PassQueen
a.k.a dumpsqueen.com

Certified Implementation Specialist - Discovery

ServiceNow CIS-Discovery

Version Demo

Total Demo Questions: 14

Total Premium Questions: 147

Buy Premium PDF

https://passqueen.com

support@passqueen.com

Topic Break Down

Topic	No. of Questions
Topic 1, Discovery Fundamentals	25
Topic 2, Discovery Configuration	46
Topic 3, Discovery Process	12
Topic 4, Discovery Patterns	52
Topic 5, Discovery Troubleshooting	12
Total	147

QUESTION NO: 1

Which selections are necessary to create a custom horizontal pattern to discover a computer operating system?

A. CI type

B. Application Pattern Type

C. Infrastructure Pattern Type

D. Process Strategy Type

ANSWER: A C

Explanation:

Creating a custom horizontal discovery pattern starts by defining the target configuration-item class, so **CI type** is required. The CI type tells Discovery which CMDB class the pattern is intended to populate and provides the context for identification and the attributes that the pattern should retrieve. For discovering a computer operating system, the target is infrastructure rather than an application service, so **Infrastructure Pattern Type** is also required. This type classifies the pattern as one used to discover infrastructure devices and their related data, including computer systems and operating-system information. Together, these selections establish both the CMDB destination and the discovery-pattern category before pattern steps, operations, and variables can be configured in Pattern Designer. ServiceNow documents that horizontal discovery patterns are categorized as infrastructure or application patterns and that patterns are associated with a CI type. See the ServiceNow documentation on Discovery patterns and creating discovery patterns.

QUESTION NO: 2

Where are the target address ranges defined for a recurring Discovery Schedule?

A. The Discovery IP Ranges related list

B. The CMDB Health dashboard

C. The MID Server record comments field

D. The Discovery credential record

ANSWER: A

Explanation:

The Discovery IP Ranges related list is correct because a Discovery Schedule uses associated IP range records to define the addresses or subnets that it scans. The schedule controls when and how Discovery runs, while the related IP ranges define the network scope for that run.

An administrator can associate one or more IP ranges with a schedule and configure the schedule with the appropriate MID Server selection, credentials, behavior, and other discovery settings. When the schedule runs, Discovery uses the specified ranges as the target population for Shazzam and subsequent discovery activity. See the ServiceNow Discovery schedule documentation.

QUESTION NO: 3

Which of the choices provides active discovery errors with a help link for each error?

A. Discovery Dashboard

B. IP Address Failure Report

C. Discovery Schedule

D. MID Server Dashboard

ANSWER: A

Explanation:

Discovery Dashboard is correct because it provides an operational, centralized view of Discovery activity and outcomes. Its Active Discovery Errors area identifies errors that currently require attention and presents a help link for each listed error. Those links are intended to help Discovery administrators understand the error condition and access relevant remediation guidance, allowing them to investigate failures efficiently without needing to begin from individual schedules or logs.

The dashboard is designed to support ongoing Discovery monitoring by surfacing actionable information, including error conditions associated with active discoveries. This makes it a useful starting point for identifying issues that can prevent credentials, probes, patterns, classification, or exploration from completing as expected. By pairing each active error with contextual help, the dashboard supports faster troubleshooting and more consistent operational response.

ServiceNow's Discovery documentation describes the Discovery Dashboard and its role in monitoring Discovery status, including active errors. See ServiceNow Discovery Dashboard documentation and the ServiceNow Discovery documentation landing page.

QUESTION NO: 4

Using the SNMP Query operation on a pattern for a custom device query, it is best practice to do what?

A. Modify the default MIB information

B. Enable SSH as a secondary protocol

C. Use live devices in production

D. Use the publish manufacturer's device MIB

ANSWER: D

Explanation:

Use the published manufacturer's device MIB. An SNMP Query operation in a Discovery pattern needs reliable object identifiers, table definitions, data types, and enumerated values so that it can query and interpret information exposed by the target device accurately. The manufacturer-published MIB is the authoritative source for this information, including vendor-specific MIB modules that extend standard SNMP objects. Using that documentation helps an implementer select supported OIDs, understand the expected response structure, and build a pattern that is maintainable when the device software or pattern is reviewed later.

This practice is especially important for custom devices because their identifying and configuration attributes are frequently exposed through enterprise-specific MIB branches rather than only through generic SNMP system objects. ServiceNow's SNMP Query pattern guidance directs implementers to use the device manufacturer's MIB information when defining custom SNMP queries. MIB modules describe managed objects within the SNMP management framework and provide the semantic details necessary to interpret returned values correctly. See the ServiceNow SNMP Query pattern documentation and the IETF's SNMP management information reference.

QUESTION NO: 5

Which of the following pattern operations query targets? (Choose two.)

