BOSTON, MA
JUNE 23-26, 2015

Integrating the RHCI Suite with IdM

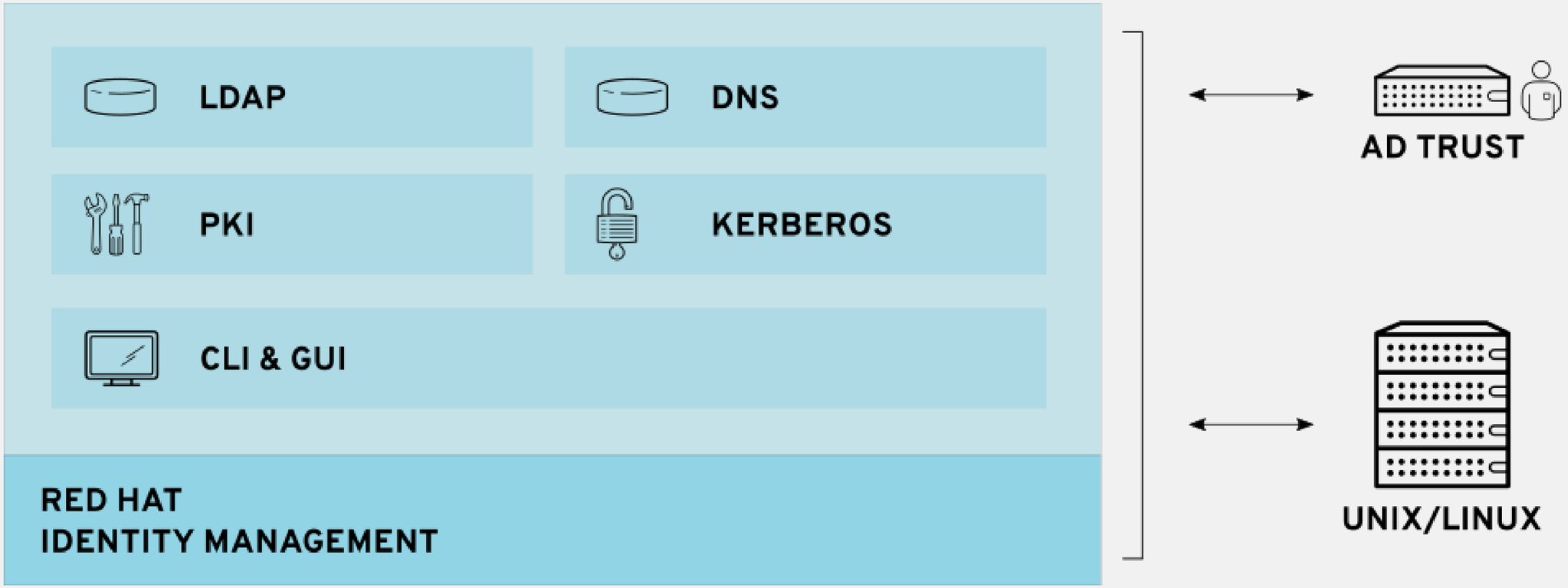
INTRODUCTION

Who are we?

Chris Keller
Solutions Architect
Red Hat, Inc.

Nathan Kinder
Engineering Manager
Red Hat, Inc.

What is IdM?



IdM Features

- Numerous Capabilities
  - Identity management for users and machines
  - HBAC
  - 2FA (OTP)
  - Centralized sudo rules management
  - Other services including DNS, NTP

What is RHCI?

- Collection of products
  - Red Hat Enterprise Virtualization
  - Red Hat CloudForms
  - Red Hat Satellite
  - Red Hat Enterprise Linux Open Stack Platform
- Lets you build a private Infrastructure-as-a-Service (IaaS) based cloud for traditional workloads
- On-ramp to a highly scalable public-cloud-like infrastructure
- Built on Red Hat Enterprise Linux

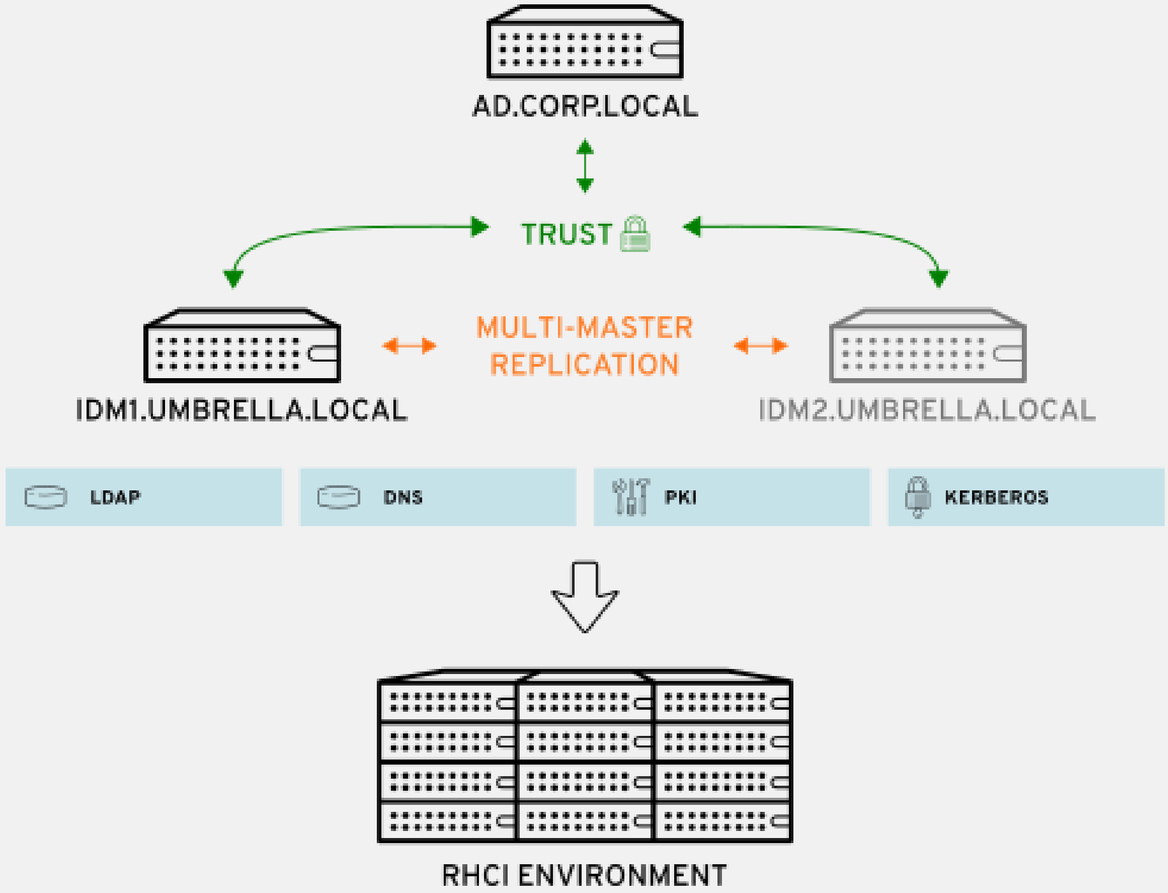
What are we integrating?

- RHEL
- Application User Interfaces
- Specific application functionality
  - Satellite system lifecycle in IdM
  - Application quotas
- Mapping application roles to groups
  - Common roles between products?
  - Overlap groups as much as possible (i.e. Administrators)

Why Centralized Authentication?

