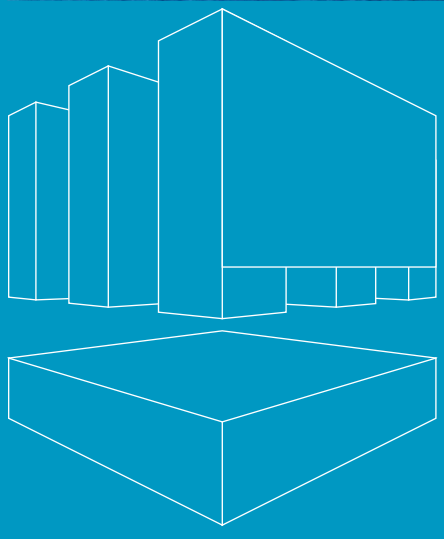
HELECLOUD

Centralised Authentication within AWS

Jon Southby
2019



HELECLOUD

Table of Contents

HELECLOUD

Abstract

Whilst startups can easily deploy serverless or immutable infrastructure using infrastructure as code, more established companies sometimes have to take a more traditional approach with a lift and shift of their existing virtual machines and applications into AWS. In this lift and shift approach the question then arises: "how do we setup a centralised authentication of those Linux and Windows EC2 instances?". AWS has a number of services that can help here, namely Simple AD which is one of 2 Directory Services offerings and EC2 Systems Manager which offers a couple of alternatives to logging directly into an instance, in the form of the Run Command, and the recently released Session Manager.

