Red Hat Satellite 6.12

在 Red Hat Satellite 中使用 Puppet 集成管理配置

使用 Puppet 集成来管理主机的配置，并将主机系统状态报告到 Satellite

Last Updated: 2024-02-18

Red Hat Satellite 6.12 在 Red Hat Satellite 中使用 Puppet 集成管理配置

使用 Puppet 集成来管理主机的配置，并将主机系统状态报告到 Satellite

Red Hat Satellite Documentation Team
satellite-doc-list@redhat.com

法律通告

摘要

本指南介绍了如何启用 Puppet 与 Satellite 集成，在主机上配置 Puppet 代理，如何导入 Puppet 模块，以及如何使用 Puppet 模块在 Red Hat Satellite 基础架构管理的主机上强制实施配置。

目录

向红帽文档提供反馈 3

第 1 章 使用 PUPPET 进行配置管理简介 4
1.1. PUPPET 如何与 SATELLITE 集成 4
1.2. 支持的 PUPPET 版本和系统要求 5
1.3. 启用与 SATELLITE 集成的 PUPPET 5
1.4. 在主机置备过程中安装和配置 PUPPET 代理 6
1.5. 在主机注册过程中安装和配置 PUPPET 代理 6
1.6. 手动安装和配置 PUPPET 代理 7
1.7. 执行配置管理 8
1.8. 禁用与 SATELLITE 集成的 PUPPET 9

第 2 章 管理 PUPPET 模块 10
2.1. 在卫星服务器上安装 PUPPET 模块 10
2.2. 更新 PUPPET 模块 10

第 3 章 将 PUPPET 类和环境导入到卫星中 11

第 4 章 创建自定义 PUPPET 环境 12

第 5 章 创建 PUPPET 配置组 13

第 6 章 配置 PUPPET 智能类参数 14
6.1. PUPPET 参数层次结构 14
6.2. 全局覆盖智能类参数 14
6.3. 为机构覆盖智能类参数 14
6.4. 为位置覆盖智能类参数 15
6.5. 在单个主机上覆盖智能类参数 15

第 7 章 将 PUPPET 类分配给主机组 17

第 8 章 将 PUPPET 类分配给单独的主机 18

第 9 章 在受管主机上强制 PUPPET 配置 20
9.1. 使用 SSH 运行 PUPPET ONCE 20
9.2. 了解自动强制设置的间隔 20
9.3. 在主机上设置 PUPPET 代理运行间隔 20
9.4. 设置全局 OUT-OF-SYNC INTERVAL 20
9.5. 设置 PUPPET OUT-OF-SYNC INTERVAL 20
9.6. 为主机组覆盖 OUT-OF-SYNC INTERVAL 21
9.7. 为单个主机覆盖 OUT-OF-SYNC INTERVAL 21

向红帽文档提供反馈

我们感谢您对文档提供反馈信息。请让我们了解如何改进文档。

您可以通过在 Bugzilla 中记录一个 ticket 来提交反馈：

1. 导航到 Bugzilla 网站。
2. 在 **Component** 字段中，使用 **Documentation**。
3. 在 **Description** 字段中，输入您要改进的建议。包括文档相关部分的链接。
4. 点 **Submit Bug**。

第 1 章 使用 PUPPET 进行配置管理简介

您可以使用 Puppet 管理和自动配置主机。Puppet 使用声明语言来描述受管主机 *的所需状态*。

Puppet 会增加您的工作效率，因为您可以同时管理多个主机。同时，它会减小您的配置工作，因为 Puppet 可以轻松地验证并可能更正主机的状态。

其他资源

- 开源 Puppet 文档
- Puppet Forge latex-latexa 存储库，适用于预建的 Puppet 模块存储库

1.1. PUPPET 如何与 SATELLITE 集成

Puppet 使用 server-agent 架构。Puppet 服务器是存储配置定义的中央组件。卫星服务器或任何胶囊通常部署为 Puppet 服务器，卫星 则充当此类 Puppet 服务器的外部节点类别(ENC)。受管主机运行与 Puppet 服务器通信的 Puppet 代理。