- Security!
- Most products have their own local user and group store
  - Managing multiple sets of users is difficult!
    - Who has access to what (difficult to audit)?
- IdM Provides centralized user and group management
  - Leverage AD users and groups
  - Align groups to roles in each application
  - Configure role/group assignment once

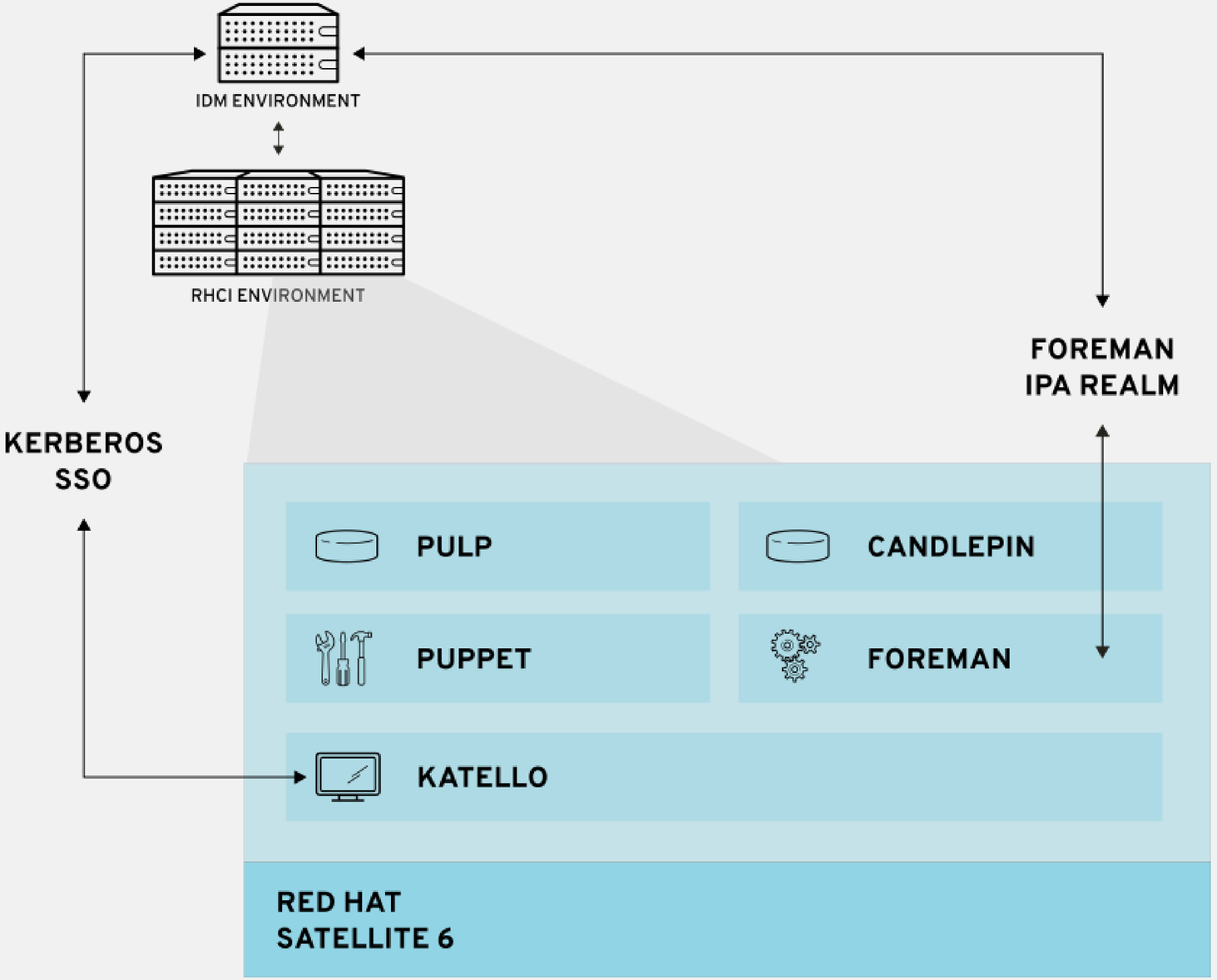
IdM Example Architecture



IdM Reference Environment

- RHEL 7.1 (Satellite, RHEL OSP)
- RHEL 6.6 (RHEV Manager, CloudForms Appliance)
- Provide for HA (multi-master replication)
- Integrate with Active Directory
- Will scale with your environment
  - Number of data centers
  - Number of hosts
- Developers and RHCI Administrators group

INTEGRATING RED HAT SATELLITE


IDM ENVIRONMENT
RHCI ENVIRONMENT
KERBEROS
SSO
FOREMAN
IPA REALM
PULP
CANDLEPIN
PUPPET
FOREMAN
KATELLO
RED HAT
SATELLITE 6

Satellite Integration

- Satellite Server
- Satellite UI
- System life-cycle management in IdM
- Users
- Groups
- Roles
  - Administrators

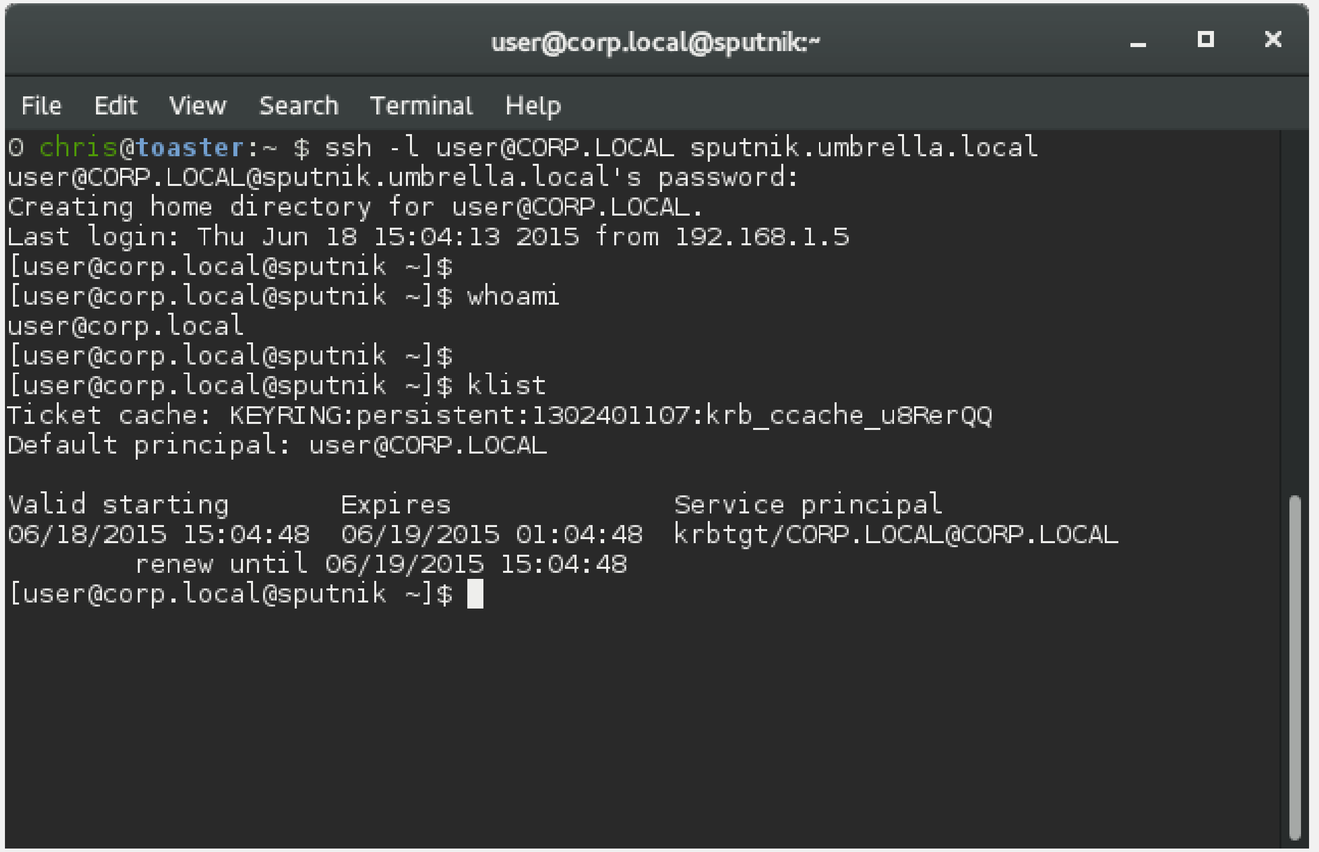
Configuring RHEL

- Install Relevant Packages
  - ipa-client, foreman-proxy, ipa-admintools
- Connect system to IdM

```
# ipa-client-install
```

  - (optional: --mkhomedir, etc)