A. WMI Query

B. Merge Table

C. Get Process

D. Parse Variable

ANSWER: A C

Explanation:

WMI Query and **Get Process** are pattern operations that obtain information directly from a Discovery target. WMI Query runs a Windows Management Instrumentation query against a Windows host, returning data that can be used by subsequent pattern steps for identification, classification, and attribute population. Get Process retrieves the running-process information from the target. This is particularly useful when a pattern needs to determine whether a specific application or executable is present and running, or to use process details as evidence for identifying an application or configuration item.

Both operations therefore perform target data collection: they communicate through the credentials and protocols available to Discovery and place the resulting information into variables or tables for later use in the pattern. Pattern design commonly starts by collecting reliable target evidence, such as WMI properties or active process data, then evaluates that evidence to identify and populate the appropriate CI. ServiceNow describes Discovery patterns as reusable sequences of operations that discover and identify infrastructure and application components. See the ServiceNow Discovery patterns documentation and the WMI Query pattern-operation documentation.

QUESTION NO: 6

A Discovery Schedule contains a /24 subnet IP Range and a Shazzam batch size of 5000.

How many times will a Shazzam probe be launched during discovery?

A. 1

B. 2

C. 5000

D. 254

ANSWER: A

Explanation:

1 is correct. A /24 IPv4 network has 256 total addresses. In the normal usable-host calculation for a conventional /24 subnet, the network address and broadcast address are not host targets, leaving 254 addresses for Discovery to assess. The Shazzam batch size determines the maximum number of IP addresses included in a single Shazzam probe launch; it does not cause one probe to be launched for every address. Since all 254 usable addresses in this range are below the configured batch size of 5000, Discovery can place the entire range into one batch and launch Shazzam once. Shazzam is the initial Discovery activity used to identify which target IPs respond and which ports or protocols can be used for subsequent classification and exploration. The configured batch size becomes relevant when the number of target addresses exceeds that value, at which point Discovery divides the addresses into multiple Shazzam probe launches. ServiceNow documents Discovery schedule configuration, including IP ranges and Shazzam behavior, in its Discovery schedule documentation. For IPv4 subnet sizing and the meaning of a /24 prefix, see RFC 4632.

QUESTION NO: 7

Which Pattern Designer operations can be used to manipulate temporary table variables? (Choose two.)

A. Filter Table

B. Merge Table

C. Change user

D. Set Credential

ANSWER: A B

Explanation:

Filter Table is used to reduce a table variable to the rows that meet specified conditions. This is useful when a command or query returns several records but a pattern must continue with only the row or rows relevant to the CI being identified.

Merge Table is used to combine table data for later pattern processing. It can help a pattern consolidate data returned from separate operations before parsing, filtering, or using the results to populate CI attributes and relationships. See the ServiceNow pattern operations reference and the ServiceNow Discovery patterns documentation.

QUESTION NO: 8

The CMDB contains which of the following record types? (Choose two.)

A. Model

B. Configuration Item (CI)

C. Asset

D. Relation Type

ANSWER: B D

Explanation:

The CMDB stores records that describe the components of an organization's IT environment and the logical relationships that connect those components. **Configuration Item (CI)** is a core CMDB record type: each record represents a managed component, such as a server, application, database, network device, or business service. CI records are organized into classes, allowing ServiceNow to store attributes appropriate to each kind of component while retaining a common configuration-management data model.

Relation Type is also a CMDB record type. It defines the semantic meaning of a connection between configuration items, such as "Depends on::Used by" or "Runs on::Runs." These relationship definitions provide the standardized relationship behavior and direction used when CMDB relationship records link individual CIs. Together, CI records and relation-type definitions enable ServiceNow to model service dependencies, support impact analysis, and provide a reliable view of the infrastructure and services that support the business.

See ServiceNow's CMDB record types documentation and the CMDB relationships documentation.

QUESTION NO: 9

Which ECC Queue value indicates that the ServiceNow instance has sent work to a MID Server?

A. Input

B. Output

C. Processed

D. Complete

ANSWER: B

Explanation:

Output is correct because an ECC Queue record with an output queue value contains work generated by the ServiceNow instance for a MID Server to process. For Discovery, this can include instructions to run a probe, execute a pattern operation, or perform another task against a target.

After the MID Server completes the work, it returns its results to the instance in an input ECC Queue record. Reviewing output and input records together helps administrators trace the request and response flow during Discovery troubleshooting. See the ServiceNow ECC Queue documentation.

QUESTION NO: 10

Which of the following related lists can assist with troubleshooting discovery from a discovery status?

Choose 3 answers

A. Discovery Log

B. Running Processes

C. ECC Queue

D. Devices

ANSWER: A C D

Explanation:

Discovery Log, ECC Queue, and Devices are the key related lists for investigating a Discovery run from its Discovery Status record. Discovery Log provides the chronological operational detail for the run, including the phases and activities performed as Discovery progresses through a target. This gives an administrator a focused record of what occurred during that specific run and where troubleshooting should begin.