Puppet 代理收集某一 *主机的事实*，并在每次运行时将其报告给 Puppet 服务器。您可以通过在主机上运行 **puppet *事实*，以 JSON 格式显示 Puppet 事实**。

Puppet 服务器 *将事实转发*到卫星，并且存储它们以备后用。根据 *事实* 和其他定义，卫星将 ENC 应答构建到 Puppet 服务器。Puppet 服务器根据 ENC 回答编译 *目录*，并将 *目录* 发送到 Puppet 代理。

Puppet 代理评估主机上的系统状态。如果 Puppet 代理发现差异，在目录中定义 *的所需状态和实际状态之间*称为 *drifts*，则会对主机的状态执行更正。*Puppet 代理将结果报告回 Puppet 服务器，后者将其报告到卫星。*

Puppet 模块

主机 的所需状态 在 目录 中定义。目录 从分配给主机的一个或多个 Puppet 模块的 Puppet 清单编译。Puppet 模块是类、清单、资源、文件和模板的集合。Puppet 模块作为主机配置定义的组件工作。

智能类参数

如果模块支持使用参数，您可以使用智能类参数覆盖 Puppet 模块的参数。您可以将卫星中的参数定义为键值对，其行为与主机参数或 Ansible 变量类似。

Puppet 环境

您还可以创建多个 Puppet 环境来控制配置定义的版本，或管理定义的不同版本，并在将它们部署到生产环境中之前测试它们。

高级集成步骤

Puppet 与 Satellite 集成涉及以下高级别的步骤：

1. *启用 Puppet 集成。*
2. *将 Puppet 代理软件包导入到卫星中。可以像使用卫星一样管理 Puppet 代理软件包，方法是 启用红帽存储库 并使用 激活密钥 和 内容视图。*
3. *在调配、注册、手动或通过 远程作业执行的主机上安装 Puppet 代理。*

其他资源

- 管理内容
- 在 管理主机指南中的注册主机
- 在管理主机 指南中 配置和设置远程作业

下列步骤概述了如何使用 Puppet 模块安装、配置和管理 ntp 服务来提供示例。

1.2. 支持的 PUPPET 版本和系统要求

在开始使用 Puppet 集成之前，请查看支持的 Puppet 版本和系统要求。

支持的 Puppet 版本

Satellite 支持以下 Puppet 版本：

- Puppet 7

系统要求

在开始将 Puppet 与 Satellite 集成前，请确定您满足系统要求。详情请参阅 开源 Puppet 文档中的 Puppet 7 系统要求。

1.3. 启用与 SATELLITE 集成的 PUPPET

默认情况下，卫星没有配置任何 Puppet 集成。您需要根据情况启用集成。这意味着，您可以配置 Satellite 以在卫星服务器或胶囊上管理和部署 Puppet 服务器。此外，您还可以将 Puppet 服务器部署到外部的卫星，并将它与卫星集成，以进行报告、事实和外部节点分类(ENC)。

流程

1. 启用 Puppet 集成并在 Satellite 服务器中安装 Puppet 服务器：

```
# satellite-installer --enable-foreman-plugin-puppet \
--enable-foreman-cli-puppet \
--foreman-proxy-puppet true \
--foreman-proxy-puppetca true \
--foreman-proxy-content-puppet true \
--enable-puppet \
--puppet-server true \
--puppet-server-foreman-ssl-ca /etc/pki/katello/puppet/puppet_client_ca.crt \
--puppet-server-foreman-ssl-cert /etc/pki/katello/puppet/puppet_client.crt \
--puppet-server-foreman-ssl-key /etc/pki/katello/puppet/puppet_client.key
```

