Availability Manager

Project 1: CMPE 283 Virtualization Technologies



**Submitted By:**

Pawan Shrivastava (SJSU ID: 009308169)

pawan.shrivastava@sjsu.edu

CMPE 283, Virtualization Technologies

Fall 2013

**Submitted To:**

Professor Shimon Shim

Computer Engineering Department

San Jose State University

**Introduction**

The goal of Disaster Recovery application in a virtualized environment is to manage the faults and failures of Virtual Machines and hosts which are connected to a shared Datacentre and also to provide functionality of continuous monitoring of those VMs and host attached to this Datacentre so if any of the host or VMs get stopped, than they can be replaced by another set of VMs from the resource pool or these VMs can be transferred to a different working host.

This kind of application is very helpful in the real world cloud computing network where there are large number of low commodity hardware involved to implement these Datacentres and virtual host. So there is high probability of any of these hardware’s might get down in such a big network so in these cases this application would be very handy in figuring out which physical device has went down so we can move all those working virtual machine on top of those hardware to move on the next available hard ware. So this application can monitor a large set of VMs in a virtual network and also take some post-failure actions.

**Background**

In the modern day of Information technology where there are lots of physical hardware’s are being compressed and segregated to one place to host all the machines in the form of servers and these servers becomes the point of hosting these virtual machines and virtual hosts. In this kind of environment there is a need of application which can monitor these vHost and VMs .vCenter which is a product of VMware perform same kind of task by providing an interface to monitor and manages those VMs and vHost .But there is a need of application which can provide automation for all the monitoring activities, also in case of any failures of any virtual machines it can replace with a new one or can move to some other working vHost.

**Functional Requirement**

1. VM is considered to be “live” when it replies to ping and “dead” when it stops responding to the ping.
2. If the VM is appearing to be “dead” than we need to start another VM from the resource pool.
3. While selecting a host, availability manager must make a compulsory check whether host is working fine or not.
4. It should always create an alarm on the virtual machine which it is monitoring and in case of power off by user only it must trigger the alarm.
5. In cases if the VM is not responding to ping it should seek for another working host to clone and migrate this VM on that host.
6. It should create snapshot after every 20 minutes and must remove the earlier snapshot as soon as a new snapshot is created.
7. It must be also able to detect the difference between VM shutdown due to a failure or manual power OFF and respond accordingly.
8. In case of VM failure ,this application must provide facility to clone the original VM > migrate it to new host >delete the old VM > Powered on the new VM only when the old VM is deleted from the previous host.

**Non Function Requirement**

1. Application should hold the correct copy of the snapshot in the memory and it must be able to clone the VM in case of failure from the most recent snapshot.
2. Ping function should be continuous and work in synchronization with the VMs and application as the ping response and reply time should be regular.
3. The system must power ON the migrated VM only when the previous VM is switched OFF.

**Design**

**VMware vSphere**  **Client** **Client**

**DATACENTER**

VHOST 1

Vmware ESXI

VHOST 2

Vmware ESXI

Virtual Machine 1 

Virtual Machine 2

Virtual Machine 3

Architecture of Lab Set up

1. In our infrastructure we have got one centralized Datacentre which is hosting 2 Vhosts Vhost 1 and Vhost 2 .All the Vhost have VMware Esxi installed on it.
2. Virtual machine are connected to the first Vhost in the initial set up and can be migrated upon the case of vHost failure.
3. Mostly Ubuntu and Windows are installed on virtual machines.
4. Vsphere client is monitoring whole of the infrastructure.
5. Vmware tools are running on all the virtual machines so they help in getting the network information of the virtual machines.

**Approach**

1. We would manage the virtual machine attached to first host and in case of failure we will transfer the VM on other available working vHost.
2. Our script will first ping the first vHost in this 2 conditions can arrive :

* If the first vHost itself is not pinging than we need to wait till the first vHost is alive than only we can ping the VMs attached to it
* If the first vHost is pinging and working fine , than I need to start ping the VMs attached to the working host.

1. In case of VM pinging fine, my script should first create an Alarm on my virtual machine so the application can understood between a manual power off by the user and a failure of VM so that it can respond accordingly. In case of manual power off the Alarm should Turn on the VM again automatically.
2. It should continue monitoring the aliveness of the virtual machine and must create the snapshot of the machine at every 20 minutes and must delete the old snapshot as the new snapshot is created.
3. We would be assuming a host failure by disconnecting the network adapter of my virtual machine and now my code will start looking for another available host on the datacentre. When it finds the another vHost it will start pinging to it and if it respond our code will start the process of CLONE-MIGRATE-DELETE-POWERON.
4. Our code will start the process of cloning of the original VM on the same Host from the most recent Snapshot. After it has completed cloning it will start migrating the clone to the other available vHost found in step 5.
5. As soon as the migration is completed it will start the process of “Deleting” the old VM
6. As the process of deleting of old VM is completed it will power on the cloned VM on the New host.
7. It will now start monitoring the VM on the new host and will continue to function same as in step 3.
8. New snapshots will be also created after every 20 minutes.

