ATMIC NETWORKS
YOUR GUIDE TO GLOBAL CERTIFIATION

"Please note that these files may not be up to date. However, the questions will help you understand the exam format and typical question patterns."

www.atmicnetworks .com

Warning: Keep connected with our support team for latest updates

Question: 1

What is the primary reason for creating an Apstra worker node?

A. To support more than one blueprint
B. To create a space for storing event logs
C. To run Zero Touch Provisioning (ZTP)
D. To ofload off-box agents and Intent-Based Analytics (IBA)

Answer: D

Explanation:

In Apstra 5.1, the worker node's primary purpose is to add scalable runtime capacity to an Apstra cluster by hosting off-box services that would otherwise consume resources on the controller. Specifically, worker nodes run containerized services such as off-box device agents (used to communicate with and manage devices) and Intent-Based Analytics (IBA) components (such as probes and analytics-related services). This design keeps the controller node focused on cluster management and control-plane functions (API handling, cluster-wide state, blueprint control workflows), while shifting resource-intensive operational services to worker nodes.
As your fabric grows—more switches, more telemetry, more devices requiring agent connectivity— CPU and memory demand increases notably, especially when IBA is enabled. Adding worker nodes allows you to scale those container workloads horizontally without redesigning the fabric or reducing analytics coverage. In a Juniper data center built on EVPN-VXLAN with Junos v24.4 leaf-spine roles, this separation helps ensure that Apstra can continuously validate intent, process streaming telemetry, and maintain device communications reliably at scale. Worker nodes therefore exist primarily to ofload and scale operational agents and IBA services, improving performance and resilience for larger deployments.

Question: 2

Which Root Cause Identifier is currently supported in Juniper Apstra software?

A. Virtual network
B. Connectivity
C. ESI imbalance
D. BGP

Answer: B

Explanation:

In Juniper Apstra 5.1, Root Cause Identification (RCI) is implemented with a currently supported model focused on connectivity. Practically, this means RCI is designed to take telemetry and state learned from the fabric (for example, interface operational status, LLDP neighbor information, and routing session status) and correlate those signals to determine the most likely underlying cause of a connectivity-impacting event. Within an EVPN-VXLAN IP fabric, many operational symptoms can appear similar at the service layer (endpoints cannot reach each other, routes disappear, overlays degrade), but RCI narrows the problem by correlating evidence across the underlay and control plane.

The "connectivity" RCI model targets common failure scenarios that directly break device-to-device reachability, such as a broken link, a miscabled link (wrong LLDP neighbors), or an operator-disabled interface. These conditions often cascade into higher-level symptoms, including BGP sessions dropping over affected links. With Junos v24.4-based leaf-spine fabrics, maintaining stable underlay connectivity is foundational for EVPN signaling and VXLAN forwarding; therefore, Apstra's connectivity-focused RCI helps operators rapidly isolate whether the primary fault lies in physical

adjacency, cabling/neighbor correctness, or administrative shutdown—reducing mean time to repair by pointing to the most probable root cause rather than only listing alarms.

Question: 3

You are attempting to attach the server connected to the my_border_001_leaf1 node's ge-0/0/5 interface to the finance-app virtual network.

Referring to the exhibit, what would you do to solve the problem?

A. You can add a generic system to the physical topology.
B. You can set the generic system to the deploy mode.
C. You can allocate an IP pool resource to the virtual network.
D. You can assign the finance-app virtual network to the my_border_001_leaf1 node.

Answer: A

Explanation:

In Apstra 5.1, servers are modeled as Generic Systems and must be represented in the blueprint topology so that Apstra can bind an endpoint (the server) to a specific switch interface and then apply the intended connectivity template / virtual network attachment. In the exhibit, the interface ge-0/0/5 on my_border_001_leaf1 is shown as missing from the assignment workflow, which indicates that Apstra does not currently have an endpoint object connected to that port in the blueprint's staged physical topology (or that the

port is not presented as an eligible connectivity point for server attachment).

The correct remediation is to add a Generic System connected to my_border_001_leaf1 ge-0/0/5 in Staged > Physical > Topology, thereby creating a modeled server link on that interface. Once the Generic System exists and the interface is a recognized server-facing connectivity point, you can assign the Tagged VXLAN “finance-app” connectivity template (or the VN assignment action driven by that template) to the server-facing interface and then commit the staged changes.

Changing “deploy mode” may affect whether Apstra actively configures a generic system-facing link, but it does not solve a missing interface in the topology model. Likewise, allocating an IP pool is unrelated to making the port available for attachment, and assigning the VN to the switch node is not how server interfaces are attached in this workflow.

Verified Juniper sources (URLs):

https://www.juniper.net/documentation/us/en/software/apstra5.1/apstra-user-guide/topics/topic- map/internal-generic-system-create.html

https://www.juniper.net/documentation/us/en/software/apstra5.0/apstra-user-guide/topics/topic- map/virtual-network-assignment-update.html

https://cloudlabs.apstra.com/labguide/Cloudlabs/6.0.0/test-drive-guide/lab1-junos-11_adding- gs.html

Question: 4

You are using Juniper Apstra to create your DC fabric. The fabric requires the use of configlets and requires a property set, which you call “test.” While creating the property set, you encounter an error message.

Referring to the exhibit, how would you correct the error?

A. Use the Builder option for input type of YAML.

B. Remove the trailing blank lines.

C. Change to JSON and click Create.

D. Use valid YAML syntax of key: value.

Answer: D

Explanation:

In Apstra 5.1, a property set is a structured data object used to parameterize configlets (config templates). The key point is that Apstra expects the property set "values" to be a dictionary/map so that the configlet can reference variables by name (for example, {{ NTP_SRV1 }} or nested keys). The exhibit shows a server-side validation error indicating that values_yaml "should be dict," which occurs when the YAML content is entered as a single scalar string (such as try_ksh) instead of a keyvalue mapping.

To correct this, rewrite the YAML using valid key: value syntax so the top-level structure is a dictionary. For example, a minimal valid property set would look like role: try_ksh (or any meaningful key name aligned to the variables your configlet expects). If multiple variables are needed, add additional keys, and if your configlet uses nested objects, represent them as nested YAML dictionaries. This correction aligns the property set with Apstra's intent-based model: values are stored as named properties and then rendered deterministically into device configuration. This is independent of Junos v24.4 specifics; Junos becomes relevant when the rendered configlet content is applied to devices, but the property set itself must first validate as a dictionary for Apstra to render the template correctly.

Verified Juniper sources (URLs):

https://www.juniper.net/documentation/us/en/software/apstra5.1/apstra-user- guide/topics/task/property-set-datacenter-design-create.html

https://www.juniper.net/documentation/us/en/software/apstra5.1/apstra-user- guide/topics/concept/property-set-datacenter-design.html

https://www.juniper.net/documentation/us/en/software/apstra5.1/apstra-user- guide/topics/ref/property-sets-

api.html

Question: 5

You want to route between tenants in a multitenant environment in Juniper Apstr

a. What are two ways to accomplish this task? (Choose two.)

A. Route between VRFs on a VTEP-enabled device.

B. Use an external device to route between tenants.

C. Use iBGP to route within the same AS number.

D. Use virtual networks to route between VRFs.

Answer: A, B

Explanation:

In Apstra 5.1 multitenancy, tenants are modeled as routing zones, and each routing zone maps to a distinct VRF to provide strict Layer 3 isolation. Because each tenant's VRF is separate, "routing between tenants" is effectively inter-VRF routing. Apstra's routing-zone behavior emphasizes that inter-tenant routing is achieved via external systems: you connect each tenant/routing zone to an external router or firewall (often attached to border leafs), and that external device performs the policy-controlled inter-VRF routing between tenants. This approach is the most common because it centralizes security and compliance controls (stateful inspection, zone policies, NAT, logging) on the firewall/router while keeping the fabric clean and consistent.

A second method is to perform inter-VRF routing on a VTEP-capable border leaf that terminates the tenant VRFs. In EVPN-VXLAN designs, border leafs are frequently the demarcation where tenant VRFs connect to outside domains; when the same border leaf hosts multiple tenant VRFs and is designed to provide L3 services for them, it can act as the routing point between VRFs (subject to your design and security requirements). Junos v24.4 supports VRFs and policy constructs required for controlled route exchange and forwarding behavior, but Apstra's intent model still expects routing-zone isolation by default—so any inter-tenant connectivity should be explicitly designed and governed, typically at the border.

Question: 6

Which element of an intent-based analytics (IBA) probe is used to specify the database objects to which the probe will apply?

A. Reference design schema

B. Graph query

C. Database nodes

D. Node-edge relationship

Answer: B

Explanation:

In Apstra 5.1, Intent-Based Analytics (IBA) is built on Apstra's graph-based source of truth, where devices, interfaces, links, routing constructs, and services are represented as nodes with relationships. An IBA probe is effectively a processing pipeline (a directed acyclic graph of stages and processors) that ingests telemetry and then performs calculations, aggregations, and anomaly detection. To make any of that work, the probe must first determine which specific objects in the graph—for example, which leaf switches, which uplinks, which BGP sessions, or which interface counters—should be included in the analysis.

The probe element that selects those objects is the graph query. A graph query is evaluated against Apstra's graph database to return a set of matching nodes/relationships; those query results then become the scope for ingestion and subsequent processing. In other words, the graph query defines "apply this probe to these devices/interfaces/sessions," and it also provides the context used to bind telemetry identities (key-value pairs describing the metric source) to the correct logical objects in the blueprint. This is why Apstra documentation describes early probe processors producing outputs whose cardinality aligns with the number of results returned by the specified graph query(s). Without a graph query, the probe would not have a deterministic, intent-aligned target set for analytics, and the same probe definition could not be reliably reused across fabrics or blueprints.

Question: 7

What are two types of policies that Juniper Apstra uses to push to switches using Security Policies? (Choose two.)

A. Filter-based forwarding (ChooseFBF)

B. Firewall filters

C. Policy-based routing (ChoosePBR)

D. Access control lists (ChooseACLs)

Answer: B, D

Explanation:

Apstra 5.1 Security Policies are intended to enforce permit/deny controls for traffic between defined endpoints such as routing zones, virtual networks, and IP endpoints. Apstra expresses this security intent in an implementation-independent way, then renders and deploys the equivalent enforcement configuration onto the appropriate devices and interfaces. In Apstra terminology, the outcome is an ACL applied at enforcement points, such as virtual network interfaces (SVIs/IRBs) for east-west controls and border leaf interfaces for external-to-internal controls.

Therefore, the two correct policy types in this context are access control lists (ACLs) and firewall filters. "ACL" is the abstract policy object Apstra compiles and applies, while on Junos v24.4 the concrete enforcement mechanism for stateless packet filtering on interfaces is typically implemented as a firewall filter. Apstra automatically places these rendered ACLs/filters where needed: when you add VXLAN endpoints (such as expanding a rack/leaf in a VN), the ACL is placed on the corresponding VN interface; when you add external connectivity points, relevant ACLs are placed on the border leaf enforcement points. This automation ensures that security intent remains consistent as the fabric scales or changes, reducing the risk of manual rule drift. In contrast, filter-based forwarding / policybased routing changes forwarding decisions rather than expressing permit/deny security intent, and is not the primary mechanism used by Apstra Security Policies for reachability control.

Question: 8

In Juniper Apstra, which statement about resources is correct?

A. User-defined resources are supported.

B. The scope of a pool is limited to a single blueprint.

C. The scope of a pool can be defined as global or blueprint specific.

D. Only the default resources are supported globally.

Answer: C

Explanation:

In Apstra 5.1, “resources” are the identifier values consumed by the fabric design and rendered into device configuration—examples include ASNs, IP addresses, VNIs, VLAN-related identifiers (where applicable), and similar allocation-driven values. These values are provided through resource pools, which are the authoritative containers Apstra draws from when assigning resources to blueprint roles (for example, leaf ASNs, spine ASNs, loopbacks, point-to-point subnets, and VNI ranges). A key architectural feature is that resource pools are not confined to one blueprint. Apstra supports pools with different scopes to match operational needs: some pools are managed centrally and reused across multiple blueprints, while other pools are created and used within the context of a specific blueprint when you want strict separation and lifecycle alignment with that blueprint.

This is why the correct statement is that a pool’s scope can be global or blueprint-specific. Global pools are appropriate when you want consistent allocation policy across fabrics (for example, enterprise-wide ASN ranges). Blueprint-specific pools are appropriate when you want per-fabric independence or when allocations are generated dynamically within the blueprint. This scope behavior is independent of Junos v24.4; Junos receives the final rendered values, but the pool scoping and allocation control are Apstra design-time constructs that ensure deterministic, conflict- free assignments at scale.

Verified Juniper sources (URLs):

https://www.juniper.net/documentation/us/en/software/apstra5.1/apstra-user-guide/topics/concept/resources.html

https://www.juniper.net/documentation/us/en/software/apstra5.1/apstra-user- guide/topics/concept/freeform-resource-management.html

https://www.juniper.net/documentation/us/en/software/apstra5.1/apstra-user- guide/topics/ref/resource-pools-api.html

Question: 9

In the Juniper Apstra UI, what are two aspects that you are able to query under the Active tab within a blueprint? (Choose two.)

A. ARP

B. Virtual Network

C. Routing Zone

D. MAC

Answer: A, D

Explanation:

In Apstra 5.1, the Active view represents the operational state of the deployed fabric (as opposed to the intended state being edited in Staged). Within Active, the Query function is designed for day-2 operations where an operator needs to quickly locate endpoint-related information and validate forwarding/neighbor state derived from the fabric. The query choices exposed in the UI are focused on operational lookup primitives rather than design objects. Specifically, Apstra supports querying MAC and ARP (and also VMs when virtual infrastructure integration is present).

MAC queries help identify where a Layer 2 endpoint is being learned in the fabric—useful for troubleshooting EVPN-VXLAN fabrics where MAC learning and advertisement can determine reachability and mobility behavior. ARP queries help identify IP-to-MAC bindings and validate whether hosts are being resolved correctly, which is critical when troubleshooting first-hop behavior (for example, IRB gateway adjacency, endpoint onboarding, or unexpected IP conflicts).

By contrast, "Virtual Network" and "Routing Zone" (VRF) are primarily design constructs managed in Staged and validated/assured by analytics and intent checks; they are not the direct query selectors in the Active > Query tool. Therefore, the two correct Active-query aspects from the given options are ARP and MAC.

Verified Juniper sources (URLs):

https://www.juniper.net/documentation/us/en/software/apstra5.1/apstra-user- guide/topics/task/query-active.html

Question: 10

Which type of generic system should you select when adding a new server inside an existing rack type?

A. Internal generic

B. Rack generic

C. External generic

D. Embedded generic

Answer: A

Explanation:

In Apstra 5.1, servers that connect to leaf switches are represented as generic systems so Apstra can model links, apply connectivity templates, attach virtual networks, and validate intent. The selection of generic system type depends on whether the endpoint is considered part of the rack's internal topology or an external attachment. When you add a new server inside an existing rack type, that server is treated as a component of the rack topology (that is, it lives "within" the rack alongside leaf switches and any other rack-internal endpoints). Apstra documentation refers to such systems as internal generic systems.

Internal generic systems are not managed like switches (no full device management), but they are first-class topology objects: they occupy ports on leaf switches, can be tagged with roles, and can be associated with link definitions that drive correct interface intent (LAG vs single link, VLAN tagging, and virtual network association). This modeling is essential in EVPN-VXLAN fabrics because correct endpoint attachment on leaf ports determines VLAN/VNI mapping and the resulting Junos v24.4 configuration rendered by Apstra.

External generic systems, by contrast, represent devices outside the rack topology (often used for external routers, firewalls, or other non-rack-contained endpoints). Because the question explicitly places the server inside an existing rack type, the correct choice is Internal generic.

