Red Hat Satellite 6.15

使用 Puppet 集成管理配置

在 Satellite 中配置 Puppet 集成，并使用 Puppet 类配置您的主机

Last Updated: 2024-06-20

Red Hat Satellite 6.15 使用 Puppet 集成管理配置

在 Satellite 中配置 Puppet 集成，并使用 Puppet 类配置您的主机

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 基础架构中管理的主机上强制实施配置。

目录

使开源包含更多

红帽承诺替换我们的代码、文档和网页属性中存在问题的语言。由于这项工作的艰巨性，这些变化正在尽可能地逐步更新。详情请查看 CTO Chris Wright 的信息。

对红帽文档提供反馈

我们感谢您对我们文档的反馈。让我们了解如何改进它。

使用 Red Hat JIRA 中的 **Create Issue** 表单提供您的反馈。JIRA 问题在 Red Hat Satellite Jira 项目中创建，您可以在其中跟踪其进度。

先决条件

- 确保您已注册了 红帽帐户。

流程

1. 单击以下链接： **创建问题**。如果 Jira 显示登录错误，则登录并在您重定向到表单后继续。
2. 完成 **Summary** 和 **Description** 字段。在 **Description** 字段中，包含文档 URL、章节号以及问题的详细描述。不要修改表单中的任何其他字段。
3. 点 **Create**。

第 1 章 使用 PUPPET 引入配置管理

您可以使用 Puppet 管理和自动化主机配置。Puppet 使用声明性语言来描述 *所需的* 主机状态。

Puppet 提高了您的生产力，因为您可以同时管理多个主机。同时，它会减少您的配置工作，因为 Puppet 可以轻松地验证并更正主机的状态。

其他资源

- 开源 Puppet 文档
- Puppet Forge HBAC-wagona 存储库预构建的 Puppet 模块

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。Satellite 服务器或任何胶囊通常通过 Puppet 服务器部署，Satellite 充当此类 Puppet 服务器的 外部节点分类器(ENC)。主机运行与 Puppet 服务器通信的 Puppet 代理。

Puppet 代理收集有关 *主机的事实*，并在每次运行时将它们报告到 Puppet 服务器。您可以通过在主机上运行 **puppet *事实* 来显示 JSON 格式的 Puppet 事实**。

Puppet 服务器 *将事实转发到* Satellite，再存储它们供以后使用。根据 *事实* 和其他定义，Satellite 构造对 Puppet 服务器的 ENC 回答。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 代理发现目录中定义 *的所需状态* 与实际状态之间的差别（称为 *drifts*），它会强制更正主机的状态。*然后，Puppet 代理将结果报告回 Puppet 服务器，后者将它们报告给 Satellite。*

Puppet 模块

主机 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 模块充当主机配置定义的组件。

智能类参数

如果模块支持使用参数，您可以使用 Smart Class 参数覆盖 Puppet 模块的参数。您可以将 Satellite 中的参数定义为 键值对，这的行为与主机参数或 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版本或管理定义变体，并在将定义部署到生产之前测试它们。

高级集成步骤

Puppet 与 Satellite 集成涉及以下高级别步骤：

1. *启用 Puppet 集成。*
2. *将 Puppet 代理软件包导入到卫星中。Puppet 代理软件包可以像通过 Satellite 的其他内容一样进行管理，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
- 管理主机 指南中的注册主机
- 管理主机指南中的配置和设置 远程作业

以下流程描述了如何使用 Puppet 模块安装、配置和管理 ntp 服务以提供示例。

1.2. 支持的 PUPPET 版本和系统要求

在开始进行 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 Puppet 7。确保用于配置主机的 Puppet 模块与 Puppet 7 兼容。

系统要求

在开始将 Puppet 与 Satellite 集成之前，请确保满足系统要求。有关更多信息，请参阅 开源 Puppet 文档中的 Puppet 7 的系统要求。

1.3. 启用 PUPPET 与 SATELLITE 集成

默认情况下，Satellite 没有配置任何 Puppet 集成。您需要根据您的具体情况启用集成。这意味着，您可以配置 Satellite，以在 Satellite 服务器或 Capsule 服务器上管理和部署 Puppet 服务器。另外，您可以在外部将 Puppet 服务器部署到 Satellite，并将其与 Satellite 集成，以报告、事实和外部节点分类(ENC)。