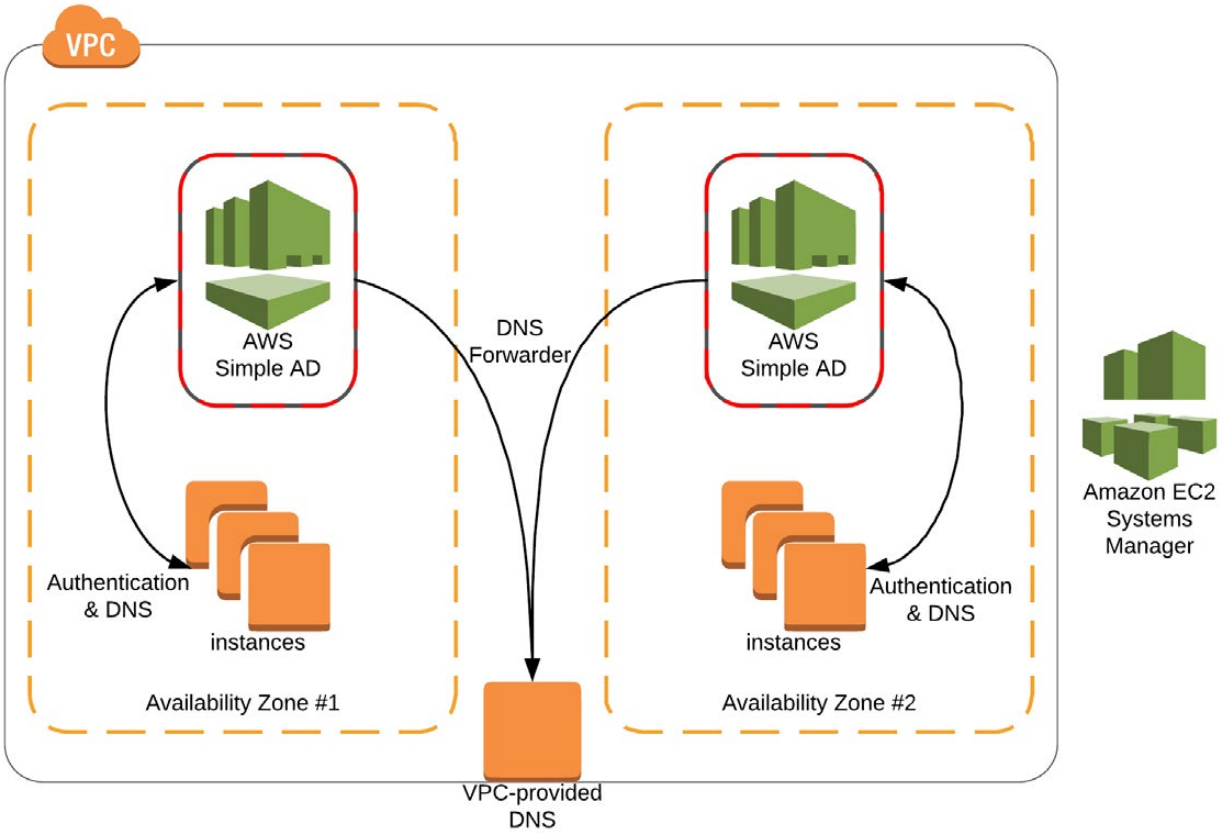


Figure 1 – Simple AD and EC2 Systems Manager

HELECLOUD

Background & Problem Statement

By default, in AWS when an EC2 instance is launched, it has a single key pair that you must use to log in to the instance securely. This presents a number of security issues:

- All users are using a single set of shared credentials
- There is no audit trail of individual access or actions carried out
- All users have the same level of access – Administrative

In order to control access at a more granular level and ensure that sessions are auditable, further controls are required. In AWS, we can use:

- Simple AD to provide centralised authentication and Simple AD groups to control the level of access. Additionally, with a Simple AD schema extension, users can authenticate (to linux instances) using public/private key authentication and be prompted for a password to use sudo.
- AWS Systems Manager, which provides two different approaches

    1. Run command can be used to run a shell script on a Linux instance or a Windows PowerShell script on a Windows instance
    2. Session Manager provides secure shell/PowerShell command line access to instances

We shall now explore each approach in detail.

HELECLOUD

Simple AD

AWS Simple AD is a standalone managed directory that is powered by a Samba 4 Active Directory (LDAP) Compatible Server. It provides a subset of the features offered by AWS Managed Microsoft AD. It also provides the ability to manage user accounts and group membership, create and apply group policies, securely connect to Amazon EC2 instances, and provide Kerberos-based single sign-on (SSO).

Simple AD is available in two sizes

- Small (upto 2,000 objects, including ~500 users, groups and computers)
- Large (upto 200,000 objects, including ~5,000 users, groups and computers)

NOTE: Once deployed the directory size cannot be modified

Prior to deployment the directory also needs

- a fully qualified domain name (FQDN)
- a NETBIOS name
- an Administrator password

Simple AD can also be used as a DNS forwarder, to forward queries from outside of AWS (via VPN or Direct Connect) to a Route53 private hosted zone. In order for this to work the fully qualified domain name (fqdn) of the Simple AD should not match the Route 53 private hosted zone fqdn. So, if you want to forward DNS queries to mgmt.helecloud.com the fqdn of the Simple AD should be something like mgmt.helelcoud.local – although AWS have recently announced that Route 53 now supports bi-directional querying between on-premises and AWS over private connections.

The Simple AD also requires a VPC and two subnets to be deployed into – the subnet selection ensures that two different Availability Zones are chosen for fault isolation and resiliency.

Once the directory has been deployed it will have 2 (fixed) IP addresses that should be used in the VPC DHCP Options Set, to enable EC2 instances to use the directory for LDAP authentication and also as a DNS forwarder to Route 53 private hosted zones.