Verified Juniper sources (URLs):

https://www.juniper.net/documentation/us/en/software/apstra5.1/apstra-user-guide/topics/topic- map/internal-generic-system-create.html

Question: 11

You want to assign resources to your blueprint during the deployment phase. In this scenario, which statement is correct?

A. To assign resources in the blueprint, you must have completed the device profile and device assignments.

B. To assign resources in the blueprint, you must have already created them under global resources.

C. All resources are created and assigned under the blueprint's Resources tab.

D. All resources are automatically assigned values from the available resource pools.

Answer: D

Explanation:

In Apstra 5.1, “resources” (such as ASNs, IP addressing, and VNIs) are allocated to blueprint elements using resource pools. The blueprint does not require you to manually craft every individual resource value; instead, Apstra’s workflow is to have you indicate which pool(s) should be used for the blueprint, and then Apstra automatically pulls and assigns the required values. This automation is fundamental to Apstra’s intent-based model: once the blueprint knows which pools to consume, it can deterministically allocate unique values across the fabric and generate consistent Junos configuration for the assigned devices.

Option D best matches this behavior because it reflects the documented mechanism: required resources are automatically pulled from the selected pool(s) and assigned in a fast, bulk transaction. This is what enables repeatable deployments—especially in EVPN-VXLAN data center fabrics— because resource collisions and manual tracking are avoided.

Option A is not the defining prerequisite for resource assignment; device profile and device assignment are important overall build steps, but the correctness of resource assignment is tied to pool selection and availability rather than being strictly gated by those tasks. Option B is incorrect because pools can be created and managed beyond only “global” contexts, and Apstra also supports creating additional pools from within the blueprint when needed. Option C is misleading because resources are governed by pools and allocation, not only by manual creation under a single tab.

Verified Juniper sources (URLs):

https://www.juniper.net/documentation/us/en/software/apstra5.1/apstra-user-guide/topics/concept/resources.html

https://www.juniper.net/documentation/us/en/software/apstra5.1/apstra-user- guide/topics/concept/freeform-resource-management.html

https://www.juniper.net/documentation/us/en/software/apstra5.1/apstra-user- guide/topics/ref/resource-pools-api.html

Question: 12

In Juniper Apstra terminology, to which network operating system concept does a routing zone refer?

A. IRB

B. VRF

C. VLAN

D. Access list

Answer: B

Explanation:

In Apstra 5.1, a routing zone is the primary construct used to represent an L3 domain for multitenant isolation. In traditional network operating system terms, that maps to a VRF (Virtual Routing and Forwarding instance). Each routing zone is placed "in its own VRF," which provides independent routing tables and isolates IP traffic so that different tenants can reuse overlapping IP subnets without conflict. This is central to modern EVPN-VXLAN data center design, where tenants typically require clean separation of routing and policy boundaries.

Within a routing zone, you can create one or more virtual networks (often mapped to VXLAN segments) that provide L2 extension across racks while still being contained by the tenant's VRF. If L3 gateway services are enabled for those virtual networks, their gateway interfaces (for example, IRB interfaces on Junos v24.4 leaf switches) are associated with the routing zone's VRF so that intersubnet routing occurs within the tenant boundary.

This terminology distinction is important: an IRB is an interface construct used to provide L3 gateway functionality for a VLAN/VXLAN segment; a VLAN is a Layer 2 segmentation mechanism; and an access list is a policy enforcement tool. A routing zone, however, defines the tenant's L3 routing context, which is precisely what a VRF provides on Junos.

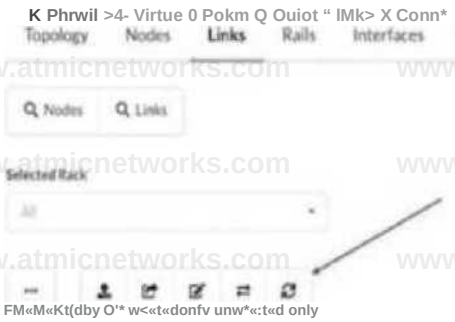
Verified Juniper sources (URLs):

https://www.juniper.net/documentation/us/en/software/apstra5.0/apstra-user- guide/topics/concept/routing-zones.html

https://www.juniper.net/documentation/us/en/software/apstra4.2/apstra-user- guide/topics/concept/routing-zones.html

Question: 13

What does clicking the indicated icon shown in the exhibit accomplish?



A. It refreshes the screen.
B. It fetches the discovered Link Layer Discovery Protocol (LLDP) data.
C. It erases the entire cable map to start over.
D. It changes the speed of existing links.

Answer: B

Explanation:

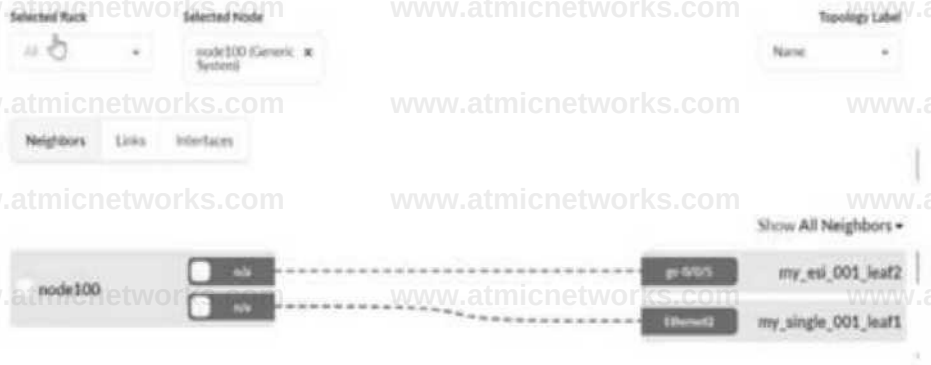
In Apstra 5.1, the Staged > Physical > Links workspace is where you build and validate the cabling (link) intent for the fabric before committing changes. During deployment and day-0/1 build, Apstra can leverage LLDP neighbor discovery from the connected devices to accelerate and validate the cabling map. The indicated toolbar icon in the Links view is used to fetch discovered LLDP data from the devices so Apstra can compare the discovered neighbor relationships with the intended topology and, depending on workflow, help populate or validate link endpoints.

This is particularly important in leaf-spine IP fabrics because correct physical connectivity underpins the entire underlay—interface states, point-to-point addressing, and BGP sessions. In an EVPN- VXLAN design running Junos v24.4, broken or mis-cabled links quickly manifest as missing underlay adjacencies and failed EVPN control-plane signaling. Pulling LLDP discovery into Apstra helps you identify mismatches early (wrong neighbor, wrong port, missing neighbor) and reduces manual cabling errors.

This action is not merely a UI refresh, it does not wipe the cable map, and it does not modify link speeds. Its operational purpose is to import discovered LLDP neighbor information into the blueprint's physical link view so Apstra can assist with accurate topology validation and deployment readiness.

Question: 14

What is correct about the selected device shown in the exhibit?



A. It is a peer switch.
B. It is an external generic system.
C. It is an internal generic system.
D. It is an access switch.

Answer: C

Explanation:

The exhibit shows node100 (Generic System) selected, with links from that generic system to two fabric leaf switches (for example, a leaf participating in an ESI pair and another leaf node). In Apstra 5.1, a Generic System represents an endpoint that is not managed as a network device by Apstra (such as a server, appliance, or host), but it is still modeled so Apstra can apply interface intent (LAG vs single link), connectivity templates, and virtual network attachments.

Because the device is shown as a generic system connected on leaf-facing ports inside the fabric topology, this aligns with an internal generic system. Internal generic systems are used for servers or endpoints that reside “inside” the rack/fabric context and consume leaf switch ports as access-facing connections. This is the common representation for endpoints in EVPN-VXLAN data center designs, where the leaf switches provide the VLAN/VNI mapping and, if required, IRB gateway services within the tenant VRF (routing zone).

An external generic system is typically used for devices outside the fabric boundary—most commonly external routers, firewalls, or upstream networks attached at border leafs—where the intent is external connectivity rather than server access. The selected node is neither a peer switch nor an access switch (those are network infrastructure roles), and the UI explicitly labels it as a Generic System, confirming the correct classification as an internal generic system.

Question: 15

You are using Juniper Apstra to create security policies that create ACLs on the fabric devices. What are two valid objects that would be used within Apstra in this scenario? (Choose two.)

A. Virtual network
B. Domain name
C. Routing zone
D. Application signature

Answer: A, C

Explanation:

In Apstra 5.1, Security Policies express traffic-permit/deny intent between defined fabric endpoints, and Apstra compiles that intent into ACL enforcement on the appropriate switches (for example, on gateway interfaces for east-west segmentation and on border leaf interfaces for north-south controls). The objects you use to define that policy intent must correspond to fabric connectivity constructs that Apstra understands as endpoints in the blueprint's logical model.

Two such valid objects are Virtual Networks and Routing Zones. A virtual network represents a tenant segment (typically mapped into EVPN-VXLAN constructs such as VNI and associated IRB gateway when L3 is enabled). Policies between virtual networks are a common way to implement micro-segmentation or tier-based segmentation (web/app/db) within the same tenant boundary. A routing zone represents the L3 tenancy boundary (mapped to a VRF) and can be used to group and control connectivity at the tenant level, especially where policy needs to be expressed for aggregated tenant domains or for controls involving external connectivity.

"Domain name" and "application signature" are not endpoint objects for Apstra Security Policies in this context. They may exist in other security ecosystems, but Apstra's security intent model for ACL generation is based on topology and blueprint objects (routing zones, virtual networks, and endpoint definitions), which can then be rendered into Junos v24.4 firewall filter–style enforcement on the fabric devices.

Verified Juniper sources (URLs):

https://www.juniper.net/documentation/us/en/software/apstra5.1/apstra-user-guide/topics/topic- map/policy-security.html

https://www.juniper.net/documentation/us/en/software/apstra5.1/apstra-user- guide/topics/concept/routing-

zones.html

https://www.juniper.net/documentation/us/en/software/apstra5.1/apstra-user- guide/topics/concept/virtual-networks.html

Question: 16

You want to gracefully take a device out of service to perform an OS upgrade. How would you accomplish this task using Juniper Apstra?

A. After selecting the device in the Staged tab, you would select Deploy Mode then Upgrade.

B. After selecting the device in the Active tab, you would select Deploy Mode then Drain.

C. After selecting the device in the Dashboard tab, you would select Deploy Mode then Upgrade.

D. After selecting the device in the Staged tab, you would select Deploy Mode then Drain.

Answer: B

Explanation:

In Apstra 5.1, the correct operational method to gracefully remove a switch from service for maintenance is to set its Deploy Mode to Drain. Drain is a day-2 operational control that tells Apstra to adjust intent so the fabric can continue operating while the targeted device is logically taken out of service as much as the design allows. This is especially relevant in EVPN-VXLAN leaf-spine fabrics where taking down a spine or a leaf can disrupt underlay BGP adjacencies and overlay reachability if traffic is not shifted first.

This action is performed from the blueprint's Active view because Drain affects the currently deployed, running fabric state (not a staged design change). Selecting the device under Active and changing Deploy Mode to Drain initiates the workflow that prepares the device for maintenance by reducing its role in forwarding and/or withdrawing dependent services according to the blueprint's modeled redundancy (for example, shifting server-facing traffic to an MLAG/ESI peer where applicable, or reducing reliance on the device for transit). After the device is drained, an OS upgrade can be performed with less impact, and the device can later be returned to service by switching Deploy Mode back to Deploy and committing the change.

The "Upgrade" action is not the deploy-mode mechanism described for graceful removal; the key is Deploy Mode → Drain from Active, which is explicitly intended for maintenance and decommissioning scenarios.

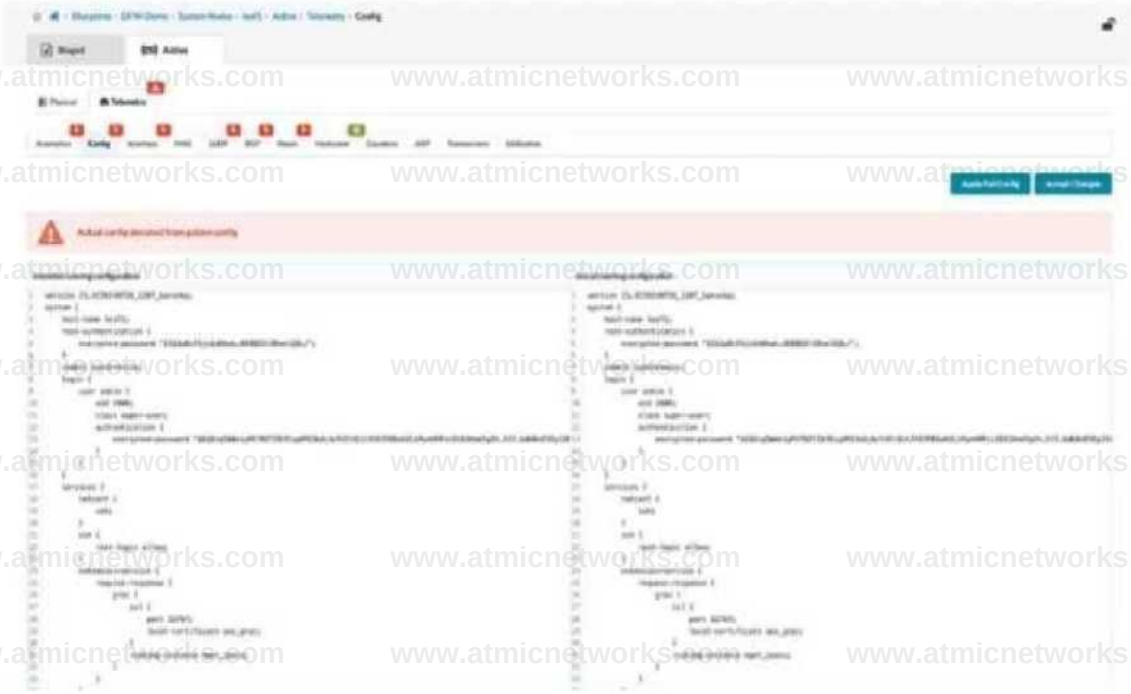
Verified Juniper sources (URLs):

https://www.juniper.net/documentation/us/en/software/apstra5.1/apstra-user- guide/topics/task/device-drain.html

https://www.juniper.net/documentation/us/en/software/apstra4.2/apstra-drain-mode/apstra-drain- mode-guide/topics/concept/apstra-drain-mode-activate-or-disable-drain.html

Question: 17

Referring to the exhibit,



what happens when an operator clicks the Accept Changes button on the right side of the screen in Juniper Apstra?

A. Apstra will add a similar configuration from a known device context to accomplish the goal of the CLI-entered configuration.

