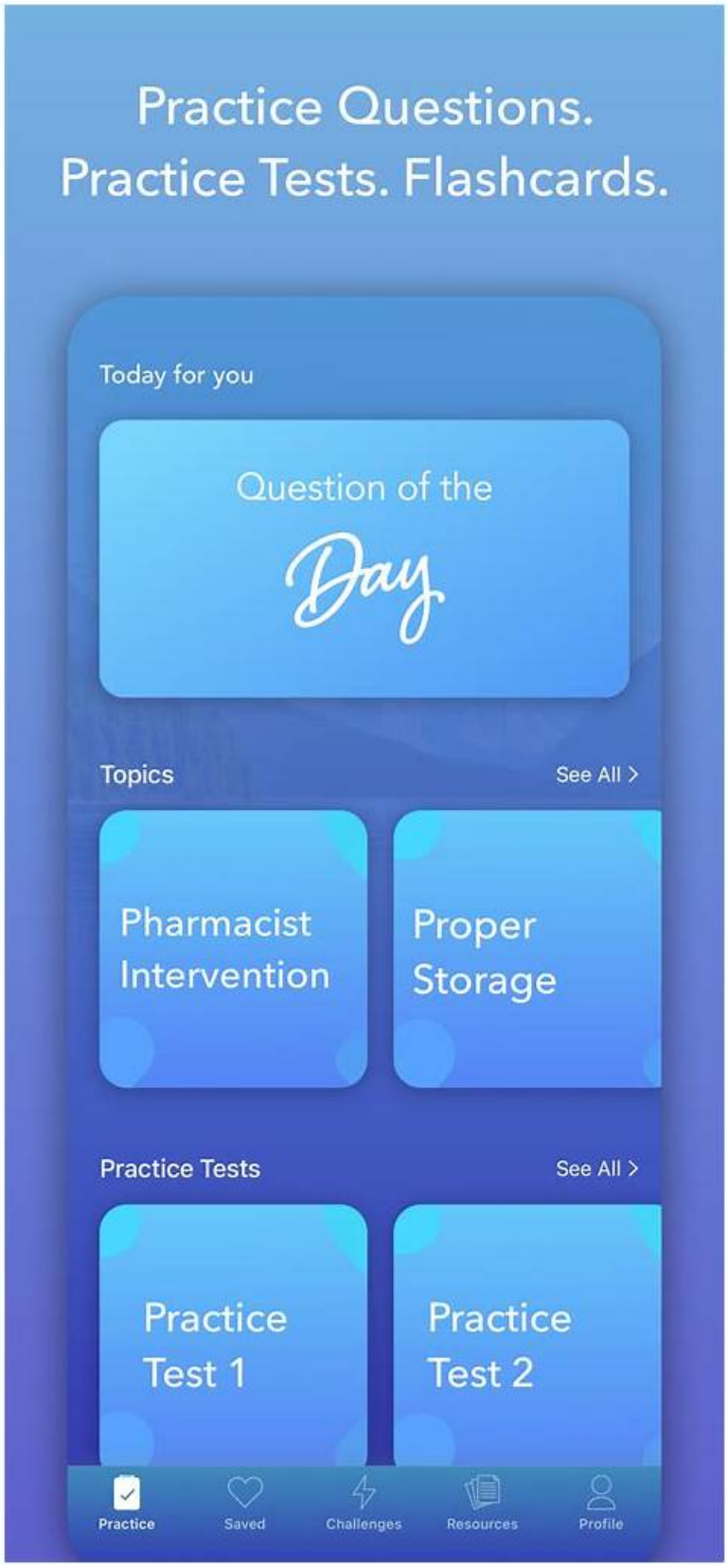
2026 Updated New PT-AM-CPE Test Topics | Certified Professional - PingAM Exam 100% Free Valid Test Registration



What's more, part of that TestPDF PT-AM-CPE dumps now are free: https://drive.google.com/open?id=11jB1TLh3Vkgyn3zpJCiLVetpHbEsBvi-

Since the content of the examination is also updating daily, you will need real and latest Ping Identity PT-AM-CPE Dumps to prepare successfully for the PT-AM-CPE Certification Exam in a short time. People who don't study from updated PT-AM-CPE questions fail the examination and loss time and money.

One of the great features of our PT-AM-CPE training material is our PT-AM-CPE pdf questions. Certified Professional - PingAM Exam exam questions allow you to prepare for the real PT-AM-CPE exam and will help you with the self-assessment. You can easily pass the PT-AM-CPE exam by using PT-AM-CPE dumps pdf. Moreover, you will get all the updated PT-AM-CPE Questions with verified answers. If you want to prepare yourself for the real Certified Professional - PingAM Exam exam, then it is one of the most important ways to improve your PT-AM-CPE preparation level. We provide 100% money back guarantee on all PT-AM-CPE braindumps products.

