Integrate Puppet Enterprise and Razor and Cisco Unified Computing System

When adding bare-metal, virtual machine, or container hosts to an application workload infrastructure, a server administrator typically has to manually install the host operating systems and configure various server, networking, storage, cluster, and other settings. This traditional manual build process can take hours or days, depending on the number of servers and configuration requirements. With Cisco UCS® Manager integrated with Puppet Enterprise and Razor, multiple complex server builds can be programmatically defined and completed automatically in less than an hour.

Solution Highlights

The Cisco Unified Computing System™ (Cisco UCS) and Puppet Enterprise and Razor provide fast, flexible definition and management for complete infrastructure-as-code solutions. They can be used by administrators, developers, and operations staff for routine infrastructure lifecycle management, including the following:

- **Day-zero configuration and server role assignment with Cisco UCS service profiles**
  - Service profiles play an integral part in Cisco UCS programmability. They provide identity for the infrastructure. The infrastructure elements are stateless, and the network, storage, and server profiles create the identity for each of these elements through Cisco UCS Manager. The combination of model-based architecture and service profiles used in Cisco UCS enables safe, fast automation.

- **Day-one rapid provisioning with Razor**
  - Razor provides a comprehensive set of programmatic interfaces for provisioning, including precise control of the target OS (Red Hat Linux, Microsoft Windows, etc.) and target workload (the server’s role).
  - Cisco UCS service profiles allow flexible definition of the intended server role, which is passed automatically to Razor. They unify the process for provisioning computing, connectivity, storage, and OS resources in a single step, so they eliminate time-consuming provisioning processes across operational silos that often take days or weeks to complete.
  - Developers can use Razor to regularly wipe and reprovision test machines on Cisco UCS and Cisco HyperFlex Systems™ platforms.
  - Razor transparently integrates with Puppet Enterprise, and provisioned nodes are immediately brought under management.

- **Day-two and beyond management and monitoring with Puppet Enterprise**
  - Puppet Enterprise can install and manage all aspects of a target workload with information passed up from Cisco UCS service profiles. This capability allows definition of a target OS and workload, which are automatically provisioned and brought into the desired state by Puppet Enterprise.
  - Puppet Enterprise’s comprehensive set of modules allows a wide range of managed workloads across all major OS, virtual machine, and containerized environments.
Using Cisco UCS Manager integrated with Puppet Enterprise and Razor, you can:

- Quickly and easily create server workload roles using service profiles
- Automatically install host operating systems with Puppet Enterprise’s Razor.
- Automatically configure hosts, including deployment and management of complete application workload stacks with Puppet Enterprise

Figure 1 provides an overview of the solution components.

**Figure 1.** Overview of Puppet Enterprise Integration with Cisco UCS Management

---

**Cisco Unified Computing System**

Cisco UCS delivers smart, programmable infrastructure that simplifies and accelerates enterprise-class application and service deployment in bare-metal, virtualized, and cloud-computing environments. Unified, model-based management, end-to-end provisioning, and migration support come together in this next-generation data center platform to accelerate and simplify application deployment with greater reliability and security. Cisco UCS Manager automates the provisioning, configuration, and monitoring of the infrastructure. It includes an open API that serves as a unified control plane for integration with Puppet and a wide range of independent software vendor (ISV) configuration, orchestration, and monitoring tools.

Read more about Cisco Unified Computing System.

**Puppet Enterprise**

Since its introduction in 2005, Puppet has become one of the industry’s most important standards for IT automation. Puppet Enterprise’s extensible plug-in architecture, representational state transfer (REST) API, and powerful declarative language provide a flexible, easy-to-use platform that transparently integrates the unique capabilities of the Cisco UCS and Cisco Nexus® solutions. The combination of Cisco UCS and Puppet Enterprise provides the following benefits:
• Exceptional cost efficiency: Cisco UCS and Cisco Nexus open architectures provide Puppet Enterprise with extremely robust and detailed levels of metadata, which Puppet Enterprise uses to eliminate costly, repetitive, and error-prone tasks.

• Faster time to value: The model-based approaches of Cisco UCS Manager and Puppet Enterprise create a flexible platform that eliminates manual, time-consuming tasks and provides the capability to quickly provision, deploy, and monitor mission-critical applications in real time.

• Security from the foundation up: The precise levels of security embedded in the Cisco UCS and Puppet Enterprise solutions provide the capability to manage an entire application’s infrastructure in a proactive and consistent way. This capability reduces risk and improves IT governance while at the same time providing a truly scalable infrastructure-as-code environment.

