PassQueen
a.k.a dumpsqueen.com

Deploy and Manage Citrix ADC with Traffic Management

Citrix 1Y0-241

Version Demo

Total Demo Questions: 15

Total Premium Questions: 153

Buy Premium PDF

https://passqueen.com

support@passqueen.com

Topic Break Down

Topic	No. of Questions
Topic 1, Advanced Policies	60
Topic 2, Content Switching	15
Topic 3, Global Server Load Balancing	13
Topic 4, Optimization	24
Topic 5, Troubleshooting	28
Topic 6, Mix Questions	13
Total	153

QUESTION NO: 1

A Citrix Administrator has configured SSL services behind an SSL load-balancing vServer. The administrator needs Citrix ADC to validate the certificate presented by each backend server.

Which two configurations are required? (Choose two.)

A. Enable server authentication on the SSL service or service group.

B. Bind the issuing CA certificate to the SSL service or service group.

C. Enable client authentication on the front-end SSL vServer.

D. Bind the public server certificate only to the front-end SSL vServer.

E. Configure SSL bridging for the backend services.

ANSWER: A B

Explanation:

Server authentication must be enabled on the SSL service or service group. This enables Citrix ADC to validate the certificate presented by the backend server when the ADC establishes the server-side TLS connection.

The CA certificate that issued the backend server certificate must be bound to the SSL service or service group. Citrix ADC uses this trusted CA certificate to build and validate the backend certificate chain. The backend server certificate must also contain a valid identity for the server being contacted, but binding the issuing CA and enabling server authentication are the required ADC-side configurations.

Binding the public-facing server certificate to the front-end SSL vServer only controls the certificate presented to clients. Enabling client authentication validates certificates presented by clients and does not validate backend server certificates. See the Citrix ADC documentation on SSL backend server authentication.

QUESTION NO: 2

A Citrix Administrator notices that the Citrix ADC is sending the IP addresses of all the active services in the DNS response.

The administrator can use the set gslb vServer _____ parameter to change this behavior.

A. EDR ENABLED

B. MIR DISABLED

C. MIR ENABLED

D. EDR DISABLED

ANSWER: D

Explanation:

EDR DISABLED is correct because EDR, or Extended DNS Response, controls whether a Citrix ADC GSLB virtual server includes the addresses of all eligible active GSLB services in its DNS responses. When EDR is enabled, the appliance can return multiple service IP addresses for a DNS query, allowing the client resolver to receive the full set of currently available endpoints. Disabling EDR changes that behavior so the GSLB virtual server returns only the IP address selected according to its configured GSLB load-balancing method, rather than listing every active service address.

The setting is configured on the GSLB virtual server with the `-EDR` parameter, for example: `set gslb vserver <name> -EDR DISABLED`. This is useful when DNS clients should receive a single, ADC-selected destination and the administrator does not want endpoint selection to be deferred to the client after receiving multiple records. Citrix documents EDR as a GSLB virtual-server setting and describes its role in controlling whether all service IP addresses are included in a DNS reply. See the Citrix ADC GSLB virtual server configuration documentation and the Citrix ADC gslb vserver command reference.

QUESTION NO: 3

A Citrix Administrator needs to export HTTP transaction records from a Citrix ADC to an AppFlow collector.

Which two configurations are required? (Choose two.)

A. Create an AppFlow action that specifies the collector.

B. Bind an AppFlow policy to the applicable vServer.

C. Bind a rewrite policy globally.

D. Create a URL transformation action.

E. Enable load-balancing spillover.

ANSWER: A B

Explanation:

An **AppFlow action** identifies the AppFlow collector to which Citrix ADC exports records. The action includes the collector destination and the export configuration needed for the appliance to send AppFlow data.

An **AppFlow policy bound to the applicable vServer** is also required. The policy determines the traffic for which AppFlow records are generated, and the virtual-server binding applies that policy to the relevant application traffic. Together, the policy and action allow Citrix ADC to select transactions and export their records to the configured collector.