**Implementation of Different Scenarios**

1. The Host Add/Remove Mechanisms

public boolean migrateToAnotherHost(String vmname1 , String newHostName1) Method is used to perform the migration of VM from one host to another .In the case of failure of the original host my code will start looking for the another available free host attached to the datacenter ,once it get the free host it will start pinging to second vHost by using public Boolean pingSecondHost() method and once the second host starts responding it will perform the migration of VM on this host.

1. The approach used to configure the failure detection for each VM .

I have used synchronized public Boolean pingVM() method to continuously monitor my VM by pinging the VM at regular intervals.As soon as the VM starts responding back to ping .Than a failure is detected and we can start with the process of CLONE-MIGRATE-POWERON

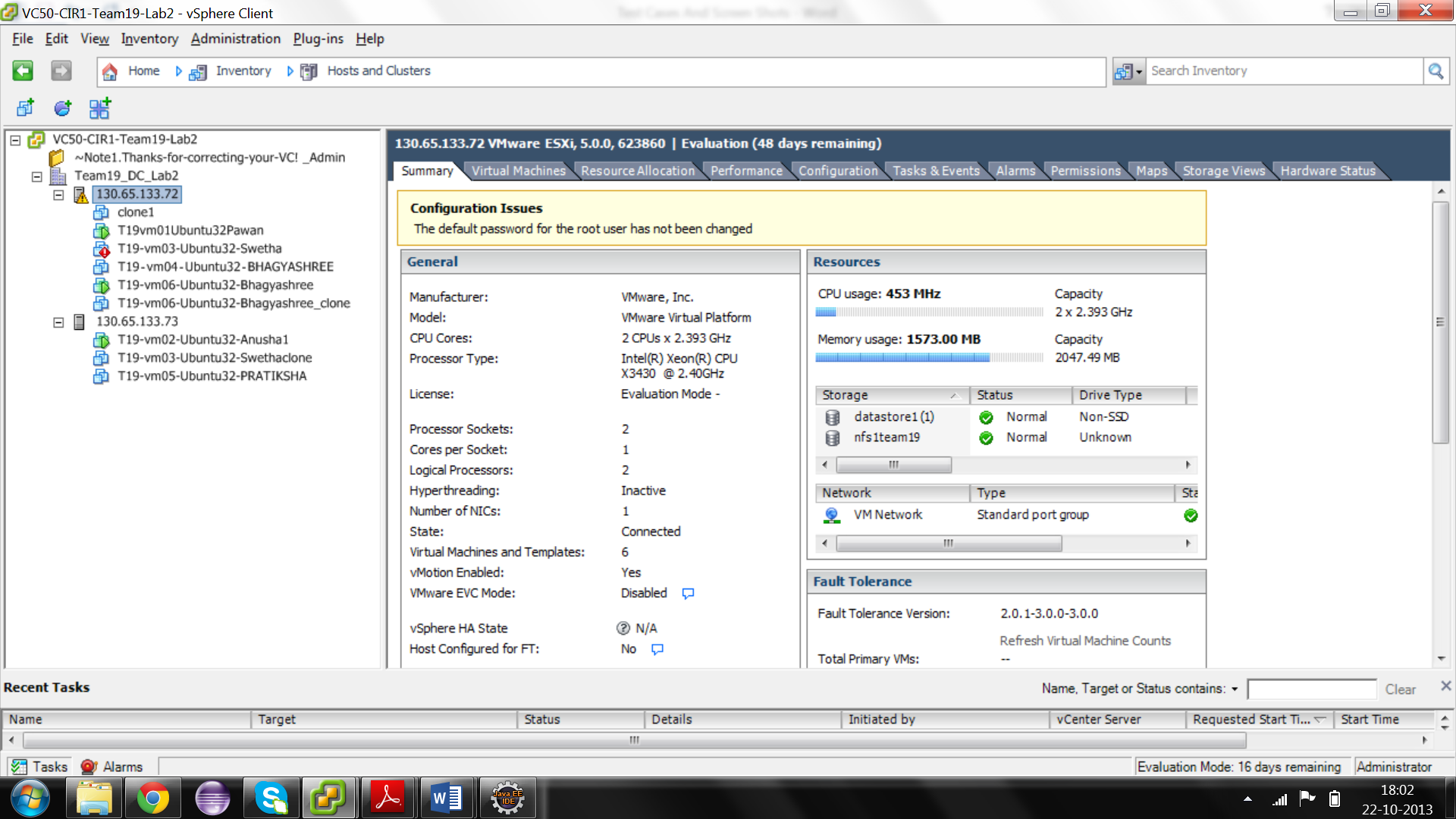
**3**. How host failures were detected

Host failures was detected using the public Boolean pingHost() method in this method I would continuously monitor by pinging on the host and whenever the host doesn’t reply to a ping that means a host failure has occurred and my code will start looking for the another available host