• Declarative management: Puppet Enterprise provides declarative management that allows the user to specify and track the desired state without having to specify a procedure to achieve the desired state (users declare how they want the system configured). Puppet Enterprise typically manages resources through agents that run on the resource or through broker services that interface with managed resources.

Read more about Puppet Enterprise.

Note: You can try Puppet Enterprise for free on up to 10 nodes.

Automate Bare-Metal Provisioning with Puppet’s Razor Module

Puppet’s Razor bare-metal provisioning module works with Cisco UCS B-Series Blade Servers and C-Series Rack Servers, Cisco UCS Mini, Cisco HyperFlex™ hyperconverged infrastructure, and Cisco UCS C3260 Rack Server composable infrastructure to create completely autonomous, end-to-end infrastructure-as-code solutions. Razor’s open-source heritage makes it highly customizable and superior to the more cumbersome menu-based approach of typical preboot execution environment (PXE)–based systems.

Combining Cisco UCS hardware and integrated management with Puppet’s Razor bare-metal provisioning capabilities provides the following benefits:

• **Operational efficiency:** Puppet’s Razor module eliminates wasteful, haphazard provisioning of hardware and operating system combinations by automatically matching the right hardware with the right OS at the right time.

• **Simplicity:** Razor bare-metal provisioning can be completed in just a few steps. When combined with other Puppet Enterprise tools, Razor can automatically discover, categorize, deploy, and then boot the most appropriate OS and Cisco UCS hardware combinations.

• **Security:** The combination of powerful Cisco UCS security features such as Trusted Platform Module (TPM) and Razor’s node tagging and policy enforcement features provides an opportunity to instantly embed security from the foundation up for security-sensitive applications.

• **Flexibility:** Razor can function as a standalone system or as part of a completely integrated Puppet Enterprise solution. Razor modules can be customized to meet a particular set of needs, or Razor can be used in conjunction with other popular Puppet Enterprise tools such as Facter and MCollective to create a truly comprehensive solution.

Read more about Puppet’s Razor.
Provisioning Infrastructure and Razor: Overview

The sample workflow provided here demonstrates how Puppet Enterprise manages nodes, including a separate Razor server. In the example, host server roles are created in Razor and Puppet Enterprise, roles are associated with servers through Cisco UCS service profiles, and Razor is used to provision hosts in the desired role (Figure 2).

Figure 2. Infrastructure Overview

The Razor server used to provision nodes also provides Dynamic Host Configuration Protocol (DHCP), Trivial File Transfer Protocol (TFTP), and PXE boot services to Cisco UCS host nodes. When a Cisco UCS node is booted by Razor using PXE, Razor gathers facts about the node, matches these facts to a policy, installs the appropriate OS, and brings the node under management by Puppet Enterprise (Figure 3).
In the following example, policy matching is simplified by using Cisco UCS Manager service profiles to directly specify the intended server role. With the manager and service profiles, the user can completely manage end-to-end server provisioning. Here is the simplified workflow:

- Create a server role (OS and application workload stack) and give it a name through tags and policies in Razor and Puppet Enterprise.
- Create a Cisco UCS Manager service profile with the role name. Service profiles allow complete specification of the server hardware configuration (networking, BIOS policies, etc.) and enable transparent migration of the profile from one physical server instance to another. When the profile is assigned to a physical server, the server boots using PXE and is provisioned by Razor.
- Razor installs a host OS for the assigned role and brings the server under management by Puppet Enterprise.
Sample Host Provisioning Workflow

Puppet Enterprise and Razor are supported on several OS distributions. For detailed instructions, see the following:

- Puppet Enterprise installation instructions
- Razor installation instructions

Several options are available for running Puppet Enterprise and Razor. The following setup assumes that an existing Puppet Enterprise and Razor installation is in place along with basic DHCP, TFTP, and PXE services from the Razor host. The following steps were performed with Puppet Enterprise and Razor installed on Red Hat Enterprise Linux (RHEL) with an active Red Hat subscription. Installation in other environments is possible, but those instructions are not provided here.

After you have Puppet Enterprise and Razor servers active in your environment, your Razor server should be part of your Puppet Enterprise managed domain. This setup allows you to manage additional server deployments through the Puppet Enterprise console (Figure 4).