Access over SSH



Configuring UI

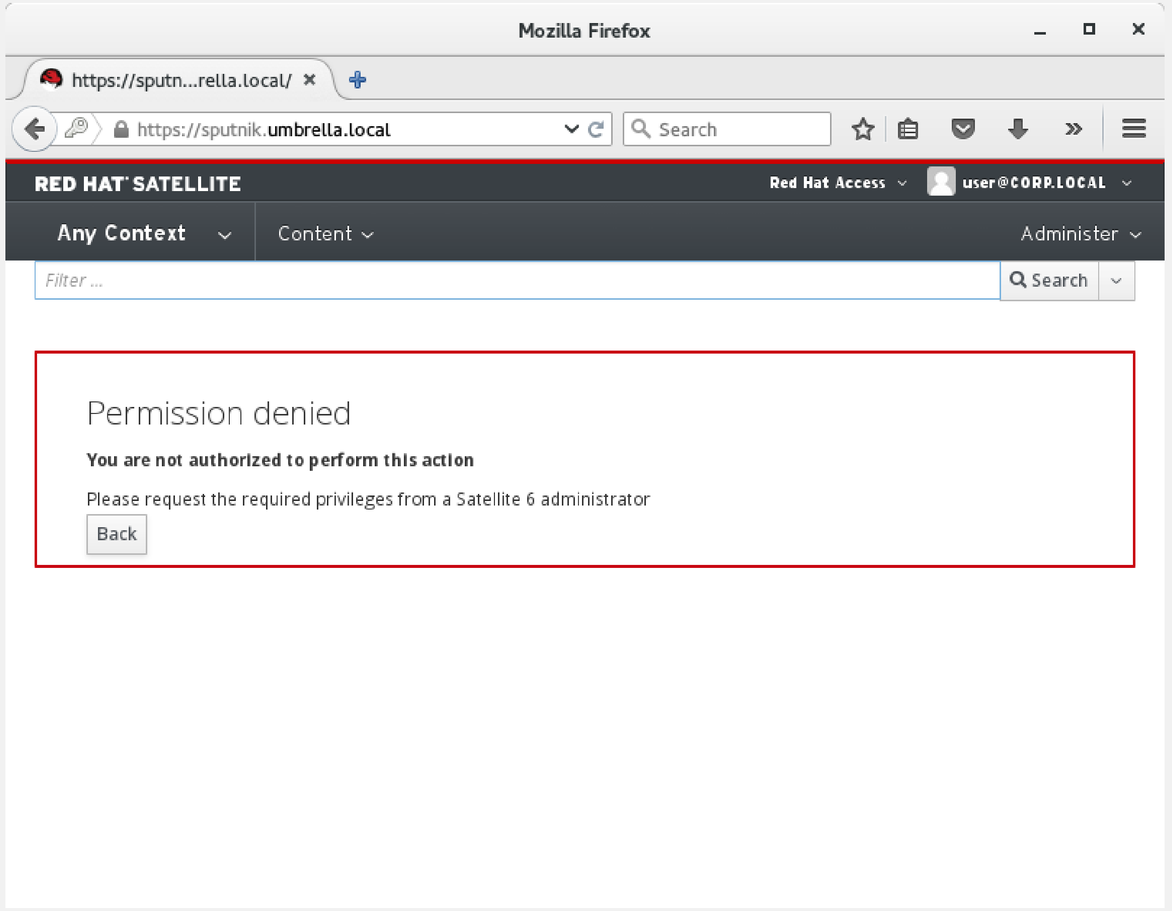
- Kerberos SSO
- Create Service Principal for Apache

```
# kinit admin
# ipa service-add HTTP/sputnik.umbrella.local@UMBRELLA.LOCAL
```

- Configure Foreman

```
# katello-installer --foreman-ipa-authentication=true
```

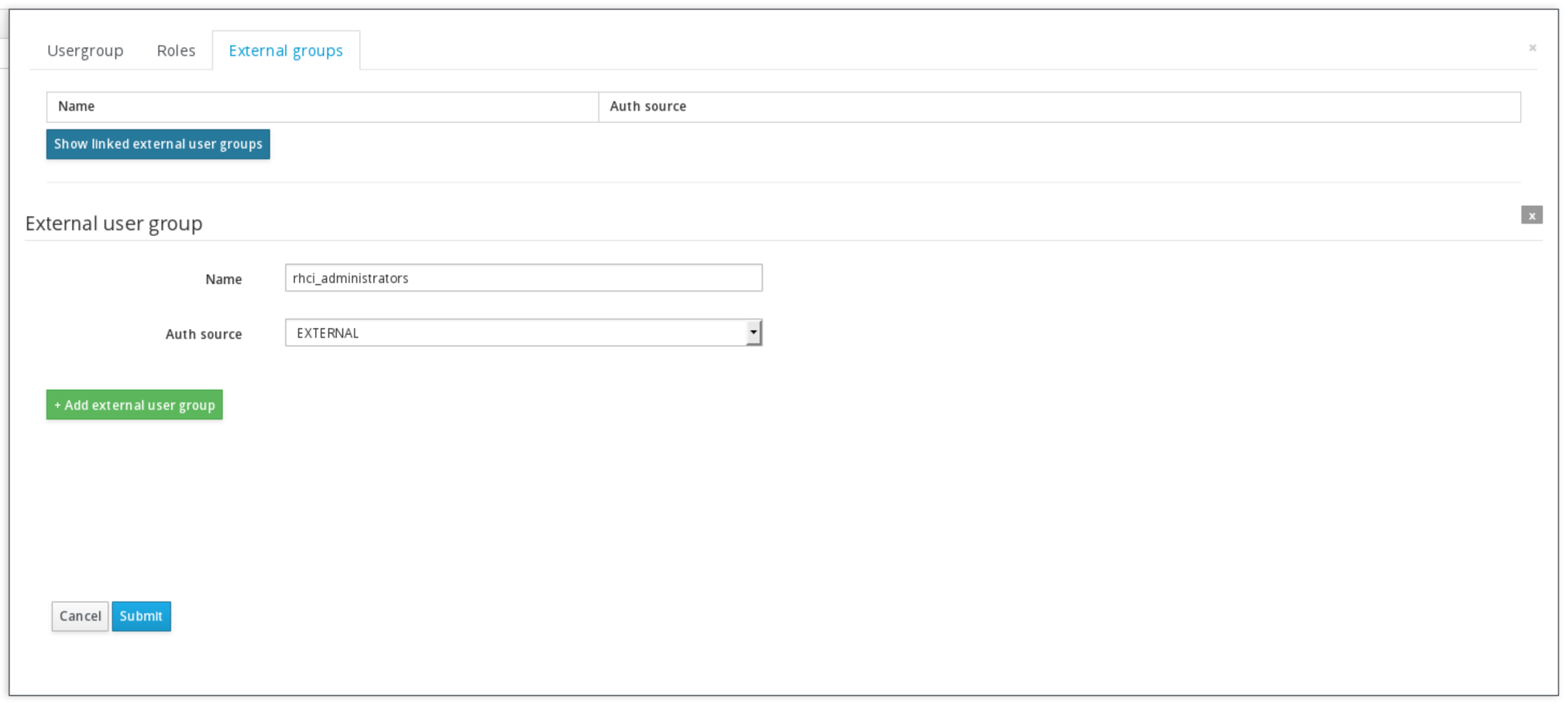
What happens after logging in?