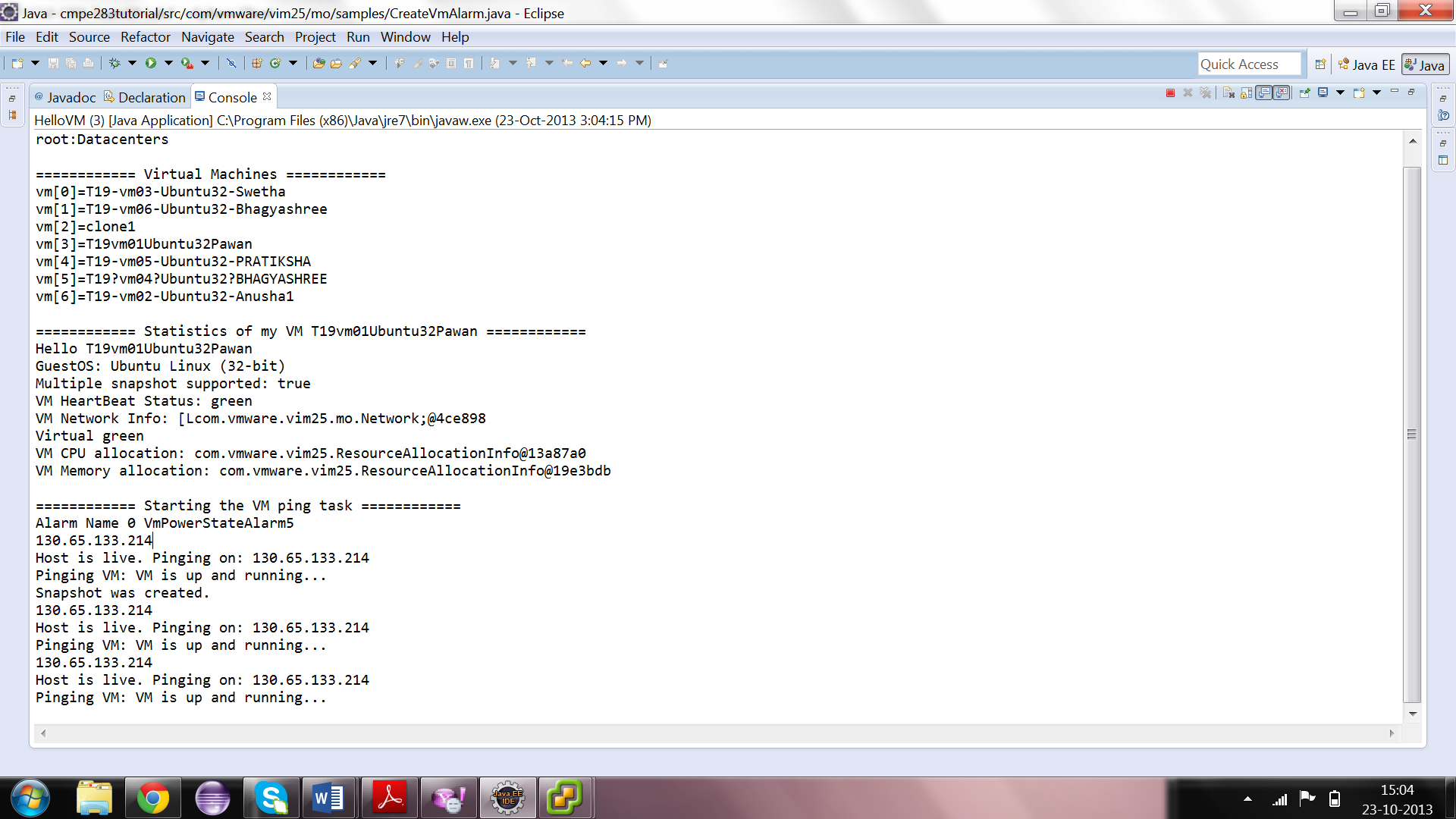
**4**. The mechanism used to convert between the image formats used by the hypervisors : Our code takes snapshot after every 20 mins and deletes the previous snapshot once the new snapshot is created in memory it will delete the old snapshot . In case of VM failure it would pick up the most recent snapshot and start cloning from that snapshot

