**AWS IAM Identity Centre (SSO) & Active Directory Integration**

**Table of Contents**

1. **Introduction to AWS IAM Identity Center**
   * What is IAM Identity Center?
   * Why Use It?
2. **Architecture Overview**
   * Active Directory (On-Prem/EC2)
   * AWS Managed Microsoft AD
   * AD Connector
   * IAM Identity Center Integration
3. **Step-by-Step Implementation**
   * Setting Up AWS Managed Microsoft AD
   * Configuring Windows Instance for AD Sync
   * Creating Users in Active Directory
   * Enabling IAM Identity Center
   * Assigning Permission Sets
4. **Logging In via SSO Portal**
   * Multi-Factor Authentication (MFA) Setup
   * Testing Access
5. **Bonus: AD Connector Setup**
   * Connecting On-Prem AD to AWS
6. **Conclusion & Best Practices**

**1. Introduction to AWS IAM Identity Center**

**What is IAM Identity Center?**

* Formerly **AWS SSO**, it centralizes access management across **multiple AWS accounts**.
* Integrates with:
  + **AWS Managed AD**
  + **Self-hosted Active Directory (on EC2/on-prem)**
  + **External identity providers (e.g., Okta, Azure AD)**

**1. 🏢 AWS Managed Microsoft AD**

* **What it is**: A fully managed Active Directory service hosted on AWS.
* **Use Case**:
  + Organizations that want AD **without managing servers**.
  + Supports **Kerberos, LDAP, and Group Policy**.
* **Setup**:

plaintext

AWS Directory Service → Setup Directory → AWS Managed Microsoft AD

* **Example**: Sync corporate users from corp.example.com to AWS SSO.

**2. 🖥️ Self-Hosted Active Directory (EC2/On-Prem)**

* **What it is**: Traditional AD running on:
  + **EC2 Windows Server** (AWS cloud) or
  + **On-premises servers**.
* **Use Case**:
  + Companies with **existing AD infrastructure**.
  + Hybrid cloud setups (e.g., AD on-prem + AWS resources).
* **Integration Methods**:
  + **AD Connector** (Proxy to on-prem AD).
  + **Manual sync** using AWS IAM Identity Center.
* **Example**: Sync on-prem AD users (onprem.example.com) to AWS SSO.

**3. 🌐 External Identity Providers (Okta, Azure AD, etc.)**

* **What it is**: Connect to **third-party identity providers** (IdPs).
* **Supported IdPs**:
  + **SAML 2.0-compliant** providers (Okta, Azure AD, OneLogin).
  + **Social logins** (Google Workspace).
* **Use Case**:
  + Companies using **Okta/Azure AD for SSO**.
  + Avoid managing users in AWS directly.
* **Setup**:

plaintext

IAM Identity Centre → Settings → Identity Source → External Identity Provider

* **Example**: Allow employees to log in to AWS using their **Okta credentials**.

**🛠️ Key Integration Features**

| **Feature** | **AWS Managed AD** | **Self-Hosted AD** | **External IdP** |
| --- | --- | --- | --- |
| **Maintenance** | AWS-managed | You manage | IdP-managed |
| **Authentication** | Kerberos/LDAP | Kerberos/LDAP | SAML/OAuth |
| **Best For** | Cloud-native AD | Hybrid setups | Existing IdP users |

**Why Use It?**

✅ **Single sign-on (SSO)** for AWS accounts.  
✅ **No manual role switching** (unlike cross-account roles).  
✅ **Scalable** for organizations with **1000s of users**.

**2. Architecture Overview**

**Components**

1. **Self-Hosted AD (EC2 Windows Server)**
   * Manages local users/groups (e.g., admin@corp.sap.com).
2. **AWS Managed Microsoft AD**
   * Fully managed Active Directory service (corp.sap.com).
3. **AD Connector**
   * Links **on-prem AD** to AWS (hybrid setup).
4. **IAM Identity Center**
   * Syncs users from AD → AWS SSO portal.

**3. Step-by-Step Implementation**

**Step 1: Set Up AWS Managed Microsoft AD**

1. Go to **AWS Directory Service → Set up directory → AWS Managed Microsoft AD**.
2. Enter:
   * **Directory name**: corp.sap.com
   * **Admin password**: Secure password for Admin user.
3. Deploy in **2 subnets** (for HA).

**Step 2: Configure Windows Instance for AD Sync**

1. Launch a **Windows EC2 instance** (non-AD).
2. Set **DNS servers** to AWS Managed AD IPs (found in Directory Service console).

powershell

ncpa.cpl → IPv4 → DNS: 10.0.209.19, 10.0.80.184

1. **Join the domain**:

powershell

System → Advanced → Domain → corp.sap.com

* + Use Admin credentials from AWS Managed AD.

**Step 3: Create Users in Active Directory**

1. Open dsa.msc (AD Users & Computers).
2. Create users (e.g., aish@corp.sap.com) with:
   * **Email** (required for SSO sync).
   * **Password** (meets complexity rules).
3. Add users to groups (e.g., AWS-Admins).

**Step 4: Enable IAM Identity Center**

1. Go to **IAM Identity Center → Enable**.
2. Under **Identity source → Actions → Change identity source → AWS Managed AD**.
3. Select corp.sap.com → **Accept terms**.

**Step 5: Assign Permission Sets**

1. Go to **AWS accounts → Assign users/groups**.
2. Select users (aish, mohit) → **Next**.
3. Create **Permission Set** (e.g., AdministratorAccess).
4. **Submit** to sync permissions.

**4. Logging In via SSO Portal**

**Accessing the Portal**

1. Navigate to your **SSO portal URL** (e.g., https://d-123456.awsapps.com/start).
2. Log in as aish@corp.sap.com.

**Setting Up MFA**

* Choose **Authenticator app** (e.g., Microsoft Authenticator).
* Scan QR code → Enter OTP.

**Testing Access**

* After login, select **AWS account → AdministratorAccess**.
* Verify permissions (e.g., create S3 buckets, EC2 instances).

**5. Bonus: AD Connector Setup**

**For On-Prem AD Integration**

1. Go to **Directory Service → AD Connector**.
2. Enter:
   * **DNS name**: onprem.ad.sap.com
   * **On-prem AD IPs** (from your DC).
   * **Service account**: onprem\admin
3. **Sync users** to IAM Identity Center.

**6. Conclusion & Best Practices**

**Key Takeaways**

✔ **Centralized access** for multi-account environments.  
✔ **No manual role switching** required.  
✔ **MFA enforced** for security.

**Best Practices**

* **Least privilege**: Avoid AdministratorAccess in production.
* **Regularly audit** permission sets.
* **Use AD groups** for bulk assignments.

**Next Steps**: Explore **GuardDuty & KMS** for enhanced security! 🚀