

## Ansible Plugin - Story #3363

### As a user, I can pull in an Ansible Role from any git repository

02/13/2018 03:35 PM - daviddavis

<b>Status:</b> NEW	<b>Start date:</b>
<b>Priority:</b> Normal	<b>Due date:</b>
<b>Assignee:</b>	<b>% Done:</b> 0%
<b>Category:</b>	<b>Estimated time:</b> 0:00 hour
<b>Sprint/Milestone:</b>	<b>Tags:</b>
<b>Platform Release:</b>	<b>Sprint:</b>
<b>Groomed:</b> No	<b>Quarter:</b>
<b>Sprint Candidate:</b> No	
<b>Description</b>	

#### History

##### #1 - 02/13/2018 03:36 PM - daviddavis

- Subject changed from As a user, I can pull in an Ansible role from any git repository to As a user, I can pull in an Ansible Role from any git repository

##### #2 - 02/13/2018 03:37 PM - daviddavis

There's been some requests by the community asking that we support this but I have no firm ideas on how. I've opened this issue to hopefully solicit feedback on how this should work from a user perspective and how we should implement this.

##### #3 - 02/13/2018 04:19 PM - daviddavis

- Tracker changed from Issue to Story

- % Done set to 0

##### #4 - 07/15/2019 11:19 PM - daviddavis

Here was some initial work I did:

```
commit df05cfb16abce7999bdc413092df2164ed1f7e9f
Author: David Davis <daviddavis@redhat.com>
Date: Thu May 3 13:04:10 2018 -0400
```

```
[WIP] Add git remote
```

```
diff --git a/pulp_ansible/app/models.py b/pulp_ansible/app/models.py
index 8d726ac..6a4f07f 100644
--- a/pulp_ansible/app/models.py
+++ b/pulp_ansible/app/models.py
@@ -68,9 +68,17 @@ class AnsiblePublisher(Publisher):
     TYPE = 'ansible'
```

```
-class AnsibleRemote(Remote):
+class AnsibleGalaxyRemote(Remote):
+    """
+    A Remote for Ansible content
+    A Remote for Ansible Galaxy
+    """
+
+    TYPE = 'ansible'
+    TYPE = 'ansible-galaxy'
+
+class AnsibleGitRemote(Remote):
+    """
+    A Remote for Ansible content in git
```

```

+ """
+
+ TYPE = 'ansible-git'
diff --git a/pulp_ansible/app/serializers.py b/pulp_ansible/app/serializers.py
index ee084d0..9eef0f 100644
--- a/pulp_ansible/app/serializers.py
+++ b/pulp_ansible/app/serializers.py
@@ -3,7 +3,7 @@ from rest_framework import serializers
     from pulpcore.plugin.serializers import ContentSerializer, RemoteSerializer, PublisherSerializer
     from pulpcore.plugin.models import Artifact

-from .models import AnsibleRemote, AnsiblePublisher, AnsibleRole, AnsibleRoleVersion
+from .models import AnsibleGalaxyRemote, AnsibleGitRemote, AnsiblePublisher, AnsibleRole, AnsibleRoleVersion

     from rest_framework_nested.relations import NestedHyperlinkedIdentityField

@@ -46,12 +46,17 @@ class AnsibleRoleVersionSerializer(ContentSerializer):
     model = AnsibleRoleVersion

-class AnsibleRemoteSerializer(RemoteSerializer):
+class AnsibleGalaxyRemoteSerializer(RemoteSerializer):
     class Meta:
         fields = RemoteSerializer.Meta.fields
-        model = AnsibleRemote
+        model = AnsibleGalaxyRemote

+class AnsibleGitRemoteSerializer(RemoteSerializer):
+    class Meta:
+        fields = RemoteSerializer.Meta.fields
+        model = AnsibleGitRemote
+
     class AnsiblePublisherSerializer(PublisherSerializer):
         class Meta:
             fields = PublisherSerializer.Meta.fields
diff --git a/pulp_ansible/app/tasks/__init__.py b/pulp_ansible/app/tasks/__init__.py
index 67e6aac..9e3f50d 100644
--- a/pulp_ansible/app/tasks/__init__.py
+++ b/pulp_ansible/app/tasks/__init__.py
@@ -1,2 +1,2 @@
-from .synchronizing import synchronize # noqa
+from .synchronizing import galaxy # noqa
     from .publishing import publish # noqa
diff --git a/pulp_ansible/app/tasks/synchronizing/__init__.py b/pulp_ansible/app/tasks/synchronizing/__init__.py
new file mode 100644
index 0000000..62e4b74
--- /dev/null
+++ b/pulp_ansible/app/tasks/synchronizing/__init__.py
@@ -0,0 +1 @@
+from .galaxy import synchronize # noqa
diff --git a/pulp_ansible/app/tasks/synchronizing.py b/pulp_ansible/app/tasks/synchronizing/galaxy.py
similarity index 98%
rename from pulp_ansible/app/tasks/synchronizing.py
rename to pulp_ansible/app/tasks/synchronizing/galaxy.py
index 1c8c2ae..c63573b 100644
--- a/pulp_ansible/app/tasks/synchronizing.py
+++ b/pulp_ansible/app/tasks/synchronizing/galaxy.py
@@ -20,7 +20,7 @@ from pulpcore.plugin.changeset import (
     SizedIterable)
     from pulpcore.plugin.tasking import UserFacingTask, WorkingDirectory

-from pulp_ansible.app.models import AnsibleRole, AnsibleRoleVersion, AnsibleRemote
+from pulp_ansible.app.models import AnsibleRole, AnsibleRoleVersion, AnsibleGalaxyRemote

     log = logging.getLogger(__name__)
@@ -49,7 +49,7 @@ def synchronize(remote_pk, repository_pk):
     Raises:
         ValueError: When remote has no url specified.
     """
-    remote = AnsibleRemote.objects.get(pk=remote_pk)
+    remote = AnsibleGalaxyRemote.objects.get(pk=remote_pk)
     repository = Repository.objects.get(pk=repository_pk)