>> New PT-AM-CPE Test Topics <<

New PT-AM-CPE Test Topics: 2026 Ping Identity Realistic New Certified Professional - PingAM Exam Test Topics Pass Guaranteed

Our PT-AM-CPE exam guide has high quality of service. We provide 24-hour online service. If you have any questions in the course of using the PT-AM-CPE exam questions, you can contact us by email. We will provide you with excellent after-sales service with the utmost patience and attitude. And we will give you detailed solutions to any problems that arise during the course of using the PT-AM-CPE practice torrent. And our PT-AM-CPE study materials welcome your supervision and criticism. With the company of our PT-AM-CPE study materials, you will find the direction of success.

Ping Identity PT-AM-CPE Exam Syllabus Topics:

Topic	Details
Topic 1	• Federating Across Entities Using SAML2: This domain covers implementing single sign-on using SAML v2.0 and delegating authentication responsibilities between SAML2 entities.
Topic 2	• Extending Services Using OAuth2-Based Protocols: This domain addresses integrating applications with OAuth 2.0 and OpenID Connect, securing OAuth2 clients with mutual TLS and proof-of-possession, transforming OAuth2 tokens, and implementing social authentication.
Topic 3	• Enhancing Intelligent Access: This domain covers implementing authentication mechanisms, using PingGateway to protect websites, and establishing access control policies for resources.
Topic 4	• Installing and Deploying AM: This domain encompasses installing and upgrading PingAM, hardening security configurations, setting up clustered environments, and deploying PingOne Advanced Identity Platform to the cloud.
Topic 5	• Improving Access Management Security: This domain focuses on strengthening authentication security, implementing context-aware authentication experiences, and establishing continuous risk monitoring throughout user sessions.

Ping Identity Certified Professional - PingAM Exam Sample Questions (Q25-Q30):

NEW QUESTION # 25
Which area of PingAM does affinity mode relate to?

- A. Self-service
- B. Authentication
- C. Load balancing
- D. Authorization

Answer: C

Explanation:
In PingAM 8.0.2, the term Affinity Mode (or session affinity) is strictly related to Load Balancing (Option B). It describes a configuration where a load balancer ensures that all requests belonging to a specific user session are consistently routed to the same PingAM server instance in a cluster.
According to the "Load Balancing" and "Deployment Planning" documentation:
Affinity is critical for performance in stateful deployments. While PingAM can operate in a "stateless" manner by retrieving sessions from the Core Token Service (CTS) on every request, this creates unnecessary overhead. Affinity Mode allows the AM server to satisfy requests using its local "In-memory" session cache.
There are two primary levels of affinity discussed in PingAM documentation:
Client-to-AM Affinity: Usually handled by the load balancer using a cookie (like the AMLB cookie) to keep the user on the same AM node.
AM-to-DS Affinity: Used when AM connects to the CTS (PingDS). This ensures that an AM server always talks to the same directory server node to avoid "replication lag" where a session might be written to one DS node but not yet visible on another.
Without affinity, the system remains functional due to the CTS, but performance decreases as every request requires a cross-network database lookup. Therefore, affinity is a core concept of the Load Balancing and high-availability architecture.

NEW QUESTION # 26
A PingAM administrator wants to deny access to an area of a protected application if the end user has been logged in for more than 10 minutes. How can this be achieved?

- A. Use a policy with a Time environment condition
- B. Use a policy with an Active session time environment condition
- C. Use a policy with a Scripted environment condition
- D. Use a policy with a Current session properties environment condition

Answer: C

Explanation:
To enforce complex authorization logic based on session duration, PingAM 8.0.2 administrators must move beyond the static "Out-of-the-Box" conditions.
Analysis of the options based on the "Policy Conditions" documentation:
Time Condition (Option A): This condition is used to restrict access based on the clock time of day or day of the week (e.g., "Allow access only between 9 AM and 5 PM"). It does not track the elapsed time of a specific user session.
Current Session Properties (Option B): This condition checks for the presence of specific key-value pairs in a session. While a session contains a startTime property, this condition is designed for matching static values (like department=HR), not for performing mathematical time calculations.
Active Session Time (Option D): This is not a standard default condition name in the PingAM 8.0.2 policy engine.
The Correct Approach (Option C): A Scripted Policy Condition is required for this use case. Within a Policy Condition script, the administrator has access to the session object. The script can retrieve the startTime (or creationTime) of the session and compare it against the current system time (currentTime).
Example logic in the script:
var sessionStartTime = session.getProperty("startTime");
var maxDuration = 10 * 60 * 1000; // 10 minutes in milliseconds
if ((currentTime - sessionStartTime) > maxDuration) { authorized = false; } By using a script, PingAM can dynamically calculate the age of the session at the moment of the access request and return a "Deny" decision if the 10-minute threshold has been exceeded.
This provides the granular control needed for high-security environments where "session freshness" is a requirement for specific sensitive resources.

NEW QUESTION # 27
When making a request to the /oauth2/access_token endpoint using the JWT profile client authentication method, which parameter is used to provide the JWT value?