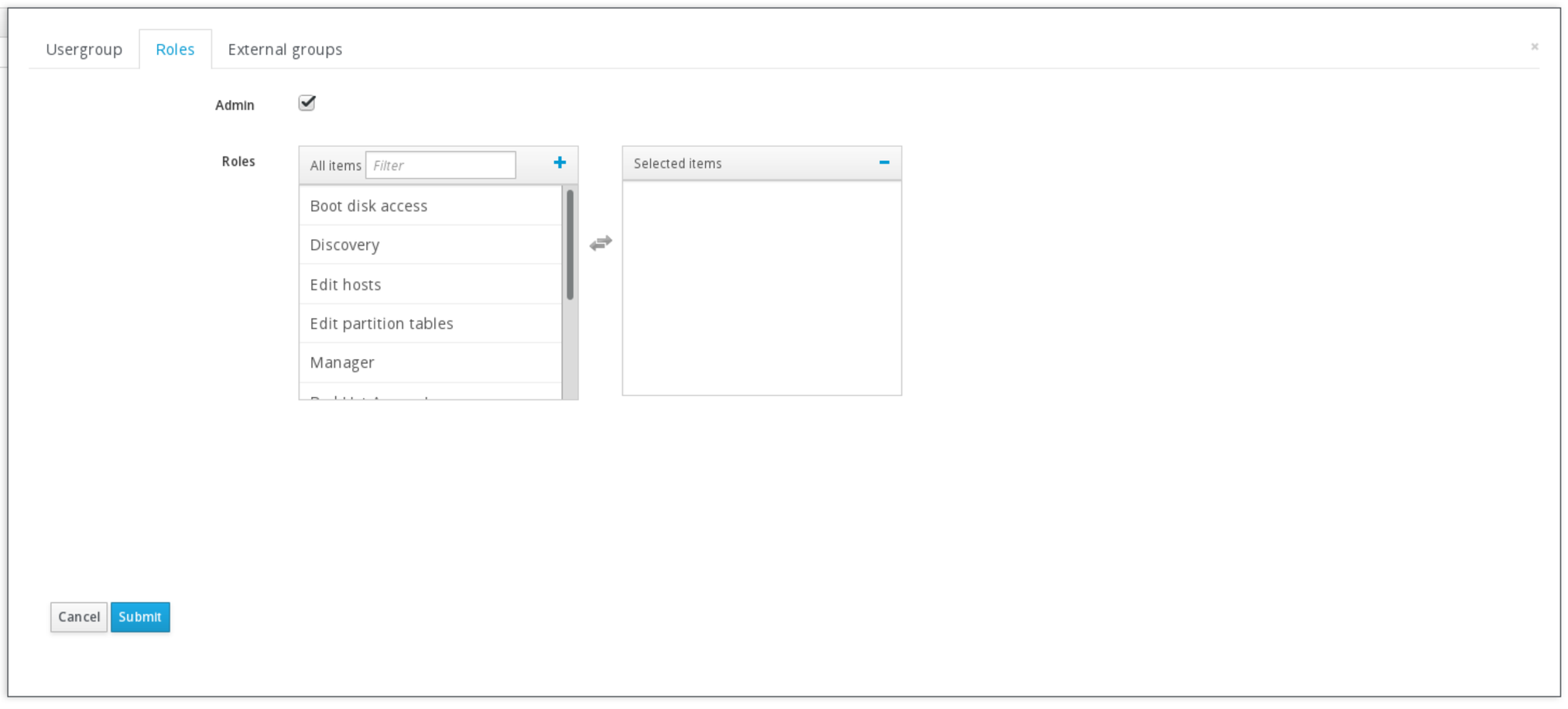
Assigning Groups to Roles

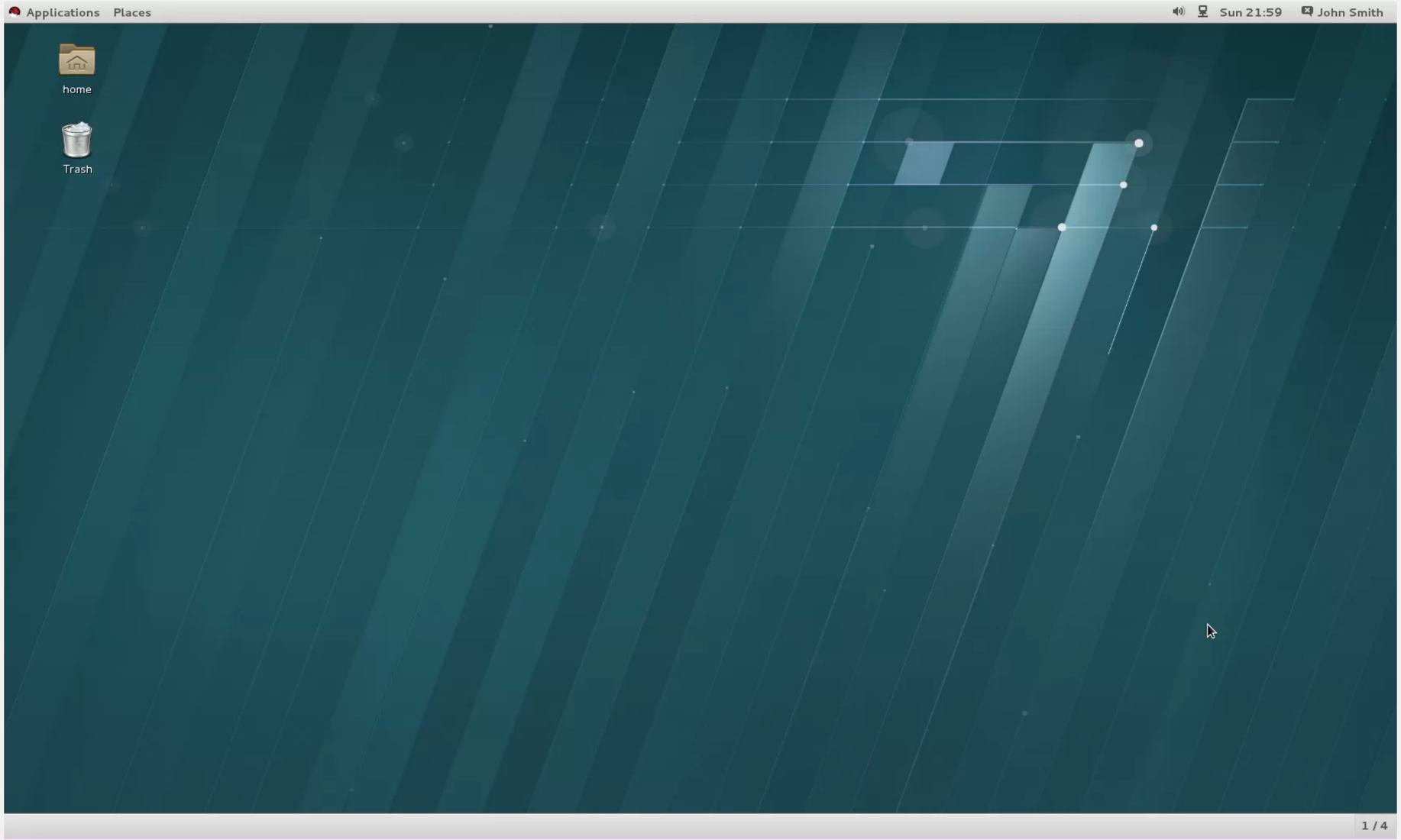
- Administrative access by group?
- Create a new User Group that includes an external user group from IdM
- rhci_administrators in Satellite which sources rhci_administrators in IdM
  - Assign this group the Admin role

Assign External Group



Assign This Group a Role




Applications
Places
Sun 21:59
John Smith
home
Trash
1 / 4

Enabling IdM Realm Support in Foreman

- Foreman can manage the lifecycle of hosts in IdM
- Can configure a series of realms (e.g. UMBRELLA.LOCAL) that can be associated with a host when initially provisioned
- IdM generates single-use password
  - Foreman embeds password in provisioning template
- Systems can be automatically enrolled in Host Groups
  - HBAC based on group membership
  - Self-service users have access to resources immediately

Realm Configuration

- Configure IdM to work with a Foreman Smart Proxy
  - Creates dedicated IdM role with appropriate permissions
  - Creates a user and retrieves keytab

```
# foreman-prepare-realm admin realm-capsule
```

Realm Configuration Continued

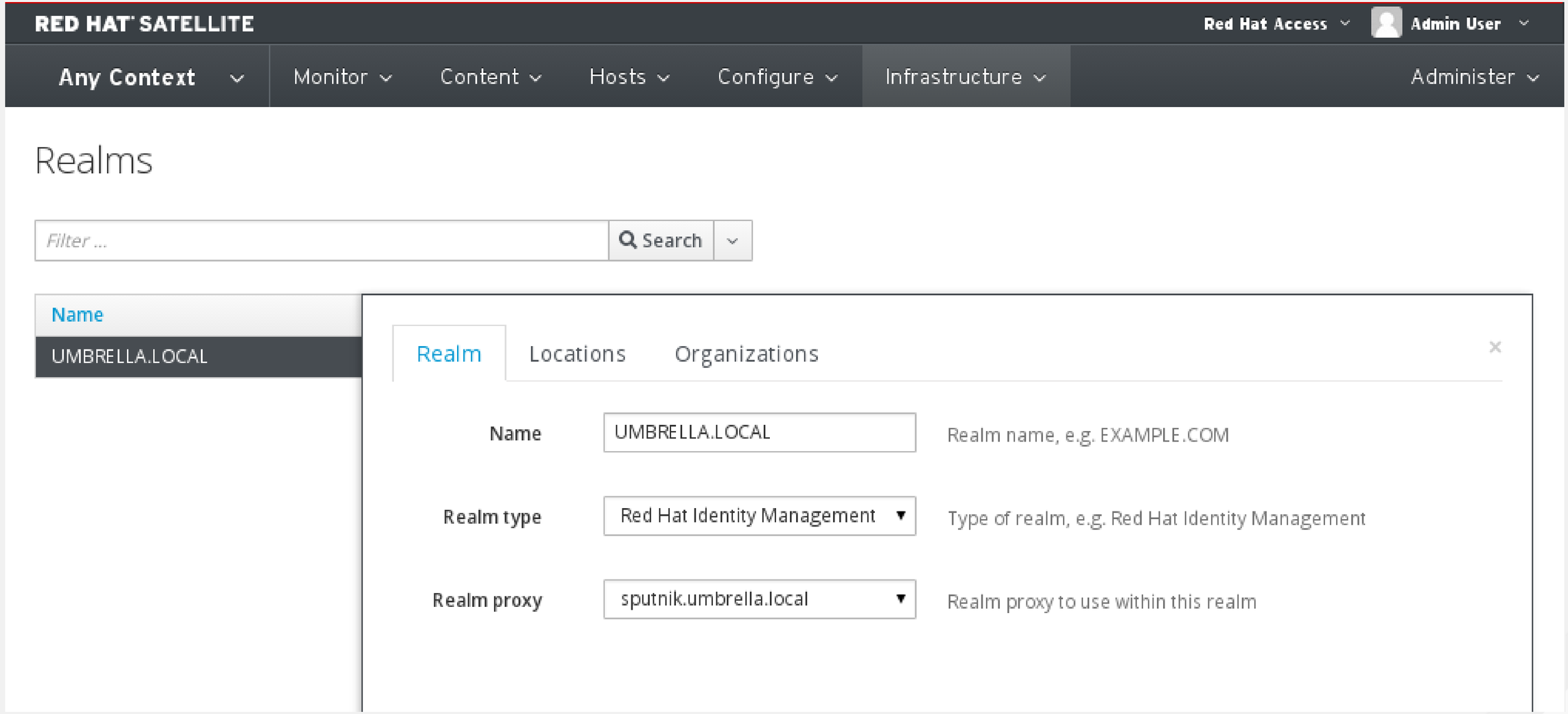
- Configure the realm in Katello

```
# katello-installer --capsule-realm true
    --capsule-realm-keytab /etc/foreman-proxy/freeipa.keytab
    --capsule-realm-principal 'realm-capsule@UMBRELLA.LOCAL'
    --capsule-realm-provider freeipa
```

