**Lab Guide: Static Inventory & Job Execution in Ansible Automation Platform (AAP) 2.5**

**🎯 Lab Objective**

Set up a fully functional automation workflow using **static inventory**, **Git-based projects**, **EC2 Ubuntu instances**, **machine credentials**, and a **job template** to automate basic tasks.

**🛠️ Prerequisites**

* AWS account with EC2 access
* Installed AAP 2.5 controller
* Access to AAP web UI
* SSH private key to access Ubuntu EC2 nodes
* A Git repository (GitHub/GitLab/Bitbucket)

**🔹 Step 1: Launch Two Managed EC2 Ubuntu Nodes**

1. Go to AWS EC2 → Launch two instances:
   * AMI: Ubuntu 22.04 LTS
   * Instance Type: t2.micro
   * Key Pair: Use or create one (.pem file needed)
   * Name tags:
     + Name: m1 for the first instance
     + Name: m2 for the second instance
   * Allow SSH access via Security Group (port 22)
2. Note down the **Public IP addresses** of both instances.

**🔹 Step 2: Create Git Repository for Project**

1. On GitHub/GitLab:
   * Create a repository: raman-first-project
   * Add your Ansible playbooks to this repo (e.g., create-user-dir.yml)

yaml

CopyEdit

# create-user-dir.yml

- hosts: all

become: yes

connection: ssh

tasks:

- name: Create a directory

file:

path: /opt/rk

state: directory

- name: Create a user

user:

name: usertest

shell: /bin/bash

uid: 6666

comment: "my rh-aaap user"

state: present

🔁 Make sure repo is publicly cloneable or set up Git credentials in AAP.

**🔹 Step 3: Create Project in AAP**

1. Go to AAP UI → **Resources → Projects → Add**
2. Fill in the details:
   * **Name**: raman-first-project
   * **SCM Type**: Git
   * **SCM URL**: https://github.com/<your-user>/raman-first-project.git
   * Check:
     + ✅ *Allow branch override*
     + ✅ *Update revision on launch*
3. Save and click **Sync** to fetch playbooks.

**🔹 Step 4: Create Static Inventory**

1. Go to **Resources → Inventories → Add**
   * Name: raman-inv-static
   * Organization: Default (or your org)
2. Inside the inventory:
   * Go to **Groups → Add**
     + Create two groups: web and db
3. Inside web group → **Hosts → Add**:
   * Name: m1
   * Add ansible\_host = <Public IP of m1>
4. Inside db group → **Hosts → Add**:
   * Name: m2
   * Add ansible\_host = <Public IP of m2>
5. Add host-level variables (click Edit Variables):

yaml

CopyEdit

ansible\_user: ubuntu

⚠️ Do **not** add ansible\_ssh\_private\_key\_file here—it will be managed via credentials.

**🔹 Step 5: Create Machine Credential**

1. Go to **Resources → Credentials → Add**
   * Name: ubuntu-ec2-key
   * Credential Type: *Machine*
   * Inputs:
     + **Username**: ubuntu
     + **Private Key**: Paste contents of your .pem file (or upload)
2. Save.

**🔹 Step 6: Create Job Template**

1. Go to **Resources → Templates → Add → Job Template**
   * **Name**: user-dir-job
   * **Job Type**: Run (enable *Prompt on Launch*)
   * **Inventory**: raman-inv-static
   * **Project**: raman-first-project
   * **Playbook**: create-user-dir.yml
   * **Credential**: ubuntu-ec2-key
   * ✅ Enable:
     + Privilege Escalation
     + Fact Storage (optional)
2. Save the job template.