- A. client_assertion
- B. client_credentials
- C. client_token_value
- D. client_id

Answer: A

Explanation:
PingAM 8.0.2 supports advanced client authentication methods defined in the OpenID Connect and OAuth 2.0 specifications, including private_key_jwt and client_secret_jwt. These methods allow a client to authenticate without sending a static password/secret in the request. Instead, the client generates and signs a JSON Web Token (JWT).
According to the "OAuth 2.0 Client Authentication" and "JWT Profile for Client Authentication" (RFC 7523) documentation, when a client sends this JWT to the /oauth2/access_token endpoint, it must use the client_assertion parameter.
The request must also include the client_assertion_type parameter, which must be set to the constant value: urn:ietf:params:oauth:client-assertion-type:jwt-bearer.
Option A (client_credentials) is a grant type, not a parameter for providing a JWT.
Option B (client_token_value) is not a standard OAuth2 parameter name.
Option C (client_id) is often included in the request, but it is the identifier of the client, not the container for the cryptographic assertion itself.
When PingAM receives a request with a client_assertion, it extracts the JWT, verifies the signature using the client's public key (stored in the client's profile or retrieved via a JWKS URI), and validates the standard claims (iss, sub, aud, exp). This method is significantly more secure than simple secrets because it proves the client possesses the private key and limits the window for replay attacks through the token's expiration claim.

NEW QUESTION # 28
When a user undergoes a session upgrade, what is the outcome?

- A. The session properties are copied to a new session, and a new session token is handed to the client
- B. A new session is created, and the original session properties are not copied
- C. A new session is created, and the original session is deleted
- D. The session is updated with new properties, but the session token remains the same

Answer: A

Explanation:
Session Upgrade in PingAM 8.0.2 is the mechanism by which a user's current authenticated session is "elevated" to a higher authentication level (Auth Level). This is commonly triggered by Step-up Authentication requirements, where a user attempts to access a highly sensitive resource that requires a stronger authentication method (such as MFA) than what was used for their initial login.
According to the PingAM documentation on "Session Upgrade Outcomes," the process is not merely a modification of the existing session. Instead, when a user successfully completes the additional authentication requirements (the "Advice"):
Creation of a New Session: PingAM generates a brand-new authenticated session. This new session is assigned a higher authentication level corresponding to the tree or module just completed.
Property Copying: To ensure a seamless user experience, PingAM copies the session properties (attributes, constants, and other metadata) from the original lower-level session into the new higher-level session. This ensures that information gathered during the initial login remains available to applications.
Token Replacement: Because the session ID is part of the session token (SSO Token), a new session implies a new token. PingAM hands the client a new session token to replace the original one. The client (browser or application) must then use this new token for subsequent requests.
If the realm is configured for server-side sessions, the new session is stored in the Core Token Service (CTS). If configured for client-side sessions, a new signed/encrypted JWT is sent to the client as a cookie. The key distinction is that the token changes, and properties are preserved through copying, which distinguishes Option B as the correct technical description of the internal AM lifecycle.

NEW QUESTION # 29
In a default PingAM configuration, what type of keystore stores the secret ID named storepass, which contains the encrypted password of the default-keystore secret store?

- A. Filesystem secret store
- B. Hardware Security Module secret store
- C. Environment and system property secret store
- D. Keystore secret store

Answer: A

Explanation:
In PingAM 8.0.2, the management of sensitive data such as passwords and cryptographic keys is handled through a unified Secret Store framework. This framework abstracts the source of the secret from the component that consumes it using Secret IDs. One of the most critical secret IDs in a standard installation is storepass.
The storepass secret ID is specifically used by the default-keystore (which is typically a "Keystore secret store" pointing to keystore.jks or keystore.p12). Before AM can access the keys within the default-keystore to sign tokens or encrypt data, it must first unlock the keystore itself using the password mapped to the storepass secret ID.
According to the PingAM "Secrets, certificates, and keys" documentation, in a default file-based configuration, PingAM initializes a Filesystem secret store as its primary global store. This store is configured to look into a specific directory within the AM configuration path (usually .../openam/secrets/). Inside this directory, AM expects to find files named after the secret IDs they contain. For the storepass ID, there is typically a corresponding file (such as storepass or .storepass) containing the cleartext or encrypted password required to open the primary keystore.
While AM can be configured to use an Environment and system property secret store (Option B) for high-portability cloud deployments, the "out-of-the-box" default behavior during a standard installation relies on the filesystem. Option A is incorrect because the storepass is the key to the keystore, not a secret inside it, and Option D refers to specialized hardware integrations not used in a default software-only setup. Therefore, the Filesystem secret store is the correct technical answer for the default location of the storepass.

NEW QUESTION # 30
.....

www.stes.tyc.edu.tw, Disposable vapes

What's more, part of that TestPDF PT-AM-CPE dumps now are free: https://drive.google.com/open?id=11jB1TLh3Vkgyn3zpJCiLVetpHbEsBvi-