Tools of Centralized Management Platform Based on Puppet and Ansible

Lin Qiao¹, Jue Bo¹, Linlang Guo², Fei Hu¹, Nan Hu¹ and Ran Ran¹

¹State Grid Liaoning Electric Power Co.,Ltd. Information Communication Branch, Shenyang City, Liaoning Province, China
²State Grid Liaoning Electric Power Supply Co. Ltd. Meterring Center, Shenyang City, Liaoning Province, China

*Corresponding author e-mail: ip31131508@163.com

Abstract. With the development and expansion of enterprise groups, large enterprise groups are gradually moving towards multinational groups. Such enterprises have many subordinate units, frequent business exchanges, business types and business scopes all over the world, and there are frequent work exchanges around the world. In particular, the decentralization of funds makes the management of funds difficult. The management risk of funds is also relatively high with the characteristics of the group companies. For large enterprise groups, especially multinational groups, the relatively centralized management of group funds is a challenge. In order to meet the management needs of large enterprise groups, the concept of centralized fund management emerged as the times require, and it first appeared in large enterprises in western countries. In order to solve the management difficulties and realize the centralized management of the company group, etc., and for the management staff to achieve overall planning and consideration, this article specifically takes Group A as an example to carry out the investigation. Through investigation, it can be found that the company's fund precipitation is very obvious, the supervision is seriously insufficient, the capital utilization rate is extremely low at about 40%, and the maximum does not exceed 50%. Group A cannot obtain the financial information of each subsidiary and branch in time. There is a lag in reconciliation information with branches. In order to solve this difficulty, this article deliberately combined puppet and Ansible to study the tools of the centralized management platform, to find the direction for the company to achieve centralized development, and to further promote the development of the group.

Keywords: Centralized Management, Management Platform, Application Practice, Tool Research

1. Introduction

With the continuous deepening of the country's reform and opening-up policies and the continuous development of the socialist market economy, the business organization as one of the main players in
the market has also continued to evolve. The type of business has evolved from a single organization to a multi-member, large-scale, cross-regional feature. Industry group or multinational corporation organization [1]. Its organizational structure has become more and more complex, showing the characteristics of multiple member units, and the geographical distribution is intricate. Therefore, it is difficult to manage, it is difficult to make timely and effective judgments, and it affects the strategic investment and development of enterprise groups [2]. Puppet is now used by many large data centers and well-known IT companies. Due to its open source nature, companies are more convenient to use, simpler to transplant, and compatible with mainstream systems. Users only need to extend the corresponding functions of Puppet according to their needs [3]. In the current Internet tide, domestic manufacturers using Puppet include many large Internet companies such as Alibaba and Sina. Because the amount of data that these large Internet companies need to store is increasing, the demand for network equipment in their data centers is also increasing. The number of administrators is increasing, and the use of automated operation and maintenance tools such as Puppet is becoming more and more widespread [4-5]. In China, Huawei, one of the equipment giants, also announced in May 2015 that it had signed related cooperation agreements with Puppet Labs. Before Puppet was introduced to Huawei, the company used script files or internally developed small management tools for batch management of devices and servers, and management staffs were in a lot of trouble. But after Puppet was introduced to Huawei, the problem was quickly solved. This shows that centralized management is particularly important [6].

In foreign countries, Puppet's scope of application is even more extensive. All Fortune 100 companies and about half of the global top 1000 companies are Puppet fans. These companies use Puppet to implement server configuration, management, and automated management of software operations [7]. Cisco was very optimistic about Puppet when it first appeared, and gave out large investments in several rounds of financing. Because Cisco is one of the investors of Puppet, among the modules that come with Puppet, there are modules developed for Cisco, so that users can easily use Puppet for management when using Cisco equipment [8]. At the same time, Cisco's R & D personnel are further developing Puppet to provide special Puppet connection blocks for users who need custom functions. Other foreign companies such as Red Hat, Google, etc. are also carrying out Puppet research and development methodically [9].

This article analyzes the current management mode and finds the shortcomings of the current management mode. In order to solve the current management difficulty, this paper uses the two technologies of Puppet and Ansible to implement the design and implementation of a centralized management platform. On this basis, Companies can change the current management mode, realize centralized management, improve work efficiency, and solve the difficult problem of management [10]. Promote the development of enterprises, and at the same time provide a basis for future centralized management.

2. Method

2.1 Ansible Framework

Ansible is a framework based on module work, which does not have the ability to execute and deploy itself. And each module that Ansible runs really has the ability to deploy and execute. The Ansible framework mainly includes:

1) connection plugins: responsible for communication with the monitored end.
2) Host inventory: The host of the specified operation is the host that needs to be monitored in the configuration file.
3) Various functional modules, such as service modules and cron modules.
4) Complete logging and other functions with the help of plug-ins.
5) Playbook: When a playbook performs multiple tasks, it is not necessary for the node to run multiple tasks at once.
2.2 Puppet Important Data Types

Puppet defines a lot of data types for internal communication, which will be mentioned in subsequent chapters, so this article chooses to put them here for simple explanation. Since they are the types necessary to complete all communication process stages, any other tool in Puppet can use these data types, so they are essential [9]. The following discusses several important data types:

1. Facts: The Facter command is used to collect system information on each host, generate Facts variables and pass them to the server. The collected system information will be used by the server to handle compilation and so on.

