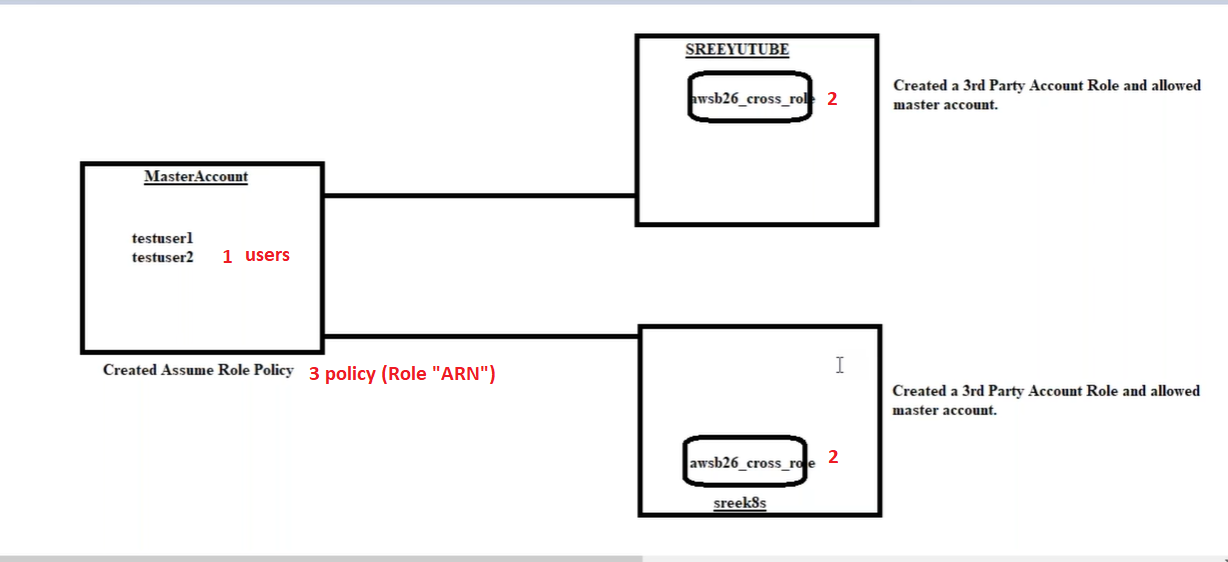
**Role switching:**

**Role is not only allowing access between two AWS services, but also used to switch roles between two Root accounts.**

**Is to give access all IAM users of “main account” to “sub account”, without creating users again in sub accounts.**



**Lab**:

1, Main Account: (dheerajpalvai@gmail.com)

Testuser1

Testuser2

IAM -> Users -> Add user -> User name "TestUser1" & "TestUser2"

2, Sub Account1: ([dheeraj.nc24@gmail.com](mailto:dheeraj.nc24@gmail.com))

* Create the role that trust Master account
* Allow Main account id in Sub Account1 Role.
* Role 3rd party account role (allow Main account id, and give “AdministratorAccess” of Sub Account1 to IAM users in main account)

IAM -> Roles -> Create role -> "Another AWS account Belonging to you or 3rd party" -> Account ID "Main account id no. where users there" -> AdministratorAccess -> Role name "Aws\_cross\_RoleSwitching"

🡪Goto "Aws\_cross\_RoleSwitching" role and copy

arn:aws:iam::989878044275:role/Aws\_cross\_RoleSwitching

now role is created by allowing Main account id

3, Sub Account2:

Role 3rd party account role (allow Main account id, by giving admin access of Sub Account2 to users)

1, Main Account:

* Now allow sub account role “ARN” in Main account, by writing Policy and adding that policy to IAM user

Group the users and “Create assume role policy”- that allows Sub Account’s 3rd party account role ARN::

* IAM -> Policies -> Create policy -> JSON -> past code -> Name: “Cross\_Account\_policy”

Google: aws assume role policy

{

    "Version": "2012-10-17",

    "Statement": {

        "Effect": "Allow",

        "Action": "sts:AssumeRole",

        "Resource": "arn:aws:iam::989878044275:role/Aws\_cross\_RoleSwitching"**---> (Sub account1 Role ARN)**

    }

}

🡺 Add this policy to test users’ group

IAM -> Groups -> Attach Policy "Cross\_Account\_policy"

**Workflow**: [**Main account**(IAMusers) – **Sub account**(Role{Main account id}==ARN) – **Main account**(ARN in Policy) – **Main account**(user attach policy]

**==>** Login to TestUser1 --- Account -> IAM -> Dashboard -> URL

* TestUser1 login as Main account

Testuser1- > Switch Roles (to sub account1) ->

Account\*: 989878044275 (sub account id)

Role\*: Aws\_cross\_RoleSwitching (sub account role, that allowed main account)

Display Name: TestUser1\_dheeraj.nc24

* Now it is switched IAM Test User From “**Main account**” to “**sub account**”

But this is not effective **method**, we get productivity issues because of too many accounts

And all listed in TestUser1

-------------------------------------------------------------------------------------------------------------------------------

**Another Method** is Active Directory

Active Directory:

By this Active Directory method, we no need to create IAM users separately

We create Active Directory and use as share folder.