B. Apstra will incorporate the new CLI into the “golden config”.

C. Apstra will not commit new changes to this device until the user clicks the Apply Full Config button.

D. Apstra will stop warning about the changes, but the changes will be overwritten at the next commit.

Answer: B

Explanation:

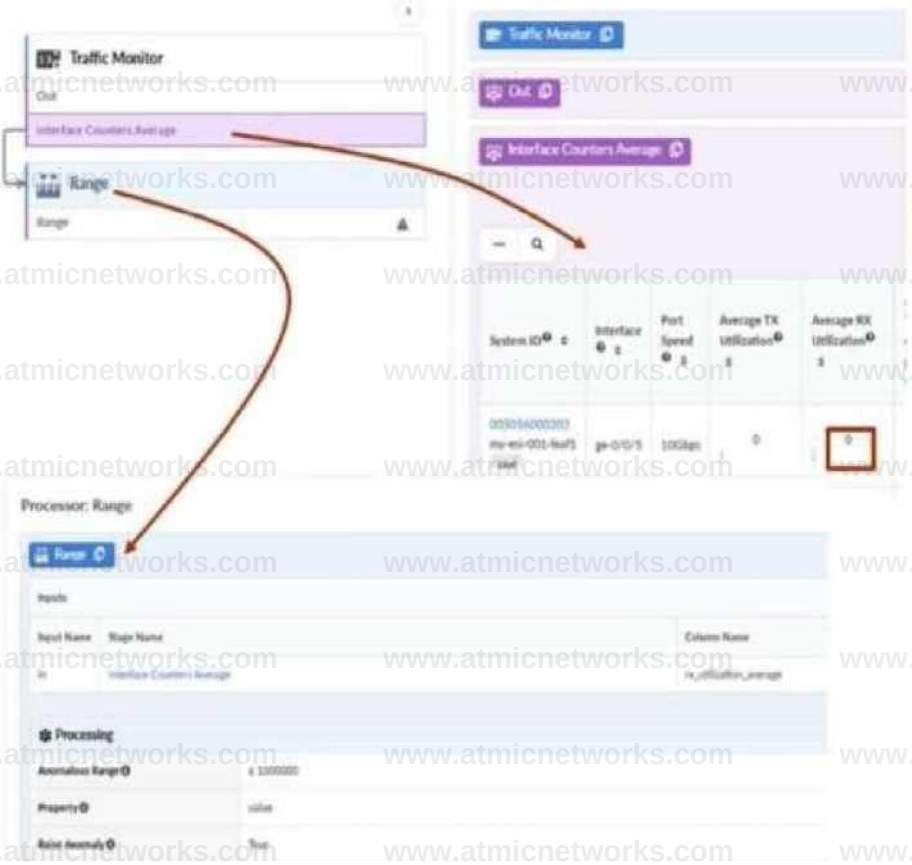
In Apstra 5.1, this screen represents a configuration deviation workflow: Apstra is comparing the intended (golden) configuration it generated from blueprint intent against the actual configuration currently on the device. When an operator makes a change directly on the switch CLI (for example, on a Junos v24.4 leaf), Apstra detects the difference and flags it as drift because it did not originate from the blueprint's intent model.

Clicking Accept Changes tells Apstra to adopt the device's current CLI state as the new accepted baseline for that device, effectively incorporating the observed CLI delta into Apstra's intended configuration for purposes of future comparison and compliance. In other words, Apstra stops treating that specific deviation as an error because it has been acknowledged and absorbed into the "golden config" (the intent-aligned configuration Apstra considers correct for that node). This is commonly used when an emergency change was made on-box and you want Apstra's source of truth to reflect it, rather than reverting it.

This differs from Apply Full Config, which is used to push Apstra's intended configuration down to the device to restore compliance. If you do not accept the change, a later commit/apply action can overwrite the CLI-entered configuration to re-align with blueprint intent.

Question: 18

The analytics probe shown in the exhibit is enabled.



The ge-0/0/5 interface on the my-esl-001-leaf1 node receives an average of greater than 1 Mbps of traffic. Which two statements are correct in this scenario? (Choose two.)

A. A probe anomaly will be raised.
B. The indicator for the Analytics tab in the blueprint will turn red.
C. The indicator for the Active tab in the blueprint will turn red.
D. A service anomaly will be raised.

Answer: A, B

Explanation:

In Apstra 5.1, an IBA probe is a defined analytics pipeline that applies to a scoped set of graph objects (here, interfaces) and evaluates telemetry against logic defined in its processors. The exhibit shows a probe that consumes Interface Counters Average and then applies a Range processor. In the processor configuration, the Anomalous Range is set to "greater than 1,000,000" (bytes per second–
equivalent for ~1 Mbps depending on the probe's metric definition), and Raise Anomaly is set to True. Therefore, when

ge-0/0/5 receives an average traffic level above the configured threshold, the probe's condition evaluates as anomalous and Apstra raises a probe anomaly for that interface. That makes statement A correct.

When probe anomalies are raised, Apstra surfaces them in the blueprint's Analytics area because they are analytics-derived findings (as opposed to configuration drift or deployment workflow issues). As a result, the blueprint's Analytics tab indicator changes state (commonly to red with a badge count) to signal active analytics anomalies requiring attention. That makes statement B correct.

This event is not classified as a service anomaly (which is associated with higher-level service intent/assurance objects) unless separately mapped by policy/logic, and it does not primarily drive the Active tab indicator, which is focused on operational state views rather than being the primary alert surface for IBA probe anomalies.

Question: 19

An operator is working on a capacity-planning exercise. The operator needs to examine the pre-built time-series information regarding link utilization. In the Juniper Apstra UI, which top-level tab would the operator have to access to find this information?

A. Active
B. Staged
C. Analytics
D. Dashboard

Answer: C

Explanation:

In Apstra 5.1, capacity planning based on pre-built time-series telemetry (such as link utilization trends) is part of Intent-Based Analytics (IBA). IBA is where Apstra ingests streaming telemetry from fabric devices, stores it as time-series data, and presents it through built-in analytics views (dashboards/widgets) and probes. Because the question specifically calls out "pre-built time series information regarding link utilization," the correct UI location is the Analytics top-level tab within the blueprint.

The Active tab is primarily oriented to operational state and day-2 workflows (for example, viewing live state, queries, and device-level operational views). The Staged tab is where you modify intent (physical/virtual design, policies, catalog items) prior to committing and deploying. The Dashboard provides a high-level blueprint overview and navigation, but the drill-down and time-series analytics views that support trending and capacity analysis are accessed via Analytics.

In an EVPN-VXLAN fabric using Junos v24.4, link utilization time-series is particularly valuable because underlay

congestion can degrade overlay performance (BGP convergence behavior, ECMP distribution effectiveness, and endpoint experience). Apstra's Analytics tab centralizes these metrics so operators can evaluate utilization baselines, identify sustained hot links, and support proactive actions (rebalancing, adding capacity, or adjusting design intent) without relying on ad-hoc perdevice CLI polling.

Verified Juniper sources (URLs):

https://www.juniper.net/documentation/us/en/software/apstra5.1/apstra-custom-telemetry- collection-guide/topics/concept/apstra-telemetry-and-intent-based-analytics.html

Question: 20

Which two statements are correct about Juniper Apstra reference designs? (Choose two.)

A. The Freeform reference design requires knowledge of device CLI.
B. The data center reference design supports only Junos devices.
C. The data center reference design requires knowledge of device CLI.
D. The Freeform reference design supports only Junos devices.

Answer: A, D

Explanation:

Apstra 5.1 provides multiple reference designs that define how intent is modeled and how configuration is produced. In the Freeform reference design, the network designer is responsible for creating and validating the device configurations using constructs such as device contexts, property sets, and config templates. Because configuration is authored rather than generated from a fixed data-center abstraction set, Freeform inherently requires practical knowledge of the device's configuration model and operational behavior—on Juniper platforms, that means being comfortable

with Junos-style configuration and show/verification workflows. This makes statement A correct.

Freeform in Apstra 5.1 is also limited to Juniper devices (Junos and Junos Evolved families). In other words, Freeform blueprints do not provide multi-vendor device onboarding and deployment in that reference design at this software level. That makes statement D correct as written in the question's choices.

By contrast, the data center reference design in Apstra 5.1 is not limited to Junos-only device families; Apstra qualifies multiple NOS families for data center operation via device profiles and packages, and it renders intent accordingly. Therefore statement B is not correct. Finally, while CLI knowledge is always useful for troubleshooting, the data center reference design does not fundamentally require an operator to hand-author device CLI to deploy a standards-based

EVPN- VXLAN fabric; that's one of the core benefits of the intent-based data center reference design.

Verified Juniper sources (URLs):

https://www.juniper.net/documentation/us/en/software/apstra5.1/apstra-user-guide/topics/concept/freeform.html

https://www.juniper.net/documentation/us/en/software/apstra4.2/apstra-freeform-4.2.1/apstra- freeform-guide/topics/concept/freeform-overview-and-design.html

https://www.juniper.net/documentation/us/en/software/apstra5.1/apstra-user- guide/topics/task/device-profile-import-freeform.html

https://www.juniper.net/documentation/us/en/software/apstra5.1/apstra-user-guide/topics/topic- map/devices-qualified.html

Question: 21

You are creating a new security policy using Juniper Apstra.

Referring to the exhibit, which application point should you select to allow or deny traffic to or from a particular VRF?

A. Routing Zone
B. External Endpoint
C. Internal Endpoint
D. Virtual Network

Answer: A

Explanation:

In Apstra 5.1, multitenancy is modeled using routing zones, which map directly to the network operating system concept of a VRF. A VRF is an isolated Layer 3 routing instance with its own routing table and forwarding context, and Apstra's routing zone is the intent-based abstraction used to define and manage that isolation consistently across the fabric. Therefore, if your goal is to allow or deny traffic to or from a particular VRF, you must select Routing Zone as the security policy application point.

This choice enables you to express policy at the tenant boundary (VRF boundary) rather than at a single segment boundary. In EVPN-VXLAN data center fabrics, a tenant VRF commonly contains multiple virtual networks (VXLAN

segments) and their associated IRB gateways on the leaf switches. Applying policy at the routing-zone level allows Apstra to compile intent and deploy enforcement consistently where traffic enters or exits that VRF context—typically as ACL constructs rendered as Junos firewall filters on the appropriate interfaces (for example, IRB interfaces for east-west controls or border interfaces for north-south controls).

By contrast, selecting Virtual Network targets a single segment (not the whole VRF), and Internal/External Endpoint targets specific endpoints or endpoint groups rather than the VRF-wide policy boundary. Hence, Routing Zone is the correct application point when policy scope is the VRF.

Question: 22

You are performing an upgrade to your switches in your network. You want to ensure that the upgrade can be performed without interrupting traffic. In the Juniper Apstra UI, which deploy mode should be used to accomplish this task?

A. Deploy
B. Undeploy
C. Drain
D. Ready

Answer: C

Explanation:

In Apstra, Deploy Mode = Drain is the operational mechanism used to gracefully remove a switch from active forwarding before performing maintenance such as an OS upgrade. Drain mode is specifically intended to drain traffic while preserving fabric stability, so that maintenance can be executed with minimal to no application impact, provided the fabric design has sufficient redundancy (for example, ECMP in the underlay and dual-homing/ESI for server attachments). In an EVPN-VXLAN IP fabric, taking a leaf or spine abruptly out of service can cause transient loss of reachability as underlay adjacencies reconverge and the overlay recalculates paths. By placing the device into Drain, Apstra adjusts intent so that traffic is shifted away from the device as much as possible, reducing dependency on it before the upgrade begins.

This is different from Undeploy, which removes Apstra-rendered configuration and is generally used for decommissioning; if a device is carrying traffic, Apstra guidance is to drain first. Ready is a predeploy state used in lifecycle workflows, not a maintenance traffic-shifting mode. Deploy keeps the device fully participating. Therefore, for a maintenance window where the goal is “upgrade with minimal interruption,” the correct mode is Drain, then perform the Junos v24.4 upgrade, and finally return the device to Deploy.

Verified Juniper sources (URLs):

https://www.juniper.net/documentation/us/en/software/apstra4.2/apstra-drain-mode/apstra-drain- mode.pdf

https://www.juniper.net/documentation/us/en/software/apstra4.2/apstra-user-guide/topics/topic- map/deploy-mode-update-datacenter.html

https://www.juniper.net/documentation/us/en/software/apstra6.0/apstra-user-guide/topics/topic- map/device-config-lifecycle.html

Question: 23

When creating a probe, an operator wants to make it easy to view that probe's output. In this scenario, which element must be created to accomplish this task?

A. A dashboard widget
B. A predefined probe
C. A processor
D. A stage

Answer: A

Explanation:

In Apstra IBA, a probe is a directed graph made of stages (data you can inspect) and processors (operations that transform/aggregate data). While stages can be inspected during probe construction, the simplest operational way to make probe results readily consumable by day-2 operators is to publish them through widgets that can be placed on Analytics dashboards. A dashboard widget is the visualization and presentation object that renders either (1) counts of anomalies or (2) the outputs produced by stages and processors in a probe. Creating a widget tied to

the probe output means the operator can open a dashboard and immediately see the metric trends, tables, or anomaly indicators without navigating into probe internals.

A predefined probe is optional content (a starting template) and is not required for visibility. Processors and stages are internal probe building blocks, but they do not, by themselves, create an operator-friendly view in the UI. In a Junos v24.4 EVPN-VXLAN fabric, this is especially useful for link utilization, drops, latency signals, or any custom telemetry pipeline: you build the probe logic once, then expose the key results in a widget that persists across operational workflows and can be shared on standardized dashboards for capacity planning and troubleshooting.

Verified Juniper sources (URLs):

https://www.juniper.net/documentation/us/en/software/apstra4.2/apstra-user-guide/topics/concept/widgets.html

https://www.juniper.net/documentation/us/en/software/apstra4.2/apstra-user-guide/topics/topic-map/widget-stage-create.html

https://www.juniper.net/documentation/us/en/software/apstra4.2/apstra-user-guide/topics/concept/probes.html

Question: 24

Juniper Apstra provides five different predefined user roles. Given this information, what is the main difference between the administrator and the user role?

A. The user role can only be assigned to specific blueprints.
B. The user role can make changes to any other role.
C. The user role cannot make any changes to other user types.
D. The user role can only make changes to the view role.

Answer: C

Explanation:

Apstra role-based access control separates fabric operations from identity and authorization administration. The administrator role includes full permissions, including the ability to manage users and roles (for example, creating users, assigning permissions, and creating/cloning/editing

custom roles where allowed). This enables administrators to govern who can access the system and what they are permitted to change across all blueprints and system settings.

The user role, in contrast, is designed for day-to-day fabric work: viewing and editing supported blueprint elements and operational objects within the scope permitted by the role, but not administering other users' access or modifying the role structure itself. In other words, a user can work on the network intent and operations, but cannot elevate privileges, change other users' roles, or otherwise manage user/role administration unless explicitly granted additional permissions through custom roles.

That makes option C the correct statement: the user role cannot make changes to other user types (that is, it lacks the

permissions needed to administer identities/roles). Options A, B, and D do not reflect Apstra's RBAC model: roles are not primarily constrained "per blueprint" in that way, and users are not intended to modify other roles—those are administrator-level capabilities.

Verified Juniper sources (URLs):

https://www.juniper.net/documentation/us/en/software/apstra6.0/apstra-user- guide/topics/concept/user-role-management.html

https://www.juniper.net/documentation/us/en/software/apstra4.2/apstra-user- guide/topics/concept/user-role-management.html

https://www.juniper.net/documentation/us/en/software/apstra5.0/apstra-user- guide/topics/concept/user-role-management.html

Question: 25

Which statement is correct about an event log?

A. It stores alerts for anomalies in the event log.
B. It can be exported to a PDF file.
C. You can view the configuration of a device.
D. It runs on the worker node.

Answer: A

Explanation:

In Juniper Apstra 5.1, the Event Log is a centralized record used for auditing and operational visibility. It includes audit events (user/system actions) and anomaly-related alerts (events generated when Apstra detects abnormal conditions). In Apstra documentation, anomaly entries are explicitly treated as "Alert" records with high severity, meaning the Event Log is a valid place to review anomaly notifications and their associated details. Therefore, the statement that the Event Log stores alerts for anomalies is correct.

The other options do not match the Event Log function. Viewing a device's configuration is handled in device configuration and blueprint operational views, not as the primary purpose of the Event Log. Export behavior, where supported, is described for formats such as CSV rather than PDF, and exporting dashboards is a separate capability from event logging. Finally, while worker nodes are used to ofload operational services (such as off-box agents and IBA components), the Event Log is a platform logging feature exposed through the controller UI/API rather than something described as "running on the worker node" as its defining trait.

