

# A Tool for Bulk Deployment of Similarly Configured Resources in VMware vSphere

Andrei Kuznetsov  
St. Petersburg Software Center  
Motorola Solutions Inc.  
Sedova st., 12  
St. Petersburg, Russia, 192019  
andrei.kuznetsov@motorolasolutions.com

Artyom Antonenko  
Institute of Computer Science and Technology  
Peter the Great St. Petersburg Polytechnic  
University  
29 Polytechnicheskaya st.  
St. Petersburg, Russia, 195251  
artonenko@gmail.com

## ABSTRACT

In this paper we consider a problem of bulk deployment of similarly configured resources in given resource provider. This problem often arises during testing of distributed enterprise applications, cloud services or infrastructure monitoring software. Manual deployment of virtual infrastructure is error-prone and time consuming when the infrastructure to be deployed is considerably large (e.g. contains more than 100 resource instances). Special bulk deployment assistant tools are very helpful for deployment of a large infrastructure to a resource provider. As a result of our examination of major bulk deployment use cases, we present a software implementation and its user interface. At this moment the tool supports configuring a virtual infrastructure in VMware vSphere, however the approach can be adopted for other resource providers (e.g. AWS, MS Azure, OpenStack, etc.).

## Categories and Subject Descriptors

K.6.2 [Installation Management]: Computing equipment management; D.2.5 [Testing and Debugging]: Testing tools

## General Terms

Experimentation

## Keywords

VMware vSphere, Bulk deployment, Virtual infrastructure

## 1. INTRODUCTION

Rapidly growing hardware virtualization technologies have enabled development of many different kinds of distributed applications, including clustered enterprise applications and cloud-based applications and services. Increasing capabilities of these applications bring on their implementation complexity and require intensive application testing. In many

cases testing of distributed applications requires special tools allowing developers to simulate high load conditions.

Let's consider testing of cloud provider (e.g. OpenStack, AWS, MS Azure, OpenShift, etc.) as an example. At some point test engineer might need to deploy 1000 resource instances (e.g. virtual machines in IaaS<sup>1</sup> cloud or application servers in PaaS<sup>2</sup> cloud) in order to test that a provider does not crash, there are no memory leaks or similar non functional problems, especially in a case if the number of deployed resources is considerably large. Another verification issue is related to the question whether a cloud service is scalable against available hardware resources. If there had been a need for implementing such a test, an engineer would have been allocating the available (virtual) hardware resources (for instance, about 100 or even more virtual machines). Then the testing engineer would have made cloud broker using all the available resources by deploying 10000 resource instances into the cloud. It's obvious that deploying such amount of resources into the cloud has to be automated in order to minimize manual routine work (and therefore, to minimize risks of deployment errors).

In this paper we consider VMware vSphere as a resource provider. The tool we describe in this paper is intended to support a software testing process in a company developing software for monitoring an infrastructure deployed in VMware vSphere. Let us note that the monitoring software is out of scope of this paper. In order to test this monitoring application it is required to deploy a homogeneous hierarchy of resources (which includes both resource pools and virtual machines) where each node of the hierarchy has a number of predefined parameters, while total number of nodes being more than 50. In the following sections we give overview of a resource hierarchy in VMware vSphere; we explain major use cases supported by our solutions and we investigate the issue of how this tool increases user performance and what are its major particularities against existing tools like *Veeam Backup & Replication*.

## 2. PROBLEM OF BULK DEPLOYMENT IN VMWARE VSPHERE

Virtual infrastructure deployment in VMware vSphere includes virtual machines and resource pools creation and configuration. Main settings of virtual machine are the follow-

<sup>1</sup>Infrastructure-as-a-Service

<sup>2</sup>Platform-as-a-Service

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

ICAIT '16, Oct. 6 – 8, 2016, Aizu-Wakamatsu, Japan.  
Copyright 2016 University of Aizu Press.

