iControl 101 - #08 - Partitions

Joe Pruitt, 2008-27-03

In a previous article, I discussed user management and the concepts of user roles. User roles form half of what we refer to as Administrative Domains. The second half of Administrative Domains are Configuration Partitions.   Configuration Partitions allow you to control access to BIG-IP system resources by BIG-IP administrative users. This article will discuss how to manage Configuration Partitions with the iControl programming interfaces.

Configuration partitions are essentially logical containers that you can use to group local traffic management objects. Think of it as different "buckets" you can place your virtuals, pools, rules, etc into. You can then control who can peek into those buckets by the user name and, for those lucky enough to be able to peek in, what they can do by their user roles.

There is always at least one partition. For new installs, or upgrades, a special partition called "Common" is created that is the default for all traffic management objects. You can create many partitions with zero or more users assigned access to them. The following examples will illustrate how to interact with the partitions. The samples use the iControl CmdLets for PowerShell so check out the PowerShell Labs Project if you want to test them out for yourself.

Initializing the iControl PowerShell SnapIn

For a new PowerShell instance, you will need to add the iControl SnapIn and initialize it for your BIG-IP.

```
PS> Add-PSSnapIn iControlSnapIn
PS> Initialize-F5.iControl -hostname bigip_address -username bigip_username -password bigip_password
PS> $ic = Get-F5.iControl
```

Querying Partitions

The Management::Partition::get_partition_list() method will return the list of all Configuration partitions along with the corresponding description attribute.

```
PS> $ic.ManagementPartition.get_list()
partition_name                                            description
--------------                                            -----------
Common                                                    Repository for system objects and shared
Accounting                                                Partition for Accounting Department
DevCentral                                                DevCentral administrative partition
Finance                                                   Partition for Finance Departmen
```

Creating Partitions

So, let's say you've got a new department that needs a separate configuration space on your BIG-IP. Creating a new Partition is as easy as this:

```
PS > $partition = New-Object -TypeName iControl.ManagementPartitionAuthZPartition
PS > $partition.partition_name = "NewDept"
PS > $partition.description = "New Department"
PS > $ic.ManagementPartition.create_partition( (,$partition) )
PS> $ic.ManagementPartition.get_list()
partition_name                                            description
--------------                                            -----------
```

```
Common                                                      Repository for system objects and shared
NewDept                                                     New Department
Accounting                                                  Partition for Accounting Department
DevCentral                                                  DevCentral administrative partition
Finance                                                     Partition for Finance Departmen
```

Deleting Partitions

Now, let's suppose that the NewDept group that insisted on a separate configuration space, has gone 2 years without adding a single object and they are now no longer anywhere to be seen. Well, it's very simple to delete that partition you created for them with the delete_partition() method.

```
PS> $ic.ManagementPartition.delete_partition( (,"NewDept") )
PS> $ic.ManagementPartition.get_list()
partition_name                                              description
--------------                                              -----------
Common                                                      Repository for system objects and shared
DevCentral                                                  DevCentral administrative partition
Accounting                                                  Partition for Accounting Department
Finance                                                     Partition for Finance Departmen
```

Active Partitions

So, you've found out how to query, create, and delete partitions. The last piece of info you need to know is how you specify which partition a object is going into when you are using all of the other 2500+ iControl methods that deal directly with the configuration objects.

When you initialize an iControl connection, you authenticate with a given username. How do you determine which partition this user context is operating in? Well, it's all right there with the other Partition methods. You can use the get_active_partition() method to query the current active partition for a given user and the set_active_partition() method to change the current active partition. A reason why you might want to do this is so that when you are logging in as admin, you can create and manage objects in various partitions buckets. Here's an example of querying the current partition and then changing it.

```
PS> $ic.ManagementPartition.get_active_partition()
Common
PS> $ic.ManagementPartition.set_active_partition( (,"DevCentral") )
PS> $ic.ManagementPartition.get_active_partition()
DevCentral
```

Conclusion

By combining the User Roles aspects of the Management::UserManagement interface and the Partition methods in the Management::Partition interface, you have all the tools you need to implement and manage Administrative Domains programmatically.

Get the Flash Player to see this player.