**A DYNAMIC RESOURCE ALLOCATION SCHEME FOR VM REQUESTS**

**A PROJECT REPORT**

*Submitted by*

**M.ABARNA(12IT01)**

**R.KALAIARASI (12it31)**

**S.O.PRIYANGA (12IT72)**

*in partial fulfillment for the award of the degree*

*of*

**BACHELOR OF TECHNOLOGY**

*In*

INFORMATION TECHNOLOGY



**THIAGARAJAR COLLEGE OF ENGINEERING, MADURAI – 15**

(A Government Aided ISO 9001 : 2008 Certified Autonomous Institute Affiliated to Anna University

**ANNA UNIVERSITY: CHENNAI 600 025**

NONEMBER 2015

**THIAGARAJAR COLLEGE OF ENGINEERING, MADURAI-15**

(A Government Aided ISO 9001 : 2008 Certified Autonomous Institute Affiliated to Anna University)



**BONAFIDE CERTIFICATE**

Certified that this project report“**A Dynamic Resource Allocation Scheme For VM Requests"** is the bonafide work of “**M.ABARNA(12IT01), R.KALAIARASI(12IT31), S.O.PRIYANGA(12IT72) ”,** who carried out the project work under my supervision during the Academic Year 2015-2016.

**SIGNATURE SIGNATURE**

Dr. S. MUTHURAMALINGAM, M.E., Ph.D., Mrs.J.JOHN SHINY M.E,

**ASSOCIATE PROFESSOR &**

**HEAD OF THE DEPARTMENT ASSISTANT PROFESSOR**

INFORMATION TECHNOLOGY INFORMATION TECHNOLOGY

THIAGARAJAR COLLEGE OF ENGG, THIAGARAJAR COLLEGE OF ENGG,

MADURAI- 625 015. MADURAI – 625 015.

Submitted for the VIVA VOCE Examination held at Thiagarajar College of

Engineering on ………………

**INTERNAL EXAMINER EXTERNAL EXAMINER**

**ACKNOWLEDGEMENT**

We express our sincere gratitude to our Honorable Correspondent **Mr.Karumuthu T.Kannan** for his encouragement and guidance in our academic life at Thiagarajar College of Engineering.We wish to express our profound gratitude to our honorable Principal **Dr.V.Abhai Kumar** for his overwhelming support provided during my course span in Thiagarajar College of Engineering.

We are grateful to our department head **Dr.S.Muthuramalingam,** for all the academic help extended in our project. We thank our project supervisor **Mrs.John Shiny,** for all the guidance for the successful completion of this project and led us from the scratch of this project’s commencement with undivided attention and skillful expertise in developing this project in a systematic and professional manner

We express our sincere thanks to all the staff members and also the lab technicians of the Department of Information Technology for their constant and dedicated service to brighten our career.

We thank God Almighty and our parents for helping me procure such a challenging and an interesting project, and in completing the same in due course without much difficulty.

**TABLE OF CONTENTS**

|  |  |  |
| --- | --- | --- |
| **CHAPTERNO.** | **TITLE** | **PAGE NO.** |
|  | **ABSTRACT** | iv |
|  | **LIST OF FIGURES** | vii |
| **1.** | **INTRODUCTION** | 1 |
| **2.** | **LITERATURE REVIEW**  2.1. AN EFFICIENT DYNAMIC RESOURCE ALLOCATION STRATEGY FOR VM ENVIRONMENT IN CLOUD  2.2.MIGRATION OF INSTANCE FOR EFFICIENT RESOURCE UTILIZATION IN PRIVATE CLOUD 2.3.PRIORITY BASED DYNAMIC RESOURCE ALLOCXATION IN CLOUD COMPUTING WITH MODIFIED WEIGHTING QUEUE  2.4.STABLE RESOURCE ALLOCATION IN GEOGRAPHICALLY DISTRIBUTEDCLOUDS. | 3  4  5  6  7 |
| **3.** | **PROBLEM DEFINITION**  3.1 PROBLEM STATEMENT  3.2 PROBLEM DESCRIPTION | 8  9  9 |
| **4.** | **PROPOSED METHOD**  4.1.DYNAMIC LOAD BALANCING  4.2. MODULES  4.3. MODULE DESCRIPTION  4.3.1.USER MODEL  4.3.2. SYSTEM MODEL  4.3.3.MAIN CONTROLLER AND BALANCERS  4.3.4.CLOUD PARTITION LOAD BALANCING STRATEGY  4.5. DATA SHARING  4.6. PROPOSED SYSTEM | 11  12  13  14  14  14  14  15  15  16 |
| **5.** | **DESIGN OF THE PROJECT**  5.1. LOAD BALANCING  5.2. METRICS  5.3. CLASS DIAGRAM  5.4. COLLABORATION DIAGRAM  5.5. SEQUENCE DIAGRAM  5.6. USECASE DIAGRAM | 18  19  20  22  24  25 |
| **6.** | **SCREENSHOTS** | 26 |
| **7.** | **CONCLUSION AND FUTURE WORK**  7.1 CONCLUSION  7.2 FUTURE WORK |  |

**LIST OF FIGURES**

|  |  |  |
| --- | --- | --- |
| **FIGURE NO** | **DESCRIPTION** | **PAGE NO.** |
| 5.1. | LOAD BALANCING | 19 |
| 5.2 | METRICS | 20 |
| 5.3 | CLASS DIAGRAM | 22 |
| 5.4 | COLLABORATION DIAGRAM | 24 |
| 5.5 | SEQUENCE DIAGRAM | 25 |
| 5.6 | USECASE DIAGRAM |  |

|  |  |  |
| --- | --- | --- |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

**ABSTRACT**

Cloud computing is an on demand service.The major problem in cloud computing is fault tolerance. When many clients request the server simultaneously, server is overloaded which causes fault. In our proposed method, the dynamic load balancing technique is used to avoid this fault tolerant in cloud computing. The dynamic load balancing algorithm checks the utilization of CPU. If CPU has less utilization, it responses the client request otherwise the request is shifted to another server. Through which we reach the minimum average load among the host. Result, show that the time of migration process is very short. In this paper, we introduce a load balancing algorithm which eliminates load imbalance problem.

1. **INTRODUCTION**

With the rapid development of cloud computing, the datacenters are widely deployed in various areas. Connected by optical networks, the datacenters are always distributed unbalanced according to the service requirement of different regions. For the uncontrollable randomness of the requirements from large amount of clients to the servers, more tasks might focus on a small amount of datacenters, which would cause part of the network servers’ paralysis. Upgrading the hardware’s processing power is not a fundamental solution.What’s more, it can improve equipment cost rapidly

Therefore, it is very useful if the service which focuses on some datacenters can be dispatched to some other idle datacenters through migrating virtual machines dynamically. Through these operations, it will develop efficient use of data centers’ resources, improve the quality of users’ service, and prolong the life of each physical host computer. This dynamic scheduling strategy is based on a load balance algorithm. Using load balance algorithm, dynamic dispatching of virtual machine is more reasonable.

The load balancing algorithm based virtual machine’s dynamic migration scheme for datacenter application with optical networks, this paper proposes can satisfy the service demand of the front users and dispatch the resources of datacenter’s virtual machines. It combines load balancing algorithm to dispatch the virtual machine’s resource. After migration of virtual machines, each datacenter’s load is balanced so well that the service on it can be run freely. At the same time, the users from the front stage can only feel their service demands are fulfilled very well and have no idea of the operation at the backstage. In addition, the load balancing algorithm in this paper is based on the analysis for average variance of each node’s load in the cluster. And the algorithm is aiming at making every node’s load in a balanced way. Since variance is a way to measure the degree of each variable’s balance, the nodes could reach the best state when the cluster’s variance reaches the minimum.

1. **LITERATURE REVIEW**

**2.1 An Efficient Dynamic Resource Allocation Strategy for VM Environment in Cloud:**

Dynamic resource allocation system that uses virtualization technology to allocate resources dynamically based on application demands. We proposed resource management system that supports green computing and avoid overload by minimizing the number of servers used. We proposed the “skewness” concept to measure the un-evenness in the multiple resource utilization of a server and avoid overload by adding different workloads and optimize utilization of server resources. In this paper data center resources allocated dynamically by using virtualization technology and it depends on application request and optimizing the number of servers in use to support green computing. To minimize skewness here combine different types of loads. Skewness can be measured by Hot and Cold spot. Hot spot: If consumption of any resources is over a hot threshold. It specifies that the server is overloaded and some VMs running on it might be migrated. Cold spot: If consumption of resources is under a cold threshold. It specifies that the server is usually idle and it should be turn off to save energy.

**2.2Migration of Instance for Efficient Resource Utilization in Private Cloud:**

For Load balancing, migration of virtual machine is introduced. Migration of VM from one cloud platform to the other in an Infrastructure-as-a-Service Cloud service model ensures Load balancing. An approach is introduced for a framework to the VM migration problem, and proposes an egalitarian approach that finds a stable matching fair to both VMs and servers, instead of favouring either side as a result of the deferred acceptance procedure. In need of secure migration of VM, A Trust\_Token based VM migration protocol is introduced, which guarantees that the user VM can only be migrated to a trustworthy cloud platform. Different from previous schemes, our solution is not dependent on an active trusted third party. Various mechanisms have introduced for secure VM migration in cloud environments.

**2.3 Priority Based Dynamic Resource Allocation in Cloud Computing with Modified Waiting Queue**

The dynamic resource allocation based on distributed multiple criteria decisions in computing cloud. In it, the author’s contribution is tow-fold, first distributed architecture is adopted, in which resource management is divided into independent tasks, each of which is performed by Autonomous Node Agents (NA) in ac cycle of three activities: VM placement, in it suitable physical machine (PM) is found which is capable of running a given VM and then assigned VM to that PM, monitoring, in it total resources use by hosted VM are monitored by NA,In VM selection, if local accommodation is not possible, a VM need to migrate at another PM and process loops back to into placement. And second, using PROMETHEE method, NA carry out configuration in parallel through multiple criteria decision analysis. This approach is potentially more feasible in large datacenters than centralized approaches. A solution for dynamic scaling of web application provided by describing an architecture to scale web application in dynamic manner, based on threshold in a virtualized cloud computing environment. Architecture consists of front-end load balancer, a no. of web application virtual machine. In it apache HTTP Load Balancer is a front-end load-balancer for routing and balancing user requests to web application deployed on Apache HTTP server that are installed in Linux virtual machine. As per the demand these virtual machines are started and provisioned by a provisioning sub-system. But the action of provisioning and de-provisioning of web server virtual machine instances control by a dynamic scaling algorithm based on relevant threshold of web application.

**2.4 Stable Resource Allocation in Geographically**

**Distributed Clouds**

Today’s public cloud providers typically deploytheir small sized data centers in multiple geographically different locations, so as to improve data center power usage effectiveness and locate resources closer to users. A major challenge is resource allocation. Many results have been reported regarding this issue from the perspectives of virtual machine consolidation, networkaware virtual machine placement, traffic engineering, dynamic capacity provisioning, and so on. However, there has not been any focus on stable resource allocations, where no resource request or data center has any migration incentives. To the best of our knowledge, this paper is the first attempt at gaining a better understanding of the structure of the Stable resource Allocation (**SEA**) problem. We introduce a formal problem statement and develop two algorithms for the 1-dimensional (1-D) and 2-D cases, respectively. Simulation results show that the proposed algorithms have good scalability and convergence.