**🔹 Step 7: Launch Job Template**

1. Go to **Templates → user-dir-job → Launch**
2. On launch prompt:
   * Confirm inventory/project/playbook
   * Click **Next/Launch**
3. Monitor job execution in **Jobs tab**

**✅ Expected Results**

* /opt/rk directory is created on both m1 and m2
* User usertest with UID 6666 and shell /bin/bash is created
* Successful task status in AAP

**📁 Git Repository Structure Example**

bash

CopyEdit

raman-first-project/

│

├── create-user-dir.yml

└── README.md

**🔎 Troubleshooting Tips**

| **Issue** | **Resolution** |
| --- | --- |
| Permission denied (SSH) | Make sure correct private key is added as Machine Credential |
| Host unreachable | Ensure EC2 instances allow SSH from AAP server’s IP |
| Invalid playbook error | Verify syntax and indentation in YAML |
| Missing privilege escalation | Enable become in playbook and template |

**🧼 Cleanup**

To avoid unnecessary AWS costs:

* Terminate EC2 instances
* Delete inventory, credentials, projects from AAP if no longer needed
* **🧪 Lab2: Parameterize Jobs using Survey in AAP**

**🎯 Objective**

Configure a Job Template in **Ansible Automation Platform 2.5** that:

* Accepts **three dynamic parameters** via Survey:
  + dir\_state: Control if directory is created/absent
  + dir\_path: Directory path to manage
  + user\_state: Control user creation/removal
* Executes tasks accordingly on managed hosts.

**🧱 Prerequisites**

1. A working **AAP 2.5 cluster** (controller + EE).
2. SSH access from AAP to target managed nodes (e.g., m1, m2).
3. A **project repo** (local/Git) with your playbook.
4. A **machine credential** (e.g., user ubuntu or ansible).
5. Inventory with at least 1–2 hosts.

**📁 Step 1: Create Your Playbook**

Save the following as parameterized\_task.yml:

yaml

CopyEdit

- hosts: all

become: yes

connection: ssh

tasks:

- name: Create a directory (controlled via survey)

file:

path: "{{ dir\_path }}"

state: "{{ dir\_state }}"

- name: Create a user (controlled via survey)

user:

name: usertest

shell: /bin/bash

uid: 6666

comment: "my rh-aaap user"

state: "{{ user\_state }}"

✅ This playbook uses 3 variables we’ll pass at runtime via Survey.

**📂 Step 2: Upload to AAP Project**

**Option A: Git Project**

1. Push the playbook to a Git repo (GitHub/GitLab).
2. In AAP → **Resources → Projects → Add**:
   * Name: SurveyDemoProject
   * Source Control: Git
   * URL: https://github.com/<your\_repo>.git
   * Branch: main
   * Save & Sync

**Option B: Manual Project**

1. Upload playbook to AAP controller's PROJECTS\_ROOT.
2. In AAP → Projects → Add:
   * Name: SurveyDemoProject
   * Source Control: Manual
   * Playbook Directory: parameterized\_lab
   * Save

**📘 Step 3: Create a Job Template**

Go to: **Templates → Add → Job Template**

**Fill Details:**

| **Field** | **Value** |
| --- | --- |
| **Name** | Parameterized Job Survey |
| **Inventory** | Choose your inventory |
| **Project** | SurveyDemoProject |
| **Playbook** | parameterized\_task.yml |
| **Execution Env.** | Default or custom EE |
| **Credentials** | Machine credential (e.g., SSH) |

Do **not** add extra variables here — we’ll pass them via **Survey**.

Click **Save**.

**📊 Step 4: Add Survey**

1. In the Job Template view, click **Add Survey**
2. Click **Add Question** three times for the following:

**✅ Survey Question 1: Directory State**

| **Field** | **Value** |
| --- | --- |
| **Prompt** | Choose the directory state |
| **Variable Name** | dir\_state |
| **Answer Type** | Multiple Choice (single select) |
| **Choices** | directory, absent |
| **Default** | directory |
| **Required** | Yes |

**✅ Survey Question 2: Directory Path**

| **Field** | **Value** |
| --- | --- |
| **Prompt** | Enter directory path |
| **Variable Name** | dir\_path |
| **Answer Type** | Text |
| **Default** | /opt/rk |
| **Required** | Yes |

**✅ Survey Question 3: User State**

| **Field** | **Value** |
| --- | --- |
| **Prompt** | Choose the user state |
| **Variable Name** | user\_state |
| **Answer Type** | Multiple Choice (single select) |
| **Choices** | present, absent |
| **Default** | present |
| **Required** | Yes |

Click **Save Survey** and then **Save Template**.

