

**RED HAT**  
**SUMMIT**

Boston, MA  
June 23-26, 2015

Red Hat Summit 2015

# Red Hat Enterprise Linux Identity Management

Diaa Radwan

## **Table of Contents**

Lab Overview .....	3
Background .....	3
Red Hat Enterprise Linux Identity Management Overview .....	3
Red Hat Enterprise Linux Identity Management Benefits: .....	4
Enhances Security .....	4
Provides eSSO (enterprise Single Sign-on) .....	4
Centralizes Administration and Control .....	4
Implements Standards-Based, Integrated Components .....	4
Reduces costs .....	4
IdM Features .....	4
IdM Lab Environment Details .....	5
IdM Lab objectives .....	5
Lab 1: Server Installation .....	6
Lab 2: Users and Password Policies .....	9
Lab 3: Two Factor Authentication .....	11
Lab 4: Client Installation .....	14
Lab 5: User Groups and Host Groups Management .....	16
Lab 6: Integrating IdM with Active Directory .....	20
Lab 7: Host Based Access Control – HBAC .....	28
Lab 8: IdM Roles Management .....	32
Lab 9: IdM Replication .....	37
Lab 10: Services and Keytabs .....	38

## Lab Overview

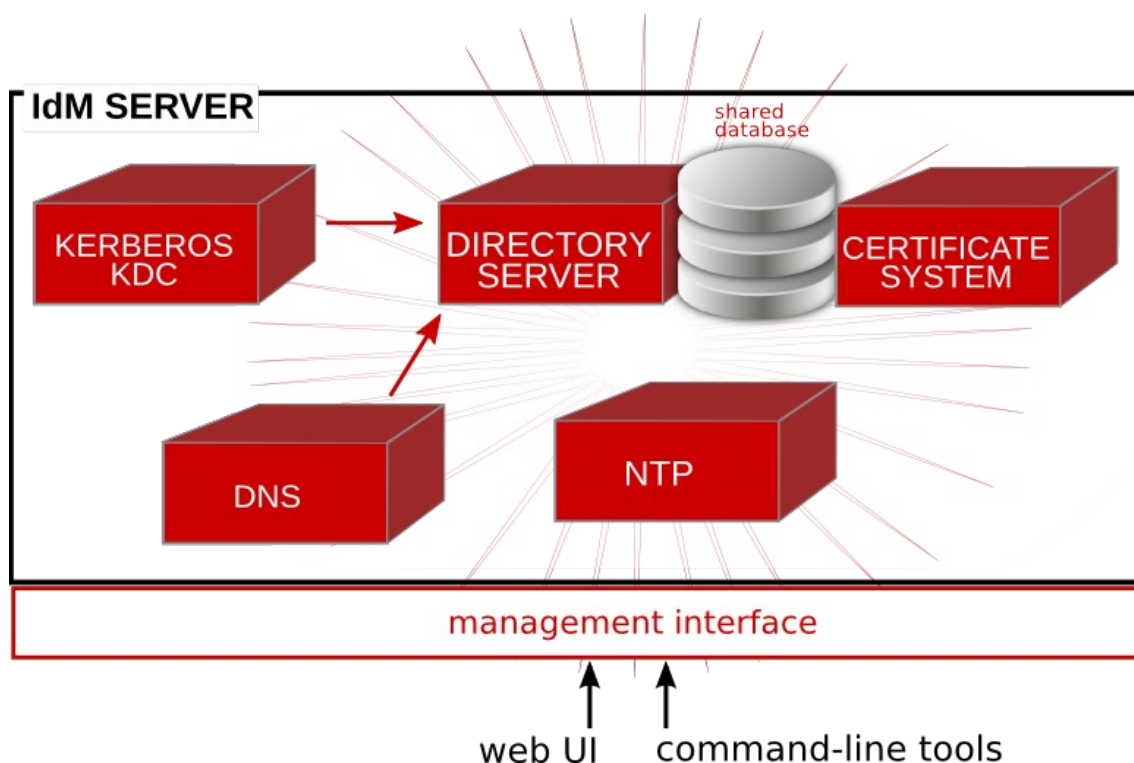
This lab guide assumes that you're following instructor-led training and that this lab guide is will try to simulate real life tasks and scenarios. It goes through a number of labs that will enable your to create full functional environment using Red Hat Enterprise Linux IdM. Also you will explore IdM features such as users, groups, policies and access control rules management. The purpose is to give you a basic hands-on overview of Red Hat Enterprise Linux Identity Management and how the components are fit together. It will use a combination of command-line tools and the IdM web interface. This lab is prepared to run on environment, the setup is descried in this document on Lab Environment Section.

Your instructor will provide you with any additional information that you will require, primarily the lab setup and required scenarios.

## Background

### Red Hat Enterprise Linux Identity Management Overview

Red Hat Enterprise Linux IdM is a way to create identity stores, centralized authentication, domain control for Kerberos and DNS services, and authorization policies — all on Linux systems, using native Linux tools. It is also supports Linux/Unix domains.



## Red Hat Enterprise Linux Identity Management Benefits:

### Enhances Security

Centralizes authentication, authorization and fine-grained access control for UNIX/Linux environments.

### Provides eSSO (enterprise Single Sign-on)

Enables users to access many different enterprise resources after their initial log-in without having to type user name and password again and again.

### Centralizes Administration and Control

Allows administrators to easily consolidate and manage identity servers in a UNIX/Linux environment; with the option to interoperate with Active Directory.

### Implements Standards-Based, Integrated Components

Integrates the capabilities of Kerberos, LDAP, DNS and x.509 certificates into a simple identity management solution.

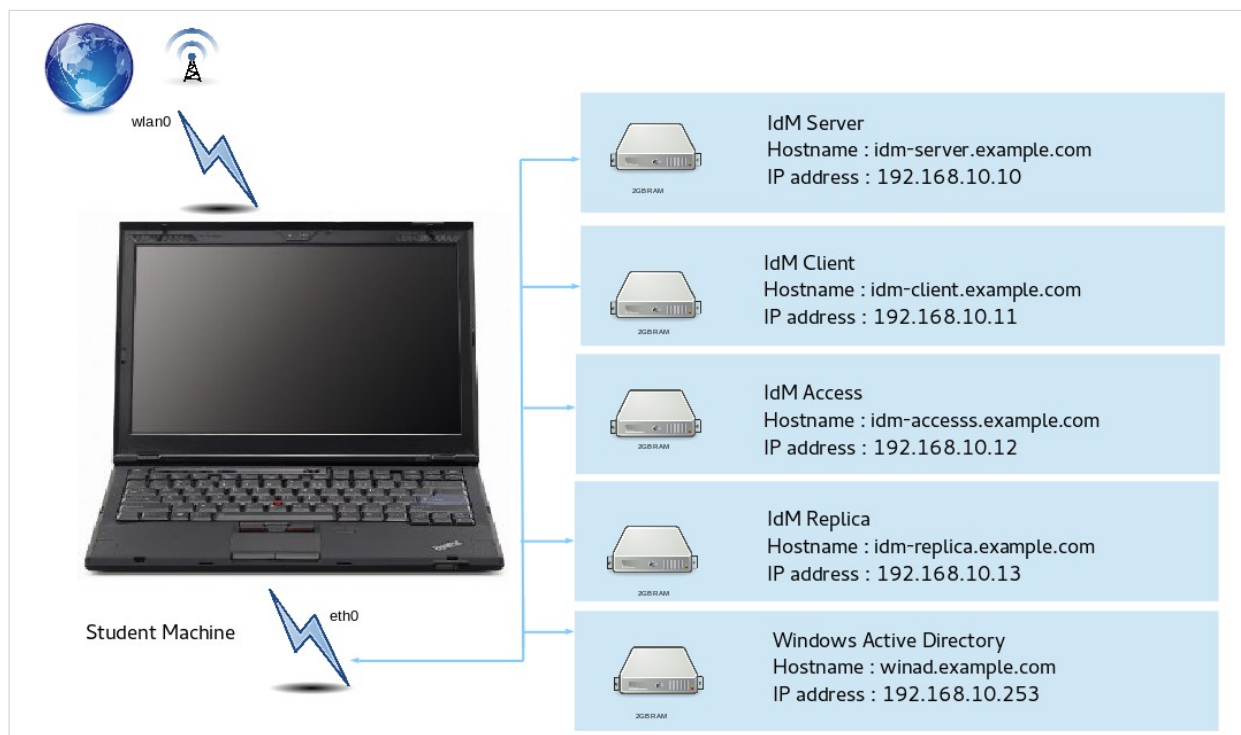
### Reduces costs

Can replace third-party user directories or Identity Management Solutions

## IdM Features

- Integrated, native user, host, and service authentication and access control.
- Consistent and manageable identity management for Linux and Unix systems.
- Interoperability with Microsoft Active Directory domains.
- Standards-based, trusted technologies.
- Easier and clearer to implement, maintain, and understand authentication and access control policies.
- Flexible access control rules based on sudo rules, host-based rules, and other criteria.
- Consistent and universal password policies for users.
- Integrate established Linux/Unix services like NFS, automount, NIS, NTP, Kerberos, and DNS into a single management domain.
- Smooth migration paths from NIS and LDAP services.
- Scalable operations with up to 20 servers and replicas and an unlimited number of clients in a single domain.

## IdM Lab Environment Details



Element	URL	Username	Password
IdM Server	http://idm-server.example.com	admin	password
IdM Server	ssh: idm-server.example.com	root	redhat
IdM client	ssh: idm-client.example.com	root	redhat
IdM access evaluation	ssh: idm-access.example.com	root	redhat
IdM Replication	ssh idm-replica.example.com	root	redhat
Windows Active Directory	Virtual Machine Console	administrator	Secret123

## IdM Lab objectives

Deploy both client and server centralized and high available authentication using Red Hat Enterprise Linux Identity Management (IdM) and provide a working central authentication server, implement additional access controls and sudo rules for client and access machines.

**Note:** Make sure that all virtual machines starting with “**IdM-\***” are running. After finishing Lab1, you can start the Windows-DC machine which is running the Active Directory

## Lab 1: Server Installation

**Target server:** idm-server.example.com

**Access:** ssh **root@idm-server.example.com**

- Log into idm-server.example.com, via ssh.
- Make sure that hosts file is properly configured, you should find this line:

```
cat /etc/hosts | grep idm
192.168.10.10    idm-server.example.com    idm-server
```

- Install the IdM packages:

```
yum -y install bind-dyndb-ldap ipa-server
```

- Run as root:

```
[root@idm-server ~]# ipa-server-install --setup-dns --ssh-trust-dns \
--mkhomedir
```

- When you prompt for these questions use the respective answers:

```
Existing BIND configuration detected, overwrite? [no]: <Yes>
Server host name [idm-server.example.com]: <Press Enter>
Please confirm the domain name [example.com]: <Press Enter>
Please provide a realm name [EXAMPLE.COM]: <Press Enter>
Directory Manager password: <Use "password">
Password (confirm): <Use Password>
IPA admin password: <Use Password>
Password (confirm): <Use Password>
Do you want to configure DNS forwarders? [yes]: Yes
Enter IP address for a DNS forwarder: 192.168.10.254
Enter IP address for a DNS forwarder: <Press Enter>
Do you want to configure the reverse zone? [yes]: Yes
Continue to configure the system with these values? [no]: Yes
Domain name:    example.com
```

