# Practical No 8

# Aim: Study and Implementation of Single Sign-On

# Theory

# What Is AWS Single Sign-On?

AWS Single Sign-On is a cloud-based single sign-on (SSO) service that makes it easy to centrally manage SSO access to all of your AWS accounts and cloud applications. Specifically, it helps you manage SSO access and user permissions across all your AWS accounts in AWS Organizations. AWS SSO also helps you manage access and permissions to commonly used third-party software as a service (SaaS) application, AWS SSO-integrated applications as well as custom applications that support Security Assertion Markup Language (SAML) 2.0. AWS SSO includes a user portal where your end-users can find and access all their assigned AWS accounts, cloud applications, and custom applications in one place.

AWS SSO provides the following features:

**Integration with AWS Organizations**

AWS SSO is integrated deeply with AWS Organizations and AWS API operations, unlike other cloud native SSO solutions. AWS SSO natively integrates with AWS Organizations and enumerates all your AWS accounts. If you have organized your accounts under organizational units (OUs) you will see them displayed that way within the AWS SSO console. That way you can quickly discover your AWS accounts, deploy common sets of permissions, and manage access from a central location.

**SSO access to your AWS accounts and cloud applications**

AWS SSO makes it simple for you to manage SSO across all your AWS accounts, cloud applications, AWS SSO-integrated applications, and custom SAML 2.0–based applications, without custom scripts or third-party SSO solutions. Use the AWS SSO console to quickly assign which users should have one-click access to only the applications that you've authorized for their personalized end-user portal.

**Create and manage users and groups in AWS SSO**

When you enable the service for the first time, we create a default store for you in AWS SSO. You can use this store to manage your users and groups directly in the console. Or, if you prefer, you can connect to an existing AWS Managed Microsoft AD directory and manage your users with standard Active Directory management tools provided in Windows Server.

Before you can set up AWS SSO, you must:

* Have first set up the AWS Organizations service and have **All features** set to enabled. For more information about this setting, see [Enabling All Features in Your Organization](https://docs.aws.amazon.com/organizations/latest/userguide/orgs_manage_org_support-all-features.html) in the *AWS Organizations User Guide*.
* Sign in with the AWS Organizations master account credentials before you begin setting up AWS SSO. These credentials are required to enable AWS SSO. For more information, see [Creating and Managing an AWS Organization](http://docs.aws.amazon.com/organizations/latest/userguide/orgs_manage_org.html) in the *AWS Organizations User Guide*. You cannot set up AWS SSO while signed in with credentials from an Organization’s member account.
* Have chosen an identity source to determine which pool of users has SSO access to the user portal. If you choose to use the default AWS SSO identity source for your user store, no prerequisite tasks are required. The AWS SSO store is created by default once you enable AWS SSO and is immediately ready for use. There is no cost for using this store. If you choose to connect to an existing Active Directory for your user store, you must have the following:
  + An existing AD Connector or AWS Managed Microsoft AD directory set up in AWS Directory Service, and it must reside within your organization's master account. You can connect only one AWS Managed Microsoft AD directory at a time. However, you can change it to a different AWS Managed Microsoft AD directory or change it back to an AWS SSO store at any time. For more information, see [Create a AWS Managed Microsoft AD Directory](http://docs.aws.amazon.com/directoryservice/latest/admin-guide/create_directory.html) in the *AWS Directory Service Administration Guide*.
  + You must set up AWS SSO in the Region where your AWS Managed Microsoft AD directory is set up. AWS SSO stores the assignment data in the same Region as the directory. To administer AWS SSO, you should switch to the Region where you have setup AWS SSO. Also, note that AWS SSO’s user portal uses the same [access URL](http://docs.aws.amazon.com/directoryservice/latest/admin-guide/access_url.html) as your connected directory.