Figure 4. Puppet Enterprise Console Showing Sample Razor Node Group and Classes
With Puppet Enterprise and Razor configured, you can assign server roles and automatically provision new hosts with the following steps:

**Step 1. Create Razor repositories with the desired host operating systems for deployment.**

a. Here is a sample Razor command line to create a Red Hat 6.5 repository from an ISO image on a local Network File System (NFS) share:

```
razor create-repo --name redhat-6.5 --task redhat/6 --iso-url file:///nfs_isos/RHEL/RHEL_65/rhel-server-6.5-x86_64-dvd.iso
```

b. You can check the repositories currently available with the razor repos command (Figure 5).

**Figure 5. Current Razor Repositories**

```
oot@razor:~# razor repos
From https://localhost:8151/api/collections/repos:

<table>
<thead>
<tr>
<th>name</th>
<th>iso_url</th>
<th>url</th>
<th>task</th>
</tr>
</thead>
<tbody>
<tr>
<td>centos-6.7</td>
<td><a href="http://centos.sonn.com/6.7/iso/x86_64/centos/6.7-x86_64-bin-DVD1.iso">http://centos.sonn.com/6.7/iso/x86_64/centos/6.7-x86_64-bin-DVD1.iso</a></td>
<td>---</td>
<td>centos/5</td>
</tr>
<tr>
<td>redhat-6.5</td>
<td>file:///nfs_isos/RHEL/RHEL_65/rhel-server-6.5-x86_64-dvd.iso</td>
<td>---</td>
<td>redhat/6</td>
</tr>
<tr>
<td>windows-2012</td>
<td>file:///nfs_isos/Windows/Windows_2012_r2_x64/12_R2/en_windows_server_2012_r2_wit_h_update_x64_dvd_6052708.iso</td>
<td>---</td>
<td>windows/2012r2</td>
</tr>
</tbody>
</table>

Query an entry by including its name, e.g. `razor repos centos-6.7`
```

[root@razor ~]# ```
Step 2. Create a server role with Razor tags and policies.

a. To determine a server’s intended role, Razor matches server facts with defined tags. Server roles are created with several tags to match against. For example, a redhat tag is used to determine that Red Hat should be installed (a later step shows how Cisco UCS service profiles populate the dmi_oem_string_2 tag being matched):
   
   ```bash
   razor create-tag --name redhat --rule '["like", ["str", ["fact", "dmi_oem_string_2", "empty"]], ".*redhat.*"]'
   ```

b. You can view currently defined tags with the `razor tags` command (Figure 6).

```
$ razor create-tag --name redhat --rule '["like", ["str", ["fact", "dmi_oem_string_2", "empty"]], ".*redhat.*"]'
```

Figure 6. Current Razor Tags

![Current Razor Tags](image)

Note: Tags used in this example are visible to Razor using custom facts. See the [GitHub updated fact module for Razor](https://github.com/CiscoUCS/razor-facts) and the [GitHub custom extensions module for Razor](https://github.com/CiscoUCS/razor-cx) for more information.
Step 3. Configure Razor to assign servers to define roles.

a. Razor policies control the way that nodes are provisioned. In this example, policies are set so that nodes with Cisco UCS service profiles matching certain tags are installed with the appropriate OS. For example, Red Hat installations use the redhat tag to match the policy:

```bash
razor create-policy --name redhat --repo redhat-6.5 --broker pe-ucs --tag redhat --hostname 'redhat-${id}.ucsdemo.cisco.com' --root-password supersecret
```

**Note:** In this example, a custom pe-ucs broker is being used to pass additional role specifiers (for example, kvm for a Red Hat Kernel-based Virtual Machine [KVM] host) to Puppet Enterprise for complete workload installation and configuration management. You can read more about Razor brokers.

b. Use the **razor policies** command to view the current policies and status (Figure 7).

![Figure 7. Razor Policies](image)

Step 4. Create service profile templates and service profiles for the desired server roles. If you’re unfamiliar with profile templates (or any other aspects of Cisco UCS), you can view the training demos at [Cisco Demo Cloud (dCloud)](https://dcloud.cisco.com). Search for “Cisco Unified Computing System.”

a. All aspects of each logical server’s computing, network, and storage configuration can be specified through its service profile. This includes virtual network interface card (vNIC) placement, PXE boot–related networking configuration, and policies such as boot order.

b. Through service profiles, the intended server role is also defined by setting the profile name to match a defined Razor policy (Figure 8).
Figure 8. Service Profile Template Named to Match a Razor Policy

**Note:** The service profile name must contain an exact match of the role name created with a Razor tag (for example, the substring “redhat” must be present in a service profile name to match the associated “redhat” Razor policy). The profile name can have other leading or trailing characters (for example, “redhat-kvm1” will still match the “redhat” policy).
c. When service profiles are associated with physical servers and the desired power state is up, each server will begin booting and is ready for provisioning (Figure 9).

