RDS Script Automation And Environment Enhancement

Submitted By

Romita Rath

13MCEN12

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

INSTITUTE OF TECHNOLOGY

NIRMA UNIVERSITY

AHMEDABAD-382481

May 2015

RDS Script Automation And Environment Enhancement

Major Project

Submitted in partial fulfillment of the requirements

for the degree of

Master of Technology in Computer Science and Engineering (Networking Technologies)

Submitted By

Romita Rath

(13MCEN12)

Guided By

Guide Name

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

INSTITUTE OF TECHNOLOGY

NIRMA UNIVERSITY

AHMEDABAD-382481

May 2015

Certificate

This is to certify that the major project entitled **"RDS Script Automation And Environment Enhancement"** submitted by **Romita Rath (Roll No: 13MCEN12)**, towards the partial fulfillment of the requirements for the award of degree of Master of Technology in Networking Technologies (Computer Science and Engineering) of Institute of Technology, Nirma University, Ahmedabad, is the record of work carried out by him under my supervision and guidance. In my opinion, the submitted work has reached a level required for being accepted for examination. The results embodied in this project, to the best of my knowledge, haven't been submitted to any other university or institution for award of any degree or diploma.

Prof. Jigna Patel
Guide & Associate Professor,
CSE Department,
Institute of Technology,
Nirma University, Ahmedabad.

Prof. Gaurang Rawal
Associate Professor,
Coordinator M.Tech - CSE -NT
Institute of Technology,
Nirma University, Ahmedabad

Dr. Sanjay Garg
Professor and Head,
CSE Department,
Institute of Technology,
Nirma University, Ahmedabad.

Dr K Kotecha
Director,
Institute of Technology,
Nirma University, Ahmedabad

Statement of Originality

I, **Romita Rath**, Roll. No. **13MCEN12**, give undertaking that the Major Project entitled "**RDS Script Automation And Environment Enhancement**" submitted by me, towards the partial fulfillment of the requirements for the degree of Master of Technology in **Computer Science & Engineering (Networking Technology)** of Institute of Technology, Nirma University, Ahmedabad, contains no material that has been awarded for any degree or diploma in any university or school in any territory to the best of my knowledge. It is the original work carried out by me and I give assurance that no attempt of plagiarism has been made. It contains no material that is previously published or written, except where reference has been made. I understand that in the event of any similarity found subsequently with any published work or any dissertation work elsewhere; it will result in severe disciplinary action.

_____

Signature of Student

Date:

Place:

Endorsed by

Prof. Jigna Patel

(Signature of Guide)

Acknowledgements

It gives me immense pleasure in expressing thanks and profound gratitude to **Prof. Jigna Patel**, Associate Professor, Computer Science Department, Institute of Technology, Nirma University, Ahmedabad for her valuable guidance and continual encouragement throughout this work. The appreciation and continual support she has imparted has been a great motivation to me in reaching a higher goal. Her guidance has triggered and nourished my intellectual maturity that I will benefit from, for a long time to come.

It gives me an immense pleasure to thank **Dr. Sanjay Garg**, Hon'ble Head of Computer Science and Engineering Department, Institute of Technology, Nirma University, Ahmedabad for his kind support and providing basic infrastructure and healthy research environment.

A special thank you is expressed wholeheartedly to **Dr K Kotecha**, Hon'ble Director, Institute of Technology, Nirma University, Ahmedabad for the unmentionable motivation he has extended throughout course of this work.

I would also thank the Institution, all faculty members of Computer Engineering Department, Nirma University, Ahmedabad for their special attention and suggestions towards the project work.

I would also like to express my special gratitude and thanks to Jatin Somarajan (Team Lead, Oracle), Darren Charloff(Senior Manager, Oracle) and Paul Kehler(Senior Manager, Oracle) for providing necessary information regarding the project and for their guidance and constant supervision. Also a special thank You to my colleagues in RSS team at Oracle Retail India Pvt. Ltd. and people who have willingly helped me out with their abilities.

- Student Name

13MCEN12

Abstract

Retail demonstraion services is responsible for providing technical infrastructure for the pre sales demo. The demos are intended for prospective clients informing them about working of Oracle retail products. The demos are also needed for clients when new release of software is out in the market. The environments on which demo are conducted need to be working properly without any errors. Since the demos are responsible for millions of dollars of contracts. Hence optimized environment is needed with regular monitoring. Therefore this project is for enhancement of current environment.

RDS environments needs monitoring optimization and automation. Monitoring optimization include optimized scripts for VM monitoring and monitoring tool to make sure the refresh checklist is followed. The script automation is needed to reduce the mundane repition of refresh tasks. The entire project is aimed to enhance the working environment of RDS team and optimize the performance.

Abbreviations

RDS	Retail Demonstration Service.
VM	Virtual Machine.
WLST	WebLogic Scripting Tool.
Db	DataBase.
Rib	Retail Integration Bus.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 General

Retail demonstraion services is responsible for providing technical infrastructure for the pre sales demo. The demos are intended for prospective clients informing them about working of Oracle retail products. The demos are also needed for clients when new release of software is out in the market. The environments on which demo are conducted need to be working properly without any errors. Since the demos are responsible for millions of dollars of contracts. Hence optimized environment is needed with regular monitoring. Therefore this project is for enhancement of current environment.

RDS environments needs monitoring optimization and automation. Monitoring optimization include optimized scripts for VM monitoring and monitoring tool to make sure the refresh checklist is followed. The script automation is needed to reduce the mundane repition of refresh tasks. The entire project is aimed to enhance the working environment of RDS team and optimize the performance.

1.2 Scope of Work

The project is intended to automate and enhance the working environment of RDS (Retail Demo Service). RDS works for providing demo environment for pre sales team. The environments have to be monitored continouslly to avoid any failure during actual demo. Such failure can lead to loss of millions of dollars. Hence the environments have to be refreshed regularly.

Basic scope of this project focusses on automating this refresh task. This will optimize the procedure and more time can be devoted to maintainance of environment. The automation subtasks include, automation of rdsAll script, automation of OVM tool, and automation of weblogic domain creation. The monitoring of VM is needed to make sure the working of environment is stable. The monitoring scripts deal with diskspace, vm status and system monitoring. The monitoring of refresh procedure is also neede to conduct the refresh steps properly.

Chapter 2

Literature Survey

2.1 General

Oracle Retail provide tightly integrated suite of retail applications [1]. Retail business has all business activities that include the sale of merchandise and services to consumers for personal, family, or household use. Retail Products span across entire retail process:

- Planning
- Buying
- Merchandinsing and Inventory Management
- Selling and customer service
- Support

2.2 Literature Review

2.2.1 Trainings Attended

Java Basics And JavaScript

The training included brushing up of basic java concepts. Since oracle majorly deal with java language, the training was intended to familiarize with java concepts.
JavaScript is a scripting language. It is most often used as part of web browsers, executions of which allow client-side scripts to interact with the user, control the browser, communicate unparallely, and modify the document content that is displayed [2]. The

Javascript training included introduction to javascript and basic scripting. The concepts were introduced and syntax and semantics were brushed up.
Javascript would be used in the project for preparing a monitoring tool, which basically ensure the steps of refresh are followed methodically.

Unix Shell Scripting

A shell script is a computer program outlined to be run by a command line interpreter, the Unix shell[3]. Shell script is much simpler and effective programming tool which can directly manipulate the underlying system. Shell script is used in workng domain to monitor VM's and Unix commands are used to interact with remote host. The training included the basic unix commands and understanding to basic shell scripting.
The project has unix script as integral part, since most interaction is on remote unix machine. Shell script is used in automation script and also in monitorning scripts for RDS.

Python Scripting

Python is an uncomplicated and influential programming language. It has methodical high-level data structures and a straightforward approach to object-oriented programming[4]. Python's simple syntax and effectual typing, along with its interpreted nature, makes it an flawless language for scripting and speedy application development in innumerable areas on majority of platforms[4].
Python training was carried out for a week and all the programming concepts were taught. Python scripts are mor optimized in terms of perforamance and hence preffered in maintainace scripting. Since it is extra work and has to be efficient to utilize resources.
Python script is used in the project for creating the monitporing scripts, which monitor the status of VM system.

Weblogic

Oracle WebLogic Server is a extensible, enterprise-apt Java Platform, Enterprise Edition (Java EE) application server[5]. The WebLogic Server groundwork assist the deployment of innumerable types of scattered applications and is an absolute foundation for fabricating applications established on Service Oriented Architectures (SOA)[5].
Weblogic is integral part of Oracle Fusion Middleware. Middleware is a computer software that bridges software constituents or applications. Middleware is used most usually to

support intricate, dispersed buisness software applications. Middleware comprises web-server, application servers, content management systems, and alike tools that support application build-out and distribution[5].
Training in Weblogic constituted detail training of configuring, troubleshooting and using weblogic server. Majority of work run with OFM, and hence used entirely in the project.

Internal Team Training

Internal Training included introduction to working of the teams. Working of DSS team-app, build and Db. And internal training for RDS team working. Introduction to refresh procedure, which needs automation.

2.2.2 Retail Demonstration Service

Introduction

Retail Demonstration Services, provide technical infrastructure for demonstration environments. RDS also builds, upgrades and carry out smoke testing of Oracle Retail software suite for demos. It also provide support for bakcup & recovery, refreshing of environments and troubleshooting. These demo environments are used by pre sales team to give demonstraion of Oracle Retail products to the prospective clients.

Working

RDS manages 60+ Physical Servers (Hosts) running OracleVM (Guests). 7-8 Virtual Machines running on a variety of Physical Servers making up one demo environment. Each environment is independent from one another. Each environment consist of 8 VMs:

- DB (Database)
- RIB (Retail Integration Bus)
- OID (Oracle Internet Directory) & MoM (Merchandise Operation Management)
- RPAS (Retail Predictive Application Server)
- Allocation, ReIM (Retail Inventory Management), SIM (Store Inventory Management)
- PoS(Point of Sales)

- ATG
- CLIENT

2.2.3 Oracle Weblogic Server

Introduction

Oracle WebLogic Server is a an application server. It is scalable, Java Platform, Enterprise Edition (Java EE) ready to be used for enterprise. The deployment of many types of distributed applications is supported by the WebLogic Server infrastructure and is also known to be an ideal base for building applications based on Service Oriented Architectures (SOA)[5]. The WebLogic Server implementation provides a standard set of APIs for creating distributed Java applications that can access a wide variety of services, such as databases, messaging services, and connections to external enterprise systems[6]. Web browsers are used by end-user clients access these applications. WebLogic Server also etitles enterprises to deploy critical applications in a sturdy, secure, ubiquitous, and scalable environment[6]. These features allow buisnesses to design clusters of WebLogic Server instances to diffuse load, and provide additional potential in case of hardware or any other failures[6]. WebLogic Server can also be configured to observe and adjust application throughput automatically without human intervention.

