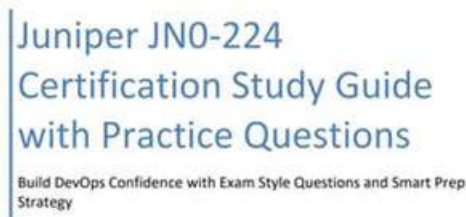


Reliable JN0-224 Test Tutorial - JN0-224 Study Reference



www.NWExam.com
The Juniper JN0-224 JNCIA-DevOps Study Guide is a structured preparation resource designed to help candidates pass the Automation and DevOps Associate certification exam. This guide provides a clear overview of the exam format, scoring structure, registration process, and effective preparation strategies.

Inside this PDF, you will find carefully crafted practice questions that reflect real exam scenarios and strengthen your understanding of automation concepts within Junos environments. Designed for IT professionals pursuing DevOps and network automation skills, this guide supports focused preparation and builds confidence for the JN0-224 certification exam.

2026 Latest Free4Dump JN0-224 PDF Dumps and JN0-224 Exam Engine Free Share: <https://drive.google.com/open?id=1MYnNhCe51AB51uQHURy7NcflmowIBtPb>

Automation and DevOps, Associate (JNCIA-DevOps) exam is one of the top-rated Juniper JN0-224 Exams. This Automation and DevOps, Associate (JNCIA-DevOps) exam offers an industrial-recognized way to validate a candidate's skills and knowledge. Everyone can participate in Automation and DevOps, Associate (JNCIA-DevOps) exam requirements after completing the Automation and DevOps, Associate (JNCIA-DevOps) exam. With the Automation and DevOps, Associate (JNCIA-DevOps) exam you can learn in-demand skills and upgrade your knowledge. You can enhance your salary package and you can get a promotion in your company instantly.

Juniper JN0-224 Exam Syllabus Topics:

Section	Weight	Objectives
Topic 1: Junos Automation Stack and DevOps Concepts	20%	- DevOps principles and culture - Junos automation architecture - Automation frameworks and tools - Configuration management and version control
Topic 2: Junos REST API	15%	- REST API Explorer tool - cURL and HTTP methods - REST API fundamentals
Topic 3: NETCONF and XML API	20%	- XML concepts, syntax and XPath - NETCONF protocol operations and capabilities - XML API usage and interaction

Topic 4: Python and PyEZ	30%	- PyEZ library installation and usage - Error handling and automation scripts - RPC execution and configuration management - Python programming basics - Jinja2 templates and JSNAPy
Topic 5: Data Serialization	15%	- Data modeling and representation - JSON structure and usage - YAML structure and usage

>> **Reliable JN0-224 Test Tutorial** <<

Pass Guaranteed 2026 Juniper JN0-224 –The Best Reliable Test Tutorial

With the advent of knowledge times, we all need some professional certificates such as JN0-224 to prove ourselves in different working or learning condition. So making right decision of choosing useful practice materials is of vital importance. Here we would like to introduce our JN0-224 practice materials for you with our heartfelt sincerity. With passing rate more than 98 percent from exam candidates who chose our JN0-224 study guide, we have full confidence that your JN0-224 actual test will be a piece of cake by them.

Juniper Automation and DevOps, Associate (JNCIA-DevOps) Sample Questions (Q10-Q15):

NEW QUESTION # 10

Which two statements about NETCONF are correct? (Choose two.)

- A. The default port for NETCONF is port 930.
- **B. The default port for NETCONF is port 830.**
- C. NETCONF cannot use the default SSH port.
- **D. NETCONF can use the default SSH port.**

Answer: B,D

Explanation:

NETCONF (Network Configuration Protocol) is used for network device management and can operate over SSH. The following are true about NETCONF:

Default Port 830 (B): By default, NETCONF uses port 830 for communication over SSH. This is the standard port reserved for NETCONF sessions.

Use of Default SSH Port (D): NETCONF can also operate over the standard SSH port (port 22) if configured to do so. This allows flexibility in network management scenarios where port 830 might not be available or used.

Options A and C are incorrect because they refer to incorrect or non-applicable port numbers for NETCONF.

Reference:

IETF RFC 6241: Specifies the use of NETCONF over SSH, including port details.

Juniper Networks NETCONF Documentation: Discusses the configuration and operation of NETCONF on Junos devices, including port usage.

NEW QUESTION # 11

You are asked to write an on-box script that will be triggered when a specific interface on a Junos device goes down.

Which type of on-box script should you use to accomplish this task?

- A. operation
- B. SNMP
- **C. event**
- D. commit

Answer: C

Explanation:

An event script is used to automate responses to system events in Junos, such as an interface going down. These scripts are

triggered automatically when a specified event occurs, making them suitable for tasks like monitoring interface status and executing actions when the status changes.

Option B (event) is correct because event scripts are designed for reacting to system events like an interface going down.

Option A (commit) is used for configuration changes, Option C (operation) is used for operational tasks, and Option D (SNMP) is not applicable in this context.

Supporting Reference:

Juniper Networks Event Scripts Documentation: Details how event scripts are used to automate responses to specific system events in Junos

NEW QUESTION # 12

Exhibit.

```
...
rest {
  http {
    port 3000;
  }
  traceoptions {
    flag all;
  }
  enable-explorer;
}
...
web-management {
  http;
  https {
    system-generated-certificate;
  }
}
...

user@router> show configuration interfaces
ge-0/0/0 {
  unit 0 {
    family inet6 {
      address 2001:db8:0:1::/64;
    }
  }
}
```

You have the configuration shown in the exhibit, but are unable to access the REST API Explorer.

Which configuration is necessary to solve this problem?

- A. Configure an IPv4 address on ge-0/0/0 0.
- B. Configure a firewall filter to explicitly permit SSH.
- C. Configure a secondary IPv6 address on ge-0/0/0.0.
- D. Configure the rest service to be active on port 80.

Answer: C

NEW QUESTION # 13

Which DevOps "Three way" principle addresses technical debt?

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

Answer: A

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 # 14

What are two Junos PyEZ configuration object methods? (Choose two.)

- A. device ()
- B. config ()
- C. commie ()
- D. lockO

Answer: B,D

Explanation:

In Junos PyEZ, the Config object provides various methods for interacting with device configurations. Two of the key methods are: lock(): This method locks the candidate configuration database to prevent other users or processes from making changes while you are modifying the configuration.

config(): This method is used to create a Config object that represents the configuration database, allowing you to load, modify, and commit configuration changes.

Option C (lock) and Option D (config) are correct because they are valid methods provided by the PyEZ Config object.

Option A (commie) and Option B (device) are incorrect as they are not methods of the Config object.

Supporting Reference:

Junos PyEZ Documentation: Details the methods available in the Config object, including lock() and config().

NEW QUESTION # 15

.....