Verified Juniper sources (URLs):

https://www.juniper.net/documentation/us/en/software/apstra5.1/apstra-user-guide/topics/topic- map/event-log.html

https://www.juniper.net/documentation/us/en/software/apstra5.0/apstra-user-guide/topics/topic- map/event-log.html

https://www.juniper.net/documentation/us/en/software/apstra5.1/apstra-user-guide/topics/topic- map/syslog-config.html

Question: 26

You have accessed your deployed blueprint and see the banner shown in the exhibit.

Which two statements are correct in this scenario? (Choose two.)

A. Devices must be assigned to profiles.
B. There are changes that are not active on the fabric.
C. Resources must be assigned to devices.
D. There are anomalies that must be addressed.

Answer: B, D

Explanation:

In Apstra 5.1, the top-level blueprint banner uses tab indicators (colored badges) to summarize blueprint status across areas such as Staged, Uncommitted, Active, and Analytics. The presence of an Uncommitted indicator signifies that there are staged modifications that have not yet been committed and therefore are not part of the active, deployed intent. That directly corresponds to the statement that changes exist which are not active on the fabric.

At the same time, the banner shows an Active indicator in an alarm state, which reflects that the running fabric has issues requiring attention—commonly surfaced as anomalies (for example, configuration deviation, interface/link faults, protocol/session issues, or service-impacting conditions). In Apstra's operational model, these issues appear as anomalies that operators should investigate and remediate to restore compliance and health. Therefore, the statement that there are anomalies that must be addressed is also correct.

The remaining options are not implied by this banner alone. Device profile assignment and resource assignment are build-time tasks, but their absence is not what the Uncommitted/Active alert indicators are specifically communicating here. The banner is highlighting uncommitted intent changes and active anomalies that affect the deployed blueprint state and assurance posture.

Verified Juniper sources (URLs):

https://www.juniper.net/documentation/us/en/software/apstra5.1/apstra-user-guide/topics/concept/uncommitted.html

https://www.juniper.net/documentation/us/en/software/apstra5.0/apstra-user-guide/topics/topic-map/anomalies-service-active.html

https://cloudlabs.apstra.com/labguide/Cloudlabs/6.0.0/test-drive-guide/lab1-junos-5_blueprints_.html

Question: 27

You are building a blueprint using Juniper Apstra and must change the cable map to match the physical environment. Where in the blueprint UI is this task accomplished?

A. Active → Physical → Links
B. Staged → Physical → Links
C. Active → Connectivity Templates
D. Staged → Connectivity Templates

Answer: B

Explanation:

In Apstra 5.1, the cabling map is part of the blueprint's intended physical topology. Cable-map edits are performed in the Staged workspace because Staged is where you modify intent (what the fabric should look like) before committing those changes and deploying them. The Staged → Physical → Links view provides both a tabular and topology-oriented representation of spine-to-leaf and other physical connections. When Apstra auto-assigns interfaces during initial build, the logical mapping may not match the real patching in the data center. The cabling map editor allows you to override interface names (and where applicable, link addressing metadata) so the blueprint accurately reflects the actual patch

panel and switchport usage.

This accuracy is critical in a Junos v24.4 leaf-spine fabric because underlay correctness depends on the real physical adjacencies: link membership, LAG expectations (where used), and the resulting BGP neighbor relationships that carry EVPN signaling for VXLAN overlays. By updating the cabling map in Staged, you ensure Apstra can correctly validate neighbor discovery, verify intent, and produce consistent device configuration aligned to the real-world wiring. After making the cabling corrections, you commit the staged changes and then deploy/apply so that Apstra's intent and the running network converge. This work is not performed under Active (which reflects deployed state) and is not a function of Connectivity Templates (which are for endpoint/service attachment rather than fabric cabling).

Verified Juniper sources (URLs):

https://www.juniper.net/documentation/us/en/software/apstra5.0/apstra-user-guide/topics/topic-map/cabling-map-edit-datacenter.html

https://www.juniper.net/documentation/us/en/software/apstra6.0/apstra-user-guide/topics/topic-map/cabling-map-edit-datacenter.html

https://www.juniper.net/documentation/us/en/software/jvd/jvd-dcfabric-5-stage/configuration_walkthrough.html

Question: 28

You are creating a template using Juniper Apstr

a. In this scenario, what is a rack-based design compared to a pod-based design?

A. A rack-based design allows the operator to select a specified number of downlinked servers, whereas a pod-based design only allows the operator to select fixed "pods" of servers.
B. A rack-based design refers to a three-stage Clos, and a pod design refers to a five-stage Clos.
C. A rack-based design is suitable for physical servers, whereas a pod-based design is used for Kubernetes deployments.
D. A rack-based design connects local servers only, whereas a pod-based design can include virtual workloads in a public cloud.

Answer: B

Explanation:

In Apstra 5.1, templates are used to describe the intended structure of a data center fabric. A rackbased template is used to build the common 3-stage Clos model (spines connected to racks containing leaf/top-of-rack switches and endpoints). In this design, you define the spine logical devices, select one or more rack types, specify rack counts, and

define the intended connectivity between spines and racks. This directly models a leaf-spine IP fabric typically used for EVPN-VXLAN in modern data centers.

A pod-based template, by contrast, is explicitly used to build 5-stage Clos networks. In Apstra's terminology, a pod-based template is essentially a "template of templates": it combines one or more rack-based templates (each representing a 3-stage pod) and adds an additional superspine layer to interconnect those pods into a larger, scalable fabric. This is the architectural distinction: rack-based describes the leaf-spine pod, while pod-based describes the multi-pod superspine architecture.

For Junos v24.4 EVPN-VXLAN deployments, the difference matters operationally because 5-stage fabrics introduce additional tiers and scaling considerations (for example, superspine connectivity and expanded ECMP domains). Apstra's template hierarchy ensures consistent intent modeling across both 3-stage and 5-stage topologies without requiring operators to manually redesign the fabric logic each time they scale out.

Verified Juniper sources (URLs):

https://www.juniper.net/documentation/us/en/software/apstra5.1/apstra-user-guide/topics/concept/templates.html

https://www.juniper.net/documentation/us/en/software/apstra5.1/apstra-user- guide/topics/task/template-create-pod-based.html

https://www.juniper.net/documentation/us/en/software/apstra5.1/apstra-user-guide/topics/topic- map/5-stage-clos.html

Question: 29

What are two types of virtual networks defined inside Juniper Apstra software? (Choose two.)

A. VLAN
8. L3 VPN
C. VXLAN
D. L2 VPN

Answer: A, C

Explanation:

In Apstra 5.1, a Virtual Network (VN) is Apstra's abstraction for a Layer 2 forwarding domain that groups endpoints into

a logical segment across the fabric. Apstra defines virtual networks as being constructed using either VLANs or VXLANs. A VLAN-based VN represents a Layer 2 domain identified by a VLAN ID and is typically used where you want traditional VLAN semantics (often in smaller environments, migration scenarios, or designs where an overlay is not required). A VXLAN-based VN represents the same Layer 2 intent but uses a VXLAN VNI for scalable overlay segmentation, which is the common approach in EVPN-VXLAN data center fabrics.

In an IP fabric architecture, VXLAN provides encapsulation to carry tenant segments over the routed underlay, while EVPN provides the control-plane signaling for MAC/IP reachability. Junos v24.4 leaf devices act as VTEPs, mapping local VLANs/bridge-domains to VNIs and participating in EVPN for advertisement and convergence. Apstra's VN construct allows you to create the segment once (as VLAN or VXLAN type), then consistently attach it to racks, ports, and endpoints through intent-driven workflows (such as connectivity templates and virtual network assignments).

"L2 VPN" and "L3 VPN" are service provider terms and are not the VN "types" in Apstra's data center reference design. In Apstra, tenant L3 separation is modeled by routing zones (VRFs), while the VN itself is specifically either VLAN-based or VXLAN-based.

Verified Juniper sources (URLs):

https://www.juniper.net/documentation/us/en/software/apstra5.1/apstra-user- guide/topics/concept/virtual-networks.html

https://www.juniper.net/documentation/us/en/software/apstra5.1/apstra-user-guide/topics/topic- map/virtual-network-create.html

Question: 30

What are two agent processes that operate within the Juniper Apstra device agent? (Choose two.)

A. Routing agent
B. Authentication agent
C. Telemetry agent
D. Deployment agent

Answer: C, D

Explanation:

In Apstra deployments that use on-box device agents, the agent package installs multiple processes inside the switch's

NOS namespace to provide an isolated runtime environment for Apstra control and telemetry collection. Two of those processes are the Telemetry Agent and the Deployment Agent. The Telemetry Agent is responsible for collecting operational information from the device— such as LLDP neighbor details, routing-related state, and interface information—and sending that telemetry upstream to Apstra. This telemetry is a key input for closed-loop assurance in EVPN-VXLAN fabrics, where Apstra correlates underlay health (interfaces, neighbors, sessions) with overlay services.

The Deployment Agent is responsible for receiving configuration content pushed from Apstra and applying it on the device. In a Junos v24.4 fabric, this is the component that enables Apstra to converge device configuration to the blueprint's intent (for example, BGP underlay, EVPN signaling, and VXLAN constructs) without requiring manual CLI workflows. Both agents are typically idle most of the time, becoming active when Apstra needs to apply configuration changes or when significant state changes trigger telemetry updates.

Other listed options—"routing agent" and "authentication agent"—are not the named Apstra device-agent processes described for the on-box agent package in Juniper documentation.

Verified Juniper sources (URLs):

https://www.juniper.net/documentation/us/en/software/apstra4.2/apstra-server-and-security-guide/topics/concept/apstra-device-agents.html

Question: 31

You staged several changes to your Juniper Apstra blueprint but have not committed them. In this scenario, what is the effect of selecting Revert?

A. All the staged changes are cleared.
B. Only the last staged change will be cleared.
C. The current active configuration will be replaced by the previous active configuration.
D. A commit is required to complete the revert operation.

Answer: A

Explanation:

In Apstra 5.1, blueprint changes follow an intent workflow: you edit intent in Staged, then review the delta in Uncommitted, and finally Commit to activate those changes and create a new revision. If you have staged changes that are visible under Uncommitted but decide not to proceed, the Revert action is used to discard them. Selecting Revert

clears the blueprint's uncommitted intent delta and returns the blueprint to the last committed state (the currently active intended design baseline). In practical terms, it removes all pending edits that were made since the last commit—whether those edits were physical (links/topology), virtual (routing zones, virtual networks), policies (security policies), or catalog-driven operations—so that none of those changes will be deployed.

Revert is not a "single-step undo" limited to only the most recent change; it is a discard of the staged/uncommitted change set. It also does not roll back device configurations on its own (that is handled by revision operations such as Time Voyager rollbacks and subsequent deployment actions). Finally, Revert does not require a commit to take effect; it is used specifically to avoid committing changes. This behavior helps maintain clean operational control in EVPN-VXLAN fabrics by ensuring only validated and intentional intent updates are promoted to the deployed network state.

Verified Juniper sources (URLs):

https://www.juniper.net/documentation/us/en/software/apstra4.2/apstra-user- guide/topics/task/blueprint-commit-revert.html

https://www.juniper.net/documentation/us/en/software/apstra6.1/apstra-user- guide/topics/task/time-voyager-rollback-blueprint-revision.html

Question: 32

What is the purpose of an EVPN Ethernet segment identifier (ESI)?

A. To provide a hop count between devices
B. To identify Layer 2 frame types for filtering purposes
C. To specify a BGP community
D. To prevent loops within a LAG connection

Answer: D

Explanation:

In EVPN multihoming, the Ethernet Segment Identifier (ESI) is the mandatory identifier used to represent a multihomed Ethernet segment—for example, a server or downstream switch that is dual-homed to two leaf devices using a single logical LAG/port-channel. By assigning the same ESI to the participating leaf-facing interfaces, the fabric recognizes those links as belonging to one Ethernet segment and can apply EVPN multihoming procedures consistently across the pair.

A key outcome of EVPN multihoming is loop prevention for multi-attached Layer 2 domains. EVPN uses the Ethernet

segment concept (identified by the ESI) along with Designated Forwarder (DF) election to ensure that only the appropriate device forwards BUM (broadcast, unknown unicast, multicast) traffic toward the multihomed segment, avoiding duplicate forwarding and L2 loops. In addition, ESI-based multihoming supports resilient forwarding behavior during failures (for example, link or node loss) while maintaining correct advertisement and convergence in the EVPN control plane.

Therefore, among the provided options, the purpose that best matches how ESI is used operationally is to prevent loops within a LAG/multihomed connection, which is fundamental to EVPN-VXLAN data center designs on Junos v24.4 leaf devices and is also explicitly supported by Apstra when modeling ESI-based dual-homing.

Verified Juniper sources (URLs):

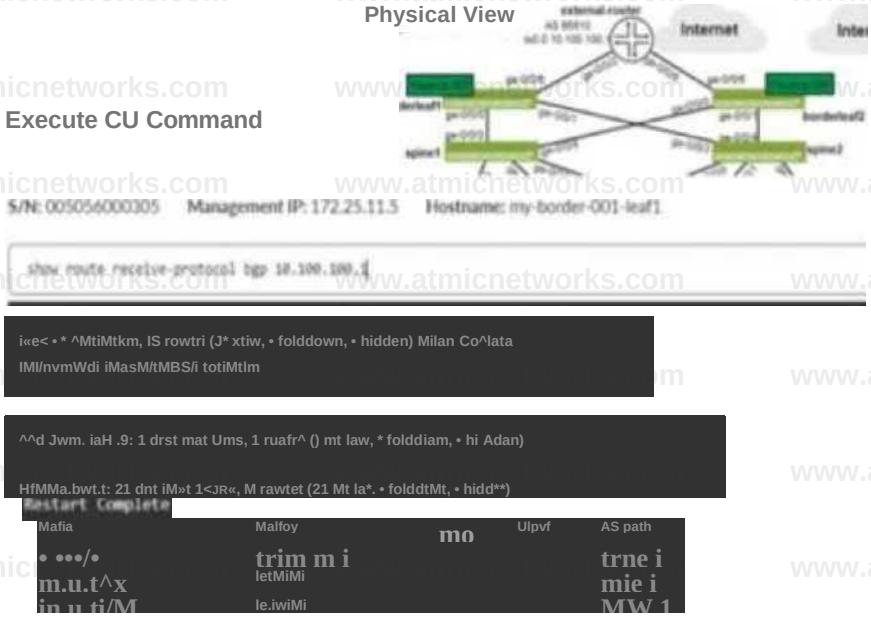
https://www.juniper.net/documentation/us/en/software/nce/evpn-lag-multihoming- guide/topics/concept/evpn-lag-guide-introduction.html

https://www.juniper.net/documentation/us/en/software/nce/evpn-lag-multihoming- guide/topics/task/evpn-lag-guide-esi-types-lacp.html

https://www.juniper.net/documentation/us/en/software/junos/evpn/topics/topic-map/evpn-mh-df- election.html

Question: 33

The same connectivity template is applied to the ge-0/0/6 interface on both borderleaf1 and borderleaf2 nodes. This connectivity template describes the intended configuration of the eBGP session between each of the borderleaf nodes and the external router. You want to ensure that the 172.23.x/24 routes are not installed in the borderleaf nodes' Finance routing table.



Referring to the exhibit, what would you change in Juniper Apstra to accomplish this task?