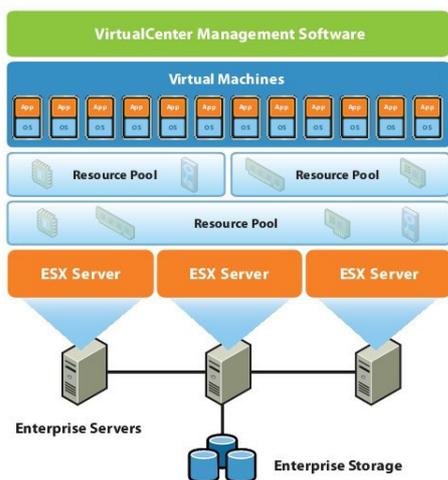


Figure 1: Resources hierarchy in VMware vSphere.<sup>4</sup>

ing: machine name, number of CPUs, memory size, virtual hard drive types and size[6]. The resource pool has only its name as one configurable parameter. If we have a few virtual instances (e.g less than 10 instances) it is not hard to configure them manually, but in a case of deploying a bulk (which might contain over 100 instances), manual configuration is time and effort consuming and might provoke errors.

In the following subsections we explain elements of VMware vSphere virtual infrastructure and standard deployment tools available for end users.

### 2.1 Resources hierarchy in VMware vSphere

Figure 1 demonstrates typical deployment of *vCenter Server*. The bottom layers (Enterprise Servers and Enterprise Storage) represent hardware; the middle layer (ESX Server) represents hypervisor software (or hardware) managing virtual resources; everything above hypervisor layer are virtual resources[5]. As we can see from the figure, the structure of *vCenter Server*'s virtual environment is hierarchical. There are resource pools which may include other resource pools and virtual machines[4]. In this way our challenge is to deploy a multi-tier structure with certain degree of nesting.

### 2.2 Standard VMware vSphere clients

In *vCenter* there is no user interface: in order to create and manage a *vCenter* infrastructure a number of special tools are provided by *VMware* including the following:

- *vSphere* desktop client;
- *vSphere* Web client;
- *PowerCLI* toolkit.

Table 1 summarizes advantages and drawbacks of the above mentioned tools in a case of being used for bulk deployment. As we can see, deploying a bulk of virtual resources with similar configuration is possible by using *PowerCLI* toolkit tools invocations in a special shell script [3]. Despite the script can generally solve the deployment problem, there are some issues which we have to consider from viewpoint of deployment tools usability and their user-friendliness:

<sup>4</sup>[http://www.assyrus.it/media/immagini/vmware\\_virtual\\_infrastructure.jpg](http://www.assyrus.it/media/immagini/vmware_virtual_infrastructure.jpg)

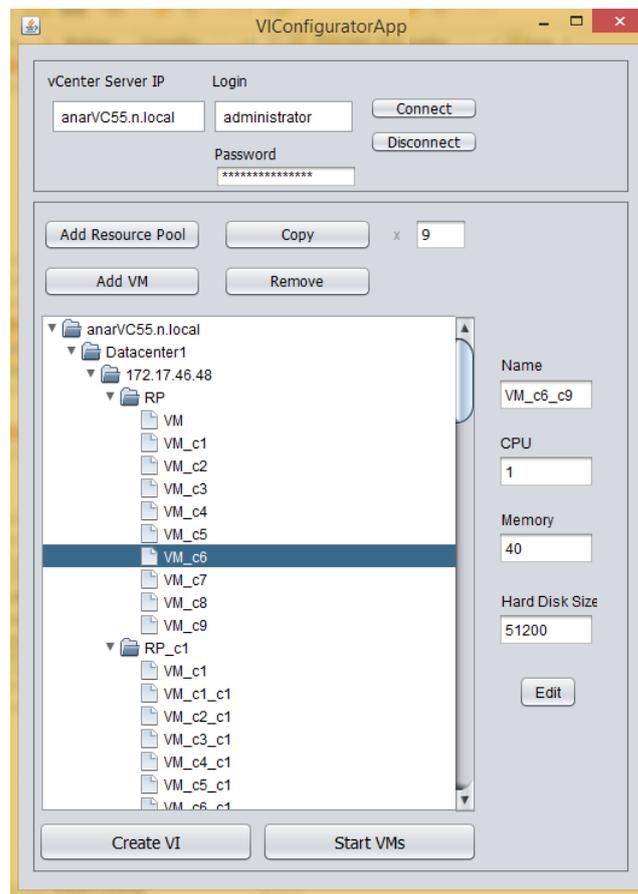


Figure 2: Main window of the developed application.

- A user has to specify a desired *hierarchical* infrastructure by using command line arguments or a configuration file. If we consider a single-level hierarchy of virtual machines only, the user has to specify four arguments in order to configure main parameters of a virtual machine (virtual machine name, number of CPUs, memory size, hard disk size). One additional parameter is required to specify number of virtual machines. If we consider a case of more levels of hierarchy, script configuration might be much more complicated and a user will required to provide even more script parameters.
- A user is unable to see a hierarchy to be deployed before it is deployed (so as to check possible errors).
- Usually for an end user it is easier to learn a new tool which provides a graphical user interface (GUI) rather than a tool with no GUI<sup>5</sup>.

