Mulesoft at Capgemini

Reference doccument

# Contents

[1. Contents 2](#_Toc476744778)

[2. Introduction 4](#_Toc476744779)

[3. Prerequisites 4](#_Toc476744780)

[4. Setting up Development and Team Environment 4](#_Toc476744781)

[4.1. Creation Docker environment (container) 5](#_Toc476744782)

[4.1.1. Install Docker 5](#_Toc476744783)

[4.1.2. Build images and Containers 5](#_Toc476744784)

[4.1.3. Access Container 5](#_Toc476744785)

[4.1.4. List of services installed 5](#_Toc476744786)

[4.2. Checking services 5](#_Toc476744787)

[4.2.1. Nginx 6](#_Toc476744788)

[4.2.2. Artifactory 6](#_Toc476744789)

[4.2.3. Jenkins: 6](#_Toc476744790)

[4.2.4. Postfix 8](#_Toc476744791)

[4.2.5. mysql 9](#_Toc476744792)

[4.2.6. MongoDB 9](#_Toc476744793)

[5. Application & Team Development 10](#_Toc476744794)

[5.1. Managing Mule Project with Maven 10](#_Toc476744795)

[5.1.1. Use software project management tool 10](#_Toc476744796)

[5.1.2. Manage dependencies 10](#_Toc476744797)

[5.2. Managing Mule Code 10](#_Toc476744798)

[5.2.1. Maintain Mule source Code 10](#_Toc476744799)

[5.2.2. Best Pratices 10](#_Toc476744800)

[5.3. Continous Integration 10](#_Toc476744801)

[5.3.1. Create A CI Job 11](#_Toc476744802)

[5.3.2. Trigger the builds 11](#_Toc476744803)

[5.3.3. Automate deployments 11](#_Toc476744804)

[5.4. Configuration Management 11](#_Toc476744805)

[5.5. Unit Test with MUnit 11](#_Toc476744806)

[5.5.1. Create acceptance criteria 11](#_Toc476744807)

[5.5.2. Fail and Pass tests 11](#_Toc476744808)

[5.5.3. Refactor test cases 11](#_Toc476744809)

[5.5.4. Refactor Mule Applications 11](#_Toc476744810)

[5.6. Deployment and Release Process 11](#_Toc476744811)

[6. Cases Study 11](#_Toc476744812)

[6.1. Case study 0 11](#_Toc476744813)

[6.2. Case study 1 Insert data in salesforce 12](#_Toc476744814)

[6.3. Case Study 2 Display frontend data from salesforce 12](#_Toc476744815)

[6.4. Case Study 4 Twitter, AI and Email 12](#_Toc476744816)

[6.5. Case Study 5 Mulesoft and Raspbery pi 12](#_Toc476744817)

[6.6. Case Study 6 Mulesoft Devops – Building with Jenkins and Artifactory 12](#_Toc476744818)

[7. References 12](#_Toc476744819)

# Introduction

Hi there! This is a manual for starting Mulesoft engineers. Maybe you just started at Capgemini or maybe you recently heard about Mulesoft.

This guide will get you started from the very beginning and will allow you to do some interesting case studies that you could encounter at a client.

Work your way through the cases one by one, our solution is available on github so you can compare yours to ours.

# Prerequisites

In order to get started you will need some tools, to understand why you need them, have a look at below picture

Anypoint Studio

(develop your integrations here)

Mule Runtime

(can be started on your local machine from Anypoint Studio)

Mule Runtime (Anypoint studio)

Code repository (Maven)

Maven

(Runs from inside anypoint studio to download libraries you need to build mulesoft and your code)

Postman

Easy tool to make webrequests to whatever you have build or any other webservices

Soap UI

Tool mainly used to make SOAP requests, even though SOAP is old it is still very much used

Mule Runtime (On Premise)

Build engine

(Jenkins)

Webserver

(Nginx)

Database

(MongodB)

FTP server

(vsftpd)

Mail server

(vsftpd)

* The tools mentioned in the blue blocks will need to be installed on your local machine
* The tools in the orange blocks are normally installed outside of your local machine (somewhere in the cloud or in some client systems) since we do not have a client here we will install them using docker on our local machine.
* The blocks in red are in the cloud and already setup, so we do not need to do the setup ourselves. We will however use these tools in this tutorial.
* The yellow blocks are not necessarily related to mulesoft but come in handy for these cases

# Setting up Development and Team Environment

The purpose is to get familiar with DevOps environment tools. As mentioned previously some tools will be installed and containerized with docker and others should be installed in the local host .

