Journal of Computer Science and Frontier Technologies
ISSN Print: 3104-4204
ISSN Online: 3104-4212
CD ISSN: 3104-4220
https://doi.org/10.63313/JCSFT.9027

Automated Deployment and Application of Intelligent Medical Assistant Private Cloud Platform Based on Ansible

Xiuxiu Zhong
Wenzhou Polytechnic, Wenzhou, 325035, China
Email: Lyphard3778@163.com

How to cite this paper: Zhong, X. (2025). Automated Deployment and Application of Intelligent Medical Assistant Private Cloud Platform Based on Ansible. *Journal of Computer Science and Frontier Technologies*, 1(3), 97-105. ISSN Print: 3104-4204; ISSN Online: 3104-4212.
https://doi.org/10.63313/JCSFT.9027
Published: 2025-11-25

Abstract

With the rapid development of cloud computing and artificial intelligence technologies, intelligent health consultation services have become an effective means to make up for the uneven distribution of medical resources. This paper builds a private cloud platform based on OpenStack, realizes automated deployment using Ansible, integrates DeepSeek intel-ligent medical consultation services, and achieves realtime monitoring and visualization of system resources through Prometheus and Grafana. Experimental results show that the platform has high availability, scalability and maintainability, and can provide users with stable and efficient remote health consultation services.

Keywords

Intelligent Medical Care; Private Cloud; OpenStack; Ansible; Automated Deployment; Monitoring System

1. Introduction

Health is an eternal concern of human beings, while the scarcity and uneven distribution of high-quality medical resources is a global problem. Especially in remote areas or during public health emergencies, people often have difficulty accessing timely and professional medical consultation. In recent years, the breakthrough progress of artificial intelligence technology, especially the outstanding capabilities of Large Language Models (LLMs) in natural language processing and knowledge reasoning, has brought new opportunities for reshaping the health consultation service model. LLM-based intelligent medical consultation systems can break through the constraints of time and space, provide users with 24/7 preliminary diagnosis suggestions, health knowledge popularization and medical guidance, effectively alleviating the pressure on the medical system.

However, putting such data-intensive and computing-intensive AI models into practical application puts extremely high requirements on their underlying infrastructure: it needs strong computing power to support model reasoning, flexible resource scheduling to cope with access fluctuations, reliable data security to protect user privacy, and efficient operation and maintenance system to ensure continuous and stable services. Although public cloud services are flexible, in the medical field with extremely sensitive data, private cloud platforms have become the first choice for many institutions due to their advantages in data sovereignty, security and controllability, and customization.

As a leading open-source cloud computing management platform, OpenStack provides a mature solution for building private clouds. However, it has many components and complex deployment. The traditional manual deployment method is time-consuming, labor-intensive and prone to errors, which is difficult to adapt to the needs of rapid iteration and scale expansion. As an agentless automated operation and maintenance tool, Ansible has become an ideal choice for implementing Infrastructure as Code (IaC) and automated deployment due to its simplicity and powerful functions.

Against this background, this paper proposes and implements an intelligent health consultation platform that automatically deploys OpenStack private cloud based on Ansible and hosts DeepSeek intelligent medical consultation services on it. The main contributions of this project include:

Designing a modular and reusable OpenStack cluster automated deployment scheme based on Ansible Roles, which greatly improves deployment efficiency and consistency.

Proposing a DeepSeek application deployment architecture based on private cloud platform, realizing the decoupling of applications and basic environment.

Building a comprehensive monitoring system integrating resource monitoring, service status detection and visual alarm, ensuring the stability and observability of the platform.

Conducting comprehensive performance testing and verification on the entire platform, providing practical experience and data reference for the construction of similar intelligent medical cloud platforms.

2. Overview of Core Components of Intelligent Medical Assistant Platform

2.1. Analysis of Ansible Architecturea

Ansible is an automated operation and maintenance tool developed based on Python. It communicates through the SSH protocol and does not require the installation of client agents on the target host (Agentless), which makes deployment and configuration extremely simple. According to different roles in the use of Ansible, it can be divided into users, Ansible toolset and objects of action. The specific workflow is shown in Figure 1.