流程

1. 启用 Puppet 集成并在 Satellite 服务器上安装 Puppet 服务器：

```
# satellite-installer \
--enable-foreman-cli-puppet \
--enable-foreman-plugin-puppet \
--enable-puppet \
--foreman-proxy-puppet true \
--foreman-proxy-puppetca true \
--puppet-server true
```

2. 如果要在 Capsule 服务器上使用 Puppet 集成，请启用 Puppet 集成并在 Capsule 服务器上安装 Puppet 服务器：

```
# satellite-installer \
--enable-puppet \
--foreman-proxy-puppet true \
--foreman-proxy-puppetca true \
--puppet-server true
```

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 Puppet 与 Satellite 集成”。
- 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 激活码 。

流程

1. 导航到 **Hosts** > **Templates** > **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 中的置备模板的 Kind 。
3. 确保 **puppet_setup** 片断包含如下：

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

请注意，此片段已包含在 Satellite 附带的模板中，如 **Kickstart default** 或 **Preseed 默认**。

4. 使用全局参数、主机组或单个主机中的 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**。
5. 为 Puppet 代理设置配置。
   - 如果您使用集成的 Puppet 服务器，请确保在创建主机时选择 Puppet Capsule、Puppet CA Capsule 和 Puppet 环境。
   - 如果您使用非集成 Puppet 服务器，请在全局参数或主机组中设置以下主机参数，或者在创建主机时设置：
     - 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。
6. 确保您的主机可以使用适当的激活密钥从卫星服务器访问 Puppet 代理软件包。

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 Puppet 与 Satellite 集成”。

- 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 激活码 。

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 **Actions** 列中的列表中，选择 **Certificates**。
7. 单击所需主机右侧的 **Sign**，为 Puppet 代理签署 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 Puppet 与 Satellite 集成”。
- 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，进入到 **Infrastructure** > **Capsules**。
8. 从所需胶囊服务器的 **Actions** 列中的列表中，选择 **Certificates**。
9. 单击所需主机右侧的 **Sign**，为 Puppet 代理签署 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，您可以使用 Puppet 开始执行配置管理。这涉及以下高级别步骤：

1. 在 Puppet 服务器上管理 Puppet 模块，该模块正在安装和更新它们。
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覆盖。
5. 将 Puppet 类或配置组分配到主机组或个别主机。
6. 配置用于在主机上运行 Puppet 代理的间隔，以及 Puppet 服务器的配置执行运行。

7. 在 Satellite Web UI 中使用报告监控配置管理。如需更多信息，请参阅管理 Red Hat Satellite 中的监控资源。

8. 配置电子邮件通知.如需更多信息，请参阅管理 Red Hat Satellite 中的配置电子邮件通知 首选项。

在分配 Puppet 类或配置组后，Satellite 会在配置的间隔中自动运行配置管理，以在主机上强制实施 Puppet 配置，或者您可以使用 **Run Puppet Once** 功能手动启动它。如需更多信息，请参阅 第 9.1 节 "使用 SSH 运行一次 Puppet"。

1.8. 禁用 PUPPET 与 SATELLITE 的集成

要停止在 Satellite 中使用 Puppet，请按照以下步骤执行。

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。

先决条件

- Puppet 在 Satellite 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 服务器上禁用 Puppet 服务器：

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

第 2 章 管理 PUPPET 模块

2.1. 在 SATELLITE 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 是提供由社区贡献的 Puppet 模块的存储库。Puppet 模块标记为 受支持，Puppet Inc 已正式支持和测试。

本例演示了如何将 ntp 模块添加到 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 服务器并安装 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 模块的一种替代方法是，将包含 Puppet 模块的文件夹复制到上述模块路径。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. 如果该模块位于上面显示的路径中，以下命令会更新一个模块：