```

```

base_version = RepositoryVersion.latest(repository)

@@ -114,7 +114,7 @@ def fetch_roles(remote):
    Fetch the roles in a remote repository

    Args:
-     remote (AnsibleRemote): A remote.
+     remote (AnsibleGalaxyRemote): A remote.

    Returns:
        list: a list of dicts that represent roles
@@ -228,7 +228,7 @@ def build_additions(remote, roles, delta):
    Build the content to be added.

    Args:
-     remote (AnsibleRemote): A remote.
+     remote (AnsibleGalaxyRemote): A remote.
+     roles (list): The list of role dict from Galaxy
+     delta (Delta): The set of Key to be added and removed.

diff --git a/pulp_ansible/app/tasks/synchronizing/git.py b/pulp_ansible/app/tasks/synchronizing/git.py
new file mode 100644
index 0000000..79f4fe2
--- /dev/null
+++ b/pulp_ansible/app/tasks/synchronizing/git.py
@@ -0,0 +1,225 @@
+import logging
+import os
+import tarfile
+
+from collections import namedtuple
+from concurrent.futures import FIRST_COMPLETED
+from contextlib import suppress
+from gettext import gettext as _
+from urllib.parse import urlparse, urlencode, parse_qs
+
+import asyncio
+from celery import shared_task
+from django.db.models import Q
+from git import Repo
+
+from pulpcore.plugin.models import Artifact, RepositoryVersion, Repository, ProgressBar
+from pulpcore.plugin.changeset import (
+    BatchIterator,
+    ChangeSet,
+    PendingArtifact,
+    PendingContent,
+    SizedIterable)
+from pulpcore.plugin.tasking import UserFacingTask, WorkingDirectory
+
+from pulp_ansible.app.models import AnsibleRole, AnsibleRoleVersion, AnsibleGitRemote
+
+log = logging.getLogger(__name__)
+
+# The natural key.
+Key = namedtuple('Key', ('namespace', 'name', 'version'))
+
+# The set of Key to be added and removed.
+Delta = namedtuple('Delta', ('additions', 'removals'))
+
+@shared_task(base=UserFacingTask)
+def synchronize(remote_pk, repository_pk, role_pk):
+    """
+    Create a new version of the repository that is synchronized with the remote
+    as specified by the remote.
+
+    Args:
+        remote_pk (str): The remote PK.
+        repository_pk (str): The repository PK.
+
+    Raises:
+        ValueError: When remote has no url specified.

