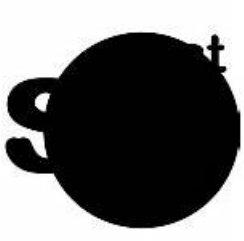
Valid 312-97 Exam Sims | 100% Free Valid Latest EC-Council Certified DevSecOps Engineer (ECDE) Questions



It follows its goal by giving a completely free demo of real ECCouncil 312-97 exam questions. The free demo will enable users to assess the characteristics of the ECCouncil 312-97 Exam product. Dumpleader will provide you with free ECCouncil 312-97 actual questions updates for 365 days after the purchase of our product.

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	• 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.
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	• 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.

>> Valid 312-97 Exam Sims <<

Top Valid 312-97 Exam Sims 100% Pass | High-quality 312-97: EC-Council Certified DevSecOps Engineer (ECDE) 100% Pass

The web-based ECCouncil 312-97 practice exam is compatible with all browsers like Chrome, Mozilla Firefox, MS Edge, Internet Explorer, Safari, Opera, and more. Unlike the desktop version, it requires an internet connection. The EC-Council Certified DevSecOps Engineer (ECDE) (312-97) practice exam will ask real EC-Council Certified DevSecOps Engineer (ECDE) (312-97) exam questions. Consistent practice with it relieves exam stress and boosts self-confidence. The web-based EC-Council Certified DevSecOps Engineer (ECDE) (312-97) practice exam does not require additional software installation. All operating systems also support this EC-Council Certified DevSecOps Engineer (ECDE) (312-97) practice test.

ECCouncil EC-Council Certified DevSecOps Engineer (ECDE) Sample Questions (Q10-Q15):

NEW QUESTION # 10
(Bruce Altman is a DevSecOps engineer at a web application development company named TechSoft Pvt.
Ltd. Due to robust security features provided by Microsoft Azure, in January of 2020, his organization migrated all the workloads from on-prem to Azure. Using Terraform configuration management tool, Bruce created a resource group and virtual machine (VM) in Azure; he then deployed a web application in the VM.
Within an hour, Bruce's team leader informed him that he detected various security issues in the application code and asked him to destroy the infrastructure that he has created in Microsoft Azure using Terraform.
Which of the following commands can Bruce use to destroy the infrastructure created using Terraform?.)

- A. terraform destroy-infra.
- B. terraform kill.
- C. terraform destroy.
- D. terraform kill-infra.

Answer: C

Explanation:
Terraform provides the terraform destroy command to remove all infrastructure resources defined in the Terraform configuration files. This command safely tears down resources such as virtual machines, networks, and resource groups by consulting the state file and executing destruction in the correct dependency order.
Commands like terraform kill, terraform kill-infra, and terraform destroy-infra do not exist in Terraform's CLI. Using terraform destroy during the Release and Deploy stage allows DevSecOps teams to quickly remediate risk by removing insecure or non-compliant infrastructure, reinforcing the importance of Infrastructure as Code and controlled lifecycle management.
═══════

NEW QUESTION # 11
(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 GIT_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 GITHUB_MULTIMAIL_CHECK_SETUP by setting it to non- empty string.

Answer: B

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 # 12
(Dave Allen is working as a DevSecOps engineer in an IT company located in Baltimore, Maryland. His team is working on the development of Ruby on Rails application. He integrated Brakeman with Jenkins to detect security vulnerabilities as soon as they are introduced; he then installed and configured Warnings Next Generation Plugin in Jenkins. What will be the use of Warnings Next Generation Plugin to Dave?.)

- A. It will regulate the function of Brakeman.
- B. It will inspect TypeScript code for readability, functionality, and maintainability issues.
- C. It will gather and manage the results from Brakeman.
- D. It will validate Jenkins compiler settings.

Answer: C

Explanation:
The Warnings Next Generation Plugin in Jenkins is designed tocollect, aggregate, visualize, and manage static analysis resultsproduced by various tools, including Brakeman. In this scenario, Dave uses Brakeman to scan Ruby on Rails applications for security vulnerabilities. Brakeman generates output files containing findings, and the Warnings Next Generation Plugin parses these results and presents them in a standardized, user-friendly format within Jenkins. This allows teams to track trends, enforce quality gates, and fail builds based on severity thresholds. The plugin does not inspect TypeScript code, validate compiler settings, or control Brakeman's execution logic. Its role is purely to manage and display analysis results. Using this plugin during the Code stage improves visibility into security issues, supports decision-making, and helps enforce security standards across the development lifecycle.
==========

NEW QUESTION # 13
(Dustin Hoffman is a DevSecOps engineer at SantSol Pvt. Ltd. His organization develops software products and web applications related to mobile apps. Using Gauntlt, Dustin would like to facilitate testing and communication between teams and create actionable tests that can be hooked in testing and deployment process. Which of the following commands should Dustin use to install Gauntlt?.)

- A. $ gem install gauntlt.
- B. $ gems install Gauntlt.
- C. $ gems install gauntlt.
- D. $ gem install Gauntlt.

Answer: A

Explanation:
Gauntlt is a security testing framework written in Ruby and distributed as a Ruby gem. The correct way to install a Ruby gem is using the gem install command followed by the lowercase gem name. RubyGems are case-sensitive and standardized to lowercase naming conventions, which makes gem install gauntlt the correct command. The gems command does not exist in Ruby's package management ecosystem, and using uppercase names such as Gauntlt can lead to installation failures. Installing Gauntlt allows DevSecOps teams to write human-readable security tests and integrate them into CI/CD pipelines, enabling automated and collaborative security validation during the Build and Test stage.
==========

NEW QUESTION # 14
(Cheryl Hines has been working as a senior DevSecOps engineer over the past 5 years in an IT company. Due to the robust features offered by Keywhiz secret management tool such as compatibility with all software, untraceable secrets, no impact of power cut or server outage, etc., Cheryl's organization is using it for managing and distributing secrets. To add a secret using Keywhiz CLI, which of the following commands should Cheryl use?)

- A. $ keywhiz.cli --devsecTrustStore --admin keywhizAdmin login
  $ keywhiz.cli add secret --name mySecretName < mySecretFile.

- B. $ keywhiz.cli --DevSecTrustStore --user keywhizAdmin login
  $ keywhiz.cli add secret --name mySecretName < mySecretFile.
- C. $ keywhiz.cli --devTrustStore --admin keywhizAdmin login
  $ keywhiz.cli add secret --name mySecretName < mySecretFile.
- D. $ keywhiz.cli --devTrustStore --user keywhizAdmin login
  $ keywhiz.cli add secret --name mySecretName < mySecretFile.

Answer: C

Explanation:
Keywhiz CLI requires authentication before secrets can be added. The correct process involves logging in using the --devTrustStore option and authenticating as an administrator using the --admin flag. Once authenticated, the add secret command is used with input redirection to securely store the secret. Options that use incorrect flag names, incorrect casing, or invalid trust store identifiers do not follow Keywhiz CLI syntax.
Adding secrets through Keywhiz instead of embedding them in code supports secure secret distribution and management, which is a fundamental aspect of DevSecOps culture. This approach ensures secrets remain protected, auditable, and available even during outages.

NEW QUESTION # 15
.....

myportal.utt.edu.tt, myportal.utt.edu.tt, myportal.utt.edu.tt, myportal.utt.edu.tt, bbs.t-firefly.com, www.stes.tyc.edu.tw, courses.fearlesstraders.in, Disposable vapes