MuleSoft Deployment Mastery

On-Premise Configuration & Mule Domains

**Hybrid
Architecture**

Runtime Manager ↔ On-Prem Server

Topics Covered:

On-Premise Runtime Setup · The Mule Agent
Hybrid Deployment Steps · Domain Projects (Shared Resources)

Prepared based on Training Slides by:
MuleSoft Expert Trainer
December 9, 2025

Contents

Chapter 1

On-Premise Deployment Fundamentals

1.1 Definition

On-premise deployment refers to hosting the hardware, software, and infrastructure required for the application on local servers managed by the organization's IT team[3]. In this model, the organization retains full control over the environment where the Mule Runtime Engine operates.

1.2 Comparison: On-Premise vs. CloudHub

While CloudHub offers a flexible, scalable, and cost-effective option where maintenance is handled by the provider[4], On-Premise is often chosen for specific regulatory or architectural constraints.

On-Premise (Customer Hosted)	CloudHub (MuleSoft Hosted)
You manage the OS, Java, and patching.[5]	Fully managed iPaaS (Platform as a Service).[4]
Full control over network security.	Runs on AWS (Public Cloud).
Vertical Scaling: Add RAM/CPU to the server.	**Horizontal Scaling:** Add more workers.
Manual updates required.	Zero-downtime updates automatic.
Best for: Legacy connectivity, GDPR/-Data Residency requirements.	Best for: Cloud-first strategies, rapid scaling.

Table 1.1: Deployment Model Comparison

1.3 Architecture: The Hybrid Model

The most common implementation is the **Hybrid Model**. The "Control Plane" (Anypoint Platform) is in the cloud, but the "Runtime Plane" (where apps run) is on your server.
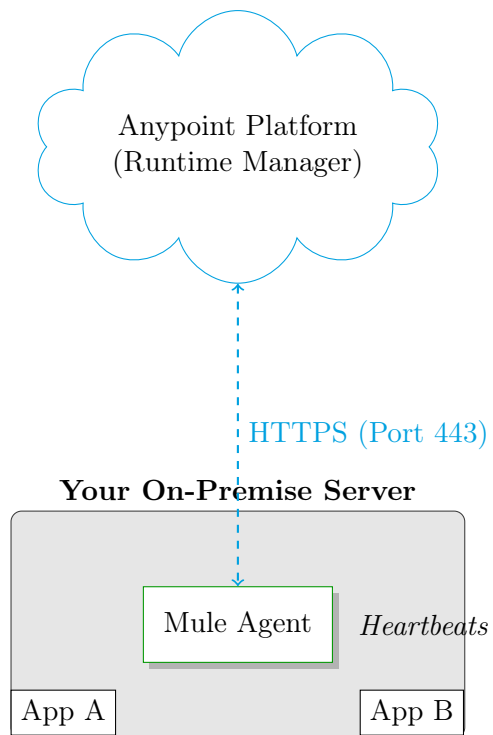
Figure 1.1: Hybrid Architecture: The Role of the Mule Agent

Chapter 2

Configuring the On-Premise Server

2.1 Prerequisites

Before starting the configuration steps[5], ensure:

1. **JDK Installed:** Mule Runtime requires OpenJDK 8 or 11.

2. **Mule Runtime Zip:** Download the standalone server zip file.

3. **Unzip:** Extract the contents (e.g., to `C:\mule-enterprise-standalone`).

2.2 Step-by-Step Configuration

2.2.1 1. Generate the Token in Runtime Manager

1. Log in to **Anypoint Platform** and navigate to **Runtime Manager**.

2. Click **Servers** on the left menu.

3. Click the **Add Server** button.

4. Enter a unique name for your server (e.g., `server-onprem`).

5. A command will be generated containing a unique token (`-H <token>`). Click **Copy Command**.

2.2.2 2. Install the Agent on the Server

1. Open your command prompt (CMD) or Terminal.

2. Navigate to the `bin` directory of your unzipped Mule Runtime.

3. Paste and run the command copied from Runtime Manager.

```
1  cd C:\mule-enterprise-standalone-4.4.0\bin
2
3  # The command format:
4  # amc_setup -H <Authentication-Token> <Server-Name>
5
6  ./amc_setup -H 60c8c4ce-efd0-4cb5-ba1b-5d975a7cdfd5---834083 server-onprem
```

Listing 2.1: The AMC Setup Command

2.2.3 3. Verify Installation

Upon running the command, you should see the following output messages:

- `Mule Agent Unpacked`

- `Communication between Anypoint Management Center and Mule Agent is authorized`

- `Mule Agent configured successfully`

2.2.4 4. Start the Server

Once the agent is installed, you must start the Mule Runtime for it to connect.

```
1 mule.bat    % Windows
2 ./mule      % Linux/Mac
```

> **Result**
>
> Go back to Anypoint Runtime Manager. The server status should change from **Created** (Grey) to **Running** (Green).