## Creation Docker environment ( container)

The script for creating this envirnonment is host in GitHub …

### Install Docker

It is recommended to install version > 1.12

<https://docs.docker.com/engine/getstarted/step_one/>

### Build images and Containers

The script consists of pulling image of each services from DockerHub via compose file and then installing required software on it.

1. First of all, get the docker script and associated configuration files from GitHub.
2. Create and build images and containers in detached mode: $ docker-compose up –d
3. Check running containers : $ docker ps –a , the status shoul be “Up”

### Access Container

To access the container run the following command where <Container ID or Name> is the Container ID or name of the container

$ docker exec –it <Container ID or Name> bash

### List of services installed

|  |  |  |  |
| --- | --- | --- | --- |
| Service | Exposed port | Initial port | Home directory |
| nginx | 8080 | 80 | /var/lib/nginx |
| node.js | 8888 | 8080 |  |
| artifactory | 8081:80 | 8081 | /opt/jfrog/artifactory  logs ->/var/opt/jfrog/artifactory/logs/catalina |
| jenkins | 8082 | 8080 | /var/lib/jenkins  ( slave agent port number : 50000) |
| postfix | SMTP : 25  POP3: 110 | 25  110 | var/lib/postfix |
| mysql | 3306 | 3306 | /var/lib/mysql |
| mongodb | 27017 | 27017 | /var/lib/mongodb  ( data directory : /data/db) |
| Mongo-ui | 8083 | 8081 | This UI interface manage mongoDB |

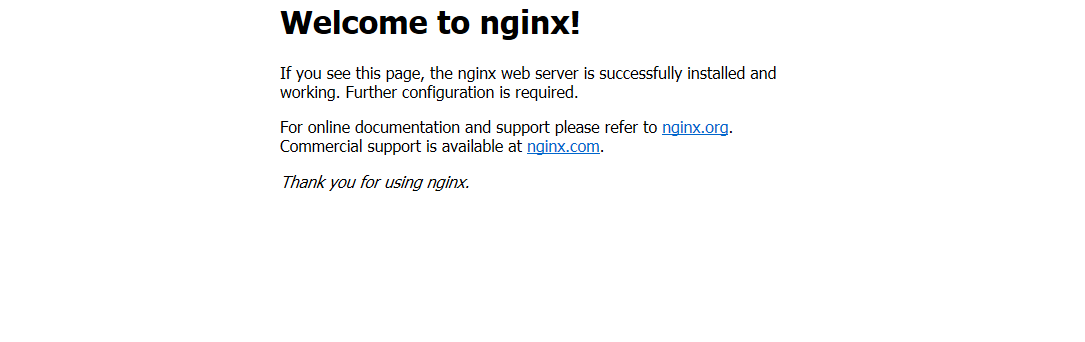
## Checking services

The Docker machine used to check the service can be obtained running the following command.

$ docker-machine ip

### Nginx

Run http://<DockerMachine IP>:8080/



## Nodejs

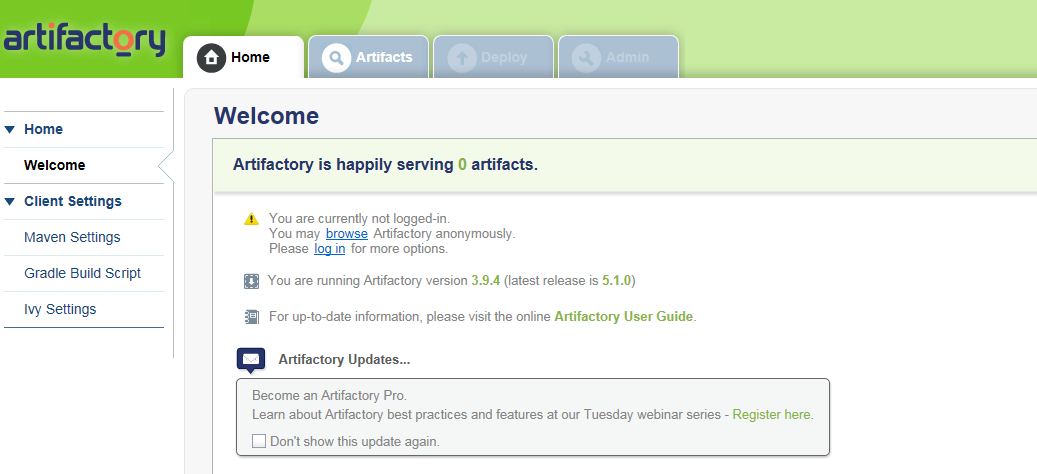
Run http://<DockerMachine IP>:8888/

Out put shoub be :

Hello from Docker container!

