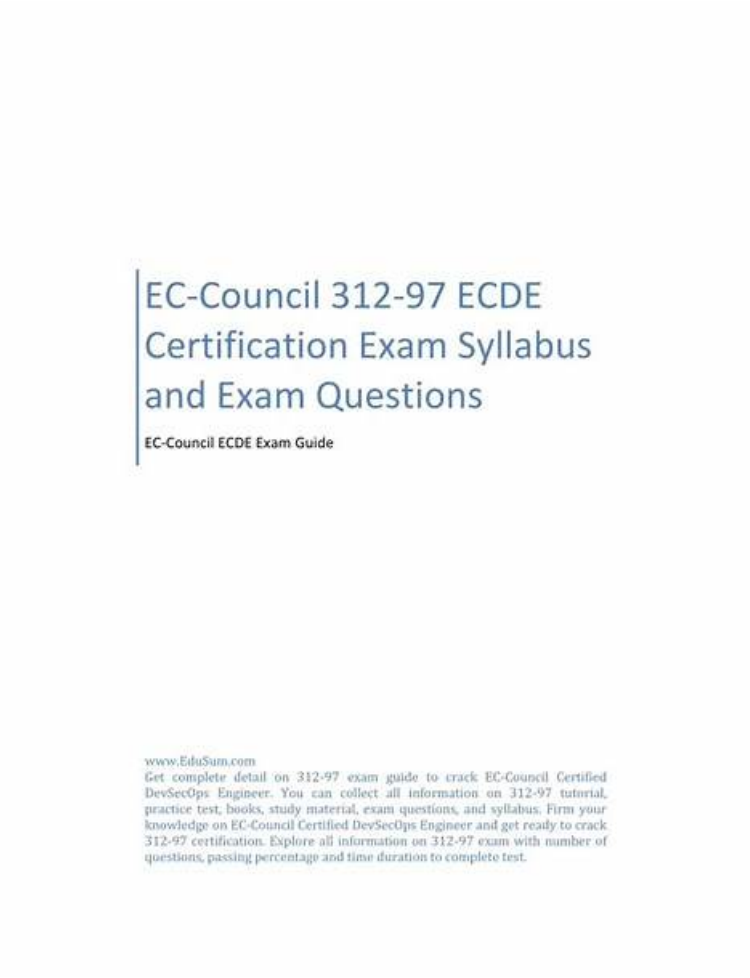
312-97テスト内容 & 312-97ウェブトレーニング



P.S. ShikenPASSがGoogle Driveで共有している無料かつ新しい312-97ダンプ：https://drive.google.com/open?id=10aF2uvRqpMi7g76mnxIibbSQ9ihTCg01

312-97スタディガイドでは、無料の試用サービスを提供しているため、購入前にいくつかのトピックやソフトウェアを開く方法について学ぶことができます。 312-97学習教材の試用期間中、サンプルの質問のPDFバージョンは無料でダウンロードできます。また、PCバージョンとオンラインバージョンの両方を明確に示すことができます。 購入または試用プロセスで312-97試験の質問に問題がある場合は、いつでもご連絡いただけます。ECCouncil 312-97トレーニングガイドで専門家をリモートで支援します。

ECCouncil 312-97 認定試験の出題範囲：

トピック	出題範囲
トピック 1	• 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.
トピック 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 • CD practices, continuous improvement, and the essential communication patterns needed for faster, reliable software delivery.

トピック 3	• 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.
トピック 4	• 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.
トピック 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.
トピック 6	• 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.

ECCouncil EC-Council Certified DevSecOps Engineer (ECDE) 認定 312-97 試験問題 (Q48-Q53):

質問 # 48
(Kenneth Danziger is a certified DevSecOps engineer, and he recently got a job in an IT company that develops software products related to the healthcare industry. To identify security and compliance issues in the source code and quickly fix them before they impact the source code, Kenneth would like to integrate WhiteSource SCA tool with AWS. Therefore, to integrate WhiteSource SCA Tool in AWS CodeBuild for initiating scanning in the code repository, he built a buildspec.yml file to the source code root directory and added the following command to pre-build phase curl -LJOhttps://github.com/whitesource/unified-agent-distribution/raw/master/standAlone/wss_agent.sh. Which of the following script files will the above step download in Kenneth organization's CodeBuild server?.)

- A. aws_agent.sh.
- B. ssw_agent.sh.
- C. wss_agent.sh.
- D. cbs_agent.sh.

正解：C

解説：
The command shown in the pre-build phase explicitly targets a script namedwss_agent.sh. The curl -LJO flags mean: -L follows redirects, -J honors the server-provided filename in the Content-Disposition header (when present), and -O writes output to a local file using the remote name. Since the requested path ends with wss_agent.sh, the downloaded file on the AWS CodeBuild server will be wss_agent.sh. This script is the WhiteSource (now commonly referred to as Mend in many environments) unified agent shell wrapper used to run SCA scans as part of a CI pipeline. Integrating SCA during the Build and Test stage helps detect vulnerable open-source dependencies and licensing/compliance issues early, when fixes are cheapest. The other filenames (ssw_agent.sh, cbs_agent.sh, aws_agent.sh) are distractors; they are not referenced by the provided command and would not be downloaded by that step.
═══════

質問 # 49
(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'.

