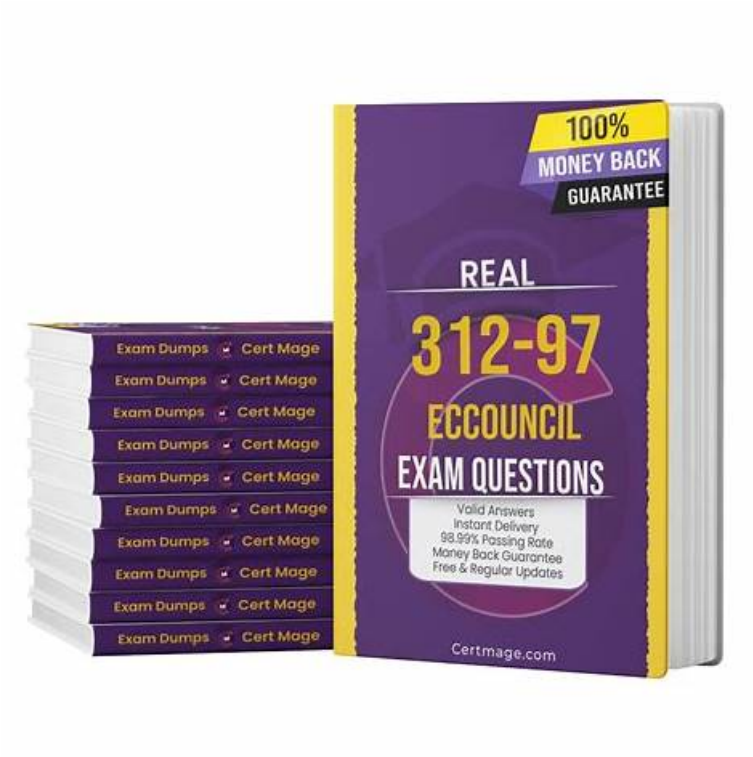
# ECCouncil 312-97 Realistic New Exam Discount




No matter you are exam candidates of high caliber or newbies, our ECCouncil 312-97 exam quiz will be your propulsion to gain the best results with least time and reasonable money. Not only because the outstanding content of EC-Council Certified DevSecOps Engineer (ECDE) 312-97 Real Dumps that produced by our professional expert but also for the reason that we have excellent vocational moral to improve our EC-Council Certified DevSecOps Engineer (ECDE) 312-97 learning materials quality.

Different from traditional learning methods, our 312-97 exam products adopt the latest technology to improve your learning experience. We hope that all candidates can try our free demo before deciding to buy our 312-97 study guide. The Q&A contained in the free demo are also compiled by our vetaren professionals who keep close on the changes of the 312-97 learning dumps according to the real exam. Come and have a try, you will get satisfied with our 312-97 training engine!

**>> New 312-97 Exam Discount <<**

## Newest New 312-97 Exam Discount - Best Accurate Source of 312-97 Exam

Success in the ECCouncil 312-97 Exam paves the way toward high-paying jobs, promotions, and skills verification. Hundreds of ECCouncil 312-97 test takers don't get success because of using ECCouncil outdated dumps. Due to failure, they lose money, time, and confidence. All these losses can be prevented by using updated and real ECCouncil Dumps of RealValidExam.

## ECCouncil 312-97 Exam Syllabus Topics:

| Topic | Details |
|---|---|
| Topic 1 | • 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 2 | • 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<br>• CD practices, continuous improvement, and the essential communication patterns needed for faster, reliable software delivery. |
| Topic 3 | • 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 4 | • 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. |
| Topic 5 | • DevSecOps Pipeline - Release and Deploy Stage: This module explains maintaining security during release and deployment through secure techniques and infrastructure as code security. It covers container security tools, release management, and secure configuration practices for production transitions. |

## ECCouncil EC-Council Certified DevSecOps Engineer (ECDE) Sample Questions (Q96-Q101):

**NEW QUESTION # 96**
(Erica Mena has been working as a DevSecOps engineer in an IT company that provides customize software solutions to various clients across United States. To protect serverless and container applications with RASP, she would like to create an Azure container instance using Azure CLI in Microsoft PowerShell. She created the Azure container instance and loaded the container image to it. She then reviewed the deployment of the container instance. Which of the following commands should Erica run to get the logging information from the Azure container instance? (Assume the resource group name as ACI and container name as aci-test- clash.))

- A. az container logs --resource-group ACI --name aci-test-clash.
- B. az container logs -resource-group ACI -name aci-test-clash.
- C. az get container logs -resource-group ACI --name aci-test-clash.
- D. az get container logs --resource-group ACI --name aci-test-clash.

**Answer: A**

Explanation:
Azure Container Instances provide built-in logging capabilities that can be accessed using the Azure CLI. To retrieve logs from a deployed container instance, the correct command isaz container logsfollowed by the resource group and container name. The proper syntax requires double-dash parameters:--resource-groupand
--name. In Erica's case, the correct command is az container logs --resource-group ACI --name aci-test-clash.
Options that use "az get container logs" are invalid because "get" is not a supported verb in this context.
Option C uses incorrect single-dash flags, which do not match Azure CLI standards. Accessing container logs during the Code stage helps engineers validate application behavior, identify runtime errors, and ensure that security instrumentation such as RASP agents are functioning correctly before progressing further in the pipeline.
═══════

**NEW QUESTION # 97**
(Michael Rady recently joined an IT company as a DevSecOps engineer. His organization develops software products and web applications related to online marketing. Michael deployed a web application on Apache server. He would like to safeguard the deployed application from diverse types of web attacks by deploying ModSecurity WAF on Apache server. Which of the following command should Michael run to install ModSecurity WAF?)

