

# Mock Interview Guide

## Ansible

### Instructions for Interviewer:

- You are playing the role of **interviewer**. Use this guide as a script.
  - Ask each question one at a time. Follow the steps: **Definition** → **Details** → **Scenario** → **Follow-up**.
  - If the interviewee struggles, use the **hint**.
  - The goal is to keep it conversational and practical. Help the interviewee think and express their learning.
  - colors assigned: Questions Answers Hint
- 

### Freshers - Level

## Ansible

### (10 Easy DevOps Interview Questions)

---

#### 1. “What is Ansible?”

✓ Expected Answer: “Ansible is an open-source automation tool used for configuration management, application deployment, and task automation.”

*Hint: “You need to install packages or configure hundreds of servers at once — this tool helps do it with a single command.”*

#### 2. “What is an Ansible playbook?”

✓ Expected Answer: “A playbook is a YAML file where you define a list of tasks to be executed on target systems.”

*Hint: “It’s like a recipe written in YAML that tells Ansible what steps to perform on which servers.”*

### **3. “What language is used to write Ansible playbooks?”**

✓ Expected Answer: “YAML (Yet Another Markup Language).”

*Hint: “It’s human-readable and often starts with ‘---’ at the top.”*

### **4. “What is an Ansible inventory?”**

✓ Expected Answer: “It’s a file that lists all the target servers (hosts) Ansible should manage.”

*Hint: “Where do you tell Ansible which machines to automate? This file answers that.”*

### **5. “What is a task in Ansible?”**

✓ Expected Answer: “A task is a single action, like installing a package or starting a service, defined in a playbook.”

*Hint: “Each step in a playbook is one of these — think of it as a job to do on the server.”*

### **6. “Do you need to install agents on nodes for Ansible?”**

✓ Expected Answer: “No, Ansible is agentless and connects to target nodes via SSH.”

*Hint: “Unlike some tools, it doesn’t need anything installed on the remote machine — just SSH access.”*

## **7. “How do you run an Ansible playbook?”**

✓ Expected Answer: “Using the ansible-playbook command.”

*Hint: “Once you’ve written your YAML file, this is the command that starts the automation.”*

## **8. “What is a module in Ansible?”**

✓ Expected Answer: “Modules are reusable scripts used by Ansible to perform specific tasks like installing software or copying files.”

*Hint: “They’re the tools Ansible uses under the hood to execute your tasks — like building blocks.”*

## **9. “What is an Ansible role?”**

✓ Expected Answer: “Roles are a way to organize playbooks into reusable, structured components.”

*Hint: “Instead of writing everything in one big file, you use this to break down tasks into folders like tasks/, vars/, handlers/.”*

## **10. “Which port does Ansible use by default to connect to remote hosts?”**

✓ Expected Answer: “Port 22 (SSH).”

*Hint: “Same port you use to SSH into a Linux server.”*

---

## SCENARIO-BASED INTERVIEW QUESTIONS

---

**1. Ask: “You ran an Ansible playbook, but it says the host is unreachable. What should you check first?”**

✓ Expected: *Check SSH connectivity, correct IP, and that the host is defined properly in the inventory file.*

*Hint: Ansible needs SSH access — start there.*

**2. Ask: “Ansible keeps skipping tasks in your playbook. What’s likely causing this?”**

✓ Expected: *The task may be using when conditions that evaluate to false, or handlers that weren’t notified.*

*Hint: Look for conditional logic.*

**3. Ask: “You edited a playbook but Ansible keeps using old variable values. What might be happening?”**

✓ Expected: *The variables may be overridden elsewhere (group\_vars, host\_vars, etc.).*

*Hint: Variable precedence can confuse if not tracked.*

**4. Ask: “You have a working playbook locally, but on a teammate’s machine it fails. Why?”**

✓ Expected: *Their Ansible version or environment might be different. Compare versions, dependencies, and inventory paths.*