2.2.4 Administration

System administration of WebLogic Server includes variety of tasks: creating WebLogic Server domains; establishing applications; relocating domains from development environments to production environments; monitoring the performance of the run-time system; and detecting and troubleshooting problems[7].
A WebLogic Server domain is a assemblage of WebLogic Server services developed for a specific tasks.

2.2.5 WebLogic Server Domain

Introduction

A domain is the elementary administration element for WebLogic Server instances. A domain can consist of more than one WebLogic Server instances that is managed by a distinct Administration Server[8]. Multiple domains can be defined based on disparate

system administrators' responsibilities, application frontiers, or geographical positions of servers. Contrarily, A solitary domain to consolidate all WebLogic Server administration activities can also be used.

Each domain has a configurations which are stored in a different configuration file called config.xml, which is stored on the Administration Server along with additional files such as logs and security files. When Administration Server is used to carry out a configuration errand, the changes made are applied only to the domain governed by that Administration Server. Hence, the servers instances, applications, and resources in one domain are impartial of servers, applications, and resources in a dissimilar domain. Configuration or distribution tasks in manifold domains cannot be carried out in a parallel.

Content of Domain

A domain can incorporate numerous WebLogic Server clusters and non-clustered WebLogic Server instances[8]. A domain can be compromised of only one WebLogic Server instance, which acts as an Administration Server ass well as a Managed server. Such a particular domain can be useful during application development, but is not approved for use in a production environment. However the extent and intend of a domain can differ notably, most WebLogic Server domains contain the following constituents[8]:

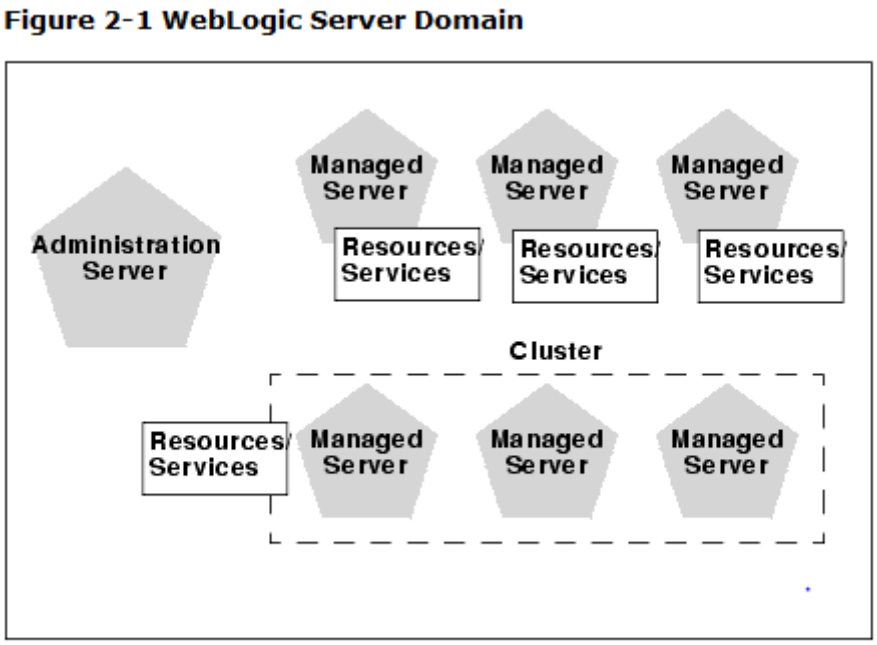


Figure 2.1: Weblogic Server Domain

- Administration Server: The Administration Server is considered as chief regulatory body for the configuration of the whole domain [9]. It preserves the domains configuration documents and manages alterations in the configuration documents of Managed Servers. The Administration Server can additionally perform as a central point to monitor all amenities in a domain.

- Managed Severs And Managed Server Cluster: Managed Servers hold application constituents, business applications, Web services, and their associated resources [9]. Managed Servers preserve a read-only copy of the domain's configuration document to increase the performance. Managed Server connects to the domain's Administration Server on boot up to harmonize its configuration document with the document that the Administration Server maintains[9].

- Resources and Services: Along with the Administration Server and Managed Servers, the resources and services that Managed Servers and deployed applications require are also present in a domain[9].

Chapter 3

rdsAll Script Automation

3.1 Introduction

The tasks for environment enhancement include monitoring and automation. Monitoring include creating monitoring scripts in python for VM monitoring. And creation of electronic refresh checklist. Automation tasks include automation of domain creation, rdsAll script and automation of OVM tool.

3.2 rdsAll Script Automation

- **Task:** rdsAll script brings up all the services in RDS. Utilise the funcionality of "RC" folder and automate rdsAll.

rdsAll script is a script written in Unix to bring up the services in VMs. The script is responsible for starting the weblogic servers for each VMs.

Why automation is needed?

Each quaterly interval of year, Oracle servers undergoes maintainance procedure. The procedure includes power outage of servers and restarting them. The restarting of servers leads to stopping of VMs and they need to be restarted as well. Since on single server there may be multiple VMs running, restarting all the services in VMs by logging in into them becomes a tedious task. Hence, the idea of automation is to write a script which will start the services of VMs automatically on restart of server.

3.2.1 Approach for automation

The basic idea for automation is to use the rdsAll script as a service. The service will start automatically on system boot up i.e. whenever the VM restarts.

Basics: The system startup files are present at location /etc directory[10]. They are:

/etc/rc

/etc/rc.d/*

/etc/rc.conf

/etc/rc.lkm

/etc/rc.shutdown

/etc/rc.local

/etc/rc.conf.d/*

/etc/rc.subr

/etc/defaults/*

After the kernel has initialized all devices at start-up, it starts init, which in turn runs /etc/rc.

/etc/rc classify the scripts in /etc/rc.d using rcorder and then runs them in that sorted sequence[10].

rc.d: The scripts that control services are in /etc/rc.d. These scripts are accordingly run at boot time. One of the following arguments are passed to rc.d scripts: start stop restart status The startup system of every Unix system vary and determines the order in which services are started. On some Unix systems this is done by accounting the files and/or putting them in distinct run level directories.

Runlevels[11]: The directory /etc/rc.d contains a set of files named rc.0, rc.1, rc.2, rc.3, rc.4, rc.5, and rc.6, or a set of directories named rc0.d, rc1.d, rc2.d, rc3.d, rc4.d, rc5.d, and rc6.d. During boot time, the system uses these files (and/or directories) to supervise the services to be started. The boot process uses these parameters to pinpoint the default runlevel and the files that will be utilized by that runlevel. Runlevels[11] refers to various levels that the system experience during a boot up. Runlevel 1 is the basic configuration (simple single user access using an text interface), while runlevel 5 is the advanced (multi-user, networking, and a GUI front end). Runlevels 0 and 6 are used for halting and rebooting the system [11].

ID	Name	Description
0	Halt	Shuts down the system
1	Single-user mode	Mode for administrative tasks
2	Multi-user mode	Does not configure network interfaces and does not expor
3	Multi-user Mode with Networking	Start the system normally
4	User-Defined	For special purposes
5	Start the system normally with GUI	As runlevel 3 + display manager
6	Reboot	Reboots the system.

Table 3.1: Runlevels and Modes

3.3 Challenge

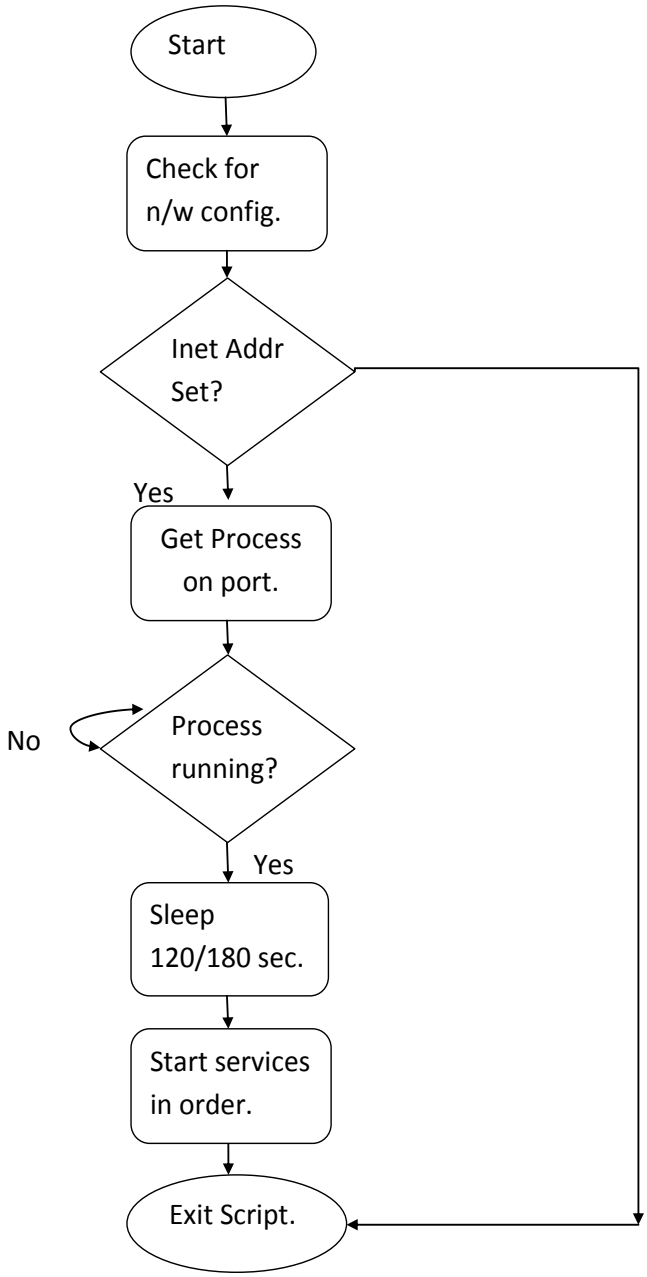
Each 8 VMs has interdependent services. Hence the services does not start up until the underlying dependent services are up and running. As in every VM need DB to be up before starting. MOM applications are dependent on RIB, and Stores on MOM. Hence these services are needed to be started in particular order. The order is across the VMs hence difficult to maintain.

