Up and Running with Red Hat Identity Manager

Presenters

Jim Wildman, Senior Solution Architect, Southeast Region
Chuck Mattern, Senior Solution Architect, Southeast Region
Tom Gamull, Solution Architect, Southeast Region

Abstract:

Red Hat Identity Management (IDM) can play a central role in user authentication, authorization and control. It can manage critical security components such as ssh keys, host based access controls and selinux contexts. These functions can be provided in a standalone environment or when in a trust relationship with a Microsoft Active Domain Controller.

Audience/Intro/Prerequisites:

This lab is geared towards system administrators and architects who want to gain a basic understanding of IDM installation and deployment

Attendees, during this session, will

- Perform basic installation and configuration of IDM
- Configuration of a replica
- User and host groups
- Sudo rules
- Ssh keys
- Host based access control
- Kerberized NFS home directories (automount)

We will use the web UI for demos and the command line interface for most of the lab exercises.

Up and Running with Red Hat Identity Manager

Overview

- All exercises will be on the local kvm network (192.168.122.x/24)
- Five RHEL 7.3 vm's
    - RHEL machines are text-console only
    - Hostnames/IPs
        - vm1: 192.168.122.11
        - vm2: 192.168.122.12
        - vm3: 192.168.122.13
        - vm4: 192.168.122.14
        - vm5: 192.168.122.15
- It is recommended that you utilize a Terminal session on your local machine and ssh to each VM you are working with as this will allow you to copy/paste as well as scroll back, neither of which you can accomplish from the VM console
- VM's point to one local repos
    - rhel-7.3 - for any distribution packages you may need
- Lab examples will use the command line.
- All examples from this document will begin with "class". Please don't name anything that way.
- You are welcome to work ahead, but please don't ask questions ahead of where we are.
- root password on all RHEL machines is: **changeme**
- For clarity all command to be entered will be in **bold** and all empty new lines will be shown as **<ENTER>**
- Before beginning, verify that forward and reverse DNS works for vm1 through vm5 of local.net

```
[root@vm2 ~]# cat /etc/resolv.conf
search local.net
nameserver 192.168.122.1
[root@vm2 ~]# host vm1.local.net
vm1.local.net has address 192.168.122.11
[root@vm2 ~]# host 192.168.122.11
11.122.168.192.in-addr.arpa domain name pointer vm1.local.net.
```

Up and Running with Red Hat Identity Manager

Lab 1: IDM Installation (15 minutes)

On machine `vm1`, install and configure IdM with DNS for a local network

- Install all the needed RPM packages.

```
[root@vm1 ~]# yum -y install ipa-server ipa-server-dns rng-tools
```

- These machines are not very active; the `ipa-server-install` utility will present a `running out of entropy` error message, then wait for more entropy to be created. Do the following before beginning the ipa-server install to make more entropy available

```
[root@vm1 ~]# rngd -r /dev/urandom -o /dev/random
```

- Run the ipa-server-install command to install the IdM server. With this install, we will enable the automatic creation of IDM user home directories on the system.
- Set both passwords to **changeme**
- Answer yes to the indicated (>) questions and provide 192.168.122.1 for the forwarder IP

```
[root@vm1 ~]# ipa-server-install --mkhomedir --allow-zone-overlap

The log file for this installation can be found in
/var/log/ipaserver-install.log

==============================================================================
=======
This program will set up the IPA Server.
This includes:
  * Configure a stand-alone CA (dogtag) for certificate management
  * Configure the Network Time Daemon (ntpd)
```

```
  * Create and configure an instance of Directory Server
  * Create and configure a Kerberos Key Distribution Center (KDC)
  * Configure Apache (httpd)
To accept the default shown in brackets, press the Enter key.

WARNING: conflicting time&date synchronization service 'chronyd' will
be disabled in favor of ntpd

Do you want to configure integrated DNS (BIND)? [no]: yes

 Enter the fully qualified domain name of the computer
 on which you're setting up server software. Using the form
 <hostname>.<domainname>
 Example: master.example.com.
 Server host name [vm1.local.net]: <ENTER>

 Warning: skipping DNS resolution of host vm1.local.net
 The domain name has been determined based on the host name.

 Please confirm the domain name [local.net]: <ENTER>

 The kerberos protocol requires a Realm name to be defined.
 This is typically the domain name converted to uppercase.

 Please provide a realm name [LOCAL.NET]: <ENTER>
 Certain directory server operations require an administrative user.
This user is referred to as the Directory Manager and has full access
to the Directory for system management tasks and will be added to the
instance of directory server created for IPA.
The password must be at least 8 characters long.

Directory Manager password: changeme
```

```
Password (confirm): changeme

The IPA server requires an administrative user, named 'admin'.
This user is a regular system account used for IPA server
administration.
IPA admin password: changeme
Password (confirm): changeme
Checking DNS domain local.net, please wait ...
Do you want to configure DNS forwarders? [yes]: <ENTER>
Following DNS servers are configured in /etc/resolv.conf: 192.168.122.1
  Do you want to configure these servers as DNS forwarders? [yes]:
<ENTER>
  All DNS servers from /etc/resolv.conf were added. You can enter
  additional addresses now: <ENTER>
  Enter an IP address for a DNS forwarder, or press Enter to skip:
  Checking DNS forwarders, please wait ...
  Do you want to search for missing reverse zones? [yes]: <ENTER>

  The IPA Master Server will be configured with:
  Hostname:       vm1.local.net
  IP address(es): 192.168.122.11
  Domain name:    local.net
  Realm name:    LOCAL.NET

  BIND DNS server will be configured to serve IPA domain with:
  Forwarders:       192.168.122.1
Forward policy:   only
Reverse zone(s):  No reverse zone

Continue to configure the system with these values? [no]: yes

The following operations may take some minutes to complete.
```