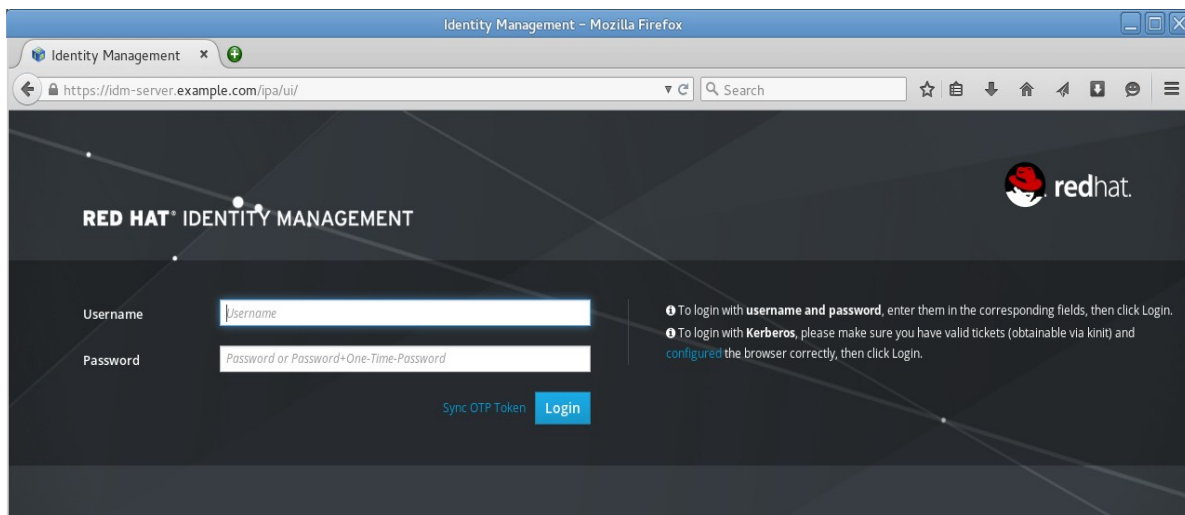
- You should get the same information at end of the dialog:

```
The IPA Master Server will be configured with:
Hostname:    idm-server.example.com
IP address:  192.168.10.10
```

## 7 | Red Hat Summit 2015 – Red Hat Enterprise Linux Identity Management

```
Domain name:   example.com
Realm name:    EXAMPLE.COM
BIND DNS server will be configured to serve IPA domain with:
Forwarders:    8.8.8.8
Reverse zone:  10.168.192.in-addr.arpa.
```

**After installation:** Check the IdM web interface via `idm-server.example.com`, use the admin username and password.



- Check main IPA configuration: `/etc/ipa/default.conf` base DN, realm.
- Obtain a kerberos ticket:

```
kinit admin
klist
```

- Check automatically created DNS records (A, SRV):

```
ipa dnszone-find
ipa dnsrecord-find --name=idm-server --all
Zone name: example.com
Active zone: TRUE
Authoritative nameserver: idm-server.example.com.
Administrator e-mail address: hostmaster.example.com.
SOA serial: 1434449021
SOA refresh: 3600
SOA retry: 900
```

- Check IdM server defaults:

```
ipa config-show  
ipa config-mod --defaultshell=/bin/bash
```

- Then on the idm-server check the logs (Just to know where to start debugging, not needed):

```
/var/log/pki-ca/debug  
/var/log/pki-ca-install.log  
/var/log/dirsrv/ (permissions!)  
/var/log/messages
```

- Common install issues:
  - Broken DNS, bad /etc/hosts configuration.
  - Files and certificates remains after the last unsuccessful install.
  - Time synchronization issues.



## Lab 2: Users and Password Policies

**Target server:** idm-server.example.com

**Access:** ssh [root@idm-server.example.com](mailto:root@idm-server.example.com)

1. Add new users (create a username with your preferences in the prompt mode), then run the other commands:

```
ipa user-add
ipa user-add --first=John --last=Smith jsmith
ipa user-add --first=Matt --last=Well --manager=jsmith \
--email=mwell@example.com --homedir=/home/mwell mwell
```

2. Modify User attributes:

```
ipa user-mod jsmith --addattr=departmentnumber=101
ipa user-show jsmith --all
ipa user-mod mwell --title="System Engineer"
```

3. Modify Users password as admin:

```
ipa user-mod mwell --password
ipa user-mod jsmith --password
```

4. Check if the system recognize the users:

```
id jsmith
getent group mwell
```

5. Check the default Password Policies:

```
ipa help pwpolicy
ipa pwpolicy-show
ipa pwpolicy-mod --maxlife=60
```

6. As jsmith **login** via ssh to idm-server, you will be prompted to change the password for first time. Then Change password with:

```
[root@idm-server ~]# ssh jsmith@localhost
ipa passwd
```

It will fail because of min life policy.

7. As Admin:

```
ipa pwpolicy-mod --minlife=0 --maxfail=3  
ipa pwpolicy-show
```

8. As mwell, login to the idm-server, change the 1<sup>st</sup> time password and then, change password with “ipa passwd ”, it will succeed as we changed the minimum lifetime of users password.

9. On the Web UI check the following:

- Add a user.
- Check password expiry.
- Edit user details.

**Reference:**

[Red Hat Documentation : Managing User Groups](#)

## Lab 3: Two Factor Authentication

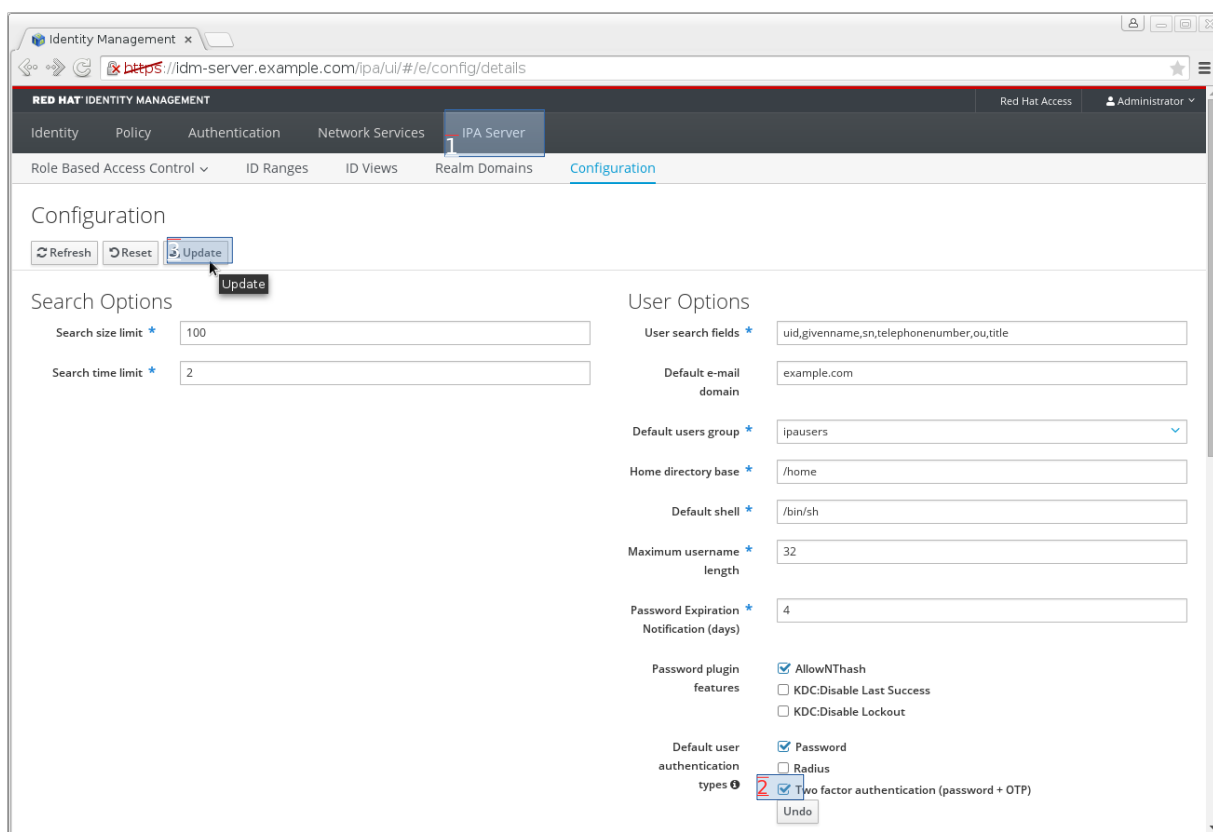
Two-factor authentication is a security process in which the user provides two means of identification, the user will be asked to provide the authentication system two elements or parts, first part is something the user know and the second one is something the user have. If you didn't finish this lab it will not impact the rest of the workshop.

**Target server:** idm-server.example.com

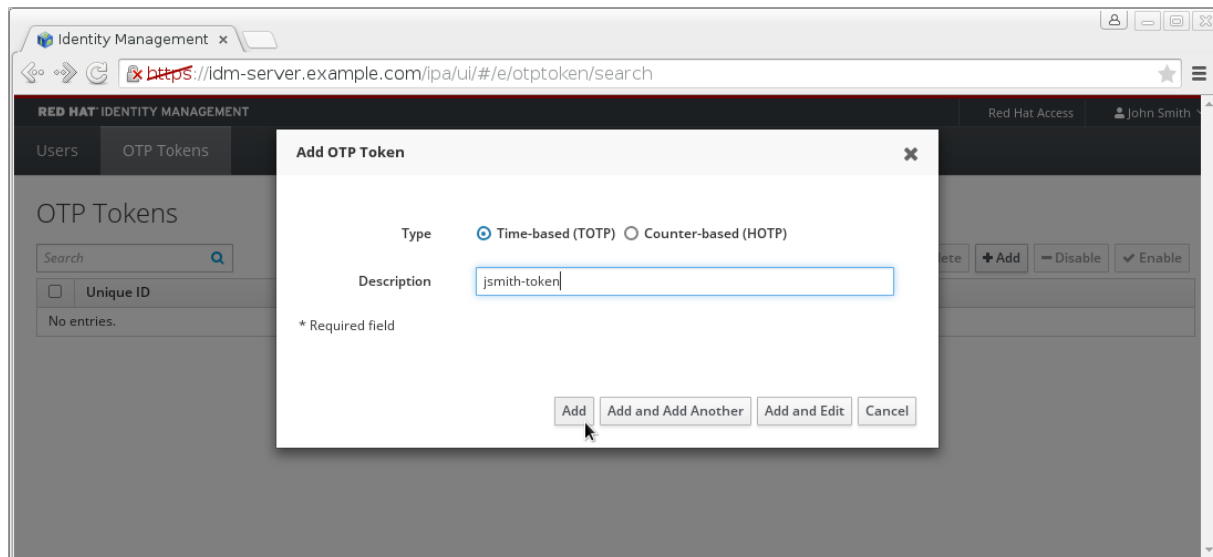
**2FA soft token:** Install FreeOTP on your smartphone, you can find the application on [App Store](#) or [Google Play](#)

Open <http://idm-server.example.com> then Login as admin, The navigate to **IPA Server** tab. Then access the “**Configuration**” subtab.