**3. PROBLEM STATEMENT AND DESCRIPTION**

* 1. **Problem Statement**

To overcome the problem on selecting underutilized virtual machine we propose a dynamic resource allocation scheme for efficient allocation of VM request.

**3.2 Problem description**

In cloud environment each PM can host number of VM andnumber of application can run on a single host. So load onthe VM is depends on the size and type of the application.Load on the host is the load of all VM running into that host,which can be change dynamically. So we proposed adynamic threshold based approach. Our approach use twothresholds that is upper and lower threshold to define the overloaded and underloaded condition. When the load onthe host is below the lower threshold, all VM running intothat host will be migrated which is known as serverconsolidation. When the load on the host is above the upperthreshold we migrate some VM so that host serve normalload. Four steps are involved in the VM migration.

**4. PROPOSED METHOD**

**4.1.Dynamic Load Balancing**

Dynamically adapts the allocation to loadchanges.Scales both in the number of physical machinesand sites and applications.Dynamically maximizing the cloud utility underCPU and memory constraints. For Load balancing in server, virtual machines are migratedfrom one host to another manually. The virtual machine livemigration is the process of moving running virtual machinesfrom one host to another host without stopping the client'sprocess. Thus distributing a workload to a multiple computingresources is load balancing. Here by, we achieves LoadBalancing in a data center by allocating workloads andmigrating a VM if over utilization or underutilization ofresources triggered in a Datacenter.

**4.2 Modules**

* User Module
* System Model
* Main controller and balancers
* Cloud Partition Load Balancing Strategy

**4.3 Modules Description**

**4.3.1 User Module**

In this module, Users are having authentication and security to access the detail which is presented in the ontology system. Before accessing or searching the details user should have the account in that otherwise they should register first.

**4.3.2System Model**

There are several cloud computing categories with this work focused on a public cloud. A public cloud is based on the standard cloud computing model, with service provided by a service provider. A large public cloud will include many nodes and the nodes in different geographical locations. Cloud partitioning is used to manage this large cloud. A cloud partition is a subarea of the public cloud with divisions based on the geographic locations. With the main controller deciding which cloud partition should receive the job. The partition load balancer then decides how to assign the jobs to the nodes. When the load status of a cloud partition is normal, this partitioning can be accomplished locally. If the cloud partition load status is not normal, this job should be transferred to another partition.

**4.4.3 Main controller and balancers:**

The load balance solution is done by the main controller and the balancers. The main controller ﬁrst assigns jobs to the suitable cloud partition and then communicates with the balancers in each partition to refresh this status information. Since the main controller deals with information for each partition, smaller data sets will lead to the higher processing rates. The balancers in each partition gather the status information from every node and then choose the right strategy to distribute the jobs.

**4.4.4. Cloud Partition Load Balancing Strategy**

When the cloud partition is idle, many computing resources are available and relatively few jobs are arriving. In this situation, this cloud partition has the ability to process jobs as quickly as possible so a simple load balancing method can be used. There are many simple load balance algorithm methods such as th Random algorithm, the Weight Round Robin, and the Dynamic Round Robin the Round Robin algorithm is used here for its simplicity.

**4.5 Resource Sharing**

Cloud computing is a technology that possesses sharingresources and applications rather than having local servers.It also provides sharing of multipleresources between several users at the same time by the use ofinternet.Physicalmachines can be divided in to number of virtual machines with thehelp of virtualization technique for addressing the efficiency of theutilization by sharing the resources of physical server intomultiple virtual machines.Migrating a VM to a PMwhere it can share memory with other hostedVMs can result in effective memory usage. Dynamic load balancing helps in resource sharing among multiple virtual machines during relocation of overloaded virtual machine to underloaded virtual machine.

**4.6 Proposed System**

We present the design and implementation ofan automated resource management system that achieves agood balance between the two goals. Two goals are overload avoidanceand green computing. Our load-balancing algorithm exhibits a fastconvergence rate.Allocate the chunks of files as uniformly aspossible among the nodes .No node manages an excessive number of chunks. Reduce network traffic caused by rebalancingthe loads of nodes as much as possible to maximize thenetwork bandwidth available to normal applications.Our proposal not only takes advantage ofphysical network locality in the reallocation of file chunksto reduce the movement cost but also exploits capablenodes to improve the overall system performance.Ensures fair resource allocation among sites andapplications.We consider computational andmemory resources and the objective is to achieve max-min fairness for computational resources under memoryconstraints. The resource allocation process dynamically and efficiently adapt to changes in thedemand for cloud services.The Resource allocation process must bescalable both in the number of machines in the cloud andthe number of sites that the cloud hosts. This means thatthe resources consumed per machine in order to achieve agiven performance objective must increase sub linearlywith both the number of machines and the number of sites.The components of the middleware layer run on allmachines. The resources of the cloud are primarilyconsumed by module instances whereby the functionalityof a site is made up of one or more modules. In themiddleware, a module either contains part of the servicelogic of a site or a site manager. It has two components: ademand profiler and a request forwarder. The demandprofiler estimates the resource demand of each module ofthe site based on request statistics. This demand estimateis forwarded to all machine managers that run instances ofmodules belonging to this site. Request forwardingdecisions take into account the resource allocation policyand constraints such as session affinity.Maingoal of our approach is to reduce the number of migrationand reduce the energy consumption that can we achievedthrough the proper load balancing.The VM placement in cloud computing are often created by considering one of the three classes:reservation, on-demand, and spot market. The hypervisor contains the list of all the physical resources available in the datacenters and information about their states. Since clients or users are provided with the computing resources only according to their requirement so the rest of it can be provided to other users. As multiple VMs can run on a single host at the same time, so, the large number of user’s requests are handled in a time and space efficient manner. The VMs are installed by means of hypervisors. A hypervisor is an operating system sits at the lowest level of the hardware environment. The instances of operating system i.e. the VMs are installed over hypervisor. A hypervisor provides a uniform abstraction of the underlying physical machine allowing multiple VMs to execute simultaneously. Resource Provisioning in cloud computing is the combination of both Resource allocation and Task Scheduling. The available resources on cloud are provisioned to the VMs in such a manner so that the resource’s tremendous capabilities are utilized efficiently and effectively without delaying in completion of tasks given by the cloud users. The service level requirements such as availability, capability of resources, performance and costs of resources must be qualified by the provisioning of resources. All kind of users must be taken into care by resource provisioning and it must fulfill the requirements of all its end users and providers.So, a developer of resource provisioning technique must take care of the needs of Cloud Users on one hand and Cloud Provider on the other hand. In this Dynamic Resource Allocation Algorithm, the virtualmachine under host, will schedule dynamically as per theuser's request. Resources will be provided to the user as perthe requirement of user. If the user wants to extend thememory, user can request for the additional resource as theyneed. Dynamically the resource will be given to the requestedvirtual machine and updated dynamically.

**5. DESIGN OF THE PROJECT**

**5.1 Load Balancing**

Load balancing algorithm to determine how to distribute the load among the servers. When a load balancer is configured to use the least response time method, it selects the service with the least number of active connections and the least average response time. Load balancing should take place when the load situation has changed. There are some particular activities which change the load configuration in Grid environment. The activities can be categorized as following:

• Selection of static or dynamic load balancing category.

• Defining the various parameters.

• Connection with the server.

• Sending threads to the server and executing results. For static load balancing first of all collect host information from user (i.e. ip address, port, request URL etc), when we execute this will try to connect to the host. If it get connected to host it will perform simulating the number requests to the host using no. of threads, then it will bring result from the server and populate into the view area. For dynamic load balancing, schedule the host information for dynamic execution, check the date and time for every second comparing the scheduled date and time, try to connect the host, simulating the number requests to the host using no. of threads, bring result from the server and populate into the view area.

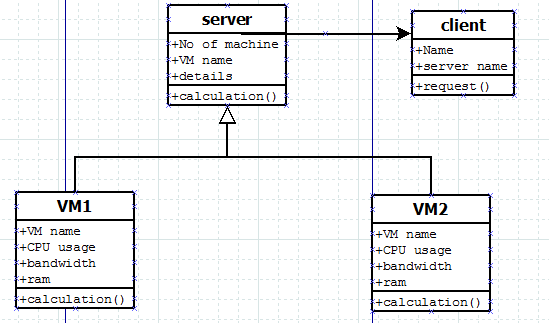
**5.2 Metrics**

From the load balancing algorithm, it is cleared that the maximize or minimize different performance parameters like CPU load, Memory capacity, Delay or network load for the clouds of different sizes.

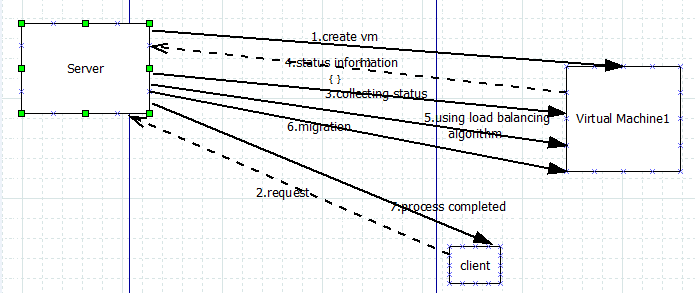
In principle, we suggest that a comprehensive benchmark should include the following staleness metrics:

* **Throughput:** Throughput is defined as the number of completed requests per second. The peak throughput is defined as the maximum throughput which can be sustained while successfully handling more than 99.99% of all requests.
* **Response Time:** Response time is the time interval between sending a request packet to a server and receiving the first response. Least Response Time load balancing allows you to distribute client requests across multiple servers. Load balancer improve server fault tolerance and end-user response time. Load balancing distributes client requests across multiple servers to optimize resource utilization.
* **Latency:** Latency is a measure of time delay experienced in a system, which is depends on the system and the time being measured. Latencies may have different meaning in different contexts. In simulation applications, 'latency' refers to the time delay, normally measured in milliseconds (1/1,000 sec), between initial input and an output. Latency is sometimes also called transport delay.
* **Downtime:**Live migration aim to minimize the downtime of virtual machine either by transferring pages before the machine gets suspended or copying minimal state (post-copy) to start the VM and using demand-paging over the network to fetch the remaining state. It starts when the first VM begins the stop-and-copy phase and ends when the last VM ends the resume phase. It depend on the order of the order of migration.