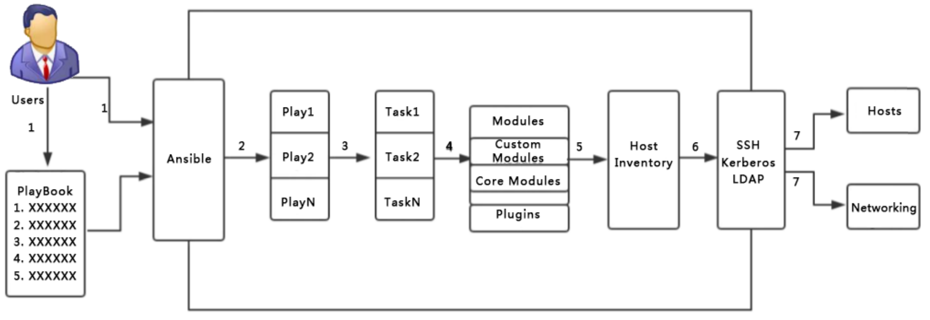

Figure 1. Ansible workflow

When users use Ansible or Ansible PlayBook (which additionally reads PlayBook files), after entering Ansible's Ad-Hoc command set or PlayBook in the server terminal, Ansible will disassemble the playbooks into Plays one by one according to pre-arranged rules, then organize the Plays into recognizable Tasks, and then call all Modules and Plugins involved in the tasks. According to the host list defined in Inventory, the task set is transmitted to the remote client in the form of temporary files or commands through SSH (default for Linux), executed, and the execution result is returned.

2.2. Maintaining the Integrity of the Specifications

OpenStack is an open-source project aimed at providing a private cloud management platform, focusing on providing Infrastructure as a Service (IaaS), allowing anyone to establish and provide cloud computing services by themselves. OpenStack is very flexible in deploying cloud environments, which can build infrastructure according to needs and expand the cluster scale flexibly. Therefore, open-source, scalability and flexible management are the three main characteristics of OpenStack. OpenStack is composed of several main components to complete specific work. Each component is composed of a separate project and provides specific services. The main components include: Keystone (Identity Service), Glance (Image Service), Nova (Compute Service), Neutron (Network Service), Horizon (UI Service), Cinder (Block Storage Service), Swift (Object Storage Service), and Heat (Orchestration Service). The architecture of OpenStack is shown in Figure 2.

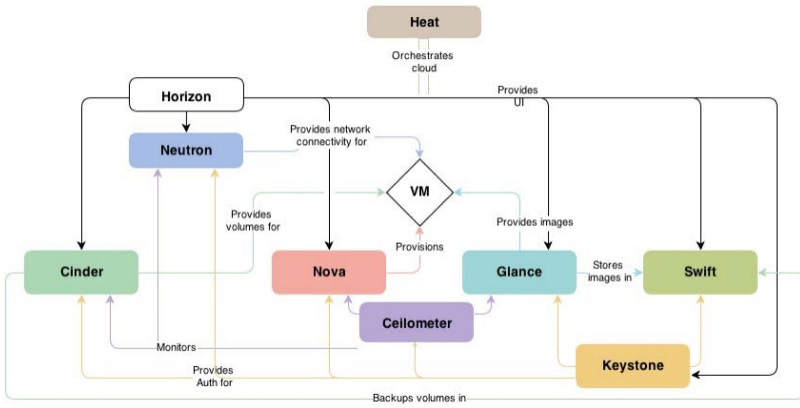

Figure 2. OpenStack Architecture Diagram

It is worth noting that the OpenStack architecture does not implement virtualization services, but uses the virtualization technology of the mature Libvirt module to realize the overall cloud platform service. Libvirt is responsible for connecting the OpenStack platform and the underlying virtualization management software, realizing the operation of virtual resources on the platform, and is an important tool for managing virtual resources in OpenStack.

The following will introduce the process of deploying OpenStack platform using Ansible.

2.3. OpenStack Deployment Process

The node planning of Ansible service is shown in Table 1.

Table 1. Node Planning

IP Address	Hostname	Node Description
192.168.130.30	ansible	Ansible Node
192.168.130.10	controller	Controller Node
192.168.130.20	compute	Compute Node

Connect three cloud hosts and modify their hostnames to ansible, controller, and compute respectively. After modification, refresh the page or execute the bash command to make the new hostname take effect.

Use the command "fdisk /dev/vdb" to partition the compute node, creating three partitions with a size of 6G each.