**Figure 9.**  Service Profiles Associated and Booting

![Service Profiles Associated and Booting](image)

| Note: | All aspects of service profile creation, including policy definition, can be automated with the Cisco UCS Manager Python software development kit (SDK). To get started automating infrastructure as code with Cisco UCS, read more about the Cisco supported [Python SDK](#) and see the comprehensive set of [Python SDK sample code](#). |
Step 5.  Optional: Observe the server boot on the Cisco UCS Manager keyboard, video, and mouse (KVM) console.

   a. After service profiles are associated with a physical server, each server should be booted by Razor using PXE. You can check the boot progress on the KVM console (Figure 10).

**Figure 10.** KVM Console Output of Razor Microkernel PXE Boot Operation
Step 6. Optional: After each node boots and is claimed by Razor, the Razor server should report facts from the node, including the intended role. In this example, the service profile name for the booted node is visible as a \texttt{dmi_oem_string\_2} fact that can be matched against Razor tags (Figure 11).

\textbf{Figure 11.} Razor Nodes and Facts Information

```
[root@razor ~]# razor nodes
From https://localhost:8151/api/collections/nodes:

+-----------------+-------------------+-------------------+-------------------+-------------------+
| name | chip_mac | tags | policy | metadata_count |
|-----------------+-------------------+-------------------+-------------------+-------------------+
| node15 | 00:25:b5:00:33:00 | centos | centos | 0 |
| node16 | 00:25:b5:00:00:06 | redhat, kvm | redhat | 1 |
+-----------------+-------------------+-------------------+-------------------+-------------------+

Query an entry by including its name, e.g. `razor nodes node15`

[root@razor ~]# razor nodes node16 facts
From https://localhost:8151/api/collections/nodes/node16:

dmi_oem_string\_1: Cisco1
\texttt{dmi_oem_string\_2}: SSPI:redhat-kvm1
\texttt{dmi_oem_string\_3}: SSPT:
\texttt{dmi_oem_string\_4}: SSYS:D23-UCS1
\texttt{virtual}: physical
\texttt{is_virtual}: false
\texttt{hardwareisa}: x86_64
\texttt{processors}:
Step 7. With Puppet Enterprise used as the node broker, nodes should also be visible in assigned node groups and under management by Puppet Enterprise.

a. In this example, a RedHat-KVM node group has been created and NTP brought under management by Puppet Enterprise on the node (Figure 12).

Figure 12. Razor Nodes and Facts Information

<table>
<thead>
<tr>
<th>redhat-host11</th>
</tr>
</thead>
<tbody>
<tr>
<td>A. View node graph</td>
</tr>
<tr>
<td>Facts</td>
</tr>
<tr>
<td>Class</td>
</tr>
<tr>
<td>ntp</td>
</tr>
<tr>
<td>puppetenterprise</td>
</tr>
<tr>
<td>puppetenterprise:profile:agent</td>
</tr>
<tr>
<td>puppetenterprise:profile:mcollective:agent</td>
</tr>
</tbody>
</table>

Solution Q&A

Q. What Cisco UCS hardware does this solution support?
A. Any Cisco UCS Manager domain can have roles assigned through service profiles that are automatically passed to Razor and Puppet Enterprise.

Q. How do I add Razor to an existing Puppet Enterprise environment?
A. Razor is part of Puppet Enterprise, and Puppet Enterprise can manage your Razor server in the same way other nodes are managed. Read more about Razor.

Q. Can I use the Cisco UCS Platform Emulator with Puppet Enterprise and Razor?
A. The platform emulator does not provide host OS boot capabilities, but all aspects of service profile and Cisco UCS Manager policy setting can be tested and verified on the platform emulator. Read more about the Cisco UCS Platform Emulator.
For More Information

Read more at the Cisco Marketplace for Puppet Enterprise and Razor on the Cisco UCS and Cisco Nexus platforms.

Visit the Cisco UCS Communities pages for additional information, video demos, and support forums.