Migrating MQTT Distributor Users to Chariot MQTT Credentials

The sample Python script below will allow you to migrate credentials from MQTT Distributor to Chariot at scale using the MQTT Distributor API and the Chariot REST API.

The script is designed to run from the Script Console in the Ignition instance where MQTT Distributor is installed and will:

- read the user configuration from the MQTT Distributor
- transform the users to the required Chariot REST API format
- add the users to the Chariot MQTT Credentials store
- read and output the Chariot MQTT Credentials

Prerequisites

The script requires a number of Python packages to be installed. We have included a download link to third party distributions for these prerequisite libraries which should be extracted into your <IgnitionInstallationDirectory>/user-lib/pylib/site-packages folder if not already present.

Running the script

1. In Ignition Designer, navigate to Tools > Script Console
2. Copy the script below
3. Set the configuration variables for your installation:
    - Set the *chariotUrl* variable to the URL for your Chariot installation
    - Set the *migrateDistributorUsers* dictionary to assign a password to each MQTT Distributor user to be migrated as user passwords are not returned via the MQTT Distributor API
4. Select the Execute button

```
import requests
import base64

#HTTP call method
def httpCall(method,url,headers,data):
    if method == "POST":
        resp = requests.post(url, headers=headers, data=data)
    if method == "GET":
        resp = requests.get(url, headers=headers, data=data)
    if method == "PUT":
        resp = requests.put(url, headers=headers, data=data)

    print "HTTPCall response: " + str(resp.status_code)
    #check for failures on httpCall
    if resp.status_code != 200:
        print "HTTPCall error detail: " + str(resp.json())

    print
    return resp.json()

################################################################################
# Configure Chariot URL and MQTT Distributor users to migrate
################################################################################

#Set URL for your Chariot installation
chariotUrl = "http://localhost:8080"

#Create dictionary for all MQTT Distributor users to be migrated with key/value pairs of UserName/Password
migrateDistributorUsers = {
         "admin":"password",
```

```
         "user1":"password1",
         "user2":"password2"
         }

################################################################################
headers = {}

#Accept Chariot EULA
print "Accept Chariot EULA"
url = chariotUrl + "/eula"
headers["accept"] = "application/json;api-version=1.0"
headers["content-type"] = "application/json"
data = "{\"isAccepted\": true}"
http_response = httpCall("POST",url,headers,data)

#Chariot login using admin user to get bearer token valid for 90 minutes
print "Chariot login and get bearer token"
url = chariotUrl + "/login"
headers["accept"] = "application/json;api-version=1.0"
headers["Authorization"] = "Basic " + base64.b64encode("admin:password")
data = {}
http_response = httpCall("POST",url,headers,data)
accessToken = http_response["access_token"]

#Read Distributor users
print "Read MQTT Distributor users"
print
distributorUsers = system.cirruslink.distributor.readConfig("Users")

#Create new format Access Control Lists (ACLs)
for i in range(len(distributorUsers)):
    subscribeList = ""
    publishList = ""
    userName = distributorUsers[i]["Username"]
    #check to see if MQTT Distributor user is included in migration list
    if userName in migrateDistributorUsers.keys():
            print "Migrating MQTT Distributor User: " + userName
            password = list(migrateDistributorUsers.values())[list(migrateDistributorUsers.keys()).index
(userName)]
            acls = distributorUsers[i]["ACLs"]
            acl_list = acls.split(",")
            for i in range(len(acl_list)):
                #check for leading spaces
                if acl_list[i][:1] == " ":
                    acl_list[i] = acl_list[i][1:len(acl_list[i])]
                #check for R subscriptions
                if acl_list[i][:2] == "R ":
                    subscription = acl_list[i][2:len(acl_list[i])]
                    subscription = "\"" + subscription + "\""
                    subscribeList = subscribeList + subscription + ","
                #check for W subscriptions
                if acl_list[i][:2] == "W ":
                    publish = acl_list[i][2:len(acl_list[i])]
                    publish = "\"" + publish + "\""
                    publishList = publishList + publish + ","
                #check for RW subscription
                if acl_list[i][:2] == "RW":
                    subscription = acl_list[i][3:len(acl_list[i])]
                    subscription = "\"" + subscription + "\""
                    subscribeList = subscribeList + subscription + ","
                    publish = acl_list[i][3:len(acl_list[i])]
                    publish = "\"" + publish + "\""
                    publishList = publishList + publish + ","

            #create Chariot user
            url = chariotUrl + "/mqttusers"
            headers["accept"] = "application/json;api-version=1.0"
            headers["content-type"] = "application/json"
            headers["Authorization"] = "Bearer " + accessToken
            data = "[{\"username\":\"" + userName + "\"," + "\"password\":\"" + password + "\"," + "\"acl\":{\"
publishTopics\":[" + publishList[:-1] +"],\"subscribeTopics\":[" + subscribeList[:-1] + "]}}]"
```

```
            print "Creating MQTT Credentials: " + str(data)
            http_response = httpCall("POST",url,headers,data)

    else:
              print "Not migrating MQTT Distributor User: " + userName
                print

#Read Chariot MQtt Users
print "Reading Chariot MQTT Credentials"
url = chariotUrl + "/mqttusers"
headers["accept"] = "application/json;api-version=1.0"
headers["Authorization"] = "Bearer " + accessToken
data = {}
chariotMqttUsers = httpCall("GET",url,headers,data)
print "Chariot MQTT Credentials: " + str(chariotMqttUsers)
```