<table>
<thead>
<tr>
<th>Page</th>
<th>Section</th>
</tr>
</thead>
<tbody>
<tr>
<td>03</td>
<td>INTRODUCTION</td>
</tr>
<tr>
<td>04</td>
<td>PLANNING PHASE</td>
</tr>
<tr>
<td>06</td>
<td>INITIAL DEPLOYMENT AND ORACLE ACCESS MANAGER INTEGRATION PHASE</td>
</tr>
<tr>
<td>10</td>
<td>APPLICATION MIGRATION PHASE</td>
</tr>
<tr>
<td>13</td>
<td>FINALIZE MIGRATION PHASE</td>
</tr>
<tr>
<td>14</td>
<td>APPENDIX A: ORACLE ACCESS MANAGER APPLICATION CLASSIFICATIONS</td>
</tr>
<tr>
<td>15</td>
<td>APPENDIX B: AUTHENTICATION FLOW PROCESS OPTIONS</td>
</tr>
<tr>
<td>16</td>
<td>APPENDIX C: OAM OBJECTS AND CORRESPONDING PINGACCESS/ PINGFEDERATE OBJECTS</td>
</tr>
<tr>
<td>19</td>
<td>APPENDIX D: MAPPING OAM LDAP AUTHORIZATION RULES TO PING ACCESS GROOVY SCRIPTS.</td>
</tr>
</tbody>
</table>
INTRODUCTION

Web access management (WAM) systems have been a part of enterprises for decades. It’s critical to control access and audit applications while reducing the friction for users to interact with applications.

As business has shifted and changed in the Internet era, identity and access management systems have to change and grow to support new use cases. Traditional IAM vendors have been challenged to respond in a timely manner to these changing business requirements. Ping Identity® offers the solution with Identity Defined Security to extend and replace traditional IAM vendor products.

Ping’s products are designed to provide a simple migration strategy from traditional IAM solutions. PingFederate® and PingAccess® integrate with Oracle Access Manager to facilitate the transfer of applications without disrupting your end users. The transition from one system to Ping can happen on a timeline that supports your business.

A typical migration to the Ping Identity Platform involves the four phases discussed below. We recommend contacting one of our professional services partners to assist and guide your organization through the migration process. One of our partners, Likeminds Consulting, assisted us with the many of the technical details included in this guide, and has supported many clients with their migration from Oracle to Ping. Learn more at http://likemindsconsulting.com/
Understanding Your Architecture
The first step in planning your migration from Oracle Access Manager is to survey your deployment and understand the different WAM pieces in play. A standard WAM deployment is shown in Figure 1.

Figure 1: A typical CA Single Sign-On WAM deployment
Typically, an Oracle Access Manager deployment uses web agents installed on web servers that sit adjacent to each application. As you plan for the migration, it’s a good time to evaluate your deployment architecture and the lessons learned from maintaining agents. The total cost of ownership related to an agent-based deployment is typically very high.

Deploying a gateway-based architecture can save deployment time as well as reduce the total cost of ownership. PingAccess can be deployed quickly, and the gateway can easily be integrated into existing applications and network traffic can be re-routed to it. A system fully migrated to PingAccess using a gateway deployment is shown in Figure 2.

*Figure 2: System migrated to PingAccess*
If the PingAccess gateway architecture is not feasible, the migration can still be performed with PingAccess agents following the same deployment architecture as the typical Oracle Access Manager installation. A system fully migrated to PingAccess using agents is shown in Figure 3.

PingAccess also supports both architectures simultaneously. Some applications can continue with the agent-based architecture while the rest of the applications are protected by the PingAccess gateway. This approach can also be used as an intermediary as the deployment architecture is shifted from agents to gateways.

**Application and Policy Inventory**

After understanding and planning the deployment architecture, you’ll take inventory of the applications to be migrated. You’ll also evaluate each application to understand the dependencies on Oracle Access Manager. A framework for application classification can be found in Appendix A.

For each application, you should catalog the realms, rules and rule groups to be migrated into PingAccess applications, resources, rules and rule sets.

