# Setup access to Mulesoft kubernetes environments

# Basic setup - tools

### NOTE: If you already have access to AWS, you can skip some sections if you know what you are doing and what's the section about. Also there is probably a better folder structure, this is just how I did it, feel free to organize it to your liking.

* **create folder structure in your home folder**

mkdir -p ~/mulesoft/tools

cd ~/mulesoft/tools

* **kubectl download and install**

cd ~/mulesoft/tools

curl -LO "https://dl.k8s.io/release/$(curl -L -s https://dl.k8s.io/release/stable.txt)/bin/linux/amd64/kubectl"

sudo install -o root -g root -m 0755 kubectl /usr/local/bin/kubectl

kubectl version --client

kubectl version --client --output=yaml

kubectl cluster-info

Note : The kubectl version should be the same as the version in the jumphost, otherwise you might see an error as "error: exec plugin: invalid apiVersion "[client.authentication.k8s.io/v1alpha1](http://client.authentication.k8s.io/v1alpha1)" . The current kubectl version in jumphost is **v1.23.9** .   
  
To download v1.23.9 version :  **curl -LO** [**https://dl.k8s.io/release/v1.23.9/bin/linux/amd64/kubectl**](https://dl.k8s.io/release/v1.28.1/bin/linux/amd64/kubectl)**sudo install -o root -g root -m 0755 kubectl /usr/local/bin/kubectl**

* **helm download and install**

cd ~/mulesoft/tools

wget https://get.helm.sh/helm-v3.8.1-linux-amd64.tar.gz

tar -xvf helm-v3.8.1-linux-amd64.tar.gz

cp linux-amd64/helm /usr/local/bin/helm

sudo cp linux-amd64/helm /usr/local/bin/helm

* **awscliv2 download and install**

mkdir ~/mulesoft/tools/awscliv2

cd ~/mulesoft/tools/awscliv2

curl "https://awscli.amazonaws.com/awscli-exe-linux-x86\_64.zip" -o "awscliv2.zip"

unzip awscliv2.zip

sudo ./aws/install

aws --ver

* **aws-iam-authenticator download and install**

mkdir ~/mulesoft/tools/aws-iam-authenticator

cd ~/mulesoft/tools/aws-iam-authenticator

curl -o aws-iam-authenticator https://s3.us-west-2.amazonaws.com/amazon-eks/1.21.2/2021-07-05/bin/linux/amd64/aws-iam-authenticator

chmod +x ./aws-iam-authenticator

mkdir -p $HOME/bin && cp ./aws-iam-authenticator $HOME/bin/aws-iam-authenticator && export PATH=$PATH:$HOME/bin

echo 'export PATH=$PATH:$HOME/bin' >> ~/.bashrc

aws-iam-authenticator help

* **aws\_utils clone the repo and install**

git clone --branch better\_auth ssh://git@git.allegiantair.com:7999/devops/aws\_utils.git

python3 -m pip install aws\_utils/

awsauth -u yavor.marinov --refresh

use 'awsauth' to login and start your token

* **rtfctl**

only needed when interacting with Mulesoft Runtime Fabric

**Install rtfctl**

curl -L https://anypoint.mulesoft.com/runtimefabric/api/download/rtfctl/latest -o rtfctl

sudo chmod +x rtfctl

mkdir ~/.local\_apps && mv rtfctl ~/.local\_apps

sudo ln ~/.local\_apps/rtfctl /usr/local/bin/rtfctl

# Access Mulesoft environment

* **authenticate with awsauth - Example**

**NOTE: Use g4aws-mule-dev\_241387238312\_AWS-Role-SysEng for QA environment**

Create **.aws** and **.kube** folder in your home directory and then proceed with the below steps.

[yavor.marinov@XYZ]$ awsauth

Allegiant Air - awsauth v0.0.10

Username (yavor.marinov):

Password:

SAML assertion includes 9 roles, now generating STS credentials.

Credential written for g4aws-mule-dev\_241387238312\_AWS-Role-SysEng, Expires in 1 hour.

Credential written for g4aws-mule-qa\_286984770123\_AWS-Role-SysEng, Expires in 1 hour.

Credential written for g4aws-ops-util\_405865296163\_AWS-Role-SysEng, Expires in 1 hour.

Credential written for g4aws-mule-stage\_510308003276\_AWS-Role-SysEng, Expires in 1 hour.

Credential written for g4aws-sdlc\_697867080916\_AWS-Role-SysEng, Expires in 1 hour.

Credential written for g4aws-mule-prod\_737124451512\_AWS-Role-SysEng, Expires in 1 hour.

Credential written for g4aws-prod\_737225616887\_AWS-Role-SysEng, Expires in 1 hour.

Credential written for g4aws-master\_759062848102\_AWS-Role-SysEng, Expires in 1 hour.

Credential written for g4aws-prod-dr\_879624941994\_AWS-Role-SysEng, Expires in 1 hour.

----------------------------------------------------------------

Your new access key pair(s) has been stored in the AWS configuration file

To use this credential, call the AWS CLI with the --profile option (e.g. aws --profile <profilename> sts get-caller-identity).

----------------------------------------------------------------

* **You can see your roles in the aws credentials file**

cat ~/.aws/credentials

* **You can see a profile identity for example - clear your profile and reassign it to the desired environment, example is STG**

export AWS\_DEFAULT\_PROFILE=

export AWS\_DEFAULT\_PROFILE="g4aws-mule-stage\_510308003276\_AWS-Role-SysEng"

* **Check your current identity setup**

aws sts get-caller-identity

* Clone Anypoint Mulesoft repositories for the desired environment

# DEV

clone ssh://git@git.allegiantair.com:7999/mule/anypoint-rtf-dev.git

# QA

clone ssh://git@git.allegiantair.com:7999/mule/anypoint-rtf-qa.git

# STG

clone ssh://git@git.allegiantair.com:7999/mule/anypoint-rtf-stage.git

# PRD

clone ssh://git@git.allegiantair.com:7999/mule/anypoint-rtf-prod.git

* Setup your kubeconfig - Example for STG afrer the repo is cloned

cd anypoint-rtf-stage

cd eks/

cat kubeconfig\_rtf-stage > ~/.kube/config

* Test it - you should see nodes and pods listed

kubectl config current-context

kubectl get nodes

kubectl get pods -A