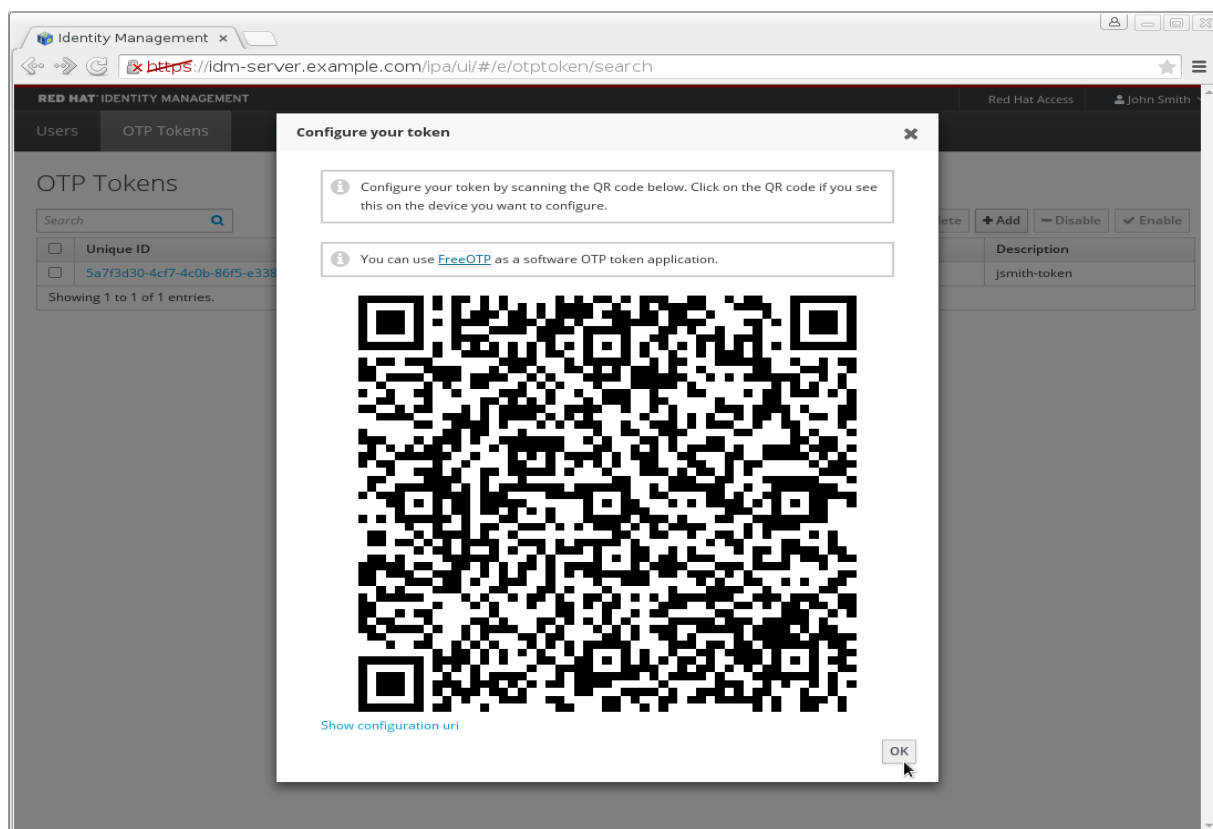
You will find the authentication “**Default user authentication types**” Choose make the “**Two factor authentication (password + OTP)**” then click update:

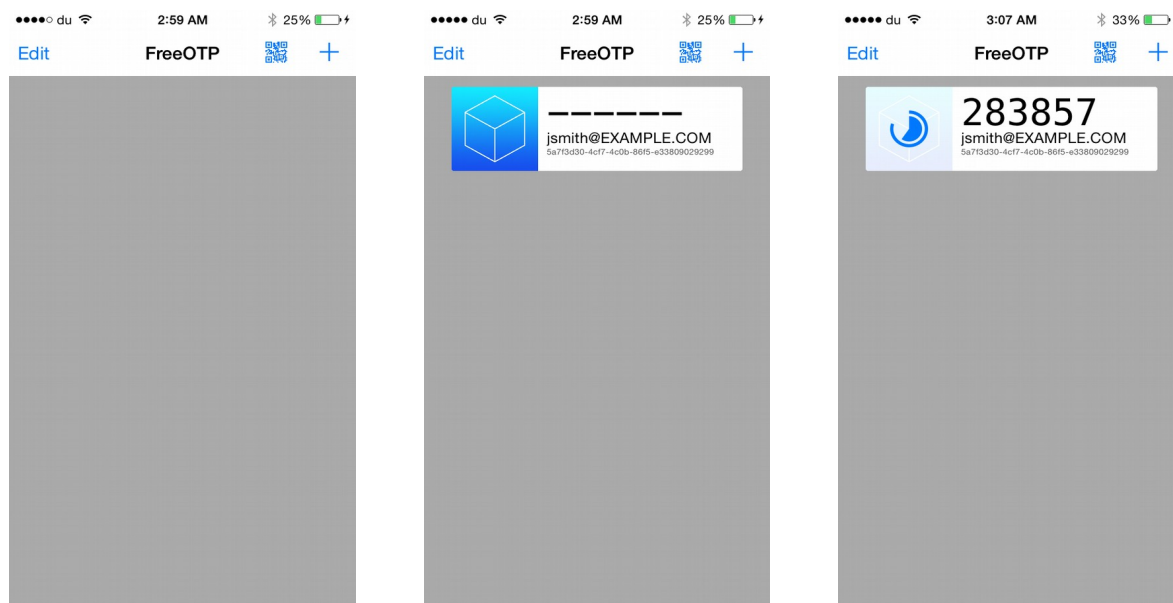


Log out from the admin session, login with jsmith, then navigate to **OTP Tokens** then click on Add. In the Add OTP token, make sure that you fill the required filled as the figure below:



After adding the token, the a QR code will show, scan this QR through FreeOTP or any other Soft token (FreeOTP is recommended):





On the smartphone, open FreeOTP, click on the QR Code Sign, then scan the QR code

After scanning the QR Code you see new token for jsmith:

Click on the new created id to generate a new token.

Now logout user jsmith and try to login using the token. The password will be the original jsmith password+ the generated number. So if the password was "redhat" the login password should be "redhat283857".

Please note that "redhat" as password will still be able to authenticate jsmith because we selected 2 authentication methods in the global IdM configuration.

## Lab 4: Client Installation

**Target server:** idm-client.example.com and idm-access.example.com

**Access:** ssh [root@idm-client.example.com](mailto:root@idm-client.example.com)

- Check in both servers resolv.conf point to IdM server (192.168.10.10):

```
echo 'nameserver 192.168.10.10' > /etc/resolv.conf
cat /etc/resolv.conf
nameserver 192.168.10.10
```

- Verify that idm-client/idm-access resolvers are pointing to idm-server

```
dig example.com
example.com. 3600 IN SOA idm-server.example.com. hostmaster.example.com. 1396857706
3600 900 1209600 3600
```

- Install the IdM client (sss):

```
yum install ipa-client
```

- on IdM server, make sure that PRT records are created/updated in new client installations:

```
ipa dnszone-mod --allow-sync-ptr=TRUE
Zone name: example.com
```

- Client installation:

```
ipa-client-install --enable-dns-updates --mkhomedir --ssh-trust-dns
User authorized to enroll computers: <admin>
Password for admin@EXAMPLE.COM: <password>
```

- Additional options to automate the installation:

```
ipa-client-install --mkhomedir --enable-dns-updates --ssh-trust-dns \
--server=idm-server.example.com --domain=example.com -p admin -w password \
--fixed-primary -U
```

- Some adjustment.

The default shell for new users is `/bin/sh`, which should probably be changed if you are using Linux only, On `idm-server`:

```
[root@idm-server ~]# ipa config-mod --defaultshell=/bin/bash
```

- Perform all the above steps on `idm-access.example.com`, to install the client on `idm-access`.
- Try to access both machines with the above created users from `idm-client.example.com`.

```
ssh jsmith@idm-access.example.com
Creating home directory for jsmith.
```

- Ssh back to `idm-client.example.com`, you should login without any passwords:

```
ssh jsmith@idm-client.example.com
```

**Note:** make sure that you have domainname and hostname in your hosts file. Example: in `idm-access.example.com`:

```
192.168.10.12 idm-access.example.com idm-access
```

## Lab 5: User Groups and Host Groups Management

**Target server:** idm-server.example.com

**Access:** ssh [root@idm-server.example.com](mailto:root@idm-server.example.com)

Activities for lab 4:

- Create users group (Either through command line or Web UI).
- Adding Group Members.
- Deleting users group.
- Explore IdM group management through command line, a new group named 'servers' will be added, then user 'mwell' will be member of servers, adding other group named 'clients' and finally adding jsmith to 'clients' group:

```
ipa group-add --desc='users server group' servers
ipa group-add-member servers --users=mwell
ipa group-add --desc='users client group' clients
ipa group-add-member clients --users=jsmith
ipa group-find
ipa group-del <group name>
ipa help group
```

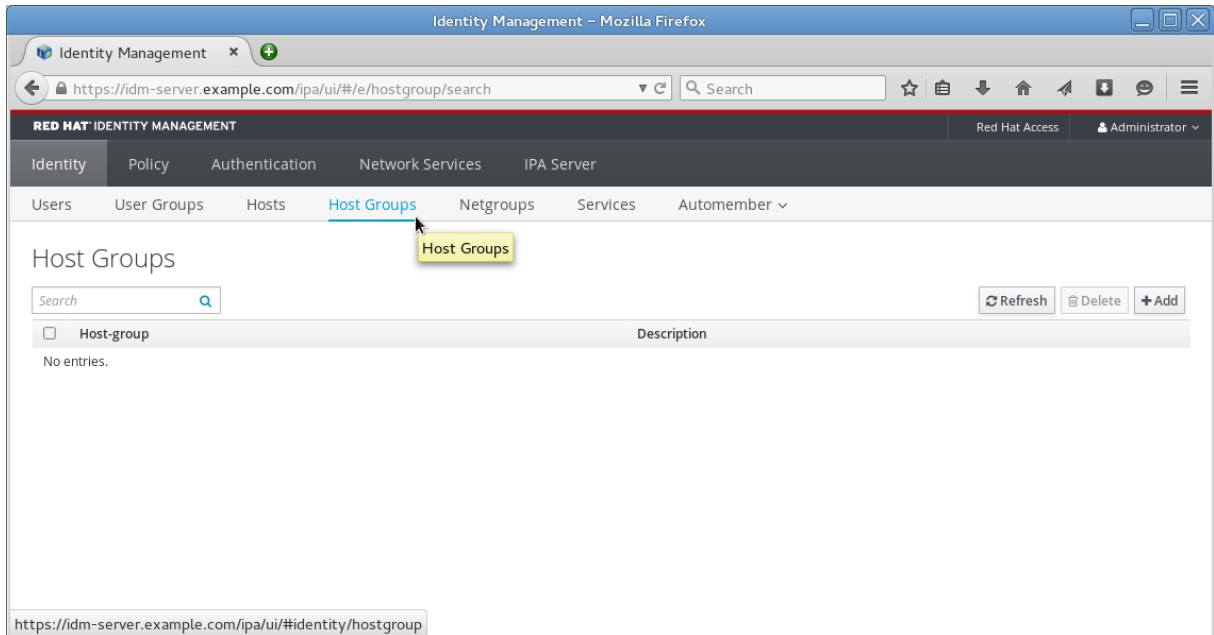
On the Web UI check the following:

- The group that you just created through the command line.
- Default User and Groups Settings, 3 default groups:
  - ipausers.
  - admins.
  - editors.
- Check the created groups on the web interface, check also the created users.
- Create two host groups:
  - restricted.
  - access.