We create “Active Directory” in windows server and assume it as on premises server, and using “AD Connector” we connect “Active Directory” in windows server to AWS

**Lab**:

1, deploy Ec2 windows server

* EC2 -> Instances -> Launch instances -> "Microsoft Windows Server 2019 Base"
* Then Ec2 -> RC -> Security -> Get Windows password -> AWSB26-VPC-key pair.pem (and copy password “2vWSe=jI%@OmwNKD7Hyvyr&=(BqbYDcq”)
* Connect -> RDP client -> Download remote desktop file -> give password (logged into server)

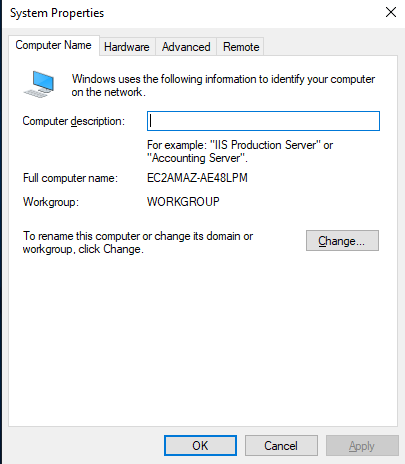
Password change

Now goto “local users and groups” for changing “.pem” generated password

Run -> lusrmgr.msc -> Users -> RC on “Administrator” -> Set password (06c61@0542)

------------------------------------------------------ system properties-------------------------------------------

Before creating Active Directory, go to system properties Run -> sysdm.cpl

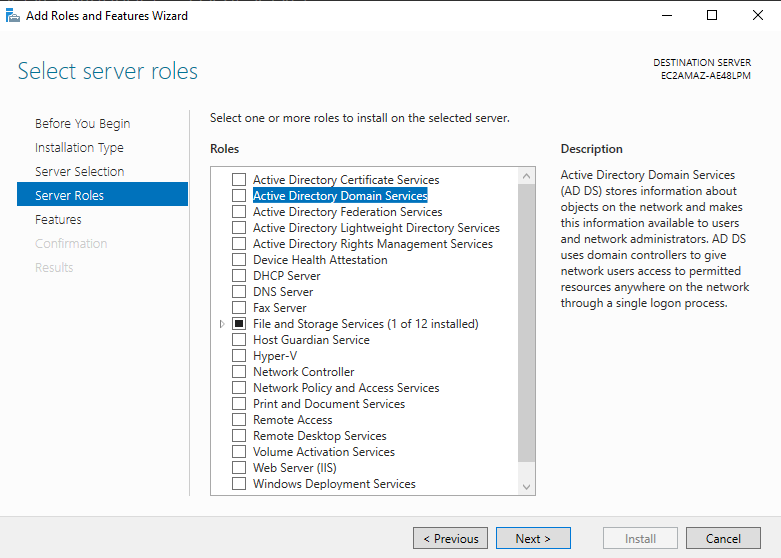


After installing Active Directory “Full computer name & Workgroup” changes with Ec2 Server “Tag” key & Value