Citrix ADC AppFlow can be used with Citrix ADM and other compatible collectors for application-performance and traffic analytics. See Citrix ADC AppFlow documentation.

QUESTION NO: 4

The Citrix ADC SDX architecture allows instances to share _____ and _____. (Choose the two correct options to complete the sentence.)

A. memory

B. a CPU

C. a physical interface

D. the kernel

ANSWER: B C

Explanation:

Citrix ADC SDX is a multi-instance platform that hosts multiple isolated Citrix ADC VPX instances on common SDX hardware. This architecture permits instances to use the same underlying physical CPU hardware while the SDX platform allocates processing resources to each instance. Resource allocation is managed through the SDX management plane, allowing administrators to assign and adjust CPU capacity according to the requirements of each VPX instance while consolidating services on one appliance.

Instances can also share a physical interface. SDX interfaces are physical ports on the appliance, and multiple VPX instances can use them through appropriate network configuration, such as VLAN separation and interface assignment. This enables traffic for independent instances to traverse common physical network connectivity without requiring a dedicated appliance or dedicated physical NIC for every instance. The combination of shared compute hardware and shared physical interfaces is a core benefit of SDX consolidation, while individual VPX instances remain separately managed Citrix ADC environments. Citrix describes SDX as a platform for provisioning and managing multiple VPX instances on an SDX appliance; see Deploy Citrix ADC VPX instances on SDX and Citrix ADC SDX hardware platforms.

QUESTION NO: 5

A Citrix Administrator needs to block all DNS requests from subnet 10.107.149.0/24.

Which expressions can the administrator use to match the required traffic?

A. CLIENT.IP.SRC(10.107.149.0) && (client.UDP.DSTPORT.EQ(53) || client TCP DSTPORT.EQ(53))

B. CLIENT.IP.SRC IN_SUBNET(10 107.149.0/24) && client.UDP.DSTPORT.EQ(53) || client.TCP.DSTPORT.EQ(53)

C. CLIENT.IP.SRC(10.107.149.0) && client UDP.DSTPORT.EQ(53) || client.TCP.DSTPORT.EQ(53)

D. CLIENT.IP.SRC.IN_SUBNET(10.107.149.0/24) && (CLIENT.UDP.DSTPORT.EQ(53) || CLIENT.TCP.DSTPORT.EQ(53))

ANSWER: D

Explanation:

`CLIENT.IP.SRC.IN_SUBNET(10.107.149.0/24) && (CLIENT.UDP.DSTPORT.EQ(53) || CLIENT.TCP.DSTPORT.EQ(53))` correctly identifies the required DNS traffic because it combines a source-network test with destination-port tests for both DNS transport protocols. `CLIENT.IP.SRC.IN_SUBNET(10.107.149.0/24)` evaluates whether the client source address belongs to the complete 10.107.149.0/24 network, covering addresses from 10.107.149.0 through 10.107.149.255. DNS commonly uses UDP destination port 53 for ordinary queries and responses, but it can use TCP destination port 53 for such cases as zone transfers and responses that require reliable or larger-message transport. The parenthesized UDP-or-TCP condition therefore matches DNS requests using either supported transport. The outer logical AND is essential to scope both port checks to clients in the specified subnet, allowing a responder or firewall policy to block precisely the intended traffic. Citrix ADC advanced policy expressions support IP-address operators such as `IN_SUBNET` and protocol-specific fields such as UDP and TCP destination ports. See the Citrix ADC policies and expressions documentation and the DNS specification in RFC 1035.

QUESTION NO: 6

Which two persistence types can a Citrix Administrator configure on a load-balancing vServer? (Choose two.)

A. SOURCEIP

B. COOKIEINSERT

C. DESTINATIONIP

D. ROUNDROBIN

E. LEASTBANDWIDTH

ANSWER: A B

Explanation:

SOURCEIP is a supported persistence type. Citrix ADC records the client source IP address and uses that persistence entry to direct subsequent requests from the same client address to the selected backend service for the configured timeout period.

