

RTF Deployment using AWS CloudFormation

April 2021

Introduction

This presentation is for MuleSoft developers, architects and SMEs who would like to move to a RTF deployment strategy using AWS as their on-prem solution.

MuleSoft currently only supports terraform for RTF deployment but this guide will illustrate how to use AWS CloudFormation instead.

A set of scripts, tools, configurations, troubleshooting steps and a guide have been created to help with the RTF deployment.

Presentation Contents

- Resource Information
- Performing the steps
- Troubleshooting

Resources Information

Resources Information

- rtf-deploy.cf contains all the resources
- s3 folder contains the configurations
- README.md contains important information and commands



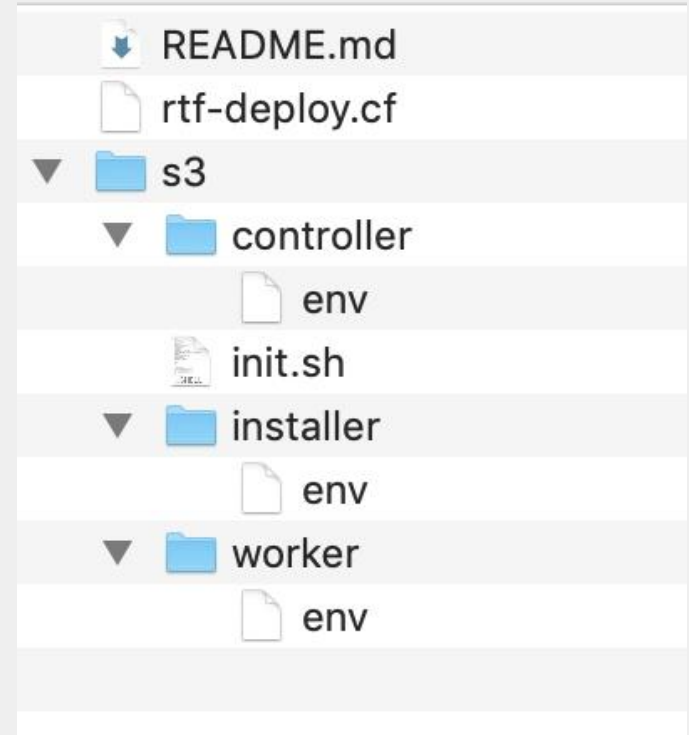
Resources Information

- rtf-deploy.cf contains info about the 1 installer node, 2 controller nodes and 3 worker nodes. Installer node is also a controller node.
- Installer node can be identified by finding the tag called “rtf-3m-controller”

```
"instancei0fda0e53484b27a31": {
  "Type": "AWS::EC2::Instance",
  "Properties": {
    "BlockDeviceMappings" : [ ...
  ],
  "DisableApiTermination": "false",
  "InstanceInitiatedShutdownBehavior": "stop",
  "EbsOptimized": "true",
  "IamInstanceProfile": { "Ref": "ec2readonlytos3ref" },
  "ImageId": "ami-0aaba9ba5c26d33c7",
  "InstanceType": "m5.large",
  "KeyName": "elogstationaws",
  "Monitoring": "false",
  "Tags": [
    {
      "Key": "Name",
      "Value": "rtf-3m-controller"
    },
    {
```