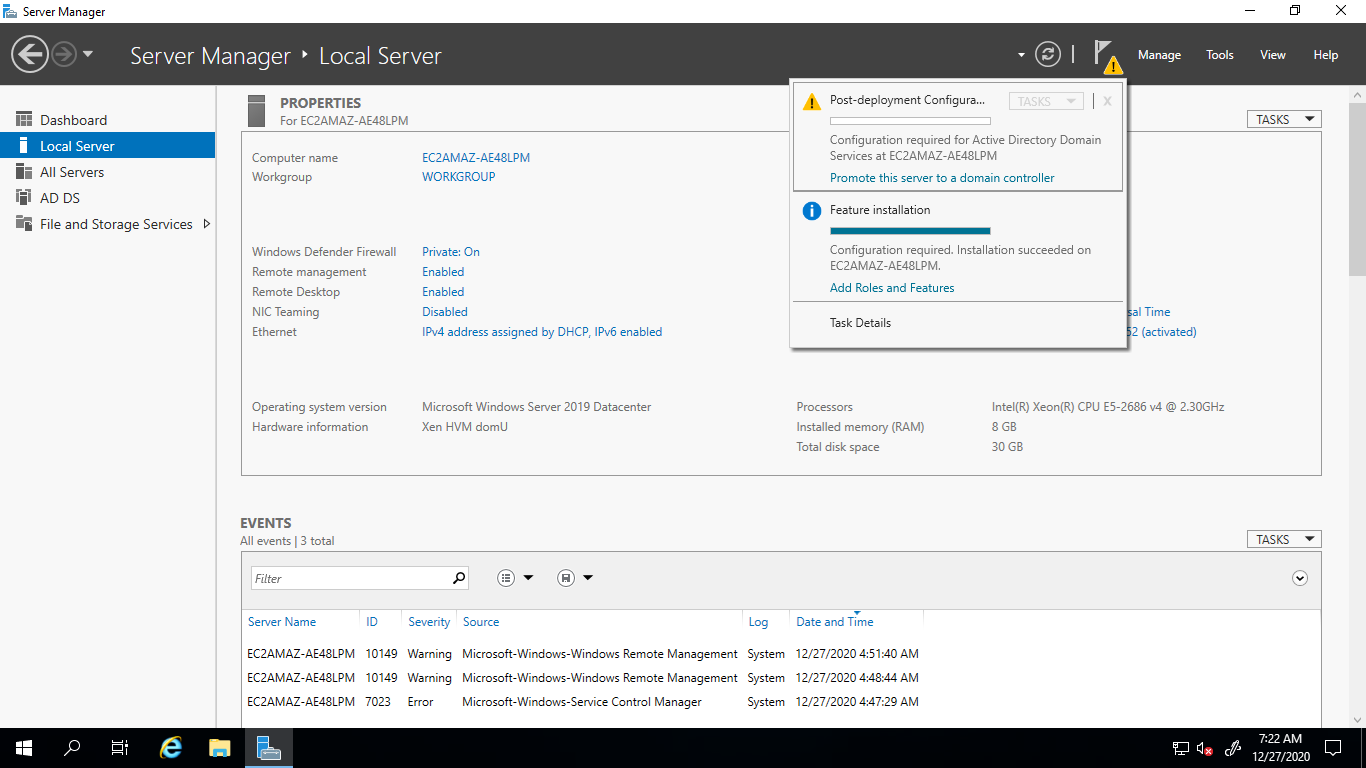
--------------------------------------------------------------------------------------------------------------------------

2, Create Active Directory

* Now go to windows -> system manager -> Local server -> click on “IE Enhanced Security Configuration: On” **On** link -> off & off
* then on top, click on "Manage" -> "Add Roles and Features" -> -> -> goto “Active Directory Domain Services” -> install