COOKIEINSERT is also a supported persistence type for HTTP traffic. Citrix ADC inserts a persistence cookie into the server response and uses the cookie returned by the client in later requests to select the same backend service. This avoids relying solely on the client IP address, which can be shared by many users behind a proxy or NAT device.

Persistence is configured on the load-balancing vServer and must be selected according to the application protocol and session-management requirements. See Citrix ADC persistence documentation.

QUESTION NO: 7

A Citrix Administrator needs to configure a rate-limiting policy setting DNS requests to a threshold of 1,000 per second.

Which set of commands does the administrator need to run to correctly configure and enable this policy?

A. > add stream selector DNSSelector1 client.udp.dns.domain
> add ns limitIdentifier DNSLimitIdentifier1 -threshold 5 -timeSlice 1000 -selectorName DNSSelector1
> add dns policy DNSLimitPolicy1 " sys.check_limit(\"DNSLimitIdentifier1\")" -preferredLocation "North America.US.TX.Dallas.. "
> bind dns global DNSLimitPolicy1 5

B. > add stream selector DNSSelector1 client.udp.dns.domain
> add ns limitIdentifier DNSLimitIdentifier1 -threshold 1000 -timeSlice 1000 -selectorName DNSSelector1
> add dns policy DNSLimitPolicy1 " sys check_limit(\"DNSLimitIdentifier1\")" -preferredLocation "North America.US.TX.Dallas.. "
> bind dns global DNSLimitPolicy1 5

C. > add stream selector DNSSelector1 client.udp.dns.domain
> add ns limitIdentifier DNSLimitIdentifier1 -threshold 5 -timeSlice 1000 -selectorName DNSSelector1
> add dns policy DNSLimitPolicy1 " sys.check_limit(\"DNSLimitIdentifier1\")" -preferredLocation "North America.US.TX.Dallas.. "

D. > add stream selector DNSSelector1 client.udp.dns.domain
> add ns limitIdentifier DNSLimitIdentifier1 -threshold 1000 -timeSlice 1000 -selectorName DNSSelector1
> add dns policy DNSLimitPolicy1 " sys check_limit(\"DNSLimitIdentifier1\")" -preferredLocation "North America.US.TX.Dallas.. "

E. > add stream selector DNSSelector1 client.udp.dns.domain
> add ns limitIdentifier DNSLimitIdentifier1 -threshold 1000 -timeSlice 1000 -selectorName DNSSelector1
> add dns policy DNSLimitPolicy1 "SYS.CHECK_LIMIT(\"DNSLimitIdentifier1\")" -preferredLocation "North America.US.TX.Dallas.."
> bind dns global DNSLimitPolicy1 5

ANSWER: E

Explanation:

The command set that creates `DNSLimitIdentifier1` with a threshold of `1000`, a `1000`-millisecond time slice, the `SYS.CHECK_LIMIT` policy expression, and a global DNS-policy binding correctly implements the requested limit. In Citrix ADC, a limit identifier defines the maximum number of matching requests permitted during its configured time slice. Therefore, a threshold of 1,000 over 1,000 milliseconds establishes a rate of 1,000 DNS requests per second. The stream selector `client.udp.dns.domain` supplies the request attribute used to identify the traffic being counted, allowing the ADC to apply the defined limiting logic to DNS requests based on the selected DNS domain value. The DNS policy calls `SYS.CHECK_LIMIT("DNSLimitIdentifier1")`, which evaluates the configured limit identifier for each relevant request. Finally, binding the policy globally is essential: creating a selector, limit identifier, and policy alone does not activate processing. The global bind with priority 5 causes the DNS feature to evaluate this rate-limit policy. Citrix documents DNS rate limiting as a combination of a stream selector, an `ns limitIdentifier`, a DNS policy using `SYS.CHECK_LIMIT`, and a DNS global binding. See Citrix DNS rate limiting documentation and Citrix ADC limit identifier command reference.

QUESTION NO: 8

Scenario: In general, it is recommended to do the following:

Use _____ if you want the Citrix ADC to reset or drop a connection based on a client or request-based parameter.