Another challenge is to update a Master copy with the scripts. When VMs are created using the Master Copy for the first time. The network is not configured. And the auto-script since runs at boot time and requires to check port no. expect the network to be configured. This creates a deadlock. The VMs don't start to allow to configure network and the script does not let the VM start since it is waiting for network configurations.

3.4 Implementation:

rc3 as runlevel is used to start the system normally with networking. Since order of start is to be maintained, the solution is to check whether the underlying dependent service is up before starting the service. To check this the port no. of weblogic server which is dependent is checked. netcat command is used to check for the port no across the VMs. When the dependent services are up a new service is started, otherwise the script waits for dependent service to come up. To deal with Master copy updation. The script was modified to check if network configurations are completed. If network configurations are not done, do not allow rdsAll script to stop VMs.

Figure 3.1: Flow chart for rdsAll script



Chapter 4

Monitoring Script

4.1 Monitoring Script

- **Task:** Understand the functioning of server monitoring scripts and script in python.

The monitoring scripts for VMs are needed to check the status of each working VM. Monitoring includes vmstatus - to check the available memory, wait time, idle time etc.To check whether the VMs are up and running successfully.To check diskspace on the VM.The scripts are written in python.The python script executes the unix commands in remote VMs and return the results.The results are examined and appropriate action is notified.

4.2 Approach

Monitoring of VMs is needed to make sure the environment is working properly. The parameters that are monitored are diskspace. Sufficient diskspace is needed for services to startup. Hence if disk space is not available the monitoring script is supposed to notify the team as possible disruption of services.

The other parameter to be monitored is status of VM. Status inline with wait time, processing time, idle time, blocks processed, I/O stream etc.

The most important parameter is to ensure the VMs are up and running always. It is monitored by pinging the VMs at regular interval to ensure they are running.

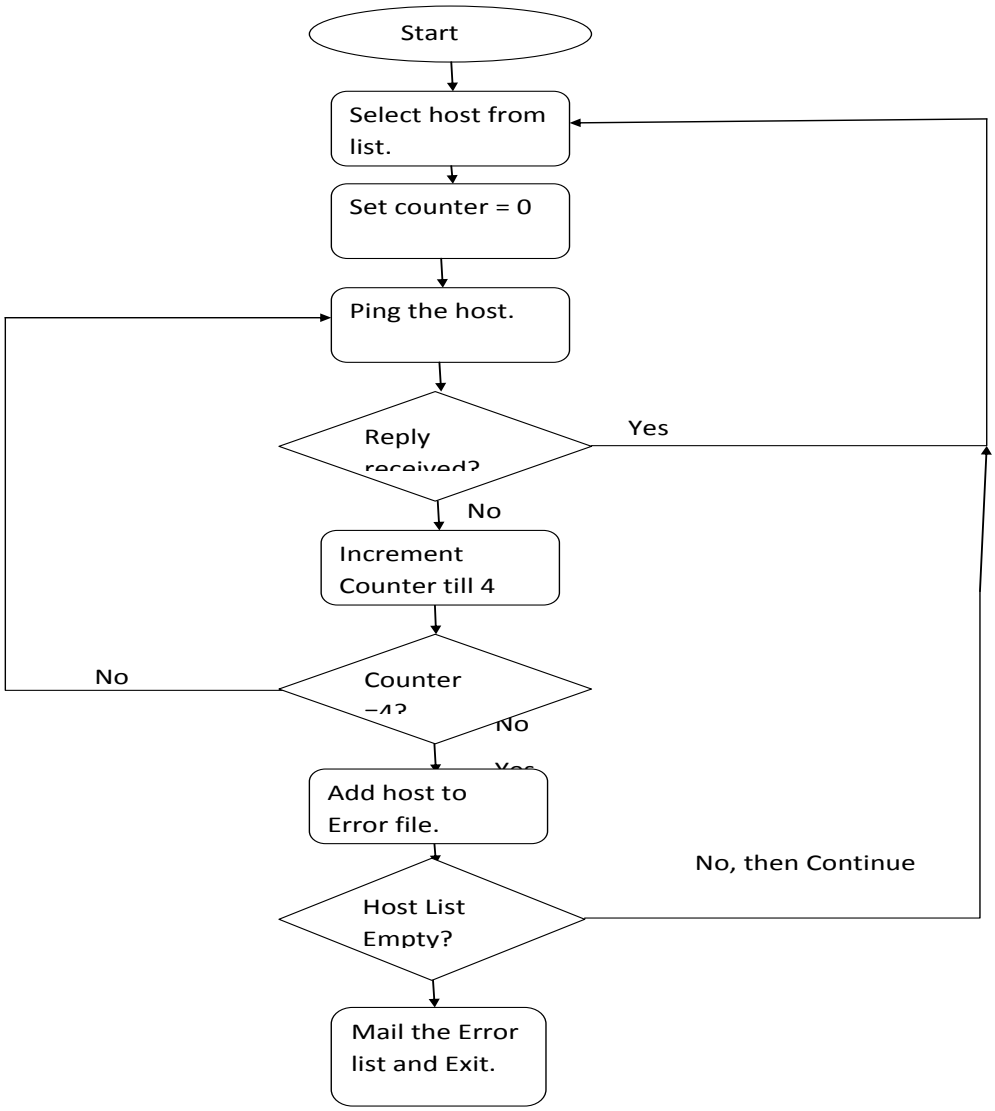
4.3 Implementation:

The scripts for monitoring are written in python. Python is used since it provides more optimized code. The python code uses shell commands to monitor the VMs, since they are based on linux operating system. The scripts are run regularly within predefined interval.

4.3.1 VM monitoring

This script checks whether the VM is up and working. The VMs are pinged to get an expected reply. If the reply matches the VM is running smoothly otherwise the error is notified via mail.

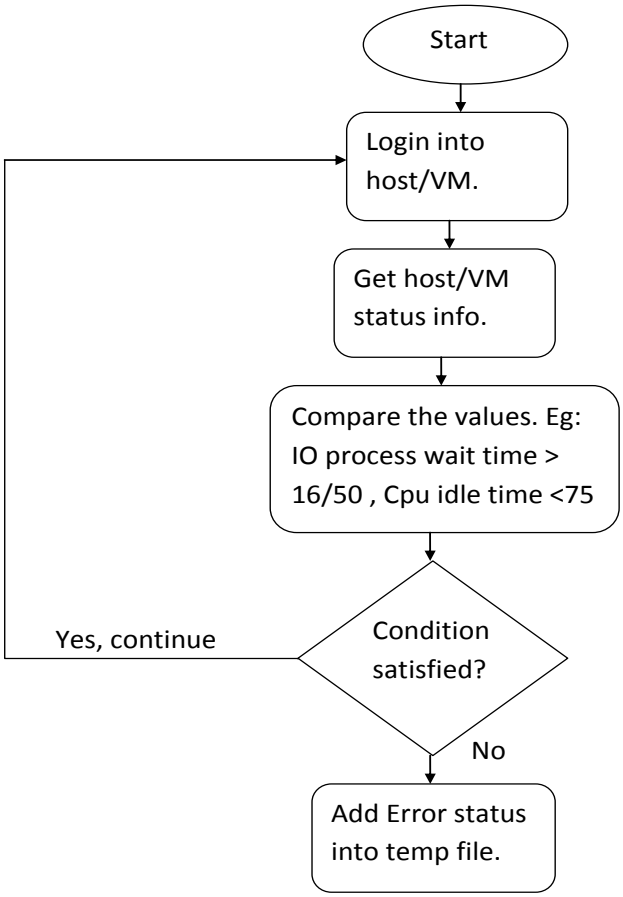
Figure 4.1: Flow chart of VM monitoring



4.3.2 System and VM monitoring

This script checks for the health of VM. The status include I/O process time, wait time for CPU, IDLE time for CPU, amount of virtual memory, amount of idle memory, No of processes waiting and in uninterrupted mode etc. Each parameter is compared with threshold limit decided for each respectively. If the status is varied than threshold appropriate error message is delivered via mail.