A. Add an aggregate prefix to the routing policy.

B. Modify the import policy to only allow the default route.

C. Select the "Expect Default IPv4 Route" checkbox.

D. Modify the export policy to only allow the default route.

Answer: B

Explanation:

The exhibit shows the border leaf receiving multiple routes via BGP from the external router, including 172.23.x/24 prefixes, and those routes appearing in the Finance VRF routing table. To stop these routes from being installed in the Finance table, you must change what the border leaf imports from that eBGP session. In Apstra, this control is implemented through a Routing Policy attached to the protocol session described by the connectivity template. By setting the Import Policy to accept default route only, Apstra renders Junos policy so that only 0.0.0.0/0 is imported into the VRF, while the 172.23.x/24 prefixes are rejected and therefore never installed in Finance.inet.0.

Option C is a common trap: the "Expect Default IPv4 Route" setting is an assurance expectation—it generates an expectation/anomaly if the default route is missing, but it does not change device configuration or filtering behavior. Export-policy changes (option D) would only affect what the border leaf advertises outbound to the external router, not what it learns inbound. Aggregation (option A) does not prevent installation of the specific learned /24s; it changes advertisement behavior rather than import filtering. The correct fix is to tighten the import policy on that external eBGP session.

Verified Juniper sources (URLs):

https://www.juniper.net/documentation/us/en/software/apstra5.1/apstra-user-guide/topics/concept/routing-policies.html

https://www.juniper.net/documentation/us/en/software/apstra6.0/apstra-user-guide/topics/concept/routing-policies.html

https://www.juniper.net/documentation/us/en/software/apstra4.2/apstra-user-guide/topics/concept/connectivity-templates.html

Question: 34

What does VXLAN use to uniquely label and identify broadcast domains?

A. VLAN ID

B. Agent Circuit Identifier (ACI)

C. Virtual Network Identifier (VNI)

D. End System Identifier (ESI)

Answer: C

Explanation:

In a VXLAN overlay, each Layer 2 broadcast domain (the logical equivalent of a VLAN/bridge domain) is identified by a 24-bit VXLAN Network Identifier (VNI) carried in the VXLAN header. This VNI is what allows the overlay to scale far beyond traditional VLAN space (12-bit VLAN IDs), enabling up to ~16 million distinct segments. In an EVPN-VXLAN data center fabric, Junos v24.4 leaf switches operate as VTEPs and map local bridge domains (often associated with VLANs on server-facing ports) to a VNI. When traffic is sent across the routed underlay, the leaf encapsulates Ethernet frames into VXLAN packets and inserts the VNI so the receiving VTEP can place the frame into the correct broadcast domain on decapsulation.

Apstra 5.1 abstracts this mapping through virtual networks and resource allocation: when you define a VXLAN-based virtual network, Apstra allocates a VNI from the appropriate pool and consistently programs the necessary constructs on all participating leaves. The key point is that VNI is the unique identifier in the VXLAN data plane used to label the broadcast domain across the IP fabric; VLAN IDs may exist locally at the edge for tagging, but the globally significant overlay identifier is the VNI.

Verified Juniper sources (URLs):

https://www.juniper.net/documentation/us/en/software/junos/evpn/topics/topic-map/sdn- vxlan.html

Question: 35

Off-box agents are consuming too much CPU and memory on your Juniper Apstra controller. In this scenario, how would you solve this problem?

A. Add more CPU and memory to the Apstra controller VM.

B. Create a worker VM to ofload off-box agents from the Apstra controller VM.

C. Use on-box agents instead of off-box agents.

D. Modify the agent profile to consume less resources.

Answer: B

Explanation:

In Apstra 5.1, off-box agents and analytics services are delivered as containerized workloads that consume CPU and memory within the Apstra cluster. When these workloads are concentrated on the controller node, the controller can become resource-constrained, impacting overall responsiveness and scaling limits. The supported architectural solution is to add a worker node (worker VM) and allow Apstra to place offbox (and, if applicable, iba) containers on that worker. This increases cluster capacity and shifts runtime load away from the controller, which should remain focused on core control-plane and management functions.

Juniper's sizing guidance also treats worker nodes as the scalable unit for off-box agent growth: each VM node (controller or worker) supports a bounded number of off-box agents, and when one VM is insufficient, the prescribed approach is to increase capacity by adding worker nodes to the Apstra VM cluster. This method scales horizontally and avoids overloading the controller with operational containers.

While increasing CPU/memory on the controller might help temporarily, the documented design pattern for sustained growth is to distribute the off-box workloads across worker nodes. Switching to on-box agents is a different operational model and not the direct remediation for controller resource pressure in an off-box deployment.

Verified Juniper sources (URLs):

https://www.juniper.net/documentation/us/en/software/apstra5.1/apstra-install-upgrade/topics/ref/apstra-server-resources.html

https://www.juniper.net/documentation/us/en/software/apstra5.0/apstra-user-guide/topics/topic-map/apstra-cluster-nodes.html

https://www.juniper.net/documentation/us/en/software/apstra4.2/apstra-user-guide/topics/topic-map/apstra-cluster-nodes.html

Question: 36

You must enable ECMP load balance in your three-stage eBGP IP Clos fabric. In this scenario, which two features are required? (Choose two.)

A. A load-balance policy applied to the forwarding table
B. multipath multiple-as applied to the BGP group
C. A load-balance policy applied to the BGP group
D. multihop applied to the BGP group

Answer: A, B

Explanation:

In a three-stage eBGP IP Clos, a leaf typically learns the same prefix through multiple spines, and those spines are commonly in different AS numbers (a standard Clos pattern for simplified operations and fault isolation). To install multiple equal-cost eBGP paths for the same destination, Junos requires BGP multipath and, when the candidate paths come from different neighbor ASs, you must use the multiple-as capability (expressed as "multipath multiple-as" in many designs) to bypass the default same-neighbor-AS restriction for multipath selection. This ensures multiple eligible eBGP next-hops can be accepted as equal-cost paths and installed.

Separately, to actually perform load balancing in the forwarding plane across those installed nexthops, Junos uses a load-balancing policy that is applied to the forwarding table via routing-options forwarding-table export <policy>. This policy (for example, per-packet or consistent-hash variants, depending on platform and intent) activates the forwarding behavior needed to distribute traffic over the ECMP set.

A "load-balance policy applied to the BGP group" is not the required mechanism for ECMP forwarding behavior in this context, and multihop is unnecessary in a Clos underlay because neighbors are typically directly connected.

Verified Juniper sources (URLs):

https://www.juniper.net/documentation/us/en/software/junos/bgp/topics/topic-map/load- balancing-bgp-session.html

Question: 37

Which three statements are correct about property sets? (Choose three.)

A. They are imported when a configlet is imported into a blueprint.
B. The key/value pairs are used for variable substitution.
C. They are used only by configlets in a blueprint.
D. The syntax used when creating property sets is specific to each supported vendor.
E. Multiple property sets can be referenced by a configlet.

Answer: A, B, E

Explanation:

In Apstra 5.1, property sets are structured data objects (YAML/JSON) used to hold values that templates can consume at render time. Their most common use is with configlets, where property set key/value pairs are referenced as variables inside the template so Apstra can perform variable substitution during configuration generation. This directly supports statement B.

Property sets are also designed to be reusable. A single configlet can reference more than one property set (for example, one set for NTP servers and another for syslog collectors), allowing clean separation of data domains and easier lifecycle updates. This supports statement E.

Operationally, when you bring design content into a blueprint, the blueprint must have the required supporting objects available. In Apstra workflows, configlets that use property sets require those property sets to be present in the blueprint context (commonly accomplished by importing the relevant property set(s) from the catalog into the blueprint as part of bringing in the configlet and its dependencies). This aligns with statement A as the blueprint-level outcome: the property sets used by an imported configlet are imported/available in the blueprint for rendering.

Statements C and D are incorrect because property sets are not limited only to configlets (they are also used with Analytics probes), and the syntax is not vendor-specific—Apstra uses standard YAML/JSON structures independent of NOS.

Verified Juniper sources (URLs):

https://www.juniper.net/documentation/us/en/software/apstra5.1/apstra-user-guide/topics/concept/property-set-datacenter-design.html

https://www.juniper.net/documentation/us/en/software/apstra4.2/apstra-user-guide/topics/concept/property-set-datacenter-design.html

https://www.juniper.net/documentation/us/en/software/apstra5.1/apstra-user-guide/topics/ref/configlet-examples.html

Question: 38

You are considering the bridged overlay EVPN-VXLAN architecture. In this scenario, how many VLANs would be enabled in the VLAN-based service type at the MAC-VRF EVPN instance level?

A. One VLAN
B. Two VLANs
C. 4000 VLANs
D. Four VLANs

Answer: A

Explanation:

In Junos EVPN, the service type determines how VLANs (broadcast domains) are mapped into EVPN constructs. With VLAN-based service, the mapping is one-to-one: a single VLAN (single broadcast domain) maps to a single EVPN instance (EVI), resulting in a separate bridge table per VLAN. When you implement EVPN-VXLAN in a bridged overlay (L2 extension over an L3 underlay), leaf devices act as VTEPs and encapsulate VLAN traffic into VXLAN using the associated VNI, but the service-type mapping rule still applies.

At the MAC-VRF hierarchy, you configure the EVPN-VXLAN parameters and choose the service type for that EVPN instance. If the service type selected is VLAN-based, then the EVI represents exactly one VLAN at that EVPN instance level. If you need multiple VLANs under the same MAC-VRF/EVI construct, Junos provides alternative service types (for example, VLAN-aware or VLAN-bundle) specifically intended to associate multiple VLANs with a single EVPN instance; that is not what VLANbased service does.

Therefore, in a bridged overlay EVPN-VXLAN design using VLAN-based service at the MAC-VRF EVPN instance level, the correct answer is one VLAN. This remains true regardless of how many total VLANs exist across the fabric; each VLAN-based EVI handles a single VLAN.

Verified Juniper sources (URLs):

https://www.juniper.net/documentation/us/en/software/junos/evpn/topics/concept/evpn-based-services.html

https://www.juniper.net/documentation/us/en/software/junos/evpn/topics/concept/mac-vrf- routing-instance-overview.html

Question: 39

In a three-stage Clos network, what are important key design aspects of a data center fabric? (Choose two.)

A. External traffic enters and exits the spine devices.
B. Fabric traffic enters and exits the border devices.
C. Servers are connected to leaf devices.
D. Servers are connected to spine devices.

Answer: B, C

Explanation:

A three-stage Clos (leaf–spine) data center fabric separates roles to keep forwarding predictable and scalable. The leaf layer is the attachment point for endpoints, so servers connect to leaf devices. This is where the fabric accepts workload traffic, applies edge policies, and provides tenant/service constructs (for EVPN-VXLAN fabrics, the leafs typically act as VTEPs and host IRB gateways as required). The spine layer provides the non-blocking transit core between leafs using L3 underlay routing and ECMP, but it is not the place where servers attach directly.

For north-south connectivity, a typical three-stage Apstra data center architecture includes border leaf switches. These border devices provide the controlled connection point between the fabric and external networks (Internet/WAN/DCI or upstream services). As a result, fabric traffic enters and exits the border devices, not the spines. This design keeps the spines dedicated to high-speed eastwest transit and simplifies operations: external routing policies, security controls, and inter-domain handoffs are concentrated at the border, while the spines remain a uniform L3 transit tier. In Junos v24.4 EVPN-VXLAN deployments, this separation also helps maintain clean overlay boundaries and consistent underlay behavior across the fabric.

Verified Juniper sources (URLs):

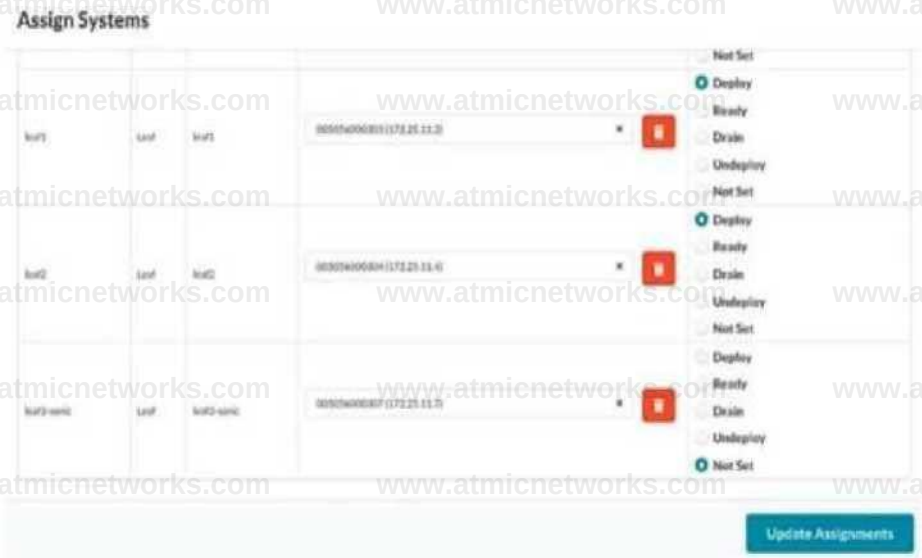
https://www.juniper.net/documentation/us/en/software/juniper-data-center-assurance/user-guide/topics/concept/dc-network-topology-on-dc-assurance.html

https://www.juniper.net/documentation/us/en/software/jvd/jvd-3-stage-datacenterdesign-with- juniper-apstra/solution_architecture.html

https://www.juniper.net/content/dam/www/assets/white-papers/us/en/design-considerations-for- spine-and-leaf-ip-fabrics.pdf

Question: 40

You have created a blueprint and are in the process of assigning systems. You require the leaf3-sonic device in the blueprint but do not want it to actively participate in the routing of the IP fabric.



In the Juniper Apstra UI, which two modes satisfy this requirement? (Choose two.)

A. Drain
B. Ready
C. Deploy
D. Undeploy

Answer: B, D

Explanation:

Apstra deploy modes control how far a device progresses in the configuration lifecycle and whether it becomes active in the fabric. If you must keep leaf3-sonic present in the blueprint (modeled, cabled, and available for future use) but you do not want it to participate in IP-fabric routing, you use modes that keep the device not active.

Ready mode assigns the device to the blueprint and applies only "Ready (Discovery 2)" level configuration—hostnames, interface descriptions, and port speed/breakout settings—while explicitly keeping the device out of fabric routing. In this mode, Apstra does not configure routing/BGP or L3 interface addressing for the IP fabric, so the switch is staged and visible for validation (for example, LLDP wiring checks) but does not forward as part of the Clos underlay.

Undeploy mode removes the complete Apstra service configuration from the device. Operationally, this also ensures the device is not active in the fabric. It is commonly used when a device must be retained in the blueprint inventory/topology but should not be participating (for example, temporarily withdrawn, decommission preparation, or held as a spare).

By contrast, Deploy makes the device active (full rendered fabric configuration, including BGP), and Drain is a

maintenance state used to gracefully remove traffic from an already-active device rather than a state for keeping it non-participatory from the outset.

Verified Juniper sources (URLs):

https://www.juniper.net/documentation/us/en/software/apstra6.0/apstra-user-guide/topics/topic-map/device-config-lifecycle.html

https://www.juniper.net/documentation/us/en/software/apstra4.2/apstra-user-guide/topics/topic-map/device-config-lifecycle.html

Question: 41

What are two available Juniper Apstra template types? (Choose two.)

A. Collapsed
B. Rack-based
C. Compressed
D. Device-based

Answer: A, B

Explanation:

In Juniper Apstra 5.1, a template is a design abstraction used to create a blueprint. It captures the intended topology shape and design rules without tying the design to a specific vendor's CLI. Apstra supports multiple template types to match common data center fabric architectures.

A rack-based template is used for the standard three-stage Clos (leaf–spine) approach. In this model, you define the spine logical devices and one or more rack types (containing leaf devices and optional endpoint constructs). This is the dominant pattern for EVPN-VXLAN IP fabrics: leaf switches provide server attachment, VXLAN encapsulation (VTEP function), and optional IRB gateways, while spines provide high-capacity L3 transit with ECMP.