Install FTP and Ansible services using Ansible, and configure passwordless login from the Ansible node to the Controller and Compute nodes. After configuring passwordless login, use the Ansible node for SSH connection testing, as shown in Figure 3.

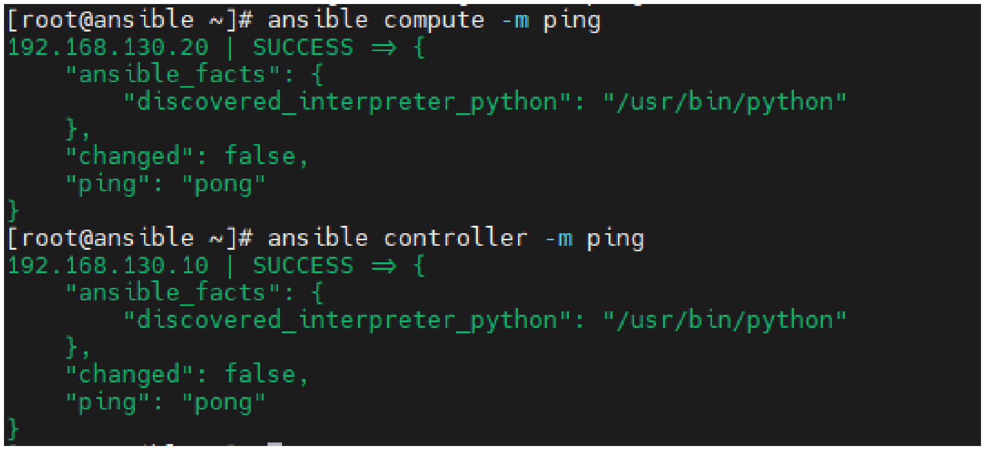

Figure 3. Ansible Node SSH Connection Test

Install source code and dependencies: Use commands to clone the content in the OpenStack-Ansible Git repository to the local, move it to the specified directory, execute the installation script located in the /opt/Openstack-ansible/scripts directory, and run the Playbook. The execution process is shown in Figure 4.

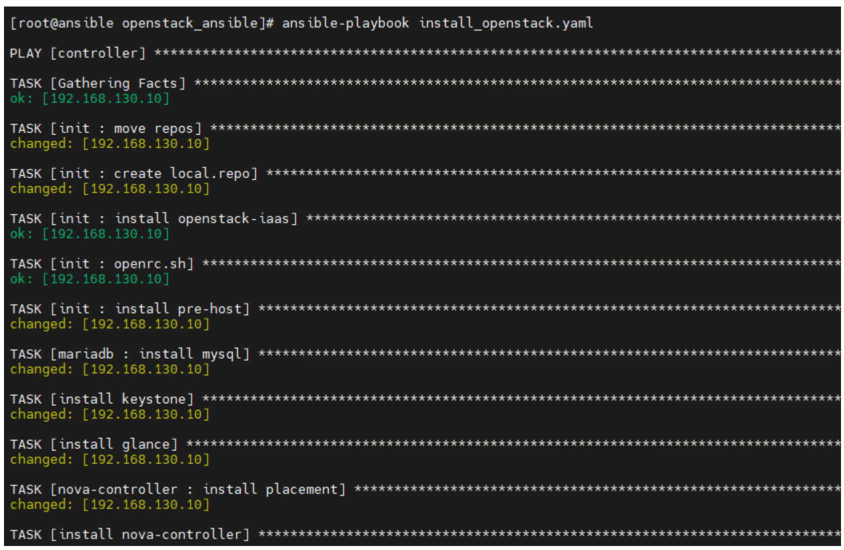

Figure 4. OpenStack Playbook Execution Process

To verify the OpenStack operation, use a web browser and enter the external load balancer IP address defined in the "external_lb_vip_address" option to access Horizon. The dashboard installed as above will use HTTPS (port 443). Log in with the administrator username and administrator password defined in the keystone_auth_admin_password /etc/openstack_deploy/user_secrets.yml file. The operation interface is shown in Figure.5.