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

第 3 章 将 PUPPET 类和环境导入到 SATELLITE 中

在将任何类分配给主机之前，将 Puppet 类和环境从安装的 Puppet 模块导入到 Satellite 服务器或任何附加的胶囊服务器。

先决条件

- 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，进入到 **Configure** > **Puppet ENC** > **Classes** 或 **Configure** > **Puppet ENC** > **Environments**。
2. 单击右上角的 **Import**，再选择您要从中导入模块的胶囊。您通常可能会在 Satellite 服务器或任何附加的胶囊服务器之间进行选择。
3. 选中 Puppet 环境，以使用左侧的复选框导入。
4. 单击 **Update**，将 Puppet 环境和类导入到 Satellite。
5. 导入会导致通知，如下所示：

> 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，进入到 **Configure > Puppet Environments**。
2. 单击 **Create Puppet Environment** 以创建 Puppet 环境。
3. 允许输入 **Name**，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，在将 Puppet 模块导入到 Satellite 之前，环境必须已作为 Puppet 服务器上的 **/etc/puppetlabs/code/environments/example_environment** 存在，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 类的指定列表，允许您组合其功能并将它们分配到单击时的主机。这等同于纯 Puppet 中配置文件的概念。

流程

1. 在 Satellite Web UI 中，进入到 **Configure** > **Puppet ENC** > **Config Groups**。
2. 单击 **Create Config Group**。
3. 选择您要添加到 config 组的类。
   a. 为 Puppet 配置组选择一个有意义的名称。
   b. **将所选的 Puppet 类添加到 已包含类 字段。**
4. **点 Submit 以保存更改。**

第 6 章 配置 PUPPET 智能类参数

6.1. PUPPET 参数层次结构

Puppet 参数的结构化为层次结构。较低级别的参数覆盖更高级别的参数：

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，特定于主机的参数在任何更高级别上覆盖参数，位置参数仅覆盖组织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

您可以将 Puppet 类导入到 Satellite 服务器后，您可以配置 Puppet 类。这个示例覆盖 ntp 服务器的默认列表。

流程

1. 在 Satellite Web UI 中，进入到 Configure > Puppet ENC > Classes。
2. 选择 ntp Puppet 类以更改其配置。
3. 选择 Smart Class Parameter 选项卡并搜索服务器。
4. 确保已选中 Override 复选框。
5. 将 Parameter Type 下拉菜单设置为 数组。
6. 将 ntp 服务器列表作为默认值 插入 ：

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

   描述阵列的替代方法是 yaml 语法：

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

7. 单击 Submit 以更改 Puppet 模块ntp 的默认配置。

6.3. 覆盖机构的智能类参数

您可以使用主机组一次性覆盖多个主机的 Puppet 参数。以下示例选择 组织上下文 来说明基于上下文的参数。

请注意，组织- 级别的 Puppet 参数会被location-level Puppet 参数覆盖。

流程

1. 在 Satellite Web UI 中，进入到 Configure > Puppet ENC > Classes。
2. 点类名称来选择类。
3. 在 Smart Class Parameter 选项卡上，选择一个参数。
4. 使用 Order 列表定义 Puppet 参数的层次结构。单个主机(fqdn)将大多数和组织上下文（组织）标记为最不相关的内容。
5. 如果要在找到第一个匹配项后添加所有进一步匹配的参数，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 字段需要顺序列表中的属性类型。
9. 可选：点 Add Matcher 来添加更多匹配项。
10. 点 Submit 以保存更改。

6.4. 覆盖位置的智能类参数

您可以使用主机组一次性覆盖多个主机的 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，进入到 Configure > Puppet ENC > Classes。
2. 点类名称来选择类。
3. 在 Smart Class Parameter 选项卡上，选择一个参数。
4. 使用 Order 列表定义 Puppet 参数的层次结构。单个主机(fqdn)标记与最相关的位置上下文（位置）。
5. 如果要在找到第一个匹配项后添加所有进一步匹配的参数，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 字段需要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，并将值设为法语 ntp 服务器。
9. 可选：点 Add Matcher 来添加更多匹配项。
10. 点 Submit 以保存更改。

6.5. 覆盖单个主机上的智能类参数

您可以覆盖单个主机上的参数。如果您有多个主机，并且只想更改单个主机，则建议这样做。

流程

1. 在 Satellite Web UI 中，进入到 Hosts > All Hosts。
2. 单击主机名以选择主机。
3. 点 Edit。
4. 在 Host 选项卡中，选择 PuppetEnvironment。
5. 选择 Puppet ENC 选项卡。
6. 单击 Override 以编辑 Puppet 参数。
7. 点 Submit 以保存更改。

