



JN0-420^{Q&As}

Automation and DevOps, Specialist (JNCIS-DevOps)

Pass Juniper JN0-420 Exam with 100% Guarantee

Free Download Real Questions & Answers **PDF** and **VCE** file from:

<https://www.pass4itsure.com/jn0-420.html>

100% Passing Guarantee
100% Money Back Assurance

Following Questions and Answers are all new published by Juniper
Official Exam Center

- ⚙️ **Instant Download** After Purchase
- ⚙️ **100% Money Back** Guarantee
- ⚙️ **365 Days** Free Update
- ⚙️ **800,000+** Satisfied Customers





QUESTION 1

What are three supported languages for operational scripts on Junos devices? (Choose three)

- A. Python
- B. XSLT
- C. SLAX
- D. Java
- E. PHP

Correct Answer: ABC

Junos automation scripts can be written in the following scripting languages:

Extensible Stylesheet Language Transformations (XSLT)

Stylesheet Language Alternative syntaX (SLAX)

Python

Reference: https://www.juniper.net/documentation/en_US/junos/topics/concept/junos-script-automationoverview.html

QUESTION 2

To which component(s) does the JET service APIs provide access?

- A. YANG modules
- B. XML APIs
- C. control plane
- D. forwarding plane

Correct Answer: C

Reference: https://www.juniper.net/documentation/en_US/jet1.0/topics/concept/jet-service-apis-overview.html

QUESTION 3

Which data serialization method does the Junos CLI use to communicate with the Junos management process (mgd)?

- A. JSON
- B. gRPC
- C. XML



D. YAML

Correct Answer: C

QUESTION 4

Click the Exhibit button. Exhibit:

Python Script:

```
1 from jnpr.junos import Device
2 from jnpr.junos.utils.config import Config
3 from jnpr.junos.exception import *
4 from jinja2 import Template
5 import yaml
6 import sys
7
8
9     with open ('vMX-1.yml','r') as fh:
10         data = yaml.load (fh.read())
11
12     with open ('test.j2','r') as t_fh:
13         t_format = t_fh.read()
14
15     template = Template (t_format)
16
17     myConfig = template.render (data)
```

vMX-1.yml file:

```
- - -
intf1:
  name: 'ge-0/0/0'
  unit: 0
  family: inet
  address: 172.17.1.1
  cidr: 24
  bfd_intvl: 500
intf2:
  name: 'ge-0/0/1'
  unit: 0
  family: inet
  address: 172.17.3.1
```



```
    cidr:24
    bfd_intvl: 500
  intf3:
    name: 'lo0'
    unit: 0
    family: inet
    address: 192.168.1.1
    cidr: 32
```

```
AS: 65000
area: 0.0.0.0
```

test.j2 file:

```
interfaces {
  {{intf1.name}} {
    unit {{intf1.unit}} {
      family {{intf1.family}} {
        address {{intf1.address}} / {{intf1.cidr}};
      }
    }
  }
  {{intf2.name}} {
    unit {{intf2.unit}} {
      family {{intf2.family}} {
        address {{intf2.address}} / {{intf2.cidr}};
      }
    }
  }
  {{intf3.name}} {
    unit {{intf3.unit}} {
      family {{intf3.family}} {
        address {{intf3.address}} / {{intf3.cidr}};
      }
    }
  }
}
```



```
    }
  }
}
routing options {
  static {
    route 0.0.0.0/0 next-hop 172.25.11.254;
  }
  autonomous-system {{AS}};
}
protocols {
  ospf {
    area {{area}} {
      interface fxp0.0 {
        disable;
      }
      interface {{intf1.name}}. {{intf1.unit}} {
        bfd-liveness-detection {
          minimum-interval {{intf1.bfd_intvl}};
        }
      }
      interface {{intf2.name}}. {{intf2.unit}} {
        bfd-liveness-detection {
          minimum-interval {{intf2.bfd_intvl}};
        }
      }
      interface {{intf3.name}}. {{intf3.unit}};
    }
  }
}
```

Referring to the exhibit, what is the type of the myConfig variable on line 17 of the Python script?

