

Ansible Authentication Failed: Permission Denied Error

Manual SSH Works

You can successfully log in using ssh supawit@10.0.53.24 with password authentication through interactive prompt.

Ansible Ping Fails

The Ansible ping module returns: **Permission denied** (publickey,gssapikeyex,gssapi-withmic,password)

Widespread Issue

This authentication failure occurs consistently across all target hosts in your infrastructure.



Three Solutions to Fix Ansible Authentication

0

Quick Fix: Use Password Authentication

Run Ansible with the --ask-pass flag to prompt for password input:

ansible all -i inventory.ini -m ping -u supawit --ask-pass

02

Recommended: Set Up SSH Key Authentication

Generate SSH keys and copy to managed hosts for seamless, secure access:

ssh-keygen -t rsa -b 4096 -C "supawit" ssh-copy-id supawit@10.0.53.24 ssh-copy-id supawit@10.0.53.26 ssh-copy-id supawit@10.0.53.28

Test connection: ssh supawit@10.0.53.24

03

Alternative: Store Credentials in Inventory

Add authentication details directly in inventory.ini (less secure):

[all]

10.0.53.24 ansible_user=supawit ansible_password=your_password 10.0.53.26 ansible_user=supawit ansible_password=your_password

Understanding Why This Authentication Issue Occurs

Root Cause Analysis

Ansible defaults to **SSH key authentication** for security and automation efficiency. When no valid SSH keys are configured, authentication fails silently.

Manual SSH sessions work because they interactively prompt for passwords, while Ansible expects automated authentication without user intervention.

Setting up SSH keys eliminates password prompts and creates a secure, seamless workflow for infrastructure management.



Best Practice: SSH key authentication is the gold standard for Ansible deployments. It provides better security than password storage and enables fully automated playbook execution.