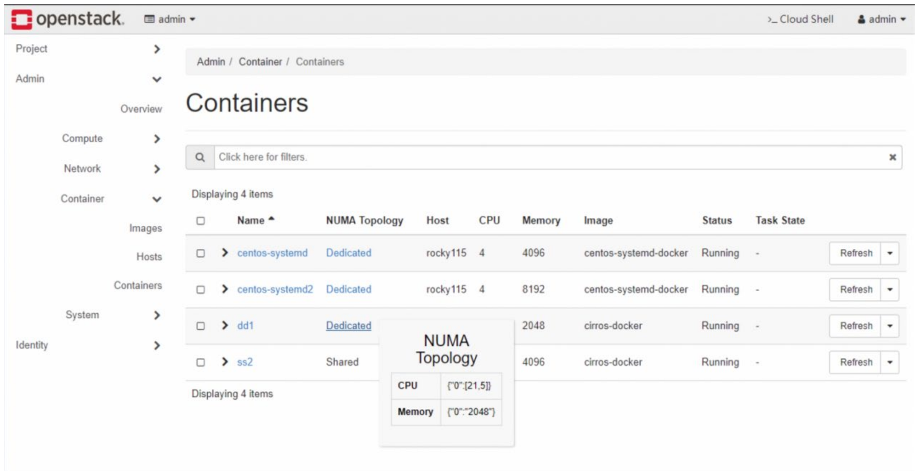

Figure 5. OpenStack Operation Interface

2.3. Deployment of DeepSeek Based on OpenStack Platform

To realize the privatization of DeepSeek on the OpenStack platform, create a cloud host using OpenStack and deploy DeepSeek on the cloud host.The core code of the data training processis shown in Figure 6.

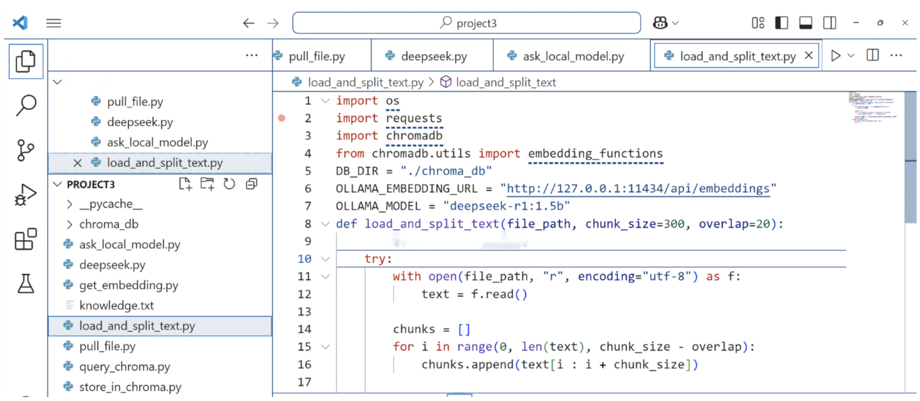

Figure 6. Data training code

3. Performance Monitoring and Testing Based on OpenStack Platform

To break through the limitations of "scattered data, weak real-time performance, and insufficient visualization" in traditional performance verification, this paper builds a full-link monitoring system of "Prometheus data collection - Grafana visualization display - alarm mechanism linkage". With quantitative indicators as the core, combined with the characteristics of medical service scenarios, it realizes the visual verification of platform performance from three dimensions: infrastructure health, AI service performance, and user access experience.

The one-click deployment of Prometheus, Grafana and Exporter components is completed through Ansible Playbook, realizing the rapid delivery and configuration unification of monitoring resources. The deployment process includes three parts:

1. Core Component Installation: Install Prometheus service (version 2.45.0) and Grafana (version 10.1.0) through Ansible's yum module. Collect host resource indicators through node_exporter, collect container operation status through cadvisor, and collect DeepSeek service reasoning indicators through custom_exporter (custom development). All Exporters are configured as system services through Ansible and set to start automatically at boot.

2. Monitoring Rule Configuration: Define monitoring targets in the Prometheus configuration file (/etc/prometheus/prometheus.yml), including Controller node, Compute node and two DeepSeek service nodes. The collection period is set to 15 seconds, and the indicator storage strategy is configured at the same time (retaining data for 15 days to avoid redundant storage resources). The configuration effect is shown in Figure 7.