Up and Running with Red Hat Identity Manager

Please wait until the prompt is returned.

- Verify the state of the firewall on the `vm1` system.

```
[root@vm1 ~]# systemctl status firewalld
```

- Enable `firewalld`, ensuring that it will be started at every system boot.

```
[root@vm1 ~]# systemctl enable firewalld
```

- If `firewalld` is not running, start the service.

```
[root@vm1 ~]# systemctl start firewalld
```

- Add port rules to the firewall so that connections to IdM are permitted.

```
[root@vm1 ~]# firewall-cmd --permanent \
--add-port={80/tcp,443/tcp,389/tcp,636/tcp,88/tcp,\
464/tcp,53/tcp,88/udp,464/udp,53/udp,123/udp}
```

- Reload the firewall so that these port exemptions are loaded

```
[root@vm1 ~]# firewall-cmd --reload
```

- Your `firewalld` default should now appear as follows:

```
[root@vm1 ~]# firewall-cmd --list-all
public (default, active)
  interfaces: eth0
  sources:
  services: dhcpv6-client ssh
  ports: 443/tcp 80/tcp 464/tcp 88/udp 464/udp 88/tcp 123/udp
389/tcp 53/tcp 53/udp 636/tcp
  masquerade: no
  forward-ports:
  icmp-blocks:
  rich rules:
```

Verify the installed IDM Server

- Open a web browser on your host machine. Login to the web console (user admin and the password you gave during installation) for the IdM server at: `http://192.168.122.10` and verify the following:
  - That you can login
  - There are no hosts nor users.

- Verify that an initial kerberos ticket can be created. Once this ticket is obtained, we will be able to run ipa commands.

```
[root@vm1 ~]# kinit admin
Password for admin@LOCAL.NET: changeme
[root@vm1 ~]# klist
Ticket cache: KEYRING:persistent:0:0
Default principal: admin@LOCAL.NET

Valid starting       Expires              Service principal
06/10/2016 22:13:38  06/11/2016 22:13:35
krbtgt/LOCAL.NET@LOCAL.NET
```

- Validate that a DNS record exists for idm.local.net.

```
[root@vm1 ~]# ipa host-show vm1.local.net
  Host name: vm1.local.net
  Principal name: host/vm1.local.net@LOCAL.NET
  Password: False
  Keytab: True
  Managed by: vm1.local.net
  SSH public key fingerprint: (output truncated)
```

- Verify that the DNS server settings are correct and that hostname resolutions are working properly.
-

```
[root@vm1 ~]# cat /etc/resolv.conf
# Generated by NetworkManager
search local.net
nameserver 192.168.122.11
nameserver 192.168.122.1

[root@vm1 ~]# host vm1.local.net
vm1.local.net has address 192.168.122.11
```

Up and Running with Red Hat Identity Manager

Congratulations, you now have a running IDM server!!!

Up and Running with Red Hat Identity Manager

Lab 2: Install IdM Client Systems (10 minutes)

- The following commands will need to be repeated for all 4 client machines (vm2, vm3, vm4, vm5). Remember to verify /etc/resolv.conf on each
- ```
  [root@vm2 ~]# yum -y install ipa-client rng-tools
  [root@vm2 ~]# rngd -r /dev/urandom -o /dev/random
  [root@vm2 ~]# ipa-client-install --mkhomedir
  Discovery was successful!
  Client hostname: vm2.local.net
  Realm: LOCAL.NET
  DNS Domain: local.net
  IPA Server: vm1.local.net
  BaseDN: dc=local,dc=net

  Continue to configure the system with these values? [no]: yes
  Synchronizing time with KDC...
  Attempting to sync time using ntpd.  Will timeout after 15
  seconds
  User authorized to enroll computers: admin
  Password for admin@LOCAL.NET:changme
  Successfully retrieved CA cert
  (output truncated)

  Client configuration complete.
  ```

