Welcome to Cert007 - Your Ultimate IT Certification Partner

- Real Exam Questions
- Free Updates
- Expert Support
- Instant Access
- Money-Back Guarantee

Visit us at **https://www.cert007.com/** for more information

Exam : **PT-AM-CPE**

Title : Certified Professional - PingAM Exam

Version : DEMO

1.In the default Cloud Developer Kit (CDK) deployment of the forgeops repository, which pods provide the user interface functionality?
A. admin-ui, end-user-ui, login-ui
B. amadmin-ui, idmadmin-ui, login-ui
C. am-ui, idm-ui, login-ui
D. am-ui, idm-ui, end-user-ui
Answer: A
Explanation:
The Cloud Developer Kit (CDK), part of the forgeops repository, represents the modern approach to deploying the Ping Identity Platform (including PingAM 8.0.2) in a containerized, Kubernetes-native environment. According to the PingAM deployment and ForgeOps documentation, the platform has transitioned from a monolithic architecture—where the user interface was embedded within the AM web application—to a decoupled, microservices-aligned architecture. In a standard CDK deployment, the user interface components are separated into their own distinct pods to allow for independent scaling, updates, and management.
The three specific pods that provide user interface functionality in a default CDK environment are: admin-ui: This pod hosts the administrative console. It is the centralized interface that administrators use to configure realms, manage identity stores, define authentication trees, and oversee the general health of both PingAM and PingIDM. By separating the administrative UI from the core engine, the platform reduces the attack surface and allows for more granular resource allocation. end-user-ui: This pod serves the self-service portal for end-users. It is responsible for providing the interface where users can manage their own profiles, update passwords, register Multi-Factor Authentication (MFA) devices, and manage their consent for OAuth2/UMA applications. This UI interacts with the back-end via REST APIs to ensure a seamless and responsive user experience. login-ui: This is a specialized pod dedicated to the authentication journey. When a user interacts with an "Intelligent Access" tree, the login-ui pod renders the callbacks (such as username prompts, password fields, or MFA challenges). This pod ensures that the presentation layer of the authentication process is modernized and distinct from the heavy processing logic of the PingAM core.
Collectively, these three pods ensure that the "User Interface" layer of the deployment is modular. This architecture is a prerequisite for high-availability deployments and is the standard configuration verified in the ForgeOps documentation for version 8.0.2 deployments.

2.A user enters their credentials, but is faced with the error message "user requires profile to login". What is a possible cause of this message?
A. Policies have not been defined to allow a user to access their profile page
B. The realm has not been set to user profile ignore mode
C. The user has not filled in the required information in their profile
D. The user has not entered the correct credentials
Answer: B
Explanation:
This error message is directly related to the User Profile configuration within a specific realm in PingAM 8.0.2. In the "Core Authentication Attributes" of a realm, PingAM defines how it should handle user identities after they have successfully provided valid credentials through an authentication tree or chain. There are primarily four modes for the User Profile setting:

Required: This is often the default. It specifies that after a user successfully authenticates, PingAM must be able to locate a corresponding user entry in the configured Identity Store. If the user exists in the datastore, the session is created. If the user does not exist, authentication fails with the error message "user requires profile to login" (or a similar profile-related exception in the logs).
Ignored: In this mode, PingAM issues an SSO session token immediately upon successful credential validation, regardless of whether a user profile exists in the back-end repository. This is useful for temporary or guest access where no permanent record is needed.
Dynamic: AM attempts to find the user; if the user is not found, it automatically creates a new profile in the identity store.
Dynamic with User Alias: Similar to dynamic creation but supports aliasing.
If an administrator sees the "user requires profile to login" error, it confirms that the credentials themselves were technically correct (the user passed the authentication nodes), but the realm is currently in Required mode (it has not been set to Ignore or Dynamic) and no matching entry exists in the identity store. This frequently happens in migration scenarios or when using external identity providers (like Social IDPs) where the "Link" or "Provisioning" step has not been properly configured in the authentication journey. To resolve this, the administrator must either pre-provision the user, set the mode to Ignore, or implement a Create Object node within the authentication tree to handle dynamic provisioning.