**Test Cases And Screen Shots:**

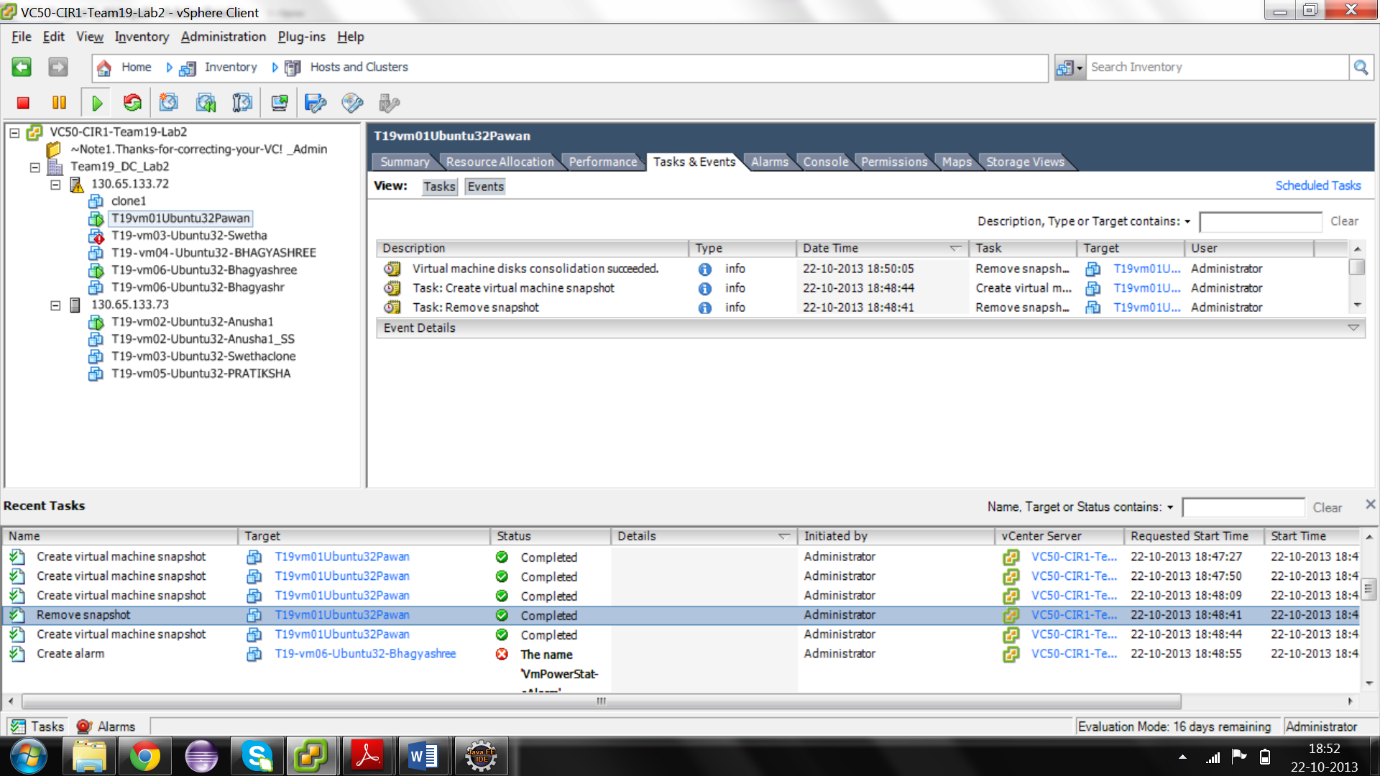
There is 1 datacenter “Team19\_DC\_Lab2” and 2 hosts “130.65.133.72” & “130.65.133.73” attached to the datacentre. My VM “T19vm01Ubuntu32Pawan”is attached to the first host.