2. 如果要在胶囊上使用 Puppet 集成，请启用 Puppet 集成并在胶囊上安装 Puppet 服务器：

```
# satellite-installer --foreman-proxy-puppet true \
--foreman-proxy-puppetca true \
--foreman-proxy-content-puppet true \
--enable-puppet \
--puppet-server true \
--puppet-server-foreman-ssl-ca /etc/pki/katello/puppet/puppet_client_ca.crt \
--puppet-server-foreman-ssl-cert /etc/pki/katello/puppet/puppet_client.crt \
--puppet-server-foreman-ssl-key /etc/pki/katello/puppet/puppet_client.key \
--puppet-server-foreman-url "https://satellite.example.com"
```

输入您的 Satellite 服务器的 URL，作为 **--puppet-server-foreman-url** 参数的值。

1.4. 在主机置备过程中安装和配置 PUPPET 代理

您可以在置备过程中在主机上安装和配置 Puppet 代理。主机上需要配置了 Puppet 代理，以便 Puppet 与您的 Satellite 集成。

先决条件

- Puppet 必须在您的 Satellite 中启用。如需更多信息，请参阅 第 1.3 节 "启用与 Satellite 集成的 Puppet"。
- 您已启用并将 **Satellite 客户端 6** 存储库同步到 Satellite。如需更多信息，请参阅 管理内容 中的 导入内容。
- 您创建了为主机启用 **Satellite 客户端 6** 存储库的激活码。如需更多信息，请参阅 管理内容 中的 管理激活密钥。

流程

1. 进入 **Hosts > Provisioning Templates**。
2. 根据您的主机置备方法选择置备模板。如需更多信息，请参阅 置备主机中的 置备 模板类型。
3. 确保 **puppet_setup** 片断如下：

```
<%= snippet 'puppet_setup' %>
```

4. 对单个主机或主机组使用 host 参数启用 Puppet：
为 Puppet 7 添加名为 **enable-puppet7** 的主机参数，作为类型布尔值设置为 **true**。
5. 可选：要直接从 yum.puppet.com 安装 Puppet 代理，请为 Puppet 7 添加名为 **enable-puppetlabs-puppet7-repo** 的主机参数，因为类型布尔值设置为 **true**。仅当您不使用其激活密钥，才会向主机提供 Puppet 代理时使用此代理。

1.5. 在主机注册过程中安装和配置 PUPPET 代理

您可以在注册过程中在主机上安装和配置 Puppet 代理。主机上需要配置了 Puppet 代理，以便 Puppet 与您的 Satellite 集成。

先决条件

- Puppet 必须在您的 Satellite 中启用。如需更多信息，请参阅 第 1.3 节 "启用与 Satellite 集成的 Puppet"。
- 您已启用并将 **Satellite 客户端 6** 存储库同步到 Satellite。如需更多信息，请参阅 管理内容 中的 导入内容。
- 您创建了为主机启用 **Satellite 客户端 6** 存储库的激活码。如需更多信息，请参阅 管理内容 中的 管理激活密钥。

流程

1. 在 Satellite Web UI 中，进入到 **Configure** > **Global Parameters** 以全局添加主机参数。或者，您可以进入到 **Configure** > **Host Groups** 并编辑或创建主机组，来仅将主机参数添加到主机组中。
2. 使用全局参数或主机组中的 host 参数启用 Puppet 代理。添加名为 **enable-puppet7** 的主机参数，选择 **boolean** 类型，然后将值设为 **true**。
3. 在全局参数或主机组中使用以下主机参数为 Puppet 代理指定配置：
   - 添加名为 **puppet_server** 的主机参数，选择**字符串**类型，并将值设为 Puppet 服务器的主机名，如 **puppet.example.com**。
   - 可选：添加名为 **puppet_ca_server** 的主机参数，选择**字符串**类型，并将值设为 Puppet CA 服务器的主机名，如 **puppet-ca.example.com**。如果没有设置 **puppet_ca_server**，则 Puppet 代理将使用与 **puppet_server** 相同的服务器。
   - 可选：添加名为 **puppet_environment** 的主机参数，选择**字符串**类型，并将值设置为您希望主机使用的 Puppet 环境。

