**Filesystems on AWS**

**LINUX AD INTEGRATION - SSSD**

**Native Authentication**

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**Version 1.01**

## INITIALS

Linux AD integration can be achieved with Native Authentication using Kerberos through SSSD configuration.

SSSD controls access to the Linux servers for both groups and users. It also ensures home directory is created once the user authenticate to the Linux environment.

SSSD does not controls privileges and permissions for users. This is achieved using Sudo files and this can be integrated with values from SSSD into sudo file.

## PROCESS

**First Step:**

The below packages would be installed on the Linux Server.

* sssd
* realmd
* authconfig
* adcli
* oddjob-mkhomedir
* oddjob
* samba-common-tools

SSSD uses realm which uses adcli to join Linux servers to AD.

Service account with **limited permission** is needed to complete the join.

Service account permission: **Able to join computer object privileges**

Once the above steps is completed, proceed to step 2

**Second Step:**

Run the below command:

**realm join -U <service account> awva-pcwad00993.tfs.toyota.com**

This would prompt for password.

If running as script, command can be written as;

**echo '<password>'| realm join -U <service account> awva-pcwad00993.tfs.toyota.com**

Once Linux AD integration is completed, sssd.conf file would be generated.

**Third Step:**

Modify the sssd conf file with the highlighted information.

In the sssd.conf file, is where you specify the access or group to be allowed.

/etc/sssd/sssd.conf

[sssd]

domains = TFS.Toyota.com

config\_file\_version = 2

services = nss, pam

[domain/TFS.Toyota.com]

ad\_server = awva-pcwad00993.tfs.toyota.com

ad\_domain = TFS.Toyota.com

krb5\_realm = TFS.TOYOTA.COM

realmd\_tags = manages-system joined-with-adcli

cache\_credentials = True

id\_provider = ad

krb5\_store\_password\_if\_offline = True

default\_shell = /bin/bash

ldap\_id\_mapping = True

use\_fully\_qualified\_names = **False**

fallback\_homedir = **/export/home/%u**

access\_provider = ad

[domain/tfs,toyota.com]

access\_provider = simple

#simple\_allow\_users = **user\_adm**

simple\_allow\_groups = **tfs\_unix\_dne**

Restart SSSD service and ensure sssd service is enabled

systemctl start sssd

systemctl enable sssd

With this completed, users allowed should be able to authenticate into the Linux servers.

**Configuring Sudo file for Permission**

Configuring Permission in sudo can be seamlessly easy to complete by using Cmnd\_Alias

/etc/sudoers

Configure with Group such as

%tfs\_unix\_dne ALL=(ALL) ALL

or Users such as

owosenj\_adm ALL=(ALL) ALL

Privileges can be reduced by using Cmnd\_Alias such as

Cmnd\_Alias BIDEPLOYMENT = /usr/bin/docker, /usr/bin/yum

and the line would be

%tfs\_unix\_dne ALL=(ALL) NOPASSWD: sudo su - <service acct>

**Please Note:** SSSD configuration and sudo privileges can be achieved through automation using Chef or Ansible and Puppet if ever introduced at TFS.