### Artifactory

Run http://<DockerMachine IP>:8081/



### Jenkins:

#### Get Administrator default password

$ docker-compose log jenkins

In output of the log copy the default password and keep it somewhere

jenkins | \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

jenkins | Jenkins initial setup is required. An admin user has been created and a password generated.

jenkins | Please use the following password to proceed to installation:

jenkins |

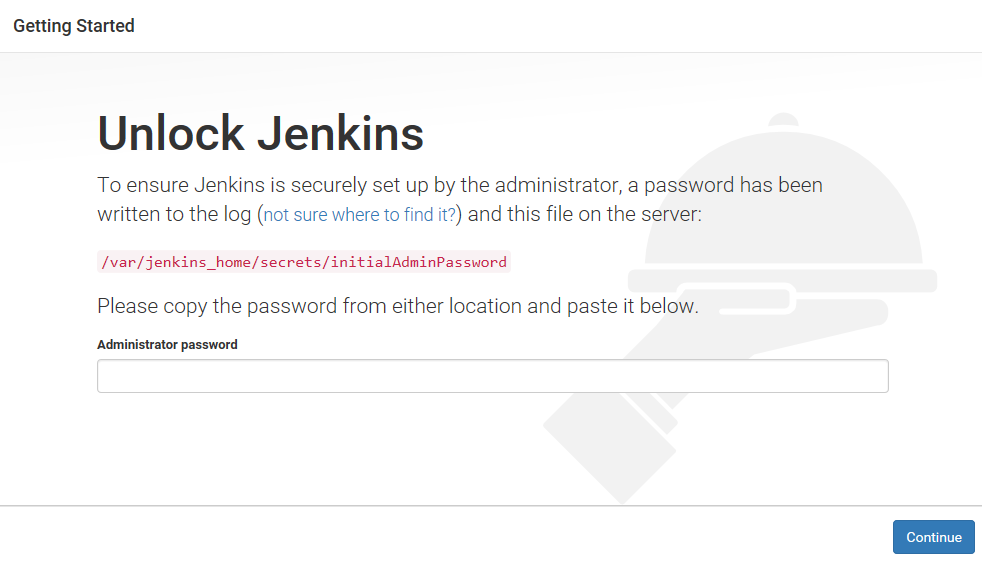
jenkins | **eca30e39e2c14f71b38b0d545c15820a**

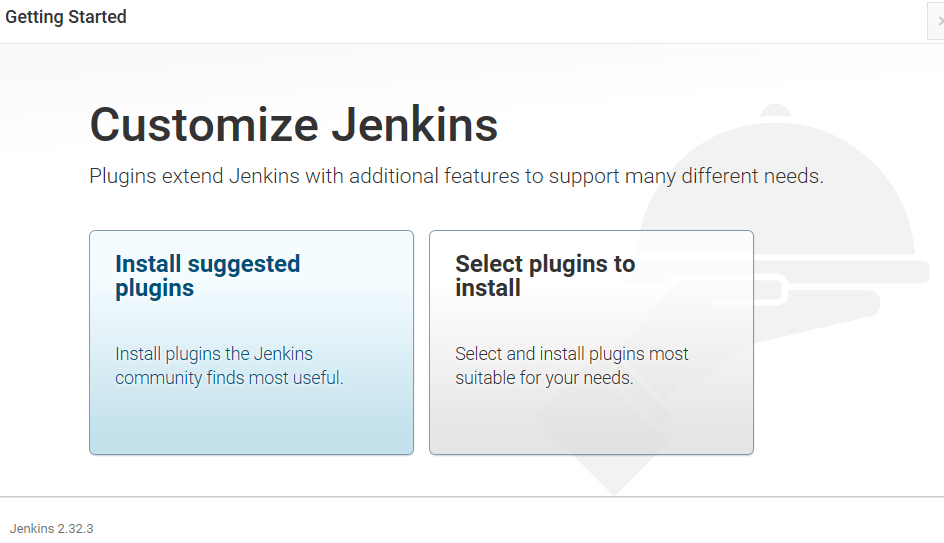
jenkins |

jenkins | This may also be found at: /var/jenkins\_home/secrets/initialAdminPassword