- A. sudo apt install libapache2-mod-security2 -y.
- B. sudo apt install libapache2-mod-security2 -x.
- C. sudo apt install libapache2-mod-security2 -w.

- D. sudo apt install libapache2-mod-security2 -z.

**Answer: A**

Explanation:
On Debian- and Ubuntu-based systems, ModSecurity for Apache is installed using the package libapache2- mod-security2. The correct command to install this package is sudo apt install libapache2-mod-security2 -y, where the -y flag automatically confirms installation prompts. The other options include invalid flags that are not recognized by the package manager and would result in command failure. Installing ModSecurity during the Operate and Monitor stage provides an additional layer of defense by inspecting incoming HTTP requests and blocking malicious traffic such as SQL injection, cross-site scripting, and protocol violations. A Web Application Firewall helps protect deployed applications from common attack vectors and supports defense- in-depth strategies in production environments.

**NEW QUESTION # 98**
(Trevor Noah has been working as a DevSecOps engineer in an IT company located in Detroit, Michigan. His team leader asked him to perform continuous threat modeling using ThreatSpec. To do so, Trevor installed and initialized ThreatSpec in the source code repository; he then started annotating the source code with security issues, actions, or concept. Trevor ran ThreatSpec against the application code and he wants to generate the threat model report. Which of the following command Trevor should use to generate the threat model report using ThreatSpec?.)

- A. $ ThreatSpec report.
- B. $ ThreatSpec Report.
- C. $ threatspec report.
- D. $ Threatspec Report.

**Answer: C**

Explanation:
ThreatSpec is a command-line tool that follows standard Unix-style conventions, where commands are lowercase. To generate a threat model report after annotating source code, the correct command is threatspec report. Commands using incorrect casing or capitalization will fail because the CLI is case-sensitive. Options A, B, and C incorrectly capitalize either the command or the subcommand. Generating threat model reports during the Plan stage allows DevSecOps teams to continuously identify, document, and visualize security threats as the code evolves. This practice embeds threat modeling directly into the development lifecycle, enabling early risk identification and more secure system design decisions.
==========

**NEW QUESTION # 99**
(Nicholas Cascone has recently been recruited by an IT company from his college as a DevSecOps engineer.
His team leader asked him to integrate GitHub Webhooks with Jenkins. To integrate GitHub Webhooks with Jenkins, Nicholas logged in to GitHub account; he then selected Settings > Webhooks > Add Webhook. In the Payload URL field, he is supposed to add Jenkins URL. Which of the following is the final Jenkins URL format that Nicholas should add in Payload URL field of GitHub to configure GitHub Webhooks with Jenkins?.)

- A. http://address:port/github-webhook/.
- B. http://address:port/github_webhook/.
- C. http://address:port/GitHub.webhook/.
- D. http://address:port/GiHhub-webhook/.

**Answer: A**

Explanation:
Jenkins exposes a predefined endpoint for receiving GitHub webhook events. This endpoint is /github- webhook/ and must be appended to the Jenkins base URL in the GitHub webhook configuration. Option C correctly matches the required endpoint format. The other options use incorrect casing, separators, or naming conventions that Jenkins does not recognize. Correct webhook configuration ensures that Jenkins jobs are automatically triggered when code changes occur in GitHub repositories. This integration supports continuous integration and immediate feedback during the Code stage of the DevSecOps pipeline.
==========

**NEW QUESTION # 100**
(Cindy Williams has recently joined an IT company as a DevSecOps engineer. She configured Bundle-Audit in Travis CI. Cindy detected vulnerability in Gemfile dependencies and resolved it by adding some line of codes. How does Bundler scan Gemfile.lock for insecure versions of gems?)

- A. By taking the information from the travis.yml file and comparing it with the known vulnerabilities.
- B. By taking the information from the Gemfile and comparing it with the known vulnerabilities.
- C. By taking the information from the travis.yml and comparing it with the unknown vulnerabilities.
- D. By taking the information from the Gemfile and comparing it with the unknown vulnerabilities.

**Answer: B**

Explanation:
Bundler-Audit is a Software Composition Analysis (SCA) tool designed specifically for Ruby applications. It scans theGemfile and Gemfile.lockto identify all declared dependencies and their resolved versions. The Gemfile specifies which gems the application depends on, while the Gemfile.lock ensures consistent dependency versions across environments. Bundler-Audit compares this dependency information against a database ofknown vulnerabilitiesto identify insecure or outdated gems. It does not rely on the Travis CI configuration file for vulnerability detection, nor does it compare against unknown vulnerabilities. Integrating Bundler-Audit into the Build and Test stage ensures that vulnerable third-party libraries are detected early, allowing developers to remediate issues before the application progresses further in the pipeline. This practice supports shift-left security and reduces the risk of introducing known vulnerabilities into production systems.
==========

**NEW QUESTION # 101**
.....

The software version is one of the three versions of our 312-97 actual exam, which is designed by the experts from our company. The functions of the software version are very special. For example, the software version can simulate the real exam environment. If you buy our 312-97 study questions, you can enjoy the similar real exam environment. So do not hesitate and buy our 312-97 preparation exam, you will benefit a lot from our products.

bookmarkick.com, rafaelshvh998203.verybigblog.com, poppyikpx929054.hazeronwiki.com, Disposable vapes