A collapsed template is used for a spine-less (spineless) topology. Instead of a separate spine tier, a collapsed design models a fabric where leaf nodes interconnect in a mesh-like arrangement (as supported by the template type) to provide underlay reachability and redundancy. This can be useful for smaller environments or edge data centers where a full spine tier is unnecessary.

"Compressed" and "device-based" are not Apstra template types. Junos v24.4 is relevant when the blueprint is

instantiated and deployed, but the template type selection is an Apstra design-time decision that determines the fabric topology class.

Question: 42

What are three valid resource types supported within Juniper Apstra? (Choose three.)

A. Integer pools
B. Routing zone pools
C. Interface pools
D. VNI pools
E. ASN pools

Answer: A, D, E

Explanation:

In Apstra 5.1, resources are values that must be allocated uniquely and consistently across a fabric so Apstra can render deterministic, conflict-free configurations. These values are managed through resource pools, which provide ranges (or sets) of assignable identifiers that Apstra can automatically allocate to blueprint elements during build and deployment.

Three valid resource pool types in Apstra are ASN pools, VNI pools, and integer pools. ASN pools supply Autonomous System Numbers used in IP fabric underlays (commonly eBGP in three-stage Clos), ensuring each device or role receives the correct AS assignment without manual tracking. VNI pools supply VXLAN Network Identifiers for overlay segmentation in EVPN-VXLAN fabrics. Apstra uses these to create scalable tenant segments where the VNI uniquely identifies the broadcast domain in the overlay, and Junos v24.4 devices (leaf VTEPs) are configured accordingly. Integer pools provide generic numeric values used mainly in Freeform-style designs or in situations where a template needs a consistent allocated integer (for example, a custom ID used by a configlet or another allocation-driven construct).

“Routing zone pools” and “interface pools” are not resource pool types in Apstra. Routing zones (VRFs) are blueprint design objects, and interfaces are physical/logical constructs, but neither is consumed as an allocatable “pool” resource type in the Apstra resource catalog model.

Question: 43

You are trying to deploy a five-stage template to a blueprint as shown in the exhibit. You cannot see your template name in the list of available templates.

In this scenario, which statement is correct?

A. The Collapsed option should be selected.
B. You must include “five-stage” in the template name for it to appear in the list.
C. The Pod Based option should be selected.
D. Only Freeform-type blueprints support five-stage templates.

Answer: C

Explanation:

In Apstra 5.1, templates are organized by template type, and the Create Blueprint screen can filter the template list by those types. A five-stage Clos design in Apstra is represented as a pod-based template (multi-pod fabric with an additional tier such as super-spines to interconnect pods).
Because of that, a five-stage template will only appear in the template selector when the Pod Based filter is chosen. If the filter is set to Rack Based or Collapsed, Apstra hides pod-based templates because those types

correspond to different topology classes: rack-based aligns with a three-stage leaf–spine pod, while collapsed aligns with a spine-less topology pattern.

This behavior is by design to prevent selecting an incompatible template for the intended topology. The template name itself does not need to contain “five-stage”; Apstra determines its type from how the template was created and stored in the catalog. Also, five-stage templates are not limited to Freeform blueprints—five-stage is a data center fabric topology choice, and it is supported within the data center reference design workflows when the appropriate template type is selected.

So, to make the five-stage template visible and selectable, choose Pod Based in the template filter.

Question: 44

Which two statements are correct about a Juniper Apstra server? (Choose two.)

A. The Juniper Apstra server uses Layer 2 to communicate with managed devices.
B. The Juniper Apstra server requires one network adapter connection for each managed device.
C. The Juniper Apstra server uses Layer 3 to communicate with managed devices.
D. The Juniper Apstra server requires a single network adapter.

Answer: C, D

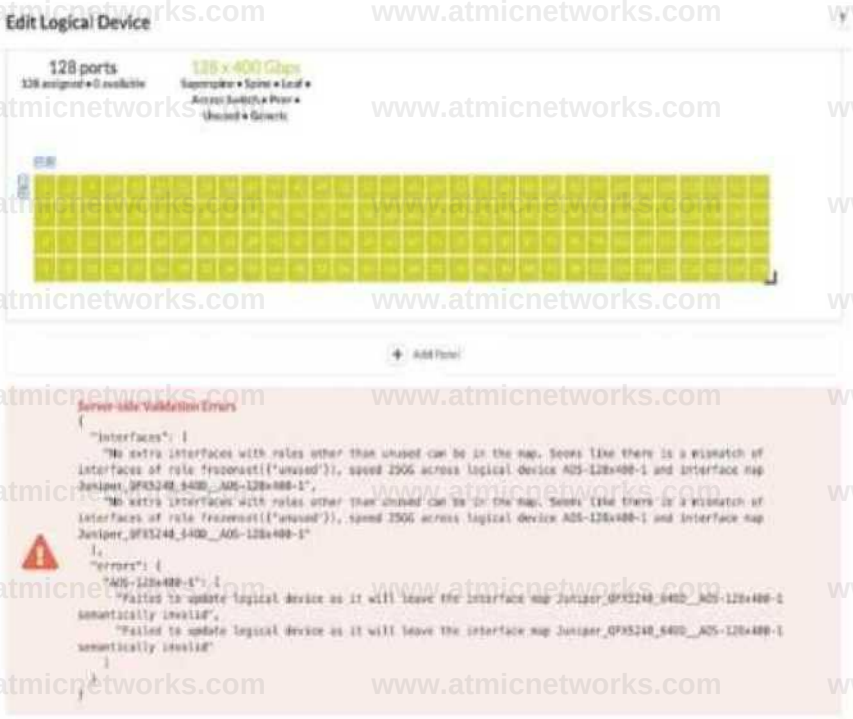
Explanation:

Apstra manages devices using IP connectivity over the management network, which is a Layer 3 relationship. Whether you are using on-box agents or off-box agents, the controller (or cluster) communicates with the fabric devices using IP reachability (for example, to exchange management traffic, retrieve discovery state, collect telemetry, and push configuration). This is why Layer 2 adjacency is not required between the Apstra server and the managed switches; the essential requirement is routable IP connectivity and appropriate access (credentials/agent connectivity) to the device management interfaces.

From a platform perspective, Apstra does not need a dedicated physical NIC per managed device. Instead, the server/VM requires connectivity to the management network through a single network adapter, and that interface can route to all managed devices. In a typical data center deployment, the Apstra controller VM sits on a management VLAN/subnet and reaches the entire fabric through routed management. This scales operationally: adding devices does not require adding additional server NICs; it only requires IP reachability and capacity planning for telemetry and agent workloads. Thus, the correct statements are that Apstra uses Layer 3 to communicate with managed devices and that it requires a single network adapter for that management connectivity model.

Question: 45

You are using Juniper Apstra to create logical devices and interface maps. You use them in three different rack types. You then modify the logical devices to support the required increased interface speeds and receive an error message when updating the logical devices.



Referring to the exhibit, which action is needed to remove the error?

A. Remove any templates that reference the logical device.
B. Remove any interface maps that reference the logical device.
C. Remove any racks that reference the logical device.
D. Remove any templates, racks, and interface maps that reference the logical device.

Answer: D

Explanation:

In Apstra 5.1, a logical device defines the abstract port layout and capabilities (including supported speeds), while an interface map binds that abstract port layout to the real, vendor-specific frontpanel ports. Rack types then consume logical devices and interface maps to model the rack's leaf/superspine roles. Once a logical device is referenced by interface maps and used inside rack types (and potentially templates that instantiate those rack

types), Apstra treats the combination as a consistent contract: port counts, roles, and speeds must remain semantically valid for every object that depends on it.

The exhibit's validation error indicates that after changing interface speeds on the logical device, the existing interface map(s) and their usage in rack types no longer match the logical device definition (for example, the map expects certain ports/speeds/roles, but the updated logical device would leave the map invalid). Because the logical device is being consumed in multiple places, the safest and required way to remove the error is to remove all dependencies—templates (if they reference the rack types), rack types, and interface maps—so Apstra can accept the new logical device definition without violating existing mappings. After updating the logical device, you then recreate or update the interface maps and re-associate them with the rack types/templates so the entire chain remains consistent under the new speed requirements.

Question: 46

You are assigning managed devices to a blueprint, for a fully functioning IP fabric. In the Juniper Apstra UI, which mode should you choose for this task?

A. Deploy
B. Ready
C. Not Set
D. Drain

Answer: A

Explanation:

In Apstra, Deploy mode is the state in which a device is intended to fully participate in the fabric. For a three-stage eBGP IP Clos (typical EVPN-VXLAN underlay), "fully functioning" means the switch receives the complete, intent-derived configuration required for production operation—underlay interface addressing, BGP peering, routing policy constructs, and any overlay-related prerequisites appropriate for its role (leaf, spine, border leaf). In Apstra's device configuration lifecycle, Deploy is the mode that causes Apstra to render and apply the full set of intended services for that node so it becomes an active member of the IP fabric and contributes to ECMP pathing and control-plane adjacency.

By contrast, Ready is commonly used when you want the device discovered and prepared (for example, basic identity and interface readiness), but not actively routing in the fabric. Drain is a maintenance state used to gracefully withdraw an already-deployed device from forwarding to minimize impact (for example, for upgrades or repairs). Not Set indicates the deploy mode has not been chosen and therefore does not represent an

operationally complete participation state.

Therefore, when your objective is an operational IP fabric where the assigned devices are actively routing and forwarding according to blueprint intent on Junos v24.4, the correct choice is Deploy.

Question: 47

You have a configuration deviation in the Juniper Apstra dashboard. What does this anomaly indicate in this scenario?

A. A device's configuration has been updated by the server.
B. A device is ready to be configured by the system.
C. A device's configuration has been changed using a method outside of Apstra.
D. A device cannot support a configuration command sent by the system.

Answer: C

Explanation:

A configuration deviation (also called a configuration anomaly) in Apstra indicates that the device's running configuration differs from Apstra's intended (golden) configuration for that node. In day-today operations, this most commonly occurs when an operator makes a change outside of Apstra's control, such as entering commands directly on the device CLI (for example, on a Junos v24.4 switch), using another automation system, or applying an out-of-band configuration method.

Apstra continuously compares the device's operational configuration against what it expects based on blueprint intent. When it detects drift, it raises a deviation anomaly so operators can decide how to restore compliance. Typical remediations are either (1) remove/revert the out-of-band change so the device matches intent again, or (2) explicitly acknowledge the change in Apstra (for example, via an accept/suppress workflow, depending on the exact UI action and version), so the deviation is no longer treated as unexpected.

While it is also possible for a deviation to be triggered by a device not accepting a rendered command (capability mismatch), the question asks what the anomaly indicates in this scenario; the primary meaning of "configuration deviation" is configuration changed outside of Apstra and therefore the network is no longer aligned with the intended state. That corresponds to option C.

Question: 48

What are three phases of the Juniper Apstra data center life cycle? (Choose three.)

A. Configuration
B. Design
C. Installation
D. Operational phase
E. Deployment

Answer: B, D, E

Explanation:

Juniper Apstra describes data center fabric management as a full lifecycle that spans three core phases: Design (Day 0), Deployment (Day 1), and Operations (Day 2). These phases map directly to how Apstra applies intent-based networking to a data center fabric.

In the Design phase, you model the intended architecture—templates (3-stage or 5-stage), rack types, logical devices, interface maps, resource pools, and high-level constructs such as routing zones and virtual networks. The objective is to capture intent in a vendor-agnostic way while ensuring consistency and validation before anything is pushed.

In the Deployment phase, Apstra turns the modeled intent into device-level implementation. This includes onboarding systems, assigning device profiles, allocating resources, rendering configurations, and pushing the resulting configuration to switches so the IP fabric becomes operational. This is where Junos v24.4 leaf/spine nodes receive underlay and overlay configuration generated from the blueprint.

In the Operational phase, Apstra continuously validates the running network against intent using telemetry and analytics (IBA), detects deviations and anomalies, supports maintenance workflows (such as drain), and provides troubleshooting tools (queries, time-series utilization, and configuration compliance).

"Configuration" and "installation" are activities that occur within the lifecycle, but the lifecycle phases themselves are Design, Deployment, and Operations.

Question: 49

You are allowed to assign tags for which three objects? (Choose three.)

A. Virtual networks
B. Interfaces
C. Generic systems
D. Property sets
E. Device profiles

Answer: A, B, C

Explanation:

In Apstra, tags are an intent-level metadata mechanism used to classify objects and drive automation and reuse. Within a data center blueprint, Apstra supports tagging multiple blueprint objects so operators can apply configuration or policy logic conditionally (for example, applying a connectivity template or a configlet based on a tag match). In this scenario, three valid taggable objects are virtual networks, interfaces, and generic systems.

Virtual network tagging is supported directly from the blueprint's virtual network table, enabling you to label virtual networks (such as "finance," "pci," or "dev") and then reference those tags elsewhere in blueprint operations and policy application. Interface tagging is also explicitly supported in the blueprint, allowing you to assign tags to switch interfaces and use those tags to control how templates, assignments, or other intent-driven operations apply to those ports. Finally, generic systems (which are modeled endpoint systems such as servers or external routers represented as "systems" in the blueprint) can be tagged so that downstream intent logic can distinguish system roles and apply the correct operations consistently across expansions and changes.

By contrast, property sets are structured data objects used for variable substitution and probe/configlet parameterization, not a primary target for operational tagging in the blueprint UI; and device profiles are catalog artifacts describing hardware/NOS compatibility rather than blueprint objects typically tagged for intent application.

Verified Juniper sources (URLs):

https://www.juniper.net/documentation/us/en/software/apstra5.1/apstra-user-guide/topics/topic-map/tag-interface-add-remove-datacenter.html

https://www.juniper.net/documentation/us/en/software/apstra5.0/apstra-user-guide/topics/task/tag-virtual-network-update.html

https://www.juniper.net/documentation/us/en/software/apstra5.0/apstra-user-guide/topics/topic-map/tag-system-add-remove-freeform.html

Question: 50

What are three port group roles that you are allowed to assign to a logical device? (Choose three.)

A. Leaf
B. Empty
C. Generic
D. Spine
E. Root

Answer: A, C, D

Explanation:

In Apstra, a logical device abstracts a physical switch's front-panel layout into one or more panels containing port groups. Each port group has a defined speed and one or more roles that describe how those ports are expected to be used in the fabric. These roles are essential because they constrain where ports may be consumed during rack type and template construction (for example, spine-facing vs server-facing vs generic connectivity).

Apstra-supported port group roles include fabric roles such as Spine and Leaf, and endpoint-facing roles such as Generic (commonly used for ports that connect to servers or external generic systems). Assigning Leaf and Spine roles ensures Apstra can correctly validate and render intent for uplinks and interconnects in a three-stage Clos or larger topologies. Assigning Generic indicates ports that can be used for non-fabric connections (such as server links, external routers modeled as generic systems, or other non-managed endpoints).

The options Empty and Root are not valid Apstra port group roles in the logical device model; Apstra uses other explicit role names (for example, Access, Peer, Unused, Generic, Leaf, Spine, Superspine depending on design type and version). In Junos v24.4 EVPN-VXLAN fabrics, getting these roles correct is foundational because Apstra relies on them to place underlay and overlay configuration

onto the right interfaces with predictable results.

Verified Juniper sources (URLs):

https://www.juniper.net/documentation/us/en/software/apstra4.2/apstra-user-guide/topics/concept/logical-devices.html

https://www.juniper.net/documentation/us/en/software/jvd/jvd-collapsed-dc-fabric-juniper-apstra- access-switches/configuration_walkthrough.html

Question: 51

You have an EVPN-VXLAN data center IP fabric, with all single-homed hosts/servers. Which two EVPN route types are present in this scenario? (Choose two.)