Next, you’ll take inventory and prepare the Oracle Access Manager policies for migration. You’ll then catalog each policy and identify the protected resources, the rules and access control lists (ACLs) used and the authentication requirements used to identify users.
Authentication and Session Management
To begin this step, you’ll evaluate and ensure you understand the authentication processes in the environment while planning the migration. If Oracle Access Manager is used to authenticate users, you must decide when to migrate authentication away from those systems. There are two recommended methods to migrate authentication from Oracle Access Manager. All authentication flows are explained in Appendix B.

The first method migrates authentication from Oracle Access Manager at the beginning of the migration process. Migrating authentication at the beginning of the process makes PingFederate the source of record for authentication and identity. If this method is selected, PingFederate can be integrated with Oracle Access Manager to provide applications with those still expecting them.

The second method migrates authentication from Oracle Access Manager at the end of the migration process. Authentication will continue to occur with Oracle Access Manager even if users are accessing PingAccess protected application. PingFederate can be configured and integrated with Oracle Access Manager to trust and accept tokens issued by that system. When it’s time to decommission Oracle Access Manager, the authentication process is migrated to PingFederate.

Next, you’ll evaluate the session management process. When working with two concurrent WAM systems, it’s important to understand how the sessions integrate. Without evaluating and integrating the sessions between the systems, end users may be unnecessarily prompted to authenticate to the system.

Project Planning
After classifying the applications, cataloging the policies and understanding the authentication and session management processes, you should create a project plan to acquire the resources to install PingFederate and PingAccess, perform integration with Oracle Access Manager and migrate the applications.

Based on the application classifications, we recommend the following migration order:

1. Start with a simple application as a proof-of-concept (PoC) to understand the migration process and remove risk further in the process.
2. Migrate low complexity applications (i.e., ones with no Oracle Access Manager customizations, or those using custom Oracle Access Manager APIs) into PingAccess.
3. Migrate the remaining medium/high complexity applications. Customizations and complex authentication schemes should be migrated and configured as well.
INITIAL DEPLOYMENT AND ORACLE ACCESS MANAGER INTEGRATION PHASE

PingFederate Installation
To install PingFederate, find the document here.

PingAccess Installation
To install PingAccess, find the documentation here.

PingAccess Integration with PingFederate
Connect PingAccess to PingFederate for authentication by following the documentation located here.

Oracle Access Manager Integration
Oracle Access Manager Integration leverages the authentication and session management features in both environments. There are two approaches to this, and depending on your deployment needs, one or both might be used at different stages of the migration. Read about the two approaches on the following pages.
**Option 1: Leverage Oracle Access Manager**

In its existing deployment, Oracle Access Manager is likely the entry point for users in the system, dictating how authentication is performed and creating sessions once successful. The PingFederate and PingAccess infrastructure can leverage this authentication point by using the PingFederate WAM Integration Kit. This approach is a good way to try some of the PingAccess access control and session management capabilities without impacting the majority of applications or network infrastructure.

Consult the [PingFederate WAM Integration Kit documentation](#) for detailed steps on how to deploy it.

Once the PingFederate WAM Integration Kit is deployed, Oracle Access Manager sessions are created or updated when accessing applications protected by PingAccess. From that point on, if the user accesses applications using only a PingAccess session, there's a risk of the Oracle Access Manager session timing out. To prevent this behavior and effectively centralize the sessions, employ a 'session keep alive' rule or site authenticator to ensure that the Oracle Access Manager session is periodically ‘refreshed’ by calling a test resource on the user’s behalf. Any updates to this session (i.e., new cookie values returned in a SetCookie HTTP response header) should be relayed back to the browser.

A summary system architecture diagram of this WAM migration phase is depicted below:
**Option 2: Token Mediation**

An alternative to leveraging sign-on interfaces in Oracle Access Manager is to migrate those to PingFederate as the sole entry point into the system. PingFederate provides a full-featured username/password sign-on mechanism, known as the HTML Form Adapter, and it easily connects to existing directory servers. This may be supplemented with other sign-on mechanisms such as Integrated Windows Authentication (IWA) and strong authentication (i.e., PingID, X.509 certificates, third-party OTP tokens, etc.).

When a mix of PingAccess and Oracle Access Manager protected applications exist in the environment, token mediation can be used to ensure a seamless end user SSO and session management experience. In this configuration, the PingAccess gateway exchanges its own web session cookie (known as the PA token) for the target Oracle Access Manager session, which is presented to a back-end web server on which an agent is still deployed. This may be most convenient in a phased migration when a majority of the applications are protected by PingAccess, but pockets of applications still exist that have third-party agents installed on their supporting web server that are difficult to remove.