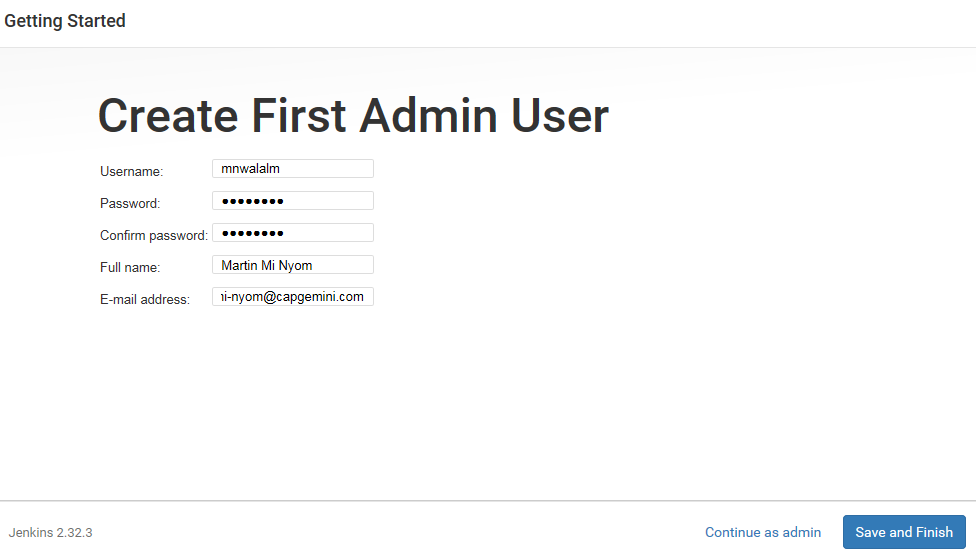
jenkins | \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

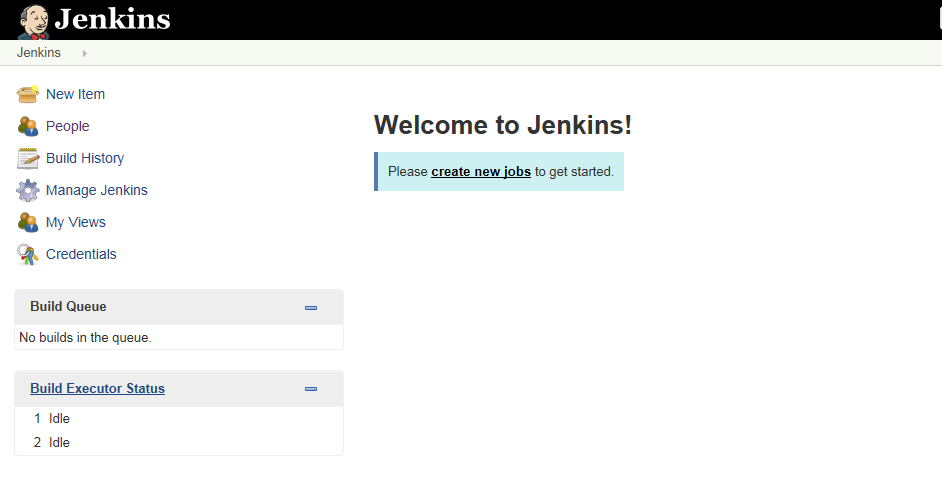
Launch http://<dockermachineIP>8082 and enter the previous password





Select “**Install suggested plugins “** and after the install of the plugins create and admin user





### Postfix

To connect to the postfix container

Run the command

telnet localhost 25

Connected to localhost.

Escape character is '^]'.

220 localhost ESMTP Postfix (Ubuntu)

En then type:

Input:

ehlo localhost

Output:

250-PIPELINING

250-SIZE 10240000

250-VRFY

250-ETRN

250-STARTTLS

250-AUTH PLAIN LOGIN CRAM-MD5 DIGEST-MD5 NTLM

250-AUTH=PLAIN LOGIN CRAM-MD5 DIGEST-MD5 NTLM

250-ENHANCEDSTATUSCODES

250-8BITMIME

250 DSN

### mysql

Check that the service is running:

$ docker exec -it <mysql container> <mysql command>

Start the mysql container if is not running . $ docker start -i mysql

And then run the following command:

$ docker exec -it mysql mysql

Welcome to the MySQL monitor. Commands end with ; or \g.