   在 BZ2177730 被解决前，您需要使用 host 参数来指定 Puppet 代理配置，即使 Puppet 服务器是 Capsule 服务器。
4. 进入到 **Hosts** > **Register Host**，并使用适当的激活码注册您的主机。如需更多信息，请参阅 管理主机 中的 注册主机。
5. 进入到 **Infrastructure** > **Capsules**。
6. 从所需胶囊服务器 **的操作** 列中的列表中，选择 **证书**。
7. 单击所需主机右侧的 **Sign** to，以签署 Puppet 代理的 SSL 证书。

1.6. 手动安装和配置 PUPPET 代理

您可以在主机上手动安装和配置 Puppet 代理。主机上需要配置了 Puppet 代理，以便 Puppet 与您的 Satellite 集成。

先决条件

- Puppet 必须在您的 Satellite 中启用。如需更多信息，请参阅 第 1.3 节 “启用与 Satellite 集成的 Puppet”。
- 主机必须分配有 Puppet 环境。
- 必须启用 **Satellite Client 6** 存储库并同步到 Satellite 服务器，并在主机上启用。如需更多信息，请参阅 管理内容 中的 导入内容。

流程

1. 以 **root** 用户身份登录主机。
2. 安装 Puppet 代理软件包。
   - 在运行 Red Hat Enterprise Linux 8 及更高版本的主机上：

     ```
     # dnf install puppet-agent
     ```
   - 在运行 Red Hat Enterprise Linux 7 及更早版本的主机上：

```
# yum install puppet-agent
```

3. 使用以下脚本，将 Puppet 代理添加到当前 shell 中的 **PATH** 中：

```
. /etc/profile.d/puppet-agent.sh
```

4. 配置 Puppet 代理。将 **environment** 参数设置为主机所属的 Puppet 环境的名称：

```
# puppet config set server satellite.example.com --section agent
# puppet config set environment My_Puppet_Environment --section agent
```

5. 启动 Puppet 代理服务：

```
# puppet resource service puppet ensure=running enable=true
```

6. 为主机创建证书：

```
# puppet ssl bootstrap
```

7. 在 Satellite Web UI 中，导航到 **Infrastructure** > **Capsules**。
8. 从所需胶囊服务器 **的操作** 列中的列表中，选择 **证书**。
9. 单击所需主机右侧的 **Sign** to，以签署 Puppet 代理的 SSL 证书。
10. 在主机上再次运行 Puppet 代理：

```
# puppet ssl bootstrap
```

1.7. 执行配置管理

在主机上部署 Puppet 代理后，您可以开始使用 Puppet 进行配置管理。这涉及以下高级别的步骤：

1. 在 Puppet 服务器上管理 Puppet 模块，用于安装和更新它们。
2. 将 Puppet 模块中的 Puppet 类和环境导入到卫星中。
3. 可选：从 Puppet 类创建配置组。
4. 在各种级别上配置智能类参数的覆盖。
5. 分配 Puppet 类或配置组到主机组或单独的主机。
6. 配置用于在主机上运行 Puppet 代理的间隔，以及运行 Puppet 服务器的配置执行。
7. 使用卫星 Web UI 中的报告监控配置管理。如需更多信息，请参阅管理 红帽卫星中的监控资源。
8. 配置电子邮件通知.有关更多信息，请参阅管理 红帽卫星中的配置电子邮件通知首选项。

分配 Puppet 类或配置组后，卫星会在配置的时间间隔中自动运行配置管理，以在受管主机上实施 Puppet 配置，或者您可以使用 **Run Puppet Once** 功能根据需要手动启动它。更多信息请参阅 第 9.1 节 “使用 SSH 运行 Puppet Once”。

1.8. 禁用与 SATELLITE 集成的 PUPPET

要中断在 Satellite 中使用 Puppet，请遵循以下步骤。

请注意，没有 **--remove-all-data** 参数的命令会删除 Satellite 数据库中所有与 Puppet 相关的数据。使用 **--remove-all-data** 参数时，命令还会移除 Puppet 服务器数据文件，包括 Puppet 环境。