```

```

+ """
+ remote = AnsibleGitRemote.objects.get(pk=remote_pk)
+ repository = Repository.objects.get(pk=repository_pk)
+ role = AnsibleRole.objects.get(pk=role_pk)
+ base_version = RepositoryVersion.latest(repository)
+
+ if not remote.url:
+     raise ValueError(_('A remote must have a url specified to synchronize.'))
+
+ with WorkingDirectory() as working_dir:
+     with RepositoryVersion.create(repository) as new_version:
+         log.info(
+             _('Synchronizing: repository=%(r)s remote=%(p)s'),
+             {
+                 'r': repository.name,
+                 'p': remote.name
+             })
+         versions = fetch_role_versions(remote, os.join(working_dir, remote.id), role)
+         content = fetch_content(base_version)
+         delta = find_delta(versions, content)
+         additions = build_additions(remote, versions, delta)
+         removals = build_removals(base_version, delta)
+         changeset = ChangeSet(
+             remote=remote,
+             repository_version=new_version,
+             additions=additions,
+             removals=removals)
+         for report in changeset.apply():
+             if not log.isEnabledFor(logging.DEBUG):
+                 continue
+             log.debug(
+                 _('Applied: repository=%(r)s remote=%(p)s change:%(c)s'),
+                 {
+                     'r': repository.name,
+                     'p': remote.name,
+                     'c': report,
+                 })
+
+
+def fetch_role_versions(remote, working_dir, role):
+    """
+    Fetch the role versions in a remote git repository
+
+    Args:
+        remote (AnsibleGitRemote): A remote.
+
+    Returns:
+        list: a list of dicts that represent roles
+    """
+    repo = Repo.clone_from(remote.url, working_dir)
+    tags = repo.tags
+    versions = set()
+
+    progress_bar = ProgressBar(message='Fetching and parsing git repo', total=len(tags),
+                               done=0, state='running')
+    progress_bar.save()
+
+    for tag in tags:
+        repo.heads.tag.checkout()
+        versions.append(Key(name=role.name, namespace=role.namespace, version=tag.name))
+        with tarfile.open(tag.name, "w:gz") as tar:
+            tar.add(working_dir, arcname=os.path.basename(working_dir))
+        progress_bar.increment()
+
+    progress_bar.state = 'completed'
+    progress_bar.save()
+
+    return versions
+
+
+def fetch_content(base_version):
+    """
+    Fetch the AnsibleRoleVersions contained in the (base) repository version.
+
+    Args:

```

```

+     base_version (RepositoryVersion): A repository version.
+
+ Returns:
+     set: A set of Key contained in the (base) repository version.
+ """
+     content = set()
+     if base_version:
+         for role_version in AnsibleRoleVersion.objects.filter(pk__in=base_version.content):
+             key = Key(name=role_version.role.name, namespace=role_version.role.namespace,
+                       version=role_version.version)
+             content.add(key)
+     return content
+
+
+def find_delta(role_versions, content, mirror=True):
+    """
+    Find the content that needs to be added and removed.
+
+    Args:
+        roles (list): A list of roles from a remote repository
+        content: (set): The set of natural keys for content contained in the (base)
+            repository version.
+        mirror (bool): The delta should include changes needed to ensure the content
+            contained within the pulp repository is exactly the same as the
+            content contained within the remote repository.
+
+    Returns:
+        Delta: The set of Key to be added and removed.
+    """
+    remote_content = set()
+    additions = (role_versions - content)
+    if mirror:
+        removals = (content - remote_content)
+    else:
+        removals = set()
+    return Delta(additions, removals)
+
+def build_additions(remote, roles, delta):
+    """
+    Build the content to be added.
+
+    Args:
+        remote (AnsibleGitRemote): A remote.
+        roles (list): The list of role dict from Git
+        delta (Delta): The set of Key to be added and removed.
+
+    Returns:
+        SizedIterable: The PendingContent to be added to the repository.
+    """
+    pass
+    # def generate():
+    #     # for metadata in roles:
+    #         # role, _ = AnsibleRole.objects.get_or_create(name=metadata['name'],
+    #                                                       # namespace=metadata['namespace'])
+
+    #         # for version in metadata['summary_fields']['versions']:
+    #             # key = Key(name=metadata['name'],
+    #                       # namespace=metadata['namespace'],
+    #                       # version=version['name'])
+
+    #             # if key not in delta.additions:
+    #                 # continue
+
+    #             # url = GITHUB_URL % (metadata['github_user'], metadata['github_repo'],
+    #                                   # version['name'])
+    #             # role_version = AnsibleRoleVersion(version=version['name'], role=role)
+    #             # path = "%s/%s/%s.tar.gz" % (metadata['namespace'], metadata['name'],
+    #                                           # version['name'])
+
+    #             # artifact = Artifact()
+    #             # content = PendingContent(
+    #                 # role_version,
+    #                 # artifacts={
+    #                     # PendingArtifact(artifact, url, path)
+    #                 # })