2. Manifests: files containing Puppet code, existing as "* .pp" files. In Puppet, Manifests are usually grouped and managed in modules.

3. Catalog: A graph of the interdependence between resources to be managed for a given host. Generally, it is compiled on the server side and sent to the client before it takes effect.

4. Report: After the client and the server communicate with each other, the report information is generated. It mainly contains key information such as authentication and configuration of both parties in this process, which is convenient for subsequent administrators to maintain. In addition to the above important data types, Puppet also supports other data types such as certificates, which will not be explained here. The most important Catalog data types are explained in detail below.

2.3 Puppet Architecture Analysis

There are two working modes inside Puppet: C / S mode and Stand-alone. Mode. The C / S mode is what is usually said. A central server and multiple clients. In Puppet, the server is called the master and the client is the agent. In the C / S mode, both the Master server and the Agent client need to run Puppet to ensure the normal work, while in the Stand-alone mode, a single process can complete all the work. In order to ensure consistency between working modes in a network environment, Puppet's internal network always maintains a transparent state. Therefore, under the current two working modes of Puppet, the Puppet framework requires the same code path regardless of whether it is through the network. The Agent client obtains the master service _compiled configuration_, generates a report and returns it to the master server after completing related operations. There are several benefits to this: □

1. Ensure the principle of least privilege. The Puppet framework stipulates that each Agent client can only get the information it is allowed, and the relevant information of any other host is not allowed to be obtained. In this way, each Agent client in a complex environment can be separated to ensure that they do not interfere with each other.

2. The user can completely separate the compile configuration file permission from the application configuration file permission. Generally speaking, Puppet will hand over the compilation of configuration files to the Master for unified processing, while the application configuration files are handled by the Agent clients. This may involve access to data center data storage, so this can ensure the security of the data.

3. When the Agent repeatedly applies a configuration to the client, Puppet allows the Agent client to disconnect from the Master server to perform the configuration. This can ensure that even if the Master server is closed or the Agent client is disconnected from it, the Agent client will keep the current configuration and operate normally without having to repeatedly obtain data from the Master server.

It is precisely because Puppet chose this architecture that its workflow becomes simpler and more efficient. In simple terms, puppet's workflow can be divided into the following parts:

1. Puppet's Agent client process will collect some host information required by the Master server (ie the facts variables mentioned earlier), and then pass this information to the Master server.

2. Puppet's built-in parser will use these host information and key information such as the Puppet local module to compile the configuration for a specific Agent client, and it will be passed to the specific Agent client after compilation.

3. The Agent client will also use Puppet's built-in parser locally to forward the configuration
information to the corresponding module, so as to change the configuration of the local host. Finally, the Agent client will generate a report based on the current situation and feed it back to the Master service, end.

3. Experiment

In order to make the research objectives more convincing, we deliberately sought a group limited company for investigation. Group A is a large cross-regional group. It has branches in many provinces in China. According to the business process of financial management of Company A within the company, all costs are uniformly paid after review by the regional company. Slow. In order to more clearly see the disadvantages of not using centralized management, we have deliberately investigated the financial utilization of Group A. You can directly see the problems existing in the company.

First, we investigate the average monetary capital, average current assets, proportion of current assets, sales revenue, and turnover rate of liquidity of Group A. The goal is to find the group's problems through precise data and find more precise solutions. Secondly, we investigated the group's bank account information to find out the actual funds in their account and compare them with the due funds in their account. Using the above algorithm to analyze and process the sample data to ensure the scientificity and accuracy of the experimental results.

Through our survey, after collating and analyzing relevant data, we have reached the corresponding conclusions and found that the company’s capital utilization rate is polar, and there is a lag in the reconciliation information between the head office and the branch; the funds are idle and the supervision is seriously insufficient.

4. Results

4.1 Experimental Results and Analysis

As shown in Table 1, Group A's monetary fund holdings in 2018 were 65,466.2 million yuan, accounting for 49.6% of current assets, and the currency holdings were excessively sufficient. Through a comparative analysis of relevant data from 2014 to 2018, it can be known that the currency funds holdings of Z Grid Corporation increased faster than the current assets, and its proportion in current assets also showed an upward trend. Due to the poor profitability of monetary funds, the funds that are idle inside the company will not be fully utilized, resulting in a low utilization rate of the funds and causing the company to suffer a certain loss of revenue. According to Figure 1, it can be seen that Group A cannot obtain the financial information of each subsidiary and branch in time, and there is a lag in the reconciliation information between the head office and the branch. Z Power Grid Corporation has nearly one hundred subordinate units, and each subordinate unit has opened accounts in multiple banks with balances, and has more idle funds and a relatively scattered distribution. The deposit of funds is very obvious, and the supervision is seriously insufficient.