Details of a typical token mediation are presented in the PingAccess documentation.

The token mediator site authenticator provides this capability in PingAccess and depends on the PingFederate STS with appropriate token generators installed, such as the WAM token translator. View the documentation here.

A summary system architecture diagram of this WAM migration phase is depicted below:
The first step in migrating an application from Oracle Access Manager to PingAccess is to ensure that the required integrations are complete and that PingAccess can provide the necessary identity information to the application. The next step is to create the PingAccess rules that enforce access in the same way that Oracle Access Manager enforces access. More information about defining rules can be found here.

The final step is to select the architecturally appropriate deployment option. Appendix C highlights a sample migration scenario from an agent-based architecture into either a gateway-based or proxy architecture.

**Migrating Application Protection to the PingAccess Gateway**

The first step is to determine if the application will continue to require the Oracle Access Manager OBSSO session token. If it's still required (for example, if the Oracle Access Manager agent will remain on the target web server), you configure token mediation so that PingAccess will acquire a target token on behalf of the user.

Next, you configure the application in PingAccess, defining where the protected web server resides, how it'll be exposed via the gateway, what the required web session management controls are, and how end user identity details will be passed down to the application if needed (i.e., HTTP headers using identity mappings).

Then you define your access policies using rules that are equivalent to (or a simplification of) those employed in Oracle Access Manager.

Last, you configure network routing to start sending request traffic to PingAccess instead of the application.

**Migrating Application Protection to a PingAccess Agent**

In scenarios where an agent model is still preferred, the PingAccess agent replaces the existing Oracle Access Manager agent.

You should start by defining the application details in PingAccess, including virtual host and path information, web session management controls and how end user identity details will be passed down to the application, if required (i.e., HTTP headers using identity mappings). The agent bootstrap configuration must be defined in PingAccess, and produces a configuration file the agent plugin will read on the target system.

Next, you define your access policies using rules that are equivalent to (or a simplification of) those employed in Oracle Access Manager.

Once configured, you uninstall the Oracle Access Manager agent on the target web server and install the PingAccess agent.

Documentation for the agents can be found here.

**Deploying New Applications**

Any new applications protected by the solution should be deployed behind PingAccess. They immediately benefit from the lighter weight integration and combined web and API access management features that PingAccess provides. While a gateway-based implementation is recommended to simplify ongoing maintenance, agents can be installed where required in order to accommodate existing deployment models.
Authentication Migration
The final phase of the migration is to remove any remaining Oracle Access Manager dependencies. If authentication responsibilities haven’t been transferred to PingFederate, finalize those changes first.

Move the authentication policies and methods to PingFederate in order to provide the same security, experience and functionality that existed with Oracle Access Manager.

Application Migration
As applications are migrated to PingAccess for protection, Oracle Access Manager agents could still be left in place. It’s critical to work with the application owners to remove those agents before Oracle Access Manager is decommissioned. As agents are removed from applications, configuration should be updated in PingAccess to provide the applications the necessary information.

When the final application requiring Oracle Access Manager OBSSO Tokens is migrated to PingAccess, you remove the token mediation configuration.

Final Migration Steps
The last steps are to remove any mechanisms for synchronizing sessions between Oracle Access Manager. Finally, Oracle Access Manager can be shut down and the servers can be recycled.

The migration to PingFederate and PingAccess is now complete. If you encounter problems or challenges, please contact support@pingidentity.com.
APPENDIX A: ORACLE ACCESS MANAGER APPLICATION CLASSIFICATIONS

Low Complexity Applications
Low complexity applications are protected by standard web/application agents and have the following characteristics:

- LDAP or Active Directory is typically used for authentication
- Form basic and certificate-based authentication schemes
- Simple realms, rules and policies with no custom authentication schemes/active expressions (such applications are easily migrated to PingAccess without any customizations)

Medium Complexity Applications
Medium complexity applications rely on minor Oracle Access Manager customization and have the following characteristics:

- Minimal Oracle Access Manager customizations, such as custom authentication schemes
- Oracle Access Manager authorization APIs called to perform authorization checks or using Oracle Access Manager APIs for responses (headers/cookies)

Custom code developed using the Oracle Access Manager SDK will need to be evaluated to determine if similar functionality is supported by PingAccess out-of-box functionality or can be developed using the PingAccess SDK.