Hint: *Consistency is key in automation tools.*

**5. Ask: “Your playbook runs fine but the target server doesn’t reflect changes. What could be missing?”**

✓ Expected: *Tasks may not trigger because Ansible sees them as unchanged. Use changed\_when or force in some modules.*

Hint: *Idempotency works both ways — check change detection.*

---

## PROJECT-BASED INTERVIEW QUESTIONS

---

**1. Ask: “How would you use Ansible to install Apache on a remote Ubuntu server?”**

✓ Expected: *Create a playbook using the apt module to install apache2, and target it to a host in the inventory file.*

Hint: *One module + one task = working web server.*

**2. Ask: “How would you copy a file from your local machine to multiple servers using Ansible?”**

✓ Expected: *Use the copy module in a playbook or ad-hoc command, specifying src and dest.*

Hint: *Automation = same result across all nodes.*

**3. Ask: “How would you write a playbook that creates a user on multiple Linux servers?”**

✓ Expected: *Use the user module with a name and state: present in your playbook.*

Hint: *Ansible has a module for everything — even users.*

**4. Ask: “How do you organize your inventory if you want to run a playbook on just your web servers?”**

✓ Expected: *Group them under [web] in the inventory and target that group in the playbook.*

Hint: *Inventory groups = powerful targeting.*

---

## Medium - Level

### Ansible

## (DevOps Interview Questions - 1 to 2 Years Experience)

---

### 1. “How does Ansible work without agents?”

✓ Answer:

Ansible uses SSH to connect to target machines, so it doesn't require any agent to be installed.

This simplifies setup and reduces management overhead.

*Hint: No background services — just a direct remote connection.*

### 2. “What's the difference between a playbook and a role?”

✓ Answer:

Playbooks define tasks to run; roles organize them into reusable, structured units.

Roles follow a standard folder layout for better scalability.

*Hint: Roles = reusable building blocks, Playbooks = action plan.*

### 3. “How does Ansible handle variables?”

✓ Answer:

Variables can be defined in many places: playbooks, inventories, facts, roles, or command-line.

Precedence determines which value takes effect.

*Hint: Where and how you define the variable affects its final value.*

#### **4. “What is an inventory file in Ansible?”**

✓ **Answer:**

An inventory file lists hosts and groups of hosts that Ansible manages. It can be static (INI/YAML) or dynamic (Python script/cloud plugin).

*Hint: It's how Ansible knows who to talk to.*

#### **5. “What is the purpose of ansible.cfg?”**

✓ **Answer:**

ansible.cfg is a configuration file to set default behavior like inventory path, SSH settings, retries, etc. It customizes how Ansible behaves.

*Hint: It's your control panel for customizing Ansible.*

#### **6. “What is a handler in Ansible?”**

✓ **Answer:**

A handler is a special task that runs only when notified by another task.

Usually used for restarting services after configuration changes.

*Hint: Triggered only if something changes.*

#### **7. “How can you make a task run only when a condition is met?”**



✓ **Answer:**

Use the when clause in a task to define conditions based on variables or facts.

This allows conditional execution.

*Hint: Like an if-statement in your automation.*

## 8. “How do you test Ansible playbooks before applying to production?”

✓ **Answer:**

Use --check (dry run), --diff to see changes, or tools like Molecule for local testing.

Also test in staging environments first.

*Hint: Look before you leap — always simulate or stage first.*

## 9. “What is idempotency in Ansible?”

✓ **Answer:**

Idempotency ensures running a playbook multiple times won’t change the system again if it’s already in the desired state.

This is a key Ansible principle.

*Hint: Run once or ten times — result stays the same.*

## 10. “How do you pass extra variables to a playbook at runtime?”

✓ **Answer:**

Use the -e flag like `ansible-playbook play.yml -e "var=value"` to

override or inject values.  
You can also pass a JSON or YAML file.