Agents

- Installer (also a controller)
- 2 more controllers
- 3 workers



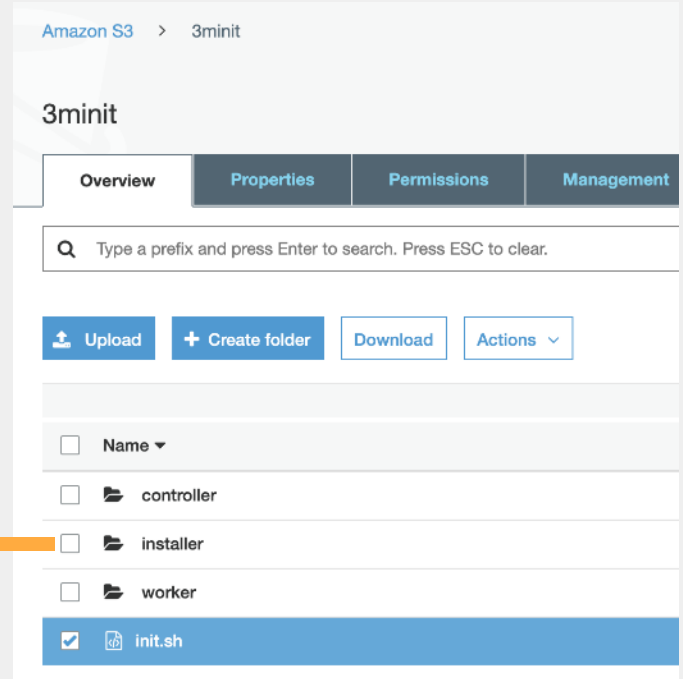
Performing the Steps

Step 1: Open s3 folder and add AppCentrica configs

- RTF_TOKEN is a random string of 16 characters, same for all 3 env configs
- RTF_REGION choose accordingly
- RTF_ACTIVATION_DATA comes from Anypoint Platform when you activate RTF
- License is obtained from MuleSoft Support Centre, encode by using Base64 (<https://docs.mulesoft.com/runtime-fabric/1.2/install-aws#base64-encode-your-mule-license-key>) to get RTF_MULE_LICENSE
- Check all 3 env configs - worker, installer and controller
- For worker and controller env file make sure you use the correct private IP of installer node

```
s3 > installer > env
1  export RTF_NODE_ROLE=controller_node
2  export RTF_INSTALL_ROLE=leader
3  export RTF_INSTALL_PACKAGE_URL=""
4  export RTF_ETCD_DEVICE_SIZE=60G
5  export RTF_DOCKER_DEVICE_SIZE=250G
6  export RTF_TOKEN="sh53WnjJtJafDMGU"
7  export RTF_NAME="rtf-3m"
8  export RTF_ACTIVATION_DATA="YW55cG9pbmQubXVsZXNvZnQuY"
9  export RTF_ORG_ID=""
10 export RTF_REGION="us-east-2"
11 export RTF_ENDPOINT="https://anypoint.mulesoft.com"
12 export RTF_AUTH_TOKEN=""
13 export RTF_MULE_LICENSE='2+w35iUhD9LYMHMU0B7Jqdigy4f'
14 export RTF_HTTP_PROXY=''
15 export RTF_NO_PROXY=''
16 export RTF_MONITORING_PROXY=''
17 export RTF_SERVICE_UID=''
18 export RTF_SERVICE_GID=''
19 export RTF_AGENT_URL=''
20 export POD_NETWORK_CIDR='10.244.0.0/16'
21 export SERVICE_CIDR='10.100.0.0/16'
```

Step 2: Upload contents of S3 folder in AWS S3



Step 3: Get Image ID based on Region

```
abains-ltm2:aws abains$ aws ec2 describe-images --owners 309956199498 --query 'sort_by(Images, &CreationDate)[*].[CreationDate,Name,ImageId]' --filters "Name=name,Values=RHEL-7.8_HVM_GA-20200225-x86_64-1-Hourly2-GP2" --region us-east-2 --output table
```

DescribeImages		
2020-02-26T16:46:55.000Z	RHEL-7.8_HVM_GA-20200225-x86_64-1-Hourly2-GP2	ami-0aaba9ba5c26d33c7

```
$ aws ec2 describe-images --owners 309956199498 --query 'sort_by(Images, &CreationDate)[*].[CreationDate,Name,ImageId]' --filters "Name=name,Values=RHEL-7.8_HVM_GA-20200225-x86_64-1-Hourly2-GP2" --region us-east-2 --output table
```