| Table 1. | A Group's 2014-2018 Monetary Funds Statement (Unit / Thousand) |
|----------|---------------------------------------------------------------|
| Years    | 2014  | 2015  | 2016  | 2017  | 2018  |
| Average monetary funds | 56503540 | 65126510 | 64825170 | 67012350 | 65484620 |
| Average monetary funds | 142635620 | 156358520 | 157465230 | 145498740 | 132015680 |
| Percentage of current assets | 39.6% | 41.6% | 41.1% | 46.0% | 49.6% |
| Sales revenue | 82543250 | 84256540 | 94321460 | 98356420 | 99468510 |
| Liquidity turnover | 1.72 | 1.80 | 1.63 | 1.46 | 1.32 |
4.2 Suggestions for a Centralized Management Platform Based on Puppet and Ansible

(1) Compare the differences between the current configuration mode and Puppet configuration. By comparison, we can understand the shortcomings of the configuration methods under the current network management platform and the necessity of introducing Puppet software.

(2) The design plan was formulated according to the work needs. Combining Puppet and the features of the current platform, two alternatives are proposed, and finally a Puppet extension based on the NETCONF protocol is selected. This solution can minimize the time that users wait for the configuration to take effect and improve the user experience.

(3) Understand the working principle of Puppet. The most important thing in the development of Puppet is to understand the working principle of Puppet's implementation configuration. By reading the Puppet source code, master its working principle and configuration process, and lay the foundation for Puppet's porting and adding company's private Type and Provider.

(4) Port the Puppet software to the current network management platform. Although Puppet software has good compatibility, the current platform is not a mainstream system, but a tailored Linux system. Therefore, during the Puppet migration process, some simple modifications need to be made to the processes such as the Puppet authentication system type to ensure that Puppet can work properly on the platform. This part of the work mainly involves the modification of Puppet source code and the installation of some dependent packages.

5 Conclusion

This article strengthens the management and construction of the company by building a centralized management platform based on puppet and Ansible. Although there are still a few minor problems with Puppet, there is no doubt that it will be used on a large scale. Google, Cisco and other companies are using Puppet abroad. Many large companies in China have also started to use Puppet, such as Alibaba, Tencent and Sina. The use of Puppet by these large companies is an absolute affirmation of their capabilities, and it is also a strong support for the further improvement of Puppet functions. Therefore, Puppet will continue to maintain enough enthusiasm in the automated operation and maintenance market in the future, and has an optimistic outlook.

Acknowledgments

This work was financially supported by Research on improving technological innovation ability and mass innovation - State Grid Liaoning Electric Power Co., Ltd. information and communication branch (2019YF-65) fund.
References

[1] Robert Caiazzo, Gregory Baud, Guillaume Clément. Impact of Centralized Management of Bariatric Surgery Complications on 90-day Mortality[J]. Annals of Surgery, 2018, 268(5):1.

[2] Elkhatib Kamal, Lounis Adouane, Abdel Aitouche. Robust Power Management Control for Stand-Alone Hybrid Power Generation System[J]. Journal of Physics Conference Series, 2017, 783(1):012035.

[3] Mylaine Breton, Sabrina T. Wong, Mélanie-Ann Smithman. Centralized Waiting Lists for Unattached Patients in Primary Care: Learning from an Intervention Implemented in Seven Canadian Provinces[J]. Healthcare Policy, 2018, 13(14):65-82.

[4] Meigen Huang, Bin Yu, Sensen Li. PUF-Assisted Group Key Distribution Scheme for Software-Defined Wireless Sensor Networks[J]. IEEE Communications Letters, 2017, PP(99):1-1.

[5] mohammed Balfaqih, Mahamod Ismail, Rosdiadee Nordin. 802.21-Assisted Distributed Mobility Management Solution in Vehicular Networks[J]. IEEE Access, 2017, PP(99):1-1.

[6] Hao Song, Xuming Fang, Li Yan. Control/User Plane Decoupled Architecture Utilizing Unlicensed Bands in LTE Systems[J]. IEEE Wireless Communications, 2017, 24(5):132-142.

[7] Marco Maatman, Jeroen Meijerink. Why sharing is synergy: The role of decentralized control mechanisms and centralized HR capabilities in creating HR shared service value[J]. Personnel Review, 2017, 46(7):1297-1317.

[8] Ahhadreza Montazerolghaem, Mohammad Hossein Yaghmaee, Alberto Leon-Garcia. OpenSIP: Toward Software-Defined SIP Networking[J].IEEE Transactions on Network & Service Management, 2017, PP(99):1-1.

[9] Adam Mileczarek, Mariusz Malinowski, Josep M. Guerrero. Reactive Power Management in Islanded Microgrid—Proportional Power Sharing in Hierarchical Droop Control[J]. IEEE Transactions on Smart Grid, 2017, 6(4):1631-1638.

[10] Yi Zheng, Shaoyuan Li, Ruomu Tan. Distributed Model Predictive Control for On-Connected Microgrid Power Management[J]. IEEE Transactions on Control Systems Technology, 2017, PP(99):1-12.