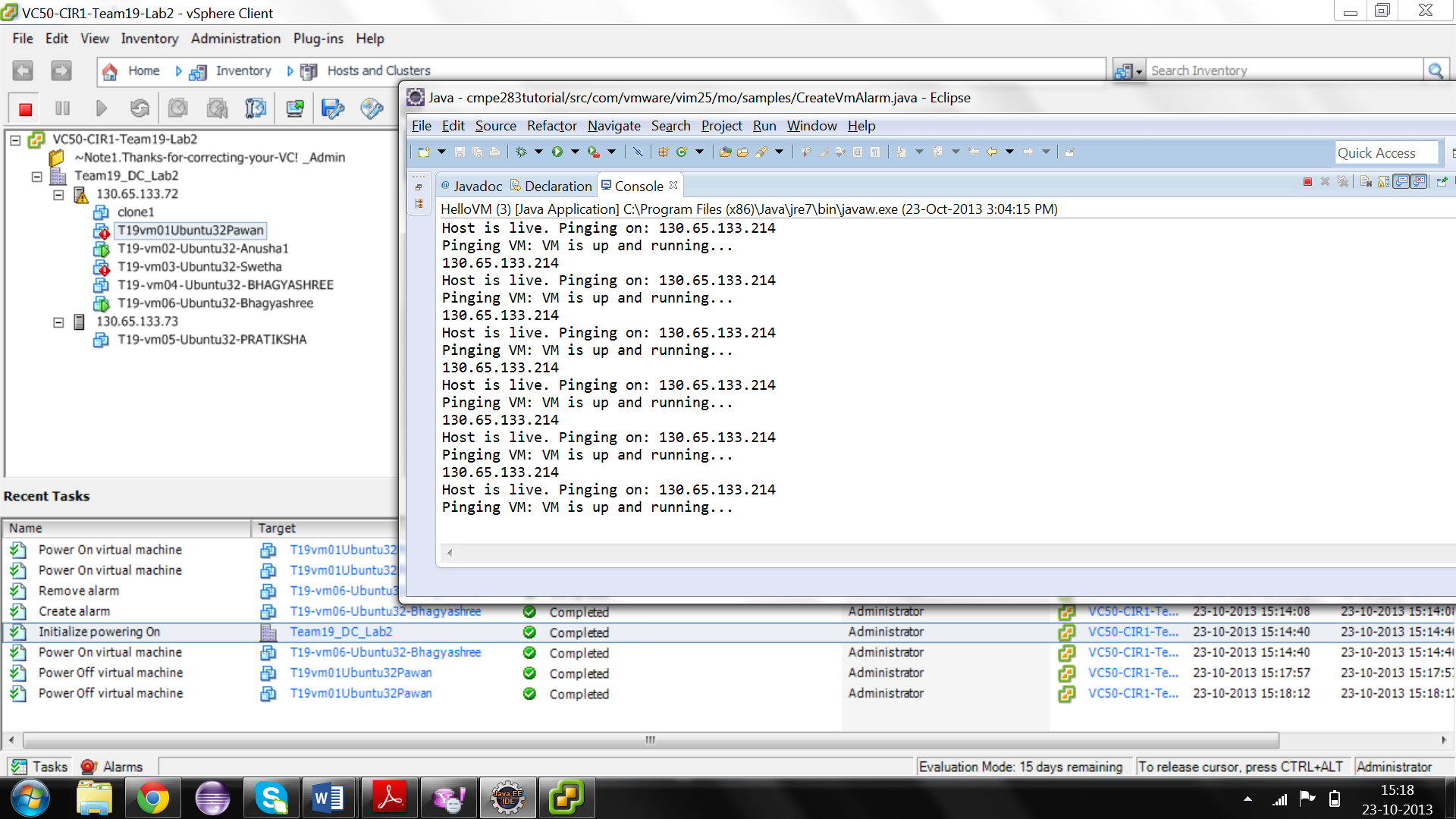


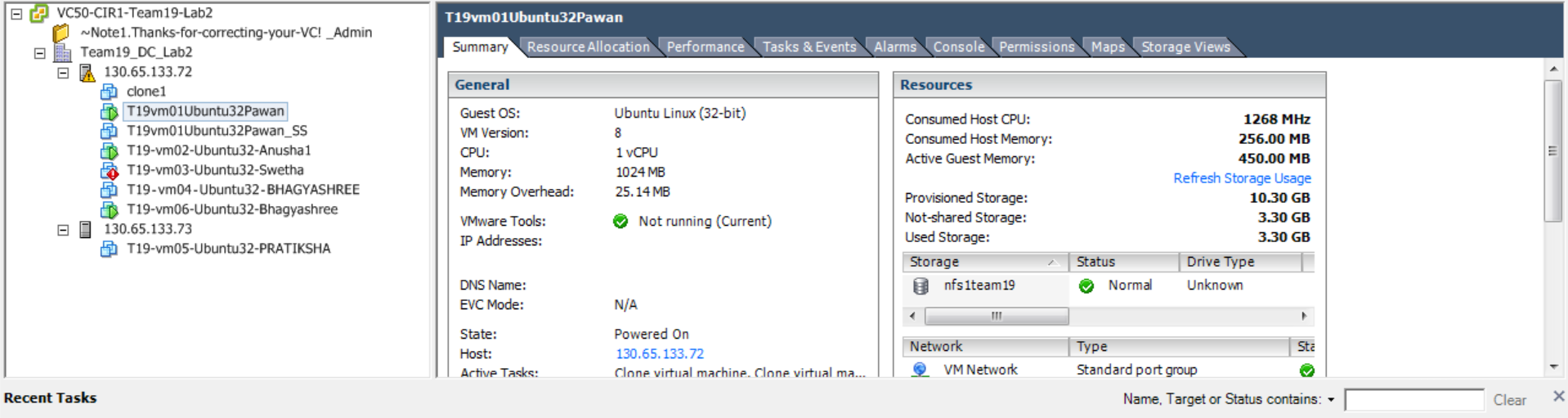
**Main Script create Alarm and if alarm already exists it will remove and create that again. It will display the datacentre, all the VMs connected to that Datacentre and then it will select my VM and list all the properties related to and start pinging my VM .It will create snapshot after every 20 mins.**