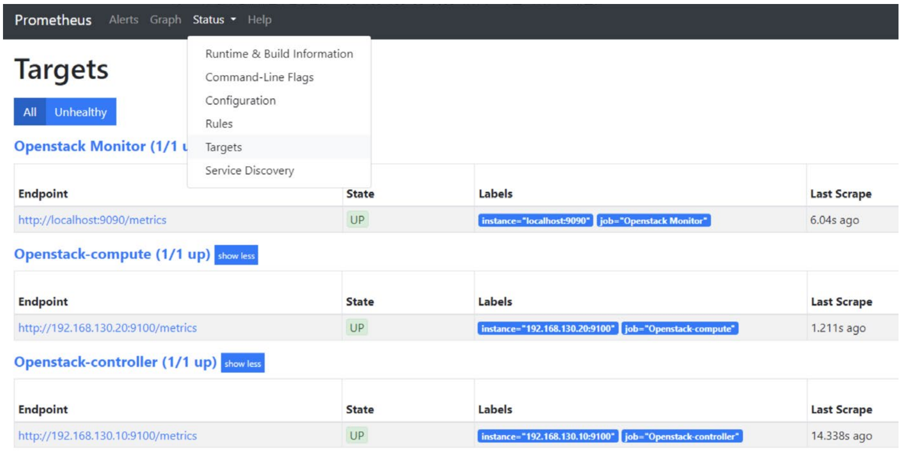

Figure 7. Prometheus Monitoring Configuration Effect

3. Grafana Visualization Configuration: Automatically create an "Intelligent Medical Assistant Platform Monitoring Dashboard" through Ansible's API call module, and import Prometheus data sources to realize real-time pull of indicator data. The final monitoring effect is shown in Figure 8.

Figure 8. Grafana Monitoring Visualization Effect

4. Conclusion

Aiming at the infrastructure bottlenecks and deployment efficiency issues in the implementation of AI services in the intelligent medical scenario, this paper proposes and implements an OpenStack + DeepSeek intelligent medical assistant private cloud platform based on Ansible automated deployment. Through the technical path of "automated construction of cloud infrastructure - standardized deployment of AI models - full-link monitoring visualization", it effectively solves the pain points of traditional medical AI platforms such as complex deployment, high operation and maintenance costs, and insufficient service stability.

Experimental verification shows that the platform meets the medical-grade application standards in terms of availability, scalability and security, and can effectively make up for the shortage of uneven distribution of high-quality medical resources, providing stable support for health consultation in remote areas and public health emergencies. Future research can focus on two directions: one is to introduce a federated learning framework to realize the secure collaborative training of multi-institutional medical data and improve the diagnostic accuracy of the model; the other is to optimize the resource scheduling strategy between OpenStack and AI models, and realize predictive resource expansion combined with user access heat maps to further reduce service response time.

Acknowledgements

I would like to express sincere gratitude to Wenzhou Polytechnic for its strong support. The superior academic atmosphere, advanced laboratory equipment and abundant library resources provided a solid foundation for this research, ensuring the smooth progress of experimental verification and literature collation.

My thanks also go to all contributors of the references. Their in-depth research on Ansible automation, OpenStack private cloud and intelligent medical technology provided important theoretical guidance and research ideas, helping me clarify the direction and avoid detours.

References

[1] Sun, C. Y. (2019). Research on Ansible Deployment and Automated Management Based on Linux Cluster. China Management Informatization, 22(9), 165-166.

[2] Lu, B. F., & Su, M. R. (2017). Research and Design of Web-Based Automated Operation and Maintenance System Based on Ansible. Digital Technology and Application, (12), 151-152.

[3] Fu, Y. Y., Liu, Y., & Yan, Z. W. (2018). Design of Private Cloud Computing Platform. Electronic Technology & Software Engineering, (12), 67.

[4] Yang, X. F. (2019). Application of OpenStack in the Construction of Private Cloud for School Teaching. Computer Programming Skills & Maintenance, (5), 97-98, 110.

[5] Yao, J., & Yan, N. (2019). Design and Implementation of OpenStack High-Availability Distributed Storage Scheme. Computer Technology and Development, (2), 35-38.

[6] Malik, A. S., Boyko, O., Atkar, N., & Young, W. F. (2001). A Comparative Study of MR Imaging Profile of Titanium Pedicle Screws. Acta Radiologica, 42, 291-293. https://doi.org/10.1080/028418501127346846

[7] Hu, T., & Desai, J. P. (2004). Soft-Tissue Material Properties under Large Deformation: Strain Rate Effect. Proceedings of the 26th Annual International Conference of the IEEE EMBS, San Francisco, 1-5 September 2004, 2758-2761.