Use _____ to redirect traffic or respond with custom messages.

Use _____ for manipulating data on HTTP requests and responses.

A. rewrite, rewrite, responder

B. responder, responder, rewrite

C. rewrite, responder, rewrite

D. responder, redirect, rewrite

ANSWER: B

Explanation:

responder, responder, rewrite is correct because Citrix ADC Responder policies are designed to evaluate traffic against policy expressions and take an immediate action when the expression evaluates as true. These actions include dropping a connection, resetting it, redirecting a client to another URL, and sending a custom HTTP response. This makes the Responder feature the appropriate policy engine when the intended outcome is to control the request flow or return a direct response based on client attributes or request content.

Citrix ADC Rewrite policies serve a different but complementary purpose: they modify HTTP protocol data while traffic continues through the appliance. Rewrite actions can insert, delete, replace, or otherwise alter HTTP headers, URLs, cookies, and other request or response content. Therefore, Rewrite is the recommended feature when the requirement is specifically to manipulate data carried in HTTP requests and responses rather than to terminate, redirect, or directly answer a request.

Citrix documents Responder actions and their policy-based processing behavior in its Responder documentation, and describes HTTP content modification in its Rewrite documentation.

QUESTION NO: 9

A Citrix Administrator needs to configure an HTTP-ECV monitor to verify that a web application returns an expected status value.

Which two monitor parameters should the administrator configure? (Choose two.)

A. Send string

B. Receive string

C. Load-balancing method

D. Persistence type

E. Spillover threshold

ANSWER: A B

Explanation:

The **send string** specifies the HTTP request that Citrix ADC sends to the monitored service. For example, the administrator can configure a request for a dedicated health-check URI that is implemented by the application.

The **receive string** specifies text that Citrix ADC expects to find in the HTTP response. The monitor marks the service as healthy only when the service responds as expected and the configured receive string is found. This lets the administrator validate application content rather than only confirming that a TCP port accepts connections. See Citrix ADC load-balancing monitors documentation.

QUESTION NO: 10

A Citrix Administrator needs to configure a Citrix ADC high availability (HA) pair with each Citrix ADC in a different subnet.

What does the administrator need to do for HA to work in different subnets?

A. Configure SyncVlan

B. Turn on fail-safe mode.

C. Turn on HA monitoring on all interfaces

D. Turn on Independent Network Configuration (INC) mode.

ANSWER: D

Explanation:

Turn on Independent Network Configuration (INC) mode. INC mode is specifically designed for a Citrix ADC HA deployment in which the two appliances require independent network settings, including placement in different subnets. When INC is enabled, each node can retain its own network configuration while participating in the HA pair. This is important where the primary and secondary appliances are deployed in separate network segments, data centers, or availability zones and cannot use an identical network configuration.

HA still provides node-state monitoring, configuration synchronization for shared configuration, and failover behavior. The administrator should ensure that the nodes have the required Layer 3 connectivity for HA communication and that the relevant routes and network reachability are configured on each appliance. INC changes the HA networking model so that node-specific settings can remain local rather than requiring the normal shared network arrangement. Citrix documents Independent Network Configuration as the HA capability for deployments where appliances use separate network configurations. See the Citrix guidance on Citrix ADC high availability and the HA node configuration documentation.

QUESTION NO: 11

Scenario: A Citrix Administrator needs to create local, limited-privilege user accounts for other administrators. The other administrators will require only:

Read-only access

The ability to enable and disable services and servers

Which built-in command policy permission level can the administrator use?

A. Operator

B. Network

C. Sysadmin

D. Read-only

ANSWER: A

Explanation:

Operator is the appropriate built-in Citrix ADC command-policy permission level. It is designed for administrators who need to monitor appliance configuration and operational status while performing limited day-to-day operational actions. In addition to permitting show commands for read-only visibility, the operator policy permits enable and disable commands. This directly supports the stated requirement to enable or disable services and servers without granting broad configuration authority.