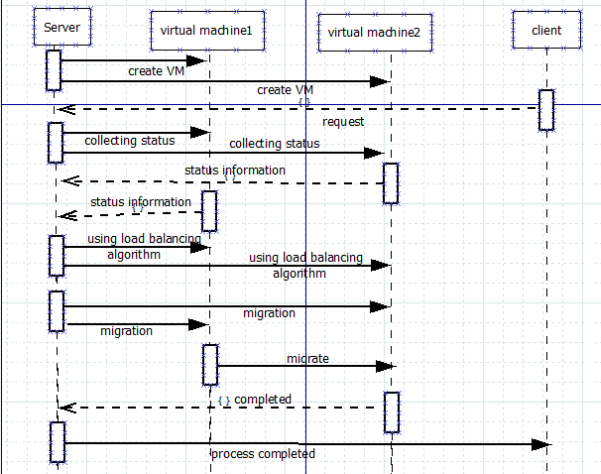
**5.3Class Diagram**



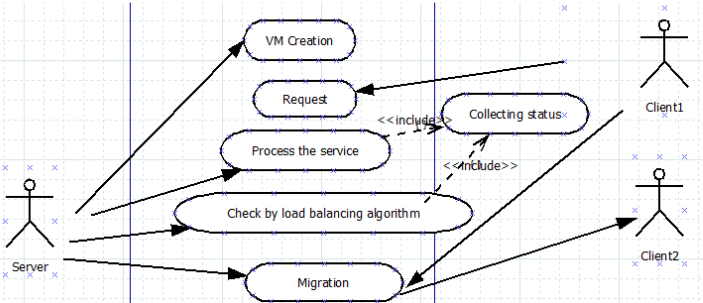
**5.4 Collaboration Diagram**



**5.5 Sequence Diagram**



**5.6 Usecase Diagram**

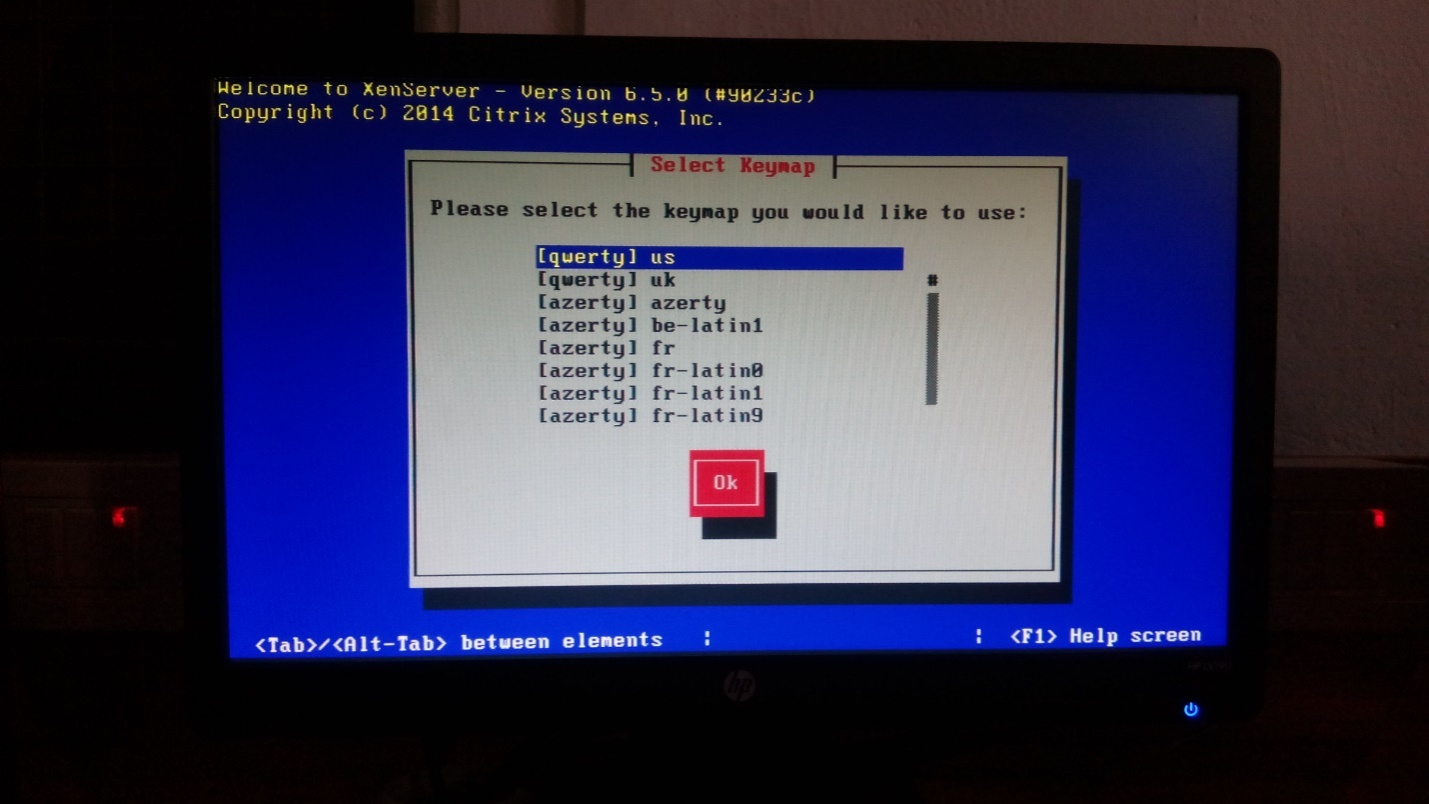


**6. SCREENSHOTS**

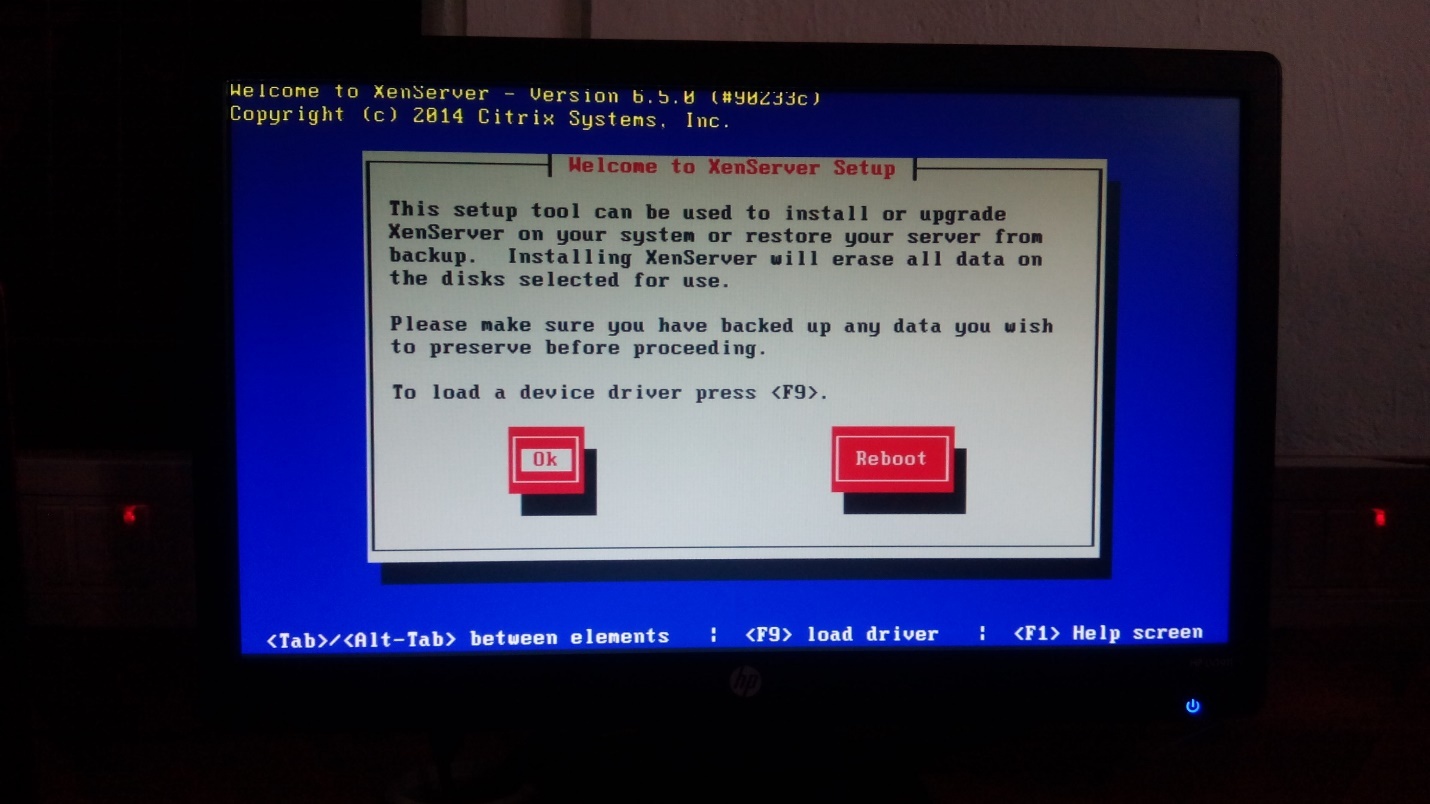
**Citrix Xen server Booting:**

This shows the seletion of keymap to use

**Select language:**

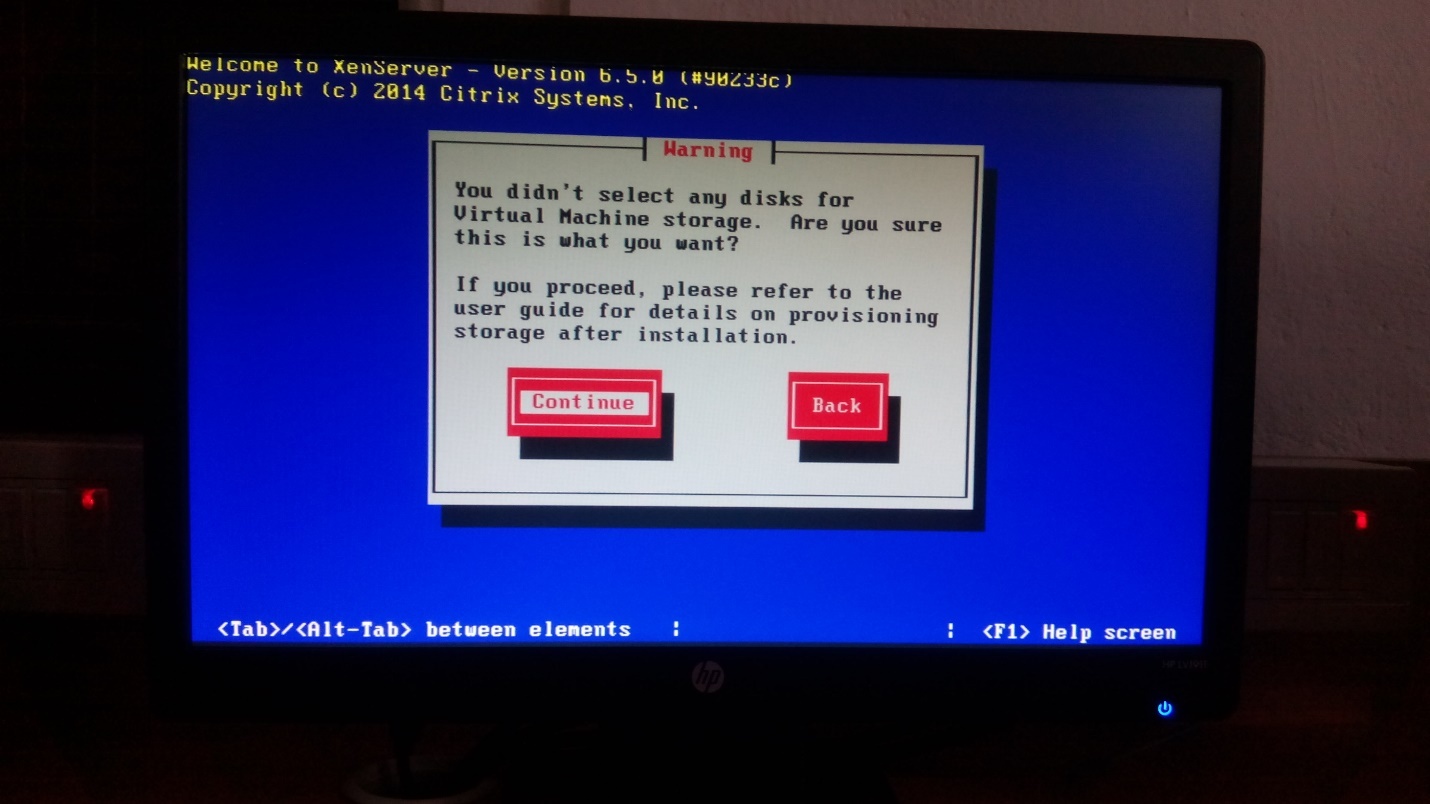


This image shows whether to load device driver or not:



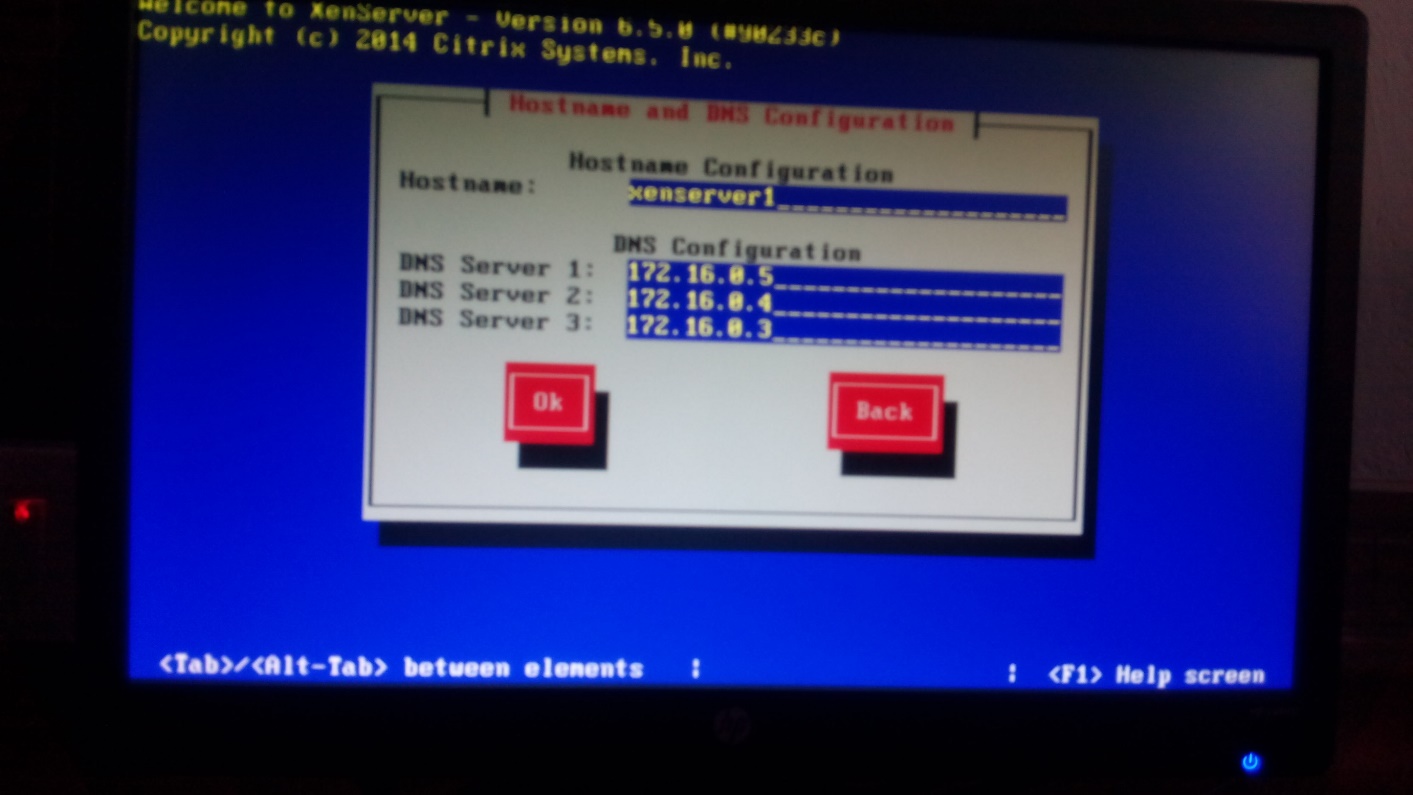
**Load device driver:**

This shows to select virtual machine storage

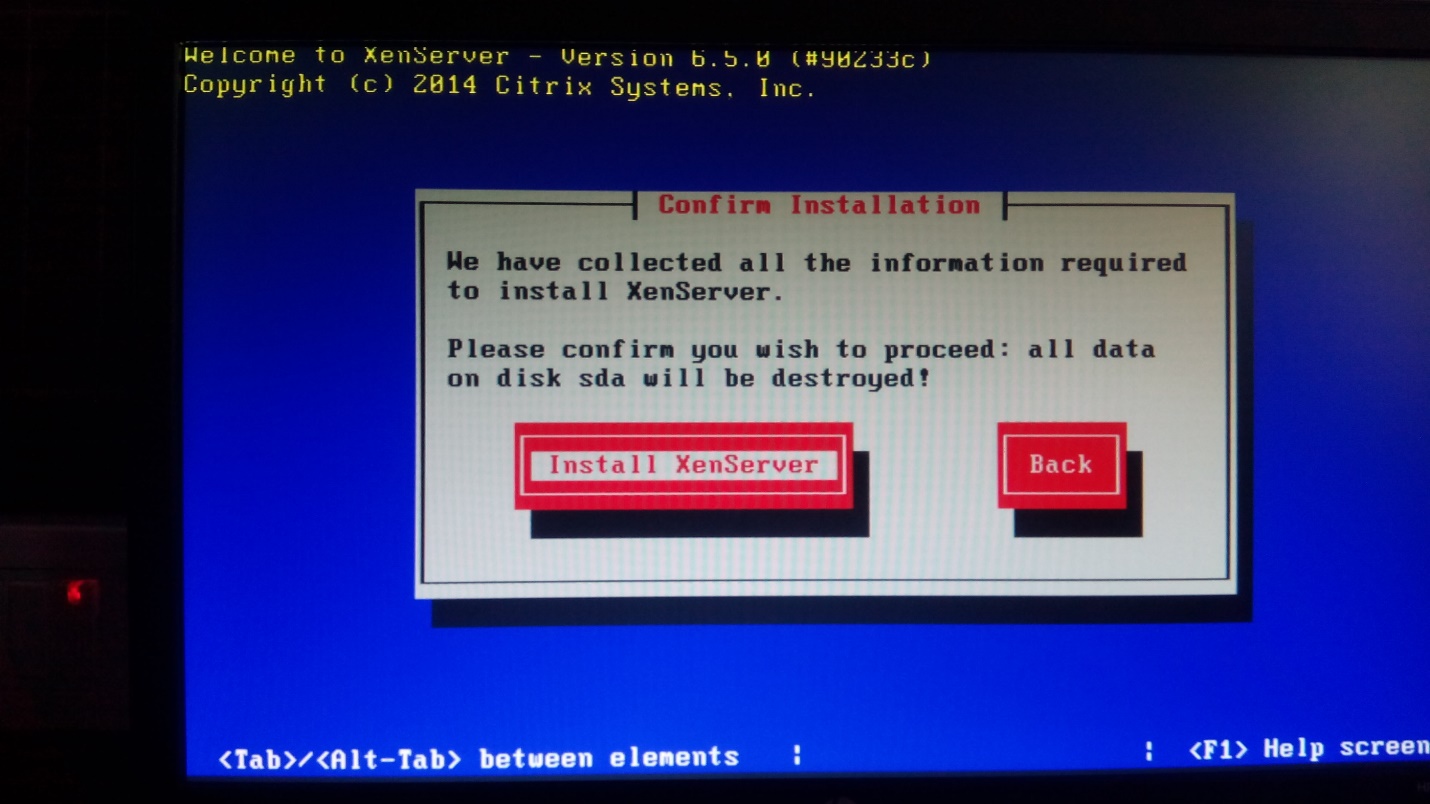


**Source are verified and Installation is Intiated:**

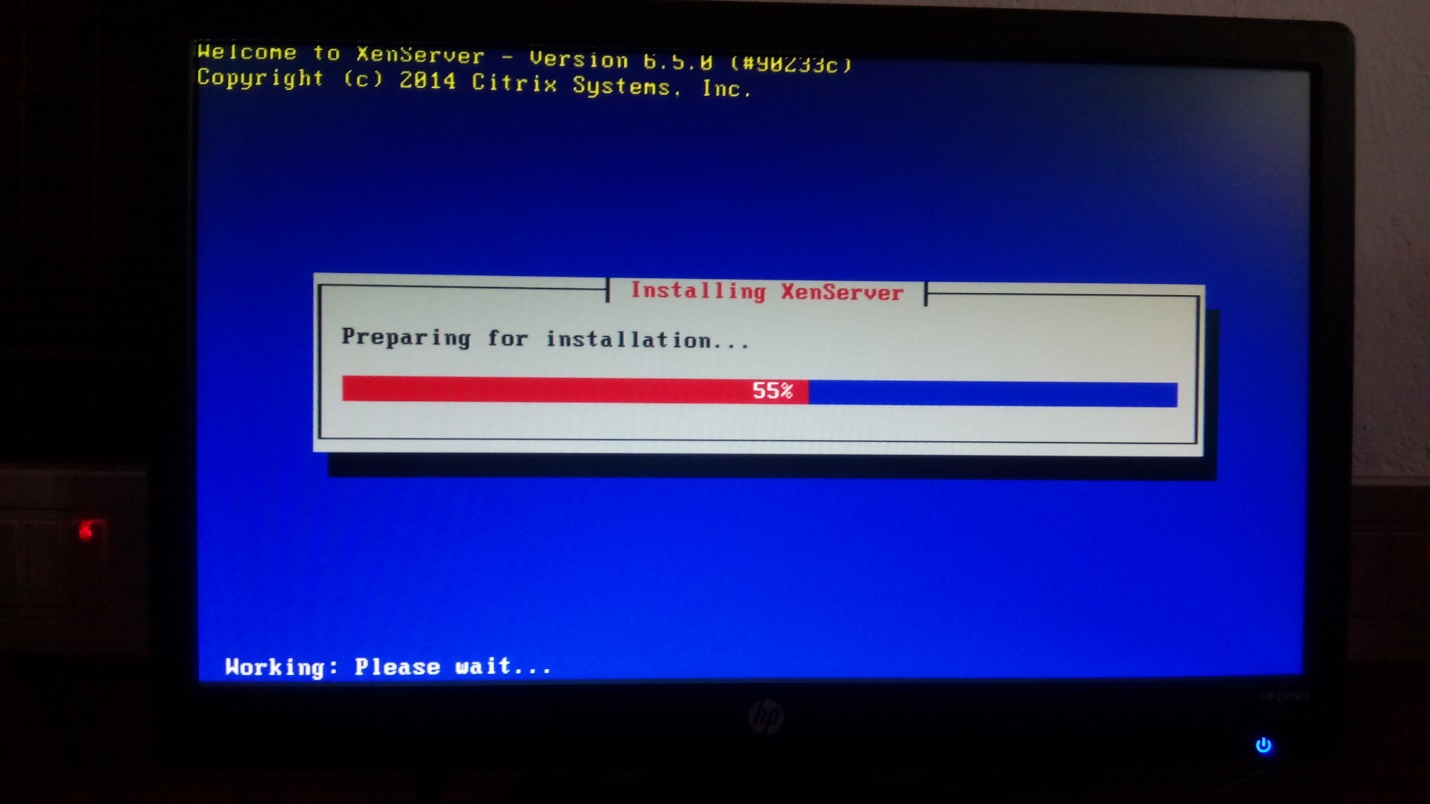