警告

如果您使用 **--remove-all-data** 参数禁用 Puppet，则之后您将无法重新启用 Puppet。这是一个已知问题，请查看 Bug 2087067。

前提条件

- Puppet 在卫星上启用。

流程

1. 如果您在任何 Capsules 上使用 Puppet 服务器，请在所有 Capsules 上禁用 Puppet 服务器：

```
# satellite-maintain plugin purge-puppet --remove-all-data
```

2. 在 Satellite 服务器中禁用 Puppet 服务器：

```
# satellite-maintain plugin purge-puppet --remove-all-data
```

第 2 章 管理 PUPPET 模块

2.1. 在卫星服务器上安装 PUPPET 模块

您可以从 Puppet Forge 安装预构建的 Puppet 模块。Puppet Forge 是一个提供由社区贡献的 Puppet 模块的存储库。标记为 支持的 Puppet 模块由 Puppet Inc 官方支持和测试。

本例演示了如何将 ntp 模块添加到 受管主机。

流程

1. 导航到 forge.puppet.com 并搜索 **ntp**。第一个模块之一是 puppetlabs/ntp。
2. 使用 SSH 连接到您的 Satellite 服务器，再安装 Puppet 模块：

```
# puppet module install puppetlabs-ntp -i
/etc/puppetlabs/code/environments/production/modules
```

使用 **-i** 参数指定路径和 Puppet 环境，如 **production**。

安装完成后，输出类似如下：

```
Notice: Preparing to install into /etc/puppetlabs/code/environments/production/modules ...
Notice: Created target directory /etc/puppetlabs/code/environments/production/modules
Notice: Downloading from https://forgeapi.puppet.com ...
Notice: Installing -- do not interrupt ...
/etc/puppetlabs/code/environments/production/modules
|-| puppetlabs-ntp (v8.3.0)
  |-- puppetlabs-stdlib (v4.25.1) [/etc/puppetlabs/code/environments/production/modules]
```

安装 Puppet 模块的另一种方式是复制含有 Puppet 模块到上述模块路径的文件夹。确保手动解析其依赖项。

2.2. 更新 PUPPET 模块

您可以使用 **puppet** 命令更新现有的 Puppet 模块。

流程

1. 使用 SSH 连接到您的 Puppet 服务器，并找出 Puppet 模块所在的位置：

```
# puppet config print modulepath
```

这会返回如下输出：

```
/etc/puppetlabs/code/environments/production/modules:/etc/puppetlabs/code/environments/com
mon:/etc/puppetlabs/code/modules:/opt/puppetlabs/puppet/modules:/usr/share/puppet/modules
```

2. 如果该模块位于以上显示的路径中，以下命令会更新模块：

```
# puppet module upgrade module name
```

第 3 章 将 PUPPET 类和环境导入到卫星中

将安装的 Puppet 模块中的 Puppet 类和环境导入到卫星服务器，或分配任何已连接的胶囊服务器，然后将任何类分配到受管主机。

前提条件

- 确保选择 **Any Organization** 和 **Any Location** 作为上下文，否则导入可能会失败。

流程

1. 在 Satellite Web UI 中，导航到 **Configure > Classes** 或 **Configure > Environments**。
2. 单击右上角的 **Import** 按钮，再选择要从中导入模块的胶囊。您通常可在您的卫星服务器或任何连接的胶囊服务器之间进行选择。
3. 选择要使用左侧的复选框导入的 Puppet 环境。
4. 单击 **更新** 按钮，将 Puppet 环境和类导入到卫星。
5. 导入应生成如下通知：

> *Successfully updated environments and Puppet classes from the on-disk Puppet installation*

