APPROACHES TO THE AUTOMATED DEPLOYMENT OF THE CLOUD INFRASTRUCTURE OF GEOGRAPHICALLY DISTRIBUTED DATA CENTERS

P.V. Fedchenkov [1,a], N.Y. Samokhin [1], S.E. Khoruzhnikov [1], O.I. Lazo [1], A.Y. Shevel [1,2]

[1] *Department of Network and Cloud Technologies, ITMO University, St.-Petersburg, 197101, Russia*

[2] *National Research Centre "Kurchatov Institute" PETERSBURG NUCLEAR PHYSICS INSTITUTE, Gatchina, 188300, Russia*

E-mail: [a] pvfedchenkov@corp.ifmo.ru

University ITMO (ifmo.ru) is developing the cloud of geographically distributed data centers under OpenStack. The article describes the deployment cycle for the developed multi-agent system on physical and virtual machines using the developed software product repository. The cloud system of distributed data centers assumes the existence of a number of subsystems such as independent agents. An agent is aimed at implementing the creation, management and provision of certain services with a defined SLA. At the same time, it is necessary to perform both the initial deployment of the system and to provide the means that perform the automated modification of the hardware or software infrastructure, that is, the complete agent life cycle. The approaches suggested in the article allow us to use the minimum number of pre-installed components for the purpose of deploying and managing an array of physical and virtual entities of data centers, including deployment and configuring using Salt and monitoring the infrastructure using Zabbix.

Keywords: automated software deployment, service reliability, cloud of data centers, multi-agent systems

1. Introduction

Cloud computing paradigm has drastically changed the way in which we consume resources. Approaches to pay for the use and standardization of resources through virtualization lead to the need to use automation tools to reduce the cost of performing routine operations. At the same time, there is the task of deploying and monitoring a set of standardized entities with different parameters of service quality for a variety of private and public cloud clients.

In the previous paper [1], we have already considered approaches to deploying the HA configurations of the OpenStack platforms, so we use it to deploy program agents system for cloud management. In the paper, approaches of automated deployment preparation, base system deployment and storage cluster deployment are described. As a storage subsystem, we use CEPH platform, which is free, open source, and backed by much of the same community as OpenStack. The stability, scalability, and value of OpenStack and CEPH have been checked by large deployments like CERN [2].

The article is structured as follows: Section 2 describes goals and objectives of automated service deployment in data centers; Section 3 describes proposed approaches for automated deployment of services; Section 4 describes monitoring approaches of deployed services.

2. Tasks solved by automated deployment

Automation of deployment follows the formal definition of the sequence of actions required to configure and run virtual entities. At the same time, the input parameters are not only the available cloud hardware resources, but also the requirements for the quality of customer service. Deployment can be initiated by the virtualization management service upon request from the client interaction system [3].

Main benefits of automated deployment are:

1. Repeatability – every deployment deploys the same with various parameters in configuration;
2. Fewer errors – removing manual steps and passing the testing procedure reduces error;
3. Work on what matters – developers can spend time working on new features, not on fixes for manual deployments;
4. Lower costs – fewer errors, fewer human hours needed for deployment.

3. Proposed approaches for automated deployment

The deployment of cloud infrastructure and services requires a number of preparatory procedures. Therefore, a series of operations are performed to ensure the possibility of using automation:

1. Preparing repository accessibility;
2. Prepare DNS and DHCP;
3. Preparing of system images using separate repositories for different types of services such as storage, computing, networking, agents;
4. Preparing of monitoring and DB facilities and messages queue service;
5. Preparing Inventory DB for hardware information, MAC-addresses, logical distribution;
6. Preparing Salt Master node [4] for connecting clients, i.e. set the deployment patterns for each of the service options on Salt Minion.

The further deployment process is aimed at using information from this list of services in order to determine whether it belongs to a particular service variant and initiate the launch of the Salt deployment scripts.

3.1 Base system deployment

In order to use a VM for further deployment of virtual entities, it is necessary to create basic images. These images are used to create VMs running in OpenStack, by obtaining the necessary configuration of hardware resources, network environment and domain names. The preparation process includes the following steps:

1. Use separate project repositories for different types of services;
2. Deployment of base system (Naulinux distribution based on Scientific Linux);
3. Automating the installation of packages after deployment for agents using custom kickstart;
4. Installing Salt Minion packages and define required naming and address of Salt Master.

Using these prepared images when deploying OpenStack virtual machines, you can verify that the installed packages from a specific repository are present, the correct configuration of the Salt system, and proceed to further configuration.

3.2 Storage subsystem deployment

The project assumes the use of CEPH clusters as a storage system and provides users with access to block devices, or to the file system using NextCloud [5] software. In order to optimize the use of resources, the developed SaltStack modules filter the disks, group them for use as an OSD. Thus, the task is to generate SaltStack rules and start their synchronization. This expands the CEPH cluster to the user's subscription.

