🧪 Lab 19 — Delegation & Local Actions in Ansible (`delegate_to`, `run_once`, `local_action`)

Author: *Sandeep Kumar Sharma*

🕐 Learning Objectives

In this lab you will learn: - What task delegation means in Ansible - How to use `delegate_to` to run tasks on a different host - How to run tasks only once using `run_once` - How to use `local_action` for control-node execution - How delegation helps in load balancer configs, file generation, and orchestration

---

👙 Learning Outcomes

After completing this lab, you will: - Execute tasks on alternate hosts while looping over others - Run orchestration-style playbooks - Understand controller-side actions - Use `delegate_to` and `run_once` effectively in automation

---

🧰 What Is Delegation in Ansible?

Delegation allows you to run a task **on a different machine** than the one defined by the play.

For example: - Play targets app servers - But configuration file must be updated only on the load balancer - Or logs must be copied to a central host

Delegation makes this possible.

Why Use Delegation?

Common use cases: - Updating a load balancer after deploying apps - Writing to a central database or monitoring server - Running tasks on control node - Creating or editing files that belong to only one machine - Orchestrating multi-node deployments

---

🏦SECTION A — Using `delegate_to` (Basic Example)

This task will run on **localhost**, even though play targets remote nodes.

Create file:

```
nano delegate-basic.yml
```

Add:

```
---
- name: Delegate task to localhost
  hosts: dev
  become: yes

  tasks:
    - name: Print message from control node
      debug:
        msg: "This runs on the control node, not on the dev hosts"
      delegate_to: localhost
```

Run:

```
ansible-playbook delegate-basic.yml
```

Expected: - Every host in `dev` triggers the task - But execution happens on *localhost*

🏦SECTION B — Using `delegate_to` for Load Balancer Updates

Create file:

```
nano delegate-lb.yml
```

Add:

```
---
- name: Update load balancer config when app nodes change
  hosts: appservers
  become: yes

  tasks:
    - name: Inform load balancer
      debug:
        msg: "App server {{ inventory_hostname }} is deployed. Updating LB."
      delegate_to: lbserver
```

Description: - All app servers trigger the task - But the task runs **only on lbserver** each time

---

🏛️ SECTION C — Using `run_once`

Sometimes tasks should run **only once**, even if multiple hosts exist.

Example: generating a configuration file.

Create file:

```
nano run-once.yml
```

Add:

```
---
- name: Run a task only once
  hosts: dev
  become: yes

  tasks:
    - name: Generate report
      debug:
        msg: "This task runs only once, no matter how many hosts you have"
      run_once: true
      delegate_to: localhost
```

Run:

```
ansible-playbook run-once.yml
```

Expected: - Only one debug message printed

🏛️ SECTION D — Using `local_action`

This is shorthand for delegation to localhost.

Create file:

```
nano local-action.yml
```

Add:

```
---
- name: Demonstrate local_action
  hosts: dev

  tasks:
    - name: Create a local file on control node
      local_action:
        module: copy
        content: "Generated by Ansible from remote nodes"
        dest: /tmp/local_output.txt
```

Run:

```
ansible-playbook local-action.yml
```

Expected: - File created on control node, not remote nodes

🧪 Hands-On Checklist

- [ ] Delegate task to localhost
- [ ] Delegate tasks to a load balancer host
- [ ] Run a task only once using `run_once`
- [ ] Use `local_action` to create a local file

Lab Summary

This lab introduced task delegation (`delegate_to`), `run_once`, and `local_action` to enable orchestration workflows, central logging, controller-side execution, and efficient multi-node automation.

Author: Sandeep Kumar Sharma