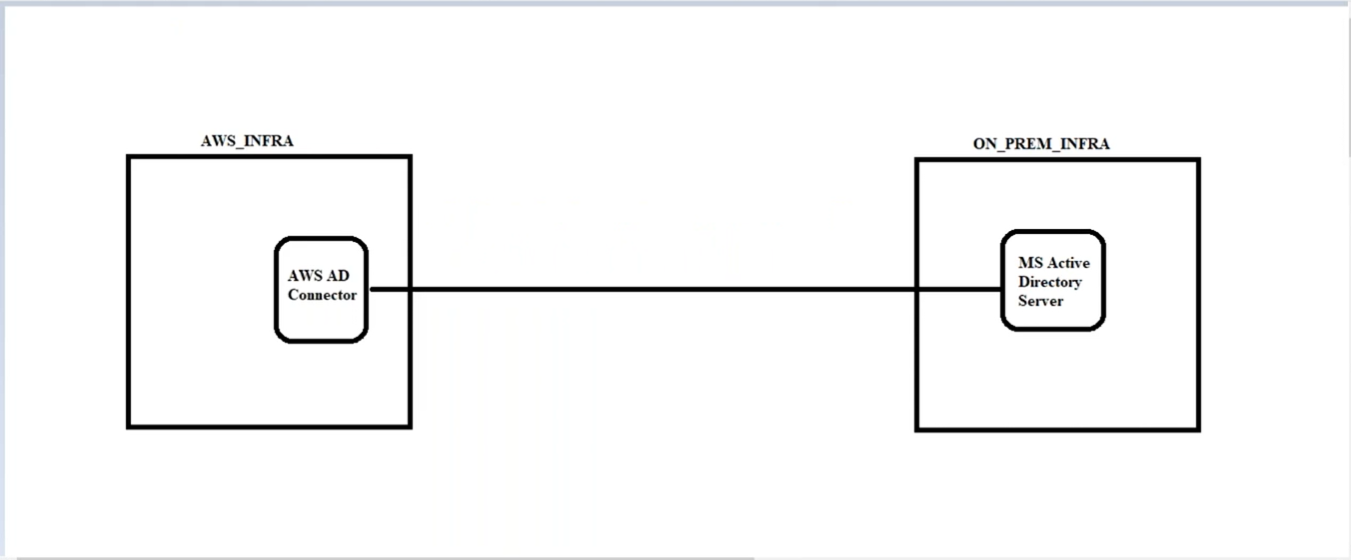
Then go to



Click on “Promote this server to a domain controller” -> Add a new forest -> Root domain name: dheerajpalvai.xyz -> Password: 06c61@0542 -> -> install (it will reboot)

------------------------------------------------------------------------------------------------------------------------------------------

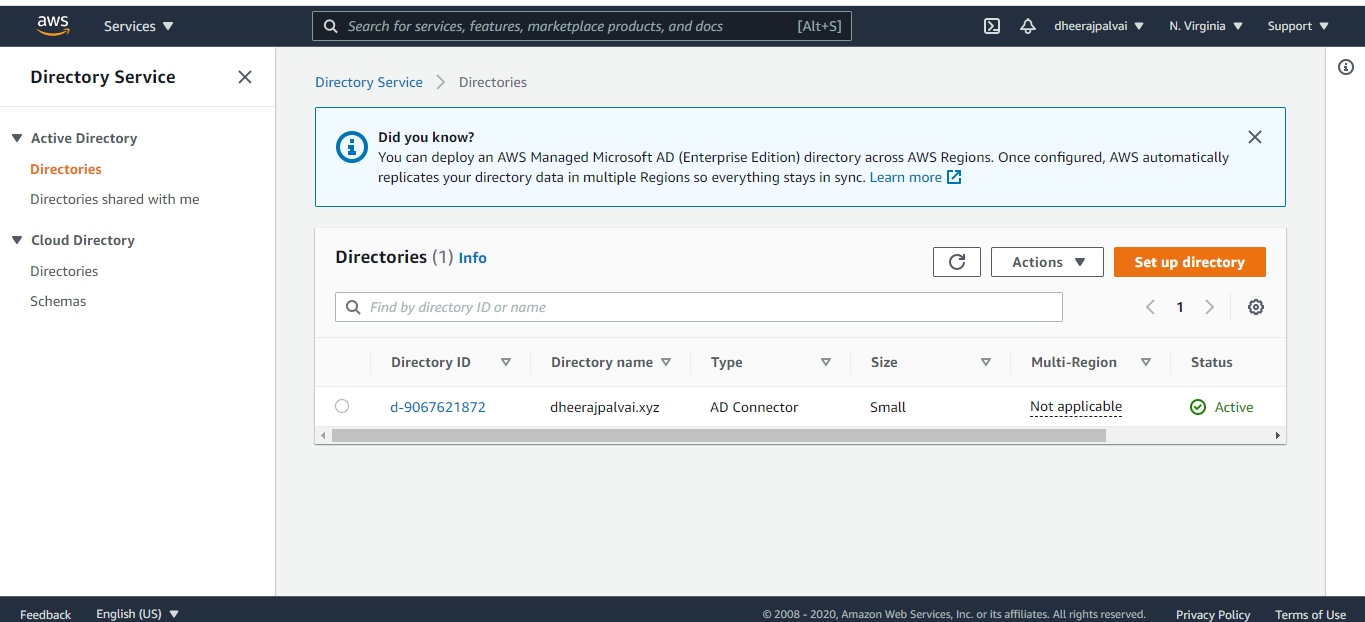
🡺 Active Directory connector in AWS will connect to Microsoft Active Directory Server in ON\_premisis



3, Directory Service

* under “Services” -> Security, Identity, & Compliance -> Directory Service -> Directories -> Set up a directory -> AD Connector -> small -> VPC & 2 Subnets ->

Directory DNS name: dheerajpalvai.xyz -> DNS IP addresses: 10.0.1.94 (Ec2 private ip) -> Service account username: Administrator -> Service account password & Confirm password: 06c61@0542



* Now “AD Connector” ready now, we sync “Active Directory” in On-premises with AWS using “AD Connector”

