LIFE CYCLE MANAGEMENT SERVICE FOR THE COMPUTE NODES OF THE TIER1, TIER2 SITES (JINR)

A.V. Baranov[1,a], A.O. Golunov[1], V.V. Mitsyn[1], I.A. Kashunin[1]

[1] *Meshcheryakov Laboratory of Information Technologies, Joint Institute for Nuclear Research, 6 Joliot-Curie, Dubna, Moscow region, 141980, Russia*

E-mail: [a] baranov@jinr.ru

Megascience experiments, such as CMS, ATLAS, ALICE, MPD, BM@N, etc., are served at the Meshcheryakov Laboratory of Information Technologies (MLIT) of the Joint Institute for Nuclear Research (JINR) using the available computing infrastructure. To ensure the guaranteed and stable operation of the infrastructure under constant load conditions, the centralized and timely maintenance of software and the rapid introduction of new compute nodes are required. As a solution to this task, a life cycle management service (LCMS) was created; its purpose is to automate the process related to the software maintenance and commissioning of new computing resources.
The paper will give an overview of the service and its components.

Keywords: grid computing, monitoring, configuration management system, control version system

Aleksandr Baranov, Alexey Golunov, Valery Mitsyn, Ivan Kashunin

1. Introduction

The depth and complexity of modern scientific knowledge in the field of nuclear physics require the involvement of many scientific institutions from different countries of the world. Experiments carried out in such collaborations are of megascience level. A high level of requirements is imposed on computer centers where important and valuable data is processed. The ability to scale resources, including through the prompt commissioning of new server hardware, is one of them. According to the profile of use, hardware can be divided into control or head machines of grid services, storage systems and working nodes (WN).

The commissioning procedure for such hardware comprises several stages: rack mounting; connection to a power supply and computer network; installation of the operating system (OS) and the required software, as well as its configuration.

The installation of hardware in large computer centers is usually handled by a separate service, and the system administrator is responsible for the rest of the stages. The following problems associated with a number of features of the operation of a computer center can prevent part of the work from being done promptly: availability of different profiles of hardware use; several servers can be located in one physical enclosure; hardware is usually added not within one server, but within a number of prepared and mounted racks.

The present article discusses a life cycle management service (LCMS) created to automate the process of installing and configuring the OS and software, as well as to simplify the WN administration process.

The service is based on the integration of various middleware, namely, Puppet [1], a well-established tool for centralized configuration management, Foreman [2], a set of tools for various tasks related to server maintenance, and GitLab, a web application for version control. A description of all the components used is given in Section 2.

2. Components

This section briefly provides an overview of the service components and describes the procedure for creating a Puppet manifest.

2.1 Puppet

Puppet is a platform-independent application, which works according to the Master-Agent architecture, it allows automating the management and configuration of the OS and software.

At the time of writing this article, the service serves more than 260 out of 1310 servers involved in data processing.

Puppet has two environments, tier1_wn and tier2_wn. WNs included in the structure of the Tier1 data center of the CMS experiment, LHC (CERN), are serviced in the tier1-wn environment. WNs that are part of Tier2 of the ALICE and COMPASS experiments, LHC (CERN), as well of the JINR Central Information and Computing Complex (CICC) for processing data, e.g. from the BMN, MPD and SPD experiments of the NICA megascience project and neutrino experiments, such as BES and JUNO [3], are serviced in tier2_wn.

For some configuration files, ssl certificates, ssh keys and others, it is required to save not only the content and access bits, but also the original timestamp of modification on all WNs. Puppet solves this task as well. The service monitors the list of files stored on separate machines and, when modified, copies them to the serviced nodes using the rsync utility. Direct access from WNs is prohibited for security reasons. Reports generated by Puppet when the Agent accesses the Master once every half hour are used as a surveillance tool.

Puppet manifests are automatically fetched from the repository into Git using GitLab CI [4]. More information on this will be presented in Subsection 2.4.

2.2 Foreman

Foreman is an open source life cycle system management solution for provisioning, configuring and monitoring physical and virtual servers.

It is used to provide the following functionality: dashboard showing Puppet's performance; automatic OS installation on the serviced servers via the network; display and storage of Puppet reports for 7 days.

There are a number of templates created to generate a configuration file used by the Anaconda OS installer. The templates contain various configurations of the disk space of all the serviced servers, as well as the required steps and their sequence when installing the OS for each machine.

The system administrator often faces the task to perform some actions on the administered servers. The "Remote Execution" functionality has been added to accomplish this task. It is used to run any command, script, etc. on the servers plugged to Foreman, as well as to force the launch of the Puppet Agent. Owing to the Foreman scheduler, the task can be performed on a one-time or regular basis.

Foreman supports several interfaces for operation: API, CLI, Web. The hammer utility is used to work with the service via the CLI interface.

2.3 Let's Encrypt

Free certificates provided by the automated and open Certificate Authority (CA) Let's Encrypt [5] are used for trusted web access via HTTPS and for the encryption of all communications between Foreman and the serviced servers. They are valid for 90 days. The added task in Cron on the server where Foreman is installed monitors the validity of the certificates and makes a request to the CA when approaching the expiration date to renew them.

2.4 GitLab CI

The mechanism of "Continuous Integration" in GitLab (GitLab CI) is used to control and facilitate the development of Puppet manifests. Figure 1 illustrates the workflow of the development of Puppet manifests.