**Hostname and DNS Configuration:**

****

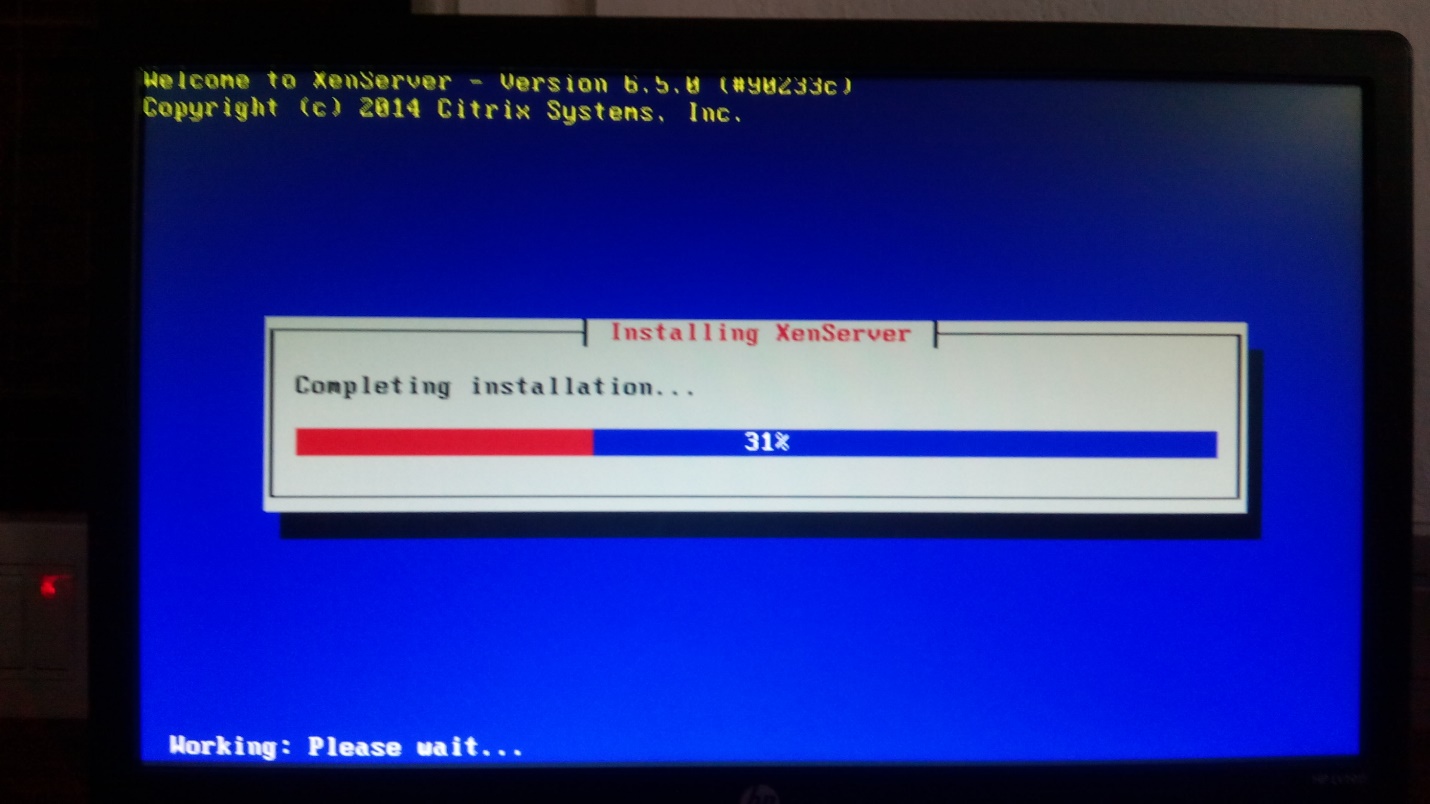
**Install XenServer**



**Installing Xen server:**



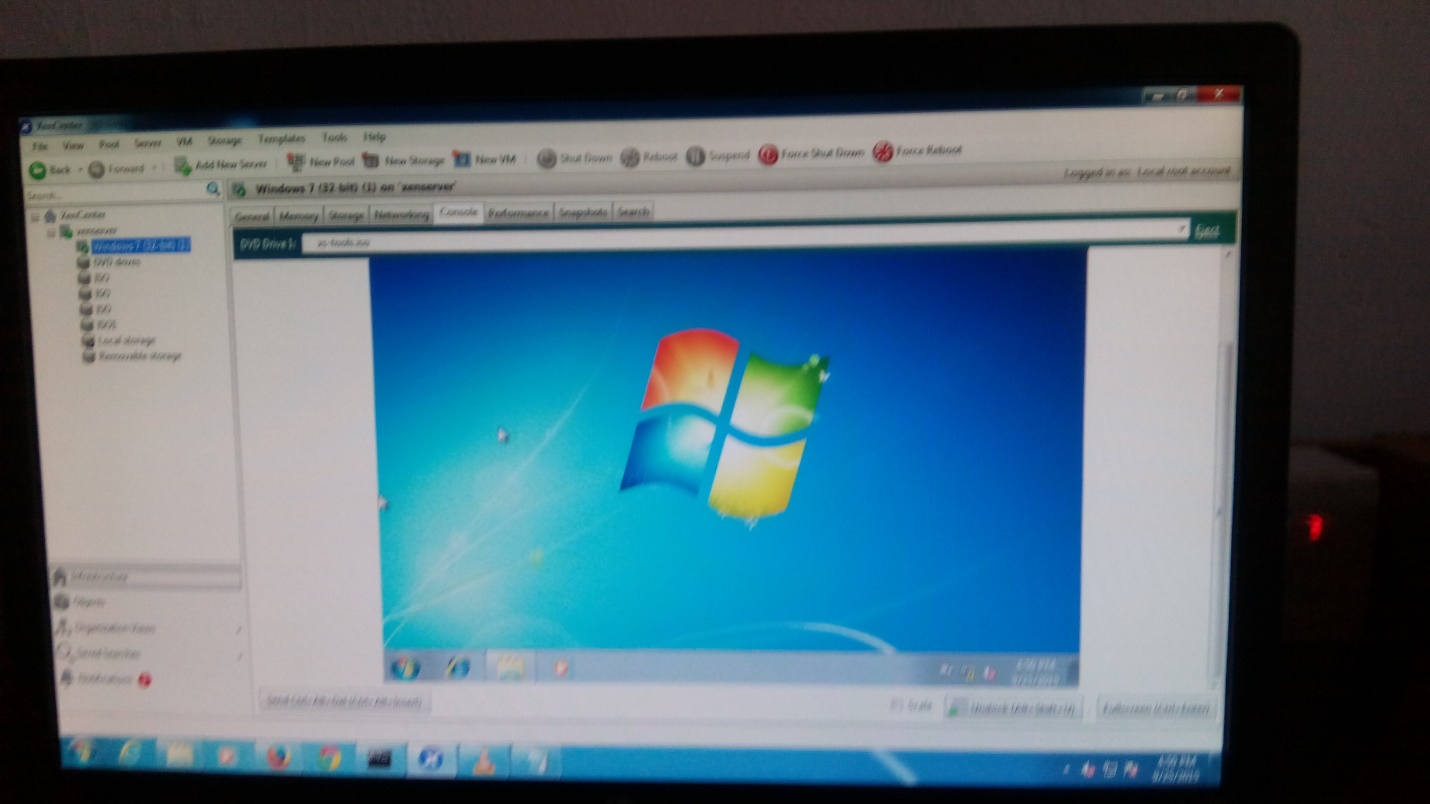