Enable AWS SSO

When you open the AWS SSO console for the first time, you are prompted to enable AWS SSO before you can start managing it. If you have already chosen this option, you can skip this step. If not, use the procedure below to enable it now. Once enabled, AWS SSO is granted the necessary permissions to create IAM service-linked roles in any of the AWS accounts within your AWS organization. No service-linked roles are created at this time. AWS SSO creates these roles later during the process of setting up SSO access to your AWS accounts.

**To enable AWS SSO**

1. Sign in to the AWS Management Console with your AWS Organizations master account credentials.
2. Open the [AWS SSO console](https://console.aws.amazon.com/singlesignon).
3. Choose **Enable AWS SSO**.
4. If you have not yet set up AWS Organizations, you will be prompted to create an organization. Choose **Create AWS organization** to complete this process.

## **User name and email address uniqueness**

When working in AWS SSO, users must be uniquely identifiable. AWS SSO implements a user name that is the primary identifier for your users. Although most people set the user name equal to a user’s email address, AWS SSO and the SAML standard do not require this. However, a large percentage of SAML-based applications use an email address as the unique identifier for users. They obtain this from assertions that a SAML identity provider sends during authentication. Such applications depend upon the uniqueness of email addresses for each user. As such, AWS SSO allows you to specify something other than an email address for user sign-in. AWS SSO requires that all user names and email addresses for your users are non-NULL and unique.

## **Groups**

Groups are a logical combination of users that you define. You can create groups and add users to the groups. AWS SSO does not support adding a group to a group (nested groups). Groups are useful when assigning access to AWS accounts and applications. Rather than assign each user individually, you give permissions to a group. Later, as you add or remove users from a group, the user dynamically gets or loses access to accounts and applications that you assigned to the group.

## **User and group provisioning**

You can create users and groups directly in AWS SSO, or work with users and groups you have in Active Directory or an external identity provider. In order for AWS SSO to assign users and groups for permissions in an AWS SSO account, AWS SSO must first be aware of the users and groups. Similarly, AWS SSO-integrated applications can work with users and groups for which AWS SSO is aware. Provisioning is the process of making user and group information available for use by AWS SSO and AWS SSO-integrated applications.

# SAML Federation

AWS SSO supports identity federation with [SAML (Security Assertion Markup Language)](https://wiki.oasis-open.org/security) 2.0. SAML 2.0 is an industry standard used for securely exchanging SAML assertions that pass information about a user between a SAML authority (called an identity provider or IdP), and a SAML consumer (called a service provider or SP). AWS SSO service uses this information to provide federated single sign-on (SSO) for those users who are authorized to use applications within the AWS SSO user portal.

