DUMPSBOSS.

Fortinet NSE 5 - FortiManager 7.6 Administrator

Fortinet FCP_FMG_AD-7.6

Version Demo

Total Demo Questions: 10

Total Premium Questions: 62

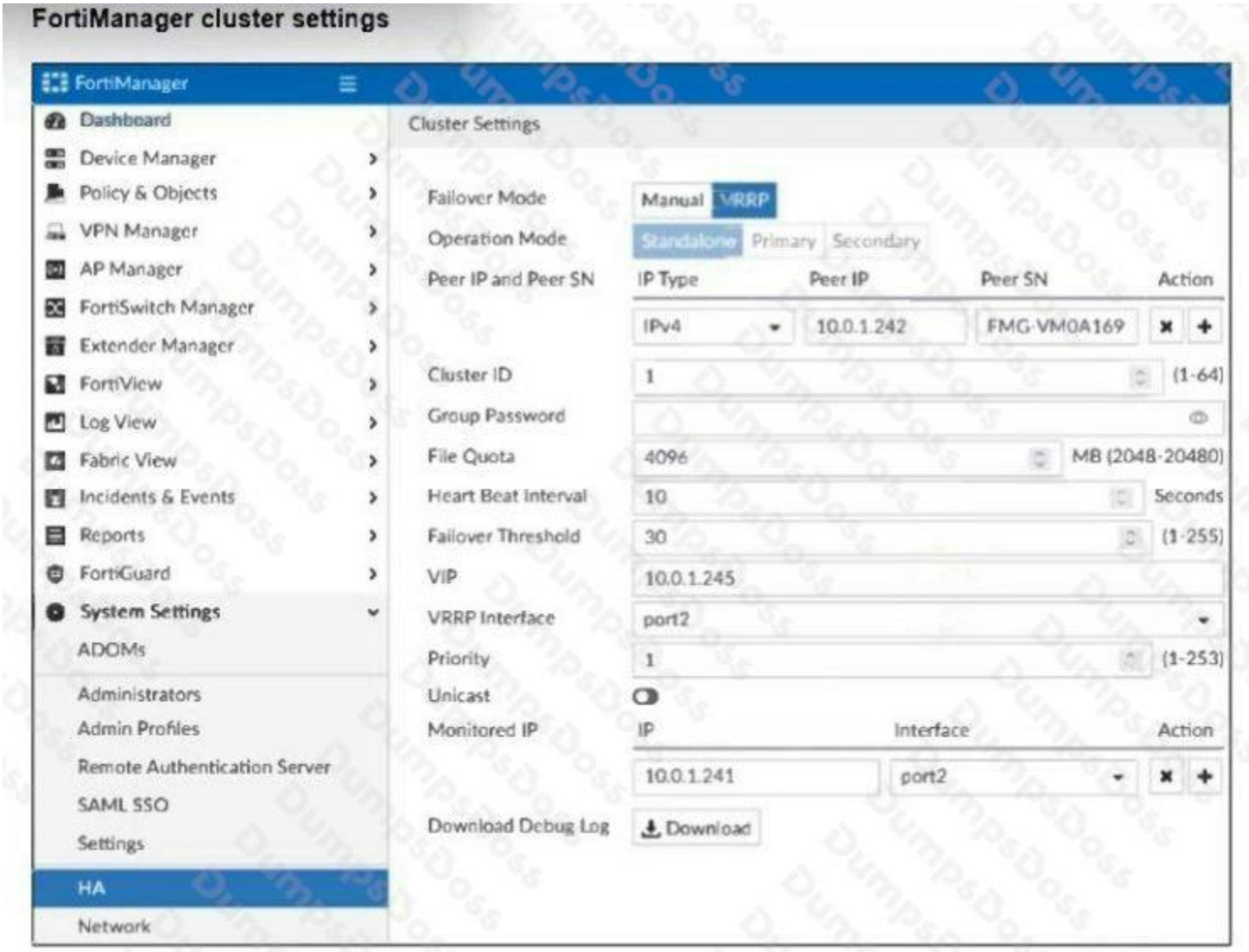
Buy Premium PDF

https://dumpsboss.co

support@dumpsboss.co

support@dumpsboss.co
dumpsboss.co

QUESTION NO: 1



If the monitored interface for the primary FortiManager device fails, what must you do to maintain high availability (HA)?