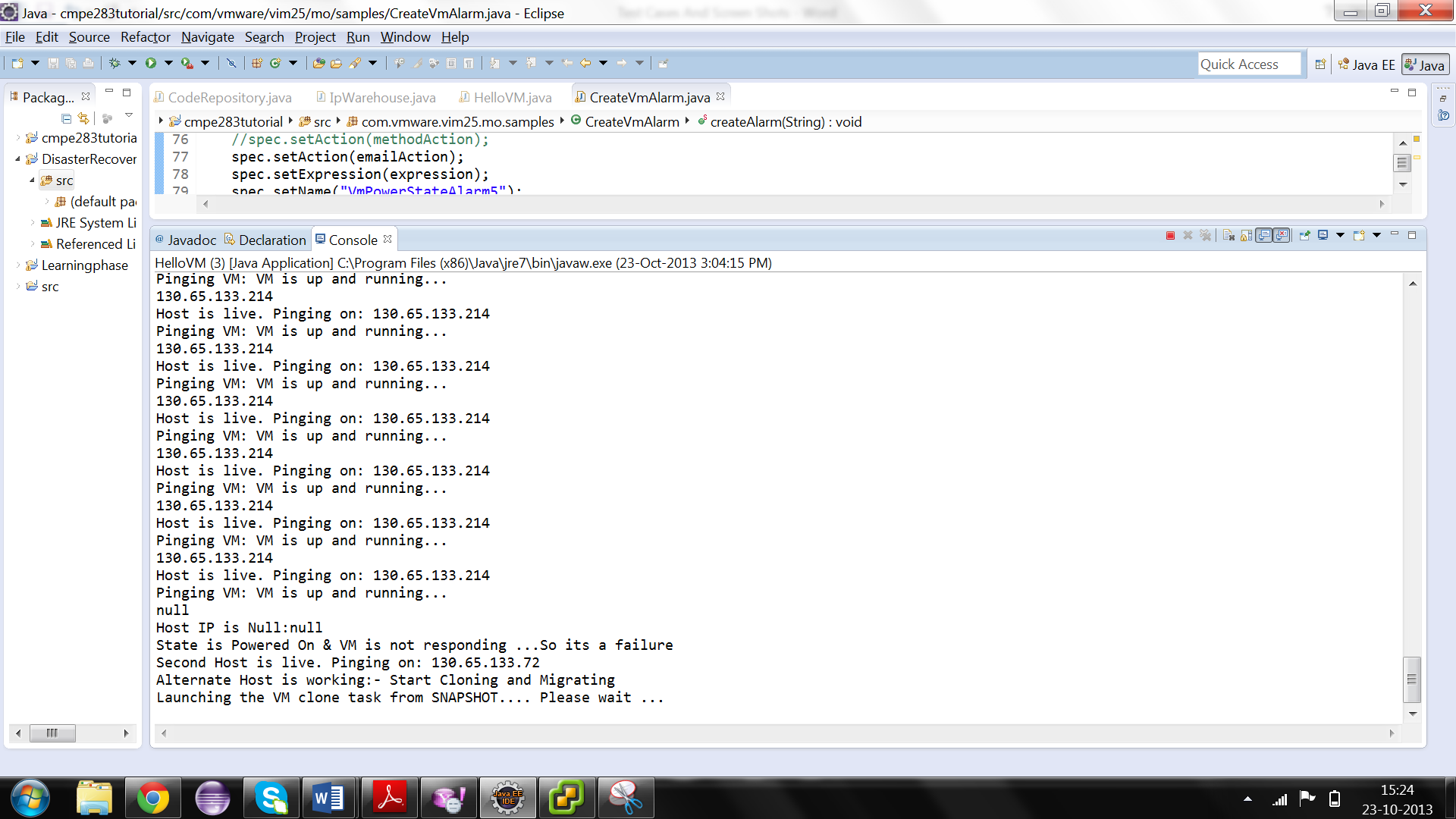
**Second thread is working and After creating first Snapshot it is removing previous One**