第 7 章 将 PUPPET 类分配给主机组

使用主机组将 ntp Puppet 类分配到多个主机。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，进入到 Configure > Host Groups 以创建主机组或编辑现有主机组。
2. 在 Host Group 选项卡中设置以下参数：
   a. 生命周期环境 描述了某些内容可用于主机的阶段。
   b. 内容视图 由产品组成，允许版本控制内容存储库。
   c. 该环境 允许您为一组主机提供自己的专用配置。
3. 导航到 Puppet ENC 选项卡。
4. 如果配置了 Puppet 配置组，请将 Puppet 类添加到已包含类或已包含配置组。
5. 点 Submit 以保存更改。

第 8 章 将 PUPPET 类分配给单个主机

流程

1. 在 Satellite Web UI 中，进入到 Hosts > All Hosts。
2. 找到您要添加 ntp Puppet 类的主机，然后单击 Edit。
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 Submit 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. 进入 Hosts > All Hosts 并选择主机。
   b. 从顶部溢出菜单中选择 Legacy UI。
   c. 在 Details 下，单击 Puppet YAML。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。

   本例假定您的主机正在运行 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来在主机上运行 Puppet 代理：

   ```
   # puppet agent -t
   ```

8. 在主机上运行以下命令检查哪个 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
```

现在，您有一个可正常工作的 ntp 模块，您可以添加到主机或主机组中，以自动推出 ntp 配置。

第 9 章 在主机上强制 PUPPET 配置

您可以从 Satellite 手动根据需要（运行一次）强制配置，或者以可配置的间隔自动执行。

9.1. 使用 SSH 运行一次 PUPPET

分配正确的作业模板到 Run Puppet Once 功能，以便在主机上运行 Puppet。

流程

1. 在 Satellite Web UI 中，进入到 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。

在主机上运行 Puppet，方法是运行作业并选择类别 **Puppet** 和模板 **Run Puppet Once - SSH Default**。或者，也可在主机详情页面上的 Schedule Remote Job 下拉菜单中，单击 Run Puppet Once。

9.2. 了解自动强制的间隔

如果最后的 Puppet 报告比 **outofsync_interval** 和 **puppet_interval** 的合并值（以分钟为单位）的合并值，则 Satellite 会将主机视为不同步。默认情况下，主机上的 Puppet 代理每 30 分钟运行一次，**puppet_interval** 设为 35 分钟，全局 **outofsync_interval** 设为 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 global **outofsync _interval** 设置为 30，**puppet_interval** 设为 60 分钟会导致主机状态变为不同步的 90 分钟。

9.3. 在主机上设置 PUPPET 代理运行间隔

设置 Puppet 代理运行并向 Satellite 发送报告的时间间隔。

流程

1. 使用 SSH 连接到您的主机。
2. 将 Puppet 代理运行间隔添加到 **/etc/puppetlabs/puppet/puppet.conf**，如 **runinterval = 1h**。

9.4. 设置全局不同步间隔

流程

1. 在 Satellite Web UI 中，进入到 Administer > Settings。
2. 在 General 选项卡中，编辑 不同步间隔。设置持续时间（以分钟为单位），之后主机被视为不同步。
   您还可以通过添加 **outofsync_interval** 参数来覆盖 主机组 或 单个主机上 此间隔。

9.5. 设置 PUPPET 不同步间隔

流程

1. 在 Satellite Web UI 中，进入到 Administer > Settings，然后点击 Config Management 选项卡。
2. 在 Puppet interval 字段中，将值设置为持续时间（以分钟为单位），然后报告使用 Puppet 的主机被视为不同步。

9.6. 覆盖主机组的同步间隔

流程

1. 在 Satellite Web UI 中，进入到 Configure > Host Groups。
2. 选择主机组。
3. 在 Parameters 选项卡上，单击 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，以分钟为单位输入新间隔。
7. 点 Submit。

9.7. 为单个主机覆盖同步间隔

流程

1. 在 Satellite Web UI 中，进入到 Hosts > All Hosts。
2. 为所选主机点 Edit。
3. 在 Parameters 选项卡上，单击 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，以分钟为单位输入新间隔。
7. 点 Submit。