- Verify the hosts are in the web interface
  ```

Up and Running with Red Hat Identity Manager

- Verify the hosts in the CLI

```
[root@vm1 ~]# ipa host-find vm
---------------
5 hosts matched
---------------
  Host name: vm1.local.net
  Principal name: host/vm1.local.net@LOCAL.NET
  Principal alias: host/vm1.local.net@LOCAL.NET
  SSH public key fingerprint: (truncated)

  Host name: vm2.local.net
  Principal name: host/vm2.local.net@LOCAL.NET
  Principal alias: host/vm2.local.net@LOCAL.NET
  SSH public key fingerprint: (truncated)

  Host name: vm3.local.net
  Principal name: host/vm3.local.net@LOCAL.NET
  Principal alias: host/vm3.local.net@LOCAL.NET
  SSH public key fingerprint:(truncated)

  Host name: vm4.local.net
  Principal name: host/vm4.local.net@LOCAL.NET
  Principal alias: host/vm4.local.net@LOCAL.NET
  SSH public key fingerprint: (truncated)

  Host name: vm5.local.net
  Principal name: host/vm5.local.net@LOCAL.NET
  Principal alias: host/vm5.local.net@LOCAL.NET
  SSH public key fingerprint: (truncated)
----------------------------
Number of entries returned 5
----------------------------
```

Lab 3: User and group creation

- We will create users from the vm1 machine.

Add a random user

- Add a user of your own choosing. The password will be reset on login.

```
[root@vm1 ~]# ipa user-add user1 --password
First name: Jim
Last name: Wildman
Password:
Enter Password again to verify:
----------------------
Added user "user1"
----------------------
  User login: user1
  First name: Jim
  Last name: Wildman
  Full name: Jim Wildman
  Display name: Jim Wildman
  Initials: JW
  Home directory: /home/user1
  GECOS: Jim Wildman
  Login shell: /bin/sh
  Kerberos principal: user1@LOCAL.NET
  Email address: user1@local.net
  UID: 1196400001
  GID: 1196400001
  Password: True
  Member of groups: ipausers
  Kerberos keys available: True
```

Add the class users

- We will add 4 users with specific UID's and GID's for later use. The ipa user-add command will accept a password echo'd into the standard input to allow unattended usage. The password for each user will be reset on first login

```
[root@vm1 ~]# for i in 1 2 3 4; do echo "changeme" | ipa user-add
classuser$i --password --first=User$i --last=Class --uid=1000$i;
echo ""; done
```

```
(Output truncated)
classuser4
-----------------------
Added user "classuser4"
-----------------------
  User login: classuser4
  First name: User4
  Last name: Class
  Full name: User4 Class
  Display name: User4 Class
  Initials: UC
  Home directory: /home/classuser4
  GECOS: User4 Class
  Login shell: /bin/sh
  Kerberos principal: classuser4@LOCAL.NET
  Email address: classuser4@local.net
  UID: 10004
  GID: 10004
  Password: True
  Member of groups: ipausers
  Kerberos keys available: True
(output truncated)
```

Verify the users were created

```
[root@vm1 ~]# ipa user-find class
---------------
4 users matched
---------------
  User login: classuser1
  First name: User1
  Last name: Class
 Home directory: /home/classuser1
(output truncated)
```

Verify that you can login as classuser1

```
[user@host ~]# ssh classuser1@vm1.local.net
The authenticity of host 'vm1.local.net (192.168.122.11)' can't be
established.
```

```
ECDSA key fingerprint is
8b:e4:53:63:5f:54:30:ec:34:ef:3f:48:63:ef:b3:72.
Are you sure you want to continue connecting (yes/no)? yes
(truncated, will have to reset password)

[root@jwildman VirtualMachines]# ssh classuser1@vm1.local.net
classuser1@vm1.local.net's password:
Last login: Mon Jun 20 08:57:15 2016 from 192.168.122.1
Could not chdir to home directory /home/classuser1: No such file or
directory
-sh-4.2$ klist
Ticket cache: KEYRING:persistent:10001:krb_ccache_NmmOGA9
Default principal: classuser1@LOCAL.NET

Valid starting       Expires              Service principal
06/20/2016 08:57:27  06/21/2016 08:57:27  krbtgt/LOCAL.NET@LOCAL.NET
-sh-4.2$ hostname
Vm1.local.net
```

Verify SSO works

```
-sh-4.2$ ssh vm2
Could not create directory '/home/classuser1/.ssh'.
Could not chdir to home directory /home/classuser1: No such file or
directory
-sh-4.2$ hostname
vm2.local.net
```

Generate and add user ssh keys.

- On the vm1 box, as root run

