Troubleshooting Ansible: Common issues and how to resolve them

Ansible is a powerful automation tool, but users may encounter various issues while working with it. Here are some common issues and their resolutions:

1. SSH Connectivity Issues

Description:

Problems connecting to remote hosts over SSH.

Symptoms:

- Connection timeouts
- Permission denied errors

Resolution:

- Ensure target machines are reachable via SSH.
- Verify SSH key configuration and permissions.
- Check `ansible_user` and `ansible_ssh_private_key_file` in inventory.
- Add SSH key to `ssh-agent` if using agent forwarding.

Code:

Shell

```
ssh-add ~/.ssh/id_rsa
```

2. Missing or Incorrect Dependencies

Description:

Tasks fail because required software is not installed.

Symptoms:

- Module or task failures due to missing dependencies.

Resolution:

- Install required dependencies on target machines.
- Use `ansible.builtin.package` to enforce dependency installation.

Code:

None

```
- name: Install dependencies

  ansible.builtin.package:

    name: "{{ item }}"

    state: present

  loop:

    - python3

    - python3-pip
```

3. YAML Syntax Errors

Description:

Playbooks cannot run due to formatting issues.

Symptoms:

- Ansible playbook fails with syntax errors.

Resolution:

- Validate YAML syntax with a YAML linter/validator.
- Ensure proper indentation, colons, and dashes.
- Remove trailing spaces or tabs.

4. Undefined Variables

Description:

Playbooks reference variables that don’t exist.

Symptoms:

- Errors indicating variables are undefined.

Resolution:

- Define variables in inventory, playbooks, or `extra_vars`.
- Use `default` filter to provide fallback values.

Code:

None

```
- name: Use default value for undefined variable

  debug:

    msg: "{{ my_variable | default('default_value') }}"
```

5. Module Not Found

Description:

Ansible cannot locate a required module.

Symptoms:

- Errors indicating a module is not found.

Resolution:

- Verify the module name is correct.
- Check compatibility with your Ansible version.
- Install missing collections.

Code:

Shell

```
ansible-galaxy collection install community.general
```

6. Permission Issues

Description:

Tasks fail due to insufficient privileges.

Symptoms:

- Access denied errors on privileged tasks.

Resolution:

- Use `become: yes` to escalate privileges.
- Configure `become_method` if required.

Code:

None

```
- name: Task requiring sudo


  become: yes
  ansible.builtin.command:

    cmd: apt-get update
```

7. Playbook Logic Errors

Description:

Playbook executes but does not behave as expected.

Symptoms:

- Tasks run incorrectly or skip unexpectedly.

Resolution:

- Use `debug` to inspect variable values.
- Check conditionals and loops carefully.

Code:

None

```
- name: Debug variable value

  debug:

    var: my_variable
```

8. Host Unreachable

Description:

Target hosts cannot be contacted by Ansible.

Symptoms:

- Ansible reports a host is unreachable.

Resolution:

- Check network connectivity.
- Verify host/IP in inventory.
- Ensure machine is online and accessible.

9. Incorrect File Paths

Description:

Playbooks reference non-existent or wrong file paths.

Symptoms:

- Errors about missing or invalid file paths.

Resolution:

- Ensure files exist at specified paths.
- Prefer absolute paths over relative ones.

10. Role or Playbook Not Found

Description:

Ansible cannot find referenced roles or playbooks.

Symptoms:

- Errors indicating missing role or playbook.

Resolution:

- Verify file paths.

- Use `ansible-galaxy` to install missing roles.

Code:

Shell

```
ansible-galaxy install geerlingguy.apache
```

11. Misconfigured Inventory

Description:

Inventory entries are incorrect or malformed.

Symptoms:

- Tasks fail or target wrong hosts.

Resolution:

- Validate inventory syntax.
- Ensure correct group/host definitions.

12. Idempotence Issues

Description:

Tasks apply changes repeatedly when they should not.

Symptoms:

- Playbook changes state every run.

Resolution:

- Ensure tasks are idempotent.
- Use conditionals to limit execution.

Code:

None

```
- name: Ensure file is present

  ansible.builtin.file:

    path: /etc/myfile

    state: touch

  when: not (ansible_facts['distribution'] == 'Ubuntu')
```