Citrix ADC command policies provide role-based command authorization for local and external users. Assigning the built-in operator policy to a local user account applies a predefined, limited command set rather than requiring a custom command policy. This follows least-privilege practice: the delegated administrators receive enough access to manage service and server availability, while more sensitive configuration and administrative capabilities remain reserved for higher-privilege roles.

Citrix documents the built-in command policies and explains that the operator policy permits show, enable, and disable operations. See Citrix ADC authorization configuration documentation and Citrix ADC default command policies documentation.

QUESTION NO: 12

Scenario: A Citrix Administrator manages an environment that has a Citrix ADC high availability (HA) pair running on two MPX appliances. The administrator notices that the state of the secondary Citrix ADC is ‘Unknown’.

What is causing the secondary state to be 'Unknown'?

A. The synchronization on the secondary appliance is disabled.

B. TCP port 22 is disabled between the primary and secondary ADCs.

C. The administrator made both Citrix ADCs primary.

D. The remote procedure call (RPC) nodes are incorrectly configured.

ANSWER: D

Explanation:

The remote procedure call (RPC) nodes are incorrectly configured. Citrix ADC HA relies on properly configured RPC-node entries so that each appliance can identify and communicate with its peer for HA control-plane operations. The RPC node configuration associates the peer appliance's NSIP address with the expected node identity. If that information is absent, points to the wrong NSIP address, or does not correspond correctly between the two appliances, the primary appliance cannot reliably obtain the peer's HA status. As a result, the peer can be displayed with an *Unknown* state rather than a normal secondary status.

For an HA pair, the appliances must have network reachability over their NSIPs, appropriate HA node definitions, and matching peer information. Administrators should verify the configured HA node and RPC-node details on both appliances, confirm that each entry uses the peer's correct NSIP address, and then restore or validate HA communication. Citrix documents HA setup and status verification, including the role of node communication, in its High Availability documentation and the RPC nodes documentation.

QUESTION NO: 13

Scenario: A Citrix Administrator is configuring a Citrix ADC high availability (HA) pair with an existing primary Citrix ADC with all resources configured. The administrator adds the secondary Citrix ADC in HA and discovers that the configuration on the existing primary was removed and is now the secondary Citrix ADC in the HA pair.

Which two configurations could the administrator have used to prevent this from happening? (Choose two.)

A. Set the primary Citrix ADC to stay primary in the Configure HA Node settings.

B. Set the secondary Citrix ADC to stay secondary in the Configure HA Node settings.

C. Enable HA monitoring on all secondary device interfaces.

D. Enable HA monitoring on all primary device interfaces.

ANSWER: A B

Explanation:

Set the primary Citrix ADC to stay primary in the Configure HA Node settings and set the secondary Citrix ADC to stay secondary in the Configure HA Node settings. These HA node status settings explicitly establish the intended roles during formation of the HA pair. Configuring the appliance that already contains the required production configuration as *STAYPRIMARY* ensures it retains the primary role. Configuring the newly introduced appliance as *STAYSECONDARY* prevents it from becoming primary during the initial role negotiation. As a result, the established primary node is the authoritative source for HA configuration synchronization, preserving its existing configuration and propagating that configuration to its peer. This is particularly important when introducing an appliance that might contain an older, default, or otherwise undesired configuration. Citrix ADC documents the HA node status controls, including STAYPRIMARY and STAYSECONDARY, in its HA node command reference: Citrix ADC HA node command reference. For the broader HA implementation and synchronization model, see Citrix ADC high availability documentation.

QUESTION NO: 14

To protect an environment against Hash DoS attacks, which two configurations can a Citrix Administrator use to block all post requests that are larger than 10,000 bytes? (Choose two.)

