Dynamic Roles in CloudStack

Boris Stoyanov
Software Development Engineer in Test
boris.stoyanov@shapeblue.com
twitter: @shapeblue
About Me

- **Break Stuff @ ShapeBlue**
- **Background:**
  - More than 10 years in Software Development and Testing
  - Specialize in:
    - Test Management
    - Automated Testing
    - Testing Frameworks
- Joined ShapeBlue and CloudStack last year
"ShapeBlue are expert builders of public & private clouds. They are the leading global CloudStack services company."
Dynamic Roles in CloudStack
Static Roles in CloudStack

• List of pre-defined roles
• All roles permissions are kept in a single file commands.properties
• Each change requires a management server restart
• How do we add a custom role with new set of permissions
Dynamic Roles

Quiz Time
Q1: What are these numbers and what’s their purpose: 1, 2, 4, 8

Answer:
These numbers represent the static roles
1 = ADMIN
2 = RESOURCE_DOMAIN_ADMIN
4 = DOMAIN_ADMIN
8 = USER
commands.properties

### Use system vm commands
createUser=startSystemVm=1
deleteUser=startSystemVm=1
updateUser=rebootSystemVm=1
listUsers=15 stopSystemVm=1
lockUser=15 destroySystemVm=1
disableUser=migrateSystemVm=3
extUser=migrateSystemVm=3

### VM commands
deployVirtualMachine=15
destroyVirtualMachine=15
rebootVirtualMachine=15

### Account commands
createAccount=listVirtualMachines=15

### CloudStack authentication commands
login=15
logout=15

### SAML SSO/SLO commands
listSamls=15
listSamlAuthorization=7
listIdps=15
authorizeSamlSso=7
listSamlAuthorization=7
listAndSwitchSamlAccount=15
Q2: What are the 7s and 15s?

Answer: all users until that number can execute the command
Q3: What does this number represent: 790

Answer: That’s about the number of lines commands.properties has in 4.9.
Static Role-based Access Control

- Pre-defined roles
- All permissions kept in a commands.properties file
- Changes are difficult to maintain
- Management server restart is required after change
- Hard to add a new role with custom permissions
Add Read-only Admin

- Root Admin
- Read-only permission
Let’s re-thing roles management

• New way of managing roles

• Add/Change roles made easy

• Apply changes without management restart
Here’s what we did

• Move all permissions to the DB
• Create a dynamic role based account checker (RBAC)
• New UI interface
• Handle migrations
Dynamic ApiChecker

1. Issue API command
2. Is Root Admin present? (Yes/No)
   - Yes: Allow API command, Continue
   - No: Is Allow Rule present? (Yes/No)
     - Yes: Check Annotation (Yes/No)
       - Yes: Deny API call
       - No: Deny API call
     - No: Allow API command, Continue
How to use it: Adding role

Use case: Root Admin wants to create a root admin read-only account, who is not allowed to see Global Settings.

• Create a custom role
• Add an “allow rule” to all list APIs
• Add ”deny rule” to all configuration APIs
• Assign the role to the read-only account
### How to use it: Adding role

<table>
<thead>
<tr>
<th>Rule</th>
<th>Permission</th>
<th>Description</th>
<th>Action</th>
</tr>
</thead>
<tbody>
<tr>
<td>*Configurations</td>
<td>Deny</td>
<td></td>
<td></td>
</tr>
<tr>
<td>list*</td>
<td>Allow</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
How to use it: Good practices

• When adding custom rules, user is allowed to select multiple APIs using “*”
• Rules can be shifted in the list in set the order of the list
• It’s a good practice to move deny rules on top of the list when allowing multiple APIs at once.
How to use it: Denied API

• What happens in UI when user hits a denied API?

• User is displayed with the following error:

  ![Status]
  The user is not allowed to request the API command or the API command does not exist.
Dynamic Role-based Access Control

- Pre-defined roles are available
- Moves all permissions into the DB
- Adds UI interface to add a new role
- Custom set of rules per API for a role
- Does not require management restart
Live demo

- One must read slide title first
Availability and Upgrade

- Dynamic RBAC is available and enabled by default on all new installations post 4.9
- Users upgrading to >4.9.x will have the feature disabled post upgrade
- Migration tool is available to do the migration and enable Dynamic RBAC
Upgrade: Running the migration tool

[root@host]# python migrate-dynamicroles.py -u cloud -p cloud -h localhost -p 3306 -f /etc/cloudstack/management/commands.properties

Apache CloudStack Role Permission Migration Tool
(c) Apache CloudStack Authors and the ASF, under the Apache License, Version 2.0
Running this migration tool will remove any default-role permissions from cloud.role_permissions. Do you want to continue? [y/N]y
The commands.properties file has been deprecated and moved at: /etc/cloudstack/management/commands.properties.deprecated
Static role permissions from commands.properties have been migrated into the db
Dynamic role based API checker has been enabled!
Migrating Roles

- After enabling Dynamic RBAC root admin role permissions looks like this:

<table>
<thead>
<tr>
<th>Rule</th>
<th>Permission</th>
<th>Description</th>
<th>Action</th>
</tr>
</thead>
<tbody>
<tr>
<td>*</td>
<td>Allow</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Click to edit

The Cloud Specialists

ShapeBlue.com

@ShapeBlue
• While other roles have explicit rules created based on the settings in commands.properties file.
Questions?
By the way....

Next CloudStack event: Cloudstack Collaboration Conference at ApacheCon North America

May 16-18, 2017
InterContinental Miami
MIAMI, FLORIDA
United States

http://events.linuxfoundation.org/events/apachecon-north-america/attend/register-
More information

• Slide deck:  http://www.slideshare.net/shapeblue

• Blog:  http://shapeblue.com/blog

• Email:  boris.stoyanov@shapeblue.com

• Web:  http://shapeblue.com