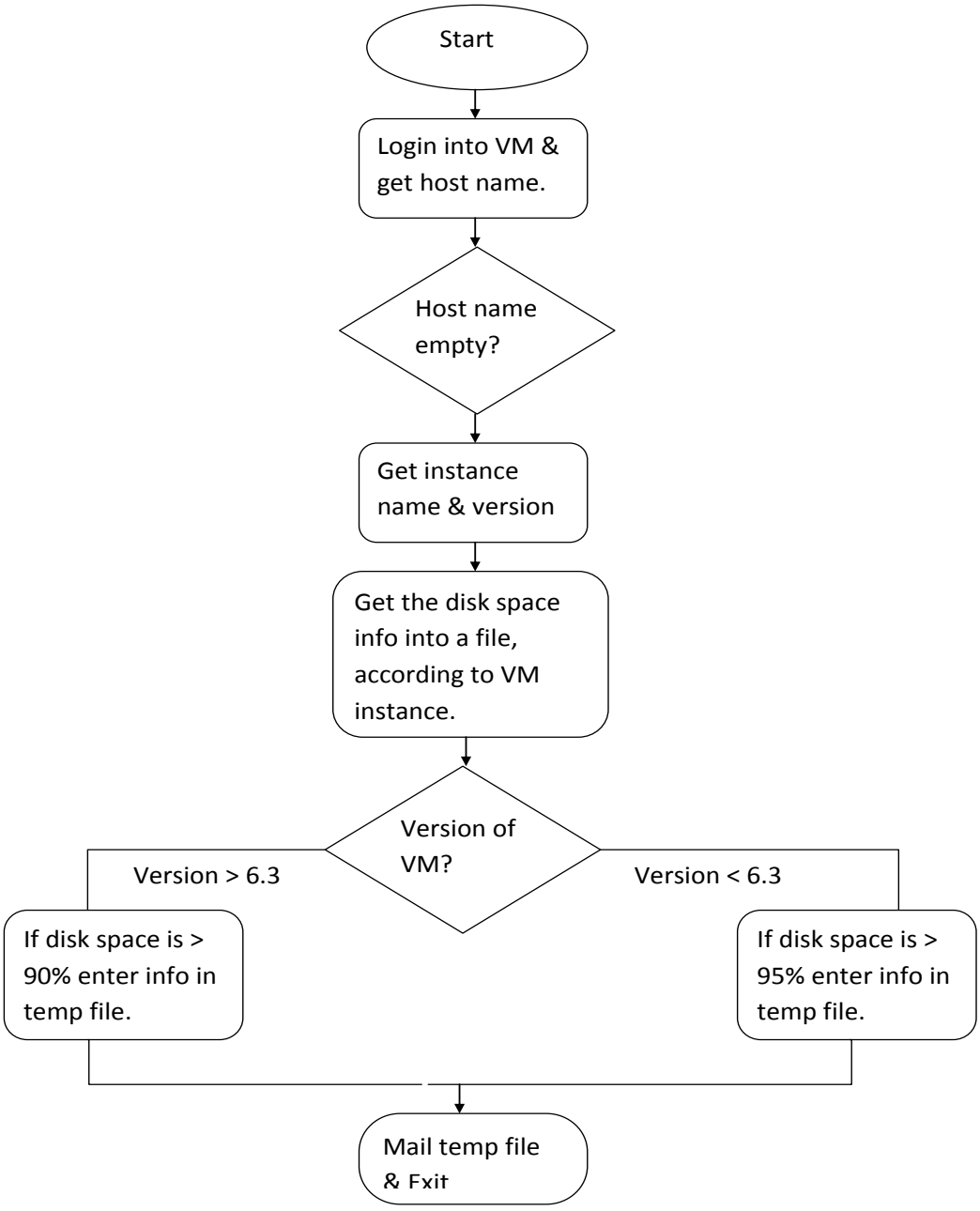
Figure 4.2: Flow chart for System monitoring



4.3.3 Disk space monitoring

This script checks the disk space in the VMs. The threshold limit of free disk space varies for every version. So sorting the input VM according to its version, the free space is compared to threshold. If the free space is less than specified limit, error message is generated.

Figure 4.3: Flow chart for Disk Space Monitoring



-

Chapter 5

Automation of Weblogic Domain Creation

5.1 Approach

Oracle Weblogic Scripting Tool(WLST) is a command-line scripting framework that can be used to create, superintend and observe Oracle Weblogic Server Domains[12]. It is based on the Java scripting interpreter, Jython. WLST provides a set of commands that are particularly used for Weblogic Server.

Using Weblogic Offline: Without connecting to a ongoing WebLogic Server instance, WLST can be used to fabricate domain templates, create a new domain rooted on prevailing templates, or expand an existing, non-functional domain [12]

5.2 Challenge

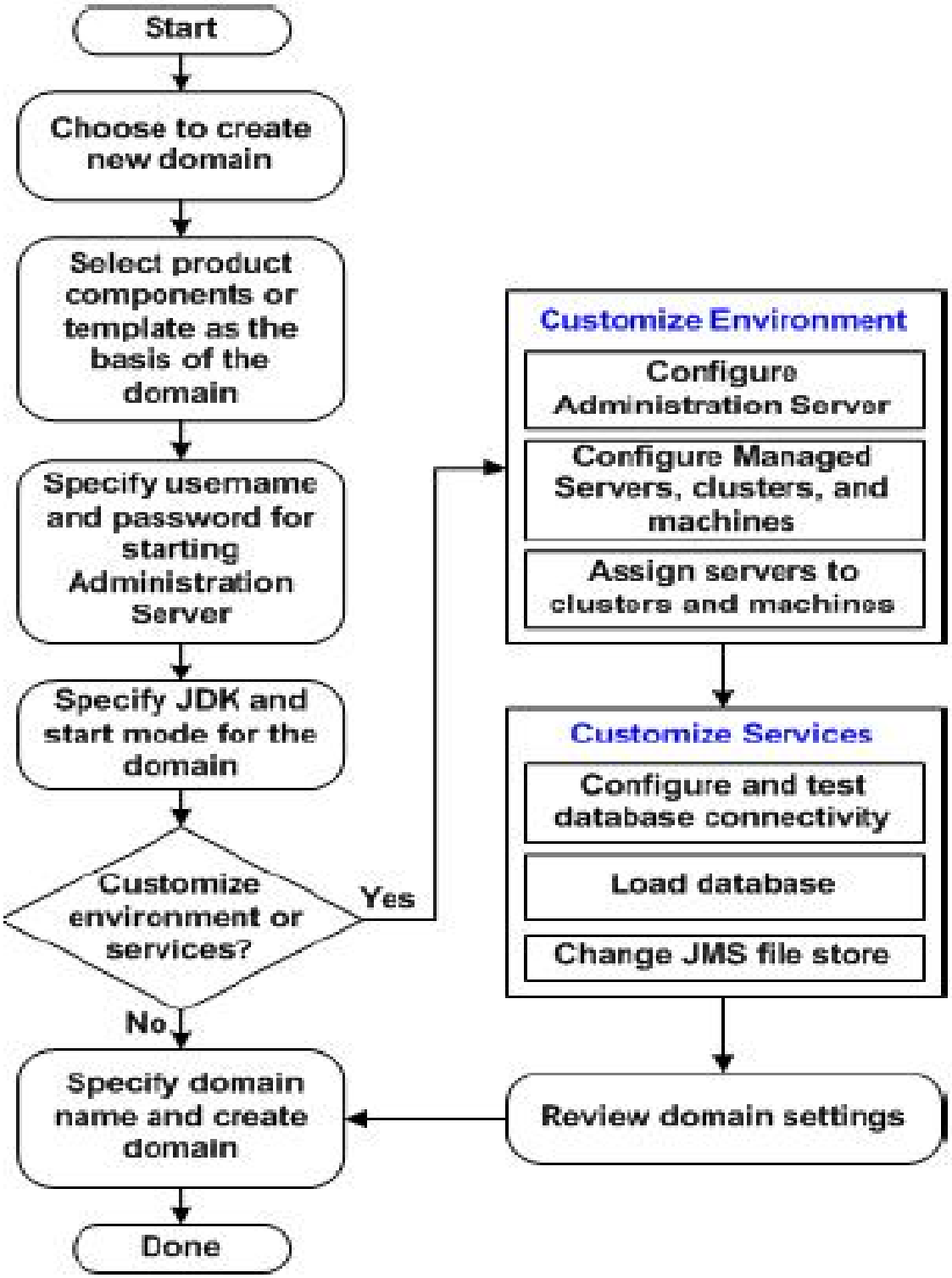
Each domain has unique identifiers and requirements. To accommodate these varied requirements is major challenge in automating Domain creation.

5.3 Implementation:

A domain template is a JAR file that contains domain configuration documents, applications, security data, startup scripts, and other information needed to create a WebLogic domain. Using WLST commands, the template is used and configured according to needs of the domain. The automation is performed using python scripting along with WLST.

Domain Creation [5]

Figure 5.1: Weblogic Domain Creation



Chapter 6

RDS Automation Tool

An Automation Tool for Refresh, Restore And Backup of instance environments.Tasks in tool include updating the instance with newer version, and network configurations. Also taking backup of current version and storing it in appropriate provided location.

6.1 Approach

The tool is based on automation of RDS tasks of refresh , restore and backup. It is web-application which runs the scripts in the background. The script consist of back-end shell scripts which perform the required tasks on VMs. The network configurations of VMs is done using Perl scripts. The desktop tool has front-end written in Flex, And integration is written in Java.

6.2 Challenge

The major challenge in creating the tool was to get all the RDS environments in uniform format. The prerequisites included collecting Environment information from all instances, Getting IP and host files for environments on shared location, to check if NAS mounts are located at every environment etc.

Another challenge in the tool was to deal with windows and unix based VMs with same scripts. Perl script deals with both environments, it helps run batch as well ass shell scripts.

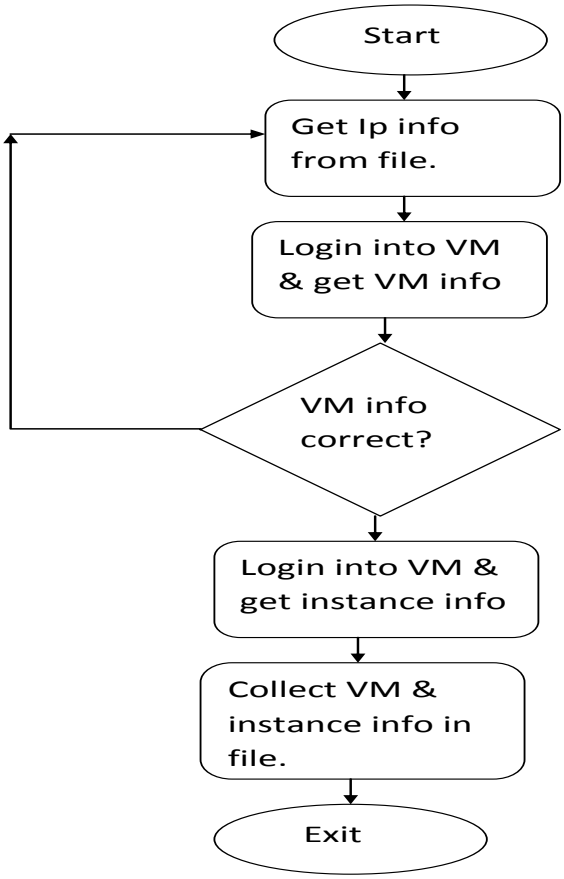
6.2.1 Prerequisite Scripts

Host-Instance

The script basically is written to collect the details of all the hosts in RDS and all instances running in each host. This files basically forms input for the tool to determine the host and its corresponding environment in shell script.