A. The FortiManager HA failover is transparent to administrators and does not require any additional action.

B. Manually promote one of the working secondary devices to the primary role: and reboot the original primary device to remove the peer IP address of the failed device.

C. Reconfigure the primary device to remove the peer IP address of the failed device from its configuration.

D. Check the integrity database of the primary device to force a secondary device to become the new primary with all active interfaces.

ANSWER: B

Explanation:

On FortiManager HA, a monitored interface going down on the primary can trigger a failover, but it isn't always completely "set and forget." If the primary is still up but its monitored link is failed, you typically need to make sure a healthy unit takes over as the active (primary) so management access and services stay available.

That's why the best answer is to manually promote a working secondary to primary. This ensures the cluster has a clear active node with working interfaces. Rebooting the old primary is a common practical step to clear the stale HA state/peer info and let it rejoin cleanly as a secondary, instead of fighting for the role or keeping bad interface status around.

The other choices don't really match how FortiManager HA is operated: failover isn't always fully transparent in every "interface-only failure" scenario, and you don't normally fix this by editing peer IPs on the primary or by "checking database integrity" to force roles.

References: https://docs.fortinet.com/document/fortimanager/7.6.0/administration-guide and https://community.fortinet.com/

QUESTION NO: 2

Company policy dictates that any time a change is made to a policy package on FortiManager an ADOM revision is created before the change installed, and that revision is held for a minimum of 90 days.

Over the past three months, each installed change has resulted in several unused policies and duplicate objects. The FortiManager administrator plans to upgrade the FortiGate devices and then upgrade the FortiManager ADOM from version 7.4 to 7.6.

Which action can the administrator take to avoid slow ADOM upgrades?

A. Check and repair the global configuration database before upgrading.

B. Export firewall policies to Excel, delete them on the ADOM, then reimport them after upgrading the ADOM.

C. Find unused firmware templates, then delete them before upgrading.

D. Limit ADOM revisions before upgrading.

ANSWER: D

Explanation:

ADOM upgrades can get painfully slow when FortiManager has a lot of historical data to carry forward—especially ADOM revisions. Each revision is basically a snapshot of the ADOM database, so if you've been creating one for every install (and keeping them for 90+ days), you're stacking up a lot of extra data that FortiManager has to process during the schema conversion.

So the practical move is to limit (or clean up) ADOM revisions before you start the ADOM upgrade. That reduces the amount of revision history FortiManager needs to migrate and speeds up the upgrade process. It also lowers the chance of hitting performance issues during the conversion.

The unused policies and duplicate objects are messy, but they're not the main thing that slows the ADOM upgrade itself. Cleaning revisions is the most direct way to make the upgrade run faster while still keeping whatever minimum history your policy requires.

References: https://docs.fortinet.com/document/fortimanager/7.6.0/administration-guide and https://docs.fortinet.com/document/fortimanager/7.6.0/release-notes

QUESTION NO: 3

The administrator uses FortiManager to push a CLI script using the Remote FortiGate Directly (via CLI) option to configure an IPsec VP

N. However, when running the script, the administrator receives the following error: config vpn ipsec phase2-interface [parameter(s) invalid. detail: object mismatch]

What must the administrator do to resolve the script error and successfully apply the IPsec configuration?

A. Add the end command after finishing the IPsec phase 1-interface configuration block.

B. Use IPsec templates to deploy provisioning templates.

C. Add a second config vpn ipsec phase2-interface block without linking it to phase1.

D. Run the script using the policy package or ADOM database method.

E. However, when running the script, the administrator receives the following error: config vpn ipsec phase2-interface [parameter(s) invalid. detail: object mismatch]
What must the administrator do to resolve the script error and successfully apply the IPsec configuration?

ANSWER: A

Explanation:

That “object mismatch” error usually happens when the FortiGate is still inside the Phase1 configuration context, but the script jumps straight into Phase2 commands. When you run a script “Remote FortiGate Directly (via CLI)”, FortiManager basically pastes the commands into the FortiGate CLI, so the CLI context matters a lot.