- A. template
- B. dictionary
- C. string
- D. list

Correct Answer: A

Reference: https://www.juniper.net/documentation/en_US/cso3.1/topics/task/operational/cd-configtemplate-working-config-designing.html



QUESTION 5

Using the RESTful API on a Junos device, you are able to retrieve configuration information in which three formats? (Choose three)

- A. YAML
- B. XML
- C. plain text
- D. GPB E. JSON

Correct Answer: BCE

Reference: https://www.juniper.net/documentation/en_US/junos/information-products/pathway-pages/restapi/rest-api.html

QUESTION 6

Click the Exhibit button. Exhibit:



```
from jnpr.junos import Device
from jnpr.junos.utils.config import Config
from jnpr.junos.exception import *
import sys

def main ():

    interface = jcs.get_input ("Enter interface to change: ")
    if not interface:
        print "invalid interface"
        sys.exit(1)

    config_xml = """
        <configuration>
            <interfaces>
                <interface>
                    <name> {0}</name>
                    <disable/>
                </interface>
            </interfaces>
        </configuration>
    """.format (interface).strip()

    dev = Device()
    dev.open()
    dev.bind( cu=Config )

    # Lock the configuration, load configuration changes, and commit
    print "Locking the configuration"
    try:
        dev.cu.lock()
    except LockError:
        print "Error: Unable to lock configuration"
        dev.close()
        return

    print "Loading configuration changes"
    try:
        dev.cu.load (config_xml, format="xml", merge =True)
    except ConfigLoadError as err:
        print err
        print "Unable to load configuration changes: "
        print "Unlocking the configuration"
```



```
        try:
            dev.cu.unlock()
        except UnlockError:
            print "Error: Unable to unlock configuration"
        dev.close()
        return

    print "Committing the configuration"
    try:
        dev.cu.commit()
    except CommitError:
        print "Error: Unable to commit configuration"
        print "Unlocking the configuration"
        try:
            dev.cu.unlock()
        except UnlockError:
            print "Error: Unable to unlock configuration"
        dev.close()
        return

    print "Unlocking the configuration"
    try:
        dev.cu.unlock()
    except UnlockError:
        print "Error: Unable to unlock configuration"

    dev.close()
if __name__ == "__main__":
    main()
```

Referring to the exhibit, what are two results of the Python operational script? (Choose two)

- A. It disables a user-selected interface
- B. It commits the configuration
- C. It deactivates a user-selected interface
- D. It does not commit the configuration

Correct Answer: AB

QUESTION 7

Click the Exhibit button. Exhibit:



```
[root@ansible-cm] # cat variables.yml
$ANSIBLE_VAULT; 1.1; AES256
3141596134396662303537353231326433363366376435376339306664313130656563646
33266
34
3730326165666565356665343137313161234569336336640a6539393270697606666
6565373765326236323535326162613531234566366539626237633962373736623865343
63066
63
6430376633306339360a34306536333131353263303634386637633062363465353835313
23141
59
3835
```

Your manager asks you to troubleshoot an Ansible problem when logging into a Junos device. You notice the Ansible play in question has a `vars_files` argument that calls the `variables.yml` entry file. The contents of that file are displayed in the exhibit.

What is happening in this scenario?

- A. The file is an SSH key
- B. The `variables.yml` file is not actually a YAML file
- C. The file is encrypted with `ansible-vault`
- D. The file is encrypted with `ansible`

Correct Answer: C

Reference: https://www.juniper.net/documentation/en_US/junos-ansible1.0/topics/task/configuration/junosansible-authenticating-users.html

QUESTION 8

Click the Exhibit button. Exhibit:



```
-- -- --
-   name: Get facts from Junos devices
    hosts: all
    connection: local
    gather_facts: no
    roles:

-   Juniper.junos

tasks:

-   name: Get facts from Junos devices

        junos_get_facts:
            host: "{{inventory_hostname}}"
            user: "user"
            passwd: "user123"
            register: response

-   name: Print Junos version
    debug:
        var: response.facts.version
```

The Ansible playbook shown in the exhibit is executed against a set of Junos network devices. Each Junos device is configured with a user account in the super-user login class. The user account uses SSH-keybased authentication with a passphrase of user123.

