December, 2014

Planning Your Implementation

Before you start installing Community Manager, there are some key factors to consider that will affect how you go about the implementation. For example, consider:

- How many containers do you want to configure?
- Are you using load balancing, and if so, where?
- Do you want to create dedicated containers for certain features, or do you want to put all features into one container? You might want to install Community Manager in the same container where Policy Manager is already installed, or in a separate container that you create. This will affect the installation features you choose and the installation steps.
- Will the containers be installed in the DMZ or inside the network?
- Which containers will have a direct database connection, and which will not?

Your decisions on these points, and other considerations such as unique security requirements or architectural features of your implementation, will affect how you structure your implementation. Based on the unique requirements of your installation, you can plan your containers for deployment and determine which features will be installed on which containers.

Review all the information in this chapter, and make sure that you take these factors into consideration in planning your implementation.

Required Community Manager Features

The core installation features required for Community Manager to work are:

- SOA Software Community Manager APIs

  This feature includes all the REST APIs offered by the Community Manager platform. These APIs are the backbone for the Community Manager back end and are needed for the developer UI portal.

  This feature does not require any other feature to be installed in the same container.

- SOA Software Community Manager Scheduled Jobs

  This feature includes the scheduled jobs to support Community Manager features. It could be running on the same container as the APIs or in a different container. This feature must be installed in all the containers on which the SOA Software Scheduled Jobs feature is installed. All features related to Scheduled Jobs should run in the same container—Policy Manager Schedule Jobs and Community Manager Scheduled Jobs.

The Scheduled jobs feature has been packaged separately so that it can run on a separate, probably non-critical container, maybe without clustering. This allows you to set up dedicated containers for scheduled jobs without using the resources from containers that are serving external/public UI requests.

- SOA Software Community Manager Default Theme

  This feature includes the developer user portal. The app developer's browser connects to the Community Manager default theme.

  Note: The Community Manager APIs feature must be installed in the same container as the Default Theme.

Additional Requirements

On every container with Policy Manager services installed, one of the following should be installed, to provide Community Manager APIs functionality:

- Community Manager APIs
- Community Manager plug-in. If Policy Manager Services and Community Manager APIs are in the same container, the CM Plug-In is not necessary. However, if you install Community Manager APIs and Policy Manager services in two different containers, you must install the Community Manager plug-in in each Policy Manager services container.

General Configuration Factors

In addition to the requirements of Community Manager installation, take into consideration general factors that will affect your implementation, such as:

- The database connectivity from the containers
- Load balancing and failover
- Security restrictions that your organization might have. For example, perhaps:
  - The DMZ cannot connect to the database server
  - Requests cannot go directly to the internal network without having a hop in the DMZ
  - The developer user sessions and the API request traffic must be kept separate on different machines

If any such organizational requirements are a factor in your implementation, keep these in mind when planning your installation and determining which features you will install on which containers.

Optional Features

Community Manager includes several key optional features. A very brief summary is included here so that you can include any relevant features in your planning.

Optional features include:

- Security Providers
- Lifecycle Management

Security Providers

Community Manager supports several key security providers. If you are using these security providers, you will need to install add-on features to support them.

Supported security providers include:

- CA SiteMinder
- PingFederate

When you are planning your implementation, and determining how many containers you will have and which features you will install on which containers, you will also need to determine which add-on features you will need to install in your containers to support one or more of the above.

For example, if you are installing the PingFederate add-on, the PingFederate plug-in must be installed in the same container where the Community Manager APIs feature is installed, and also where the Network Director is installed.

Lifecycle Management

The SOA Software Lifecycle as a Service (LaaS) Add-On for Community Manager allows you to configure a custom workflow for certain objects in Community Manager, with custom forms to collect properties, and to exchange information between Community Manager and Lifecycle Manager.

This feature is an optional add-on and requires installation of the Lifecycle Manager product.

Checklist

Answer the questions below to help you plan your implementation.

1. What features do you need/want?
2. What are the different categories of the containers you want to configure?
3. How many containers do you want to configure?
4. How many containers do you want to have for each feature?
5. Do you want everything in one container, or do you need to separate:
   - Policy Manager console from Policy Manager services?
   - Policy Manager services from Community Manager services?
   - Network Director?
   - Scheduled jobs?
6. How do you want to group features to configure specific container configurations?

Scenarios: Container Configuration

Below are some theoretical scenarios that might influence your implementation planning.

- Joe wants a separate container for the Policy Manager console, because he doesn’t want to use the path for the Community Manager console.
- Mary wants a separate container for the Network Director dependencies, because she doesn’t want to combine API traffic with user traffic.
- Tom wants to add a separate container for Community Manager because it’s an external-facing site; he doesn’t want to combine that with the administration functionality of the Policy Manager console.
- Jack wants the Community Manager user interface, but his company wants one instance for developers and another for administrators, so he will need to configure at least two Community Manager containers.
- Stephen wants to make sure Admin users are kept in a user store separate from developer users.
- Susan doesn’t want scheduled jobs to add a lot of overhead to the same container that’s serving the user traffic or API traffic, so she wants to configure a separate, dedicated container for scheduled jobs.
- Maria needs to make sure that transaction traffic/usage/load is kept separate from web-based interactive messaging. For example, calls from a Network Director must not be impacted by the web-based interactive load.
- Kanchana needs to make sure that failover strategies are in place.
- Abhishek needs to design an implementation to support a multi-site scenario.
- Jennifer needs to include the PingFederate add-on, so she will need to install the PingFederate plug-in in the same container where the Community Manager APIs feature is installed, and also in the container where the Network Director is installed.
- Matt wants to make sure “User” traffic is separated from “Admin” traffic—admin traffic must be restricted to a secure network.

Scenarios: Feature Breakout

Once you’ve decided on your container configuration, you’ll need to determine which containers will include which features. Then you can determine how many containers you want to configure for each of those container categories, and which features go into which container configurations. Below are some theoretical scenarios:

- You have a container category that has Network Director dependencies, and you want a lot of load balancing there because the API traffic is of critical importance in your company.
- You don’t want load balancing on scheduled jobs. One dedicated container is enough.
- You don’t need a separate container for the Administrator; Administrator work is minimal and can be performed at quiet times.
- The DMZ cannot connect to the database server.

–⧫–