Getting Started with Environment Manager

The Environment Manager module enables team members to select and provision the exact test environment they need: an environment with each of the AUT's dependent components set to the desired state. For instance, you might provision an environment with a virtual API simulating network congestion and error responses + a real database + a virtual mainframe that's returning positive responses.

Building Out the Environment

To start with Environment Manager, someone on your team (typically an admin or architect), builds *system* diagrams that map out the architecture of your AUT and all its dependent components (e.g., APIs, 3rd-party services, databases, applications, and other endpoints).The following graphic shows an example system diagram:
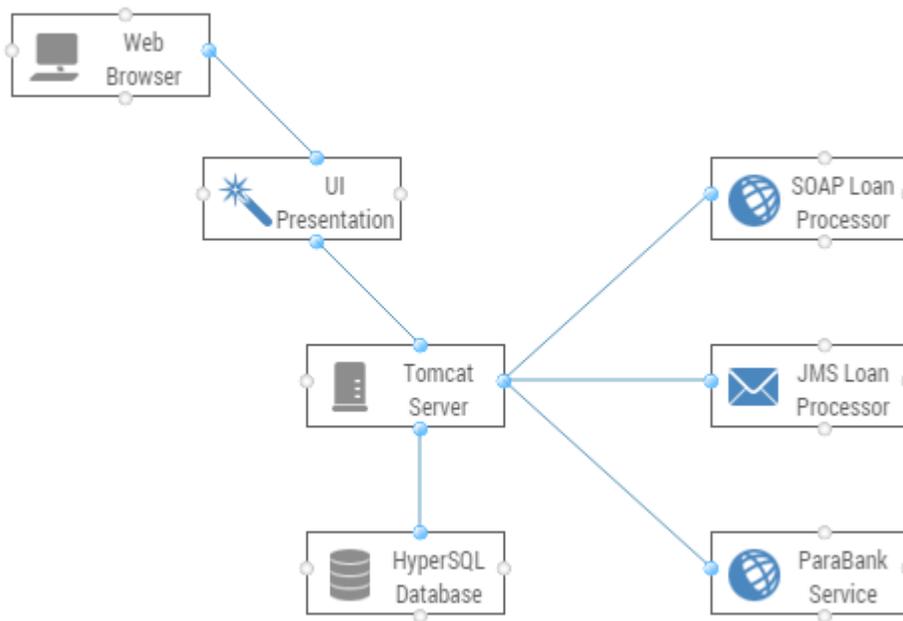
Then, one or more *environments* are defined for each system. An environment may contain the complete system architecture, or only the subset of components relevant for a particular test environment.

Next, *component instances* are added to capture the various states that each dependent component might take in the given test environment. For instance, one of the environment's 3rd-party services might be represented by 10 different virtual versions of that service—each with different combinations of performance and data profiles—as well as the real version of that same service.