The script first logs in into VM using IP from shared input sheet, it gets VM name i.e. host name and list of instances running on that particular host and saves into file on shared location.

Figure 6.1: Flow chart for host-instance script

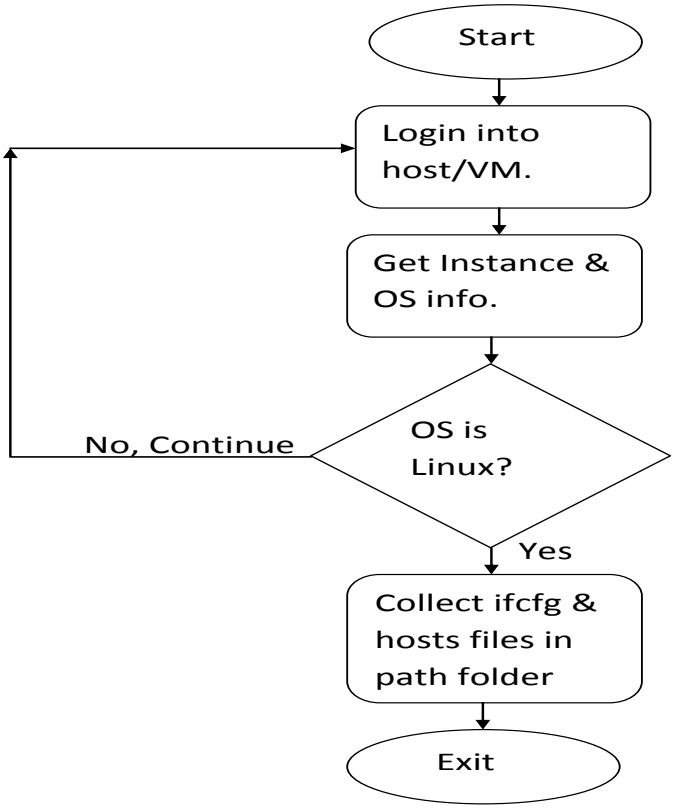


Environment Info.

This script collects all the ifcfg-eth0 files from all the VMs in every instance and stores it at shared location. Also host files are collected. These files are needed to configure the network on every update, refresh or backup. Hence, the files are to be kept up-to-date.

The script logs in into VM and checks if it is unix machine, for windows batch script is needed. If unix machine in gets ifcfg-eth0 file from network-scripts and hosts file from etc and copies both these files into shared location

Figure 6.2: Flow chart of Environment Info.

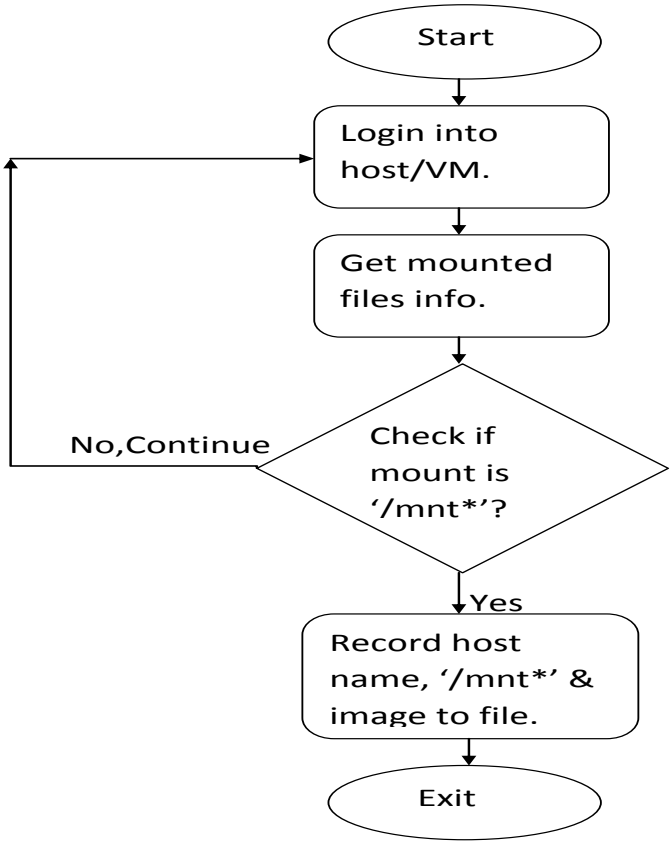


Check RDS mount

The shared location for keeping source files and automation scripts is at NAS mount, which is mounted on each VM as particular location. The presence of mount on that location is necessary for automation to work. Hence this script was written to check for rds mount.

The script logs in into VM and checks if NAS is mounted. If mounted them record the path to the mount. If path differs then change it according to needed path.

Figure 6.3: Flow chart for check-mount script



6.3 Implementation

6.3.1 Back-end Script-Shell scripts

The part of back-end scripts are written in shell and other half in Perl. The shell script are used for restore and backup part of the tool.

Restore Script

Restore of environment is done if the client requests for older version of environment. The backup of the instance according version is maintained, and if requested is restored on the host.

The environment files are needed to be copied on instance location and unzipped. Then these files are restored and appropriate changes are to be done. The VMs are then started and the Perl scripts of the project take care of network configuration. The services are started using rdsAll automation project.

The restore procedure differs based on type of server. RDS environments contain two types of servers- file based and LVM based. File based are normal file servers, LVM based use concept of logical volumes to host data files.

The script takes instance name and backup name as input. Instance name is needed to get the host name and backup is the file that is to be restored. If the environment is up, then it has to be brought down to begin restore. Then the server type is checked, if it is file based or LVM based server.

For File based server the backup files are to be copied in the restore file location. The copy can be performed locally in the VM if backup is present at the mount mounted on the VM. Else, host to host copy is performed from NAS to the VM file location. After completing the copy, the zipped files are extracted. The config files are gathered and VMs are started.

Figure 6.4: Flow chart for Restore Script I

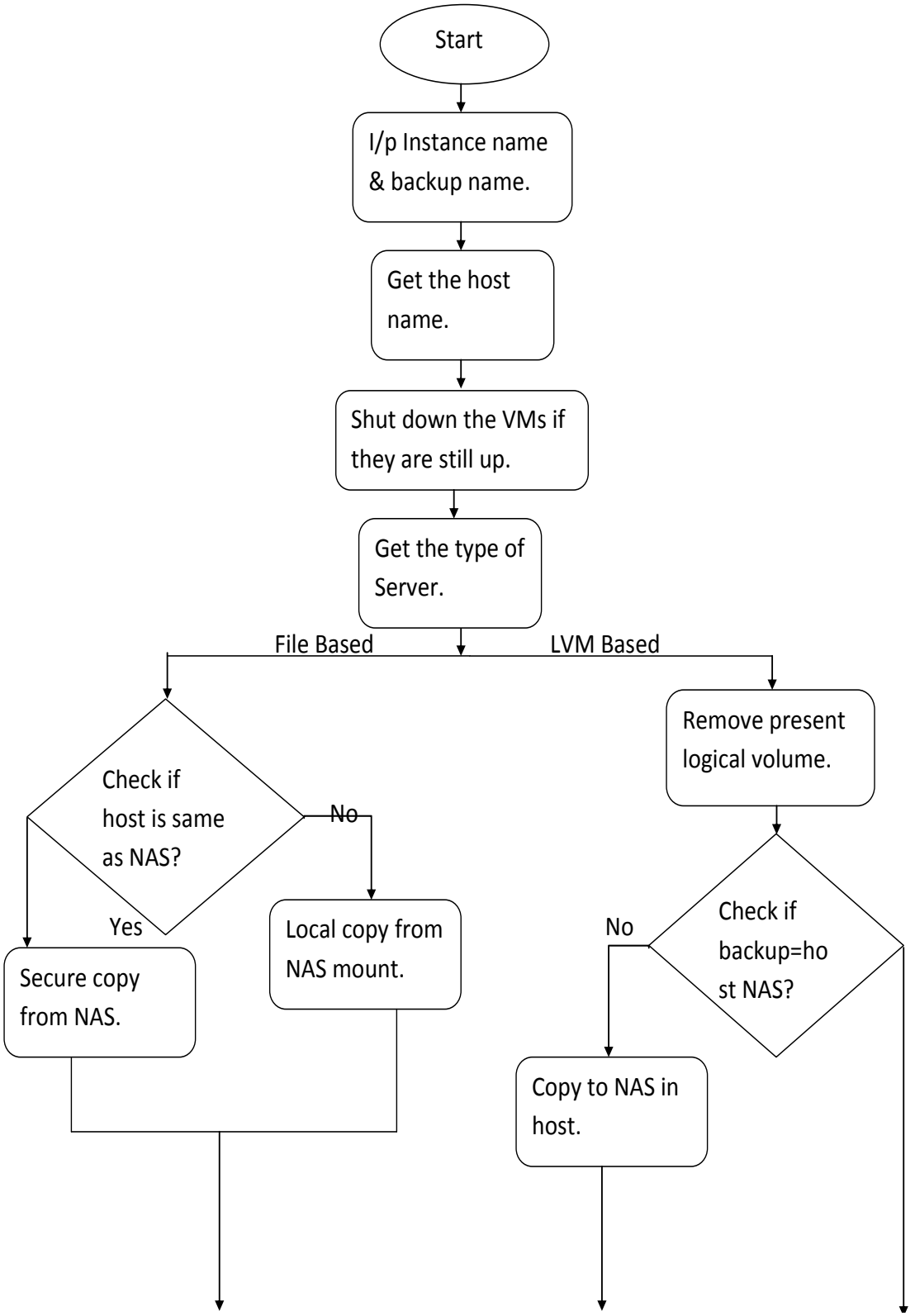
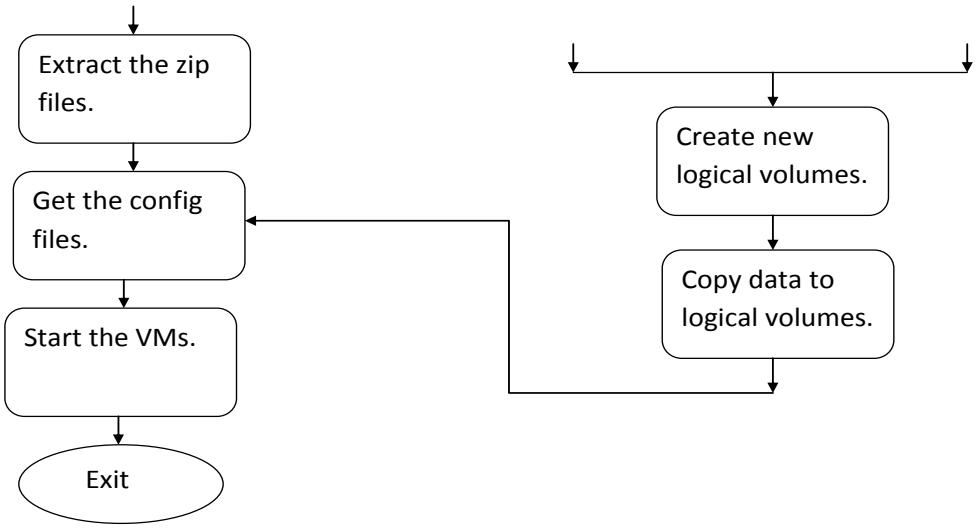