Chapter 3

Deploying Applications

3.1 The Deployment Flow

Once the server is "Green" in Runtime Manager, you can deploy applications directly from the cloud console.

1. Click **Applications** → **Deploy Application**.

2. **Deployment Target:** Select **Hybrid** (not CloudHub).

3. Select your server (`server-onprem`) from the list.

4. **Upload File:** Upload the compiled JAR file of your Mule application.

5. Click **Deploy**.

3.2 Internal Workflow

1. Runtime Manager sends the JAR file to the **Mule Agent** running on your server. 2. The Agent places the file in the `apps` directory (`/mule-standalone/apps`). 3. The Mule Runtime detects the new anchor file, unzips the JAR, and initializes the flow. 4. The Agent reports the status ("Started") back to the Cloud Console.

Chapter 4

Mule Domain Projects

4.1 What is a Domain Project?

A **Mule Domain Project**[12] is a special type of project used to share global resources across multiple Mule applications deployed on the same runtime.

4.2 Why is it used?

- **Port Sharing:** If you deploy 5 apps on one server, they cannot all listen on port 8081. They will crash with `Address already in use`.

- **Solution:** Define the HTTP Listener Config (Host: 0.0.0.0, Port: 8081) **once** in the Domain. All 5 apps reference this domain and share the connection.

- **Shared Connections:** Useful for sharing Database or JMS connection pools.

4.3 Step-by-Step Implementation

4.3.1 1. Create the Domain Project

1. In Anypoint Studio: **File → New → Mule Domain Project**.

2. Name it (e.g., `mule-domain`).

3. In the `mule-domain-config.xml`, define your shared HTTP Listener.

```
1  <domain:mule-domain xmlns:http="http://www.mulesoft.org/schema/mule/http"
2      xmlns:domain="http://www.mulesoft.org/schema/mule/ee/domain">
3
4      <http:listener-config name="Shared_HTTP_Listener_config" doc:name="HTTP
   Listener config">
5          <http:listener-connection host="0.0.0.0" port="8081" />
6      </http:listener-config>
7
8  </domain:mule-domain>
```

Listing 4.1: mule-domain-config.xml

4.3.2 2. Configure the Child Application

1. Open your regular Mule Application (`sample-mule-app`).

2. Open `mule-project.xml` (or right-click project → Properties).

3. Change **Domain** from `default` to `mule-domain`.

4. In your flow's HTTP Listener, select the configuration from the dropdown. It will now see `Shared_HTTP_Listener_config`.

```
<flow name="sample-mule-appFlow">
    <http:listener config-ref="Shared_HTTP_Listener_config" path="/test"/>
    <set-payload value="Hello from Domain!" />
</flow>
```

Listing 4.2: sample-mule-app.xml

Chapter 5

Summary & Interview Guide

5.1 Common Mistakes

> **Common Deployment Errors**
>
> - **Mule Agent Version:** Using an outdated agent version that prevents communication with the Cloud console.
>
> - **Firewalls:** Forgetting to whitelist outbound traffic on Port 443. The server must be able to reach `anypoint.mulesoft.com`.
>
> - **Port Conflicts:** Deploying two apps with distinct HTTP Listeners on port 8081 without using a Domain Project.

5.2 Interview Questions

5.2.1 Q1: What is the command used to connect a local server to the Anypoint Platform?

Answer: The `amc_setup` command located in the `bin` directory. It requires a unique token generated from Runtime Manager.

5.2.2 Q2: How do you solve "Address already in use" errors when deploying multiple apps on-premise?

Answer: Use a **Mule Domain Project**. Move the HTTP Listener Configuration to the domain project so multiple applications can share the same host and port (e.g., 8081) but distinguish traffic via their `basePath`.

5.2.3 Q3: What is the difference between the Control Plane and Runtime Plane in a Hybrid setup?

Answer:

- **Control Plane:** (Cloud) Manages metadata, deployment commands, and monitoring (Runtime Manager).

- **Runtime Plane:** (On-Prem) The physical server where the Mule Runtime Engine executes the actual payload processing.

5.3 Summary

- **On-Premise Deployment** gives you full control over the infrastructure but requires manual maintenance.

- The **Hybrid Model** uses the **Mule Agent** to bridge your local server with the Cloud Control Plane.

- **Domain Projects** are essential for On-Premise environments to allow resource sharing (Ports/DBs) across multiple applications.