Step 4: Select availability zones for Agents

```
"subnet0ef1dbde7b21db9ae": {
  "Type": "AWS::EC2::Subnet",
  "Properties": {
    "CidrBlock": "172.31.2.0/24",
    "AvailabilityZone": "us-east-2c",
    "VpcId": {
      "Ref": "vpcmain"
    }
  },
  "Tags": [
    {
      "Key": "Name",
      "Value": "rtf-3m-subnet"
    },
    {
      "Key": "ROLE",
      "Value": "RuntimeFabric-terraform"
    }
  ]
},
```

```
{
  "Key": "Name",
  "Value": "rtf-3m-controller-1"
},
{
  "Key": "ROLE",
  "Value": "RuntimeFabric-terraform"
}
],
"Volumes": [
  {
    "Device": "/dev/xvdb",
    "VolumeId": {
      "Ref": "volumevol0ef66e88d6e77b8e9"
    }
  },
  {
    "Device": "/dev/xvdc",
    "VolumeId": {
      "Ref": "volumevol0bd79cf47ae90f2cf"
    }
  }
],
"NetworkInterfaces": [
  {
    "DeleteOnTermination": "true",
    "DeviceIndex": 0,
    "SubnetId": {
      "Ref": "subnet0ef1dbde7b21db9ae"
    }
  }
],
```

Step 5a: Create Role in AWS IAM (To read from S3)

Roles > ec2readonlytos3

Summary

Role ARN	arn:aws:iam::525806787271:role/ec2readonlytos3
Role description	Allows EC2 instances to call AWS services on your behalf. Edit
Instance Profile ARNs	arn:aws:iam::525806787271:instance-profile/ec2readonlytos3
Path	/
Creation time	2020-09-28 11:25 EDT
Last activity	Not accessed in the tracking period
Maximum session duration	1 hour Edit

Permissions | Trust relationships | Tags | Access Advisor | Revoke sessions

▼ Permissions policies (1 policy applied)

[Attach policies](#)

Policy name ▼	Policy type ▼
▶ AmazonS3ReadOnlyAccess	AWS managed policy

Step 5b: Reference role in CF: IamInstanceProfile

```
"Type": "AWS::EC2::Instance",
"Properties": {
  "DisableApiTermination": "false",
  "InstanceInitiatedShutdownBehavior": "stop",
  "EbsOptimized": "true",
  "IamInstanceProfile": "arn:aws:iam::525806787271:instance-profile/ec2readonlyos3",
  "ImageId": "ami-0aaba9ba5c26d33c7",
  "InstanceType": "m5.large",
```

Step 5c: Access S3 from EC2

```
"UserData": { "Fn::Base64" : { "Fn::Join" : [ "", [
  "#!/bin/bash ", "\n",
  "mkdir -p /opt/anypoint/runtimefabric ", "\n",
  "cd /opt/anypoint/runtimefabric ", "\n",
  "yum install zip unzip -y", "\n",
  "curl 'https://awscli.amazonaws.com/awscli-exe-linux-x86_64.zip' -o 'awscliv2.zip'", "\n",
  "unzip awscliv2.zip", "\n",
  "./aws/install", "\n",
  "/usr/local/bin/aws s3 cp s3://3minit/installer/env env", "\n",
  "cat env >> /etc/environment ", "\n",
  "source /etc/environment ", "\n",
  "/usr/local/bin/aws s3 cp s3://3minit/init.sh init.sh", "\n",
  "chmod +x init.sh ", "\n",
  "/opt/anypoint/runtimefabric/init.sh >> outputofinit.log ", "\n"
```

Step 6: Change Security Group Owner ID

- Change this to the owner of the Amazon Account or the user/service account used for creation of CloudFormation Script

```
    "SourceSecurityGroupOwnerId": "525806787271"
```


Step 7: Change KeyName to pem file being used

```
"ImageId": "ami-0aaba9ba5c26d33c7",  
"InstanceType": "m5.large",  
"KeyName": "elogstationaws",  
"Monitoring": "false",
```