<sup>5</sup>The best example of this statement in our practice is *vi* console text editor on Unix. It is usual that, once a user opens this editor, he or she asks about this text editor "How do I exit *vi*?". We have never heard about *notepad.exe* users on Windows wondering how to exit the *notepad.exe*.

**Table 1: Standard VMware vSphere clients and their applicability for bulk deployment.**

Tool	Advantages	Disadvantages
vSphere Desktop Client	There is a GUI	Bulk deployment requires too many manual routine user actions
vSphere Web Client	There is a GUI. No need to install.	Bulk deployment requires too many manual routine user actions
PowerCLI	Bulk deployment automation.	No GUI. All configuration parameters (e.g. VM configuration) are passed as command-line arguments (it makes script invocation complicated and increase the risks of user errors).

### 2.3 Deploying large infrastructure with vSphere Client

For demonstration purposes we consider a problem of deploying  $N = 10$  virtual machines organized as  $M = 10$  resource pools and its solution with vSphere Client (procedures for desktop and Web clients are the same). The following are the steps to be followed by a user:

1. Connect to vCenter Server by using the server name (IP-address or DNS name), user name and password. As soon as the tool is connected to the server, vCenter Server hierarchy view is loaded with using data obtained from the vCenter Server.
2. Select the host in the vCenter Server hierarchy view and click “New Resource Pool” button;
3. Specify a name and location for this resource pool;
4. Select the host in the vCenter Server hierarchy view and click “New Virtual Machine” button;
5. Select the configuration for the virtual machine(Typical or Custom);
6. Specify a name and location for this virtual machine;
7. Choose the resource pool you have created;
8. Select a destination storage for the virtual machine files;
9. Choose virtual machine version;
10. Specify the future guest operating system;
11. Select the number of virtual CPUs for the virtual machine;
12. Configure the virtual machine’s memory size;
13. Configure network connections which will be used by the virtual machine;
14. Specify type of SCSI controller;
15. Select a virtual disk;
16. Specify the virtual disk size and provisioning policy;
17. Click the button “Finish”;
18. Repeat steps 3-17  $N = 10$  times.
19. Repeat steps 2-18  $M = 10$  times.

Note that steps 3-17 (=15 steps) are executed  $N \times M = 100$  times in total. Each step is 1 or 2 mouse clicks, so the user need to perform at least 1500 actions. Assuming that each action takes at least 1 second, the task cannot be completed faster than in 1500 seconds (25 minutes).

Despite that vSphere Client supports cloning of virtual machines (which allows the user to perform fewer actions), this option even more time consuming for the reason that clone operation takes 3 to 25 minutes to complete (depending on the virtual machine configuration). It’s easy to calculate that populating 100 virtual machines via clone function will take at least 300 minutes (5 hours).

We believe that the infrastructure configuration time can be reduced significantly by improving “clone” function in the following ways:

- Clone function should be quick (i.e. no modifications should be applied immediately)
- Clone function should be available for resource pools (not only for virtual machines)

The following sections describe our implementation of the “clone” function and provide evaluation of the proposed tool implementing quick “copy” operation.

## 3. MAJOR USE CASES SUPPORTED BY OUR IMPLEMENTATION

This section explains major use cases supported by our application. Suppose that all virtual resources to be deployed have similar configuration. Despite such an assumptions is effectively a kind of constraint, this constraint allows us to introduce a “copy” operation applied to the resources in virtual infrastructure (resource pools and virtual machines). By using the “copy” operation the application allows users to create any number of copies of a certain resource (and, what is especially important, all of its subresources) in a couple of clicks. The procedure is demonstrated in the following subsections.

### 3.1 Deploying $N$ Virtual Machines

The following are the steps to be performed by a user in order to deploy  $N$  (in our example we use  $N = 100$ ) virtual machines into a *vCenter Server* infrastructure by using the developed tool:

1. Connect to *vCenter Server* by using the server name (IP-address or DNS name), user name and password. As soon as the tool is connected to the server, *vCenter Server* hierarchy view is loaded with using data obtained from the *vCenter Server* (see Figure 2).