To fix it, you need to properly close the Phase1 block with `end` (or at least exit back to the top level) before starting `config vpn ipsec phase2-interface`. Once the Phase1 section is correctly ended, the Phase2 section can be created and linked to the Phase1 name without the parser thinking you’re still editing the wrong object.

You can review the CLI structure for IPsec Phase1/Phase2 and how the CLI config mode nesting works in the FortiOS documentation here: https://docs.fortinet.com/document/fortigate/7.6.0/administration-guide.

QUESTION NO: 4

Refer to the exhibit.

```
FortiManager # diagnose fmupdate view-serverlist fds
Fortiguard Server Comm : Enabled
Server Override Mode   : Strict
FDS server list        :
Index   Address                     Port         TimeZone       Distance       Source
-------------------------------------------------------------------------------------
*0      10.0.1.50                   8890         -5             0              CLI
 1      96.45.33.89                 443          -5             0              FDNI
 2      96.45.32.81                 443          -5             0              FDNI
 ...
 9      fds1.fortinet.com           443          -5             0              DEFAULT
```

How does FortiManager get antivirus and IPS updates?

A. It uses all URLs in the list that contain the fds host name.

B. It gets updates from the server with IP address 10.0.1.50.

C. It connects to all servers marked as FortiGuard Distribution Network through Internet (FDNI) sources.

D. It connects to the public FortiGuard servers listed in the configuration.

ANSWER: C

Explanation:

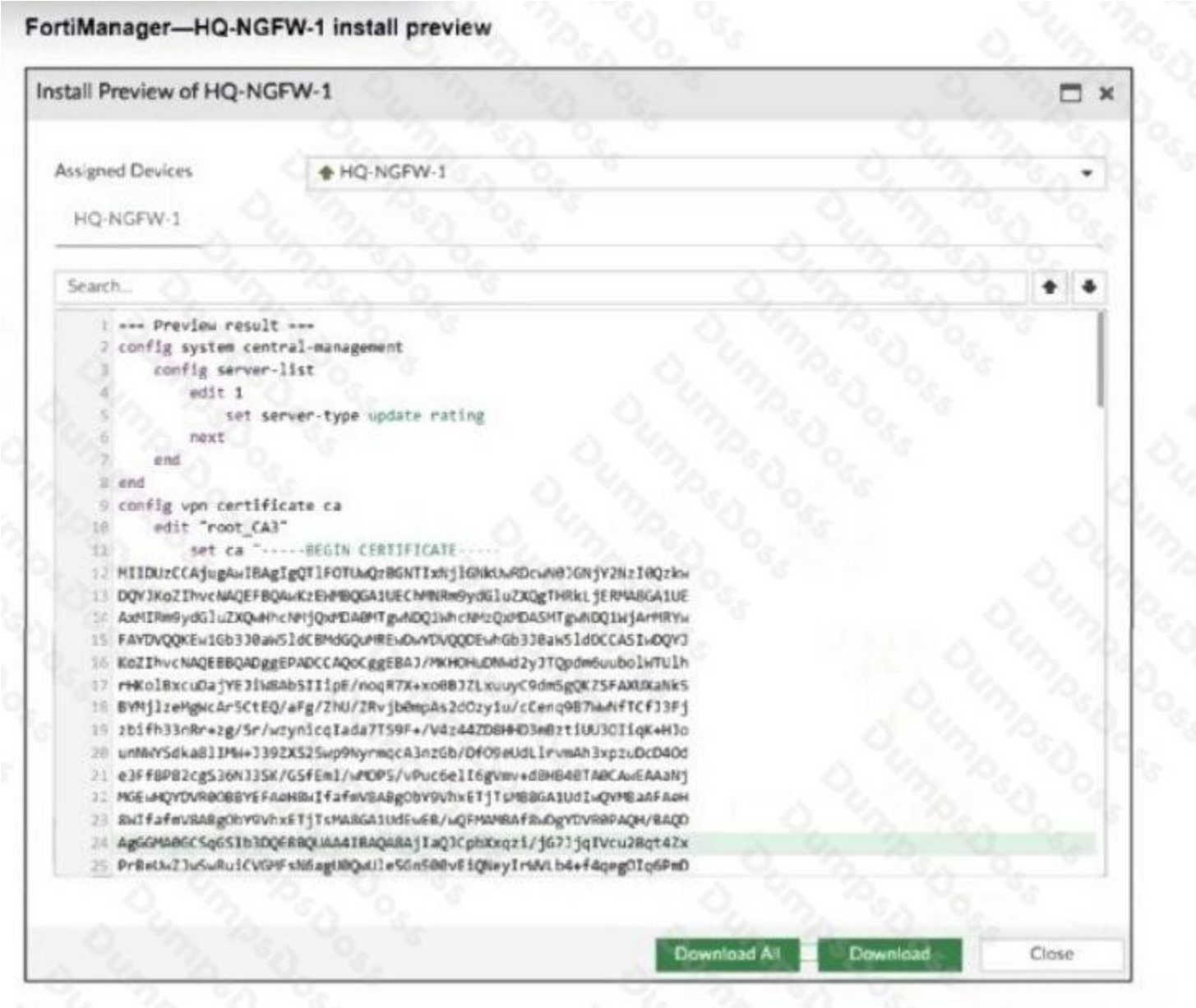
FortiManager pulls AV and IPS updates from the FortiGuard Distribution Network (FDN). In practice, that means it doesn't "try every URL in the list" or blindly use only the public servers you see configured. Instead, it uses the FortiGuard Distribution Network Internet (FDNI) sources that are marked as FDN servers, and it will connect to the appropriate ones based on availability and routing.