A. Type 3
B. Type 7
C. Type 2
D. Type 4

Answer: A, C

Explanation:

In an EVPN-VXLAN fabric where all hosts are single-homed (each endpoint is attached to only one leaf/VTEP), the EVPN control plane still needs to advertise endpoint reachability and enable BUM handling across the overlay. Two EVPN route types are fundamental in this case: Type 2 and Type 3.

EVPN Route Type 2 (MAC/IP Advertisement) is used to advertise learned MAC addresses and, optionally, associated IP addresses for endpoints connected to the local leaf. This enables remote VTEPs to learn where a given host resides (which VTEP to send unicast traffic to) without relying on data-plane flooding for MAC learning. In Junos v24.4 EVPN-VXLAN deployments, Type 2 routes are the core mechanism for distributing endpoint reachability (MAC and MAC+IP bindings) within the EVPN domain.

EVPN Route Type 3 (Inclusive Multicast Ethernet Tag / IMET) is used to establish the flooding scope for BUM traffic in EVPN-VXLAN. In VXLAN fabrics that use ingress replication (common in data centers), Type 3 routes help build the list of remote VTEPs that should receive replicated BUM traffic

for a given segment.

By contrast, Type 4 (Ethernet Segment) routes are associated with EVPN multihoming (ESI-based) and DF election; with only single-homed hosts, Type 4 is not required. Type 7 is not part of the baseline single-homed EVPN-VXLAN host advertisement set in this context.

Verified Juniper sources (URLs):

https://www.juniper.net/documentation/us/en/software/junos/evpn/topics/concept/evpn-bgp-multihoming-overview.html

https://www.juniper.net/documentation/us/en/software/junos/evpn/topics/topic-map/assisted- replication-evpn.html

Question: 52

You are asked to deploy a collapsed fabric architecture. Which two statements are correct about this deployment? (Choose two.)

A. All EVPN-VXLAN overlay functions are provided by the leaf devices.
B. Leaf devices are full-mesh connected.
C. Top-of-rack switches are full-mesh connected.
D. Top-of-rack switches provide VXLAN support.

Answer: A, B

Explanation:

In Apstra, a collapsed fabric (also described as “spineless”) consolidates traditional fabric tiers so that the primary fabric devices perform combined roles. Instead of a dedicated spine tier providing transit between leafs, the fabric is formed by leaf devices connected directly to each other using mesh links. This means a collapsed fabric uses a full-mesh topology at the leaf level, replacing the usual leaf-to- spine connections found in a three-stage Clos. Therefore, the statement that leaf devices are fullmesh connected is correct.

Because the collapsed fabric devices serve the fabric roles, they also provide the EVPN-VXLAN overlay functions (VTEP behavior, EVPN control-plane participation, and VXLAN encapsulation/decapsulation) necessary for tenant segmentation and service delivery. Juniper’s

collapsed fabric validated designs further describe the collapsed fabric switches as serving all fabric roles (including border-leaf behaviors when external connectivity is required), reinforcing that overlay functions reside on these fabric leaf devices.

The remaining statements are not generally true for the collapsed fabric definition. Top-of-rack (access) switches—when present in certain collapsed designs—are not defined by default as fullmesh connected, and VXLAN support is not a requirement for those TOR/access switches unless the specific architecture explicitly uses them as VTEPs. The defining characteristics are the consolidated fabric roles and the leaf-level full-mesh.

Verified Juniper sources (URLs):

https://www.juniper.net/documentation/us/en/software/apstra5.0/apstra-user-guide/topics/concept/templates.html

https://www.juniper.net/documentation/us/en/software/apstra4.2/apstra-user-guide/topics/concept/rack-types.html

https://www.juniper.net/documentation/us/en/software/jvd/jvd-collapsed-dc-fabric-with- apstra/jvd-collapsed-dc-fabric-with-apstra.pdf

Question: 53

You are allowed to assign tags for which three objects? (Choose three.)

A. Virtual networks
B. Interfaces
C. Generic systems
D. Property sets
E. Device profiles

Answer: A, B, C

Explanation:

In Juniper Apstra 5.1, tags are a lightweight metadata mechanism used to classify objects and enable conditional automation (for example, driving dynamic configlets or simplifying filtering/searching in the UI). Apstra supports tagging several blueprint-operational objects that commonly participate in

day-1/day-2 workflows.

Virtual networks can be tagged so operators can group, search, and apply automation consistently across sets of segments. This is useful in EVPN-VXLAN fabrics where virtual networks represent VLAN- or VXLAN-backed broadcast domains and you may want policies or configlet logic to apply to all “finance” or “pci” segments as a group.

Interfaces can be tagged directly within a blueprint (for example, leaf access ports, uplinks, or specific border-facing ports). Interface tags are often used to drive template-based configuration behavior and to simplify operational actions across many ports without relying on fragile naming conventions.

Generic systems (internal or external) can also be tagged. Apstra documentation explicitly describes using tags to specify roles for internal generic systems, enabling you to differentiate server types or attachment roles and then apply the correct intent (connectivity templates, VN attachments, or policies) in a repeatable way.

By contrast, property sets are structured data objects used for parameterization (YAML/JSON values for templates/probes), and device profiles describe hardware/NOS capabilities; they are not the standard blueprint objects for tag assignment in this scenario.

Verified Juniper sources (URLs):

https://www.juniper.net/documentation/us/en/software/apstra5.1/apstra-user- guide/topics/task/tag-virtual-network-update.html

https://www.juniper.net/documentation/us/en/software/apstra5.1/apstra-user-guide/topics/topic- map/tag-interface-add-remove-datacenter.html

https://www.juniper.net/documentation/us/en/software/apstra5.1/apstra-user-guide/topics/topic-map/internal-generic-system-create.html

Question: 54

What is the purpose of a Juniper Apstra rack?

A. It stores information on how pods connect to super spines.

B. It stores information on how leaf nodes connect to generic devices

C. It stores IP address and ASN pool information.

D. It stores device port data rates and vendor information.

Answer: B

Explanation:

A Juniper Apstra rack is a physical entity that contains one or more network devices, such as leaf nodes, access switches, or generic systems. A rack is used to organize and manage the network devices in the Apstra software application. A rack has the following characteristics:

It stores information on how leaf nodes connect to generic devices. This is because a rack can include generic systems, which are devices that are not managed by Juniper Apstra, but are connected to the network. A generic system can be a server, a firewall, a load balancer, or any other device that has a network interface. A rack stores the information on how the leaf nodes, which are the devices that provide access to the end hosts, connect to the generic devices, such as the port number, the link speed, the LAG mode, and the roles1.

It has a rack type, which defines the type and number of leaf devices, access switches, and/or generic systems that are used in the rack. A rack type is a resource that is created in the data center design phase, and it does not specify the vendor or the model of the devices. A rack type can be predefined or custom-made, and it can be used to create multiple racks with the same structure and configuration2.

It has a rack build, which assigns the specific vendor and model of the devices to the rack. A

rack build is created in the staged phase, and it uses the rack type as a template. A rack build can also assign the resources, such as the IP addresses, the ASNs, and the VNIs, to the devices in the rack3.

It has a rack deployment, which applies the network configuration and services to the devices in the rack. A rack deployment is performed in the active phase, and it uses the rack build as a reference. A rack deployment can also monitor the network performance and compliance of the devices in the rack4.

The following three statements are incorrect in this scenario:

It stores information on how pods connect to super spines. This is not true, because a rack does not store any information on the pod or the super spine level of the network. A pod is a cluster of leaf and spine devices that form a 3-stage Clos topology, and a super spine is a device that connects multiple pods in a 5-stage Clos topology. A rack only stores information on the leaf and the access level of the network1.

It stores IP address and ASN pool information. This is not true, because a rack does not store any information on the IP address and ASN pools. IP address and ASN pools are resources that are created in the data center design phase, and they contain a range of IP addresses and ASNs that can be assigned to the devices and the virtual networks. A rack only uses the IP address and ASN pools to assign the resources to the devices in the rack build2.

It stores device port data rates and vendor information. This is not true, because a rack does not store any information on the device port data rates and vendor information. The device port data rates and vendor information are specified in the rack build, which assigns the specific vendor and model of the devices to the rack. A rack only uses the rack build to apply the network configuration and services to the devices in the rack deployment3.

Reference:

Racks (Staged)

Rack Types (Datacenter Design)

Rack Builds (Staged)

Racks (Active)

Question: 55

Within Managed Devices in the Juniper Apstra UI, you notice that several devices have the OOS- Quarantined status. The devices cannot be added to any blueprint. Which action would solve this problem?

A. Acknowledge the device.

B. Fix the hardware issues with the quarantined devices.

C. Install the agent, even though connectivity is established.

D. Upload a new pristine configuration.

Answer: A

Explanation:

When an agent installation is successful, devices are placed into the Out of Service Quarantined (OOS-QUARANTINED) state using the Juniper Apstra UI. This state means that the device is not yet managed by Apstra and has not been assigned to any blueprint. The device configuration at this point is called Pristine Config.

To make the device ready for use in a blueprint, you need to acknowledge the device, which is a manual action that confirms the device identity and ownership. Acknowledging the device changes its status to Out of Service Ready (OOS-

READY)12. Reference:

Managing Devices

AOS Device Configuration Lifecycle

Question: 56

Which attribute enables Juniper Apstra to scale and manage thousands of devices with a single server instance?

A. Apstra is installed as a cloud resource.

B. Apstra is based on NGINX.

C. Apstra is available as an OVA.

D. Apstra is a distributed state system.

Answer: D

Explanation:

The attribute that enables Juniper Apstra to scale and manage thousands of devices with a single server instance is that Apstra is a distributed state system. This means that Apstra uses a graph database to store the network topology and configuration data in a distributed and replicated manner across multiple server nodes. This allows Apstra to handle large-scale networks with high performance, reliability, and availability. Apstra also uses a stateful orchestration engine that ensures the network state is always consistent with the intent of the blueprint, which is the logical representation of the network design and behavior. Apstra can automatically detect and resolve any discrepancies between the desired and actual network state, as well as handle any changes or failures in the network. The other options are incorrect because:

A . Apstra is installed as a cloud resource is wrong because Apstra can be installed either as a cloud resource or as an on-premises resource. Apstra is available as a virtual machine image that can be deployed on various hypervisors, such as VMware ESXi, QEMU/KVM, Microsoft Hyper-V, or Oracle VirtualBox. Apstra can also be deployed on public cloud platforms, such as Amazon Web Services (AWS) or Microsoft Azure. However, the installation method does not affect the scalability of Apstra, which is determined by the distributed state system architecture.

B . Apstra is based on NGINX is wrong because Apstra is not based on NGINX, but on Python and Django. NGINX is a web server and reverse proxy that Apstra uses to serve the web user interface and the REST API. However, NGINX is not the core component of Apstra, and it does not affect the scalability of Apstra, which is determined by the distributed state system architecture.

C . Apstra is available as an OVA is wrong because Apstra is available as an OVF, not an OVA. An OVF (Open Virtualization Format) is a standard format for packaging and distributing virtual machine images. An OVA (Open Virtual Appliance) is a single file that contains the OVF and the virtual disk images. Apstra provides an OVF file that can be imported into various hypervisors, such as VMware ESXi, QEMU/KVM, Microsoft Hyper-V, or Oracle VirtualBox. However, the availability of Apstra as an OVF does not affect the scalability of Apstra, which is determined by the distributed state system architecture. Reference:

JUNIPER APSTRA ARCHITECTURE

Apstra Server Requirements/Reference

Juniper Networks Apstra 4.0 enhances the experience of users and operators

Question: 57

Using Juniper Apstr

a. which component is defined in a template?

A. the leaf-to-spine interconnection

B. the speed of the links between the spine devices and the leaf devices

C. the number of spine devices in a topology

D. the definition of IP pools

Answer: A

Explanation:

According to the Juniper documentation1, a template is a configuration template that defines a network's policy intent and structure. A template can be either rack-based or pod-based, depending on the type and number of racks and pods in the network design. A template includes the following details:

Policies: These are the parameters that apply to the entire network, such as the overlay control protocol, the ASN allocation scheme, and the underlay type.

Structure: This is the physical layout of the network, such as the type and number of racks, pods, spines, and leaves. The structure also defines the leaf-to-spine interconnection, which is the number and type of links between the leaf and spine devices. The leaf-to-spine interconnection can be either single or dual, depending on the redundancy and bandwidth requirements.

Therefore, the correct answer is A. the leaf-to-spine interconnection. This is a component that is defined in a template, as it determines the physical connectivity of the network. The speed of the links, the number of spine devices, and the definition of IP pools are not components that are defined in a template, as they are either derived from the device profiles, the resource pools, or the blueprint settings. Reference: Templates Introduction | Apstra 4.2 | Juniper Networks

Question: 58

You want to make a widget appear on the main dashboard in Juniper Apstr a. In this scenario, which statement is correct?

A. When creating the widget, select the Add to Blueprint Dashboard option.

B. On the blueprint dashboard, click on the Add Widget option.

C. Widgets automatically appear on the blueprint dashboard.

D. Set the Default toggle switch to On for the desired widget.

Answer: D

Explanation:

In Juniper Apstra, a widget is a graphical element that displays data from an intent-based analytics (IBA) probe. A widget can be used to monitor different aspects of the network and raise alerts to any anomalies. A widget can be viewed by itself or added to an analytics dashboard. A dashboard is a collection of widgets that can be customized and organized according to the user's preference1.

The main dashboard in Juniper Apstra is the blueprint dashboard, which is the default view that shows the network information and configuration for the active blueprint. A blueprint is a logical representation of the network design and intent. The blueprint dashboard can display the systemgenerated dashboards, the user-generated dashboards, and the individual widgets that are relevant

to the network2.

To make a widget appear on the main dashboard in Juniper Apstra, the user needs to set the Default toggle switch to On for the desired widget. This will add the widget to the blueprint dashboard, where it can be viewed along with other network information. The user can also remove the widget from the blueprint dashboard by setting the Default toggle switch to Off for the widget3. Therefore, the statement D is correct in this scenario.

The following three statements are incorrect in this scenario:

When creating the widget, select the Add to Blueprint Dashboard option. This is not true, because there is no such option when creating a widget in Juniper Apstra. The user can only select the widget type, the probe, and the display mode when creating a widget4. To add the widget to the blueprint dashboard, the user needs to set the Default toggle switch to On for the widget after creating it3.

On the blueprint dashboard, click on the Add Widget option. This is not true, because there is no such option on the blueprint dashboard in Juniper Apstra. The user can only view, edit, or delete the existing widgets and dashboards on the blueprint dashboard2. To add a widget to the blueprint dashboard, the user needs to set the Default toggle switch to On for the widget from the widgets table view3.

Widgets automatically appear on the blueprint dashboard. This is not true, because widgets do not automatically appear on the blueprint dashboard in Juniper Apstra. The user needs to manually add the widgets to the blueprint dashboard by setting the Default toggle switch to On for the widgets that they want to see on the blueprint dashboard3. The only exception is the widgets that are part of the system-generated dashboards, which are automatically created and added to the blueprint dashboard based on the state of the active blueprint2.

Reference:

Widgets Overview

Blueprint Summaries and Dashboard

Widgets Introduction

Create Widget

Question: 59

What are two system-defined user roles that are available in Juniper Apstra? (Choose two.)