Here is an example environment derived from the above system; note that it contains a number of component instances for each of the system components, and that there are several preconfigured *environment instances* that automatically set all of 5 of the configurable components to a specific state.
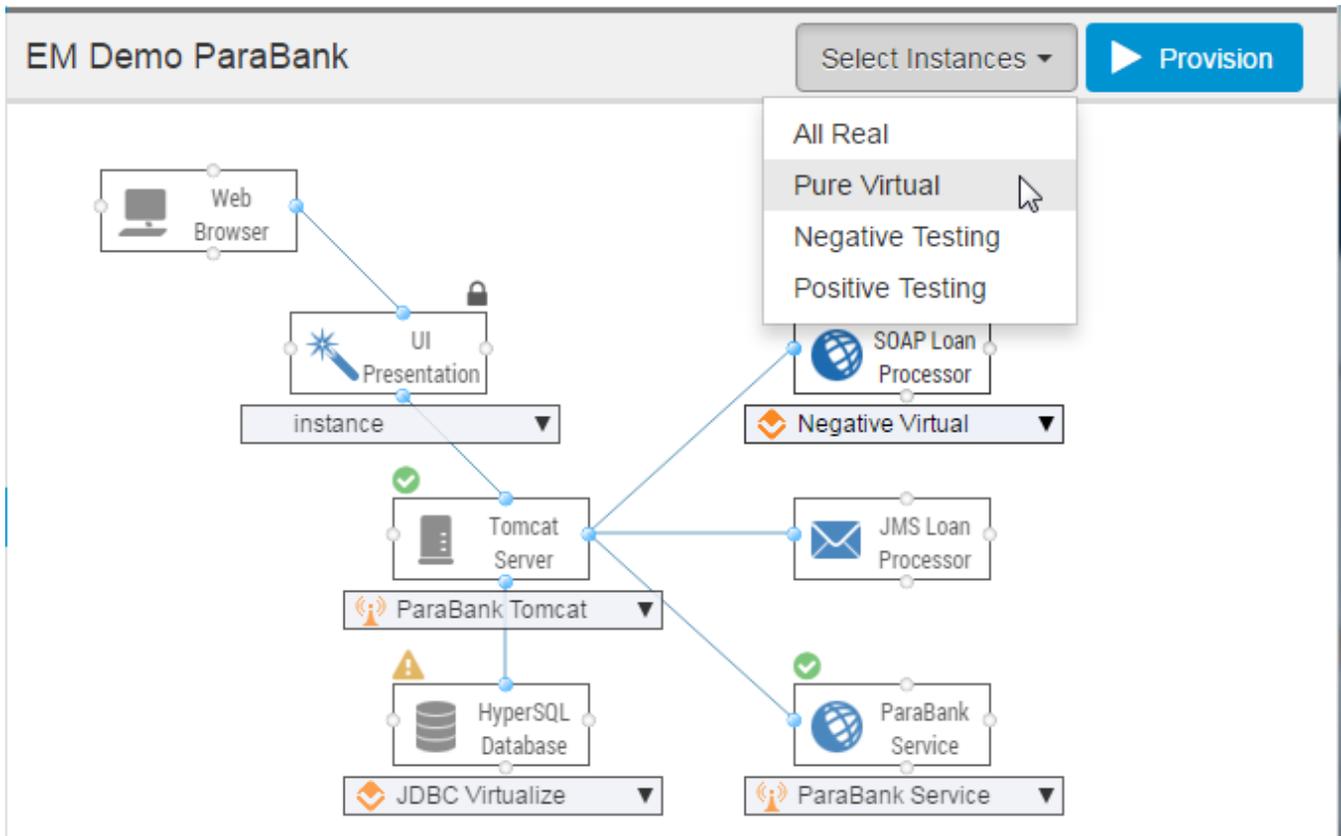
Self-Service Provisioning

Once environments with appropriate component instances are set up, team members can "self-provision" the test environments they need by setting each of the components to the desired state (or simply selecting environment instance snapshots that automatically set all components to preconfigured states) and clicking **Provision**.

Insight into System Architecture and Health

In addition to enabling test environment provisioning, the system and environment diagrams also help team members better understand the architecture of the AUT and its dependencies. Moreover, "health checks" help team members proactively identify test environment issues (e.g., dependencies that are offline, behaving unexpectedly, etc.) before they compromise the accuracy of test results. They also provide additional validation that virtual assets are accurately representing real interactions.
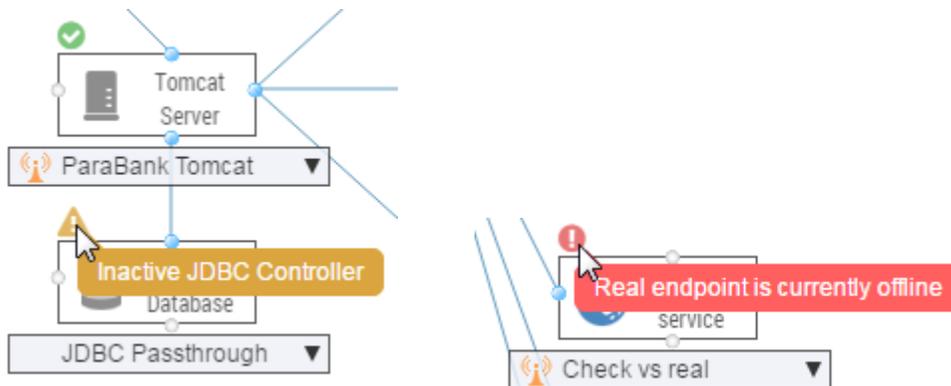
Introducing Environment Manager-Specific UI Features

As you interact with Environment Manager, note the following UI features:

- Putting a system or environment in "Edit mode" (by clicking the **Edit Diagram** toolbar button) will make a number of additional options available. You must have admin or system-level permissions to enter Edit mode.

- When you're in Edit mode, toolbar buttons such as **Save**, **Cancel**, and **Provision** are available in the top left side of the page.

- To leave Edit mode, click the **Close Editor** toolbar button:

- Messages such as provisioning status are shown in the bottom-right corner of the page. To see additional details, click the **+** sign. In some cases, you can click the message to drill down into additional details.