第 4 章 创建自定义 PUPPET 环境

您可以在 Satellite 中创建 Puppet 环境。

流程

1. 在 Satellite Web UI 中，导航到 **Configure > Puppet Environments**。
2. 单击 **Create Puppet Environment** 以创建 Puppet 环境。
3. 输入一个名称 、 字母数字字符和下划线，如 **example_environment**。
4. 可选：设置位置上下文。
5. 可选：设置机构上下文。
6. 单击 **Submit** 以创建 Puppet 环境。

请注意，在您将 Puppet 模块导入卫星之前，环境必须已存在为 Puppet 服务器上的 **/etc/puppetlabs/code/environments/example_environment** 文件夹，并包含已安装的 Puppet 模块。

第 5 章 创建 PUPPET 配置组

Puppet 配置组是 Puppet 类的命名列表，允许您组合它们的功能，并在单击时将其分配到受管主机。这等同于纯 Puppet 中的配置文件的概念。

流程

1. 在 Satellite Web UI 中，进入 **Configure > Config Groups**。
2. 单击 **创建配置组** 按钮。
3. 选择您要添加到配置组中的类。
   a. 为 Puppet 配置组选择一个有意义的 **Name**。
   b. 将选定的 Puppet 类添加到 **已包含类** 字段。
4. 点 **Submit** 保存更改。

第 6 章 配置 PUPPET 智能类参数

6.1. PUPPET 参数层次结构

Puppet 参数以层次结构形式进行构建。较低级别的参数覆盖更高等级的参数：

1. 全局参数
2. 机构参数
3. 位置参数
4. 主机组参数
5. 主机参数

例如，特定于主机的参数覆盖了在任何更高级别上的参数，位置参数仅覆盖机构或全局级别上的参数。当您使用位置或机构对主机进行分组时，此功能特别有用。

6.2. 全局覆盖智能类参数

在将 Puppet 类导入到卫星服务器后，您可以配置 Puppet 类。这个示例覆盖了 ntp 服务器的默认列表。

流程

1. 在 Satellite Web UI 中，导航到 **Configure > Classes**。
2. 选择 **ntp** Puppet 类以更改其配置。
3. 选择 **智能类参数** 选项卡并搜索 **服务器**。
4. 确保选中 **覆盖** 复选框。
5. 将参数 **类型** 下拉菜单设置为 数组。
6. 将 ntp 服务器列表插入 **默认值** ：

```
["0.de.pool.ntp.org","1.de.pool.ntp.org","2.de.pool.ntp.org","3.de.pool.ntp.org"]
```

描述数组的替代方法是 **yaml** 语法：

```
- 0.de.pool.ntp.org
- 1.de.pool.ntp.org
- 2.de.pool.ntp.org
- 3.de.pool.ntp.org
```

7. 添加值后，单击**提交按钮**。这会更改 Puppet 模块 **ntp** 的默认配置。

6.3. 为机构覆盖智能类参数

您可以使用主机组来为多个主机覆盖 Puppet 参数。以下示例选择 机构 上下文来演示基于上下文的参数。

请注意，组织级 Puppet 参数会被 位置-level Puppet 参数覆盖。

流程

1. 在 Satellite Web UI 中，导航到 **Configure > Classes**。
2. 单击类名称来选择类。
3. 在 **Smart Class Parameter** 选项卡中，选择一个参数。
4. 使用 **Order** 列表来定义 Puppet 参数的层次结构。单个主机(**fqdn**)标记最相关组织上下文（**机构**）最不相关。
5. 如果在找到第一个匹配项后添加了所有其他匹配参数，请检查 **Merge Overrides**。
6. 如果要同时包含默认值，请检查 **Merge Default**，即使定义了更具体的值。
7. 如果要为所选参数创建唯一值列表，请检查 **Avoid Duplicates**。
8. **matcher** 字段要求 顺序 列表中 的属性类型。
9. 使用 **Add Matcher** 按钮添加更多匹配者。
10. 点 **Submit** 保存更改。