  
**If the VM is powered off by user , it will trigger the alarm and Alarm will switch on the VM again.**

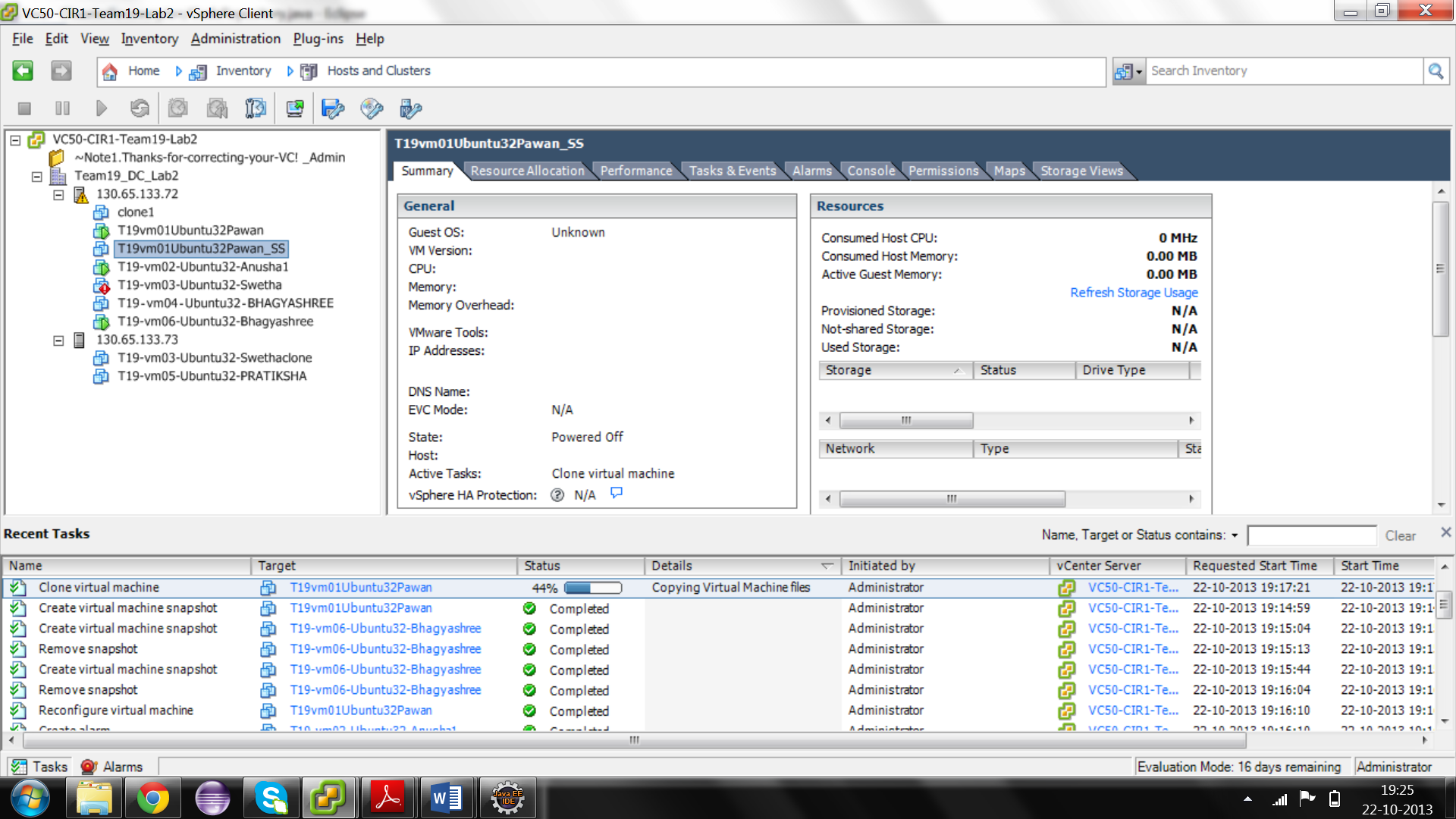




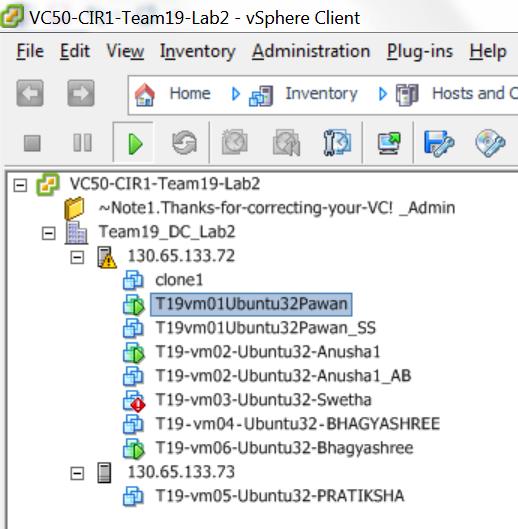
**When VM starts stops responding to ping, it will assume that VM is failed and it will start searching for other available live node to migrate the VM. It will first start cloning of VM on the same host from the most current snapshot**



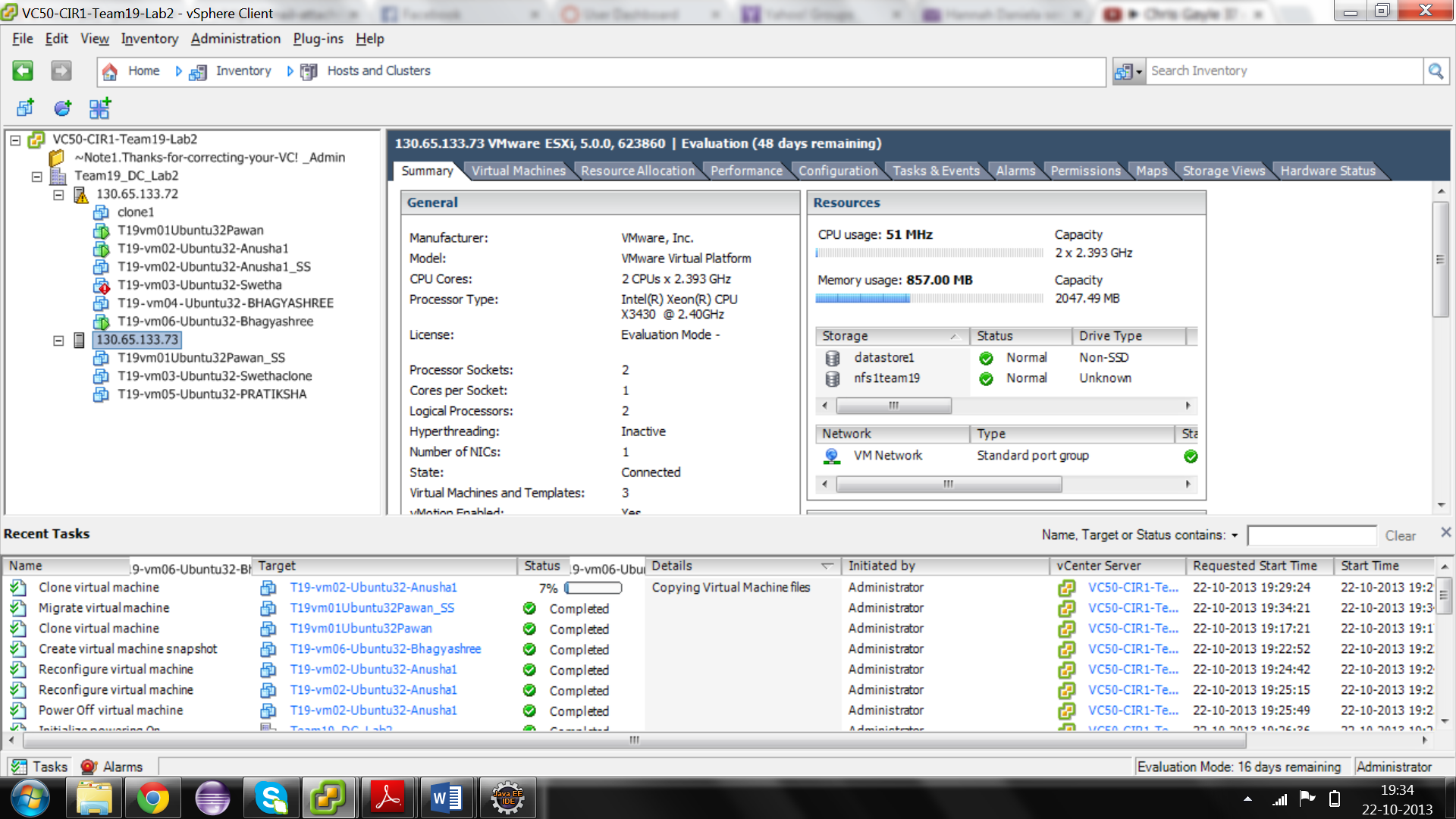
**Below we can see cloning in Progress**