What is the result of executing this playbook against the Junos network devices?

- A. The playbook executes, but fails with a "ConnectAuthError" due to the missing passphrase argument to the junos_get_facts module
- B. The playbook fails due to a YAML syntax error.
- C. The playbook executes and prints the version of Junos running on each network device as the value of response.facts.version
- D. The playbook executes and prints "VARIABLE IS NOT DEFINED!" as the value response.facts.version

Correct Answer: C

QUESTION 9

Click the Exhibit button.

Exhibit:

Junos Event Policy Configuration



```
[edit]
user@router# show event-options
policy int-down-1 {
    events snmp_trap_link_down;
    attributes-match {
        snmp_trap_link_down.interface-name matches "(^ge-0/0/0$)";
    }
    then {
        change-configuration {
            commands {
                "set system host-name int-test-1";
            }
            user-name user;
            commit-options {
                log "int down test policy 1";
            }
        }
    }
}
policy int-down-2 {
    events snmp_trap_link_down;
    attributes-match {
        snmp_trap_link_down.interface-name matches "(^ge-0/0/0$)";
    }
    then {
        ignore;
    }
}
policy int-down-3 {
    events snmp_trap_link_down;
    attributes-match {
        snmp_trap_link_down.interface-name matches "(^ge-0/0/0$)";
    }
    then {
        change-configuration {
```



```
        commands {
            "set system host-name int-test-3";
        }
        user-name user;
        commit-options {
            log "int down test policy 3";
        }
    }
}

policy int-down-4 {
    events snmp_trap_link_down;
    attributes-match {
        snmp_trap_link_down.interface-name matches "^ge-0/0/0$";
    }
    then {
        event-script hostname.slax;
    }
}
```

hostname.slax event script

version 1.1;

ns junos = "http://xml.juniper.net/junos/*/junos";

ns xnm = "http://xml.juniper.net/xnm/1.1/xnm";

ns jcs = "http://xml.juniper.net/junos/commit-scripts/1.0";

import "../import/junos.xsl";

match / {

```
<event-script-results> {
    var $hostname-cfg-change = <configuration> {
```



```
<system> {  
    <host-name> "int-test-4";  
}  
  
}  
  
var $connection = jcs:open();  
var $results := { call jcs: load-configuration ( $connection,  
$configuration = $hostname-cfg-change );}  
if ( $results//xnm:error ) {  
    for-each ( $results//xnm:error ) {  
        expr jcs:syslog( "daemon.error", "Event script error: ",  
message );  
    }  
}  
expr jcs:close($connection);  
}  
}
```

Referring to the exhibit. What will the hostname be if interface ge-0/0/0 goes down?

- A. int-test-2
- B. int-test-1
- C. int-test-4
- D. int-test-3

Correct Answer: B

QUESTION 10

With which three processes does the JET service process communicate? (Choose three)

- A. snmpd
- B. ancpd
- C. cosd
- D. mgd
- E. rpd

Correct Answer: ADE

QUESTION 11



By default, Python event scripts and SNMP scripts configured on a Junos device execute as which user?

- A. wheel
- B. snmpd
- C. nobody
- D. root

Correct Answer: C

QUESTION 12

Click the Exhibit button. Exhibit:



```
{
  "configuration" : [
    {
      "attributes" : {"junos:changed-seconds" : "1327069760",
        "junos:changed-localtime" : "2017-08-28
11:17:55 UTC"},
      "system" : [
        {
          "services" : [
            {
              "ssh" : [
                {
                  "root-login" : [
                    {
                      "data" : "allow"
                    }
                  ],
                  "max-sessions-per-connection" : [
                    {
                      "data" : "32"
                    }
                  ]
                }
              ]
            }
          ]
        }
      ]
    }
  ]
}
```

Referring to the exhibit, the configuration components are displayed in which format?