6.4. 为位置覆盖智能类参数

您可以使用主机组来为多个主机覆盖 Puppet 参数。以下示例选择 位置上下文 来说明基于上下文的参数。

流程

1. 在 Satellite Web UI 中，导航到 **Configure > Classes**。
2. 单击类名称来选择类。
3. 在 **Smart Class Parameter** 选项卡中，选择一个参数。
4. 使用 **Order** 列表来定义 Puppet 参数的层次结构。单个主机(**fqdn**)将最和位置（位置 ）标记最相关**位置**。
5. 如果在找到第一个匹配项后添加了所有其他匹配参数，请检查 **Merge Overrides**。
6. 如果要同时包含默认值，请检查 **Merge Default**，即使定义了更具体的值。
7. 如果要为所选参数创建唯一值列表，请检查 **Avoid Duplicates**。
8. **matcher** 字段要求 顺序 列表中 的属性类型。例如，您可以选择 **Paris** 作为 位置上下文，并将值设为 French ntp 服务器。
9. 使用 **Add Matcher** 按钮添加更多匹配者。
10. 点 **Submit** 保存更改。

6.5. 在单个主机上覆盖智能类参数

您可以覆盖单个主机上的参数。如果您有多个主机，且只需要更改一个主机，则建议这样做。

流程

1. 在 Satellite Web UI 中，导航到 **Hosts > All Hosts**。
2. 单击主机名来选择主机。
3. 点 **Edit**。
4. 在 **Host** 选项卡上，选择 Puppet **环境**。
5. 选择 **Puppet ENC** 选项卡。
6. 单击 **Override** 按钮，以编辑 Puppet 参数。
7. 点 **Submit** 保存更改。

第 7 章 将 PUPPET 类分配给主机组

使用主机组将 ntp Puppet 类分配给多个主机。您基于此主机组部署的每个主机都安装了此 Puppet 类。

流程

1. 在 Satellite Web UI 中，导航到 **Configure > Host Groups** 来创建主机组或编辑现有组。
2. 在 **Host Group** 选项卡中，设置以下参数：
   a. **Lifecycle Environment** 描述了主机某些内容版本可用的阶段。
   b. **内容视图** 由产品组成，并允许版本控制内容存储库。
   c. 通过 **环境**，您可以为一组具有自己的专用配置的主机。
3. 导航到 **Puppet ENC** 选项卡。
4. 将 Puppet 类添加到 已包含类，或者在配置了 Puppet 配置组时添加到 已包含的 Config Groups 中。
5. 点 **Submit** 保存更改。

第 8 章 将 PUPPET 类分配给单独的主机

流程

1. 在 Satellite Web UI 中，导航到 **Hosts > All hosts**。
2. 单击 **您要添加 ntp** Puppet 类的主机的编辑按钮。
3. 选择 **Puppet ENC** 选项卡并查找 ntp 类。
4. 单击 **ntp** 旁边的 **+** 符号，将 ntp 子模块 添加到包含类 的列表。
5. 点底部的提交按钮保存您的更改。

   提示

   如果单个主机的 Puppet 类选项卡为空，请检查它是否已分配给正确的 Puppet 环境。

6. 验证 **Puppet 配置。**
   a. **导航到 Hosts >All Hosts 并选择主机。**
   b. **在 top overflow 菜单中，选择 Legacy UI。**
   c. **在详细信息 下，单击 Puppet YAML 按钮。这会生成类似如下的输出：**

   ```
   ---
   parameters:
     // shortened YAML output
   classes:
     ntp:
       servers:
   '["0.de.pool.ntp.org","1.de.pool.ntp.org","2.de.pool.ntp.org","3.de.pool.ntp.org"]'
   environment: production
   ...
   ```

7. 验证 ntp **配置。**
   使用 SSH 连接到您的主机，并检查 /etc/ntp.conf 的内容。

   本例假设您的主机正在运行 CentOS 7。其他操作系统可能会将 ntp 配置文件存储在不同的路径中。

   提示

   您可能需要通过执行以下命令在主机上运行 Puppet 代理：

   ```
   # puppet agent -t
   ```

8. **在主机上运行以下命令，检查哪些 ntp 服务器用于时钟同步：**

   ```
   # cat /etc/ntp.conf
   ```

   这会返回类似如下的输出：

```
# ntp.conf: Managed by puppet.
server 0.de.pool.ntp.org
server 1.de.pool.ntp.org
server 2.de.pool.ntp.org
server 3.de.pool.ntp.org
```