```

```

+         # yield content
+     # return SizedIterable(generate(), len(delta.additions))
+
+
+def build_removals(base_version, delta):
+    """
+    Build the content to be removed.
+
+    Args:
+        base_version (RepositoryVersion): The base repository version.
+        delta (Delta): The set of Key to be added and removed.
+
+    Returns:
+        SizedIterable: The AnsibleRoleVersion to be removed from the repository.
+    """
+    def generate():
+        for removals in BatchIterator(delta.removals):
+            q = Q()
+            for key in removals:
+                role = AnsibleRoleVersion.objects.get(name=key.name, namespace=key.namespace)
+                q |= Q(ansibleroleversion__role_id=role.pk, ansibleversion__version=key.version)
+            q_set = base_version.content.filter(q)
+            q_set = q_set.only('id')
+            for file in q_set:
+                yield file
+        return SizedIterable(generate(), len(delta.removals))
diff --git a/pulp_ansible/app/viewsets.py b/pulp_ansible/app/viewsets.py
index 82ac625..cb3b6f3 100644
--- a/pulp_ansible/app/viewsets.py
+++ b/pulp_ansible/app/viewsets.py
@@ -14,9 +14,11 @@ from pulpcore.plugin.viewsets import (
     PublisherViewSet)

 from . import tasks
-from .models import AnsibleRemote, AnsiblePublisher, AnsibleRole, AnsibleRoleVersion
-from .serializers import (AnsibleRemoteSerializer, AnsiblePublisherSerializer,
-                          AnsibleRoleSerializer, AnsibleRoleVersionSerializer)
+from .models import (AnsibleGalaxyRemote, AnsibleGitRemote, AnsiblePublisher, AnsibleRole,
+                    AnsibleRoleVersion)
+from .serializers import (AnsibleGalaxyRemoteSerializer, AnsibleGitRemoteSerializer,
+                          AnsiblePublisherSerializer, AnsibleRoleSerializer,
+                          AnsibleRoleVersionSerializer)

 class AnsibleRoleFilter(filterset.FilterSet):
@@ -83,18 +85,15 @@ class AnsibleRoleVersionViewSet(ContentViewSet):
     headers=headers)

-class AnsibleRemoteViewSet(RemoteViewSet):
-    endpoint_name = 'ansible'
-    queryset = AnsibleRemote.objects.all()
-    serializer_class = AnsibleRemoteSerializer
+class AnsibleGalaxyRemoteViewSet(RemoteViewSet):
+    endpoint_name = 'ansible/galaxy'

     @detail_route(methods=('post',))
     def sync(self, request, pk):
         remote = self.get_object()
         repository = self.get_resource(request.data['repository'], Repository)
-        if not remote.url:
-            raise serializers.ValidationError(detail=_('A url must be specified.'))
-        result = tasks.synchronize.apply_async_with_reservation(
+
+        result = tasks.galaxy.synchronize.apply_async_with_reservation(
             [repository, remote],
             kwargs={
                 'remote_pk': remote.pk,
@@ -104,6 +103,28 @@ class AnsibleRemoteViewSet(RemoteViewSet):
         return OperationPostponedResponse(result, request)

+class AnsibleGitRemoteViewSet(RemoteViewSet):
+    endpoint_name = 'ansible/git'
+    queryset = AnsibleGitRemote.objects.all()

```

```

+     serializer_class = AnsibleGitRemoteSerializer
+
+     @detail_route(methods=('post',))
+     def sync(self, request, pk):
+         remote = self.get_object()
+         repository = self.get_resource(request.data['repository'], Repository)
+         role = self.get_resource(request.data['role'], AnsibleRole)
+
+         result = tasks.galaxy.synchronize.apply_async_with_reservation(
+             [repository, remote],
+             kwargs={
+                 'remote_pk': remote.pk,
+                 'repository_pk': repository.pk,
+                 'role_pk': role.pk
+             }
+         )
+         return OperationPostponedResponse(result, request)
+
+
+ class AnsiblePublisherViewSet (PublisherViewSet):
+     endpoint_name = 'ansible'
+     queryset = AnsiblePublisher.objects.all()

```