Option B (a single IP like 10.0.1.50) would only make sense if you had a local FortiGuard server or explicit override, which isn't what the question is describing. The key idea is that FortiManager relies on the FortiGuard distribution network sources, not just one fixed server.

You can cross-check how Fortinet describes FortiGuard/FDN update behavior in the docs here: https://docs.fortinet.com/document/fortimanager/7.6.0/administration-guide and the FortiGuard overview here: https://www.fortinet.com/support/support-services/fortiguard-security-subscriptions.

QUESTION NO: 5

Refer to the exhibit.



An administrator assigned a new policy package to FortiGate HQ-NGFW-1. In the installation preview, they noticed some settings they did not modify and are unsure about the changes.

Based on the exhibit, which two things will happen if they continue with the installation? (Choose two.)

A. FortiGate HQ-NGFW-1 can use FortiManager firmware templates to upgrade firmware and ratings.

B. FortiGate HQ-NGFW-1 can contact the FortiManager acting as FortiGuard Distribution Server (FDS) to download FortiGuard updates.

C. FortiGate HQ-NGFW-1 will use the root_CA3 certificate in firewall address objects or policies.

D. FortiManager will install the CA certificate named root_CA3 to authenticate FortiGate-to-FortiManager communication protocol (FGFM) tunnel connections with FortiGate HQ- NGFW-1.

ANSWER: B C

Explanation:

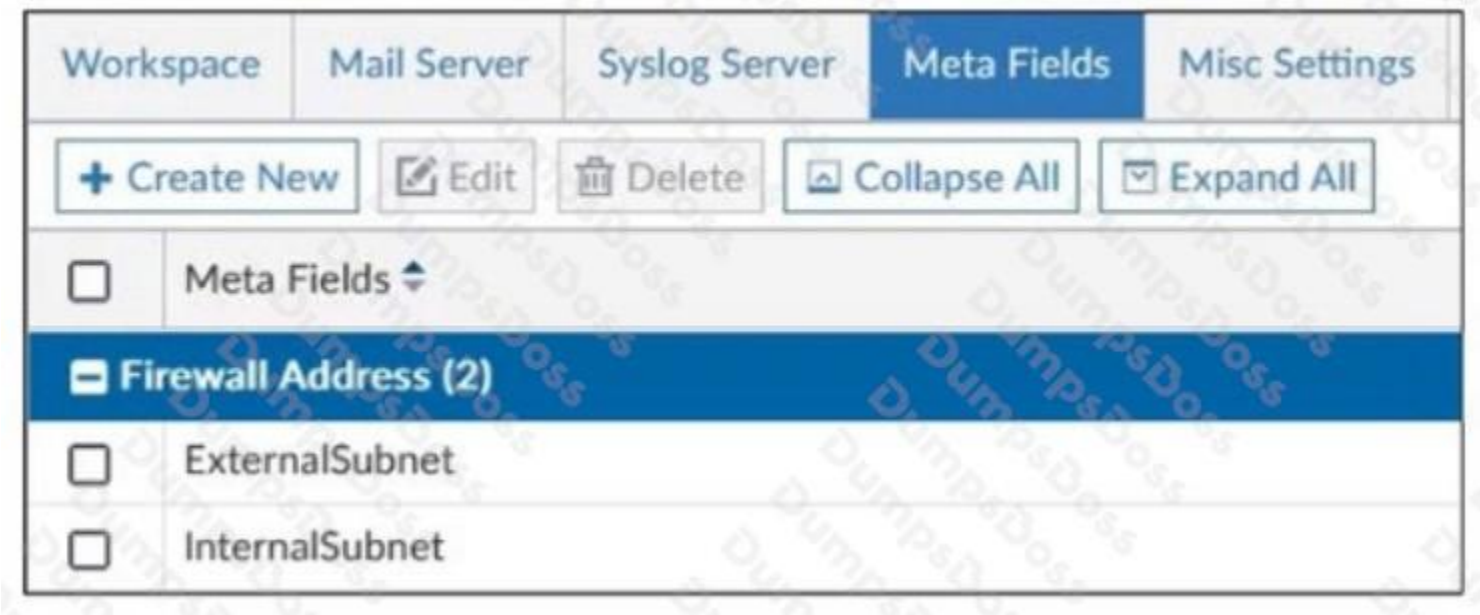
From the install preview, FortiManager is clearly pushing more than just firewall policies. It's also pushing "global" items like FortiGuard/FDS-related settings and certificate objects that are referenced by the policy package. So if the admin proceeds, the FortiGate will be configured to use FortiManager as a FortiGuard Distribution Server (FDS), meaning the FortiGate can pull FortiGuard updates (AV/IPS/etc.) through FortiManager instead of going straight to the public FortiGuard network.

The preview also shows a CA certificate object (root_CA3) being installed/updated as part of the package content. That means the FortiGate will have that certificate available for use in policy-related features (for example, if a policy, SSL inspection profile, or an address object references it). This is different from using a certificate specifically to authenticate the FGFM management tunnel—FGFM authentication is handled separately and isn't what this "root_CA3 in policy package" change implies.

References: https://docs.fortinet.com/document/fortimanager/7.6.0/administration-guide and https://docs.fortinet.com/document/fortigate/7.6.0/administration-guide.

QUESTION NO: 6

Refer to the exhibit.



An administrator created two new meta fields in FortiManager. Which operation can you perform with these parameters?

A. You can add them to objects as custom attributes.

B. You can export them to be used in other ADOMs.

C. You can use them as variables in scripts.

D. You can invoke them using the $ character.

ANSWER: A C D

Explanation:

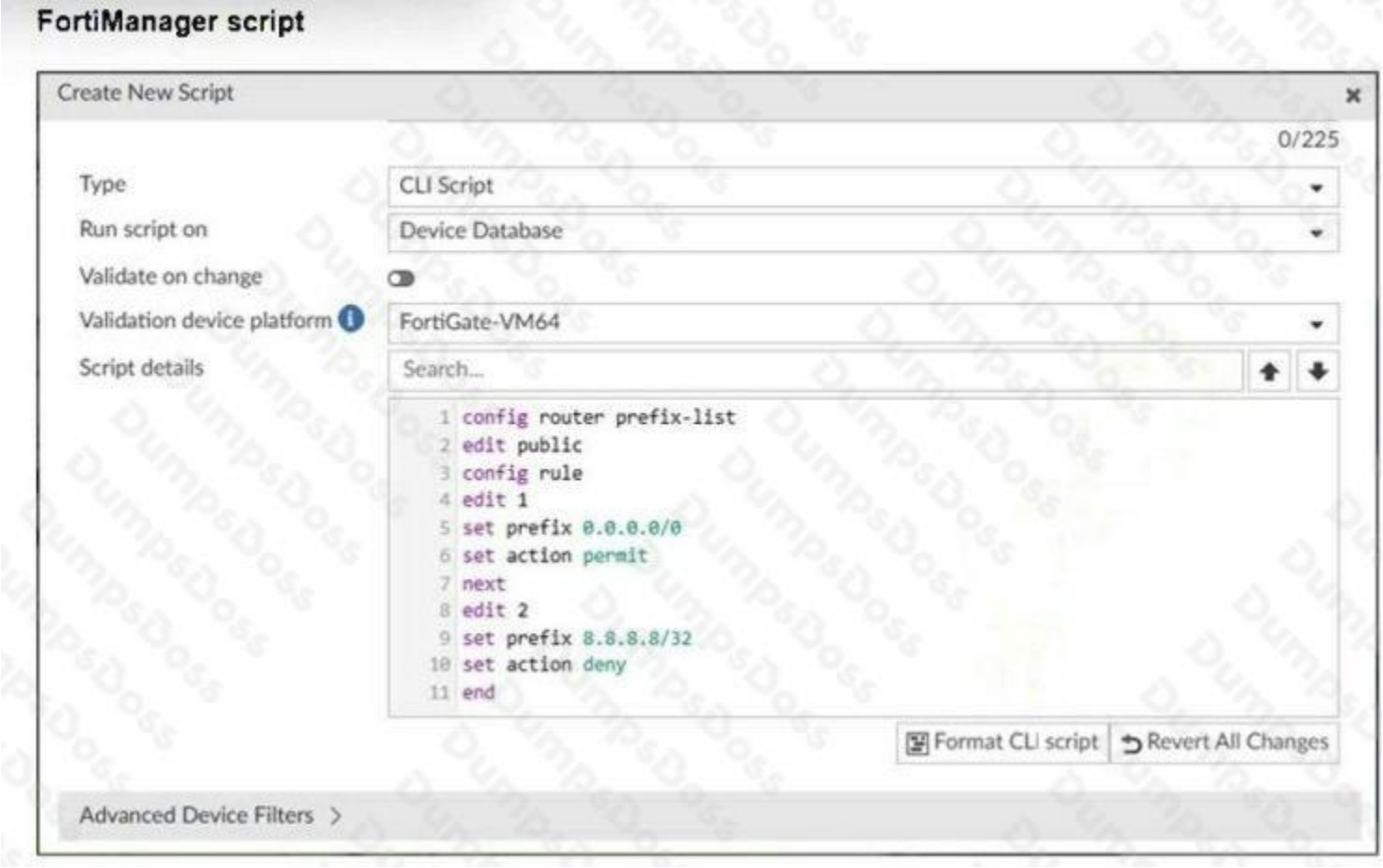
In FortiManager, meta fields are basically extra “tags” you can attach to objects (like addresses, policies, services, etc.) so you can store your own custom info. That’s why you can use them as custom attributes on objects.

Meta fields also become really handy in automation. When you write scripts or use templates, you can reference those meta field values as variables. In practice, FortiManager lets you call variables using the $ style, so you can pull the meta field value into a script or CLI template dynamically.

What you can’t really do is “export meta fields to be used in other ADOMs” as a normal operation. Meta fields are defined per ADOM (or global, depending on how you create them), but there isn’t a typical workflow where you export just meta fields and import them into another ADOM as a feature.

References: https://docs.fortinet.com/document/fortimanager/7.6.0/administration-guide

QUESTION NO: 7



Which two results occur if you run the script using the Device Database option? (Choose two.)

A. The device Config Status is tagged as Modified.

B. The script history shows the successful installation of the script on the remote FortiGate.

C. The successful execution of a script on the Device Database creates a new revision history.

D. The administrator must install these changes on a managed device using the Install Wizard.

ANSWER: A C

Explanation:

When you run a script with the **Device Database** option in FortiManager, you're only changing FortiManager's copy of the configuration (the database), not the live FortiGate. Because of that, FortiManager considers the device's database config to be different from what's currently installed on the FortiGate, so the device's **Config Status** becomes **Modified**.

Also, since the database was changed, FortiManager records that change as a new configuration state, which means it will **create a new revision** in the revision history. This is basically FortiManager keeping track of "what the config looks like now" after the script edits the database.

What you *don't* get is a script history entry showing it was installed on the remote FortiGate, because nothing was pushed to the device yet. If you want the FortiGate to actually receive those changes, you'd still need to install (push) the config afterward.