4, AWS Single Sign-On: Permissions

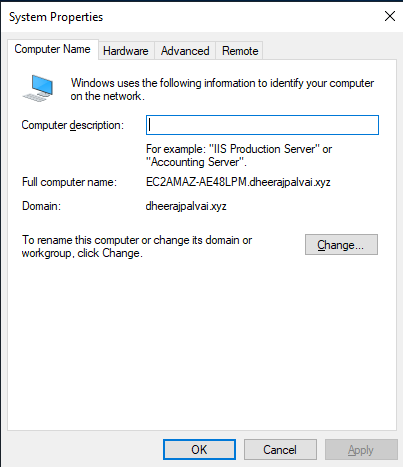
Giving permissions to AD using single sign-on

Under "Services" -> Security, Identity, & Compliance -> AWS Single Sign-On

* Now we should connect(integrate) “Directory Service(AWS account)” to “Active Directory” in “AWS Single Sign-On” [as shown in design]
* Under "Services" -> Security, Identity, & Compliance -> AWS Single Sign-On -> Settings -> click on change “**Identity source:** AWS SSO | [Change](https://console.aws.amazon.com/singlesignon/home?region=us-east-1#/settings$identitySourceWizard)” -> select Active Directory “Active Directory” -> connect “**Existing directories”** which we created in “Directory Service” -> connected.

------------------------------------------------------ system properties -------------------------------------------

Now Again Go to windows Ec2 server and check system properties, go to system properties Run -> sysdm.cpl

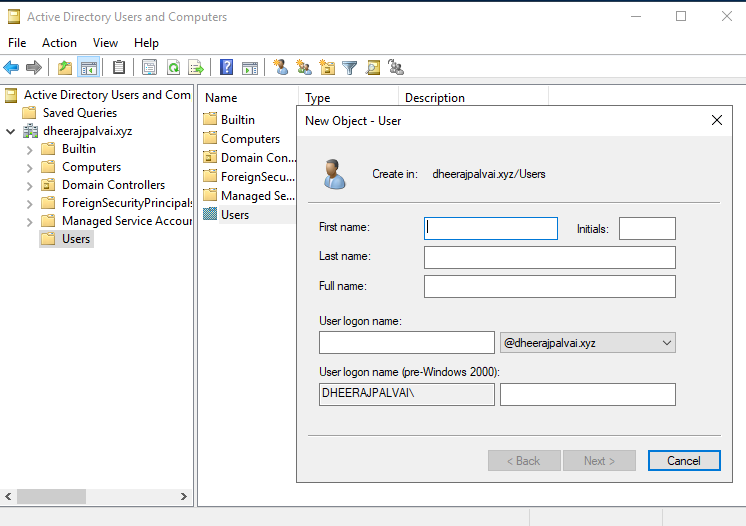


Now “Full computer name & Workgroup” changes

---------------------------------------------------------------------------------------------------------------------

5, Create Users in Active Directory

Ec2 open windows server-> (Run -> dsa.msc) or (windows -> Windows Administrative Tools -> “Active Directory Users and Computers” opens -> click on domain name “dheerajpalvai.xyz” -> Users -> RC -> New -> user



Create 3 users:

Full name: Anand & Bala & Candra

User logon name: Anand & Bala & Candra

4, AWS Single Sign-On: Permissions

I,AWS accounts -> Permission sets -> Create permission set -> Create a custom permission set -> Name: Ec2fullaccess -> Attach AWS managed policies -> [AmazonEC2FullAccess](https://console.aws.amazon.com/iam/home#/policies/arn:aws:iam::aws:policy/AmazonEC2FullAccess)

AWS accounts -> Permission sets -> Create permission set -> Create a custom permission set -> Name: s3fullaccess -> Attach AWS managed policies -> [AmazonS3FullAccess](https://console.aws.amazon.com/iam/home#/policies/arn:aws:iam::aws:policy/AmazonS3FullAccess)

Ii, AWS accounts -> AWS organization (we can see all Root Accounts, because we have included all sub accounts to main account) -> select Subaccount to access users with permissions -> Assign users -> Users -> search for users “Anand & bala & candra”

Iii, now we need portal or URL to login with users

Settings -> User portal-> User portal URL <https://d-9067621872.awsapps.com/start>

AWS management console opens -> give username & password -> click on account -> “Management console”

**Workflow**: [**Ec2 Windows** – **Ec2 Windows** (Create Active Directory) – **Ec2 Windows** (Create Users) – **Directory Service** (under IAM- Create Directory Service {adopts Ec2 windows AD}) – **AWS Single Sign-On** (under IAM – Connects to Directory Service **&** add users and create permissions)]