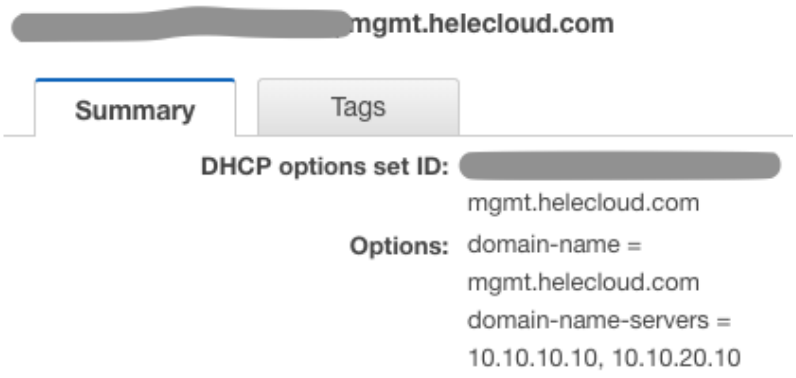


Figure 2 – example DHCP Options Set

To administer the directory, from a Windows instance, connect using the Remote Server Administration Tools (Active Directory Users and Computers); or from a linux instance, connect using the ldb-tools package.

HELECLOUD

To add Windows instances to the domain, use the traditional - properties of My Computer, and under domain and workgroup, enter the details of the domain and Administrator credentials, followed by a reboot.

In Linux a little more setup is required, first the following packages need to be installed

```
sudo apt-get install -y realmd sssd sssd-tools samba-common krb5-user
packagekit samba-common-bin samba-libs adcli
```

A Kerberos session needs to be established with the Simple AD, first the krb5.conf file needs the following additions

```
[libdefaults]
      default_realm = MGMT.HELECLOUD.COM

[realms]
      MGMT.HELECLOUD.COM = {
            kdc = 10.10.10.10
            kdc = 10.10.20.10
            admin_server = 10.10.10.10
      }
```

Figure 3 – example /etc/krb5.conf

kinit is then used to establish a Kerberos session with the Simple AD

```
kinit administrator@MGMT.HELECLOUD.COM
```

and then the instance is joined to the domain

```
sudo realm --verbose join MGMT.HELECLOUD.COM
```

After the instance is joined to the domain, we need to setup the System Security Services Daemon (sssd) so that when a user connects using ssh their session is authenticated by the Simple AD

```
[nss]
filter_users =
root,named,avahi,haldaemon,dbus,radiusd,news,nscd,centos,ubuntu

[pam]

[sudo]

[ssh]

[sssd]
domains = mgmt.helecloud.com
```

HELECLOUD

```
config_file_version = 2
services = nss, pam, ssh, sudo
use_fully_qualified_names = False

[domain/mgmt.helecloud.com]
auth_provider = ad
chpass_provider = ad
enumerate = True
ad_gpo_access_control = permissive
ad_domain = mgmt.helecloud.com
krb5_realm = MGMT.HELECLOUD.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 = /home/%u
access_provider = ad
ldap_user_certificate = noSuchAttribute
ldap_user_ssh_public_key = sshPublicKey
```

Figure 4 – example /etc/sssd/sssd.conf

The sssd and ssh services need to be restarted, and then you can login using a user from the Simple AD.

In order to provide public/private key authentication, a schema extension is required for Simple AD, so the following needs to be imported using the ldbadd command:

```
dn:
CN=sshPublicKey,CN=Schema,CN=Configuration,DC=mgmt,DC=helecloud,DC=com
changetype: add
objectClass: top
objectClass: attributeSchema
attributeID: 1.3.6.1.4.1.1466.115.121.1.40
cn: sshPublicKey
name: sshPublicKey
lDAPDisplayName: sshPublicKey
description: Users public SSH key
attributeSyntax: 2.5.5.5
oMSyntax: 22
```

HELECLOUD

```
isSingleValued: FALSE


dn:
CN=ldapPublicKey,CN=Schema,CN=Configuration,DC=mgmt,DC=helecloud,DC=com
changetype: add
objectClass: top
objectClass: classSchema
governsID: 1.3.6.1.4.1.1466.115.121.1.41
cn: ldapPublicKey
name: ldapPublicKey
lDAPDisplayName: ldapPublicKey
description: Used to store SSH keys in LDAP
subClassOf: top
objectClassCategory: 3
mayContain: sshPublicKey
mayContain: uid
```