*Hint: Command-line variables override almost everything else.*

---

## SCENARIO-BASED INTERVIEW QUESTIONS

---

*1. Ask: “Your playbook runs successfully, but the latest configuration changes aren't reflected on the servers. What could be wrong?”*

✓ Expected: *Handlers may not be triggered, or cached facts might be used. Also, some modules skip tasks if not forced.*

*Hint: Handlers and idempotency can silently skip changes.*

*2. Ask: “You want to run a play only for hosts in the web group that are also in the production group. How would you do that?”*

✓ Expected: *Use an inventory intersection like `web:&production` in the playbook's `hosts:` field.*

*Hint: Use group math: `group1:&group2`.*

*3. Ask: “How can you make a task execute only when a variable exists and is not empty?”*

✓ Expected: *Use a `when` condition: `when: variable is defined and variable != ''`.*

*Hint: Guard your logic with a clean conditional check.*

**4. Ask: “You have common roles reused across projects. How do you avoid repeating the same code?”**

✓ Expected: Use Ansible roles with shared defaults, tasks, and import them in multiple playbooks.

*Hint: Roles = reusable building blocks.*

**5. Ask: “Your playbook needs to fetch dynamic data like instance IPs from AWS. How would you do that?”**

✓ Expected: Use dynamic inventory or Ansible facts with cloud modules or external inventory scripts.

*Hint: Static files won’t help — fetch live data on run.*

---

## PROJECT-BASED INTERVIEW QUESTIONS

---

**1. Ask: “Design a playbook that installs Nginx, copies a config file, and starts the service only if installation is successful.”**

✓ Expected: Use apt module → copy module → service with when condition based on install success or notify handler.

*Hint: Use register + when, or notify → handler.*

**2. Ask: “You have different vars for dev, staging, and prod. How do you manage this in Ansible?”**

✓ **Expected:** *Use group\_vars or host\_vars directories and override values based on environment groups.*

**Hint:** *Group-based variable overrides are clean and scalable.*

**3. Ask: “How do you design a playbook that deploys code from Git, ensures required packages are installed, and runs a restart only on change?”**

✓ **Expected:** *Use git module → package module → restart using handlers triggered only on changes.*

**Hint:** *Handlers ensure restarts only when needed.*

**4. Ask: “You need to run a task across multiple servers but limit it to 2 servers at a time. How would you do this?”**

✓ **Expected:** *Use serial: 2 in the play definition to limit concurrency.*

**Hint:** *Think: staged rollouts to avoid full outages.*

---

## Hard - Level

### Ansible

### (DevOps Interview Questions - 3+ Years Experience)

---

**1. “How would you make Ansible idempotent when using shell or command modules?”**

✓ **Answer:**

Use `creates=`, `removes=`, or `when: conditions` to avoid re-running tasks.

These options help ensure tasks don't apply repeatedly.

*Hint: Shell/command tasks don't know current state — you must help them.*

**2. “How can you scale Ansible to manage thousands of servers?”**

✓ **Answer:**

Use dynamic inventory, limit concurrency with forks, enable pipelining, and consider AWX/Tower.

Split inventory and roles for modular management.

*Hint: What if you had to patch 3,000 servers tonight?*

**3. “What are facts in Ansible and how can you optimize their usage?”**

✓ **Answer:**

Facts are system information collected via setup.

To optimize, use `gather_facts: false` when not needed, or cache facts.

*Hint: Unneeded facts slow down every playbook run.*

#### **4. “How would you implement error handling in Ansible?”**

✓ **Answer:**

Use `ignore_errors`, `failed_when`, `block/rescue/always`, or conditionally skip tasks.

Helps handle partial failures and cleanups.

*Hint: What happens when a step fails mid-run — do you catch or ignore?*

#### **5. “What are Ansible callbacks, and give an example use case?”**

✓ **Answer:**

Callbacks modify output or trigger actions during playbook execution.

Example: Slack notifications or custom logging.

*Hint: Want real-time build updates on Slack?*

#### **6. “How would you use Ansible Vault in a CI/CD pipeline?”**

✓ **Answer:**

Decrypt vault files at runtime using vault password from environment variables or secret stores.

Avoid manual interaction by scripting decryption securely.