```
[root@vm1 ~]# ssh-keygen -t dsa -f ~/.ssh/classuser
Generating public/private dsa key pair.
Enter passphrase (empty for no passphrase): <RETURN>
Enter same passphrase again: <RETURN>
Your identification has been saved in /root/.ssh/classuser.
Your public key has been saved in /root/.ssh/classuser.pub.
The key fingerprint is:
d5:2a:25:c9:ed:83:79:f8:49:97:c6:d6:02:f2:92:22
root@vm1.local.net
```

Up and Running with Red Hat Identity Manager

```
The key's randomart image is:
+--[ DSA 1024]----+
|                 |
|       . o .     |
|        = = .    |
|         % + o   |
|    E . S B B .  |
|     . . * * .   |
|          o      |
|                 |
|                 |
+-----------------+
```

- Copy the public key onto your clipboard and go to a root window on idm server and add the key. You can add it to all the users similarly

```
[root@vm1 ~]# cat .ssh/classuser.pub
ssh-dss (truncated) UMQ== root@vm1.local.net
(copy the ssh key and paste below)
[root@vm1 ~]# ipa user-mod classuser1 --sshpubkey="ssh-dss…..."
---------------------------
Modified user "classuser1"
---------------------------
  User login: classuser1
  First name: User1
  Last name: Class
  Home directory: /home/classuser1
  Login shell: /bin/sh
  Email address: classuser1@local.net
  UID: 10001
  GID: 10001
  SSH public key: (truncatted)
  Account disabled: False
  Password: True
  Member of groups: ipausers
  Kerberos keys available: True
  SSH public key fingerprint:
5C:CD:6E:3A:0A:20:69:BB:E4:16:19:4C:3F:BC:62:58
root@vm1.local.net
```

```
(ssh-dss)
```

- From the vm1 box, you can use the ssh key to login to any of the guest vm's. You will get an error about missing home directories. We will fix that in a later lab.

```
[root@vm1 ~]# ssh -i ~/.ssh/classuser classuser1@vm2.local.net
Last login: Mon Jun 20 08:57:27 2016 from 192.168.122.1
Could not chdir to home directory /home/classuser1: No such file
or directory
-sh-4.2$ klist
Ticket cache: KEYRING:persistent:10001:krb_ccache_NmmOGA9
Default principal: classuser1@LOCAL.NET

Valid starting       Expires              Service principal
06/20/2016 08:57:33  06/21/2016 08:57:27
host/vm2.local.net@LOCAL.NET
06/20/2016 08:57:27  06/21/2016 08:57:27
krbtgt/LOCAL.NET@LOCAL.NET
```

Create user groups.

- We want to create 2 user groups to demonstrate host based access control. We will put the odd users in one group and the even users in another.

```
[root@vm1 ~]# ipa group-add classevenusers
----------------------------
Added group "classevenusers"
----------------------------
  Group name: classevenusers
  GID: 1196400003
[root@vm1 ~]# ipa group-add classoddusers
(truncated)
[root@vm1 ~]# ipa group-add-member classevenusers
--users=classuser2
  Group name: classevenusers
  GID: 1196400003
  Member users: classuser2
-------------------------
Number of members added 1
-------------------------
```

```
[root@vm1 ~]# ipa group-add-member classevenusers
--users=classuser4
[root@vm1 ~]# ipa group-add-member classoddusers --users=classuser1
[root@vm1 ~]# ipa group-add-member classoddusers --users=classuser3
[root@vm1 ~]# ipa group-find class --all
----------------
2 groups matched
----------------
  Group name: classevenusers
  GID: 1196400003
  Member users: classuser2, classuser4

  Group name: classoddusers
  GID: 1196400004
----------------------------
Number of entries returned 2
----------------------------
```

Create host groups

- Create similar host groups with vm1 and 3 in classoddhosts, vm2 and 4 in classevenhosts

```
[root@vm1 ~]# ipa hostgroup-add classevenhosts
[root@vm1 ~]# ipa hostgroup-add classoddhosts
[root@vm1 ~]# ipa hostgroup-add-member classoddhosts
--hosts=vm1.local.net
[root@vm1 ~]# ipa hostgroup-add-member classevenhosts
--hosts=vm2.local.net
[root@vm1 ~]# ipa hostgroup-add-member classoddhosts
--hosts=vm3.local.net
[root@vm1 ~]# ipa hostgroup-add-member classevenhosts
--hosts=vm4.local.net
[root@vm1 ~]# ipa hostgroup-add-member classoddhosts
--hosts=vm5.local.net
  Host-group: classevenhosts
  Member hosts: vm2.local.net, vm4.local.net
-------------------------
Number of members added 1
-------------------------
```

- Verify that the groups were created correctly