High Complexity Applications
High complexity applications are similar to the medium complexity applications, but have more customization or specific integration requirements. High complexity applications have the following characteristics:

- More complex forms of authentication schemes like two-factor authentication, IWA, and impersonation
- Integration with applications like PeopleSoft, SAP, WebLogic, WebSphere, etc.

Custom code developed using Oracle Access Manager's SDK will have to be evaluated to determine if similar functionality is supported by PingAccess out-of-box functionality or can be developed using the SDK. While PingAccess provides a number of out-of-box connectors/adapters for various other tools that require SSO integration, the migration may require some additional planning.
Oracle Access Manager as the Authenticator

When Oracle Access Manager is used for user authentication, there are two request flows to understand. The first flow focuses on the user accessing an application protected by Oracle Access Manager before accessing an application protected by PingAccess. The second flow focuses on the user accessing an application protected by PingAccess and then accessing an application protected by Oracle Access Manager.

Scenario 1: User Accesses an Application Protected by Oracle Access Manager and then an Application Protected by PingAccess

The user initiates a web session by accessing a Oracle Access Manager protected application and then clicks on a URL protected by PingAccess. The following diagram shows the request flow:

Request Flow:

1. The user clicks a URL protected by Oracle Access Manager.
2. The agent intercepts the request and collects the credentials and authenticates the user with the policy server.
3. The user is then redirected to the protected web page. The agent creates an OBSSO cookie for the user.
4. The user now clicks a URL protected by PingAccess.
5. PingAccess redirects to PingFederate for authentication.
6. Integration with PingFederate (i.e., WAM Integration Kit integrated with Oracle Access Manager) decodes the OBSSO cookie and then validates the session with the policy server.
7. Upon successful validation, PingFederate performs an OpenID Connect based sign-on to PingAccess.
8. PingAccess validates the OpenID Connect sign-on and generates a PingAccess web session cookie (PA Token) during a redirection to the requested resource. All rules associated with the resource are evaluated for authorization, and if successful, the resource content is returned to the client.
Scenario 2: User Accesses an Application Protected by PingAccess and then an Application Protected by Oracle Access Manager

The user initiates the session by accessing a PingAccess protected application and then clicks on a URL protected by Oracle Access Manager. The following diagram depicts the request flow:

Request Flow:

1. The user accesses a URL protected by PingAccess.
2. PingAccess redirects to PingFederate for authentication.
3. Integration with PingFederate (i.e., WAM Integration Kit integrated with Oracle Access Manager) checks if there's an existing OBSSO cookie, and if there's not, it redirects the user to a URL protected by Oracle Access Manager for authentication.
4. The agent intercepts the URL and shows the sign-on page. The user enters credentials and is authenticated by the policy server. Oracle Access Manager then creates an OBSSO cookie.
5. PingFederate integration with Oracle Access Manager decodes the OBSSO cookie and completes an OpenID Connect based sign-on to PingAccess.
6. PingAccess validates the OpenID Connect sign-on and generates a PingAccess web session cookie (PA token) and evaluates all the rules associated with the resource for authorization.
7. The user now clicks a URL protected by Oracle Access Manager.
8. The agent intercepts and validates the OBSSO cookie with policy server.
9. If the OBSSO cookie is still valid and the user is authorized, the application content returned to the browser.

PingFederate as an authenticator

When PingFederate is used for authenticating users, the requirement is that users must interact with an application protected by PingAccess first. The request flow below shows that the authentication event places both the PingAccess JSON web token (JWT) and the Oracle Access Manager OBSSO token as cookies in the browser. After authentication, the user can access either a PingAccess protected application or a Oracle Access Manager protected application.
Scenario 3: User Accesses an Application Protected by PingAccess and then an Application Protected by Oracle Access Manager

The following steps describe how SSO is achieved between PingAccess and Oracle Access Manager when a user accesses a resource protected by PingAccess and authenticated by PingFederate.