- /etc/foreman-proxy/freeipa.keytab was created via foreman-prepare-realm command
- Restart the foreman-proxy service

```
# systemctl restart foreman-proxy.service
```

Creating IdM Realm in Satellite



Assigning Systems to Host Groups

- Setup automatic membership rules based on a system's attributes
- When a system joins a Satellite Host Group, the system is joined to corresponding IdM Host Group as well
  - Allows for HBAC, sudo policies, etc.
  - Foreman Host Group is available as a parameter in IdM known as userclass
- In IdM, setup an automembership rule

```
# ipa automember-add --type=hostgroup app_servers
```

- Define an automembership condition based on the userclass attribute

```
# ipa automember-add-condition –-key=userclass
  --type=hostgroup --inclusive-regex=^app_server app_servers
```

- Note: automember rules only applied during initial add

INTEGRATING RED HAT ENTERPRISE VIRTUALIZATION

RHEV Integration

- RHEV-M Server
- RHEL Hypervisors
- RHEV UI
- Users
- Groups
  - Quota Management
- Roles
  - Administrator
  - User (Provisioning)

Configuring RHEV-M Server

- Install Relevant Packages
  - ipa-client
- Connect system to IdM

```
# ipa-client-install
```

  - (optional: --mkhomedir, etc)

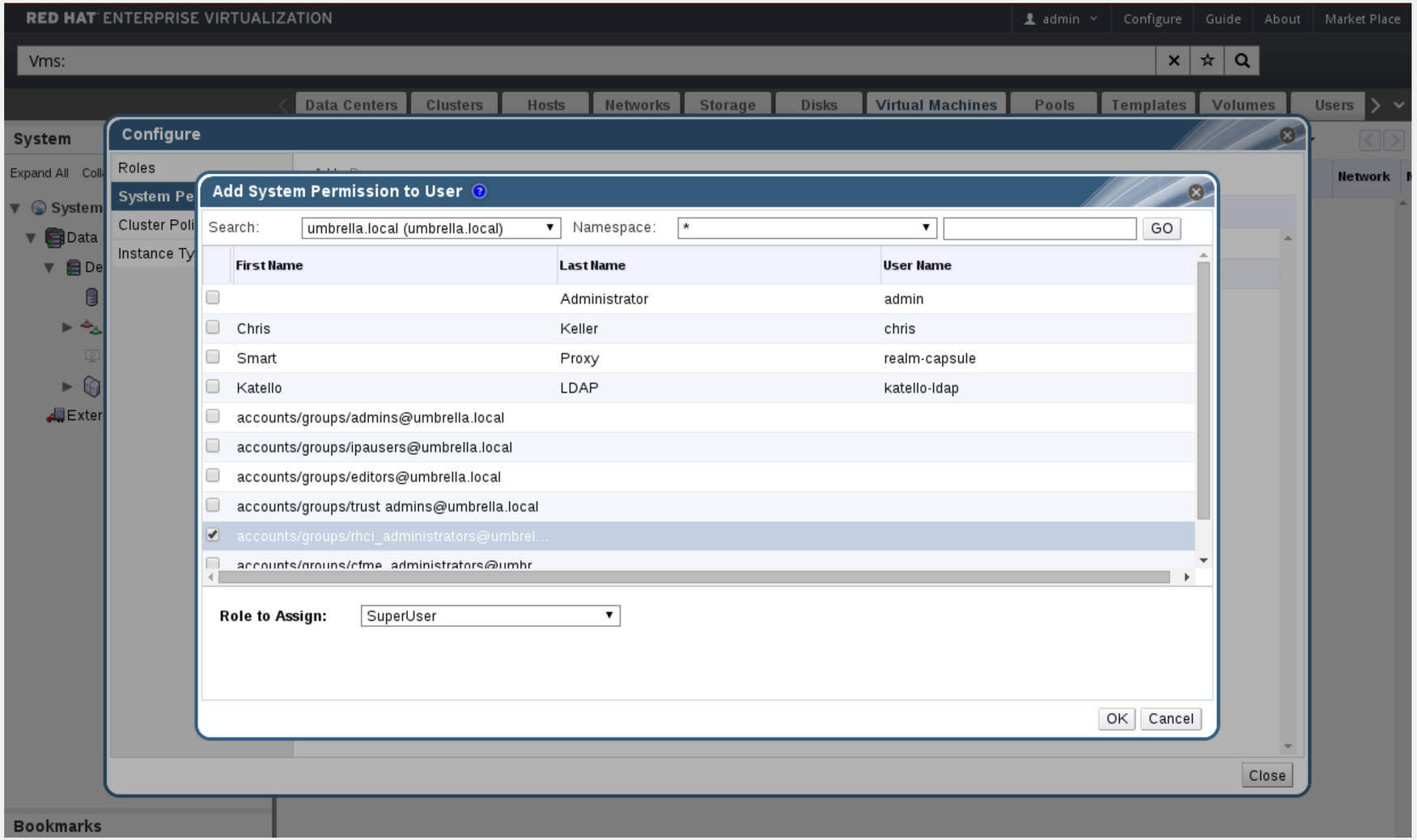
Configuring RHEV Manager

- Configure engine to use IPA

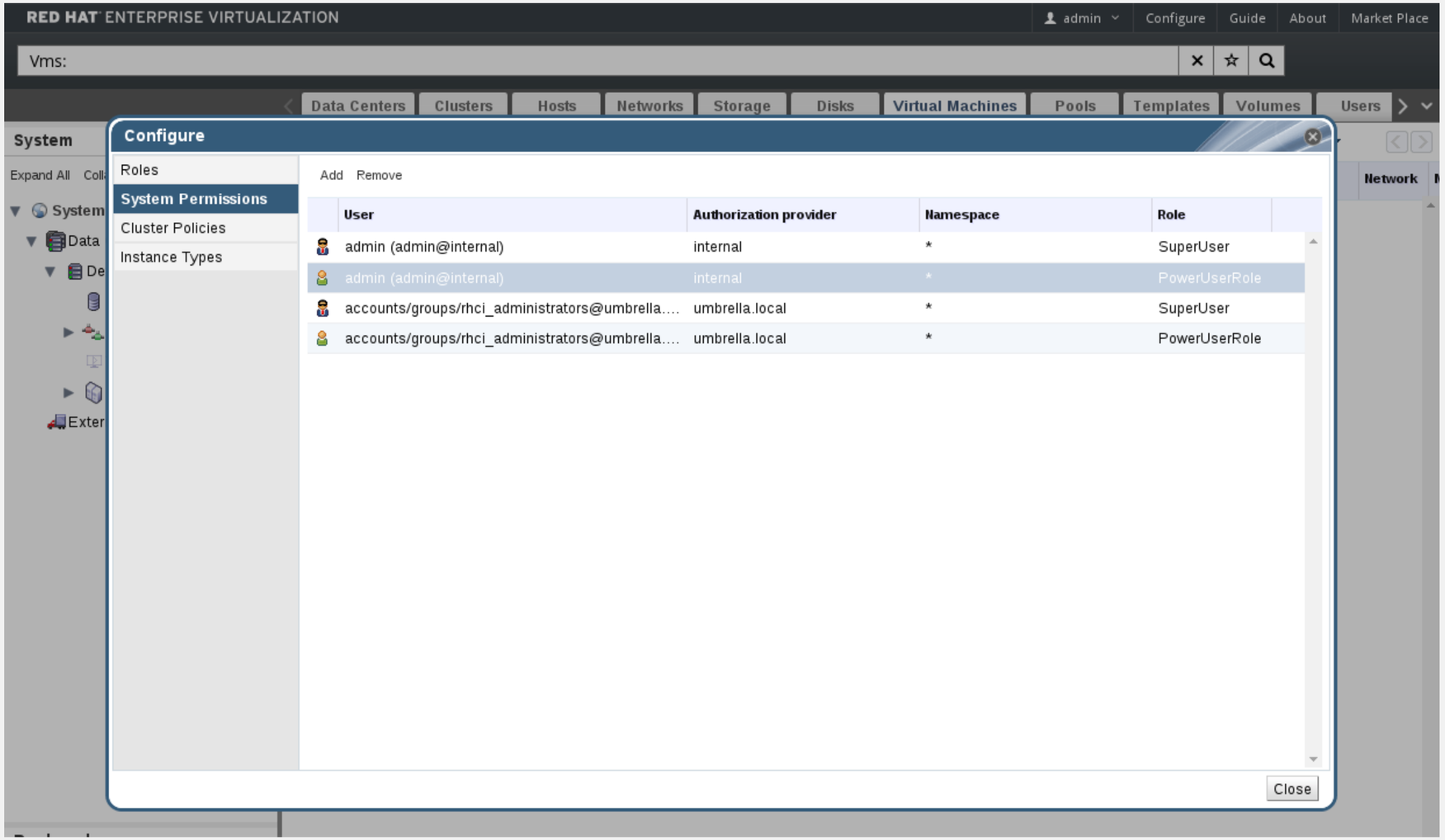
```
# engine-manage-domains add --domain=UMBRELLA.LOCAL
  --provider=IPA --user=admin
```

- Engine is now configured to use external users and groups
  - Need to align users/groups to roles

