Pass Guaranteed Quiz 2026 Juniper Authoritative JN0-224: Automation and DevOps, Associate (JNCIA-DevOps) Test Question

Juniper JN0-224 certification exam is a high demand exam tests in IT field because it proves your ability and professional technology. To get the authoritative certification, you need to overcome the difficulty of JN0-224 Test Questions and complete the actual test perfectly. Our training materials contain the latest exam questions and valid JN0-224 exam answers for the exam preparation, which will ensure you clear exam 100%.

Juniper JN0-224 Exam Syllabus Topics:

Topic	Details
Topic 1	• Python • PyEZ: This domain examines Python programming with PyEZ library for Junos automation, including JSNAPy, Jinja2 templates, RPC calls, exception handling, and device configuration management.
Topic 2	• NETCONF • XML API: This domain focuses on XML syntax, XPath expressions, NETCONF protocol, and XML API functionality for programmatic device configuration and communication.

Topic 3	• Rest API: This domain covers Junos REST API implementation, REST API Explorer tool, and cURL usage for HTTP-based device management and configuration.
Topic 4	• Data Serialization: This domain addresses YAML and JSON formats used for structured data representation and exchange in network automation workflows.
Topic 5	• Junos Automation Stack and DevOps Concepts: This domain covers fundamental automation tools, frameworks, APIs, and DevOps culture applicable to Junos platform operations and network management.

JN0-224 Dumps Torrent | Test JN0-224 Questions Answers

We will provide you with professional advice before you buy our JN0-224 guide materials. If you have problems in the process of using our JN0-224 study questions, as long as you contact us anytime and anywhere, we will provide you with remote assistance until that all the problems on our JN0-224 Exam Braindumps are solved. When you send us a message, we will reply immediately and we will never waste your precious time on studying our JN0-224 practice quiz.

Juniper Automation and DevOps, Associate (JNCIA-DevOps) Sample Questions (Q30-Q35):

NEW QUESTION # 30
Which statement about the NETCONF content layer is true?

- A. It uses XML for RPC request and response payloads.
- B. It uses JSON for RPC request and response payloads.
- C. It uses HTML for RPC request and response payloads.
- D. It uses YAML for RPC request and response payloads.

Answer: A

NEW QUESTION # 31
What is an example of correct XML syntax?

- A. <Devices><hostname>vMX1<De vices/X/hostname>
- B. <Device3><ho3tname>vMX1</hostname><Devices/>
- C. <Devices><hostname>vMX1<Devices/>
- D. <Devices><hostname>vMX1</ho3tnameX/Device3>

Answer: A

NEW QUESTION # 32
You are using the curl tool to include multiple RPCs in a REST API request.
Which HTTP method must be used for this request?

- A. PUT
- B. POST
- C. HEAD
- D. GET

Answer: B

NEW QUESTION # 33
Which DevOps "Three way" principle addresses technical debt?

- A. continuous experimentation and learning
- B. flow
- C. feedback
- D. continuous experimentation

Answer: C

Explanation:
In the context of the DevOps "Three Ways" principles, the feedback principle directly addresses the management of technical debt.
The "Three Ways" are core principles guiding DevOps practices, and they are as follows:
Flow: Refers to the smooth and fast flow of work through the system, from development to operations.
Feedback: Emphasizes creating effective, fast, and continuous feedback loops between teams to catch issues early, address technical debt, and ensure quality.
Continuous experimentation and learning: Encourages constant experimentation, innovation, and learning from failures to improve systems and processes over time.
Feedback and Technical Debt:
Feedback loops play a crucial role in addressing technical debt. Technical debt refers to the implied cost of additional work that arises when code or system design decisions are made for short-term gains, such as quick fixes or temporary patches. Over time, technical debt can accumulate and degrade system performance, reliability, and maintainability.
The feedback loop ensures that issues related to technical debt (such as poor code quality, design shortcuts, or performance bottlenecks) are caught early in the process, ideally before they become major problems. Continuous monitoring, testing, and reviewing help identify and resolve technical debt incrementally rather than letting it accumulate unchecked.
Automation in feedback loops: In DevOps, automated testing, continuous integration (CI), and monitoring tools provide immediate feedback to developers, highlighting areas where technical debt is increasing. This feedback is crucial for making proactive decisions about refactoring code or improving infrastructure without waiting for problems to manifest in production.
For instance, the feedback loop might expose slowdowns in application performance after each new feature is added. This would trigger a review to either refactor the feature code or improve system resources, preventing further technical debt accumulation.
Flow and Technical Debt:
While flow focuses on the smooth transition of work through the pipeline, it indirectly helps with technical debt by ensuring continuous and streamlined processes. However, feedback mechanisms are the primary tools for identifying and resolving technical debt.
Continuous Experimentation and Learning:
This principle promotes innovation and learning from failures but does not directly address technical debt. The focus here is more on risk-taking and improvement rather than managing or eliminating technical debt.
Reference from DevOps Practices:
The Phoenix Project, a book often referenced in DevOps, discusses how feedback loops are essential for maintaining system integrity and managing technical debt effectively. By improving feedback mechanisms, teams can address small issues before they become costly to fix.
The DevOps Handbook also highlights the importance of feedback in managing technical debt, emphasizing that fast feedback allows for continuous improvement and avoids the accumulation of bad practices that would otherwise lead to technical debt.
Juniper Automation and DevOps Context: Juniper's automation frameworks integrate feedback mechanisms using tools like continuous monitoring and automated testing. These tools help engineers track the health of network systems, identify configuration drifts, and resolve issues before they lead to significant technical debt.
Additional Resources:
The Phoenix Project by Gene Kim
The DevOps Handbook

NEW QUESTION # 34
Which two statements about NETCONF layers are correct? (Choose two.)

- A. NETCONF layers use the operations layer to send RPCs to a remote NETCONF server.
- B. NETCONF layers use the messages layer to receive RPCs from a remote NETCONF server.
- C. NETCONF layers use the operations layer to receive RPCs from a remote NETCONF server.
- D. NETCONF layers use the messages layer to send RPCs to a remote NETCONF server.

Answer: C,D

Explanation:
NETCONF (Network Configuration Protocol) is a standard protocol defined for managing network devices. NETCONF operates in a layered architecture, which includes the following key layers:
Operations Layer: This layer deals with the actual operations like <get-config>, <edit-config>, <copy-config>, and others. It receives RPC (Remote Procedure Call) requests from a remote NETCONF client and processes these requests.
Messages Layer: This layer is responsible for encoding the RPCs and sending them over the network. It handles the communication between the NETCONF client and server, ensuring that the RPC messages are correctly formatted (usually in XML) and transmitted.
Statement B is correct because the Messages layer is responsible for sending RPCs to a remote NETCONF server.
Statement C is correct because the Operations layer is where the NETCONF server receives and processes the RPCs sent by the client.
Supporting Reference:
Juniper Networks NETCONF Documentation: Provides a detailed breakdown of the NETCONF protocol layers and their functions.
RFC 6241: The official specification for NETCONF, which describes the layered architecture, including the operations and messages layers.

NEW QUESTION # 35
.....