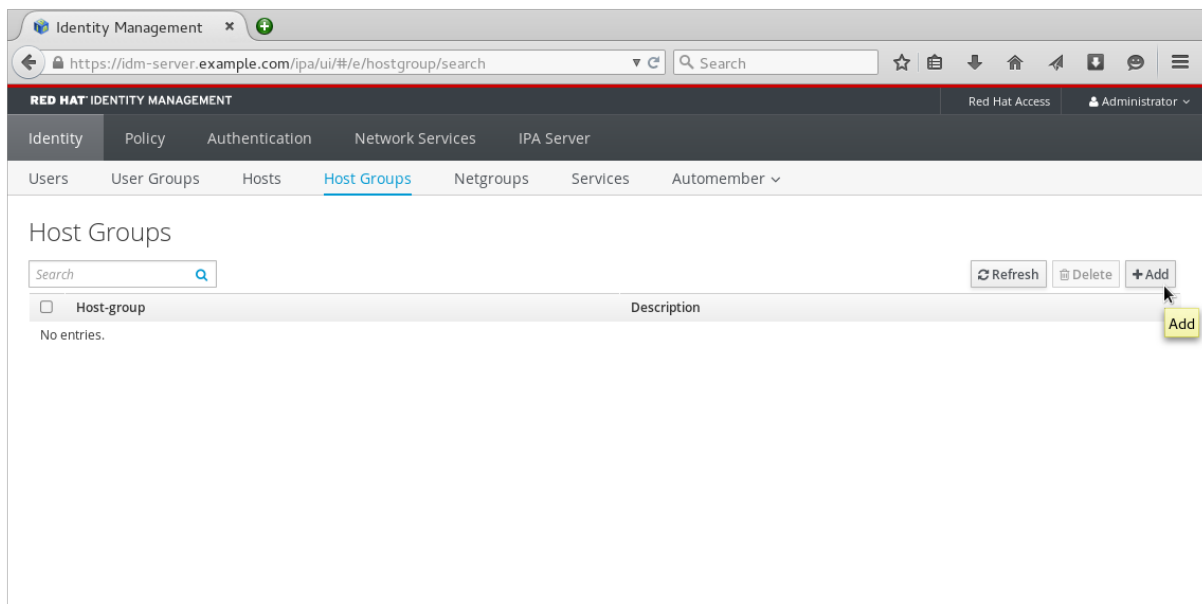


Through the web interface follow Identity ► Host groups ► Add.

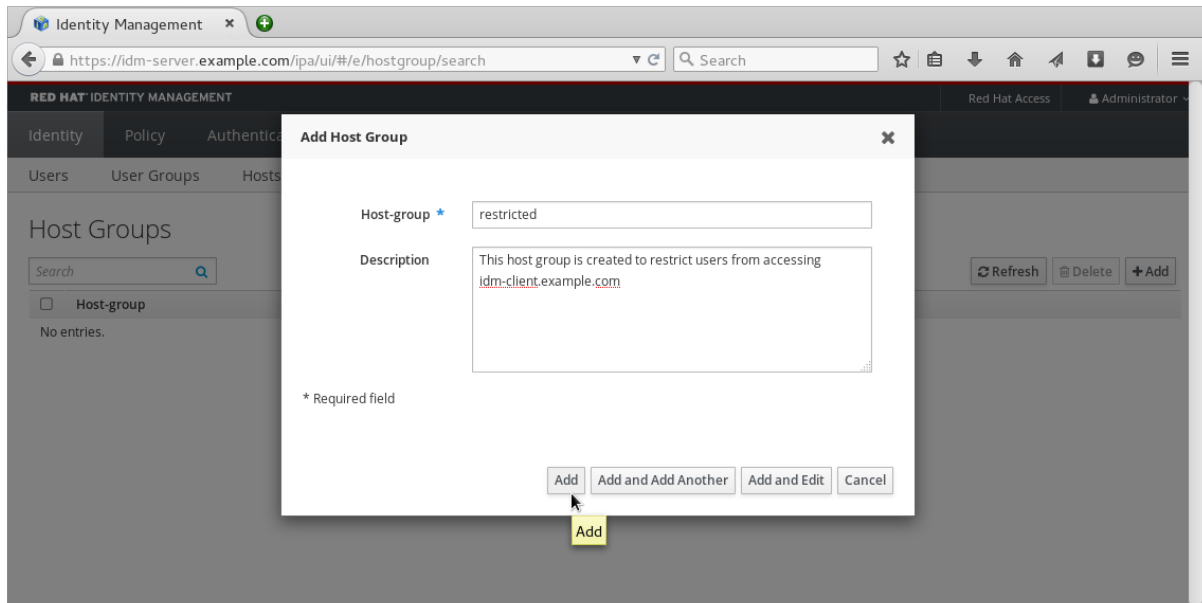
Visit the Host Groups sub tab:



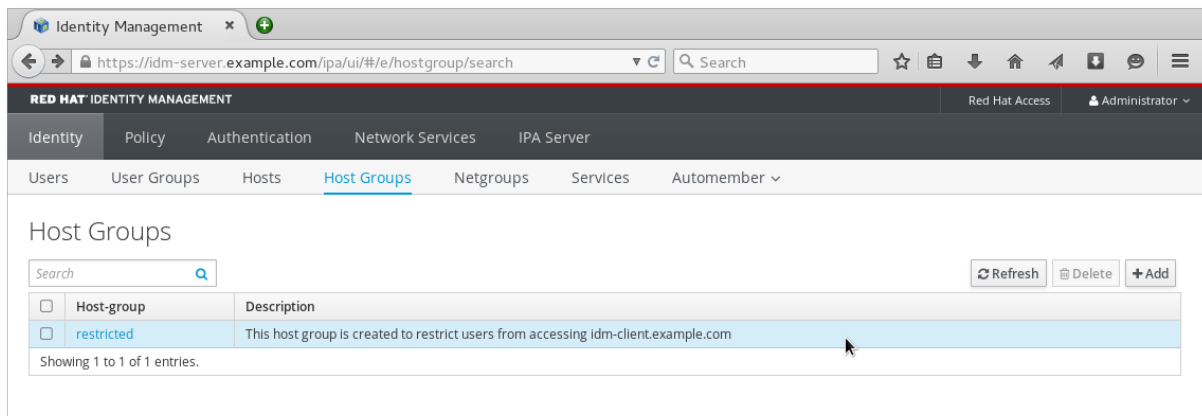
Then find “**Add**” button to add new Host Group:



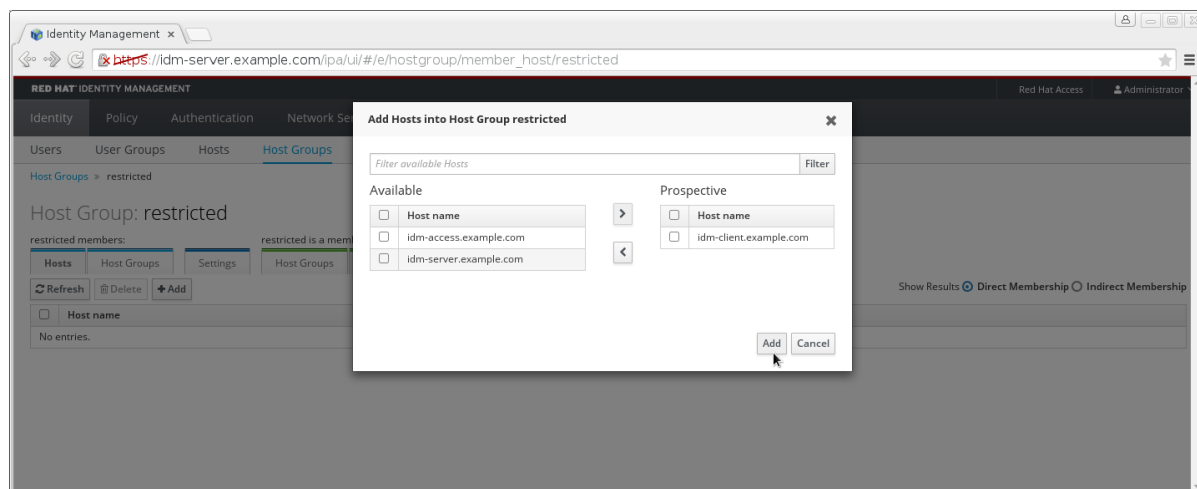
As required create new group called “**restricted**” and “**access**”:



Now the host group is created, click on “**restricted**” to add the hosts



Then you will find a created host group named “**restricted**”, click on the “**restricted**” then “**Add**”.



- Both **idm-client** and **idm-server** should be in restricted group.
- Create other host group following the same steps, name this group “**access**”.
- The **idm-access** machines should be in “**access**” group, follow the same steps in creating and adding machines to restricted group.

**Reference:** [Red Hat Documentation : Managing User Groups](#)

## Lab 6: Integrating IdM with Active Directory

**Target server:** idm-server.example.com, idm-client.example.com and winad.example.com

**Access:** ssh [root@idm-server.example.com](mailto:root@idm-server.example.com), ssh [root@idm-client.example.com](mailto:root@idm-client.example.com) and console access to winad.example.com

One of the available machines is running Windows Active Directory, the machine is ready with AD. Make sure that you have access to the Windows machine using username “administrator” and the password is “Secret123”. Also we will install the AD trust and winbind clients.

The IdM integration with Active Directory is sensitive to DNS setup, we want to allow example.com that is hosted on idm-server.example.com to be transferable to the AD

### Run on IdM server:

```
[root@idm-server ~]# yum install -y ipa-server-trust-ad samba-winbind-clients
[root@idm-server ~]# ipa dnszone-mod example.com --allow-transfer=192.168.10.253
Zone name: example.com.
Active zone: TRUE
Authoritative nameserver: idm-server.example.com.
Administrator e-mail address: hostmaster.example.com.
SOA serial: 1433680513
SOA refresh: 3600
SOA retry: 900
SOA expire: 1209600
SOA minimum: 3600
Allow query: any;
Allow transfer: 192.168.10.253;
```

### On winad.example.com:

Open the powershell/cmd and run the following to add the example.com zone:

```
dnscmd 127.0.0.1 /ZoneAdd example.com /Secondary 192.168.10.10
```

Just in case you didn't know how to open PowerShell, here is the icon:



Running the dnscmd command should return the same output:

```
Windows PowerShell
Copyright (C) 2009 Microsoft Corporation. All rights reserved.

PS C:\Users\Administrator> dnscmd 127.0.0.1 /ZoneAdd example.com /Secondary 192.168.10.10
DNS Server 127.0.0.1 created zone example.com:

Command completed successfully.

PS C:\Users\Administrator> _
```

On the Windows Desktop, You will find DNS icon (shortcut), it will open DNS service on windows, we want to verify the new resources created, double click on DNS icon and follow the DNS tree as shown blow:

Name	Type	Data	Timestamp
_kerberos	Service Location (SRV)	[0][100][88] idm-server.exa...	static
_kerberos-master	Service Location (SRV)	[0][100][88] idm-server.exa...	static
_kpasswd	Service Location (SRV)	[0][100][464] idm-server.ex...	static
_ldap	Service Location (SRV)	[0][100][389] idm-server.ex...	static

The red lines shows that idm-server.example.com is added in the forward lookup zones.