3.When a user undergoes a session upgrade, what is the outcome?
A. A new session is created, and the original session is deleted
B. The session properties are copied to a new session, and a new session token is handed to the client
C. The session is updated with new properties, but the session token remains the same
D. A new session is created, and the original session properties are not copied
Answer: B
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.

4.Examining the following JSON object, what is a valid value for the type part (shown in bold font) of the claim value in a PingAM implementation?
JSON
JSON
"act": {
"sub": "(type!subject)"
}
A. agent
B. usr
C. uid
D. user
Answer: B
Explanation:
The JSON object structure provided refers to the Actor (act) claim used in OAuth 2.0 Token Exchange (RFC 8693) within PingAM 8.0.2. This claim is essential for scenarios involving delegation or impersonation, where one entity (the actor) is performing an action on behalf of another (the subject). In PingAM, the sub (subject) field within the act claim follows a specific internal format: (type!subject). According to the PingAM 8.0.2 documentation regarding Token Exchange Configuration, the type part of this string is a mandatory prefix that identifies the category of the identity acting as the delegate. The documentation explicitly defines two primary valid values for this type field:
usr: This specifies that the subject is a user/identity from an identity store. For instance, if a user is acting on behalf of another user, the claim would appear as "(usr!username)".
age: This specifies that the subject is an OAuth 2.0/OpenID Connect-related agent or client. Examples include an OAuth 2.0 client, a Remote Consent Service agent, or a Web/Java Agent internal client. An example would be "(age!myClientID)".
While "user" and "agent" are the descriptive terms for these categories, the actual technical values recognized and emitted by PingAM in the claim string are the three-letter shorthand codes. Therefore, usr (Option B) is the correct valid value. Choosing "user" (Option D) would be technically incorrect in the context of the exact string format required by the AM engine. This formatting ensures that when the token is introspected or validated, the resource server can correctly parse whether the actor is a human user or a machine client.

5.Which of the following multi-factor authentication protocols are supported by PingAM?
A) Open authentication
B) Security questions
C) Web authentication
D) Universal 2nd factor authentication
E) Push authentication
A. B, C, and D
B. A, B, and E
C. A, C, and E
D. A, B, and C

Answer: C

Explanation:

PingAM 8.0.2 provides a robust framework for Multi-Factor Authentication (MFA) centered around modern, secure protocols and the Intelligent Access (Authentication Trees) engine. When discussing supported "protocols" in the context of MFA in PingAM documentation, the focus is on standardized methods for secondary verification.

The primary supported MFA pillars in PingAM 8.0.2 are:

Open Authentication (OATH): AM supports the OATH standards, specifically TOTP (Time-based One-Time Password) and HOTP (HMAC-based One-Time Password). This is implemented through the "OATH" authentication nodes, allowing users to use apps like ForgeRock Authenticator, Google Authenticator, or YubiKeys in OATH mode.

Web Authentication (WebAuthn): This is the implementation of the FIDO2 standard. It allows for passwordless and secure second-factor authentication using biometrics (like TouchID/FaceID) or hardware security keys (like YubiKeys). It is the successor to older standards and is natively supported via WebAuthn nodes.

Push Authentication: This is a proprietary but highly secure protocol used specifically with the ForgeRock/Ping Authenticator app. It allows a "Push" notification to be sent to a registered mobile device, which the user then approves or denies.

Why others are excluded from the selection: While PingAM supports Security Questions (KBA) and Universal 2nd Factor (U2F), they are often categorized differently in the 8.0.2 documentation. Security Questions are considered a "User Self-Service" or "Legacy" validation method rather than a modern MFA protocol. U2F is technically superseded by and included within the WebAuthn framework in PingAM 8.0.2. Thus, the most accurate grouping of distinct, core MFA protocols supported in the current version is A, C, and E, making Option C the correct answer.