您现在有一个正常工作的 ntp 模块，您可以添加到主机或主机组以自动推出 ntp 配置。

第 9 章 在受管主机上强制 PUPPET 配置

您可以从 Satellite 手动执行配置（运行一次）或以可配置的间隔自动执行。

9.1. 使用 SSH 运行 PUPPET ONCE

为 Run Puppet Once 功能分配正确的作业模板，以便在受管主机上运行 Puppet。

流程

1. 在 Satellite Web UI 中，导航到 Administer > Remote Execution Features。
2. 选择 **puppet_run_host** 远程执行功能。
3. 分配 **Run Puppet Once - SSH Default** 作业模板。

在受管主机上运行 Puppet，并选择类别 **Puppet** 和模板 **Run Puppet Once - SSH Default**。或者，也可在主机详情页面的 Schedule Remote Job 下拉菜单中选择 Run Puppet Once 按钮。

9.2. 了解自动强制设置的间隔

如果最后的 Puppet 报告早于 **outofsync_interval** 和 **puppet_interval** 设置的值，则 Satellite 将主机视为不同步。默认情况下，受管主机上的 Puppet 代理每 30 分钟运行一次，**puppet_interval** 设为 35 分钟，而 global **outofsync_interval** 被设置为 30 分钟。

主机被视为不同步的有效时间是 **outofsync_interval** 和 **puppet_interval** 的总和。例如，将全局 **outofsync _interval** 设置为 30，**puppet_interval** 设为 60 分钟，会在其后将主机状态变为不同步。

9.3. 在主机上设置 PUPPET 代理运行间隔

设置 Puppet 代理运行时的间隔，并将报告发送到 Satellite。

流程

1. 使用 SSH 连接到您的受管主机。
2. 添加 Puppet 代理运行间隔到 **/etc/puppetlabs/puppet/puppet.conf**，如 **runinterval = 1h**。

9.4. 设置全局 OUT-OF-SYNC INTERVAL

流程

1. 在 Satellite Web UI 中，导航到 Administer > Settings。
2. 在 General 选项卡中，编辑 Out of sync interval。设置一个持续时间（以分钟为单位），在哪些主机被视为不同步状态。
   您还可以通过添加 **outofsync_interval** 参数在 *主机组* 或独立 *主机上* 覆盖这个间隔。

9.5. 设置 PUPPET OUT-OF-SYNC INTERVAL

流程

1. 在 Satellite Web UI 中，导航到 Administer > Settings，然后点 Config Management 选项卡。
2. 在 Puppet 间隔 字段中，将值设为持续时间（以分钟为单位），之后使用 Puppet 报告的主机将被视为不同步。

9.6. 为主机组覆盖 OUT-OF-SYNC INTERVAL

流程

1. 在 Satellite Web UI 中，进入 Configure > Host Groups。
2. 选择主机组。
3. 在 Parameters 选项卡中，点 Add Parameter。
4. 在 Name 字段中输入 outofsync_interval。
5. 从 类型 下拉菜单中，选择 整数。
6. 在 Value 字段中输入新间隔（分钟）。
7. 单击 Submit 按钮。

9.7. 为单个主机覆盖 OUT-OF-SYNC INTERVAL

流程

1. 在 Satellite Web UI 中，导航到 Hosts > All Hosts。
2. 点 Edit for a selected host。
3. 在 Parameters 选项卡中，点 Add Parameter。
4. 在 Name 字段中输入 outofsync_interval。
5. 从 类型 下拉菜单中，选择 整数。
6. 在 Value 字段中输入新间隔（分钟）。
7. 单击 Submit 按钮。