Next configuration will be applying DNS forwarding for queries related to winad.example.com to AD DNS server. On the idm-server.example.com run:

```
[root@idm-server ~]# ipa dnsforwardzone-add winad.example.com \
--forwarder=192.168.10.253 --forward-policy=only
Zone name: winad.example.com.
Active zone: TRUE
Zone forwarders: 192.168.10.253
Forward policy: only
```

Then adding Windows AD A record in the IdM server:

```
[root@idm-server ~]# ipa dnsrecord-add example.com dc.winad \
--a-ip-address=192.168.10.253
Record name: dc.winad
A record: 192.168.10.253
```

Also we need to add the NS record for winad (AD domain name)

```
[root@idm-server ~]# ipa dnsrecord-add example.com winad --ns-hostname=dc.winad
Record name: winad
NS record: dc.winad
```

Verify that SRV records are resolvable on IdM server:

```
[root@idm-server ~]# dig SRV _ldap._tcp.winad.example.com
; <<>> DiG 9.9.4-RedHat-9.9.4-18.el7_1.1 <<>> SRV _ldap._tcp.winad.example.com
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 25329
;; flags: qr rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 1, ADDITIONAL: 2

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 4096
;; QUESTION SECTION:
;_ldap._tcp.winad.example.com.  IN      SRV

;; ANSWER SECTION:
_ldap._tcp.winad.example.com. 468 IN      SRV      0 100 389 dc.winad.example.com.

;; AUTHORITY SECTION:
winad.example.com.          86400  IN      NS       dc.winad.example.com.

;; ADDITIONAL SECTION:
dc.winad.example.com.      3468  IN      A       192.168.10.253
```

After verifying that IdM server can resolve SRV records of AD, we will verify that SRV records of IdM server are resolvable from AD server (you will type the yellow underlined nslookup commands):

```
PS C:\Users\Administrator> nslookup.exe
Default Server: localhost6.localdomain6
Address: ::1

> set type=srv
> _ldap._tcp.example.com
Server: localhost6.localdomain6
Address: ::1

_ldap._tcp.example.com SRV service location:
  priority      = 0
  weight        = 100
  port          = 389
  svr hostname  = idm-server.example.com
idm-server.example.com internet address = 192.168.10.10
>
```

We can verify that the record it add to the ldap using ldapsearch:

```
ldapsearch -Y GSSAPI -b cn=dns,dc=example,dc=com idnsname=example.com.
```

The next configuration required will be installing the AD trust IdM server, it will add all necessary objects and configuration to allow IdM server to create a trust to the Active Directory domain.

```
ipa-adtrust-install -U --netbios-name="EXAMPLE" --enable-compat -a "password"
```

Before adding the trust relationship, we need to make sure that both server are in the same timezone, using PowerShell on Windows first command will show the configured timezone and the second command will set it to EST, windows will need to be restarted, reboot the windows machine after running the below commands:

```
PS C:\Users\Administrator> tzutil.exe /g
Eastern Standard TimePS C:\Users\Administrator>
PS C:\Users\Administrator> tzutil.exe /s "Eastern Standard Time"
PS C:\Users\Administrator>
```

On IdM server, run `timedatectl` to set the timezone to EST:

```
[root@idm-server ~]# timedatectl set-timezone America/New_York
```

Then, we can start adding the trust relationship (If IdM and AD are having different timezone it will fail):

```
ipa trust-add --type=ad winad.example.com --admin Administrator --password
-----
Re-established trust to domain "winad.example.com"
-----
  Realm name: winad.example.com
  Domain NetBIOS name: WINAD
  Domain Security Identifier: S-1-5-21-3652195975-17874612-2275940394
  SID blacklist incoming: S-1-5-20, S-1-5-3, S-1-5-2, S-1-5-1, S-1-5-7, S-1-5-6, S-1-5-5,
S-1-5-4, S-1-5-9, S-1-5-17, S-1-5-16, S-1-5-15, S-1-5-14,
  SID blacklist outgoing: S-1-5-20, S-1-5-3, S-1-5-2, S-1-5-1, S-1-5-7, S-1-5-6, S-1-5-5,
S-1-5-4, S-1-5-9, S-1-5-8, S-1-5-17, S-1-5-16, S-1-5-15, S-1-5-14
  Trust direction: Two-way trust
  Trust type: Active Directory domain
  Trust status: Established and verified
```

Add AD Admins group to IdM, Create IdM group that will be flagged as external group:

```
[root@idm-server ~]# ipa group-add --desc='ad_domain admins external map' \
ad_admins_external --external
-----
Added group "ad_admins_external"
-----
```

```
Group name: ad_admins_external
Description: ad_domain admins external map
```

Create a POSIX compliant group to be linked to the external group:

```
[root@idm-server ~]# ipa group-add --desc='ad_domain admins' ad_admins
-----
Added group "ad_admins"
-----
Group name: ad_admins
Description: ad_domain admins
GID: 1861200012
```

Add members of Domain Admins to the created IdM group:

```
[root@idm-server ~]# ipa group-add-member ad_admins_external --external \
'WINAD\Domain Admins'
[member user]: <Press Enter>
[member group]: <Press Enter>
Group name: ad_admins_external
Description: ad_domain admins external map
External member: S-1-5-21-1850929294-2263411558-1060893033-512
-----
Number of members added 1
-----
```

Adding members from external AD group to IdM POSIX compliant group:

```
[root@idm-server ~]# ipa group-add-member ad_admins --group ad_admins_external
Group name: ad_admins
Description: ad_domain admins
GID: 1861200016
Member groups: ad_admins_external
-----
Number of members added 1
```

The last commands that we need run are to add the AD users to IdM:

```
ipa group-add --desc='ad_domain users external map' ad_users_external --external
ipa group-add --desc='ad_domain users' ad_users
ipa group-add-member ad_users_external --external 'WINAD\Domain Users'
ipa group-add-member ad_users --group ad_users_external
```



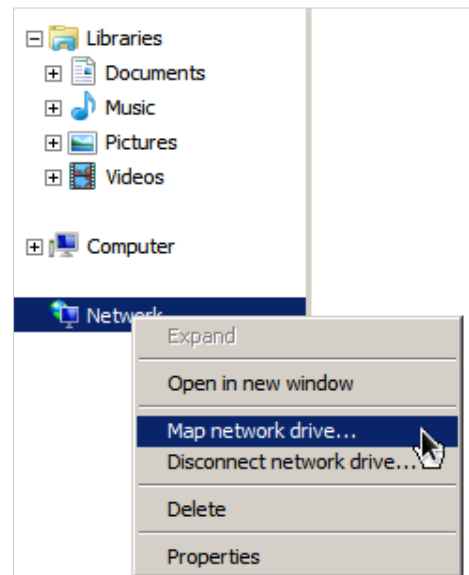
Now, this is the testing time; wbinfo will retrieve the SID associated with the username specified:

```
kinit admin
kvno -S HTTP `hostname`
ipa trust-show winad.example.com
kdestroy
klist
kinit Administrator@WINAD.EXAMPLE.COM
klist
kvno -S cifs dc.winad.example.com
wbinfo -n 'WINAD\Domain Admins'
S-1-5-21-66505577-848503339-3105483033-512 SID_DOM_GROUP (2)
```

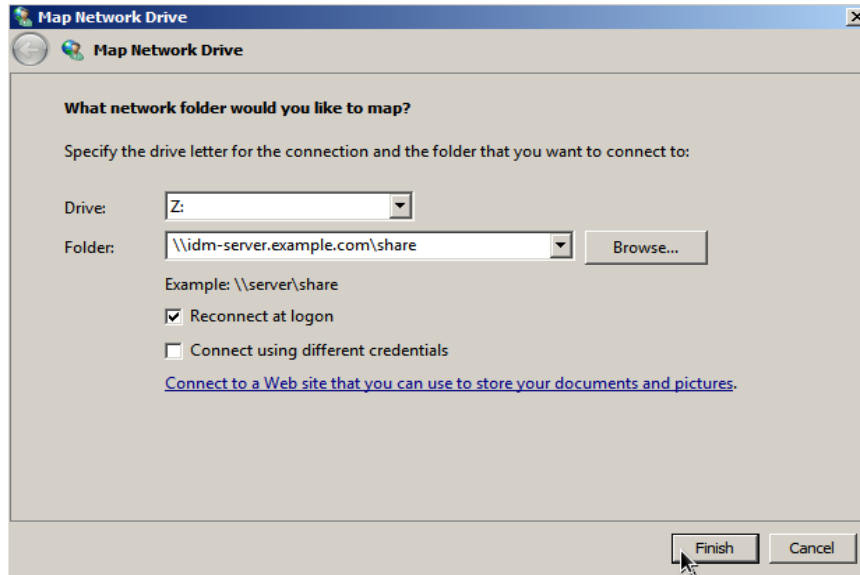
We can create a shared disk to AD Admins, these commands will create and new directory “/linuxshare” and make it available to AD admins:

```
mkdir /linuxshare
SID=`wbinfo -n 'WINAD\Domain Admins'|awk '{print$1}'`
net conf setparm 'share' 'comment' 'Trust test share'
net conf setparm 'share' 'read only' 'no'
net conf setparm 'share' 'valid users' "$SID"
net conf setparm 'share' 'path' '/linuxshare'
cd /linuxshare
touch IdM-rocks
```

The share will be available to Windows Admins, later we can avail users shares if needed, on windows machines open Computer then map the share to a Windows drive following the same procedures:



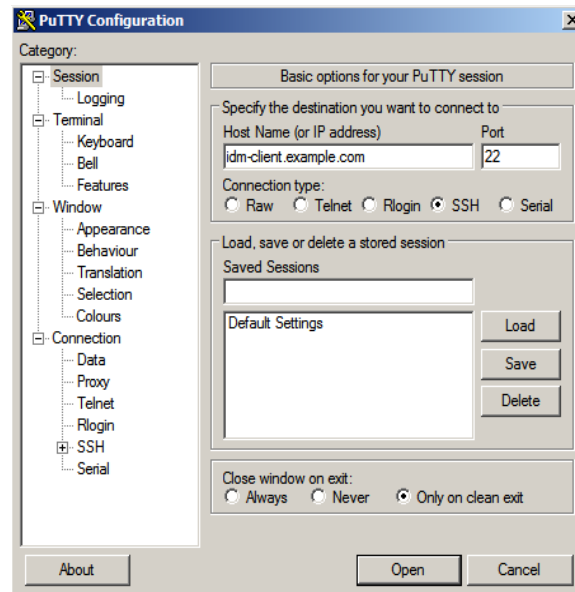
A new dialog will open as dialog will be opened to define the share it will ask for the user password, use the administrator as user and the password is “**Secret123**”:



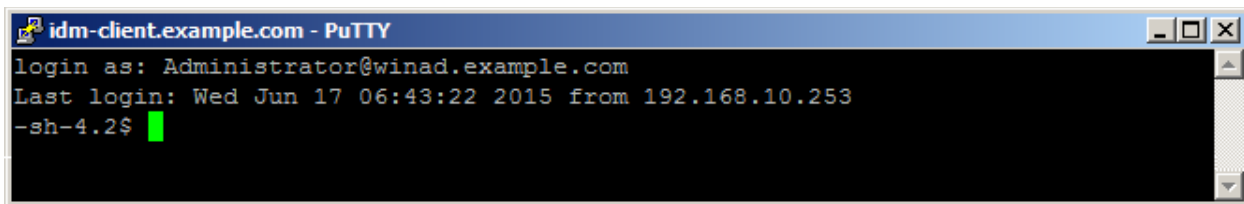
You will find the contents of “**/linuxshare**” available, the file that we created “**IdM-rocks**” will be there accessible. You can create folders on Windows and check them back on the idm-server.example.com.

Now, the administrator user can login to Linux machines without passwords, remember that we didn't configure the Host Based Access Control, so all users can login to all servers it is not recommend to run this configuration in the production. Next lab we will have a HBAC configured and it will show how to define new rules and examining the existing rules.

