BIG-IQ Central Management API automation and programmability - BULK Discovery, Import and Re Import - Perl

carldubois, 2016-29-06

Summary

In conjunction with the announcement of BIG-IQ 5.0 we are excited to bring the field greater flexibility when centrally managing BIG-IP devices utilizing well defined workflows and the BIG-IQ iControl REST API. We understand that automation and programmability is becoming more the norm these days as the network is evolving into a software defined application aware environment.

This article is an addendum to “BIG-IQ Central Management API automation and programmability - Python” and will demonstrate bulk device trust, discovery to populate resolver groups and import bigip configuration of many BIG-IP device as defined in a csv configuration file.

This automation can be run as a standalone utilities that will run directly in the BIG-IQ shell for adding devices to inventory in a sequential yet automated fashion.

API Reference

Trust: https://'bigiq-ip'/mgmt/cm/global/tasks/device-trust

Discovery: https://'bigiq-ip'/mgmt/cm/global/tasks/device-discovey

Import: ADC - https://'bigiq-ip'/mgmt/cm/adc-core/tasks/declare-mgmt-authority

Import: Firewall - https://'bigiq-ip'/mgmt/cm/firewall/tasks/declare-mgmt-authority

---

Let’s get started -

When using the BIG-IQ it is suggested to make a directory called scripts under /shared and securely copy this distribution into /shared/scripts/.

Contents:

../config/bulk_discovery.csv

../src/perl/bulkDiscovery.pl

../src/perl/bulkReImport.pl

Everything is predicated by a main loop which will invoke each task by calling supporting perl subroutines self-contained in the script. All rest calls, using curl (https://curl.haxx.se/), made are highlighted below.

Establishment of device trust is completed in the main loop while the process of discovery and import configurations are completed in subroutine blocks within the script.

```
#======================================================
# Main loop
# Process trust, discovery, and imports
#======================================================
for $bigip (@bigips) {
   my %postBodyHash = ("address"=>$bigiq, "userName"=>$user, "password"=>$pw,"clusterName"=>"", "useB

   my $postBody = encode_json(\%postBodyHash);

   my $trustCmd = "curl -s -k -u $bigiqCreds -H \"$contType\" -X POST -d \'$postBody\'
   https://localhost/mgmt/cm/global/tasks/device-trust";

          if (discoverModules($bigiq, $machineId)) {
             if (importModules($bigiq)) {
              }
          }

          # upgrade the framework if nessasary
          if (handleFrameworkUpdade ($trustTask, $bigip)) {
          }
} end of all devices
```

```
#======================================================
# Discover specified modules.
#======================================================
sub discoverModules {
    my %postBodyHash = ("moduleList" => \@moduleList, "status" => "STARTED");

    # POST a new discovery task
    $postBodyHash{"deviceReference"}{"link"} = "cm/system/machineid-resolver/$machineId";

    my $newDiscoverTaskCmd = "curl -s -k -u $bigiqCreds -H \"$contType\" -X POST  -d \'$postBodyJson\
} end of discoverModules

#======================================================
# A subroutine for importing individual module.
#======================================================
sub importModule {

# POST a new import task
$postBodyHash{"deviceReference"}{"link"} = "cm/system/machineid-resolver/$machineId";

my $postBody = encode_json(\%postBodyHash);

my $newImportTaskCmd = "curl -s -k -u $bigiqCreds -H \"$contType\" -X POST  -d \'$postBodyJson\' \"$d

# if we ecounter  conflicts, we mark them to use BigIQ, patch the task back to started, and poll agai

if (($currentStep eq "PENDING_CONFLICTS") or ($currentStep eq "PENDING_CHILD_CONFLICTS"))
    if (resolveConflicts($mip, $module, $currentStep,     $importSelfLink, @conflicts))
} # end of importModule

#==========================================================
# sub routine calling importModule to colocate all modules
#============================================================
sub importModules {
  $ltmSuccess = importModule($mip, "ltm", "https://localhost/mgmt/cm/adc-core/tasks/declare-mgmt-auth
  $asmSuccess = importModule($mip, "asm", "https://localhost/mgmt/cm/asm/tasks/declare-mgmt-authority
}
```

And last but not least Re Import of BIGIP configuration objects for greater than one BIGIP device.

This script can be run periodically based on Linux cron to ensure your device configurations managed by BIGIQ are up to date. On occasion other Element Management Systems could modify BIGIP object base and BIGIQ should be aware of these changes.

If you refer to the below main loop, the discovery and import call's are the same. So two things actually happen that differs from inital bulk discovery and import.

1. Trust establishment is removed as it already contains mutaul certificate trust.

2. We test if the discovery and import tasks exists, if they do we can just PATCH discovery and import tasks to enforce a re import.

That's about it. Refer to the code snippet below.

```
#=======================================================
# Main loop
# Process Re Discovery, and Imports
#=======================================================
for $bigip (@bigips) {
    ## get the device properties
    my $deviceCmd = "curl -s -k -u $bigiqCreds -H \"$contType\" -X GET https://localhost/mgmt/shared/
               ## call disc routine using ip and machine id.
               if (discoverModules($bigiq, $machineId)) {
                 ## call import routine using up and machine id.
                 if (importModules($bigiq, $machineId))
                 }
               }
} # end for devices
```

Just to re iterate the above the discovery and import routines used for Re Import just PATCH the existing task created during inital discovery and import. Here are the PATCH requests.

```
#=======================================================
# Discover specified modules.
#=======================================================
sub discoverModules {
    ## get the discovery task based on the machineId
    my $findDiscoverTaskCmd = "curl -s -k -u $bigiqCreds -H \"$contType\" -X GET \"https://localhost/

    ## If it exists PATCH the task
    if (defined $discoveryTask->{"items"}[0])
    {
        # PATCH the existing discovery task
        my $discoveryTaskSelfLink = $discoveryTask->{"items"}[0]->{"selfLink"};
        $postBodyJson = encode_json(\%postBodyHash);
        my $discoverCmd = "curl -s -k -u $bigiqCreds -H \"$contType\" -X PATCH -d \'$postBodyJson\' $
}

#=======================================================
# A subroutine for importing individual module.
#=======================================================
sub importModule {
     ##  get import task based on the machineid
     my $findImportTaskCmd = "curl -s -k -u $bigiqCreds -H \"$contType\" -X GET \"$dmaUrl?\\\$filter=



   ## If exists PATCH the task
   if (defined $findImportTask->{"items"}[0])
    {
        # PATCH the existing import task
        $importTaskLink = $findImportTask->{"items"}[0]->{"selfLink"};

        my $importCmd = "curl -s -k -u $bigiqCreds -H \"$contType\" -X PATCH -d \'$postBodyJson\' $im
}

#=========================================
# sub routine for calling importModule to collocate all modules.
#=========================================
sub importModules {
$ltmSuccess = importModule($mip, $machineId, "ltm", "https://localhost/mgmt/cm/adc-core/tasks/declare
 $asmSuccess = importModule($mip, machineId, "asm", "https://localhost/mgmt/cm/asm/tasks/declare-mgmt

}
```

If you are interested in this code for collaboration or solution, seach on key words "bigiq" "api" "python" or "perl" in code share section on dev central or here is the reference link:

https://devcentral.f5.com/codeshare/big-iq-big-ip-rest-api-bulk-device-discovery-perl-972

We will also create a repository on github for easy accessability. Please visit us soon and often for periodic updates.