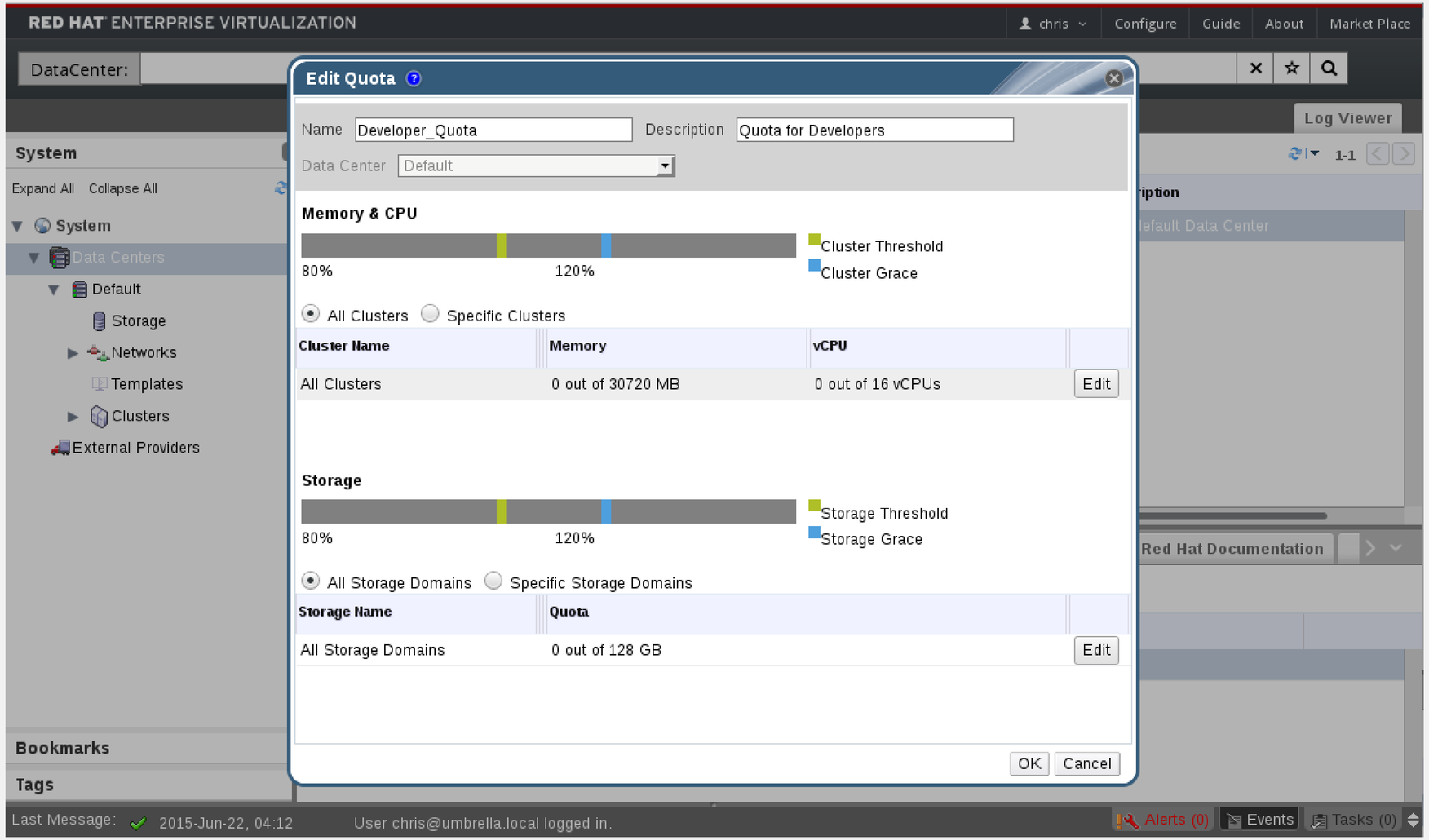
Linking Users/Groups to Roles



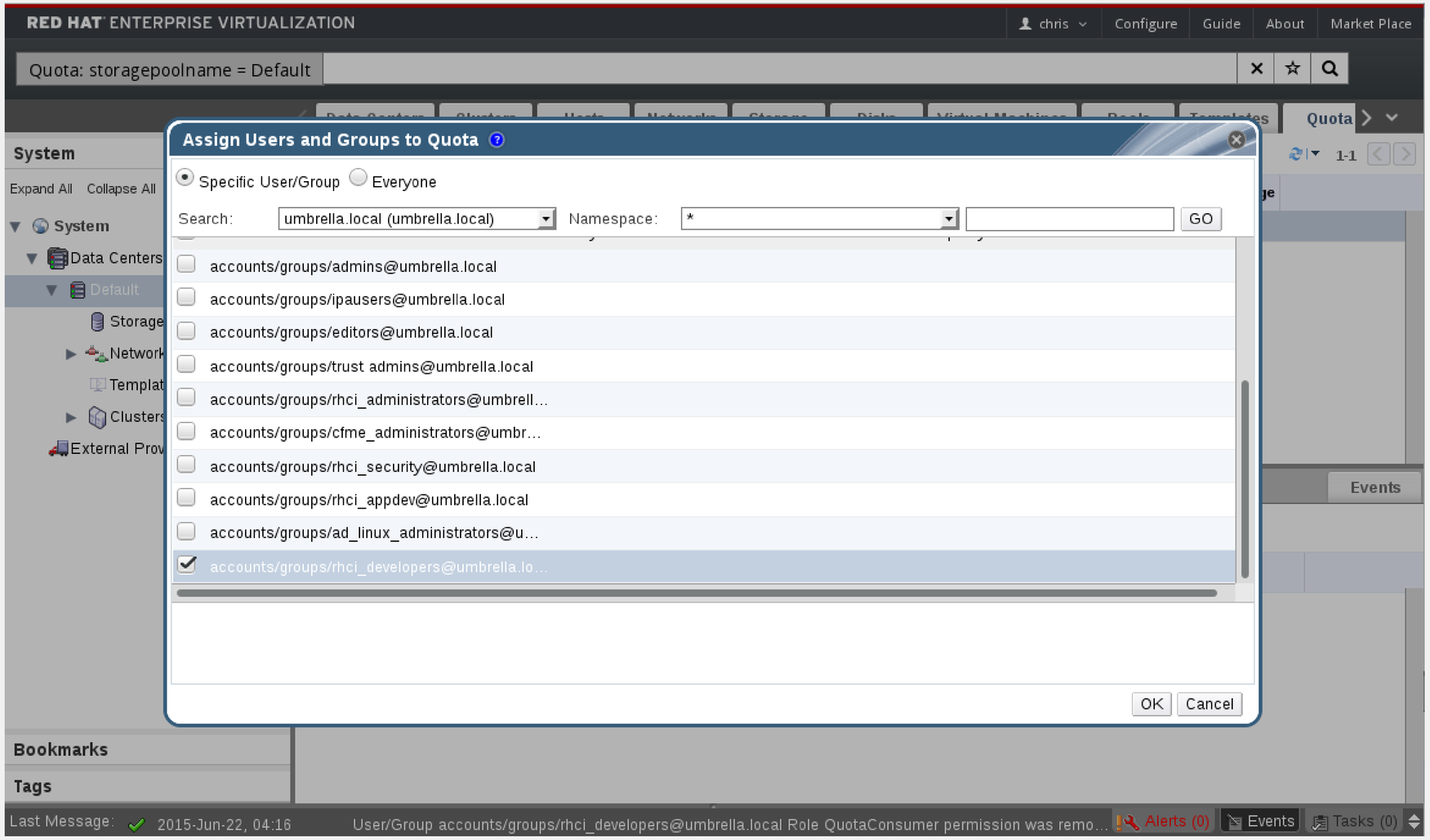
Administrator Role Configured



Creating a Developer Quota



Assigning a Quota



INTEGRATING RED HAT CLOUDFORMS

CloudForms Integration

- CloudForms Appliance
- CloudForms UI
- Users
- Groups
- Roles
  - Administrator
  - User (Provisioning)

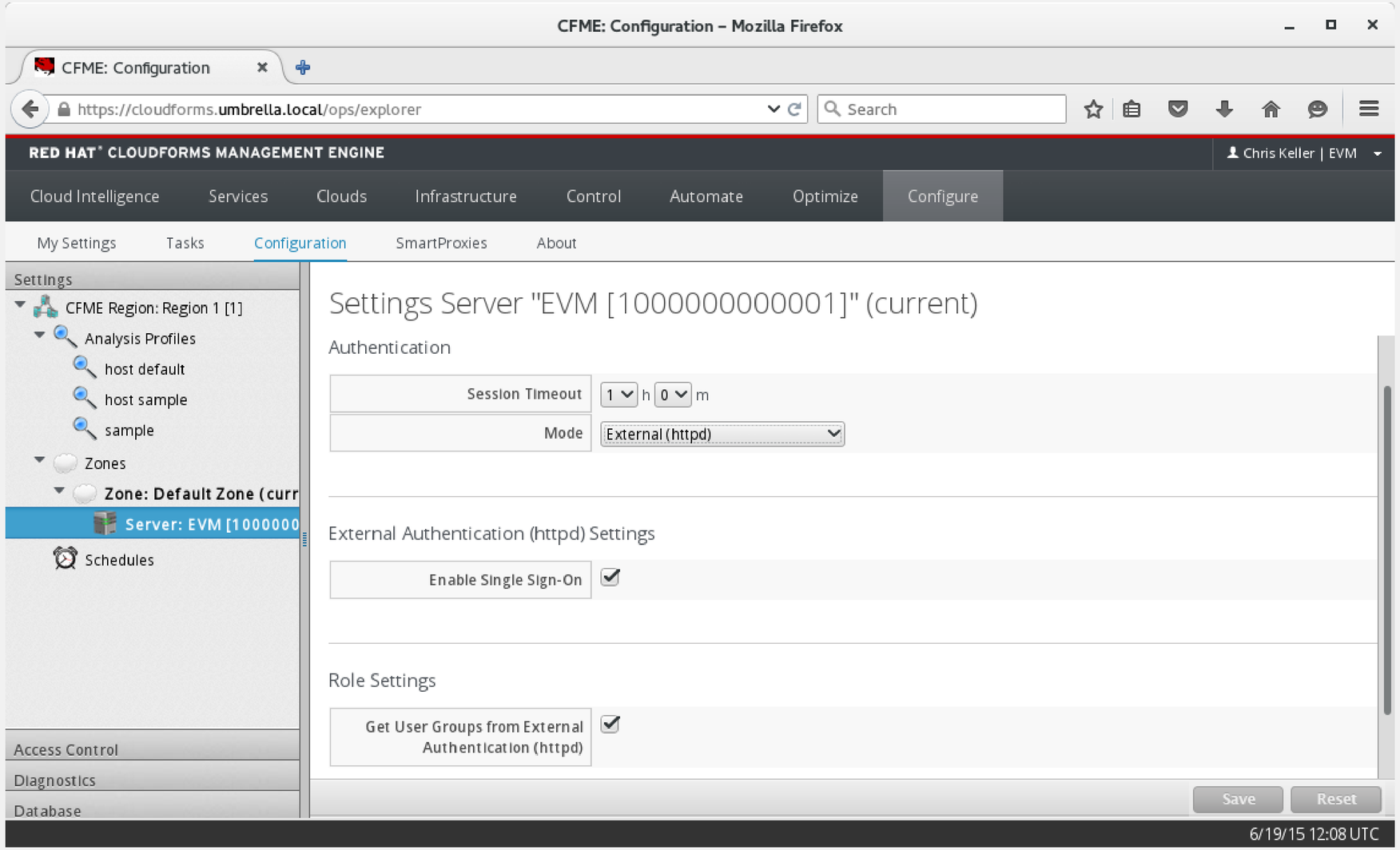