```
[root@vm1 ~]# ipa hostgroup-find class --all
```

```
--------------------
2 hostgroups matched
--------------------
  dn: cn=classevenhosts,cn=hostgroups,cn=accounts,dc=local,dc=net
  Host-group: classevenhosts
  Member hosts: vm2.local.net, vm4.local.net
  ipauniqueid: 185963e0-2889-11e7-a9d7-5254000dbeb5
  mepmanagedentry: cn=classevenhosts,cn=ng,cn=alt,dc=local,dc=net
  objectclass: ipaobject, mepOriginEntry, top, ipahostgroup,
groupOfNames, nestedGroup

  dn: cn=classoddhosts,cn=hostgroups,cn=accounts,dc=local,dc=net
  Host-group: classoddhosts
  Member hosts: vm3.local.net, vm5.local.net
  ipauniqueid: 1a3f5480-2889-11e7-b48f-5254000dbeb5
  mepmanagedentry: cn=classoddhosts,cn=ng,cn=alt,dc=local,dc=net
  objectclass: ipaobject, mepOriginEntry, top, ipahostgroup,
groupOfNames, nestedGroup
----------------------------
Number of entries returned 2
----------------------------
```

Up and Running with Red Hat Identity Manager

Lab 4: Configure sudo

- We will configure sudo to allow members of the classevenusers group (from Lab 3) to run a few commands as root.
- Add commands to be run via sudo

```
[root@vm1 ~]# ipa sudocmd-add /sbin/lvdisplay
----------------------------------
Added Sudo Command "/sbin/lvdisplay"
----------------------------------
  Sudo Command: /sbin/lvdisplay
[root@vm1 ~]# ipa sudocmd-add /sbin/vgdisplay
----------------------------------
Added Sudo Command "/sbin/vgdisplay"
----------------------------------
  Sudo Command: /sbin/vgdisplay
```

- Create a rule to manage the commands and the users

```
[root@vm1 ~]# ipa sudorule-add vgcommand
---------------------------
Added Sudo Rule "vgcommand"
---------------------------
  Rule name: vgcommand
  Enabled: TRUE
```

- Add the commands and the users to the rule

```
[root@vm1 ~]# ipa sudorule-add-allow-command vgcommand
--sudocmds=/sbin/lvdisplay
  Rule name: vgcommand
  Enabled: TRUE
-------------------------
Number of members added 1
-------------------------
[root@vm1 ~]# ipa sudorule-add-allow-command vgcommand
--sudocmds=/sbin/vgdisplay
  Rule name: vgcommand
  Enabled: TRUE
-------------------------
Number of members added 1
-------------------------

[root@vm1 ~]# ipa sudorule-add-runasuser vgcommand --users=root
```

```
  Rule name: vgcommand
  Enabled: TRUE
  RunAs External User: root
-------------------------
Number of members added 1
[root@vm1 ~]# ipa sudorule-add-user --group=classevenusers
Rule name: vgcommand
  Rule name: vgcommand
  Enabled: TRUE
  User Groups: classevenusers
  RunAs External User: root
-------------------------
Number of members added 1
-------------------------
[root@vm1 ~]# ipa sudorule-find
-------------------
1 Sudo Rule matched
-------------------
  Rule name: vgcommand
  Enabled: TRUE
  User Groups: classevenusers
  RunAs External User: root
---------------------------
Number of entries returned 1
---------------------------
```

- Test the new rule by logging in as classuser2 and running the commands

```
[jwildman@jwildman ~]$ ssh classuser2@192.168.122.12
classuser2@192.168.122.12's password: classuser
Last login: Sun Jun 12 20:47:48 2016 from 192.168.122.1
-sh-4.2$ sudo /sbin/lvdisplay

We trust you have received the usual lecture from the local
System
Administrator. It usually boils down to these three things:

    #1) Respect the privacy of others.
    #2) Think before you type.
    #3) With great power comes great responsibility.

[sudo] password for classuser2:
```

```
classuser2 is not allowed to run sudo on vm2.  This incident will
be reported.
-sh-4.2$ sudo /sbin/vgdisplay
[sudo] password for classuser2:
classuser2 is not allowed to run sudo on vm2.  This incident will
be reported.
```

- Why did the command fail?
- Add the classevenhosts host group to the command