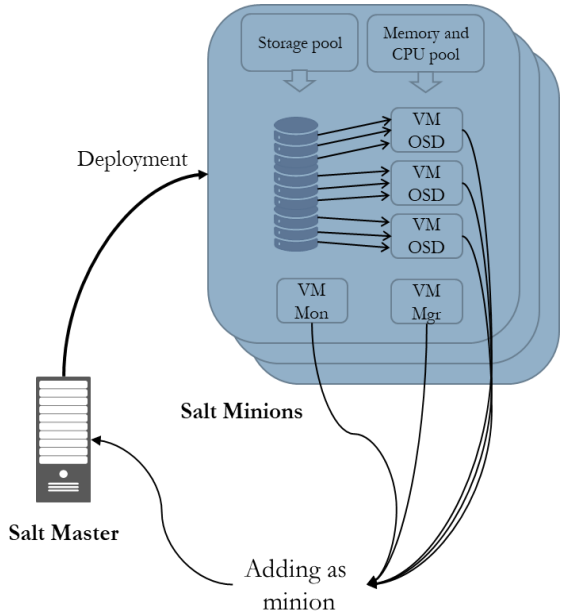

Figure 1. Deployment procedure for storage subsystem

A feature of our approach is the deployment of CEPH clusters for each client, which provides increased reliability of the storage. In this case, the virtual machines for cluster deployment and configuration are created using the OpenStack platform. The dedicated CEPH cluster connected to the Cinder OpenStack module is used as a backend for block devices of virtual machines. That is, block devices used as object storage devices (OSDs) for creating client storage are created in the CEPH cluster, which has its own replication. Figure 1 represent deployment procedure. Storage pool in the picture is the dedicated CEPH cluster; memory and CPU pools are provided by OpenStack platform for VM creation.

The approach of using CEPH in CEPH is aimed at solving the following tasks:

1. Providing increased reliability for user data due to an additional level of data replication;
2. Creating clusters of various sizes using block devices of a smaller size than the capacity of physical disks;
3. Improving the system's flexibility to change by using the level of abstraction of network interaction and hardware resources of the OpenStack platform.

A number of measurements aimed at assessing the performance of deployed clusters: we measure the speed of sequential and random data reading and writing using the rados bench utility [6]. The following results in table 1 were obtained for the dedicated CEPH and the CEPH client cluster (we use pool with replicas number of 1 for testing).

Table 1. Performance of dedicated and client`s CEPH clusters

	Sequential read, MB/s	**Random read, MB/s**	**Write, MB/s**
Dedicated CEPH	867.133	988.468	61.189
Client`s CEPH	988.047	1063.033	65.7225
Difference, %	12.3	7.1	6.2

So, a certain degree of degradation of the data exchange rate due to the use of nesting is visible, but this decrease in performance is within, for example, the percentage of losses for virtualization (10-15%).

4. **Monitoring deployed entities**

Virtual objects created during the automated deployment process need to be monitored for checking resource availability, utilization and for billing. Salt Stack is used to connect the created objects to the monitoring subsystem. With creation of virtual objects, such as CEPH cluster, or user`s gateway for data with Nextcloud instance, automated procedure add that entity into Zabbix.

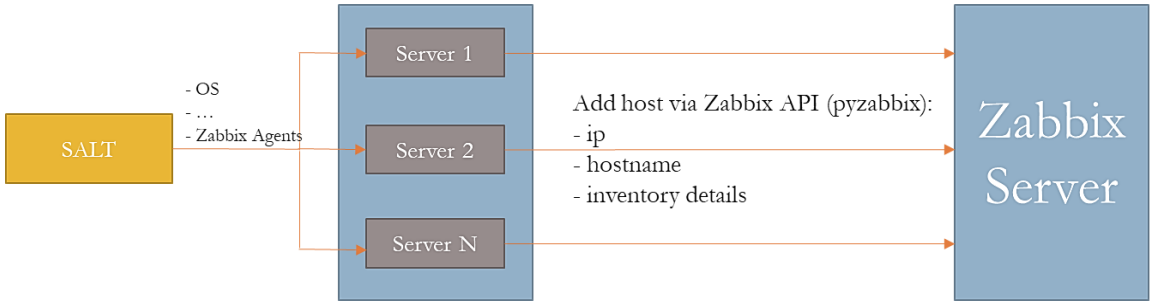

Figure 2. Provision of deployed entities into monitoring subsystem

Another task is to solve the problems that arise in the system. In general, the system may fail one of the servers and you need to determine which of the virtual objects were running on it before the error message appears. Modules have been developed that search for such information and assess the degradation of the system as a whole.

5. Conclusion

In the article, approaches of preparation and automated deployment are described. We use described methods in our project that aims on development management systems of geographically distributed Data Centers. The project is in active debugging stage now. The system is planned to be in preproduction stage first half of 2019.

6. Acknowledgment

The research has been carried out with the financial support of the Ministry of Education and Science of the Russian Federation under grant agreement № 03.G25.31.0229 “Development of new technological components for management systems of geographically distributed Data Centers”.

References

[1] Fedchenkov P.V., Khoruzhnikov S.E., Grudinin V.A., Sadov O.L., Shevel A.E., Kairkanov A.B., Lazo O.I., Oreshkin A.A. Service reliability in the cloud of data centers under Openstack // CEUR Workshop Proceedings – 2017, Vol. 2023, pp. 282-287

[2] T. Bell, B. Bompastor, S. Bukowiec, J. Castro Leon, M.K. Denis, J. van Eldik, M. Fermin Lobo, L. Fernandez Alvarez, D. Fernandez Rodriguez, A. Marino, B. Moreira, B. Noel, T. Oulevey, W. Takase, A. Wiebalck, and S. Zilli. 2015. Scaling the CERN OpenStack cloud. J. Phys. Conf. Ser. 664, 2 (2015), 022003. https: //doi.org/10.1088/1742-6596/664/2/022003

[3] Almutairi, A., Sarfraz, M., Basalamah, S., Aref, W., & Ghafoor, A. (2012). A distributed access control architecture for cloud computing. IEEE software, 29(2), 36-44.

[4] SaltStack Community. Available at: https://www.saltstack.com/resources/community (accessed 3 October 2018).

[5] Nextcloud. Available at: https://nextcloud.com (accessed 5 October 2018).

[6] Zhang X., Gaddam S., Chronopoulos A. T. Ceph distributed file system benchmarks on an openstack cloud // Cloud Computing in Emerging Markets (CCEM), 2015 IEEE International Conference on. – IEEE, 2015. – С. 113-120.