Request Flow:

1. The user accesses a URL protected by PingAccess.
2. PingAccess redirects to PingFederate for authentication. The HTML Form Adapter (or other configured sign-on mechanism) presents the sign-on page. The user enters credentials and is authenticated.
3. PingFederate completes an OpenID Connect based sign-on to PingAccess and is redirected to protected resource.
4. The user now clicks a URL to a resource hosted on a web server with an Oracle Access Manager agent installed that is behind a PingAccess gateway.
5. The PingAccess gateway takes the PingAccess web session cookie (PA token) and sends it in a WSTrust STS request to PingFederate to request a Oracle Access Manager token (the OBSSO cookie).
6. The PingAccess gateway puts the OBSSO cookie in the HTTP request and then proxies it to the Oracle Access Manager protected resource.
7. The Oracle Access Manager agent checks the session with the policy server. Upon successful authentication and authorization, the web agent allows access to the resource.
8. The web server returns the resource to the browser.
APPENDIX C: SAMPLE ORACLE ACCESS MANAGER PROTECTED APPLICATION MIGRATION SCENARIO

The following table shows the list of Oracle Access Manager objects and the corresponding PingAccess/PingFederate objects:

<table>
<thead>
<tr>
<th>Oracle Access Manager</th>
<th>PingAccess</th>
<th>PingFederate</th>
<th>Remarks</th>
</tr>
</thead>
<tbody>
<tr>
<td>Agent</td>
<td>Agent/Sites</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Agent configuration object</td>
<td>Web Session</td>
<td></td>
<td>Not all settings are supported/required in PingAccess</td>
</tr>
<tr>
<td>Host configuration object</td>
<td>PingFederate server details</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Authentication requirements</td>
<td></td>
<td>Adapters, connections</td>
<td></td>
</tr>
<tr>
<td>User directories</td>
<td></td>
<td>Data stores, password credential validation</td>
<td></td>
</tr>
<tr>
<td>Password policies</td>
<td>n/a</td>
<td>n/a</td>
<td>PingFederate supports the underlying directory server password policy and can enforce reset passwords for expired password accounts and detect locked accounts. These features are supported by the PingFederate HTML Form Adapter1</td>
</tr>
<tr>
<td>Domains</td>
<td>Applications</td>
<td></td>
<td>All protected sites are independent in PingAccess</td>
</tr>
<tr>
<td>Realms, sub-realms</td>
<td>Resources</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Rules</td>
<td>Rules, rule sets</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Responses</td>
<td>Identity mapping</td>
<td>OpenID Connect policies</td>
<td></td>
</tr>
<tr>
<td>Policies</td>
<td>Rules</td>
<td></td>
<td>Issuance criteria in access token and OpenID Connect policies</td>
</tr>
</tbody>
</table>
APPENDIX D: MAPPING OAM LDAP AUTHORIZATION RULES TO PING ACCESS GROOVY SCRIPTS

Oracle Access Manager 10/11g provides authorization rules based on standard LDAP query to determine the user access to resources.

For example, the following OAM LDAP authorization rule allows and authorizes users who are either with an orgRole that contains a string "ABC1000" or orgRole "XYZ1000". ldap:///o=test.com??sub?([(orgrole="ABC1000")|(orgrole="XYZ1000")]). Ping Access does provide an equivalent groovy script for the same authorization rule.

/* Groovy Script will allow the user access to the resource only if the orgRole contains "ABC1000" or "XYZ1000" */

import com.fasterxml.jackson.databind.node.ArrayNode
userContext = exc?.user?.userContext
ArrayNode orgrole = userContext?.get("orgrole")
if(testForValue(orgrole,"ABC1000") || testForValue(orgrole,"XYZ1000") )
{
  anything()
}
else{
  not(anything())
}

boolean testForValue(ArrayNode list, String valueToTest)
{
  exc?.log.debug "In method testForValue"
  iterator = list.iterator()
  while (iterator.hasNext())
  {
    value = iterator.next().asText()
    exc?.log.debug " Value: " + value
    if (value.toString().contains(valueToTest))
    {
      exc?.log.debug "returning true"
      return true
    }
  }
  not(anything())
}

ABOUT LIKEMINDS CONSULTING: LikeMinds consulting is a leading provider of consulting, systems integration and managed services and focuses on Identity Management, Application Security, Governance, Risk and Compliance solutions. We have focused on providing our customers with a full range of services which span through our core beliefs of Advising, Integrating, Maintaining and Accelerating the Complete Identity and Security Solution. Visit http://likemindsconsulting.com/