- A. XML
- B. JSON
- C. YAML
- D. YANG

Correct Answer: B

QUESTION 13



Which format does an Ansible playbook use?

- A. JSON
- B. YAML
- C. XML
- D. CSV

Correct Answer: B

Reference: https://www.juniper.net/documentation/en_US/junos-ansible1.0/topics/task/program/junosansible-playbooks-creating-executing.html

QUESTION 14

Click the Exhibit button. Exhibit:

```
from jnpr.junos.device import Device

dev = Device (host='router', user='user', passwd='user123', port=22)
dev.open ()
response = dev.rpc.get_interface_information (format="text", terse="yes")
print (response)
dev.close ()
```

Referring to the exhibit, which exception is raised if the RPC takes 40 seconds to return a complete response?

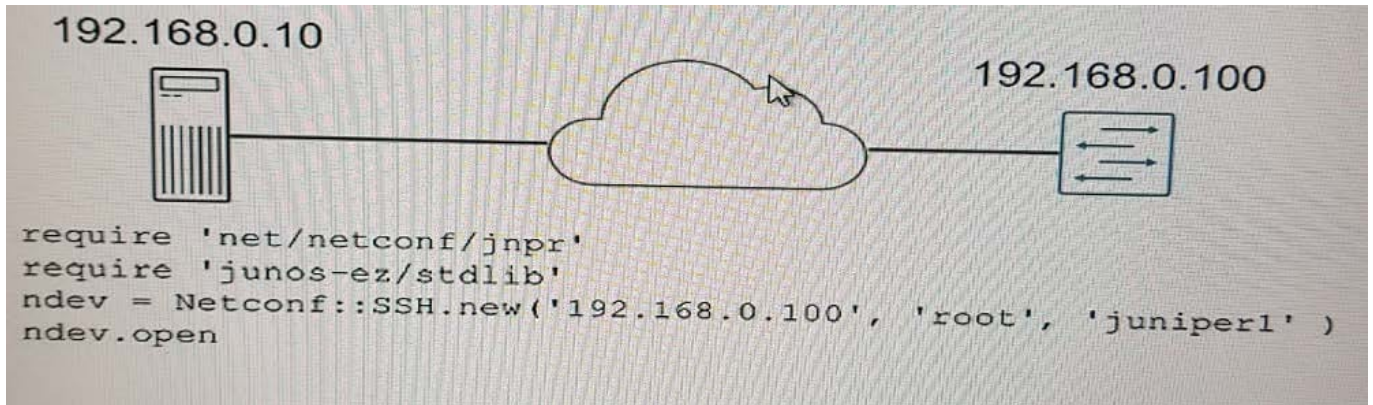
- A. `jnpr.junos.exception.RpcError`
- B. `jnpr.junos.exception.ConnectTimeoutError`
- C. `jnpr.junos.exception.RpcTimeoutError`
- D. `jnpr.junos.exception.LockError`

Correct Answer: C

Explanation: By default, the timeout value is 30 seconds, if this is exceeded a `jnpr.junos.exception.RpcTimeoutError` error is received.

QUESTION 15

Click the Exhibit button. Exhibit:



You are using RubyEZ to interact with a Junos device: however, you are not successfully connecting to the device.

Referring to the exhibit, what is the problem?

- A. Argument passed to Netconf::SSH.new statement must be referenced as variables
- B. Netconf::SSH.new statement only expects an IP address of the target device
- C. Netconf::SSH.new statement arguments must be hashes
- D. A Junos::Ez::Provider statement is missing before the Netconf::SSH. New statement

Correct Answer: A

[Latest JN0-420 Dumps](#)

[JN0-420 VCE Dumps](#)

[JN0-420 Study Guide](#)