```
[root@vm1 ~]# ipa  sudorule-add-host vgcommand
[member host]: <RETURN>
[member host group]: classevenhosts
  Rule name: vgcommand
  Enabled: TRUE
  User Groups: classevenusers
  Host Groups: classevenhosts
  RunAs External User: root
-------------------------
Number of members added 1
-------------------------
[root@vm1 ~]# ipa sudorule-find --all
-------------------
1 Sudo Rule matched
-------------------
  dn: ipaUniqueID=ba28f820-2889-11e7-b737-5254000dbeb5,
cn=sudorules,cn=sudo,dc=local,dc=net
  Rule name: vgcommand
  Enabled: TRUE
  User Groups: classevenusers
  Host Groups: classevenhosts
  Sudo Allow Commands: /sbin/lvdisplay, /sbin/vgdisplay
  RunAs External User: root
  ipauniqueid: ba28f820-2889-11e7-b737-5254000dbeb5
  objectclass: ipasudorule, ipaassociation
----------------------------
Number of entries returned 1
```

- If you are logged in as an even user on an even box, logout and back in. Then execute the LVM commands that were added.
- As root, clear the sssd cache on the target VM

```
[root@vm2 ~]# sss_cache -E
```

Up and Running with Red Hat Identity Manager

```
[jwildman@jwildman ~]$ ssh classuser2@192.168.122.12
classuser2@192.168.122.12's password:

-sh-4.2$ sudo /sbin/vgdisplay
  --- Volume group ---
  VG Name               rhel_vm2
  System ID
  Format                lvm2
  (output truncated)
```

Lab 5: Configure Host Based Access Control

- Host Based Access Control (HBA) enables you to restrict which users have access to which hosts or groups of hosts. We are going to configure HBAC to allow classevenusers to access classevengroup, classoddusers to access classoddgroup
- Create the odd to odd rule

```
[root@vm1 ~]# ipa hbacrule-add class_odd_to_odd
--------------------------------
Added HBAC rule "class_odd_to_odd"
--------------------------------
  Rule name: class_odd_to_odd
  Enabled: TRUE
```

- Add the hosts

```
[root@vm1 ~]# ipa hbacrule-add-host class_odd_to_odd
--hostgroups=classoddhosts
  Rule name: class_odd_to_odd
  Enabled: TRUE
  Host Groups: classoddhosts
-------------------------
Number of members added 1
-------------------------
```

- Add the users

```
[root@vm1 ~]# ipa hbacrule-add-user class_odd_to_odd
--groups=classoddusers
  Rule name: class_odd_to_odd
  Enabled: TRUE
  User Groups: classoddusers
  Host Groups: classoddhosts
-------------------------
Number of members added 1
-------------------------
```

- Restrict the access to ssh only

```
[root@vm1 ~]# ipa hbacrule-add-service class_odd_to_odd
--hbacsvcs=sshd
  Rule name: class_odd_to_odd
```

```
  Enabled: TRUE
  User Groups: classoddusers
  Host Groups: classoddhosts
  Services: sshd
-------------------------
Number of members added 1
-------------------------
```

- Create the even to even rule

```
[root@vm1 ~]# ipa hbacrule-add class_even_to_even
---------------------------------
Added HBAC rule "class_even_to_even"
---------------------------------
  Rule name: class_even_to_even
  Enabled: TRUE
```

- Add the hosts

```
[root@vm1 ~]# ipa hbacrule-add-host class_even_to_even
--hostgroups=classevenhosts
  Rule name: class_even_to_even
  Enabled: TRUE
  Host Groups: classevenhosts
-------------------------
Number of members added 1
-------------------------
```

- Add the users

```
[root@vm1 ~]# ipa hbacrule-add-user class_even_to_even
--groups=classevenusers
  Rule name: class_even_to_even
  Enabled: TRUE
  User Groups: classevenusers
  Host Groups: classevenhosts
-------------------------
Number of members added 1
-------------------------
```

- Restrict it to ssd access

Up and Running with Red Hat Identity Manager

```
[root@vm1 ~]# ipa hbacrule-add-service class_even_to_even
--hbacsvcs=sshd
  Rule name: class_even_to_even
  Enabled: TRUE
  User Groups: classevenusers
  Host Groups: classevenhosts
  Services: sshd
-------------------------
Number of members added 1
-------------------------
```

- Display the rules. Anybody see a problem?

```
[root@vm1 ~]# ipa hbacrule-find
--------------------
3 HBAC rules matched
--------------------
  Rule name: allow_all
  User category: all
  Host category: all
  Service category: all
  Description: Allow all users to access any host from any host
  Enabled: TRUE

  Rule name: class_even_to_even
  Enabled: TRUE
  User Groups: classevenusers
  Host Groups: classevenhosts
  Services: sshd

  Rule name: class_odd_to_odd
  Enabled: TRUE
  User Groups: classoddusers
  Host Groups: classoddhosts
  Services: sshd
----------------------------
Number of entries returned 3
```

- Disable the allow all rule

```
[root@vm1 ~]# ipa hbacrule-disable
Rule name: allow_all
-----------------------------
```

```
Disabled HBAC rule "allow_all"
------------------------------
```

- Test the rules