2. Select the host in the *vCenter Server* hierarchy view and click “Add VM” button;
3. Select the created virtual machine;
4. (Optionally) Change the virtual machine parameters if necessary (name, number of CPUs, memory size, hard disk size);
5. Set number of copies to a desired value (99 in our example);
6. Click “Copy” button;
7. Click “Create VI” button in order to deploy the just configured hierarchy to a remote *vCenter Server*.

### 3.2 Deploying $N$ Virtual Machines Organized as $M$ Resource Pools

The following are the steps to be followed by a user in order to deploy  $N$  (in our example we use  $N = 100$ ) virtual machines into a *vCenter Server* infrastructure under a condition that the VMs to be deployed are organized as  $M$  (we use  $M = 10$ ) resource pools:

1. Connect to *vCenter Server* by using the server name (IP-address or DNS name), user name and password. As soon as the tool is connected to the server, *vCenter Server* hierarchy view is loaded with using data obtained from the *vCenter Server* (see Figure 2).
2. Select the host in *vCenter Server* hierarchy view and create a resource pool by clicking “Add Resource Pool” button.
3. (Optionally) Change the resource pool name;
4. Select the created resource pool and create one virtual machine;
5. (Optionally) Change the virtual machine parameters;
6. Select the virtual machine and create copies by specifying a desired value (9 in our example) in the corresponding edit box. Then click “Copy” button;
7. Select the resource pool and create ( $M - 1$ ) copies by specifying a desired value (9 in our example) in the corresponding edit box. Then click “Copy” button. Resource pool is copied with all its subresources (i.e. 10 virtual machines), such that in total the infrastructure contains  $M = 10$  resource pools, where each resource pool contains 10 virtual machines;
8. Deploy the specified infrastructure by clicking “Create VI” button;

As a result of this scenario, a bulk of 100 virtual machines partitioned into 10 resource pools are created in the *vCenter Server* infrastructure.

## 4. IMPLEMENTATION AND EVALUATION

We implemented a bulk deployment assistant tool as Java 8 desktop application. For GUI implementation Java Swing library is used while for implementing communication with *vCenter Server* we use *VMware VI (vSphere)* Java SDK [1] library, version 5.5.

We emphasized two particular aspects while evaluating the developed application:

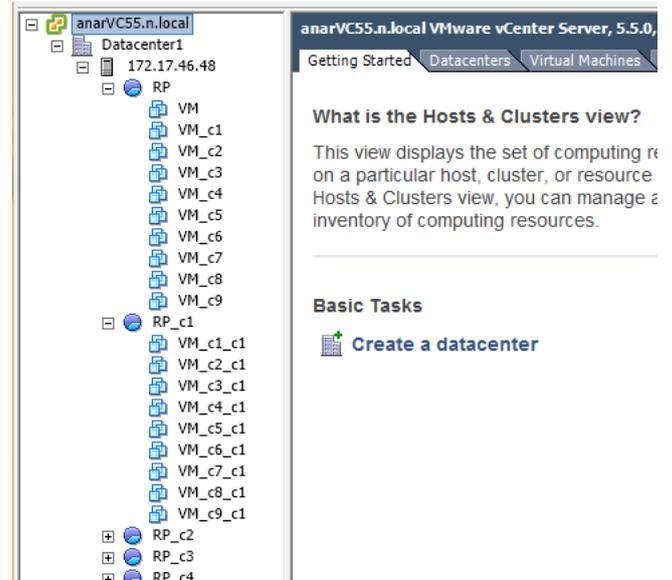


Figure 3: vSphere Client. View of the deployed infrastructure.

1. **Correctness.** This aspect is connected to checking whether an infrastructure created by a user using our application exactly matches a virtual infrastructure deployed into *vCenter Server*.
2. **User performance.** In order to go towards user efforts evaluation, we decided to compare the number of user actions (measured as mouse clicks) required to deploy a given infrastructure configuration into *vCenter Server* by using the standard *vSphere Client* against the case of using the developed deployment assistant tool.

In order to validate deployment correctness we use a manual scenario-based testing approach: the user configures an infrastructure with the help of our assistant tool, then vSphere Client is opened so as to allow viewing the actual infrastructure configuration just deployed. Figure 3 is an example of the actual infrastructure being observed with *vSphere Client* for a desired configuration presented in Figure 2. As you can see, the desired resource hierarchy matches the actual resource hierarchy, as well as each resource configuration within the hierarchy (e.g. virtual machine parameters) is as expected.

