

ESB^{MT} Manual



Santiago Gómez Sáez, Vasilios Andrikopoulos, Steve Strauch, Johannes Wettinger
Institute of Architecture of Application Systems (IAAS)
University of Stuttgart
Web: <http://www.iaas.uni-stuttgart.de>

Introduction

The ESB^{MT} is an extended variant of the Apache ServiceMix, which is an Enterprise Service Bus (ESB). The extension of ServiceMix is focused on multi-tenancy support. This manual describes the configuration and deployment of the ESB^{MT} as well as the integration with the [4CaaS](#) platform in terms of monitoring, accounting, and the 4CaaS marketplace.

Configuration and Deployment

The deployment of the ESB^{MT} is realized using Chef and has been tested on Flexiscale, AWS EC2, and HP Cloud using the following parameters:

- Ubuntu Linux 10.04 LTS Server 64bit
- 2 CPU Cores
- 4 GB RAM

The following subsections describe the deployment of the ESB^{MT} using Chef solo, i.e., Chef without a client/server infrastructure. However, the approach can be adapted to be used in conjunction with Chef server.

Preparation of the VM

Once the VM is provisioned, login to the machine using SSH and run the following commands as “root”, e.g., using the “sudo” command:

- `echo "deb http://apt.opscode.com/ lucid-0.10 main" >> /etc/apt/sources.list.d/opscode.list`
- `sudo mkdir -p /etc/apt/trusted.gpg.d`
- `gpg --keyserver keys.gnupg.net --recv-keys 83EF826A`
- `gpg --export packages@opscode.com | sudo tee /etc/apt/trusted.gpg.d/opscode-keyring.gpg > /dev/null`
- `aptitude update`
- `aptitude -y install chef`

Then, the file “solo.rb” has to be written to the current working directory:

```
file_cache_path "/var/chef"  
cookbook_path  "/var/chef/cookbooks"  
role_path     "/var/chef/roles"
```

All the cookbooks that are required for the deployment can be retrieved from:

[Download ESBMT cookbooks.zip](#)

These cookbooks need to be unzipped to “/var/chef/cookbooks”. With this, the machine is prepared to run the Chef recipes.

Running the Chef Recipes

To run the Chef recipes for the deployment of the ESB^{MT}, a number of attributes need to be set.

These attributes are defined in JSON files that can be retrieved from:

[Download ESBMT attributes.zip](#)

These files need to be unzipped to the current working directory.

Then, the Chef client needs to be called two times using the following commands:

- `chef-solo -c ./solo.rb -j ./attrs_1.json`
- `chef-solo -c ./solo.rb -j ./attrs_10.json`

The first call installs the PostgreSQL database server, which is required by the ESB^{MT}. The second call deploys the ESB^{MT} itself.

The “attrs_10.json” file can be customized to deploy arbitrary application components (ACs) such as SoapUI test suites to deploy additional tenants or service assemblies that are deployed into the ServiceMix instance. This can be done by customizing the JSON code, starting from line 66 where the application components are defined.

Integration with 4CaaSt Monitoring

The integration with the 4CaaSt monitoring has been achieved by a number of KPI definitions inside the “ExtServiceMix_PIC” cookbook. The corresponding attributes file is located in “/attributes/default.rb” and contains the following KPI definitions:

```
kpis = {  
  "jmx_url" => "service:jmx:rmi:///jndi/rmi://localhost:1099/karaf-root",  
  "kpis" => {  
    "ExtServiceMixNode_ProcessCpuTime" => {  
      "on" => "java.lang:type=OperatingSystem",  
      "att" => "ProcessCpuTime",  
      "period" => "1"  
    },  
    "ExtServiceMixNode_SystemLoadAverage" => {  
      "on" => "java.lang:type=OperatingSystem",  
      "att" => "SystemLoadAverage",  
      "period" => "1"  
    },  
    "ExtServiceMixNode_TotalPhysicalMemorySize" => {
```

```

        "on" => "java.lang:type=OperatingSystem",

        "att" => "TotalPhysicalMemorySize",

        "period" => "1"

    },

    "ExtServiceMixNode_TotalSwapSpaceSize" => {

        "on" => "java.lang:type=OperatingSystem",

        "att" => "TotalSwapSpaceSize",

        "period" => "1"

    }

}

}

```

These definitions are read by the 4CaaS platform to connect to the ServiceMix instance using its JMX interface. The goal of the monitoring is to measure the following KPIs:

- CPU load
- System load (average)
- Physical memory size (total)
- Swap space size (total)

Integration with 4CaaS Accounting

The integration with the accounting facilities of the 4CaaS platform is realized using a separate component implemented in Java (JAR file). This component is integrated with the “ExtServiceMix_PIC” cookbook, which is responsible for deploying the ESB^{MT}. The accounting component comes with a default configuration that is located in “/opt/accounting_config.properties”:

```

# All the configurable properties are stored here

### JMX Connection Settings ###

jmx.host=localhost

jmx.port=1099

jmx.username=smx

jmx.password=smx

### Metering Instruction Id ###

MeteringInstructionId=bb3dcbf0-c6d1-11e2-8b8b-0800200c9a66

### File Path for the mapping file ###

map.filepath=/opt/4caast-esbmt-tenants.map

```

```
### REST URI to PUT JSON object ###
```

```
rest.uri=http://109.231.78.229:8181/
```

```
### Period to re-check for endpoints modifications ###
```

```
schedule.time=30000
```

This configuration can be customized in case the interface changes.