**Clone VM is created on the same host and is now ready to migrate on the other live host with name32T19vm01Ubuntu32Pawan\_SS**

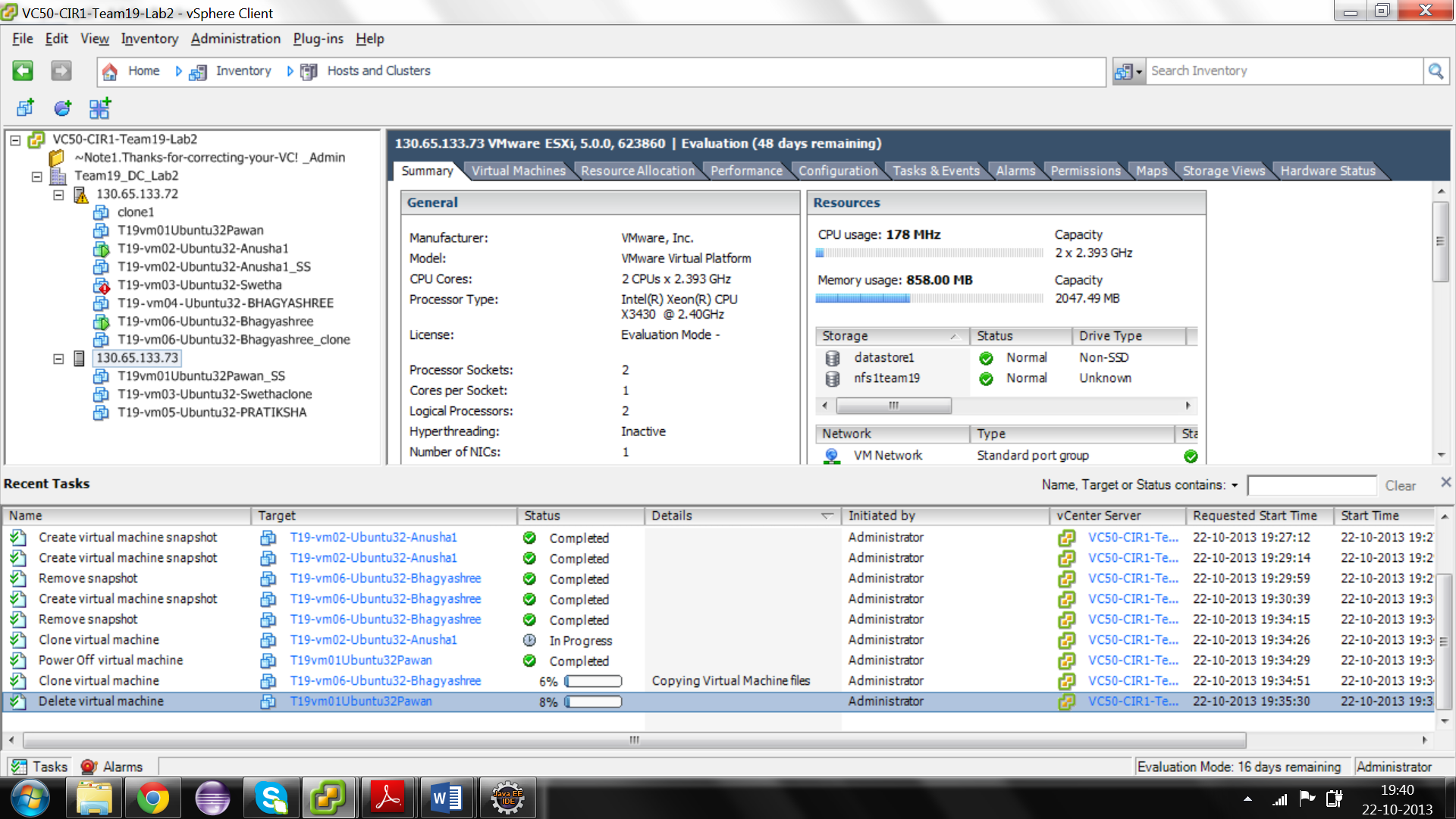


**Clone VM is transferred to another host 130.65.133.73**





**Deleting of old VM is in progress**



Old VM is deleted from the previous host and is started on the new VM.