*Hint: CI/CD runs headless — you can't type a password.*

## 7. “How can you use Ansible to perform rolling updates with zero downtime?”

✓ **Answer:**

Use serial with inventory groups, service health checks, and notify handlers.

Only a subset of servers is updated at a time.

*Hint: Avoid taking down your whole app — update in batches.*

## 8. “Explain the role of `delegate_to` and how it's different from `local_action`.”

✓ **Answer:**

`delegate_to` runs a task on another host, not the one in context.

`local_action` runs it on the control node.

*Hint: Need to run something elsewhere — not on the target?*

## 9. “How do you write dynamic inventory scripts, and when are they needed?”

✓ **Answer:**

Scripts or plugins fetch inventory from sources like AWS, GCP, or CMDBs.

Use when infrastructure is dynamic or large-scale.

*Hint: Your servers change daily — how does Ansible keep up?*

## 10. “What’s the difference between `include_tasks`, `import_tasks`, and `import_playbook`?”

✓ Answer:

`include_tasks` is dynamic (evaluated at runtime) `import_tasks` and `import_playbook` are static (evaluated at parse time)  
Use dynamic includes for conditional logic.

*Hint: Want conditional task inclusion? Choose wisely.*

---

## SCENARIO-BASED INTERVIEW QUESTIONS

---

**1. Ask:** “*You need to apply different configurations to servers based on their datacenter location, all from one playbook. How would you do it?*”

✓ Expected: *Use conditionals with host/group vars (when: datacenter == 'us-east') or dynamic facts.*

*Hint: Let Ansible adapt to where it runs.*

**2. Ask:** “*Your team needs to audit changes made by Ansible. How can you track and log what Ansible did on each host?*”

✓ Expected: *Enable callback plugins like `log_plays`, or use verbose output + centralized logging of runs via syslog or custom plugins.*

*Hint: Ansible can log more than console — plug it in.*



**3. Ask: “You want to refactor a large playbook used by multiple teams into reusable components. What’s your approach?”**

✓ Expected: *Break into roles with clear inputs/outputs, use defaults, handlers, and meta files, and store in a role repository.*

Hint: *Roles = modular, reusable, and shareable units.*

**4. Ask: “One task in a role is failing but you don’t want it to fail the entire playbook. What’s the cleanest way to handle it?”**

✓ Expected: *Use ignore\_errors: yes or handle with failed\_when: false if specific failure conditions are acceptable.*

Hint: *Fail gracefully — only when it’s critical.*

**5. Ask: “How would you ensure that secrets (like DB passwords) used in your playbooks aren’t exposed?”**

✓ Expected: *Use Ansible Vault to encrypt secrets and variables, and access them with ansible-vault CLI.*

Hint: *Secrets should be stored encrypted, not plain YAML.*

---

## PROJECT-BASED INTERVIEW QUESTIONS

---

**1. Ask: “Design an Ansible setup to manage 200+ servers across 3 environments (dev/staging/prod), ensuring safe and staged rollouts.”**

✓ Expected: *Use group\_vars/env separation, dynamic inventory, serial execution, and tagging for staged changes.*

Hint: *Controlled rollouts = inventory + logic.*

**2. Ask: “You want to integrate Ansible with Jenkins for CI/CD. How would you do it?”**

✓ Expected: *Write Ansible playbooks and call them in Jenkins pipelines using ansible-playbook with correct inventory and credentials.*

Hint: *Think: Infrastructure as Code meets CI pipeline.*

**3. Ask: “You’re provisioning cloud instances and configuring them using Ansible. How do you handle hosts that don’t exist yet at playbook start?”**

✓ Expected: *Use dynamic inventory scripts or provision infrastructure in a separate task before config runs.*

Hint: *Provision first, configure next.*

**4. Ask: “You want to deploy a microservice stack with dependencies (DB → API → Web). How would you enforce that order in Ansible?”**

✓ Expected: *Use serial, depends\_on, or group-based targeting and organize playbooks to control execution order.*

Hint: *Service order matters — structure your plays for it.*