References: https://docs.fortinet.com/document/fortimanager/7.6.0/administration-guide and https://docs.fortinet.com/document/fortimanager/7.6.0/new-features

QUESTION NO: 8

Which two statements about the integrity of databases on FortiManager are correct? (Choose two.)

A. Scheduled backups run database integrity commands automatically.

B. The diagnose dvm check-integrity command attempts to fix a corrupted file system.

C. The diagnose cdb check adom-integrity command can correct issues related to locked devices.

D. You should fix all database integrity issues before performing a script.

E. Not following the correct upgrade path may cause inconsistencies in the databases.

ANSWER: C D

Explanation:

On FortiManager, the database integrity tools are mainly about checking and repairing FortiManager's own configuration databases (like the ADOM and device manager data), not the underlying disk or file system. That's why saying *diagnose dvm check-integrity* fixes a "corrupted file system" is off—file system issues are a different troubleshooting area.

The *diagnose cdb check adom-integrity* command is used to verify and repair ADOM database consistency, and it can help clean up common database problems like stale/locked entries that can block normal operations (for example, devices appearing stuck/locked in the database).

Also, if FortiManager is already reporting integrity errors, it's smart to fix them before running scripts or making bulk changes. Scripts can touch a lot of objects at once, and existing database inconsistencies can cause failures, partial changes, or more messy cleanup afterward.

References: https://docs.fortinet.com/document/fortimanager/7.6.0/cli-reference and https://docs.fortinet.com/document/fortimanager/7.6.0/administration-guide

QUESTION NO: 9

Which two conditions trigger FortiManager to create a new revision history? (Choose two.)

A. When FortiManager installs device-level changes on a managed device

B. When changes to the device-level database are made on FortiManager

C. When FortiManager is auto-updated with configuration changes made directly on a managed device

D. When a provisioning template is assigned to a managed device on the device-level database

ANSWER: A C

Explanation:

FortiManager creates a new revision when it has a clear “checkpoint” event for a device configuration. One common trigger is when FortiManager installs (pushes) device-level changes to a managed FortiGate. That install operation represents a known, complete configuration state, so FortiManager saves it as a new revision you can roll back to later.

The other typical trigger is when FortiManager detects and imports changes that were made directly on the FortiGate (for example, an admin edits the config on the device). When FortiManager auto-updates/synchronizes its database with those device-side changes, it records that newly discovered configuration state as another revision.

By contrast, simply editing objects in the device-level database on FortiManager doesn’t necessarily create a new revision until those changes are installed, and assigning a provisioning template is more of a setup action—it doesn’t automatically mean the device’s running configuration has reached a new “saved checkpoint” state.

References: https://docs.fortinet.com/document/fortimanager/7.6.0/administration-guide and https://docs.fortinet.com/document/fortimanager/7.6.0/release-notes

QUESTION NO: 10

What are two expected results when both FortiManager and FortiGate are behind network address translation (NAT) devices? (Choose two.)

A. During discovery, the FortiManager NATed IP address is not set by default on FortiGate.

B. FortiGate is discovered by FortiManager through the FortiGate NATed IP address.

C. If the FortiGate-FortiManager communication protocol (FGFM) tunnel is torn down, FortiManager will try to reestablish the FGFM tunnel.

D. FortiGate can announce itself to FortiManager only if the FortiManager non-NATed IP address is configured on FortiGate under central management.

ANSWER: A D

Explanation:

When both sides sit behind NAT, FortiManager can't automatically "guess" the right public (NATed) address to use for management. So during discovery, FortiGate won't automatically have FortiManager's NATed IP filled in for central management—you usually have to set the reachable address (or use an FQDN) yourself. That's why option A makes sense.

Also, for FortiGate to successfully initiate (announce) the FGFM connection, it must be configured with an address it can actually reach. In NAT scenarios, that typically means you configure the FortiManager public/NATed IP (or FQDN) on FortiGate under central management, not the private/non-routable address. The intent behind option D is that you must explicitly configure the correct FortiManager address on the FortiGate for the "call-home" to work.

For reference, see Fortinet's docs on FortiGate–FortiManager (FGFM) communication and central management setup: https://docs.fortinet.com/document/fortimanager/7.6.0/administration-guide and FortiGate central management basics: https://docs.fortinet.com/document/fortigate/7.6.0/administration-guide.