Virtualization Problem Statement - Solution

1. Analysis of Problem Statement:

The company consists of various data centers across different locations. Each data center have several virtual machines running on it, to make a complete use of resources available with that datacenter. During there maintenance activity, we need to power off these VM on each datacenter. Hence our main task is to develop a remotely accessible web application to gracefully power off all those virtual machines and services of the host on each data center and then power off host as well. Similarly, during turning on virtual machines we need to regain the original status as it was prior to the shut down.

2. Details of technology used :

1) **VMWare Workstation** (for demo purpose).
2) **ESXi ISO**
   Its major purpose is to create VM's. It assigns the unique mac address and IP address to this VM's.
3) **The python** libraries :
   - pyvmomi
   - vconnector
   The python script is used for automation using various functions provided by this library.
4) **Firebase Database** keeps the track of MAC address of DC's, inventory and status of the VM's on each DC and information about Company.
5) **PHP, JS, HTML:** Technologies used for the webapp.

3. Required Softwares:

1) VMware technology :
   a) VMWare Workstation(Prototyping purpose)
   b) VMWare CLI
   c) ESXi ISO
2) Firebase Database
3) Python
4) PHP, HTML, JS, etc.

4. Achieved cost saving :

Due to automation, human intervention is reduced and that human resource can be used for further betterment of the company. It will also reduce the amount of money spent on human resources, thus achieving cost saving.

5. Architecture :

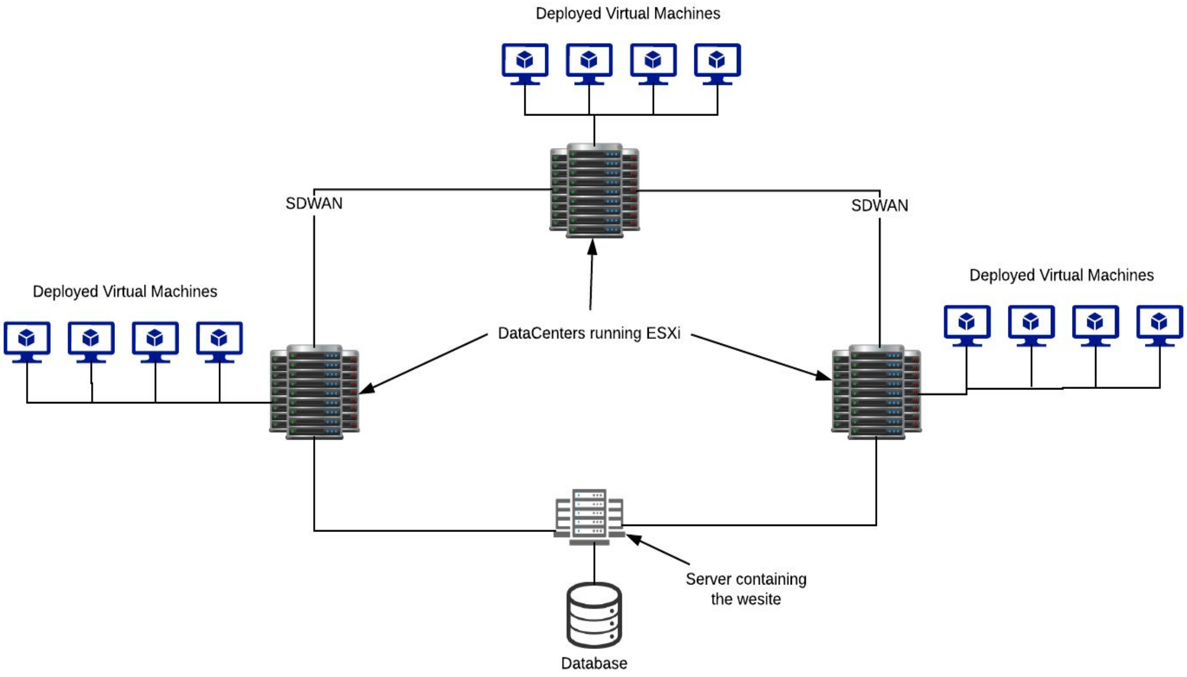

Deployed Virtual Machines
SDWAN
SDWAN
Deployed Virtual Machines
Deployed Virtual Machines
DataCenters running ESXi
Server containing the wesite
Database

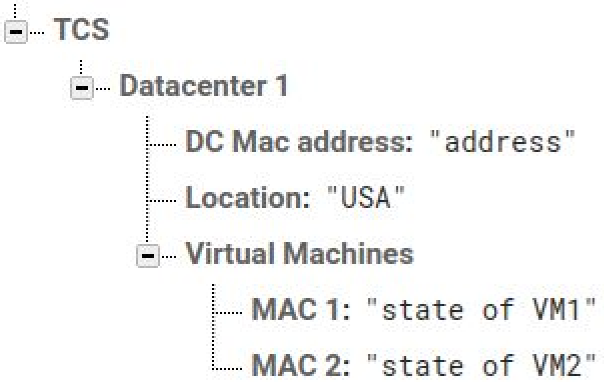
Fig. 5.1 Architecture of the system

DC’s are connected by SDWAN which can be accessed remotely. ESXi ISO is deployed on each DC’s, that creates VM’s. According to the system requirement, a web application(a website) will be added to the network by deploying it on a server in the network. When the core team member wishes to power off the system, he will just select from the web application; the centers to be powered off. In the backend, all the selected DC’s will be checked for VM’s. Once it is done, inventory of machines along with their states at the current moment will be stored in the database. A glimpse of the network is shown in the fig. 5.1.