On the Windows Desktop you will find putty (a ssh client) use idm-client.example.com as the Host Name:



Then use `Administrator@winad.example.com` as the login name, you should log in without password request



**Reference:** [Red Hat Documentation : Windows Integration Guide](#)

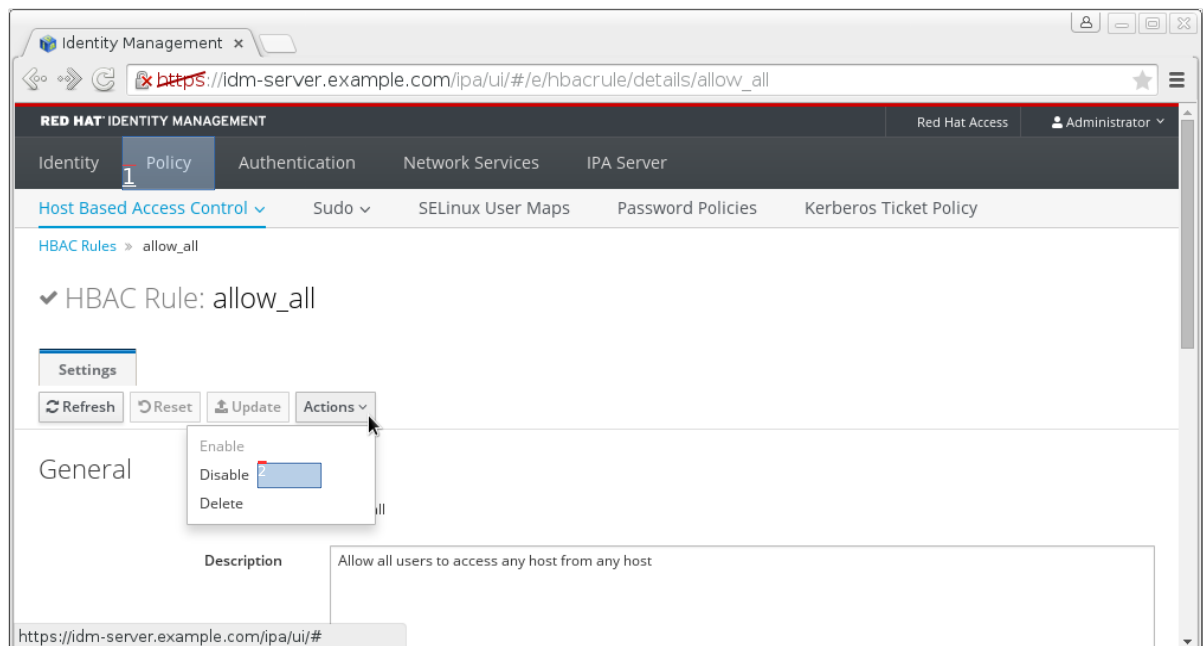
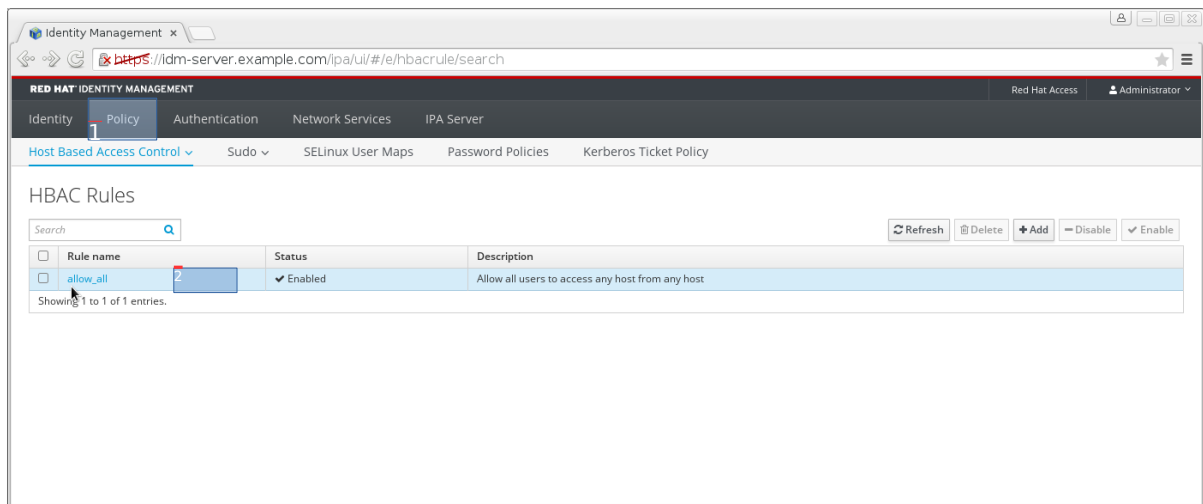
## Lab 7: Host Based Access Control – HBAC

**Target server:** idm-server.example.com, idm-client.example.com and idm-access.example.com

**Access:** ssh root@idm-server.example.com, ssh root@idm-client.example.com and ssh root@idm-client.example.com

In this Lab we will restrict/allow access based on host groups that we defined in the previous labs. By default IdM is having allow access permission to all resources, we could disable it during the installation time through **--no\_hbac\_allow**.

Disable the default allow\_all rule through web interface.



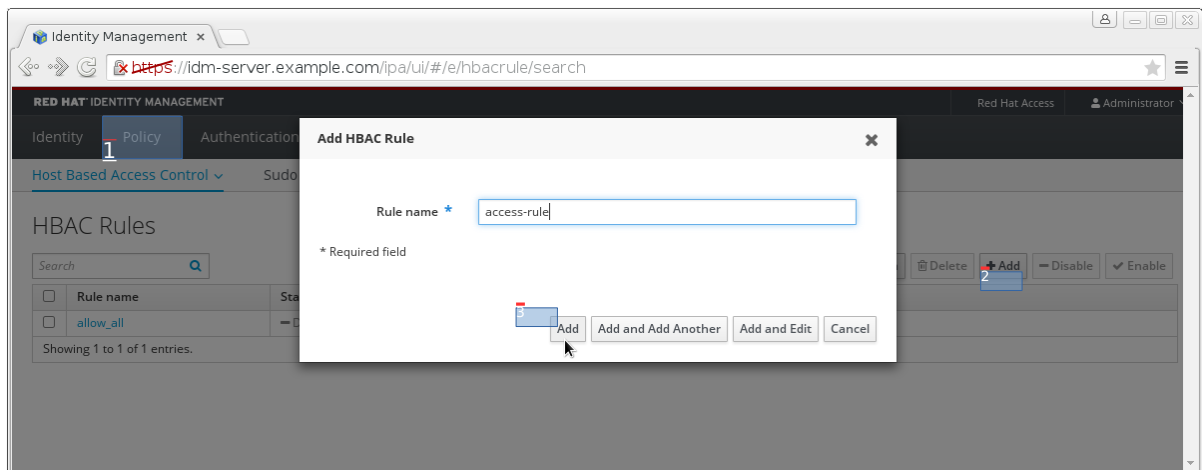
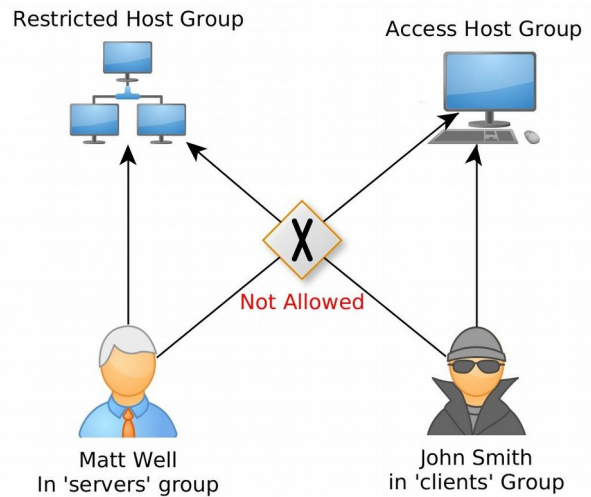
We want to grant access permissions to users in “**servers**” group to access all machines considering the following:

- Users in “**servers**” group can access “**restricted**” host group servers.
- Users in “**clients**” group can login into “**access**” host groups only.

The HBAC defines who can access which resources within the environment, not the level of access. This is called host-based access control because the rule defines what hosts are allowed to access other hosts within the domain.

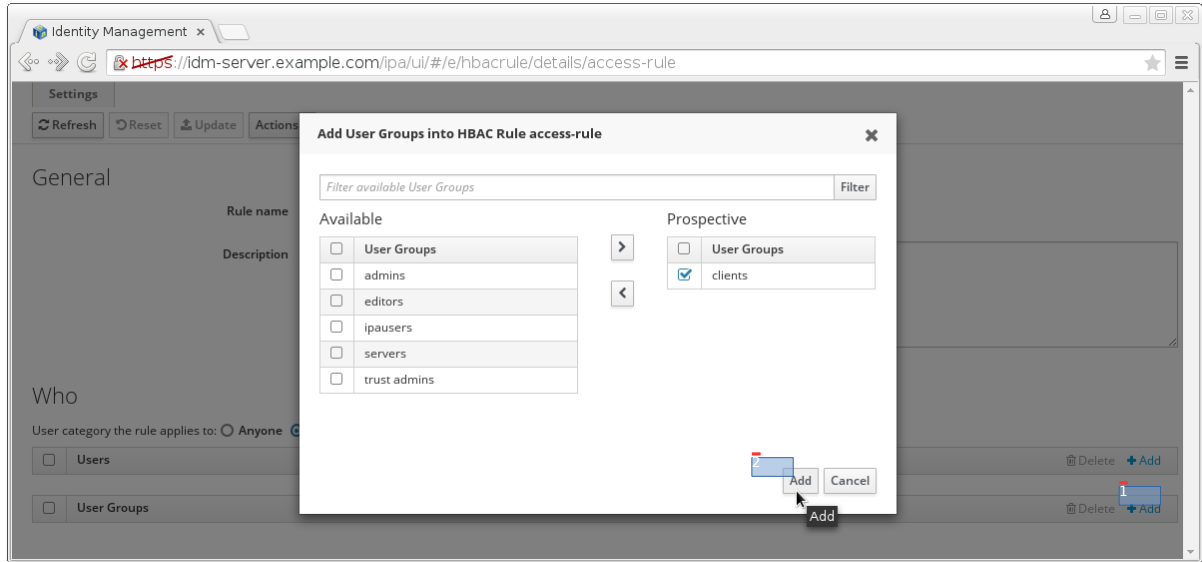
Four basic elements to construct HBAC rule:

- Who: The rule applies to.
- Where: Hosts users can access.
- How: What login services can be accessed.
- Setting Host-Based Access control Rules.
- HBAC Rule with name “**access-rule**” through the web interface.

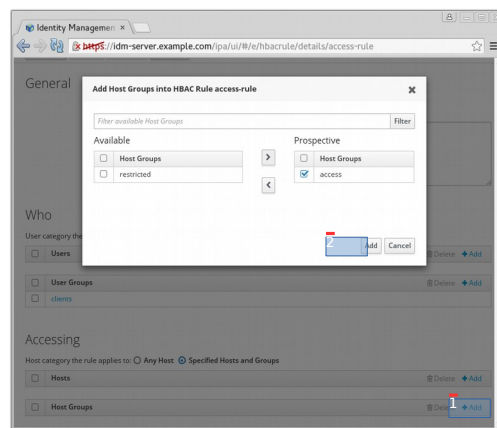


Click on the **access-rule** HBAC and add users or users groups that this rule will be applied on.

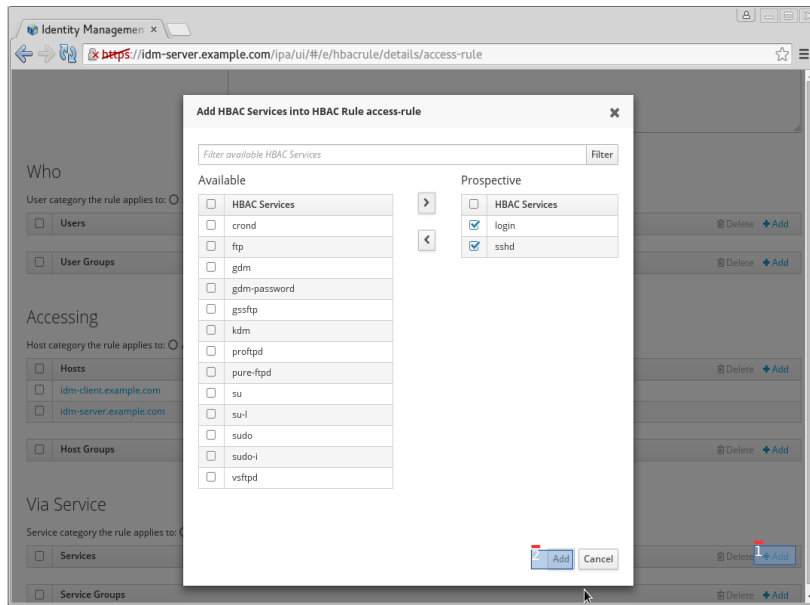
Add “**clients**” users group to the access-rule in WHO field.