```
[root@vm1 ~]# ssh classuser2@192.168.122.12
classuser2@192.168.122.12's password:
Last login: Sun Jun 12 21:25:23 2016 from 192.168.122.1
-sh-4.2$ exit
logout
Connection to 192.168.122.12 closed.
[root@vm1 ~]# ssh classuser2@192.168.122.13
classuser2@192.168.122.13's password:
Connection closed by UNKNOWN
[root@vm1 ~]# ssh classuser1@192.168.122.13
classuser1@192.168.122.13's password:
Last failed login: Sun Jun 12 21:50:54 EDT 2016 from
idm.local.net on ssh:notty
Last login: Sun Jun 12 20:15:44 2016 from 192.168.122.1
-sh-4.2$ exit
logout
Connection to 192.168.122.13 closed.
[root@vm1 ~]# ssh classuser1@192.168.122.12
classuser1@192.168.122.12's password:
Connection closed by UNKNOWN
```

Lab 6: Configure IdM Replication

- IdM is backed by a multimaster, LDAP database. RHEL supports up to 60 IdM replicas out of the box with RHEL 6 and 7 clients. We are going to configure a simple 2 node replica
- Starting in RHEL 7.3, the replication installation process has changed.
  - Register as a client
  - Run ipa-replica-install to promote to a replica
  - We will use vm2

```
[root@vm2 ~]# yum -y install ipa-server ipa-server-dns
```

- Open up the firewall rules and enable the entropy

```
[root@vm2] firewall-cmd --permanent
--add-port={80/tcp,443/tcp,389/tcp,636/tcp,88/tcp,
464/tcp,53/tcp,88/udp,464/udp,53/udp,123/udp}
firewall-cmd --reload
[root@vm2] rngd -r /dev/urandom -o /dev/random
```

- Install the replica

```
[root@vm2 ~]# ipa-replica-install --setup-ca --allow-zone-overlap
WARNING: conflicting time&date synchronization service 'chronyd'
will
be disabled in favor of ntpd

Password for admin@LOCAL.NET:
ipa          : ERROR    Reverse DNS resolution of address
192.168.122.11 (vm1.local.net) failed. Clients may not function
properly. Please check yo
ur DNS setup. (Note that this check queries IPA DNS directly and
ignores /etc/hosts.)
Continue? [no]: yes
Run connection check to master
Connection check OK
```

- Check for the user ‘test’ on vm1

```
[root@vm1 ~]# ipa user-show test
ipa: ERROR: test: user not found
```

Up and Running with Red Hat Identity Manager

- Add the user on the replica

```
[root@vm2 ~]# kinit admin
Password for admin@LOCAL.NET: changeme
[root@vm2 ~]# ipa user-add test
First name: Jim
Last name: Wildman
----------------
Added user "test"
----------------
  User login: test
  First name: Jim
  Last name: Wildman
  Full name: Jim Wildman
  Display name: Jim Wildman
  Initials: JW
  Home directory: /home/test
  GECOS: Jim Wildman
  Login shell: /bin/sh
  Kerberos principal: test@LOCAL.NET
  Email address: test@local.net
  UID: 1196500500
  GID: 1196500500
  Password: False
  Member of groups: ipausers
  Kerberos keys available: False
```

- Check on vm1 again

```
[root@vm1 ~]# ipa user-find test
--------------
1 user matched
--------------
  User login: test
  First name: Jim
  Last name: Wildman
  Home directory: /home/test
  Login shell: /bin/sh
  Email address: test@local.net
  UID: 1196500500
  GID: 1196500500
  Account disabled: False
  Password: False
```

```
  Kerberos keys available: False
----------------------------
Number of entries returned 1
----------------------------
```

.

Lab 7: Configure automount

Automount allows a single home directory to be shared across all machines in the environment. The shared directory is automatically mounted to a machine when the user logs in. IdM can be used to store the configurations. IdM provides Kerberized tokens to enable secure mounting of NFSv4 home directories. For automount to work correctly with IdM, the machine providing the home directories must be a member of the IdM Kerberos realm. We will use vm1 to host the NFS shares.

Configure IdM for Automount

- On vm1, add all 3 client vm's as a as service principals for nfs services

```
[root@vm1 ~]# ipa service-add nfs/vm3.local.net
-----------------------------------------
Added service "nfs/vm1.local.net@LOCAL.NET"
-----------------------------------------
  Principal: nfs/v3.local.net@LOCAL.NET
  Managed by: vm3.local.net
```

(repeat for vm4 and 5)

- We're going to keep it simple and just add the default map

```
[root@vm1 ~]# ipa automountmap-add default auto.home
------------------------------
Added automount map "auto.home"
------------------------------
  Map: auto.home
```

- Add auto.home to auto.master so it knows to mount the home directories