Figure 6.5: Flow chart for Restore Script II

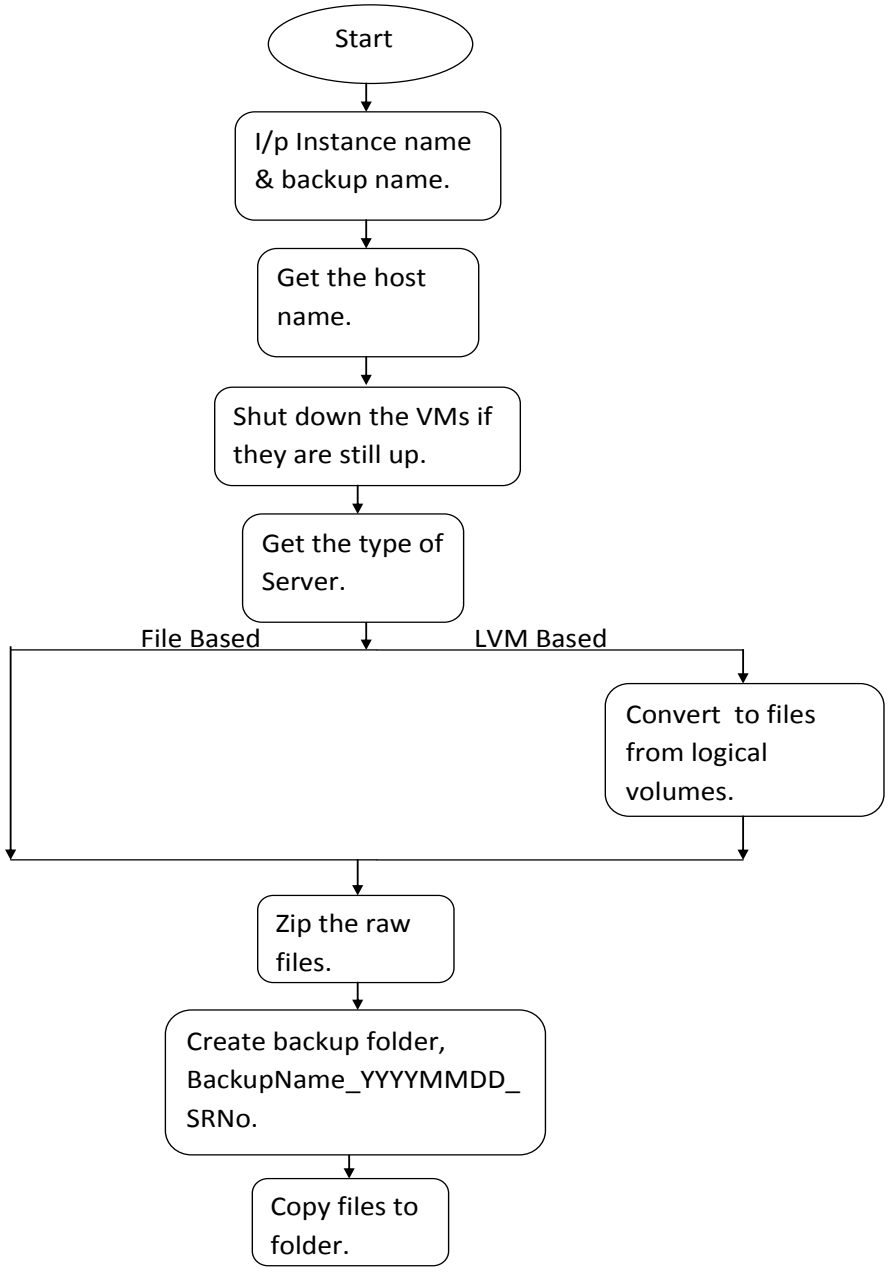


Backup Script

The other use of tool is to take backup of current environment. The backup is saved at shared location on NAS. The backup script takes instance name, backup name and SR no as input. For backup as weel the environment need to be brought down if its running in the host. After shutting down, the data files are zipped. Backup folder is created at local NAS mount. The format for folder is predefined, i.e. BackupName-YYYYMMDD-SrNo.

For LVM based servers the data is to be converted into files from logical volume before zipping them. The zipped files are then copied to new created folder and original files are deleted or restarted based on request.

Figure 6.6: Flow chart for Backup Script



6.3.2 Front-end Scripts

Front End Scripts are written in Flex. Flex is a highly productive, open source application framework for building and maintaining expressive web applications that deploy consistently on all major browsers, desktops, and devices [13]. It provides a modern, standards-based language and programming model that supports common design patterns suitable for developers from manybackgrounds[13].

Flex has basic scripting part in xml and it uses jsp for navigation and authentication purpose. The look and feel is structured using XML. The flex project is imported in java project which integrates the front-end to the back-end scripts.