Then add the resources that will have these rules applied either host groups or specific hosts (to access-rule), in this lab the access group should be used, if you didn't create it, you can add host instead of host group, select the `idm-access.example.com`.



Now we want to add the service that will be allowed, select the sshd and login services:



- In previous steps we created the access-rule that will allow “**clients**” users group to access servers in “**Access**” host group, Since Access host group doesn't have any other server except idm-access.example.com; we allowed access to idm-access.example.com or any server that will be added to this host group
- Create additional HBAC with name “**restricted-rule**” that allows “**servers**” users group to access servers in “**restricted**” host group using the previous steps used to create the “**allow-rule**”. So the steps are adding “**servers**” user group, Accessing “**restricted**” host group and services via “ssh and login”
- Testing Host-Based Access control Rules:
  - User mwell can ssh to idm-client.example.com successful.
  - User mwell will find access denied message if tried to ssh to “**access**”.
  - User jsmith can login via ssh to access.example.com.

**Reference:** [Red Hat Documentation : Managing Host Based Access Control](#)

## Lab 8: IdM Roles Management

IdM Role Management provides rights or permissions that users have been granted to perform operations within IdM on other users or objects:

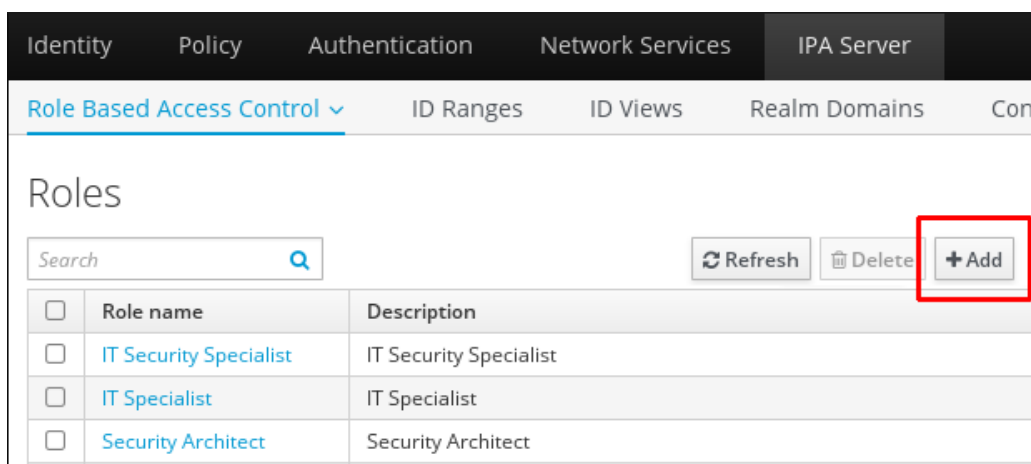
- Who can perform the operation.
- What can be accessed.
- What type of operation can be performed.
- Existing Predefined Roles.

Role-based access control grants a very different kind of authority to users compared to self-service and delegation access controls. Role-based access controls are fundamentally administrative, with the potential to, for example, add, delete, or significantly modify entries.

In this lab we will provide privileges to mwell or his group to change his/their group membership

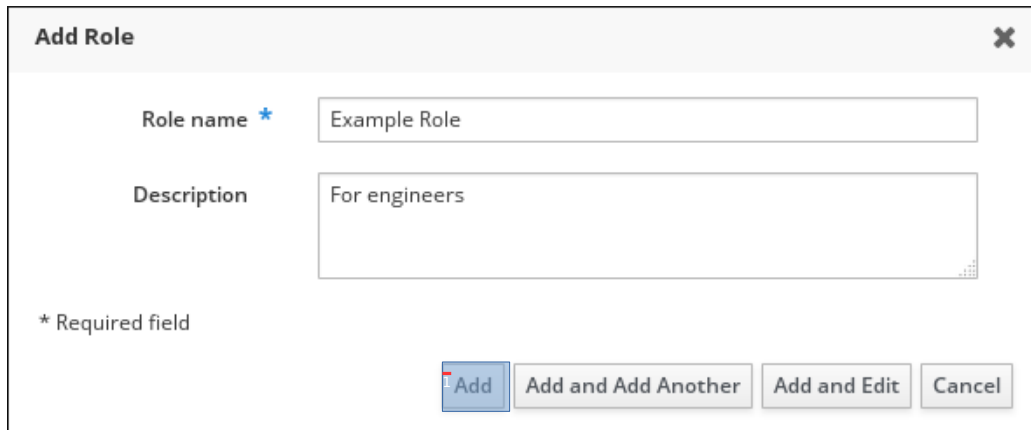
Open the “**IPA Server**” tab in the top menu, and select the “**Role Based Access Control**” subtab.

Click the “**Add**” link at the top of the list of role-based ACIs:



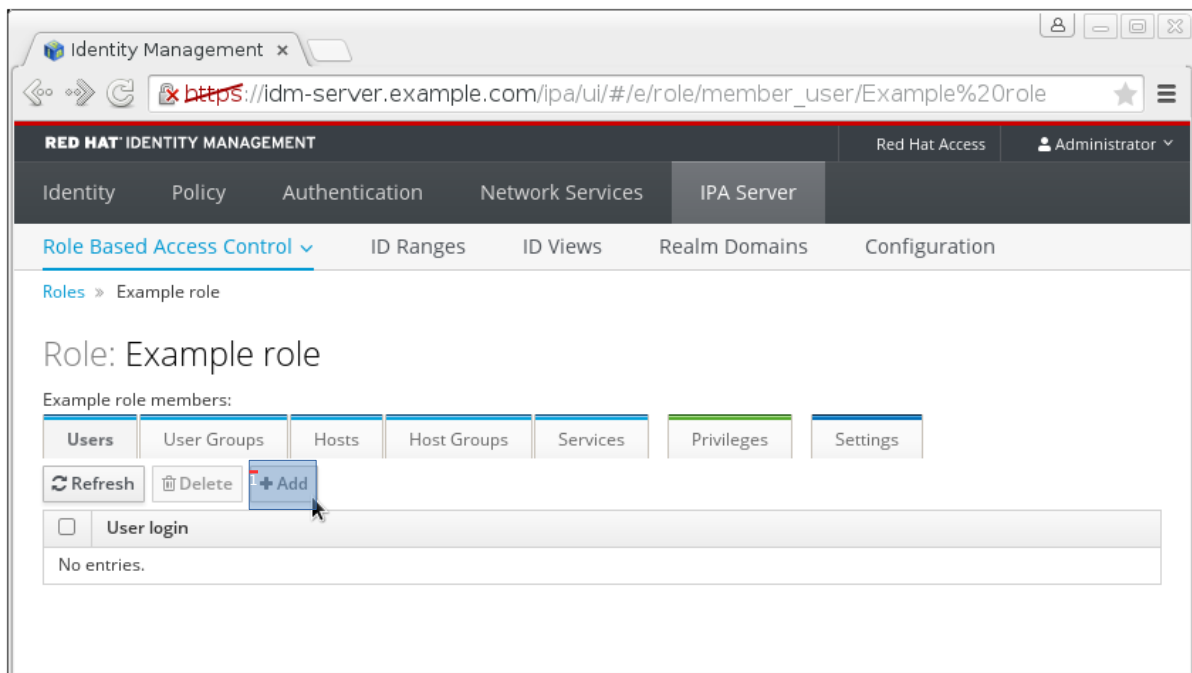


Enter the role name and a description:

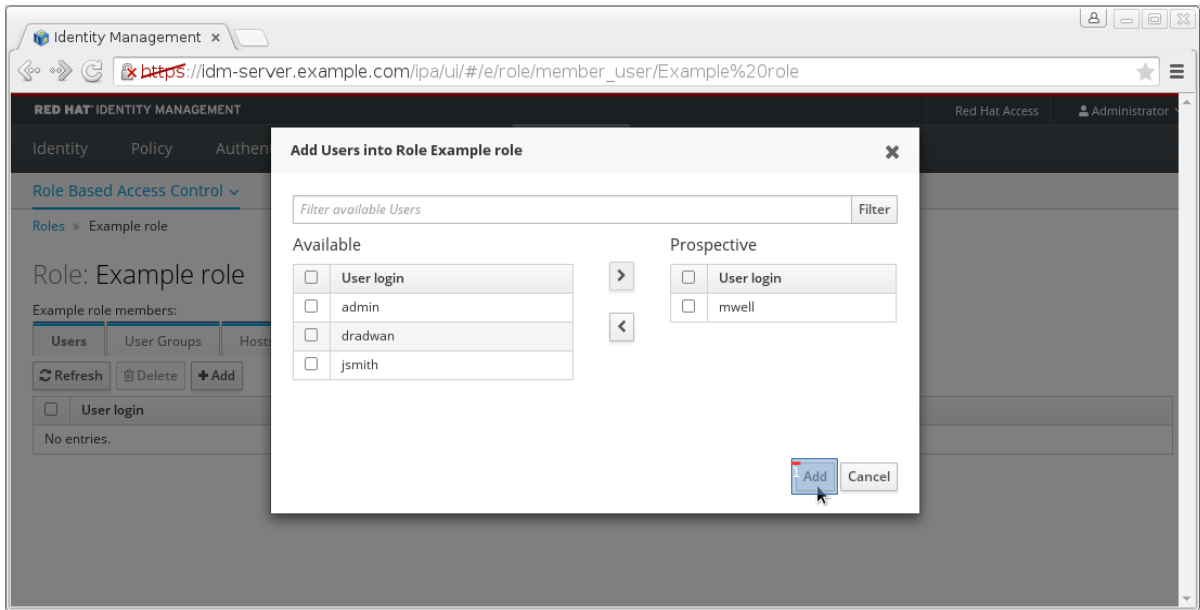


Click the **“Add and Edit”** button to save the new role and go to the configuration page.