For user performance evaluation we executed a scenario (which is complex enough and which includes the resource pools and virtual machines configuration). This scenario was performed twice. In the first run we used the *vSphere* desktop client; in the second run we used the developed deployment assistant tool. The task was to deploy a virtual infrastructure with the following configuration: (root resource pool)  $\xrightarrow[\times 10]{\text{contains}}$  (child resource pool)  $\xrightarrow[\times 10]{\text{contains}}$  (virtual machine with a given configuration). In total there are 100 configured virtual machines and 11 configured resource pools. In the first run there were about 2500 mouse clicks, while in the second one there were about 30 mouse clicks. Such a tangible difference shows that the developed tool significantly simplifies bulk deployment of similarly configured

resources in *VMware vSphere* from the viewpoint of minimizing user efforts.

## 5. CONCLUSION

The tests we arranged for the developed application allow us to expect a significant reduction of efforts and user actions required in the time and effort consuming process for deploying virtual infrastructure in *VMware vSphere*. It is possible due to the implemented feature for massive copying of configured virtual machines and resource pools. The structure of virtual environment is represented in the form of a tree-like hierarchy which allows user to see the configured infrastructure.

Despite the described tool is a kind of highly specialized solution with limited functionality, it is appropriate for a concrete problem. We believe that our consideration is in a nice harmony with a Stroustrup's "clean code does one thing well"[2] – our tool does one thing well. The proposed application is not targeted to replace the existing virtual infrastructure configuration tools, it should be used side-by-side with the other useful tools.

After comparing our tool to a selection of existing *vSphere*-oriented tools we would agree that a problem of deploying a bulk can be solved by using replication features provided by some of them (for instance, *Veeam Backup & Replication* allows to deploy many instances of virtual resource from a backup image file). It is important to mention that the difference between our solution and the *Veeam Backup & Replication* tools (as well as other similar applications) is in defining the objectives. The primary objective of the *Veeam Backup & Replication* tools is to improve virtual infrastructure reliability and failure recovery time in a given virtual infrastructure. The primary objective of our tool is to configure a potentially huge number of resources with similar configurations within the context of a given resource provider, and *VMware vSphere* is the first one we targeted in our work<sup>6</sup>.

## 6. REFERENCES

- [1] S. Jin. *VMware VI and vSphere SDK: Managing the VMware Infrastructure and vSphere*. Prentice Hall PTR, Upper Saddle River, NJ, USA, 1st edition, 2009.
- [2] R. Martin. *Clean Code: A Handbook of Agile Software Craftsmanship*. Pearson Education, 2008.
- [3] VMware, Inc. *VMware vSphere PowerCLI User's Guide*. VMware, Inc, 3401 Hillview Ave. Palo Alto, CA 94304, 1 edition, 2013. EN-001263-00, Available at [https://www.vmware.com/support/developer/PowerCLI/PowerCLI55/doc/vsp\\_powercli\\_55\\_usg.pdf](https://www.vmware.com/support/developer/PowerCLI/PowerCLI55/doc/vsp_powercli_55_usg.pdf).
- [4] VMware, Inc. *vSphere Resource Management*. VMware, Inc, 3401 Hillview Ave. Palo Alto, CA 94304, 2 edition, 2014. EN-001584-00, Available at <http://pubs.vmware.com/vsphere-55/topic/com.vmware.ICbase/PDF/vsphere-esxi-vcenter-server-552-resource-management-guide.pdf>.
- [5] VMware, Inc. *vCenter Server and Host Management*. VMware, Inc, 3401 Hillview Ave. Palo Alto, CA 94304, 2 edition, 2016. EN-001520-04, Available at <http://pubs.vmware.com/vsphere-55/topic/com.vmware.ICbase/PDF/vsphere-esxi-vcenter-server-552-host-management-guide.pdf>.
- [6] VMware, Inc. *vSphere Virtual Machine Administration*. VMware, Inc, 3401 Hillview Ave. Palo Alto, CA 94304, 2 edition, 2016. EN-001518-03, Available at <http://pubs.vmware.com/vsphere-55/topic/com.vmware.ICbase/PDF/vsphere-esxi-vcenter-server-552-virtual-machine-admin-guide.pdf>.

<sup>6</sup>Support for other resource providers, including *AWS*, *OpenStack*, *OpenShift*, *MS Azure* etc. can also be provided in the future implementations.