ECC Queue is valuable because Discovery relies on MID Server communication through External Communication Channel messages. Reviewing the messages associated with the Discovery run helps trace requests sent to a MID Server and the responses returned to the instance, including message state, payload information, and timing. This is especially useful for validating the flow of work between the instance and the MID Server.

Devices presents the discovered target devices associated with the status. It lets an administrator review the target-level results of the run, inspect whether individual devices were reached and classified, and drill into records requiring additional investigation. Together, these related lists connect run activity, MID Server message processing, and discovered-device outcomes. ServiceNow's Discovery documentation describes Discovery's MID Server-based operation and provides troubleshooting guidance at ServiceNow Discovery documentation and Troubleshoot Discovery.

QUESTION NO: 11

Which of the following are CMDB Health KPIs? (Choose three.)

A. Completeness

B. Correctness

C. Compliance

D. Availability

E. Capacity

ANSWER: A B C

Explanation:

Completeness, **Correctness**, and **Compliance** are CMDB Health KPIs. Completeness measures whether expected CI attributes contain values. Correctness evaluates whether values conform to configured data-quality requirements. Compliance measures whether CIs meet defined standards, such as required relationship or lifecycle expectations.

These KPIs help CMDB owners identify records and classes requiring remediation. See the ServiceNow CMDB Health documentation.

QUESTION NO: 12

What is the default thread count for a MID Server?

A. 5

B. 1

C. 50

D. 25

ANSWER: D

Explanation:

The default thread count for a MID Server is **25**. This setting governs the maximum number of worker threads that the MID Server can use to process work received from the ServiceNow instance. Worker threads execute activities such as Discovery probes and patterns, orchestration and integration tasks, and other ECC Queue work assigned to that MID Server. A limit of 25 provides a practical default level of parallel processing while helping avoid excessive CPU, memory, network, or target-system load on the host running the MID Server.

The value is controlled through the MID Server configuration parameter `mid.max.threads`. Administrators can tune this parameter when capacity planning or troubleshooting demonstrates that the default is not appropriate for the MID Server host and its workload. Any adjustment should account for available host resources, the nature and volume of jobs, and the capacity of devices or systems being contacted. ServiceNow documents MID Server parameters and the role of worker threads in its product documentation: MID Server parameters and MID Server overview.

QUESTION NO: 13

From an SNMP Query pattern operation, which of the choices are valid Variable Types? (Choose two.)

A. Test

B. Table

C. Scalar

D. CI Type

ANSWER: B C

Explanation:

For an SNMP Query operation in a Discovery or Service Mapping pattern, **Scalar** and **Table** are the applicable variable types. A scalar variable holds one returned value, such as a device name, system description, uptime value, or a single value obtained from a specific SNMP object identifier. This is appropriate when the query is expected to produce one result that later pattern steps can reference directly.

A table variable is used when an SNMP query returns repeating or indexed data. SNMP tables commonly contain multiple rows, for example interface details, routing entries, or storage-related records. The pattern can then iterate through the returned rows and use individual columns to identify components, create relationships, or populate attributes. Selecting the variable type to match the SNMP response structure ensures that the query output is available in the correct form for subsequent pattern logic.

ServiceNow pattern documentation describes the supported pattern-variable structures and their use in operations. See the ServiceNow Pattern Variables reference and the ServiceNow product documentation portal.

QUESTION NO: 14

What entry point type must a horizontal pattern have to execute from a process classifier?

A. A subnet entry point type.

B. HTTP(S) entry point type if the pattern is running on a web server application.

C. TCP entry point type or ALL entry point type.

D. It does not matter, it is triggered for all entry point types.

ANSWER: C

Explanation:

A horizontal discovery pattern that is intended to run from a process classifier must use a **TCP entry point type or ALL entry point type**. A process classifier evaluates a discovered process and, when its criteria match, can launch a relevant horizontal pattern to identify and classify the application or software represented by that process. For this invocation path, Discovery requires a pattern entry point that is compatible with process-based, TCP-oriented classification. The TCP entry point lets the pattern start with the target and port context associated with the classified process. The ALL entry point type is also eligible because it allows the pattern to be invoked in all supported discovery contexts, including from a process classifier.

Choosing the appropriate entry point is important because it controls the contexts in which the pattern is available for execution. In this case, configuring TCP or ALL ensures that the process classifier can select and launch the pattern during horizontal discovery, enabling the pattern to perform its identification steps and update the appropriate configuration item information. ServiceNow documents the relationship between horizontal patterns, entry point types, and process classification in its horizontal discovery patterns documentation and process classification documentation.