

Ansible Dynamic Library

Github Repo : <https://github.com/PatchMon/PatchMon-ansible/tree/main>

A dynamic inventory plugin for Ansible that queries the PatchMon HTTP JSON API and exposes hosts as an Ansible inventory.

Description

The `dynamic_inventory` plugin allows you to use PatchMon as a dynamic inventory source for Ansible. It queries the PatchMon API to retrieve host information including hostnames, IP addresses, and group assignments, and automatically generates an Ansible inventory.

Requirements

- **Ansible:** `>= 2.19.0`
- **Python:** `3.6+`
- **Dependencies:** `requests >= 2.25.1`

Installation

Install from Ansible Galaxy

```
ansible-galaxy collection install patchmon.dynamic_inventory
```

Install from Source

1. Clone the repository:

```
git clone https://github.com/PatchMon/PatchMon-ansible.git
cd PatchMon-ansible/patchmon/dynamic_inventory
```

2. Build the collection:

```
ansible-galaxy collection build
```

3. Install the collection:

```
ansible-galaxy collection install patchmon-dynamic_inventory-*.tar.gz
```

4. Install dependencies:

```
pip install -r requirements.txt
```

Configuration

Create an inventory configuration file (e.g., `patchmon_inventory.yml`):

```
---
plugin: patchmon.dynamic_inventory
api_url: http://localhost:3000/api/v1/api/hosts/
api_key: your_api_key
api_secret: your_api_secret
verify_ssl: false
```

Configuration Options

Option	Description	Required	Default
<code>plugin</code>	Name of the plugin	<input type="checkbox"/>	<code>patchmon.dynamic_inventory</code>
<code>api_url</code>	URL of the PatchMon API endpoint that returns JSON host data	<input type="checkbox"/>	—
<code>api_key</code>	API key for authentication	<input type="checkbox"/>	—
<code>api_secret</code>	API secret for authentication	<input type="checkbox"/>	—
<code>verify_ssl</code>	Whether to verify SSL certificates when contacting the API	<input type="checkbox"/>	<code>true</code>

Usage

Basic Usage

Run Ansible commands with the inventory file:

```
# List all hosts
ansible-inventory -i patchmon_inventory.yml --list

# Ping all hosts
ansible all -i patchmon_inventory.yml -m ping

# Run a playbook
ansible-playbook -i patchmon_inventory.yml playbook.yml
```

Configure as Default Inventory

Add to your `ansible.cfg`:

```
[defaults]
inventory = patchmon_inventory.yml

[inventory]
enable_plugins = patchmon.dynamic_inventory.dynamic_inventory
```

Using in Playbooks

Create a playbook (e.g., `ping.yml`):

```
---
- name: Test connectivity to all hosts
  hosts: all
  gather_facts: no
  tasks:
    - name: Ping hosts
      ansible.builtin.ping:
```

Run the playbook:

```
ansible-playbook ping.yml
```

API Response Format

The plugin expects the PatchMon API to return JSON in the following format:

```

{
  "hosts": [
    {
      "hostname": "server1.example.com",
      "ip": "192.168.1.10",
      "host_groups": [
        {
          "name": "web_servers"
        },
        {
          "name": "production"
        }
      ]
    },
    {
      "hostname": "server2.example.com",
      "ip": "192.168.1.11",
      "host_groups": [
        {
          "name": "db_servers"
        },
        {
          "name": "production"
        }
      ]
    }
  ]
}

```

Inventory Mapping

- **Hostname:** The `hostname` field is used as the Ansible host name
- **IP Address:** The `ip` field is mapped to the `ansible_host` variable
- **Groups:** Each entry in `host_groups` creates an Ansible group, and hosts are assigned to these groups

Examples

Example 1: List Inventory

```
ansible-inventory -i patchmon_inventory.yml --list
```

Output:

```
{
  "_meta": {
    "hostvars": {
      "server1.example.com": {
        "ansible_host": "192.168.1.10"
      },
      "server2.example.com": {
        "ansible_host": "192.168.1.11"
      }
    }
  },
  "all": {
    "children": [
      "ungrouped",
      "web_servers",
      "db_servers",
      "production"
    ]
  },
  "db_servers": {
    "hosts": [
      "server2.example.com"
    ]
  },
  "production": {
    "hosts": [
      "server1.example.com",
      "server2.example.com"
    ]
  },
  "web_servers": {
    "hosts": [
      "server1.example.com"
    ]
  }
}
```

```
    ]
  }
}
```

Example 2: Target Specific Groups

```
# Run on web servers only
ansible-playbook -i patchmon_inventory.yml playbook.yml --limit web_servers

# Run on production hosts only
ansible-playbook -i patchmon_inventory.yml playbook.yml --limit production
```

Example 3: Using Environment Variables

For security, you can use Ansible vault or environment variables:

```
---
plugin: patchmon.dynamic_inventory
api_url: http://localhost:3000/api/v1/api/hosts/
api_key: "{{ lookup('env', 'PATCHMON_API_KEY') }}"
api_secret: "{{ lookup('env', 'PATCHMON_API_SECRET') }}"
verify_ssl: false
```

Authentication

The plugin uses HTTP Basic Authentication with the provided `api_key` and `api_secret`. Make sure these credentials have the necessary permissions to query the PatchMon API.

SSL Verification

By default, SSL certificate verification is enabled (`verify_ssl: true`). For development or self-signed certificates, you can disable it by setting `verify_ssl: false`. **Note:** Disabling SSL verification is not recommended for production environments.

Troubleshooting

Test API Connectivity

```
# Test the API endpoint directly
curl -u "api_key:api_secret" http://localhost:3000/api/v1/api/hosts/
```

Debug Inventory

```
# Show detailed inventory information
ansible-inventory -i patchmon_inventory.yml --list --debug

# Test with verbose output
ansible-inventory -i patchmon_inventory.yml --list -v
```

Common Issues

1. **Authentication Errors:** Verify that your `api_key` and `api_secret` are correct
2. **Connection Errors:** Check that the `api_url` is accessible and the API is running
3. **JSON Parsing Errors:** Ensure the API returns valid JSON in the expected format
4. **Missing Hosts:** Verify that the API response contains a `hosts` array

Development

Testing

Test the plugin locally:

```
# Test inventory parsing
ansible-inventory -i patchmon_inventory.yml --list

# Test with a playbook
ansible-playbook -i patchmon_inventory.yml ping.yml
```

Contributing

Contributions are welcome! Please follow these steps:

1. Fork the repository

2. Create a feature branch
3. Make your changes
4. Submit a pull request

License

AGPL-3.0-or-later

See the [LICENSE](#) file for details.

Authors

- Steve Libonati stevelibonati@yahoo.com

Links

- **Repository:** <https://github.com/PatchMon/PatchMon-ansible>
- **Issues:** <https://github.com/PatchMon/PatchMon-ansible/issues>

Revision #1

Created 2026-02-12 01:08:31 UTC by lby

Updated 2026-02-12 01:09:36 UTC by lby