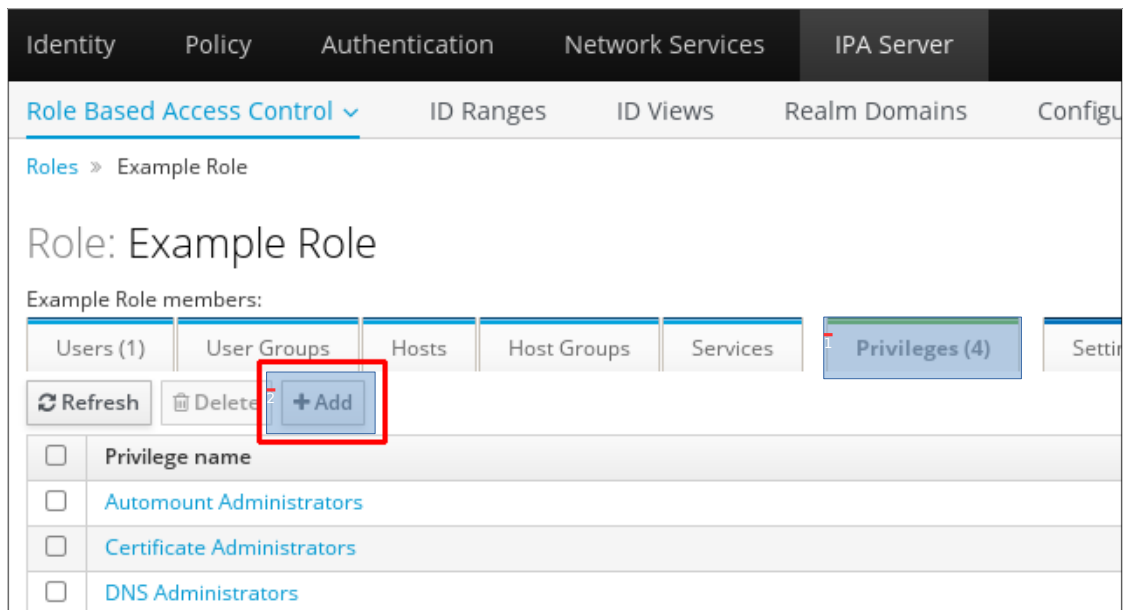
Click on the Role that you just created, then click on **“Add”**



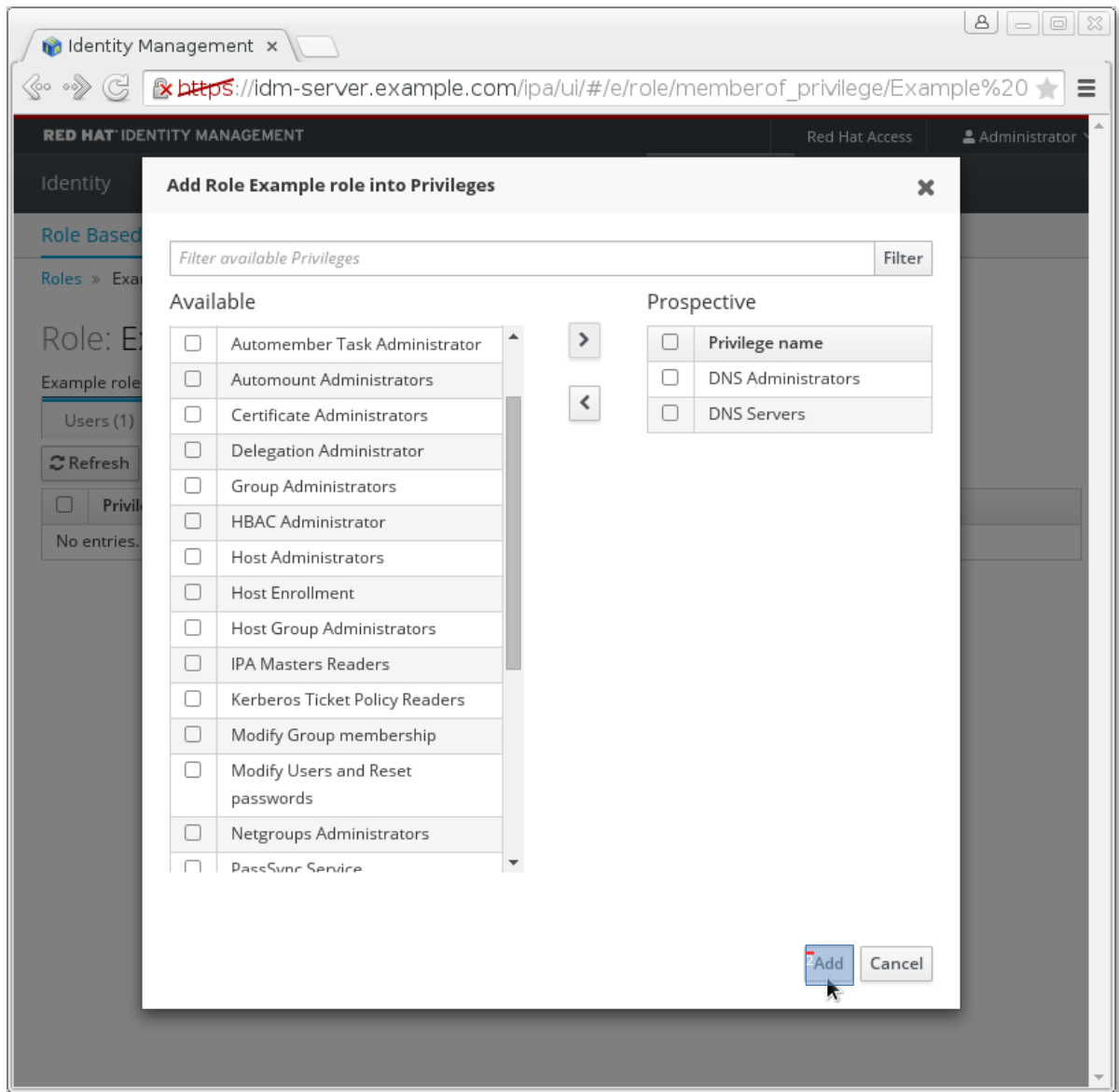
Select the users on the left and use the ">" button to move them to the "Prospective" column.



At the top of the "Privileges" tab, click "Add".

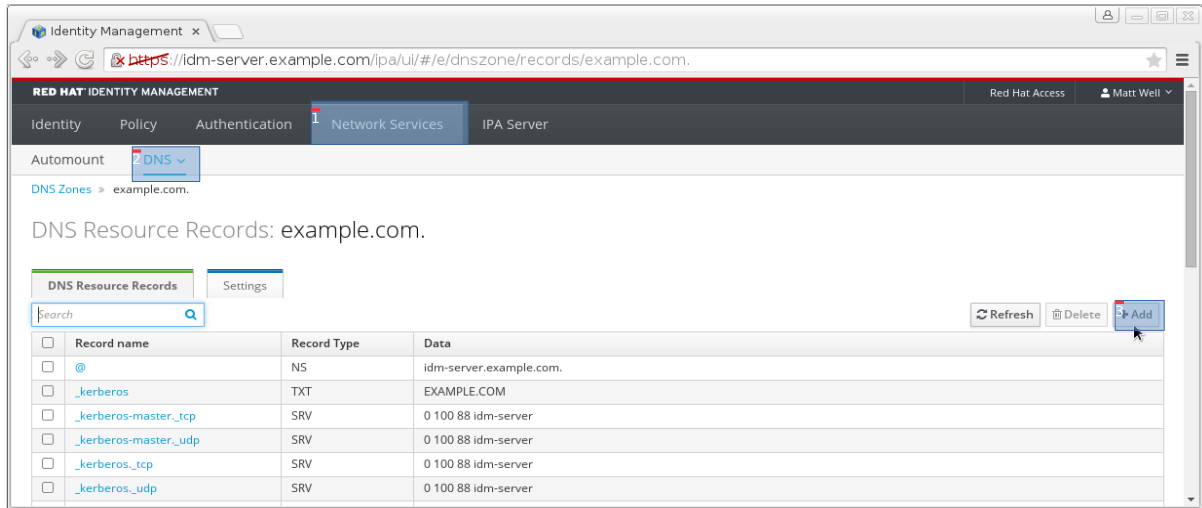


Select the privileges on the left and use the ">" button to move them to the "**Prospective**" column.

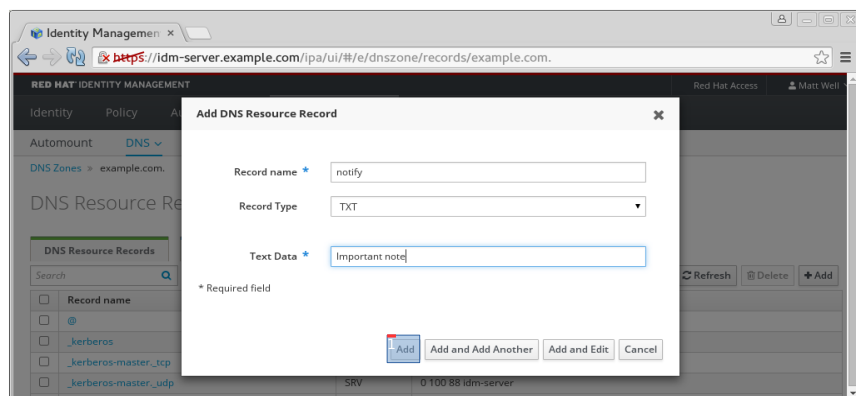


Click the "**Add**" button to save.

Log out the admin user, and login with mwell user. Navigate through “**Network Services**”, then DNS subtab menu ► then click on example.com. After getting example.com resources; click on “**Add**”:



We need to test that user “**mwell**” can add new records, create new record “**notify**” TXT record with text data “Important note” :



User mwell, will be able to add dns records to example.com domain.

```
[root@idm-server ~]# dig TXT notify.example.com | grep Important
notify.example.com.      86400 IN      TXT      "Important" "note"
```

**Reference:** [Red Hat Documentation : Roles Management](#)

## Lab 9: IdM Replication

**Target server:** idm-server.example.com and idm-replica.example.com

**Access:** ssh [root@idm-server.example.com](#) ssh [root@idm-replica.example.com](#)

On the idm-replica.example.com run:

```
yum install ipa-server bind-dyndb-ldap
```

On The idm-server.example.com run:

```
ipa-replica-prepare idm-replica.example.com --ip-address 192.168.10.13
```

Copy the replication info to the replica:

```
scp /var/lib/ipa/replica-info-idm-replica.example.com.gpg \  
root@192.168.10.13:/var
```

On idm-replica.example.com run:

```
ipa-replica-install --no-forwarders --skip-conncheck --setup-dns \  
/var/replica-info-idm-replica.example.com.gpg
```

Other options:

```
ipa-replica-install --forwarder=<our forward DNS> --setup-dns <replica  
file.gpg>
```

Replication verification.

```
ipa-replica-manage list  
ipa-replica-conncheck --replica idm-replica.example.com
```

**Reference:** [Red Hat Documentation Managing the Server-Replica Relationships](#)

## Lab 10: Services and Keytabs

**Target server:** idm-server.example.com or idm-client.example.com

**Access:** ssh **root@idm-server.example.com** ssh **root@idm-client.example.com**

Log in to idm-access machine:

```
yum install httpd mod_nss mod_wsgi mod_auth_kerb ipa-admintools
```

Prepare content for idm-access:

```
cp workshop.conf /etc/httpd/conf.d/workshop.conf
cp workshop.wsgi /var/www/cgi-bin/workshop.wsgi
chmod +x /var/www/cgi-bin/workshop.wsgi
```

Create the IPA service entry for idm-access:

```
kinit
Password for admin@EXAMPLE.COM:
ipa service-add HTTP/`hostname`
ipa service-show HTTP/`hostname`
```

Retrieve a keytab for httpd service on idm-access:

```
ipa-getkeytab -p HTTP/`hostname` -k http.keytab -s idm-server.example.com
klist -kt http.keytab
```

Configure idm-access to use the keytab:

```
mv http.keytab /etc/httpd/conf/
chown apache:apache /etc/httpd/conf/http.keytab
chmod 0400 /etc/httpd/conf/http.keytab
service httpd restart
```

Access idm-client and run:

```
yum install firefox xorg-x11-xinit.x86_64
exit
ssh root@ idm-client.example.com -X
firefox
```

In Firefox, access idm-access.example.com/test, when you exit Firefox check:

```
klist
```

It might not work as selinux will deny the http-keytab.

```
Cd /root
grep httpd_t /var/log/audit/audit.log | audit2allow -m http-keytab > http-
keytab.te
grep httpd_t /var/log/audit/audit.log | audit2allow -M http-keytab
semodule -i http-keytab.pp
```

Now, check again Firefox, after authentication it should print:

```
Hello!
Received connection from 192.168.10.11
YAY! Kerberos authentication works!
Remote user is admin@EXAMPLE.COM
```

To Allow authentication for this small web application without password:

- In the address bar of Firefox, type 'about:config' to display the list of current configuration options.
- In the Filter field, type negotiate to restrict the list of options.
- Double-click the network.negotiate-auth.trusted-uris entry to display the Enter string value dialog box.
- Enter the name of the domain against which you want to authenticate, for example, example.com.
- Repeat the above procedure for the network.negotiate-auth.delegation-uris entry, using the same domain.
- Restart Firefox, you should see the “YAY” message with no user-name and password.