A. > add policy expression expr_hashdos_prevention “http.REQ.METHOD.EQ(\”POST\”)&& http.REQ.CONTENT_LENGTH.GT(10000)”
> add rewrite policy drop_rewrite expr_hashdos_prevention DROP
> bind rewrite global drop_rewrite 100 END –type REQ_OVERRIDE

B. > add policy expression expr_hashdos_prevention “http.REQ.METHOD.EQ(\”POST\”)&& http.REQ.CONTENT_LENGTH.GT(10000)”
> add responder policy pol_resp_hashdos_prevention expr_hashdos_prevention DROP NOOP
> bind responder global pol_resp_hashdos_prevention 70 END –type REQ_OVERRIDE

C. > add policy expression expr_hashdos_prevention “http.REQ.METHOD.EQ(\”POST\”) || http.REQ.CONTENT_LENGTH.GT(10000)”
> add responder policy pol_resp_hashdos_prevention expr_hashdos_prevention DROP NOOP
> bind responder global pol_resp_hashdos_prevention 70 END –type REQ_OVERRIDE

D. > add policy expression expr_hashdos_prevention “http.REQ.METHOD.EQ(\”POST\”) || http.REQ.CONTENT_LENGTH.GT(10000)”
> add rewrite policy drop_rewrite expr_hashdos_prevention DROP
> bind rewrite global drop_rewrite 70 END –type REQ_OVERRIDE

E. > add policy expression expr_hashdos_prevention “http.REQ.METHOD.EQ(\”POST\”) || http.REQ.CONTENT_LENGTH.GT(10000)”
> add responder policy pol_resp_hashdos_prevention expr_hashdos_prevention DROP NOOP
> bind responder global pol_resp_hashdos_prevention 100 END –type REQ_OVERRIDE

F. > add policy expression expr_hashdos_prevention “http.REQ.METHOD.EQ(\”POST\”) || http.REQ.CONTENT_LENGTH.GT(10000)”
> add rewrite policy drop_rewrite expr_hashdos_prevention DROP
> bind rewrite global drop_rewrite 100 END –type REQ_OVERRIDE

ANSWER: A B

Explanation:

Both configurations correctly identify only requests that meet both required conditions: the HTTP method is `POST` and the declared request content length is greater than 10,000 bytes. The `&&` operator is essential because it requires the method and size tests to be true together. `HTTP.REQ.CONTENT_LENGTH.GT(10000)` evaluates the Content-Length value numerically, enabling the ADC to enforce the specified threshold before the request reaches the protected application.

The configuration using a rewrite policy applies the built-in `DROP` action to matching requests and binds that policy globally in the request-override processing point. The configuration using a responder policy likewise evaluates the same named expression and uses the responder `DROP` action, also at global request-override scope. In either case, `END` terminates further policy evaluation after the matching request has been dropped. This provides a lightweight edge control that limits oversized POST submissions associated with Hash DoS exposure. Citrix documents request-policy processing and rewrite actions in its Rewrite documentation and describes request matching and response actions in its Responder documentation.

QUESTION NO: 15

In a global server load balancing (GSLB) active-active environment, the connection proxy is used as the site persistence method.

What is used to source the traffic when the connection is proxied?

A. Subnet IP (SNIP)

B. LDNS IP Address

C. Client source IP

D. Virtual IP (VIP)

ANSWER: A

Explanation:

Subnet IP (SNIP) is used as the source address for traffic that the Citrix ADC sends when it connection-proxies a client request to a service at the selected GSLB site. Connection proxy allows the ADC receiving the client connection to proxy that connection to the chosen site, which is particularly useful when the client must continue using the same entry point even though GSLB selects a service in another location. For the server-side leg of this proxied flow, the ADC originates the connection by using its SNIP address. The SNIP is the appliance address intended for ADC-originated communication with backend servers and other network resources, and it provides a routable return address for the proxied server-side connection. This behavior means that, unless a separate feature or configuration explicitly preserves client source addressing, the backend service sees the ADC's SNIP as the source of the proxied traffic. Citrix documents connection proxy as a GSLB site-persistence mechanism and documents SNIPs as the addresses used by the ADC for connections it initiates to servers. See Citrix ADC Global Server Load Balancing documentation and Citrix ADC IP addressing documentation.