**Completing Installation:**

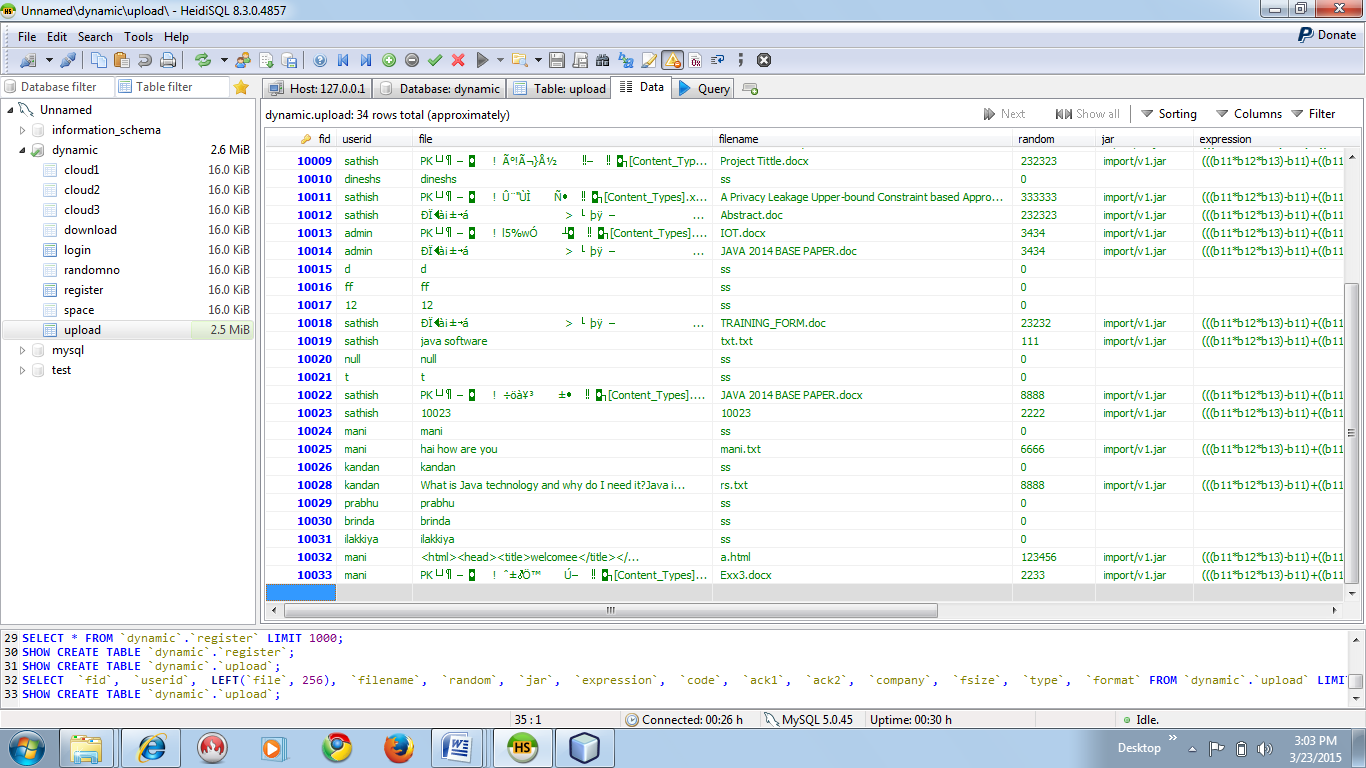


**Xenserver Configuration:**

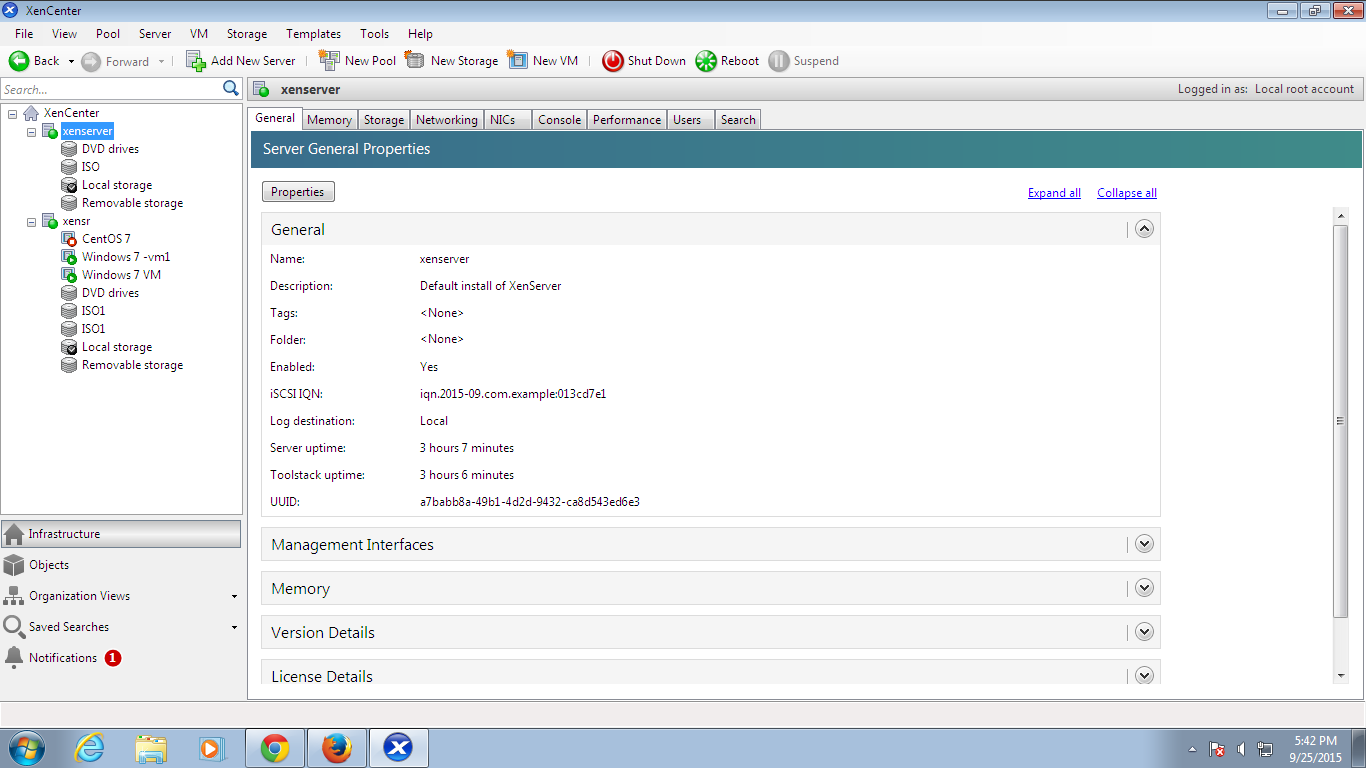


**Xen Center:**

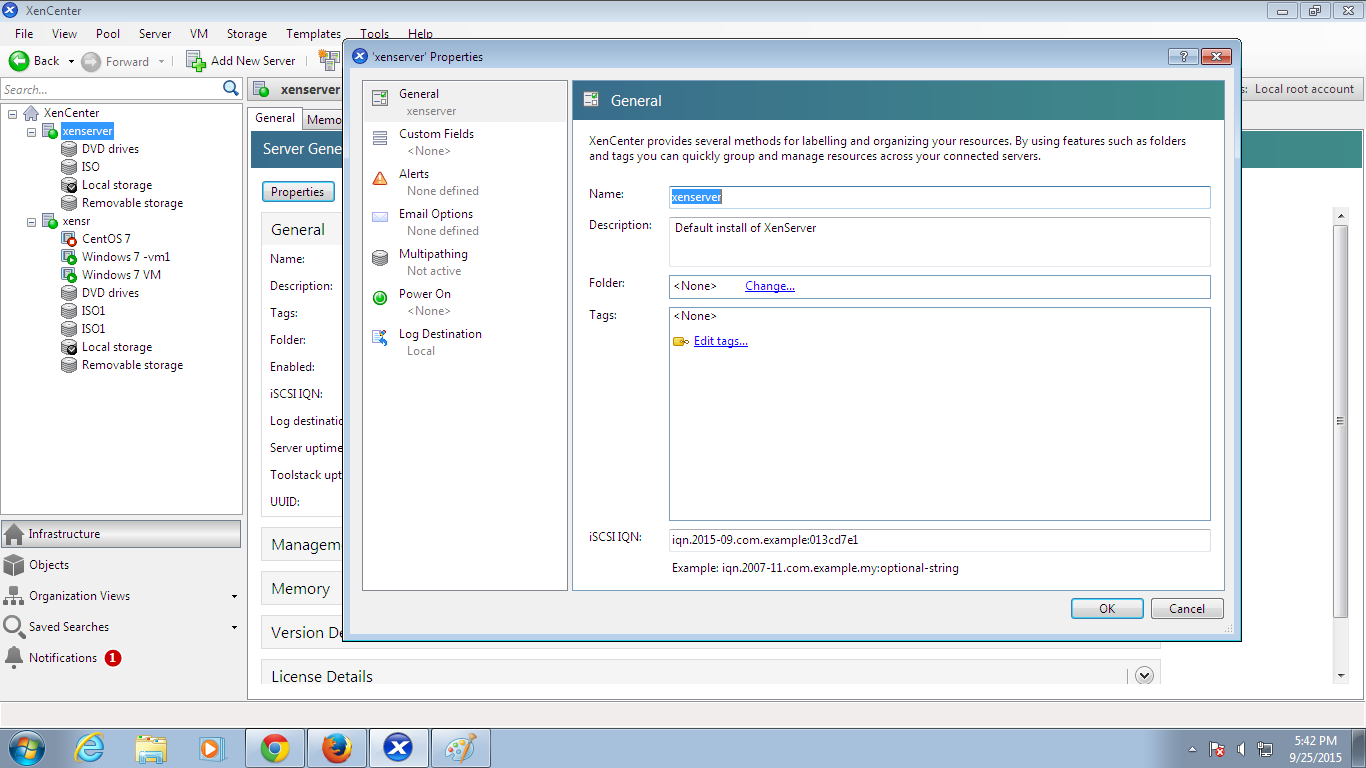


****

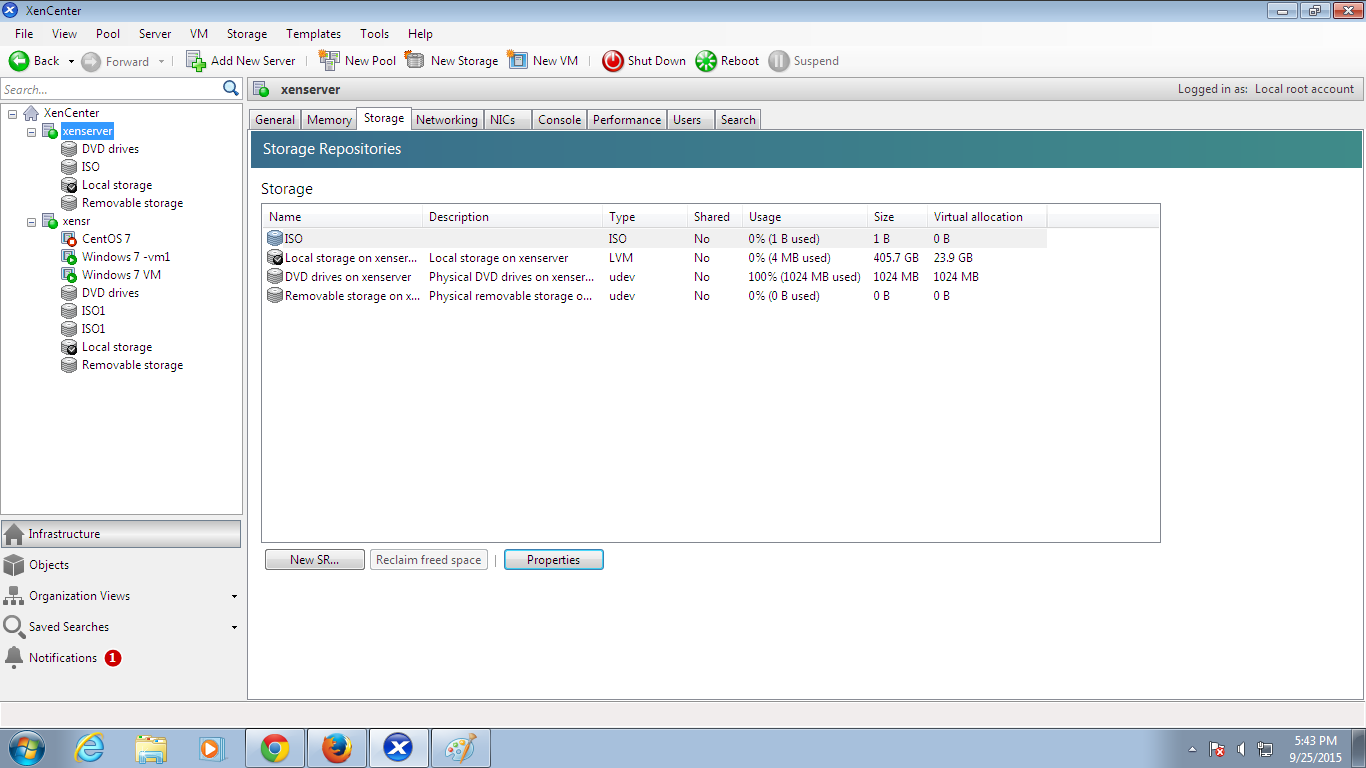
