refer to installation paths, not build uniqueness, while ${Profile_ID} and
${zap_scan} are not standard Jenkins variables for uniquely identifying builds. Using ${BUILD_ID} supports better auditing, historical analysis, and correlation between detected vulnerabilities and the exact build in which they were found, which is critical during the Build and Test stage of a DevSecOps pipeline.
═══════

NEW QUESTION # 32
(Matt LeBlanc has been working as a DevSecOps engineer in an IT company that develops software products and web applications for IoT devices. His team leader has asked him to use GitRob tool to find sensitive data in the organizational public GitHub repository. To install GitRob, Matt ensured that he has correctly configured Go >= 1.8 environment and that $GOPATH/bin is in his $PATH. The GitHub repository URL from which he is supposed to install the tool ishttps://github.com/michenriksen/gitrob. Which of the following command should Matt use to install GitRob?.)

- A. $ go get gitrob github.com/michenriksen/gitrob.
- B. $ go git github.com/michenriksen/gitrob.
- C. $ go get github.com/michenriksen/gitrob.
- D. $ go git gitrob github.com/michenriksen/gitrob.

Answer: C

Explanation:

In Go-based tool installation, the standard method to download, compile, and install a Go package is using the go get command followed by the repository import path. Since Matt has already ensured that Go version 1.8 or later is installed and that $GOPATH/bin is included in the system PATH, running go get github.com /michenriksen/gitrob will fetch the GitRob source code, build the binary, and place it in the appropriate bin directory. Options B, C, and D are invalid because go get does not accept multiple positional arguments in that manner, and go git is not a valid Go command. Installing GitRob during the Code stage enables DevSecOps teams to scan repositories for accidentally committed credentials, API keys, and other sensitive information, helping prevent data leakage from public repositories.
==========

NEW QUESTION # 33
(Maria Howell is working as a senior DevSecOps engineer at Global SoftSec Pvt. Ltd. Her team is currently working on the development of a cybersecurity software. There are 5 developers who are working on code development. Howell's team is using a private GitHub repository for the source code development. Which of the following commands should Howell use to grab the online updates and merge them with her local work?.)

- A. $ git get remotename branchname.
- B. $ git push remotename branchname.
- C. $ git pull remotename branchname.
- D. $ git grabs remotename branchname.

Answer: C

Explanation:
The git pull command is used to fetch changes from a remote repository and automatically merge them into the current local branch. In collaborative development environments, especially when multiple developers are committing code to a shared repository, regularly pulling updates is essential to stay synchronized and avoid merge conflicts. The syntax git pull <remote-name> <branch-name> correctly specifies the source of the updates. Commands such as git get and git grabs do not exist in Git, and git push performs the opposite action by sending local changes to the remote repository rather than retrieving updates. Using git pull during the Code stage supports continuous collaboration and ensures that developers integrate the latest changes securely and efficiently.
==========

NEW QUESTION # 34
(Brett Ryan has been working as a senior DevSecOps engineer in an IT company in Charleston, South Carolina. He is using git-mutimail tool to send email notification for every push to git repository. By default, the tool will send one output email providing details about the reference change and one output email for every new commit due to a reference change. How can Brett ensure that git-multimail is set up appropriately?)

- A. Running the environmental variable GIT_MULTIMAIL_CHECK_SETUP by setting it to empty string.
- B. Running the environmental variable GITHUB_MULTIMAIL_CHECK_SETUP by setting it to non- empty string.
- C. Running the environmental variable GITHUB_MULTIMAIL_CHECK_SETUP by setting it to empty string.
- D. Running the environmental variable GIT_MULTIMAIL_CHECK_SETUP by setting it to non-empty string.

Answer: D

Explanation:
The git-multimail tool provides a mechanism to verify whether it has been installed and configured correctly before being relied upon for production notifications. This verification is done using an environment variable namedGIT_MULTIMAIL_CHECK_SETUP. When this variable is set to anon-empty string, git-multimail performs a setup validation and outputs diagnostic information toconfirm that configuration values, hooks, and parameters are correctly defined. This helps prevent silent failures where commits occur but email notifications are not sent. Options that referenceGITHUB_MULTIMAIL_CHECK_SETUPare incorrect because git-multimail is not limited to GitHub and does not use that variable name. Additionally, setting the variable to an empty string does not trigger the setup check. Ensuring proper configuration during the Code stage is important because it supports auditability, traceability, and timely communication among development and security teams. Therefore, Brett must run the environment variable GIT_MULTIMAIL_CHECK_SETUPwith a non-empty value to ensure the tool is set up appropriately.
==========

NEW QUESTION # 35

(Patricia Cornwell has been working as a DevSecOps engineer in an IT company that provides custom software solutions. She would like to use GitMiner to mine the secret credentials such as usernames and passwords, API credentials, and other sensitive data from GitHub. Therefore, to start the scanning, she cloned the repo to the local machine by using the git clonehttp://github.com/UnkL4b/GitMinercommand; then, she moved to the current directory using $ cd GitMiner command. Which of the following commands should Patricia use to install the dependencies?)

- A. pip3 install -r requirement.txt.
- B. pip3 install -d requirement.txt.
- C. pip3 install -q requirement.txt.
- D. pip3 install -m requirement.txt.

Answer: A

Explanation:
GitMiner is a Python-based tool, and like most Python projects, it manages its dependencies through a requirements file named requirements.txt. The correct way to install all dependencies listed in this file is by using the pip3 install -r requirements.txt command. The -r flag instructs pip to read package names and versions from the specified file and install them accordingly. The other flags shown in the options do not correspond to dependency installation from a requirements file and would result in command errors or unexpected behavior. Installing dependencies correctly is a prerequisite for running GitMiner successfully.
During the Code stage, tools like GitMiner help identify hard-coded secrets and sensitive information early, reducing the risk of credential leakage and preventing security incidents later in the DevSecOps pipeline.

NEW QUESTION # 36
.....

No matter how good the product is users will encounter some difficult problems in the process of use. Our 312-97 real exam materials are not exceptional also, in order to enjoy the best product experience, as long as the user is in use process found any problem, can timely feedback to us, for the first time you check our 312-97 Exam Question performance, professional maintenance staff to help users solve problems. Our 312-97 learning reference files have a high efficient product maintenance team, and they can send the 312-97 exam questions to you in a few minutes.