Newest Test 312-97 Dates | 312-97 100% Free Latest Exam Notes

You will have a sense of achievements when you finish learning our 312-97 study materials. During your practice of the 312-97 preparation guide, you will gradually change your passive outlook and become hopeful for life. We strongly advise you to have a brave attempt. You will never enjoy life if you always stay in your comfort zone. And our 312-97 Exam Questions will help you realize your dream and make it come true.

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	• 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 3	• DevSecOps Pipeline - Operate and Monitor Stage: This module focuses on securing operational environments and implementing continuous monitoring for security incidents. It covers logging, monitoring, incident response, and SIEM tools for maintaining security visibility and threat identification.
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.

Topic 5	• 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 6	• 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.

312-97 Latest Exam Notes - 312-97 Premium Files

One of the best things about Exams-boost is the convenience it offers. You can access our ECCouncil 312-97 dumps PDF format from anywhere and fit you're studying into your busy schedule. No more traveling to a physical classroom, wasting time and money on gas or public transportation. With the EC-Council Certified DevSecOps Engineer (ECDE) (312-97) PDF questions, you can study on your own time, in your own place, and at your own pace.

ECCouncil EC-Council Certified DevSecOps Engineer (ECDE) Sample Questions (Q29-Q34):

NEW QUESTION # 29
(Andrew Gerrard has recently joined an IT company that develops software products and applications as a DevSecOps engineer. His team leader asked him to download a jar application from the organization GitHub repository and run the BDD security framework. Andrew successfully downloaded the jar application from the repository and executed the jar application; then, he cloned the BDD security framework. Which of the following commands should Andrew use to execute the authentication feature?.)

- A. ./gradlew -Dcucumber.options="-tags @authentication -tags ~@skip".
- B. /gradlew -Dcucumber.options="-tags @authentication -tags @skip".
- C. /gradlev -Dcucumber.options="-tags @authentication -tags @skip".
- D. ./gradlev -Dcucumber.options="-tags @authentication -tags ~@skip".

Answer: A

Explanation:
The BDD Security framework is executed through Gradle wrapper commands, and the correct wrapper script on Unix-like systems is ./gradlew (dot-slash indicates "run the wrapper from the current directory"). Options using /gradlew or /gradlev imply an absolute path at filesystem root and are typically incorrect for a cloned project. Also, the wrapper name isgradlew, notgradlev. For executing only the authentication feature (or scenarios tagged for authentication), Cucumber tag expressions are used through the -Dcucumber.options system property. The command must include --tags @authentication to select authentication-tagged scenarios. To skip scenarios tagged "skip," the exclusion operator is used as --tags ~@skip (meaning "exclude @skip").
Options A and B incorrectly include --tags @skip which wouldincludeskipped tests rather than exclude them.
Therefore, ./gradlew -Dcucumber.options="--tags @authentication --tags ~@skip" is the correct choice to run authentication scenarios while excluding anything marked to skip.
==========

NEW QUESTION # 30
(Helena Luke has been working as a DevSecOps engineer in an IT company located in Denver, Colorado. To seamlessly secure source code during build time and enhance the runtime protection functionalities to the source code, she would like to integrate Jscrambler with GitLab. Therefore, she selected a predefined template and successfully downloaded the Jscrambler configuration file. She then placed the file in the project's root folder and renamed it as .jscramblerrc. To prevent the exposure of sensitive information, she opened the Jscrambler configuration file and removed the access and secret keys from it. In which of the following formats does the Jscrambler configuration file exist?.)

- A. HTML.
- B. YAML.
- C. XML.

- D. JSON.

Answer: D

Explanation:
The Jscrambler configuration file .jscramblerrc is written inJSON format. JSON is widely used for configuration because it is lightweight, human-readable, and easily parsed by tools in CI/CD pipelines.
Removing access and secret keys from this file is a recommended security practice to prevent credential leakage when the repository is shared or stored in version control. Instead, credentials are typically injected through environment variables or secure CI/CD secrets. XML, YAML, and HTML are not the formats used by Jscrambler for its primary configuration file. Using JSON-based configuration during the Code stage allows consistent integration with GitLab pipelines while maintaining secure handling of sensitive data.
==========

NEW QUESTION # 31
(Joyce Vincent has been working as a senior DevSecOps engineer at MazeSoft Solution Pvt. Ltd. She would like to integrate Trend Micro Cloud One RASP tool with Microsoft Azure to secure container-based application by inspecting the traffic, detecting vulnerabilities, and preventing threats. In Microsoft Azure PowerShell, Joyce created the Azure container instance in a resource group (ACI) (named "aci-test-closh") and loaded the container image to it. She then reviewed the deployment of the container instance. Which of the following commands should Joyce use to get the logging information from the container?.)