Configuring the Appliance

- No need to Install Relevant Packages
  - ipa-client is already installed on the appliance
- Connect system to IdM & configure external auth

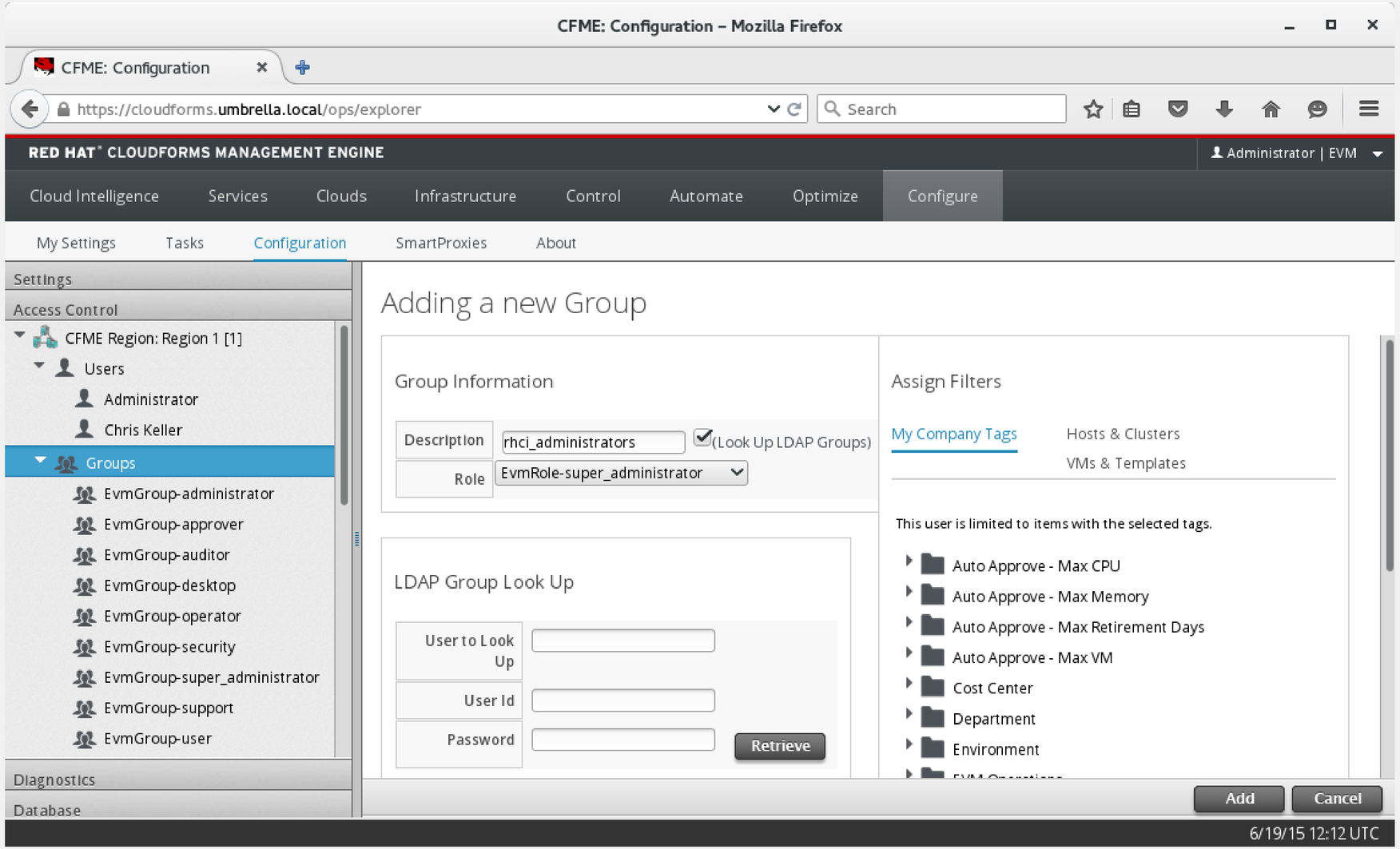
```
# /bin/appliance_console_cli --host cloudforms.umbrella.local
--ipaserver idm1.umbrella.local --iparealm UMBRELLA.LOCAL
--ipaprincipal admin --ipapassword <secret>
```