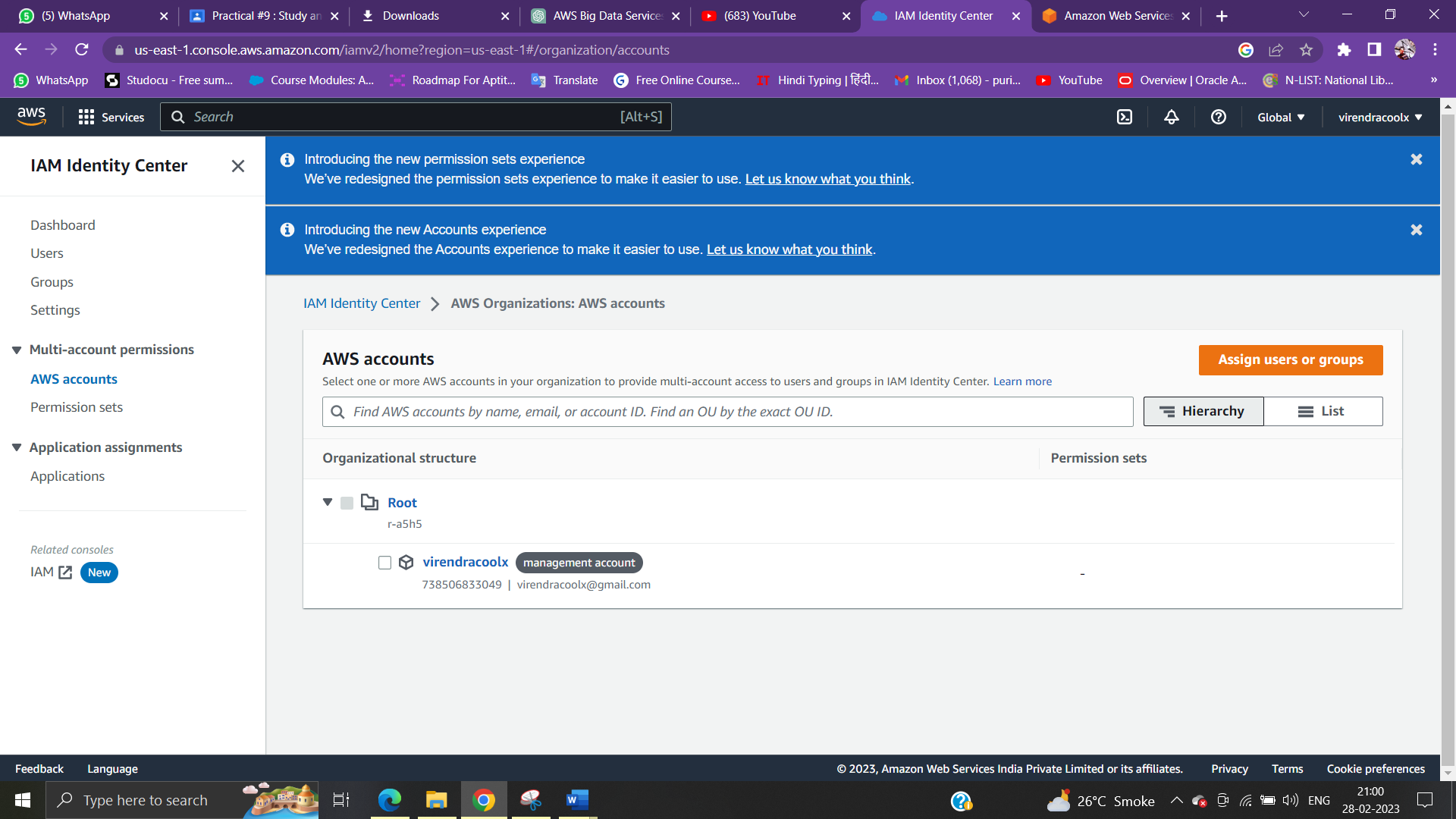
# User Authentications

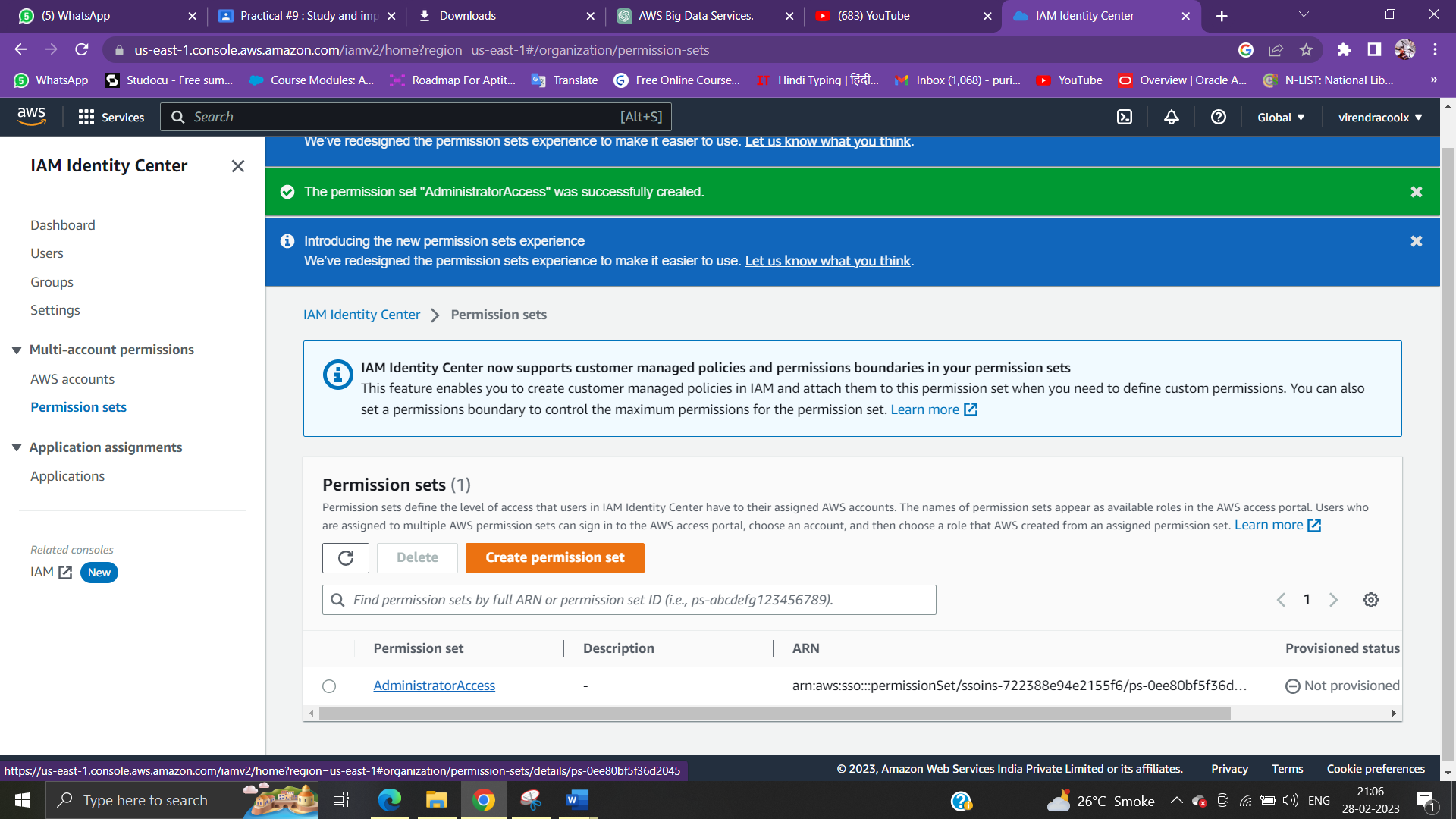
A user signs in to the user portal using their user name. When they do, AWS SSO redirects the request to the AWS SSO authentication service based on the directory associated with the user email address. Once authenticated, users have SSO access to any of the AWS accounts and third-party software-as-a-service (SaaS) applications that show up in the portal without additional sign-in prompts. This means that users no longer need to keep track of multiple account credentials for the various assigned AWS applications that they use on a daily basis.

# Permission Sets

A permission set is a collection of administrator-defined policies that AWS SSO uses to determine a user's effective permissions to access a given AWS account. Permission sets can contain either [AWS managed policies](http://docs.aws.amazon.com/IAM/latest/UserGuide/access_policies_managed-vs-inline.html#aws-managed-policies) or custom policies that are stored in AWS SSO. Policies are essentially documents that act as containers for one or more permission statements. These statements represent individual access controls (allow or deny) for various tasks that determine what tasks users can or cannot perform within the AWS account.

Permission sets are stored in AWS SSO and are only used for AWS accounts. They are not used to manage access to cloud applications. Permission sets ultimately get created as [IAM roles](http://docs.aws.amazon.com/IAM/latest/UserGuide/id_roles.html) in a given AWS account, with trust policies that allow users to assume the role through AWS SSO.





URL Portal:

https://virendracoolx.awsapps.com/start