- A. az container logs --resource-group ACI --name aci-test-closh.
- B. azure container logs -resource-group ACI -name aci-test-closh.
- C. az container logs -resource-group ACI -name aci-test-closh.
- D. azure container logs --resource-group ACI --name aci-test-closh.

Answer: A

Explanation:
Azure Container Instances (ACI) exposes container logs via the Azure CLI using the az container logs command. To retrieve logs, you must provide the resource group and the container group name using the long- form parameters --resource-group and --name. Option A matches the correct CLI structure and parameter format: az container logs --resource-group ACI --name aci-test-closh. Options B and D incorrectly use single- dash forms (-resource-group and -name), which are not valid for these long option names. Options C and D incorrectly use azure instead of az; the Azure CLI command group is invoked with az, not azure. Getting logs after deployment review is a critical Operate and Monitor activity: it helps confirm the container started correctly, diagnose runtime errors, and validate that runtime protection (such as a RASP/micro-agent) is functioning. This visibility supports faster incident response and helps ensure the containerized workload remains secure and stable in its runtime environment.
==========

NEW QUESTION # 32
(Timothy Dalton has been working as a senior DevSecOps engineer in an IT company located in Auburn, New York. He would like to use Jenkins for CI and Azure Pipelines for CD to deploy a Java-based app to an Azure Container Service (AKS) Kubernetes cluster. Before deploying Azure Kubernetes Service (AKS) Cluster, Timothy wants to create a Resource group named Jenkins in southindia location. Which of the following commands should Timothy run?.)

- A. azure group create --n Jenkins --loc southindia.
- B. az group create --name Jenkins --location southindia.
- C. azure group create --name Jenkins --location southindia.
- D. az grp create --n Jenkins --loc southindia.

Answer: B

Explanation:
Azure resource groups are created using the Azure CLI command az group create. The --name parameter specifies the resource group name, and --location defines the Azure region. Option A uses the correct CLI prefix (az), command group (group create), and valid parameters. Options B, C, and D are incorrect due to invalid command abbreviations or incorrect CLI prefixes (azure instead of az). Creating a resource group is a foundational step in the Release and Deploy stage, as it provides a logical container for AKS clusters, networking components, and related resources, enabling organized, secure, and manageable deployments.
==========

NEW QUESTION # 33
(Sarah Wheeler is an experienced DevSecOps engineer. She recently joined an IT company that develops software products for customers stretched across the globe. Sarah would like to use a security testing tool that protects the application from false positives, network sniffing, tampering with code, etc. The tool should monitor the incoming traffic to the server and APIs for suspicious activities and help her team in remediating them during runtime. Which of the following tools should Sarah select that will help her team in precisely detecting and remediating the security issues in the application code during runtime?.)

- A. DAST.
- B. RASP.
- C. IAST.
- D. SAST.

Answer: B

Explanation:
Runtime Application Self-Protection (RASP) operates from within the application runtime environment, monitoring incoming traffic, API calls, and execution behavior in real time. Because it has deep visibility into application logic and execution context, RASP can accurately detect attacks such as injection, tampering, and abnormal behavior while minimizing false positives. SAST analyzes source code statically, DAST tests running applications externally, and IAST combines some runtime insight with testing but does not actively block threats. RASP's ability to detect and remediate attacks during runtime makes it ideal for protecting applications in production environments, aligning with the Operate and Monitor stage of the DevSecOps pipeline.

NEW QUESTION # 34
.....

With 312-97 test answers, you are not like the students who use other materials. As long as the syllabus has changed, they need to repurchase new learning materials. This not only wastes a lot of money, but also wastes a lot of time. Our industry experts are constantly adding new content to 312-97 test dumps based on constantly changing syllabus and industry development breakthroughs. We also hired dedicated IT staff to continuously update our question bank daily, so no matter when you buy 312-97 Study Materials, what you learn is the most advanced. Even if you fail to pass the exam, as long as you are willing to continue to use our 312-97 test answers, we will still provide you with the benefits of free updates within a year.

www.thingstogetme.com, www.stes.tyc.edu.tw, Disposable vapes