**🚀 Step 5: Run the Job**

1. Go to **Templates**
2. Click ▶️ **Launch** on Parameterized Job Survey
3. Fill survey:
   * Directory State: absent
   * Directory Path: /opt/rk
   * User State: present
4. Click **Next → Launch**

**✅ Step 6: Verify Output**

* AAP job output will show whether:
  + Directory was created or deleted
  + User was created or removed
* You can check results via:
  + AAP Job stdout
  + SSH into the target host and verify /opt/rk and usertest

**🧪 Additional Test Scenarios**

Try re-running the job with different combinations:

| **dir\_state** | **dir\_path** | **user\_state** | **Expected Outcome** |
| --- | --- | --- | --- |
| directory | /opt/demo | present | /opt/demo is created, usertest is present |
| absent | /opt/temp | absent | /opt/temp deleted, usertest removed |
| directory | /data/logs | absent | /data/logs created, usertest removed |

**LAB3 : Using dynamic inventory instead of static :**

**🟩 Step 1: Create a New Inventory in AAP**

1. Navigate to the AAP **UI**.
2. Go to **Inventories** → **Add** → **Inventory**.
3. Fill in:
   * **Name**: raman-dynamic-inventory
   * **Organization**: Your org
   * **Description**: (Optional) Dynamic EC2 Inventory
4. Click **Save**.

**🟩 Step 2: Create AWS Cloud Credential**

1. Go to **Credentials** → **Add** → **Credential**.
2. Fill in:
   * **Name**: raman-amazon-acc-creds
   * **Credential Type**: Amazon Web Services
   * **Organization**: Your org
3. Under **Input Configuration**:
   * **Access Key**:
   * **Secret Access Key**:
4. Click **Save**.

**🟩 Step 3: Add Dynamic Source to Inventory**

1. Go to your inventory: raman-dynamic-inventory.
2. Click **Sources** → **Add Source**.
3. Fill in:
   * **Name**: raman-dynamicInv-source
   * **Source**: Amazon EC2
   * **Credential**: raman-amazon-acc-creds
4. Enable the following options:
   * ✅ **Overwrite**
   * ✅ **Overwrite Variables**
   * ✅ **Update on Launch**

**🟩 Step 4: Configure Source Variables**

-- add below variables in source variables ( not in inventory level variables )

plugin: amazon.aws.aws\_ec2

# The values for profile, access key, secret key and token can be hardcoded like:

regions:

  - us-east-2

#filters:

  ## All instances with their `Environment` tag set to `dev`

  # tag:Environment: dev

---------------------------------------------------------

plugin: amazon.aws.aws\_ec2

regions:

  - us-east-2

filters:

  tag:Env: training

    --------------------------------------------------

    -- after testing by launching inv source :

        --- we go to console and tag the servers that i want to detect by inv dynamic

        plugin: amazon.aws.aws\_ec2

regions:

  - us-east-2

filters:

  instance-state-name: running

keyed\_groups:

  - key: tags.Role

    prefix: role

compose:

  ansible\_host: public\_ip\_address

Note : make sure to tag all servers of one group as role=web and all other group servers as role=db

**🟩 Step 7: Validate Inventory Groups and Hosts**

1. Go to **Inventories** → raman-dynamic-inventory.
2. Click **Hosts** and **Groups** tabs.
3. Confirm:
   * Hosts have public IPs set as ansible\_host
   * Hosts are grouped like role\_web, role\_app based on EC2 tag