A. authorized

B. root

C. viewer

D. user

Answer: CD

Explanation:

Juniper Apstra provides four system-defined user roles that are available in the Apstra GUI environment. They are: administrator, device_ztp, viewer, and user1. Based on the web search results, we can infer the following statements:

viewer: This role includes permissions to only view various elements in the Apstra system, such as blueprints, devices, design, resources, external systems, platform, and others. Users with this role cannot create, edit, or delete any element12.

user: This role includes permissions to view and edit various elements in the Apstra system, such as blueprints, devices, design, resources, external systems, platform, and others. Users with this role cannot create or delete any element12.

authorized: This is not a system-defined user role in Juniper Apstra. It is a term used to describe users who have been authenticated by an external system, such as LDAP, Active Directory, TACACS+, or RADIUS3.

root: This is not a system-defined user role in Juniper Apstra. It is a term used to describe the superuser account on a Linux system, which has full access to all commands and files. Creating a user in the Apstra GUI does not provide that user access to the Apstra platform via SSH. To access the Apstra platform via SSH, you must create a local Linux system user4. Reference:

User / Role Management Introduction

User/Role Management (Platform)

AAA Providers

User Profile Management

Question: 60

You have a virtual network that needs controlled access to other virtual networks in the same routing zone. Using the Juniper Apstra UI. which feature would be used to accomplish this task?

A. interface policy

B. anti-affinity policy

C. routing policy

D. security policy

Answer: D

Explanation:

A security policy is the feature that would be used to accomplish the task of controlling access to other virtual networks in the same routing zone using the Juniper Apstra UI. A security policy allows you to define rules that specify which traffic is allowed or denied between different virtual networks, IP endpoints, or routing zones. A security policy can be applied to one or more virtual networks in the same routing zone, and it can use various criteria to match the traffic, such as source and destination IP addresses, protocols, ports, or tags. A security policy can also support DHCP relay, which enables the forwarding of DHCP requests from one virtual network to another. The other options are incorrect because:

A . interface policy is wrong because an interface policy is a feature that allows you to configure the interface parameters for the devices in a blueprint, such as interface names, speeds, types, or descriptions. An interface policy does not affect the access control between different virtual networks in the same routing zone.

B . anti-affinity policy is wrong because an anti-affinity policy is a feature that allows you to prevent certain devices or logical devices from being placed in the same rack or leaf pair in a blueprint. An anti-affinity policy is used to enhance the availability and redundancy of the network, not to control the access between different virtual networks in the same routing zone.

C . routing policy is wrong because a routing policy is a feature that allows you to configure the routing parameters for the devices in a blueprint, such as routing protocols, autonomous system numbers, route filters, or route maps. A routing policy does not affect the access control between different virtual

networks in the same routing zone, unless the routing policy is used to filter or modify the routes exchanged between different routing zones. Reference:

Security Policy

Interface Policy

Anti-Affinity Policy

Routing Policy

Question: 61

What is the purpose of an interface map in Juniper Apstra?

A. An interface map associates a logical device with a device profile.

B. An interface map specifies a connection between the interfaces of two devices.

C. An interface map specifies the number of ports and the port speeds of a logical device

D. An interface map specifies the connections between racks in a template.

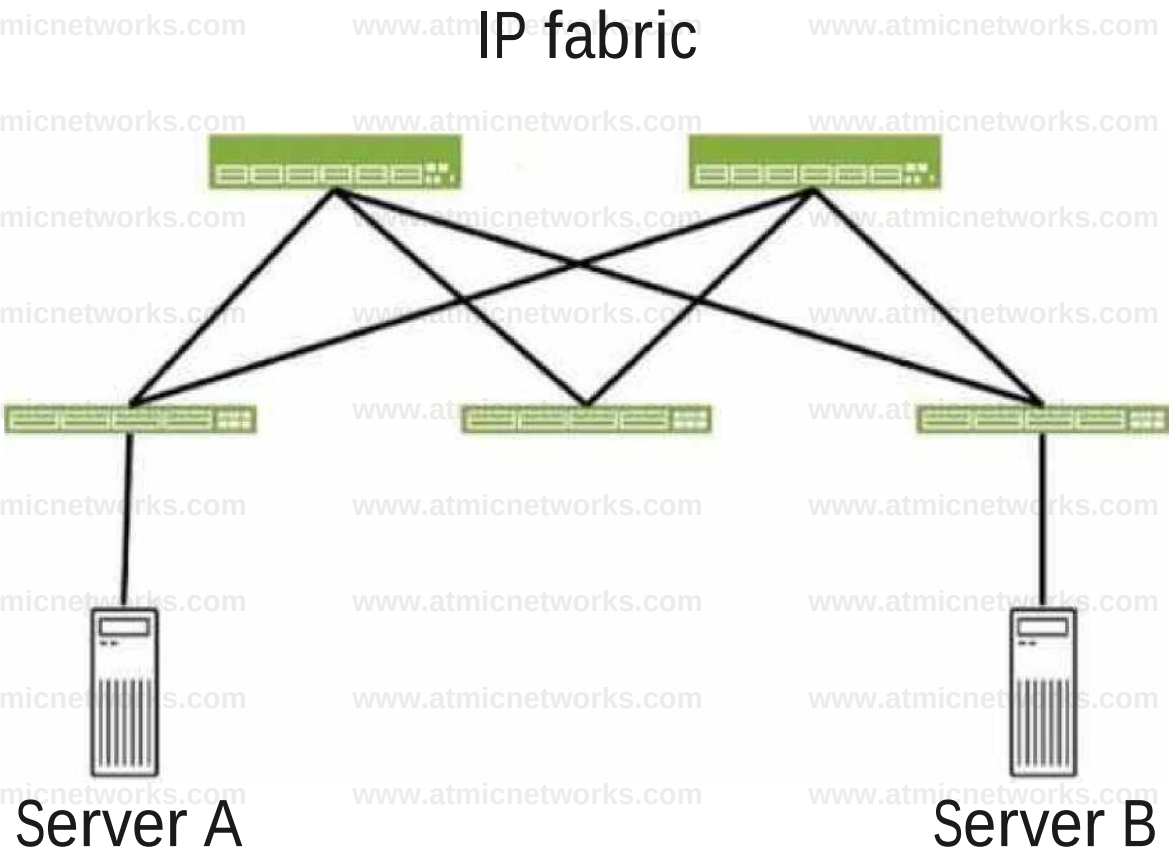
Answer: B

Explanation:

According to the Juniper documentation1, an interface map is a configuration template that maps interfaces between logical devices and physical hardware devices (represented with device profiles) while adhering to vendor specifications. An interface map specifies a connection between the interfaces of two devices, such as a leaf and a spine, a leaf and a server, or a leaf and an external gateway. An interface map can also specify port transformations, such as breaking out a 40 GbE port into four 10 GbE ports, or disabling unused ports. An interface map can be used to achieve the intended network configuration rendering and to enable features such as LAG, ESI-LAG, or MLAG. Therefore, the correct answer is B. An interface map specifies a connection between the interfaces of two devices. Reference: Interface Maps (Datacenter Design)

Question: 62

Exhibit.



Referring to the exhibit, how many broadcast domains will an Ethernet frame pass through when traversing the IP fabric from Server A to Server B?

A. 1

B. 4

C. 2

D. 3

Answer: C

Explanation:

Referring to the exhibit, the image shows a simplified diagram of an IP fabric network connecting two servers, labeled as Server A and Server B. The IP fabric is a network architecture that uses a Clos topology to provide high

bandwidth, low latency, and scalability for data center networks. The IP fabric consists of spine and leaf devices that use BGP as the routing protocol and VXLAN as the overlay technology1.

A broadcast domain is a logical portion of a network where any device can directly transmit broadcast frames to other devices at the data link layer (OSI Layer 2). A broadcast frame is a frame that has a destination MAC address of all ones (FF:FF:FF:FF:FF:FF), which means that it is intended for all devices in the same broadcast domain. A broadcast domain is usually bounded by a router, which does not forward broadcast frames to other networks2.

In the exhibit, there are two broadcast domains that an Ethernet frame will pass through when traversing the IP fabric from Server A to Server B. The first broadcast domain is the one that contains Server A and the leaf device that it is connected to. The second broadcast domain is the one that contains Server B and the leaf device that it is connected to. The IP fabric itself is not a broadcast domain, because it uses IP routing and VXLAN encapsulation to transport the Ethernet frames over the Layer 3 network. Therefore, the statement C is correct in this scenario.

The following three statements are incorrect in this scenario:

A . 1. This is not true, because there are not one, but two broadcast domains that an Ethernet frame will pass through when traversing the IP fabric from Server A to Server B. The IP fabric itself is not a broadcast domain, because it uses IP routing and VXLAN encapsulation to transport the Ethernet frames over the Layer 3 network.

B . 4. This is not true, because there are not four, but two broadcast domains that an Ethernet frame will pass through when traversing the IP fabric from Server A to Server B. The spine devices and the leaf devices that are not connected to the servers are not part of the broadcast domains, because they use IP routing and VXLAN encapsulation to transport the Ethernet frames over the Layer 3 network.

D . 3. This is not true, because there are not three, but two broadcast domains that an Ethernet frame will pass through when traversing the IP fabric from Server A to Server B. The IP fabric itself is not a broadcast domain, because it uses IP routing and VXLAN encapsulation to transport the Ethernet frames over the Layer 3 network.

Reference:

IP Fabric Overview

Broadcast Domain - NetworkLessons.com

Question: 63

Which two actions are required during Juniper Apstra's deploy phase? (Choose two.)

A. Assign device profiles to the blueprint.

B. Assign user roles to the blueprint.

C. Assign interlace maps to the blueprint.

D. Assign resources to the blueprint.

Answer: AD

Explanation:

The deploy phase is the final step in the Juniper Apstra data center fabric design and deployment process. In this phase, you apply the Apstra-rendered configuration to the devices and verify the intent of the blueprint. Based on the web search results, we can infer the following actions are required during the deploy phase12:

Assign device profiles to the blueprint. This action associates a specific vendor model to each logical device in the blueprint. Device profiles contain extensive hardware model details, such as form factor, ASIC, CPU, RAM, ECMP limit, and supported features. Device profiles also define how configuration is generated, how telemetry commands are rendered, and how configuration is deployed on a device. Device profiles enable the Apstra system to render and deploy the configuration according to the Apstra Reference Design34.

Assign resources to the blueprint. This action allocates the physical devices, IP addresses, VLANs, and ASNs to the logical devices, networks, and routing zones in the blueprint. Resources can be assigned manually or automatically by the Apstra system. Assigning resources ensures that the blueprint has all the necessary elements to generate the configuration and deploy the fabric5 .

Assign user roles to the blueprint. This action is not required during the deploy phase. User roles are defined at the system level, not at the blueprint level. User roles determine the permissions and access levels of different users in the Apstra system. User roles can be system-defined or custom- defined .

Assign interface maps to the blueprint. This action is not required during the deploy phase. Interface maps are defined at the design phase, not at the deploy phase. Interface maps are objects that map the logical interfaces of a logical device to the physical interfaces of a device profile. Interface maps enable the Apstra system to generate the correct interface configuration for each device in the fabric . Reference:

Deploy

Deploy Device

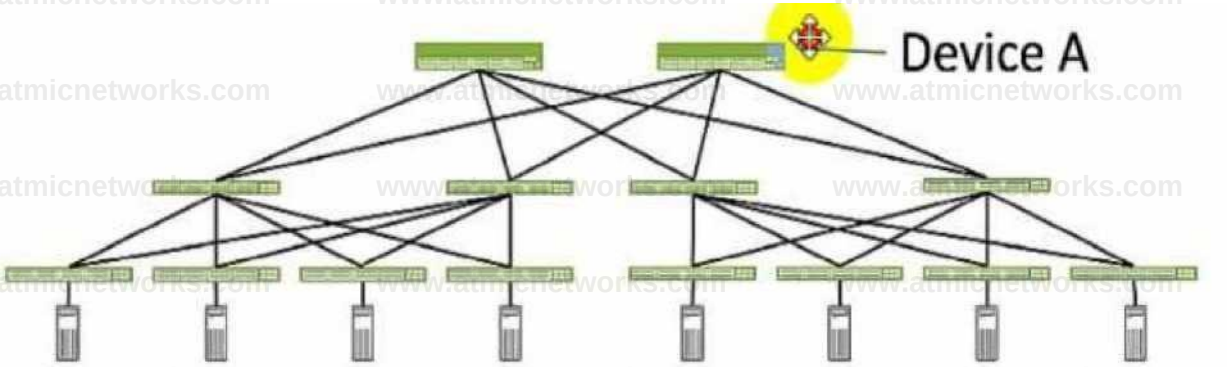
Device Profiles

Juniper Device Profiles

Resources

Question: 64

Exhibit.



Referring to the exhibit, which role does Device A serve in an IP fabric?

A. leaf

B. spine

C. super spine

D. server

Answer: B

Explanation:

Device A serves as a spine in an IP fabric. An IP fabric is a network architecture that uses a spine-leaf topology to provide high performance, scalability, and reliability for data center networks. A spineleaf topology consists of two layers of devices: spine devices and leaf devices. Spine devices are the core devices that interconnect all the leaf devices using equal-cost multipath (ECMP) routing. Leaf devices are the edge devices that connect to the servers, storage, or other network devices. In the exhibit, Device A is connected to four leaf devices using multiple links, which indicates that it is a

spine device. The other options are incorrect because:

A . leaf is wrong because a leaf device is an edge device that connects to the servers, storage, or other network devices. In the exhibit, Device A is not connected to any servers, storage, or other network devices, but only to four leaf devices, which indicates that it is not a leaf device.

C . super spine is wrong because a super spine device is a higher-level device that interconnects multiple spine devices in a large-scale IP fabric. A super spine device is typically used when the number of leaf devices exceeds the port density of a single spine device. In the exhibit, Device A is not connected to any other spine devices, but only to four leaf devices, which indicates that it is not a super spine device.

D . server is wrong because a server device is a compute or storage device that connects to a leaf device in an IP fabric. A server device is typically the end host that provides or consumes data in the network. In the exhibit, Device A is not connected to any leaf devices, but only to four leaf devices, which indicates that it is not a server device. Reference:

IP Fabric Underlay Network Design and Implementation

IP Fabric Overview

IP Fabric Architecture

Question: 65

A member of your organization made changes to a predefined interface map using Juniper Apstra.

Which two statements are correct in this scenario? (Choose two.)

A. Changes to interface maps in the global catalog do not affect interface maps that have already been imported into blueprint catalogs

B. Any changes made to predefined interface maps are discarded when Apstra is upgraded.

C. Changes made to predefined interface maps will not have an impact on the Apstra software.

D. Changes to interface maps in the global catalog will raise anomalies that may need to be addressed at the next commit.

Answer: AB

Explanation:

According to the Juniper documentation1, an interface map is a configuration template that maps interfaces between logical devices and physical hardware devices (represented with device profiles) while adhering to vendor specifications. An interface map can be either predefined or custom. A predefined interface map is one that ships with Apstra software and supports most qualified Juniper devices. A custom interface map is one that

is created by the user to meet specific requirements. An interface map can be stored in either the global catalog or the blueprint catalog. The global catalog contains all the interface maps that are available for use in any blueprint. The blueprint catalog contains the interface maps that are imported from the global catalog and used in a specific blueprint.

When a member of your organization makes changes to a predefined interface map, the following statements are correct:

Changes to interface maps in the global catalog do not affect interface maps that have already been imported into blueprint catalogs. This means that the existing blueprints that use the original version of the interface map will not be impacted by the changes. However, if you want to use the updated version of the interface map in a new or existing blueprint, you need to import it again from the global catalog.

Any changes made to predefined interface maps are discarded when Apstra is upgraded. This means that the changes will not be preserved across different versions of Apstra software. If you want to retain a customized interface map through Apstra upgrades, you need to clone the predefined interface map, give it a unique name, and customize it instead of changing the predefined one directly.

Therefore, the correct answer is A and B. Changes to interface maps in the global catalog do not affect interface maps that have already been imported into blueprint catalogs and any changes made to predefined interface maps are discarded when Apstra is upgraded. Reference: Edit Interface Map | Apstra 4.2 | Juniper Networks