The Git repository, which contains Puppet manifests, is logically divided into two branches: Master and Developed. By default, a commit to the Master branch is prohibited by access rights. After a commit is made to the Developed branch of the repository, Gitlab CI validates the Puppet manifest for syntax errors and copies it to the LCMS testbed. The testbed polygon is a full copy of the production service used for the development and testing of new functionality. The testbed is plugged to the physical server for tasks related to checking the manifest for correct work. If the Check stage is marked as done, "Merge Request" is made from the Developed branch to the Master branch. After approving "Merge Request", GitLab CI runs the second validation of the manifest and copies it to the internal catalog of the Puppet Master on the LCMS production service.

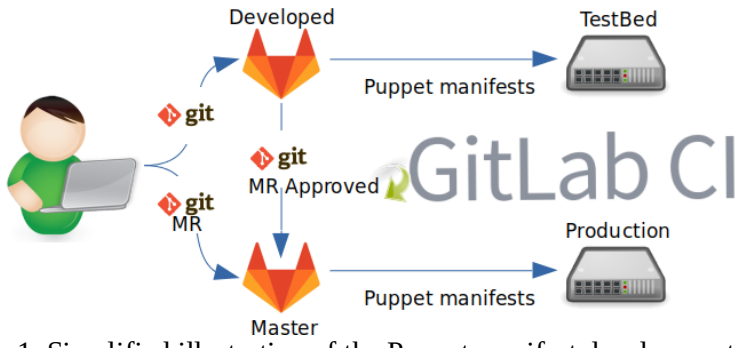

Figure 1. Simplified illustration of the Puppet manifest development process

3. Service monitoring

Monitoring is an important part of the LCMS service. This functionality is needed for operational control over the state of both individual components and the entire service as a whole. The monitoring of the service can be conditionally divided into three parts.

The first part is implemented on the basis of Prometheus [6], an open source solution for monitoring the software components of the LCMS.

The second part is based on the Wazuh [7] cluster system to search for malware, rootkits and suspicious anomalies, as well as to protect against network attacks, such as SSH brute-force.

The third part is based on Grafana [8], a web application for data visualization.

3.1 Prometheus

Prometheus is the general name for a technology that comprises a set of tools for monitoring both hardware servers and the installed software.

In a simplified version, the Prometheus installation consists of a time series database, various data exporters and Alertmanager. Prometheus send alerts to Alertmanager on the basis of rules specified in configuration files. Alertmanager alerts can be sent to different channels (e.g. email or messenger).

Prometheus, which collects metrics from exporters and Alertmanager, is serviced by the JINR cloud service [9] team.

Node_Exporter is used to collect a variety of OS and hardware metrics from the servers where the LCMS is deployed. StatsD_Exporter is utilized to collect metrics about the work of the Puppet Master. The data is taken from the telemetry module provided by Foreman.

All configuration files of Alertmanager are stored in the Git repository. Gitlab CI is used to operate with them for checking before applying them on the Alertmanager side. For operations, such as add/suppress notifications from monitoring, one must have access rights to the repository.

All notifications are delivered to the messenger channel for prompt action in the case of an incident.

3.2 Wazuh

Wazuh is a free, open source and enterprise-ready security monitoring solution for threat detection, integrity monitoring, incident response and compliance.

Wazuh is an offshoot of the pre-existing OSSEC product. It has been redesigned, and its capabilities for data visualization have been expanded by adding such components as Elasticsearch, Kibana, Filebeat, Wazuh (Master-Agent).

Daily reports are sent to email. They contain information about the checks performed, found vulnerabilities, errors in the OS and prevented network attacks. In addition to Kibana, Grafana is used to visualize data from the Wazuh system. This is done for the convenience of displaying various information about the operation of the entire service in a unified place. Data from Wazuh is collected from system logs using the Promtail collector and written to the Loki database.

3.3 Grafana

Data acquired from Prometheus and Wazuh is displayed in Grafana. Besides visual information, the ability to view the Wazuh log by the displayed event in the Grafana dashboard has been added. The Loki database is connected as a data source to Grafana.

4. Conclusion and further plans

By combining all the components considered in the article, we have managed to obtain a solution that will simplify and accelerate the implementation of new compute nodes under constant demand for novel resources.

The life cycle management service has been operating for more than six months and is successfully managed by one system administrator. Further plans are related to the transition of the rest of the servers to Puppet since it is planned to expand the use of the service not only for the working nodes, but also for the storage servers and head machines of the Tier1 and Tier2 sites. It will entail the creation of a description of the states of the services working on them for Puppet, as well as new templates for Foreman. Moreover, if the load on the service increases, the migration of Puppet and Foreman to the Kubernetes environment will be considered.

References

[1] Puppet Home Page, https://puppetlabs.com/ (accessed 21.07.2021)

[2] Foreman Home Page, https://theforeman.org/ (accessed 21.07.2021)

[3] A. Baginyan, et al. GRID AT JINR // CEUR Workshop Proceedings, ISSN: 1613-0073. 2019. Vol. 2507. P. 321

[4] Gitlab CI Home Page, https://docs.gitlab.com/ee/ci/ (accessed 21.07.2021)

[5] Let's Encrypt Home Page, https://letsencrypt.org/ (accessed 21.07.2021)

[6] Prometheus Home Page, https://prometheus.io/ (accessed 21.07.2021)

[7] Wazuh Home Page, https://wazuh.com/ (accessed 21.07.2021)

[8] Grafana Home Page, https://grafana.com/ (accessed 21.07.2021)

[9] A. Baranov, et al. Creating a Unified Educational Environment for Training IT Specialists of Organizations of the JINR Member States in the Field of Cloud Technologies // Modern Information Technology and IT Education, Springer. 2020. Vol. 1201. P. 149