Your MySQL connection id is 16

Server version: 5.7.17 MySQL Community Server (GPL)

Copyright (c) 2000, 2016, Oracle and/or its affiliates. All rights reserved.

Oracle is a registered trademark of Oracle Corporation and/or its

affiliates. Other names may be trademarks of their respective

owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql>

### MongoDB

Run $ docker-compose logs mongo-ui

Attaching to mongo-express

mongo-express | Welcome to mongo-express

mongo-express | ------------------------

mongo-express |

mongo-express |

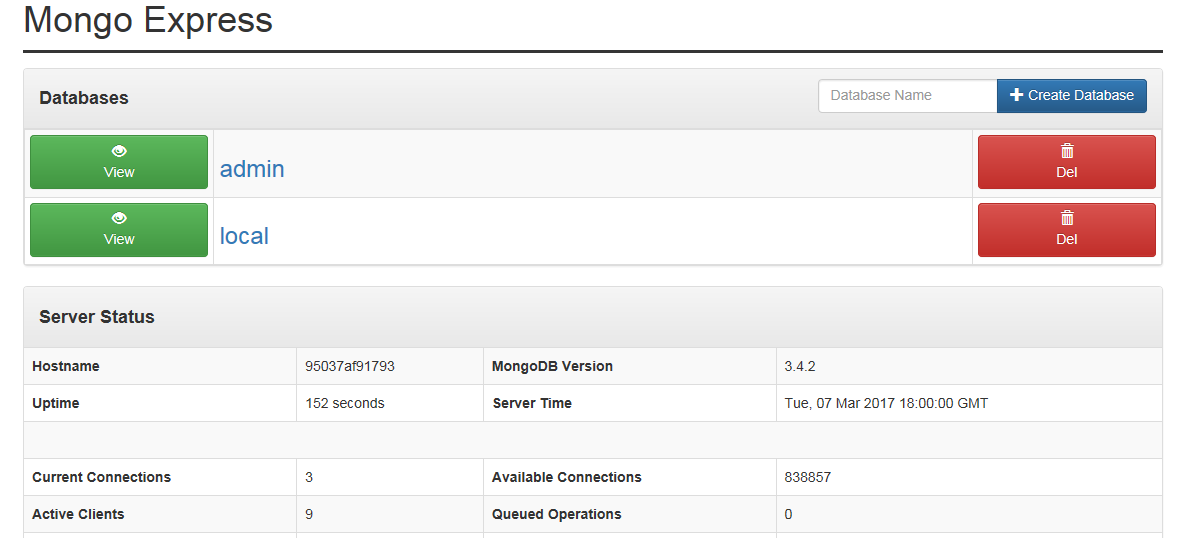
mongo-express | Mongo Express server listening at http://0.0.0.0:8081

mongo-express | Server is open to allow connections from anyone (0.0.0.0)

mongo-express | Database connected

mongo-express | Admin Database connected

then launches http://<dockermachineIP>8083



# Application & Team Development

This section should focuse on providing developers/Integration team with the information they need to work on the Mule code base.

## Managing Mule Project with Maven

### Use software project management tool

### Manage dependencies

TODO: The location in which each artefact will be packaged is determined by the dependencies declared in the POM files of each module.

## Managing Mule Code

### Maintain Mule source Code

### Best Pratices

## Continous Integration

* Jenkins
* Travis CI. I have mention Travis CI because it hosts by GitHub  and has  Notification feature when commit is done , in forming team member for new change  in real time.

### Create A CI Job

### Trigger the builds

### Automate deployments

## Configuration Management

* Chef
* Puppet

## Unit Test with MUnit

### Create acceptance criteria

### Fail and Pass tests

### Refactor test cases

### Refactor Mule Applications

## Deployment and Release Process

**Pre-Conditions**

* Make sure that the project builds without any errors.
* There is no uncommitted code locally.

Which tools to use?Maven or ANT or other.Should use the script approach for deployment and release

# Cases Study

## Case study 1 Insert data in salesforce

**Prerequisites**

**Salesforce**

Create a 30 day free trial account here:

<https://www.salesforce.com/uk/form/signup/freetrial-sales-pe.jsp?nc=70130000000NN9e>

You can sign up with your details, or alternatively with facebook or gmail

## Case Study 2 Display frontend data from salesforce

## Case Study 4 Twitter, AI and Email

## Case Study 5 Mulesoft and Raspbery pi

## Case Study 6 Mulesoft Devops – Building with Jenkins and Artifactory

# References