- What just happened?
  - ipa-client-install
  - SSSD/PAM configuration
  - Apache configuration updated
  - SELinux Booleans

Configure CloudForms



Adding LDAP Backed Groups w/ Roles



Configuring Quota

- CloudForms uses the notion of tagging
  - Virtual machines, physical assets, accounts, etc
  - Tags can be manually assigned or dynamically created
- Quotas work based off tags
  - Tags can be assigned based off group membership

Quota Example

Editing My Company Tags for "EVM Groups"

Tag Assignment

Select a customer tag to assign: Cost Center * ▼ <Select a value to assign> ▼

	Category	Assigned Value
	Auto Approve - Max Memory *	8GB
	Auto Approve - Max VM *	2
	Line of Business	Application Developers

* Only a single value can be assigned from these categories

1 EVM Group Being Tagged

	Name	Read Only	Number of Users	Role	Sequence
	rhci_developers	False	0	EvmRole-user_self_service	14

INTEGRATING RHEL OPEN STACK PLATFORM

Keystone

- Keystone focal point for identity in OpenStack
  - Used by all OpenStack for authentication, authorization, service catalogs, etc
- Supports a variety of identity providers
  - SQL (Keystone acts as identity provider)
  - LDAP
  - External
- Keystone best suited for authorization, not necessarily authentication

Basic SQL Provider

- Leverages SQL database for identity
- User entry is stored in database that contains password hash
- Data is sent via clear text
- Password based authentication services (i.e. LDAP) have additional security capabilities
  - Dictionary checking
  - Password change intervals
  - Password history
  - Account lockouts

LDAP Provider

- Keystone only supports simple BIND operations
- Works just like the SQL authentication source mentioned previously
- LDAP supports strong authentication via SASL
  - Keystone does not support SASL bind operations
- Offloads user provisioning and maintenance
  - Allows for centralized identity source that can be shared with other applications

External Provider

- Allows for stronger form of authentication vs. simple password based authentication
- Keystone expects the web server to handle authentication
  - Can utilize a host of Apache authentication modules
  - Apache supplies keystone with authenticated user name via REMOTE_USER environment variable
- User still stored in Keystone (or LDAP store) but no password credentials
- Obvious benefit for security

Federation Extension

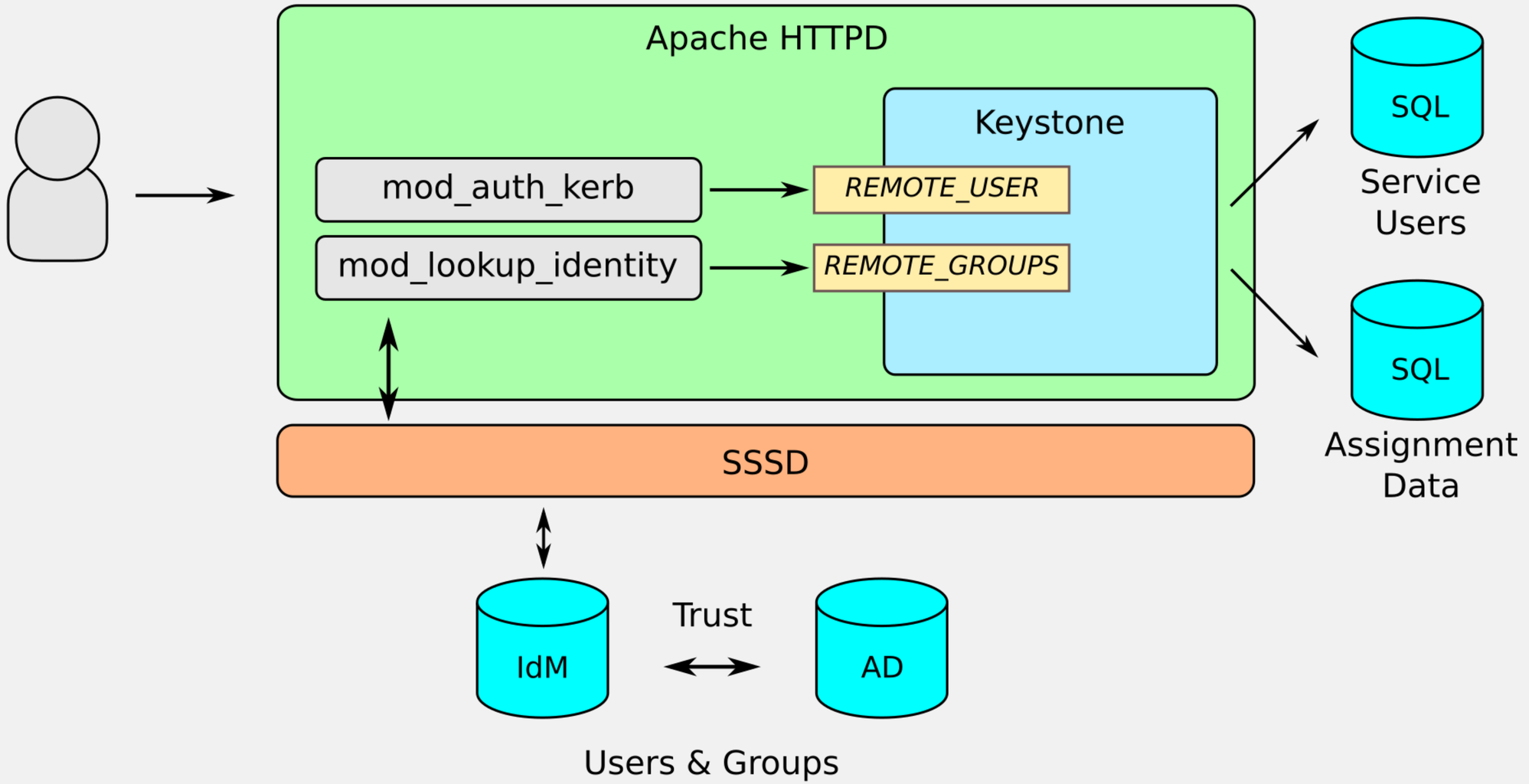
- Simpler for Keystone
- No need for LDAP schema extensions or LDAP connection management
- Have Apache provide pertinent information on authenticated user along with token request
  - Keystone can then map user info to applicable project and roles
- How does this work?

Integrating with IdM

- mod_identity_lookup – Helps to eliminate the need for identity lookup in Keystone
- Utilizes SSSD from underlying platform to provide user and group information
  - Information can be source from various providers
    - **IdM**, LDAP and/or Active Directory
- SSSD provides additional capabilities that Keystone does not
  - Credential and attribute caching
  - Connection pooling
  - Multiple identity sources
- Allows for a more scalable and performant Keystone service

SSSD & Cross Realm Trusts

- Leverage AD accounts via cross realm trusts in IdM
  - Users can use their TGT from AD to fetch Kerberos enabled services that are setup in IdM, such as Keystone and Horizon
- OpenStack specific groups defined locally in IdM
  - SSSD is able to extract group information from PAC
  - Matching external groups in IdM setup to match AD
- Multiple trusts allow users from multiple forests to leverage the same Keystone server


Apache HTTPD
Keystone
mod_auth_kerb
mod_lookup_identity
REMOTE_USER
REMOTE_GROUPS
SQL
Service Users
SQL
Assignment Data
SSSD
IdM
Trust
AD
Users & Groups

Thank You!

- Chris Keller
  ckeller@redhat.com
- Nathan Kinder

  nkinder@redhat.com

LEARN. NETWORK.
EXPERIENCE OPEN SOURCE.