**Creating VM**



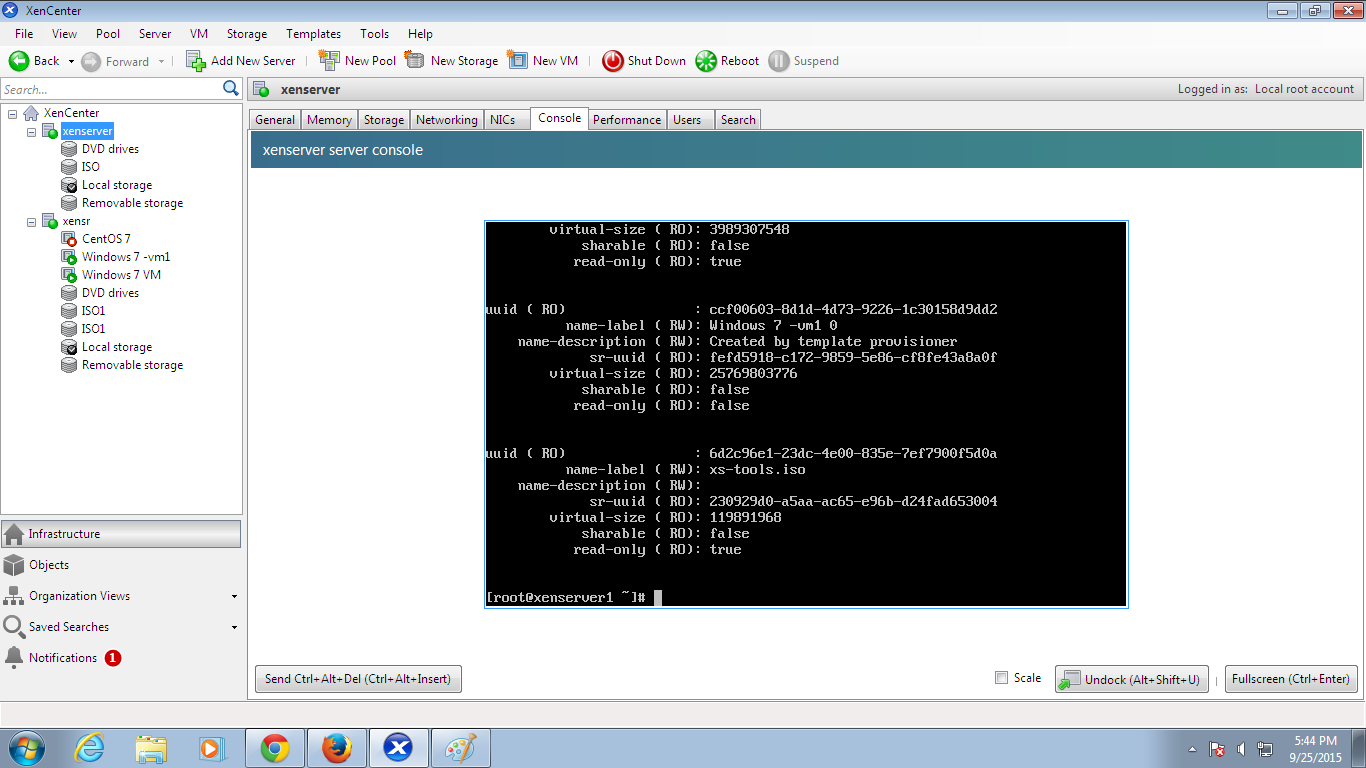
**Go to Xenserver properties for any modifications:**

****

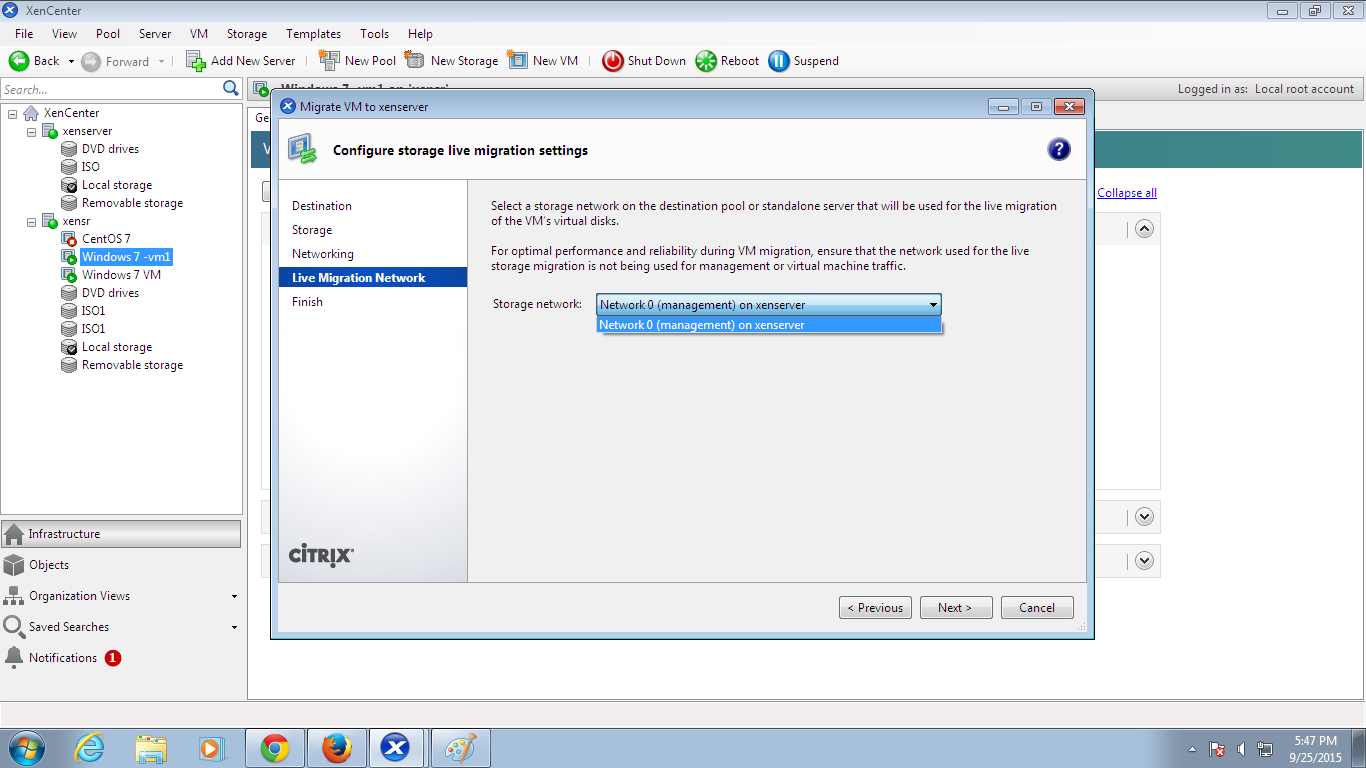
**Go to storage repositories for storage information:**