6. Solution brief description :

To achieve this automation we planned to deploy a web application in the same network.The web application will show the active DC’s in the given network. So when the core team member wishes to power off the DC’s, the following steps will take place:

1. Login to the web application to access the DC’s data.
2. MAC addresses of all the DC’s will be fetched and displayed.
   (These MAC addresses were already stored in the database as when a company registers a new DC, it will be entered, ONCE)
3. Selecting the DC to power off or selecting the select all option to power off all of them.
4. Once a DC is selected, using the mac address of selected DC so the current states of all VM’s will be fetched using the IPV6 fetching connection to the ESXi.
5. The states of all the VM’s will also be saved in the database as shown in the fig.6.1
6. Using python command at first all VM’s will shut down and later the corresponding hypervisor (Host) will be shut down.
8. While turning on the system, the states of all the VM’s according to the MAC addresses will be fetched from the database and their states will be restored as it was.
9. Logout of the admin.



- TCS
  - Datacenter 1
    - DC Mac address: "address"
    - Location: "USA"
    - Virtual Machines
      - MAC 1: "state of VM1"
      - MAC 2: "state of VM2"

Fig 6.1 Database Structure

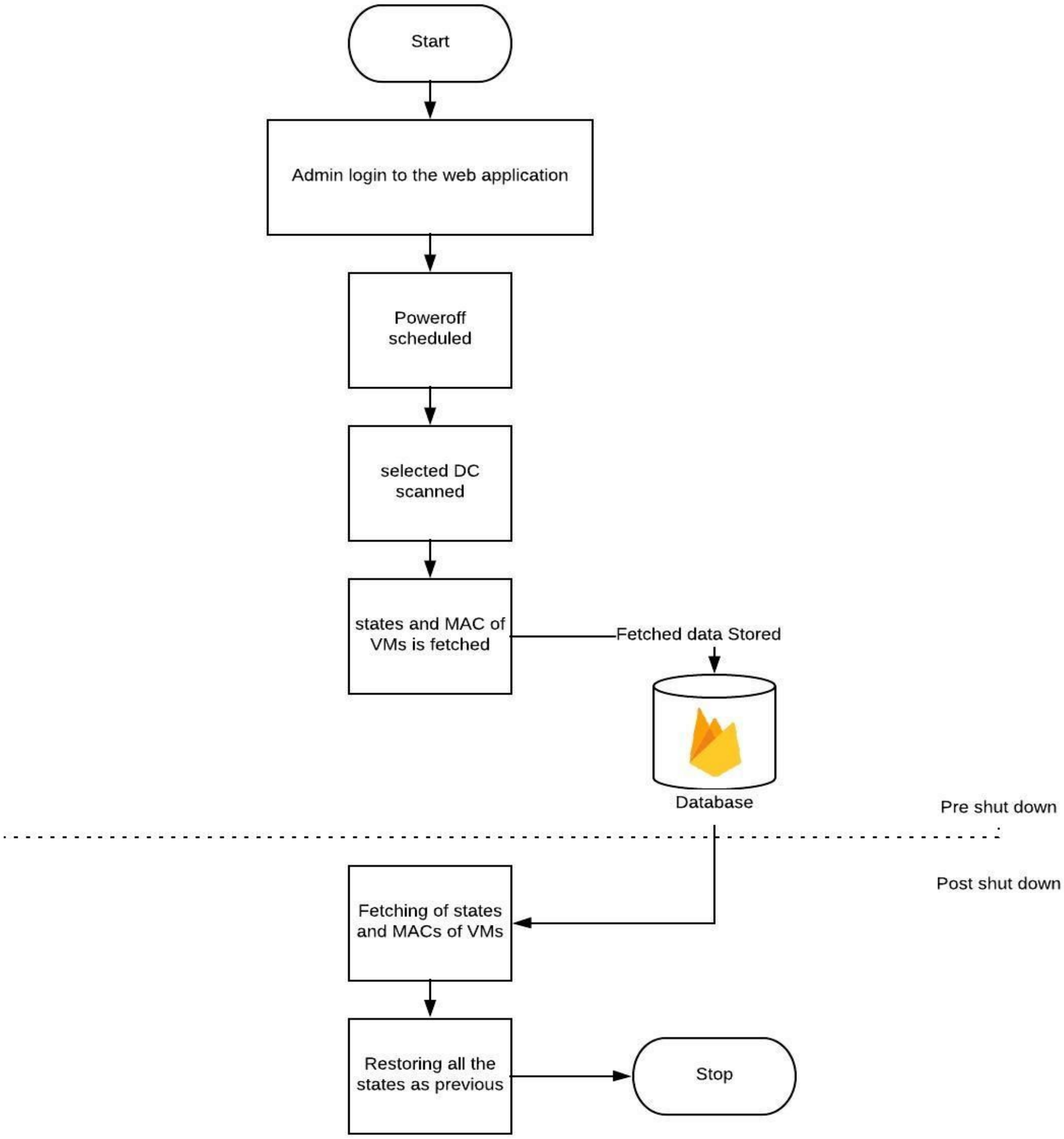

Start
Admin login to the web application
Poweroff scheduled
selected DC scanned
states and MAC of VMs is fetched
Fetched data Stored
Database
Pre shut down
Post shut down
Fetching of states and MACs of VMs
Restoring all the states as previous
Stop

Fig 6.2 Flow of system

7. Future Scope of Automation :

The system designed above demands the MAC addresses of all the DC's to be registered once the DC is added to the network. This process can be automated by making available all the MAC addresses of all the connected DC's by using a DHCP server connected in the SD-WAN network.

8. Conclusion :

A web-based application is developed that can be remotely accessed by the core team of the company, which will provide them the facility of automatically powering off all the VM’s running on different DC’s, display the current status of the VM on DC and retain the original status of VM’s when turned on. Thus automation is achieved in virtualization.