6.4 Snap Shots

Figure 6.7: Screen shot - Login Page

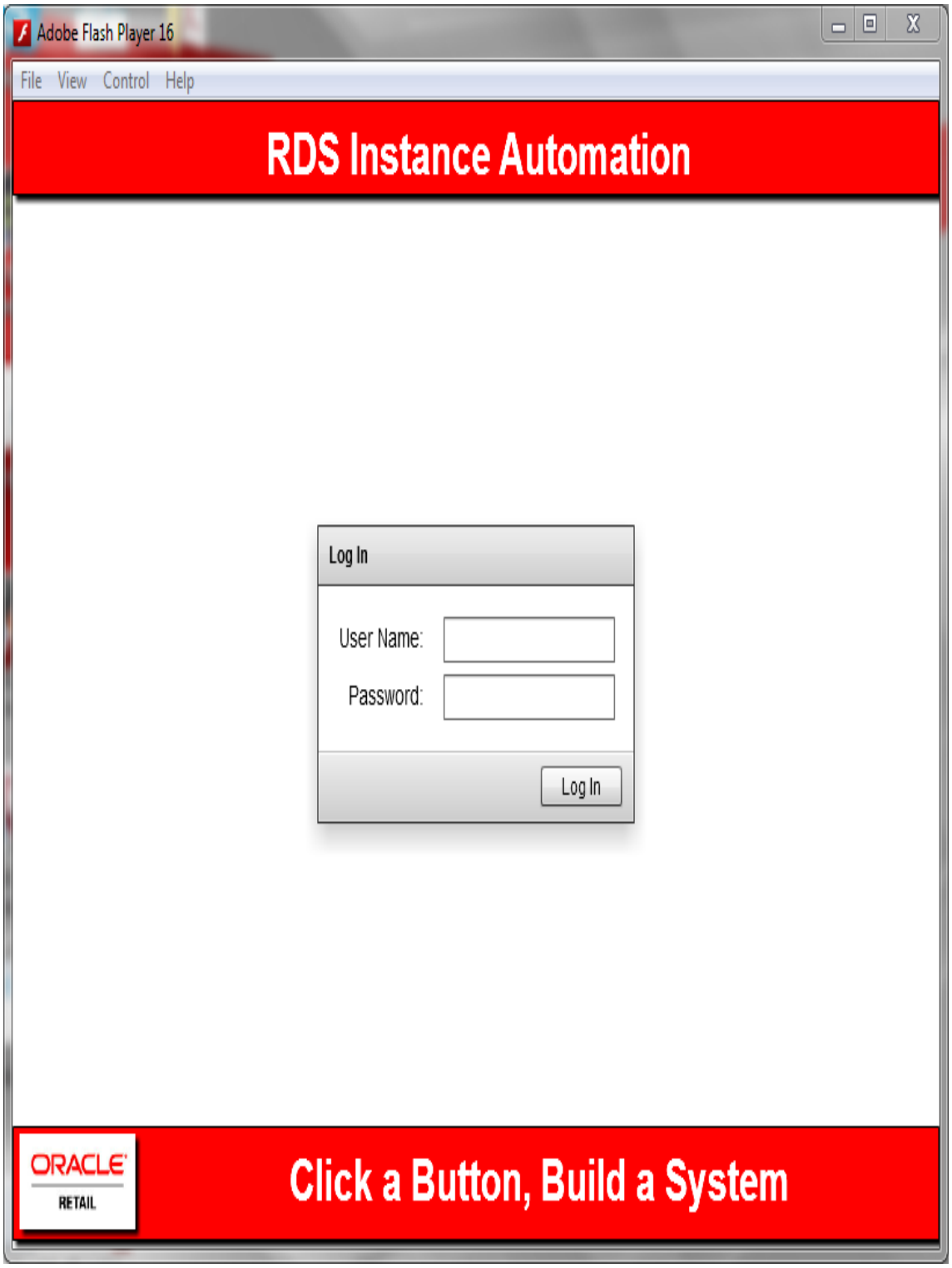


Figure 6.8: Screen shot - Request Type Page

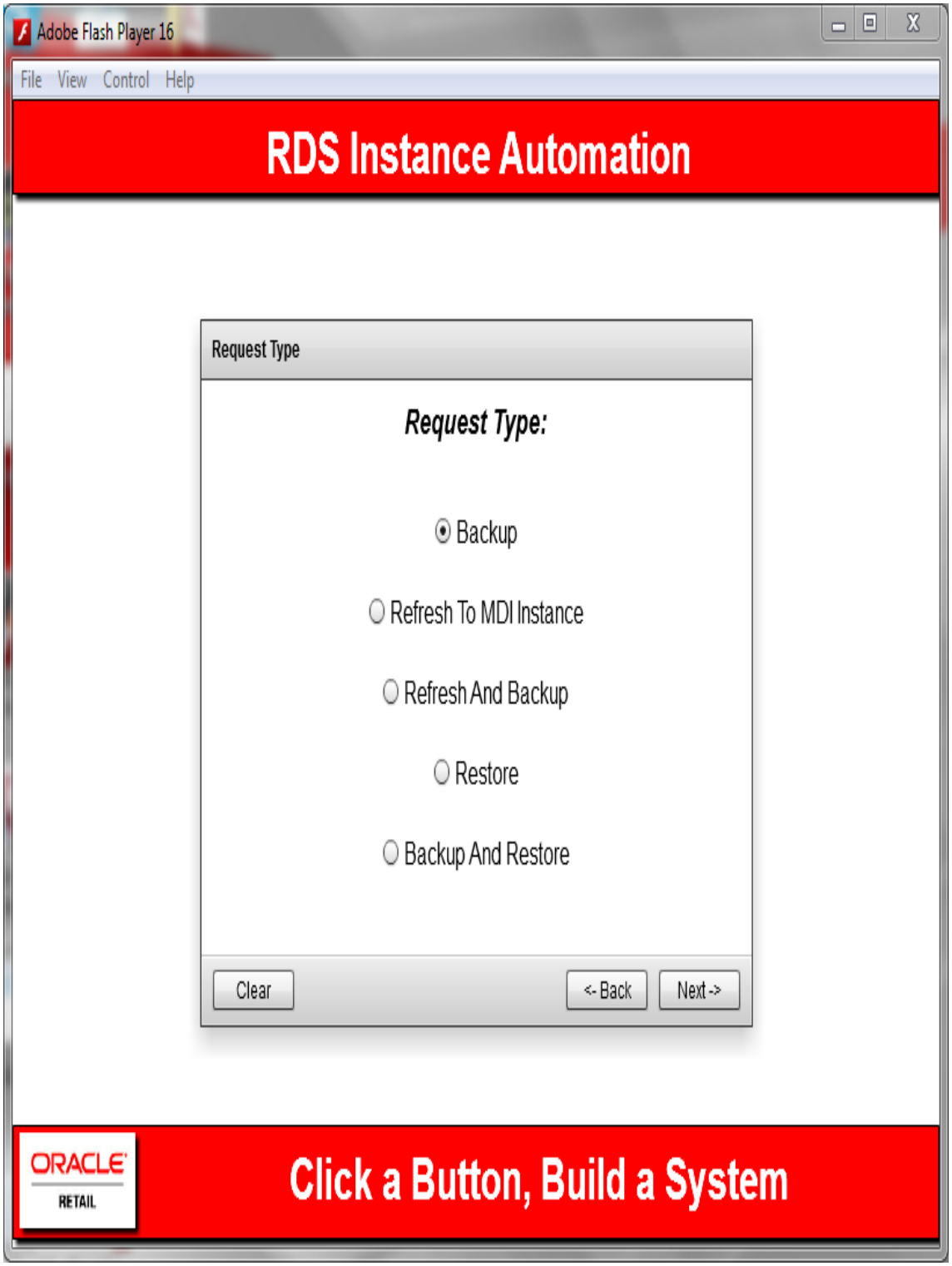


Figure 6.9: Screen shot - Backup Input Page

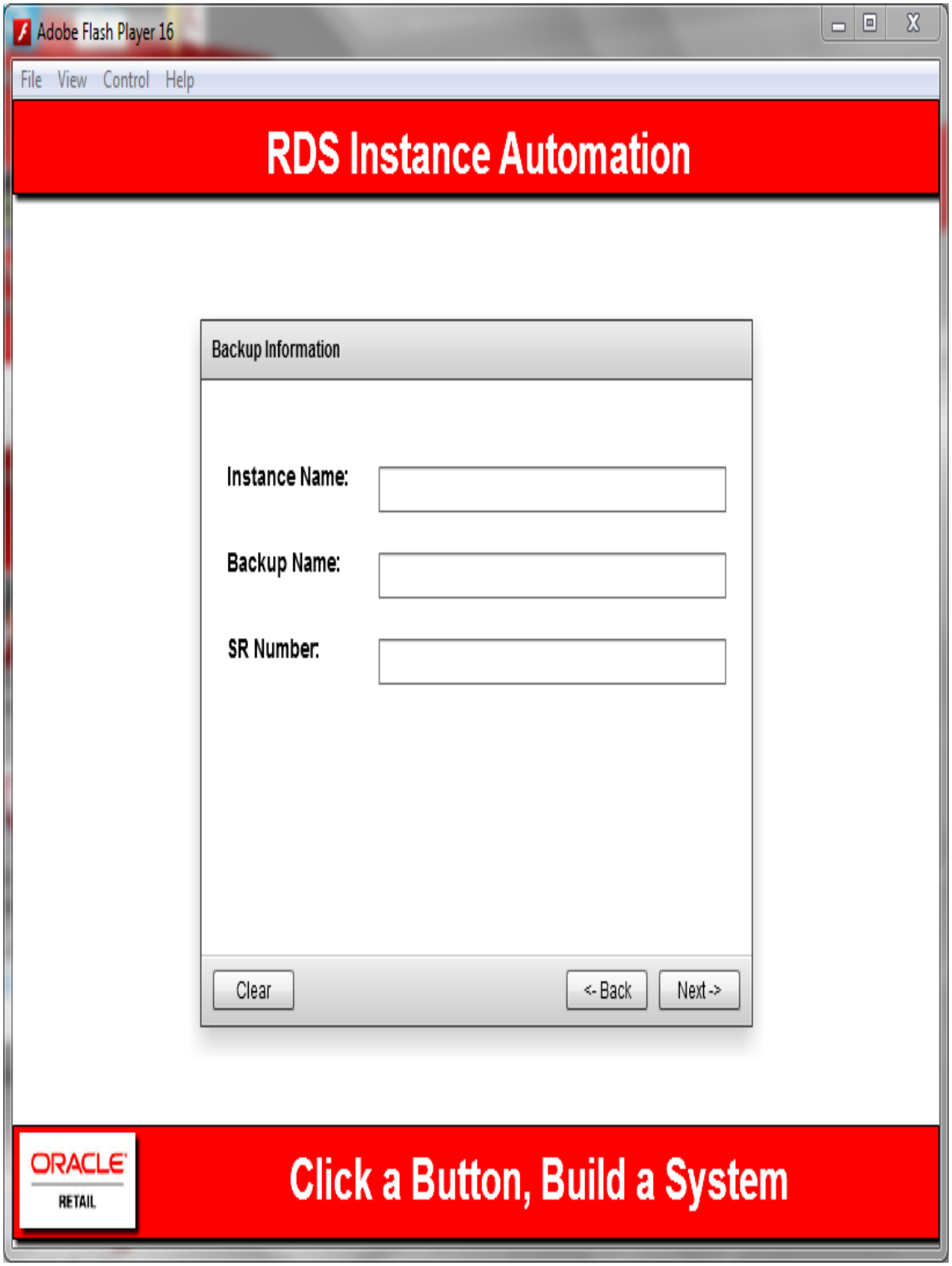


Figure 6.10: Screen shot - Restore Input Page

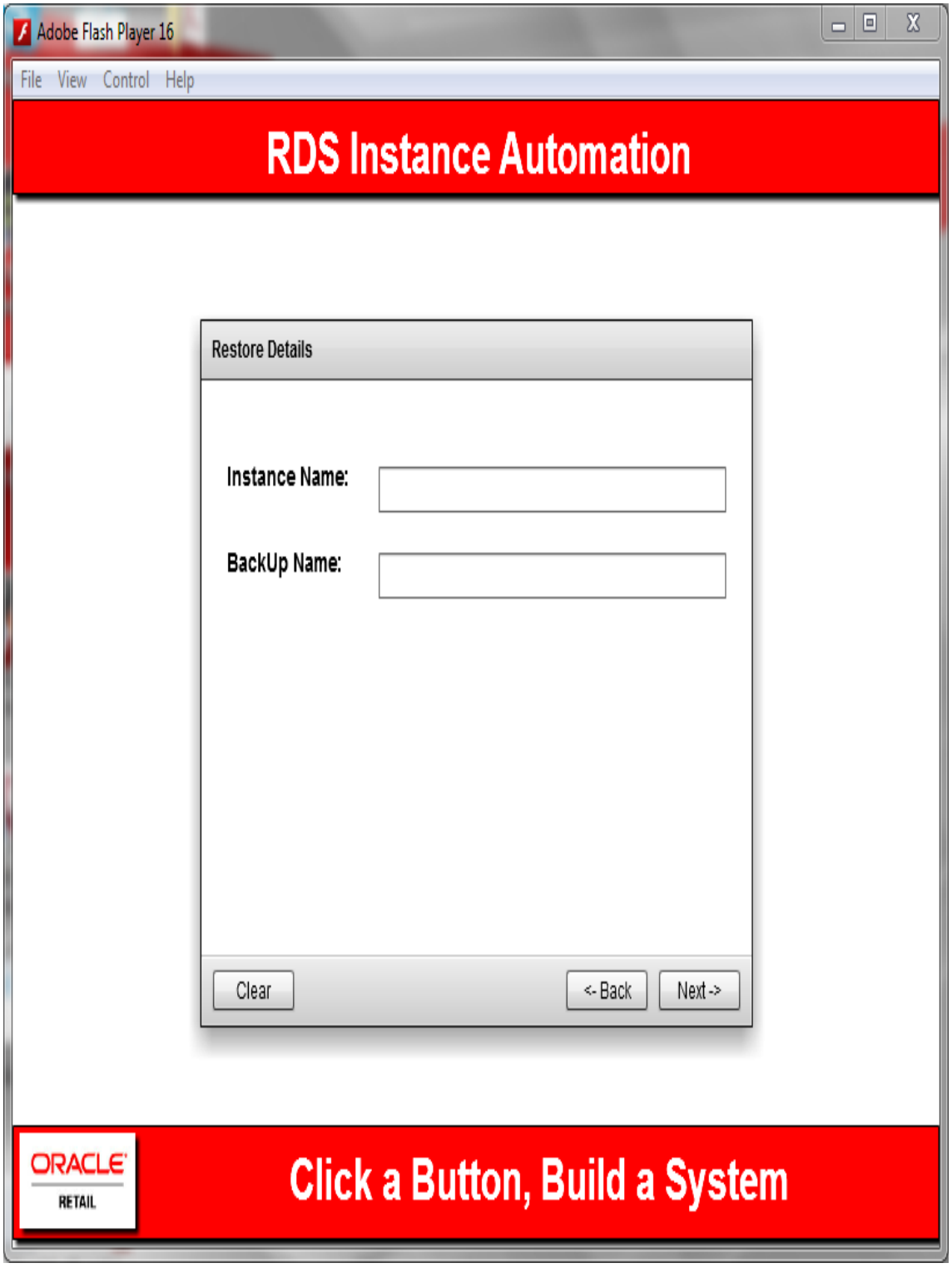


Figure 6.11: Screen shot - Backup And Restore Input Page

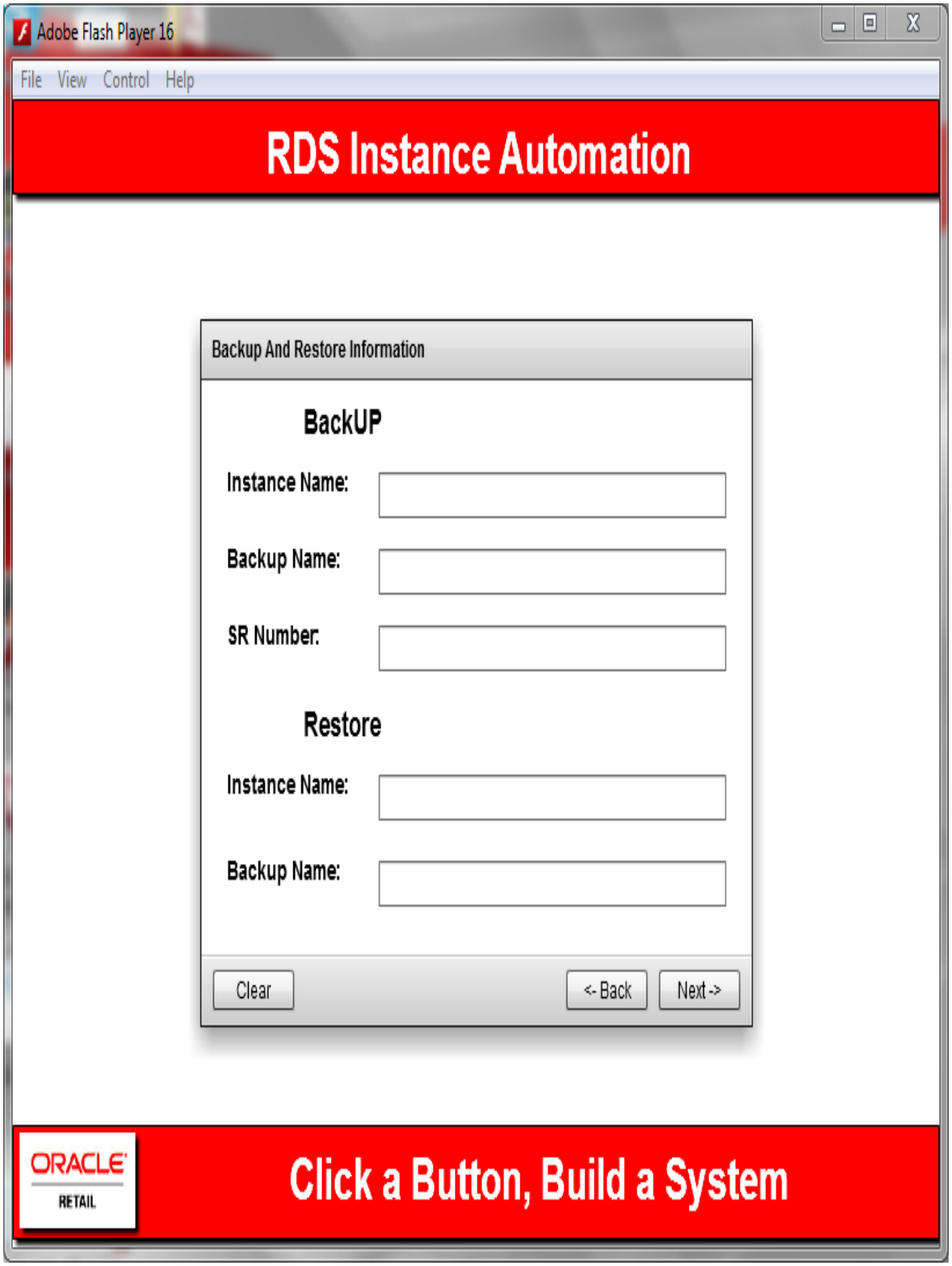
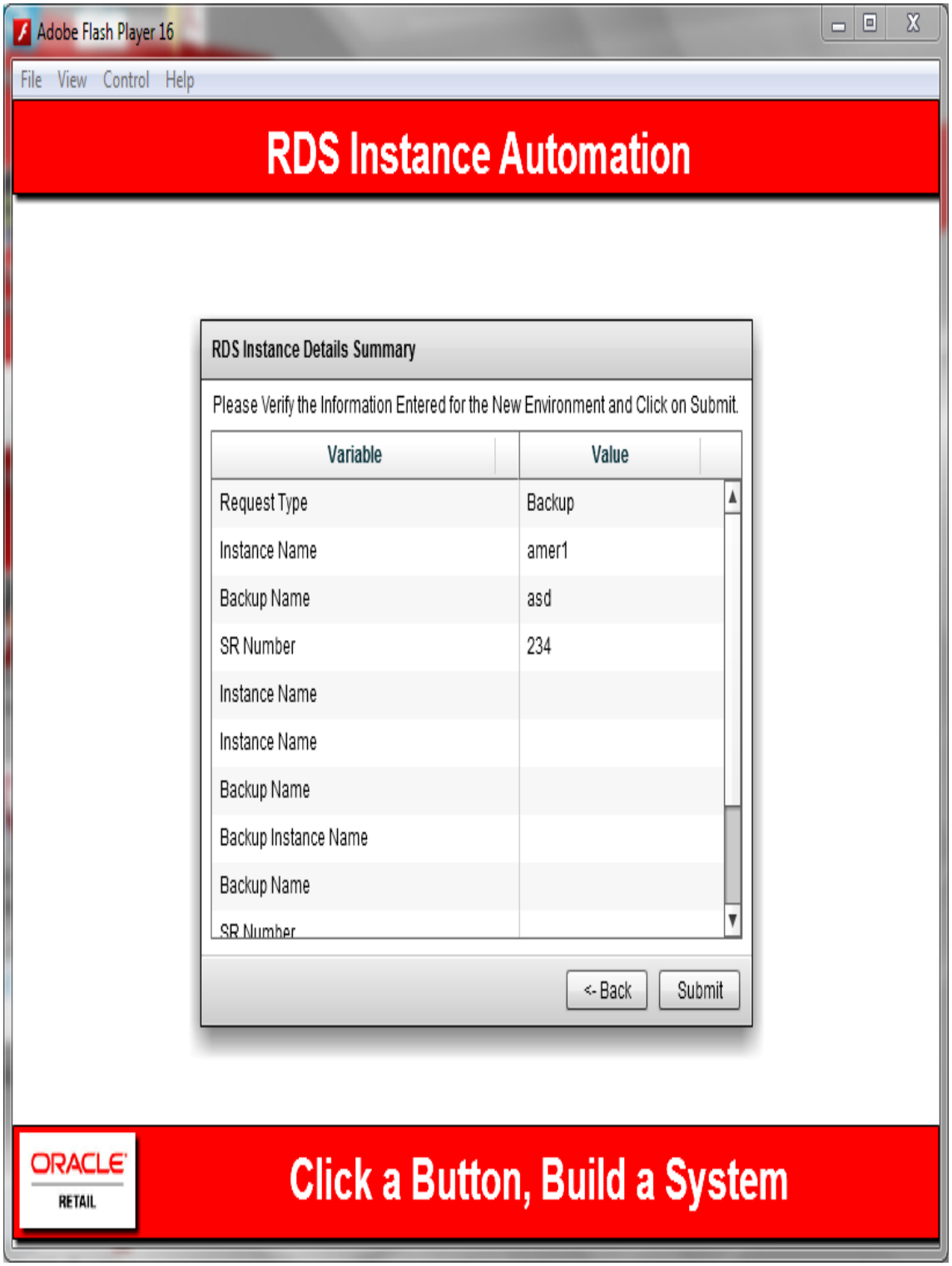


Figure 6.12: Screen shot - Confirmation Page



References

[1] *http://www.oracle.com/oms/retail/oracle-viewpoint-commerce-2156261.pdf.*

[2] *http://en.wikipedia.org/wiki/JavaScript.*

[3] *en.wikipedia.org/wiki/Shell_script.*

[4] S. H, *A Byte of Python.*

[5] *http://www.oracle.com/technetwork/middleware/weblogic/documentation/index.html.*

[6] *https://docs.oracle.com/middleware/1213/wls/INTRO/intro.htm#INTRO124.*

[7] *https://docs.oracle.com/middleware/1213/wls/INTRO/sysadmin.htm#INTRO139.*

[8] *https://docs.oracle.com/cd/E13222_01/wls/docs81/adminguide/overview_domain.html.*

[9] *http://docs.oracle.com/cd/E13222_01/wls/docs100/domain_config/understand_domains.html.*

[10] *https://www.netbsd.org/docs/guide/en/chap-rc.html.*

[11] *http://www.linux.com/news/enterprise/systems-management/8116-an-introduction-to-services-runlevels-and-rcd-scripts.*

[12] *https://docs.oracle.com/cd/E29542_01/nav/wlst.htm.*

[13] *http://www.adobe.com/in/products/flex.html.*