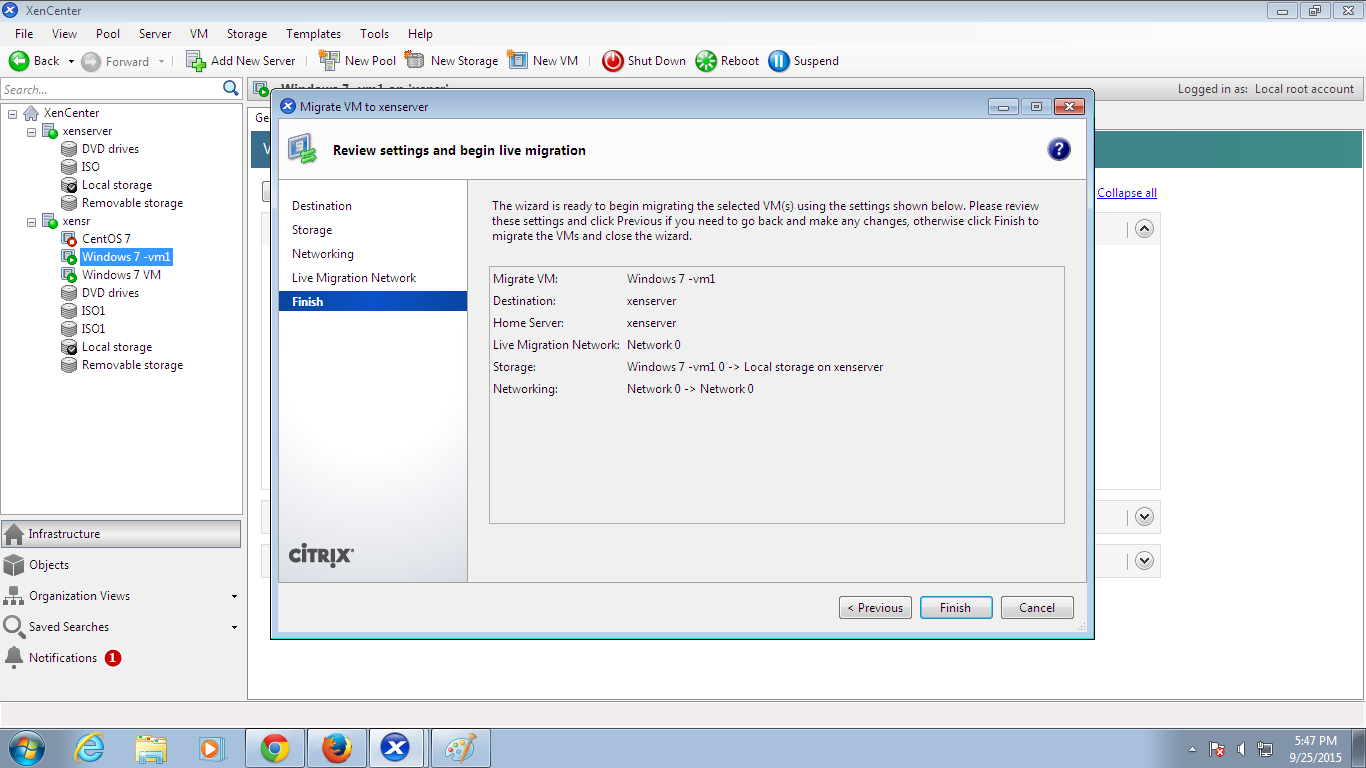
**Xenserver server console:**



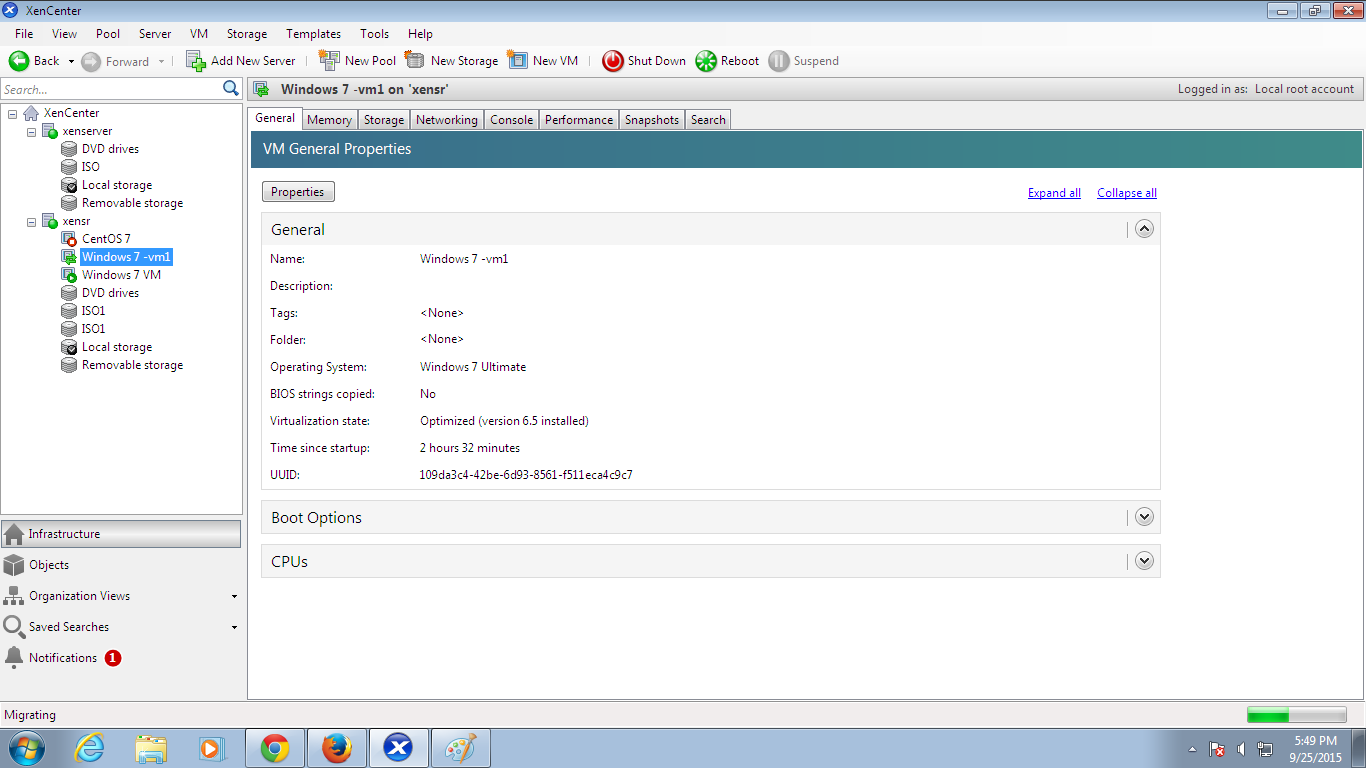
**Migrating VM to server:**



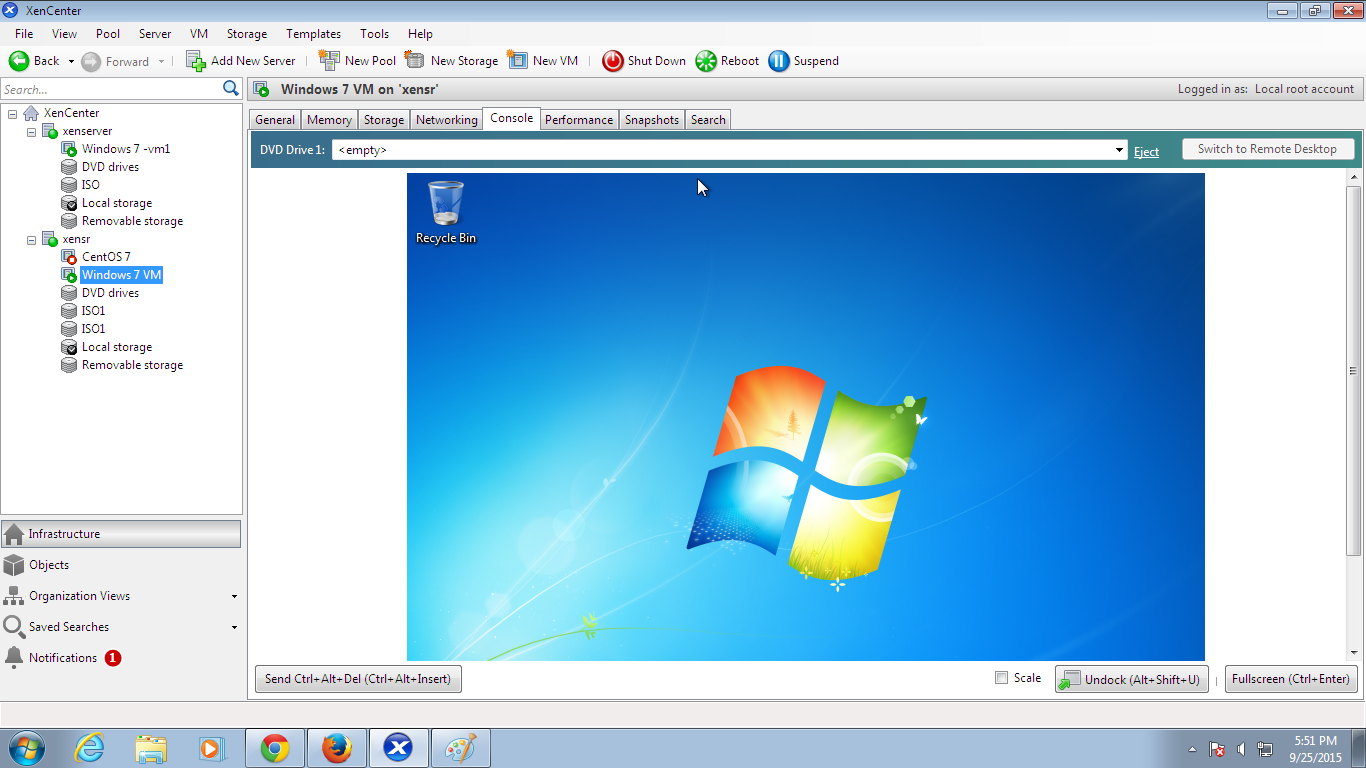
**Click finish when migration gets over:**

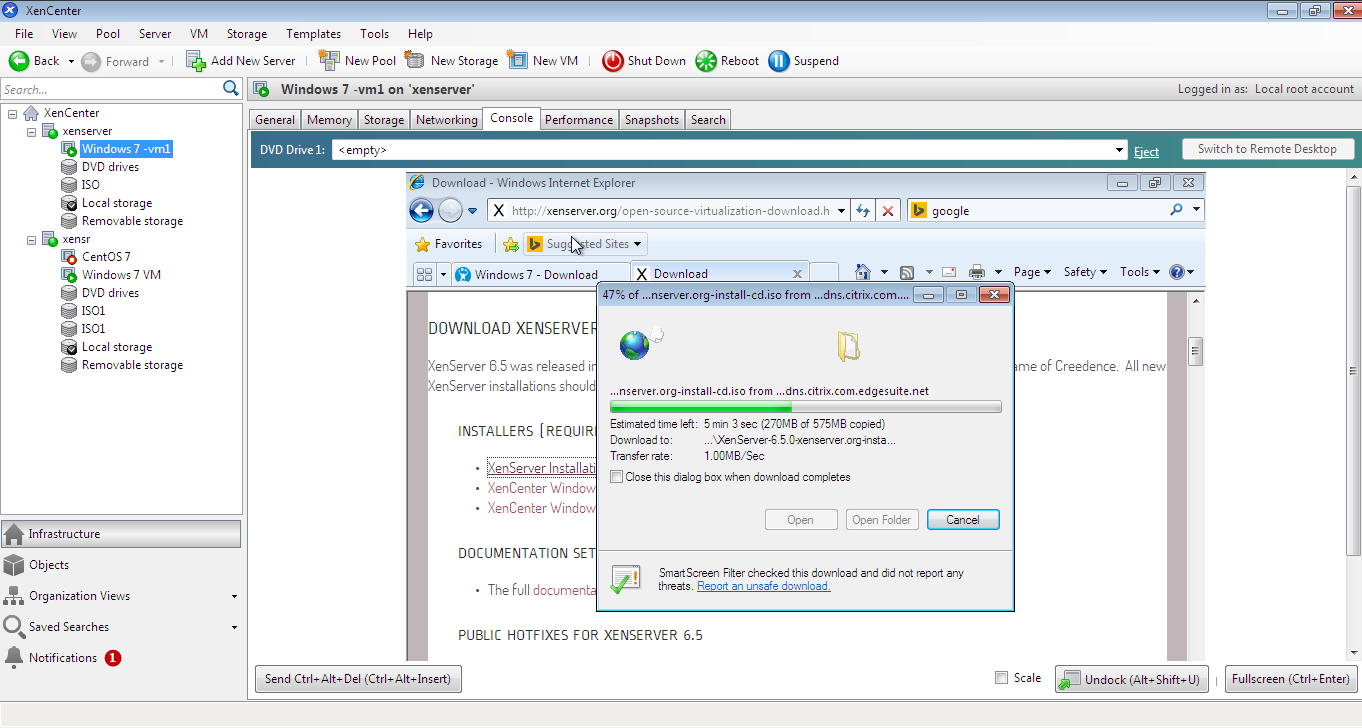


**Virtual Machine Migrated:**

****

**Starting Virtual machine:**



**Fig 6.8 Downloading ISO image**

**7.CONCLUSION**

**7.1 CONCLUSION**

The Dynamic load balancing is a technique to use the cloud computing in efficient manner. The algorithm used in this approach can automatically monitor the load balancing with the use of load balancer The data replication, job migration and the static load balancing is avoided in our method. The CPU and Memory can be utilized properly and the reliable VM in the cloud pool can be identified. Nowadays allocation of resource in Infrastructure-as-a-Service environment has more attention because of its ability to satisfy the user demand. Since cloud computing offers truthful mechanism that provide virtual resource as per the users requirement. The proposed algorithm and gossip based technique solves the migration problem in a short time with the multiple instance. The major challenge of cloud computing is that overload can be avoided and green computing is performed.

**7.2 FUTURE WORK**

In future, we will develop a Dynamic resource allocationalgorithm for underutilized VM in datacenter. This algorithmshould have criteria for selecting a VM, selecting a nodewhich has sufficient space for underutilized virtual machines.This ensures more Load Balancing in private cloud.Using the load balancer cloudblock also makes it simple to expand or smoothly transition your physical servers to the cloud service.