```
[root@vm1 ~]# ipa automountkey-add default --key "/home" --info
auto.home auto.master
--------------------------
```

```
Added automount key "/home"
--------------------------
  Key: /home
  Mount information: auto.home
```

- Add the key for the home directory so the nfs mount will be performed using the kerberos key

```
[root@vm1 ~]# ipa automountkey-add default --key "*" --info
"-fstype=nfs4,rw,sec=krb5,soft,rsize=8192,wsize=8192
vm3.local.net:/exports/home/&" auto.home
-----------------------
Added automount key "*"
-----------------------
  Key: *
  Mount information:
-fstype=nfs4,rw,sec=krb5,soft,rsize=8192,wsize=8192
vm1.local.net:/exports/home/&
```

Configure vm3 as an nfs server

- Login to vm1 and get a kerberos ticket for admin

```
[root@vm3 ~]# kinit admin
Password for admin@LOCAL.NET:
```

- Get the keytab from the idm server

```
[root@vm3]# ipa-getkeytab -s vm1.local.net -p nfs/vm3.local.net -k
/etc/krb5.keytab
Keytab successfully retrieved and stored in: /etc/krb5.keytab
```

- Configure NFS for Kerberos

```
[root@vm3]# echo "SECURE_NFS=yes" >> /etc/sysconfig/nfs
```

- Create our home directories, configure the exports and start nfs

```
[root@vm3 ~]# mkdir -p /exports/home
[root@vm3 ~]# for i in 1 2 3 4
do mkdir /exports/home/classuser$i
chown 1000$i.1000$i  /exports/home/classuser$i
```

```
touch /exports/home/classuser$i/classuser$i.txt
done
[root@vm3 ~]# ll /exports/home
total 0
drwxr-xr-x. 2 classuser1 classuser1 6 Jun 11 22:57 classuser1
drwxr-xr-x. 2 classuser2 classuser2 6 Jun 11 22:57 classuser2
drwxr-xr-x. 2 classuser3 classuser3 6 Jun 11 22:57 classuser3
drwxr-xr-x. 2 classuser4 classuser4 6 Jun 11 22:57 classuser4
[root@vm3 ~]# echo "/exports/home  *(rw,sec=sys:krb5:krb5i:krb5p)"
>> /etc/exports
[root@vm3 ~]# firewall-cmd --permanent --add-service=nfs;
firewall-cmd --reload
success
success
[root@vm3 ~]# firewall-cmd --list-all
public (default, active)
  interfaces: eth0
  sources:
  services: dhcpv6-client nfs ssh
  ports:
  masquerade: no
  forward-ports:
  icmp-blocks:
  rich rules:
[root@vm3 ~]# systemctl enable nfs-server
Created symlink from
/etc/systemd/system/multi-user.target.wants/nfs-server.service to
/usr/lib/systemd/system/nfs-server.service.
[root@vm3 ~]# systemctl restart nfs-server
[root@vm3 ~]# showmount -e vm3.local.net
Export list for vm3.local.net:
/exports/home *
```

Configure the clients for automount

- Login to each of the clients, get a kerberos ticket as admin and configure nfs and automount

```
[root@vm4 ~]# kinit admin
Password for admin@LOCAL.NET:
[root@vm4 ~]# ipa-getkeytab -s vm1.local.net -p nfs/vm3.local.net
-k /etc/krb5.keytab
Keytab successfully retrieved and stored in: /etc/krb5.keytab
```

Up and Running with Red Hat Identity Manager

```
[root@vm2 ~]# ipa-client-automount --location=default
Searching for IPA server...
IPA server: DNS discovery
Location: default
Continue to configure the system with these values? [no]: yes
Configured /etc/sysconfig/nfs
Configured /etc/idmapd.conf
Started rpcidmapd
Started rpcgssd
Restarting sssd, waiting for it to become available.
Started autofs
```

- Login to one of the clients you have configured and you should have a home directory that you can edit


```
[jwildman@jwildman ~]$ ssh classuser2@192.168.122.14
-sh-4.2$ hostname
vm2.local.net
-sh-4.2$ whoami
classuser2
-sh-4.2$ df
Filesystem                                1K-blocks    Used
Available Use% Mounted on
/dev/mapper/rhel_vm2-root                   3205120 1030148
2174972  33% /
devtmpfs                                     497996       0
497996   0% /dev
tmpfs                                        508456       0
508456   0% /dev/shm
tmpfs                                        508456    6756
501700   2% /run
tmpfs                                        508456       0
508456   0% /sys/fs/cgroup
/dev/vda1                                    508588  127184
381404  26% /boot
vm1.local.net:/exports/home/classuser2      3205120 1038928
2166192  33% /home/classuser2
tmpfs                                        101692       0
101692   0% /run/user/10001
-sh-4.2$ pwd
/home/classuser1
```