正解：C

解説：
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.
═══════

質問 # 50
(Brett Ryan has been working as a senior DevSecOps engineer in a multinational company that develops web applications. The team leader of the software development team requested Brett to detect insecure JavaScript libraries in the web application code. Brett would like to perform the vulnerability scanning on web application with grunt-retire. Which of the following commands would enable grunt plugin?)

- A. grunt.loadNpmTask('grunt-retire');.
- B. grunt-loadNpmTasks('grunt-retire');.
- C. grunt-loadNpmTask('grunt-retire');.
- D. grunt.loadNpmTasks('grunt-retire');.

正解：D

解説：
In Grunt, plugins installed via npm must be explicitly loaded in the Gruntfile to make their tasks available.
This is done using the grunt.loadNpmTasks() function, which instructs Grunt to load tasks provided by a specific plugin package.
For the grunt-retire plugin, which scans JavaScript libraries for known vulnerabilities, the correct command is grunt.loadNpmTasks('grunt-retire');. Options that omit the dot notation or use the singular form loadNpmTask are syntactically incorrect and will prevent the plugin from loading.
Enabling grunt-retire during the Code stage allows developers to identify insecure third-party JavaScript libraries early, supporting software composition analysis and reducing the risk of introducing vulnerable dependencies into the application.
═══════

質問 # 51
(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. SAST.

- D. IAST.

正解：B

解説：
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.

質問 # 52
(Andrew Gerrard has recently joined an IT company located in Fairmont, California, as a DevSecOps engineer. Due to robust security and cost-effective service provided by AWS, his organization has migrated all the workloads from on-prem to AWS cloud in January of 2020. Andrew's team leader has asked him to integrate AWS Secret Manager with Jenkins. To do so, Andrew installed the "AWS Secret Manager Credentials provider" plugin in Jenkins and configured an IAM policy in AWS that allows Jenkins to take secrets from AWS Secret manager. Which of the following file should Andrew edit to add access id and secret key parameters along with the region copied from AWS?.)

- A. /etc/sysconfig file/Jenkins.
- B. /etc/filebeat/filebeat.yml.
- C. /etc/file/Jenkins.
- D. /etc/sysconfig/Jenkins.

正解：D

解説：
On Linux systems, Jenkins environment variables such as AWS access key ID, secret access key, and default region are commonly configured in the /etc/sysconfig/Jenkins file. This file allows administrators to define environment variables that are loaded when the Jenkins service starts. By placing AWS credentials and region information in this file, Jenkins jobs and plugins-such as the AWS Secrets Manager Credentials Provider- can securely access AWS resources. The other options reference invalid paths or unrelated configuration files (such as Filebeat). Editing /etc/sysconfig/Jenkins ensures consistent credential availability across Jenkins jobs while supporting secure integration with AWS services during the Code stage.

質問 # 53
.....

- 312-97認定テキスト □312-97出題範囲 ❣ 312-97学習指導 □ ☀ www.goshiken.com □☀□サイトにて ➠ 312-97 □問題集を無料で使おう312-97ソフトウエア
- 信頼できる312-97テスト内容 - 合格スムーズ312-97ウェブトレーニング | 効果的な312-97技術内容 EC-Council Certified DevSecOps Engineer (ECDE) □ □ www.xhs1991.com □の無料ダウンロード【 312-97 】ページが開きます312-97出題範囲
- 検証する312-97テスト内容 - 合格スムーズ312-97ウェブトレーニング | 素晴らしい312-97技術内容 □ 【 312-97 】を無料でダウンロード⇒ www.goshiken.com ⇐ウェブサイトを入力するだけ312-97学習範囲
- 高品質312-97テスト内容 - 資格試験のリーダー - コンプリートECCouncil EC-Council Certified DevSecOps Engineer (ECDE) □ サイト✔ www.passtest.jp □✔□で【 312-97 】問題集をダウンロード312-97学習範囲
- 312-97トレーニング □312-97ウェブトレーニング □312-97トレーニング □時間限定無料で使える□ 312-97 □の試験問題は➤ www.goshiken.com □サイトで検索312-97ソフトウエア
- 312-97関連日本語版問題集 □312-97難易度 □312-97トレーニング □ ▶ www.xhs1991.com ◀を開いて《 312-97 》を検索し、試験資料を無料でダウンロードしてください312-97ブロンズ教材
-

さらに、ShikenPASS 312-97ダンプの一部が現在無料で提供されています：https://drive.google.com/open?id=10aF2uvRqpMi7g76mnxIibbSQ9ihTCg01