Step 8: Make changes to CF Script

```
"UserData": { "Fn::Base64" : { "Fn::Join" : [ "", [
  "#!/bin/bash ", "\n",
  "mkdir -p /opt/anypoint/runtimefabric ", "\n",
  "cd /opt/anypoint/runtimefabric ", "\n",
  "curl -O https://3minit.s3.us-east-2.amazonaws.com/controller/env ", "\n",
  "cat env.sh >> /etc/environment ", "\n",
  "source /etc/environment ", "\n",
  "curl -O https://3minit.s3.us-east-2.amazonaws.com/init.sh ", "\n",
  "chmod +x init.sh ", "\n",
  "/opt/anypoint/runtimefabric/init.sh >> outputofinit.log ", "\n"
]] }
```

```
"instancei0646f55cdd6a48d85": {
  "Type": "AWS::EC2::Instance",
  "Properties": {
    "BlockDeviceMappings" : [
      {
        "DeviceName" : "/dev/sda1",
        "Ebs" : {
          "VolumeType": "gp2",
          "VolumeSize" : "90"
        }
      }
    ],
    "DisableApiTermination": "false",
    "InstanceInitiatedShutdownBehavior": "stop",
    "EbsOptimized": "true",
    "ImageId": "ami-0aaba9ba5c26d33c7",
```

Step 9: Use AWS CloudFormation

The screenshot displays the AWS CloudFormation console interface. At the top, the AWS logo and 'Services' dropdown are visible. Below the navigation bar, there are icons for file operations and a 'Close' button. The main area is titled 'Resource types' and lists various AWS services such as ACMPCA, AccessAnalyzer, AmazonMQ, Amplify, ApiGateway, ApiGatewayV2, AppConfig, AppFlow, AppMesh, AppSync, and ApplicationAutoScaling. To the right, a diagram titled 'File: 'template1'' shows a VPC resource ('vpcmain') connected to two subnets ('subnet0c9...' and 'subnet0ef...'). Each subnet is associated with an EC2 instance ('instance1...' and 'instance2...'). The diagram uses dashed lines to show dependencies and solid lines for direct connections. At the bottom, the template code is displayed in a text editor, showing the JSON structure for the VPC resource. The language is set to JSON. Below the code editor, there are tabs for 'Components' and 'Template'.

```
1 - {
2   "AWSTemplateFormatVersion": "2010-09-09",
3   "Resources": {
4     "vpcmain": {
5       "Type": "AWS::EC2::VPC",
6       "Properties": {
```

Final Product

Final Product

The screenshot displays the 'Runtime Manager' interface. On the left is a navigation sidebar with categories: Applications, Servers, Alerts, VPCs, VPNs, Runtime Fabrics, and Load Balancers. The main content area shows details for a runtime instance 'rtf-3m3'. The instance is 'Active' (indicated by a green dot). Other details include Version 1.7.26, Appliance Version 1.1.1599670979-d08f56e, Ownership Owned, vCPU 4.65 cores, and Memory 44.1 GB. Below the details are three tabs: 'Health Details' (selected), 'Associated Environments', and 'Inbound Traffic'. The 'Health Details' tab shows a green banner stating 'All systems operational' with a timestamp 'Updated: 3 minutes ago'. Below this is a table of health checks:

Health Check	Status
Appliance Status	✓ Healthy
Create/Manage Applications Deployments	✓ Healthy
Inbound Traffic To Applications	✓ Healthy

Troubleshooting

Troubleshooting

- Check AWS CloudFormation for issues
- Open the ec2 servers(agents) and navigate to /opt/anypoint/runtimefabric and view contents of outputofinit.log

```
[ec2-user@ip-172-31-0-154 runtimefabric]$ cd /opt/anypoint/runtimefabric/  
[ec2-user@ip-172-31-0-154 runtimefabric]$ ls  
~  env  init.sh  installer  installer.tar.gz  outputofinit.log  rtfctl  
[ec2-user@ip-172-31-0-154 runtimefabric]$ tail -10 outputofinit.log  
Done.  
  
17 / 17: Wait for connectivity  
=====  
time="2020-09-23T22:23:25Z" level=info msg="Waiting up to 10m0s for readiness\n"  
Runtime Fabric is ready.  
Done.  
  
Runtime Fabric installation complete.  
[ec2-user@ip-172-31-0-154 runtimefabric]$ █
```