Figure 5 – example schema extension ssh.ldif

You will need the ldb-tools and samba packages installed and then run

```
ldbadd -H "ldap://10.10.10.10" ssh.ldif --user "Administrator" --password
"{SimpleAD Administrator password}"
```

Furthermore, the following needs to be added to the sshd_config file of the instances

```
AuthorizedKeysCommand /usr/bin/sss_ssh_authorizedkeys
AuthorizedKeysCommandUser nobody
```

Figure 6 – example /etc/ssh/sshd_config

Now, we can attach the users public ssh key to their user in SimpleAD

```
dn: CN=Adam,OU=mgmt,DC=mgmt,DC=helecloud,DC=com
changeType: modify
add: objectClass
objectClass: ldapPublicKey

dn: CN=Adam,OU=mgmt,DC=mgmt,DC=helecloud,DC=com
changeType: modify
add: sshPublicKey
sshPublicKey: ssh-rsa AAAAB3NzaC1y…BWvBRJHsOE95j Adam
```

Figure 7 – example public-key.ldif

HELECLOUD

Add it to an existing Simple AD user

```
ldbmodify -H “ldap:// 10.10.10.10” public-key.ldif --user
“Administrator” --password “{SimpleAD Administrator password}”
```

and then test that the users public key can be retrieved from the Simple AD using

```
/usr/bin/sss_ssh_authorizedkeys Adam
```

Once authentication is working, you can control which Simple AD users can login to an EC2 instance using Simple AD groups and the `ad_access_filter` in `/etc/sssd/sssd.conf`

```
ad_access_filter =
(memberOf=cn=winbastion,ou=mgmt,dc=mgmt,dc=helecloud,dc=com)
```

Figure 8 – example /etc/ssh/sssd.conf file

Systems Manager

AWS Systems Manager gives you visibility and control of your infrastructure on AWS.

The main benefit of using Systems Manager is that there is no need to:

- setup SSH keys
- deploy and manage a bastion (a hardened instance used to provide access to instances without having to make all instances public)
- open security group ports to allow access

Before you can use Systems Manager, the instances you want to access need to be configured and setup. This consists of 2 steps:

1. Verify user permissions and create an instance profile role
2. Verify the latest SSM agent is installed (it comes pre-installed on some AMI's) on the instances

The user permissions are to ensure the user can access Systems Manager and the profile role permissions can access Systems Manager. The instances require the **AmazonEC2RoleforSSM** permissions added to their profile role. Once this has been added you should see the instances appear under Managed instances in AWS Systems Manager.

Run Command

The run command can be used to run a pre-defined script (known as a document within Systems Manager) or to run a custom/one off command

To run a set of commands against an instance, from within Systems Manager, select Run command and then AWS-RunShellScript. Under command parameters specify the command(s) to be run. Then specify the instances that it should be run against using either tags or manually select the instances. Then click

HELECLOUD

run, that's it! Once run the output will be available to view – if the output exceeds 2500 chars then you need to specify an S3 bucket for the command to output to.

You can also have the command output stored in CloudWatch and configure Systems Manager to send notifications via Amazon Simple Notification Service.

Session Manager

Another tool within AWS Systems Manager is Session Manager which can also be used to establish secure interactive command line access to instances.

To connect, select Session Manager, click Start session and select the instance you wish to connect to, then start session, a new window opens with your active session.

The session output can be captured to S3 and/or CloudWatch by configuring the preferences before connecting.

For both the Run Command and Session Manager, to control which instances a user can connect to, an appropriate IAM policy will need to be applied to the user, this could state that the instance name needs to meet certain conditions or that it must have certain tags.

In addition to these capabilities, Systems Manager also provides Automation (define tasks as documents to be automated), Patching (to patch Windows and Linux instances), Distributor (software packager & installer) as well as Maintenance windows so that these tasks can be run at a certain time.

Conclusion

As is often the case with AWS there are a number of different ways to achieve centralised secure access to instances and the correct choice will be down to your individual requirements.

At HeleCloud we can help you make informed decisions about the best way to tackle these challenges.