Design and Qualitative Assessment of Environment Build and its Automation

Thesis submitted in partial fulfillment of the requirements for the award of degree of

Master of Engineering
in
Information Security

Submitted By
Harmeet Kaur
(Roll No. - 801633004)

Under the Supervision of
Dr. Maninder Singh
Professor (CSED)

COMPUTER SCIENCE AND ENGINEERING DEPARTMENT
THAPAR INSTITUTE OF ENGINEERING AND TECHNOLOGY
PATIALA – 147004
JUNE 2018

CERTIFICATE

I hereby certify that this work which is being presented in the thesis entitled, *" Design and Qualitative Assessment of Environment Build and its Automation"*, in the partial fulfillment of the requirements for the award of the degree of Master of Engineering in *Information Security* submitted in Computer Science and Engineering Department of Thapar Institute of Engineering & Technology, Patiala, is an authentic record of my own work carried out under the supervision of Dr. Maninder Singh and refers other researcher's work which are duly listed in the reference section.

This matter presented in the thesis has not been submitted for award of any other degree of this or any other University.

Harmeet Kaur

(Harmeet Kaur)

This is to certify that the above statement made by the candidate is correct and true to the best of my knowledge.

Dr. Maninder Singh

Professor

CSED

ACKNOWLEDGEMENT

I would like to express my deepest appreciation to Dr. Maninder Singh, my mentor and thesis supervisor for his constant support and motivation. He had been instrumental in guiding me throughout the thesis with his valuable insights, constructive criticisms and interminable encouragement.

I am thankful to my project mentor Mr. Justin Bye, Principal Software Developer, and my manager Mr. Paul Kehler, Software Development Director at Oracle Retail for giving me an opportunity to work on this project, for providing timely help and feedback and their help in solving the problems faced. I would also like to thank my team members, Mr. Santhosh Kumar NC and Mr. Venkata Puneeth for their valuable support and guidance during my internship.

I would like to thank all the faculty members and staff of the department who were always there at the hour and provided with all the help and facilities, which I required for the completion of this work.

I express my thanks to my family for their love, support and encouragement without which I could not complete this dissertation. I would thank to all my friends who helped me in all possible ways towards the completion of my work.

Finally I thank the Almighty who gave me the strength to complete the work.

Harmeet Kaur

(Harmeet Kaur)

801633004

ABSTRACT

DevOps is an integral part of every software company, including Oracle Retail. The Environment Creation Team here has major responsibilities of providing the infrastructure for the Development, Quality Assurance and Production teams for development, testing and further deployment of their Enterprise Applications. The infrastructure is provided in the form of Environments. For each Environment certain number of servers need to be build which commonly include setting up the DB (Database) Server, SFTP (SSH File transfer protocol) Server, IDM (Identity Management) Server, APP (Application Server) Server.

Environment Creation Services (ECS) Team while building each server takes care of setting up of Operating System with pre requisites, Installation and configuration of Oracle Fusion Middleware Products, Implementation Single Sign On configuration for Oracle Retail Products, Installing of Retail Merchandising Applications.

This work contribution to Environment Creation Team by designing the Single Sign On for Environments and doing qualitative analysis of manual Build as well as automated builds, verifying and validating advantages of automation over the manual build. The work done is for the internal purposes of the Environment Creation Team, subject to the Oracle Copyright.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

Chapter 1

Introduction

Before DevOps, Application Development teams were responsible for gathering the business requirements for the software and writing code. Followed by separate Quality Assurance team which were in charge of testing of the developed code in separate QA Environments and if the code meets the requirements it was handed over to Operations Team for the deployment. Operations teams are further divided in to various fragments like Database Teams, App teams, Build Teams.

But many problem prevails when teams works separately. When the code is passed on from one team to another it adds bottlenecks. Same code might work for Dev and QA but not work for Operations team. This may be due to many factors like the choice of system, opposing goals, less knowledge of the other team's roadblocks, etc. This ultimately led to inefficiency and finger pointing among the teams.

So to address these challenges DevOps came into picture by establishing collaborative teams which share responsibility for maintaining the system that runs the software and preparing the software to run on that system with increased quality feedback and automation issues. DevOps is formed by combining two words Development and Operations. Main idea behind DevOps is that deployments should be treated as part of a development workflow, not as an afterthought. Details of the environment, the server IP address, the database version, the environment variables to be set, etc. must all be noted and verified accordingly. As a result of which "Automation" comes into picture.

1.1 Environment Creation Services

At Oracle the environment creation services is a part of DevOps team whose task is to create and deliver infrastructure in the form of Environments. An environment is a group of servers where each server has a specific purpose and set of software installed on it. Environment are delivered to Development, Quality Assurance, and Production teams for their Applications to be installed and tested.

For each type of environment creation, we need to set up the certain number for servers.

For example for the Merchandising Environment we need to set up servers namely DB (Database) Server, SFTP (SSH File transfer protocol) Server, IDM (Identity Management) Server, APP (Application Server) Server.

1.2 Basic Roles of each Server

- DB Server – Installation of database server software. Creation of container and pluggable database (multitenant concept).
- SFTP Server – The main purpose is to manage files on the server. Manage the directories by moving files to the required place which may also be uploaded by the customer. Deals with the setting up the SFTP Server daemon process to move files for Oracle Retail Products.
- IDM Server – Identity Management Server is set up for Authentication, Authorization of the Users and the Roles which can log in to Applications in the environment via Single Sign-On (SSO).
- APP Server – Actual Merchandising application is installed here.

1.3 Manual build of environment

Manual build of environment consist of setting of the required number of servers. Some common tasks involved in building of a server includes Creating users, Installing and Securing Java, SSL Certificates, Installing WebLogic Server, Fusion Middleware installation, RCU Schema Creation, Domain Installation and Configuration, Creating Monitoring users, Securing Domains & Securing Cookies, Deploy the Custom SOA Composite, Apply CPU Patches.

1.4 Automation in DevOps

“Manual Vs Automatic" has been a crucial and challenging topic since couple of years in IT Industry. Human intervention cannot be completely eliminated from computer-based systems, but automation can help during various phases of system development in order to reduce problems in the finished product. Software automation is a process of writing a computer program that intend to automate engineering tasks and operations in a software process using well-defined strategies and systematic

solution that would otherwise need to be done manually. Once environment building tasks have been automated, they can be run quickly and repeatedly in cost effective way.

1.5 WebLogic Server

WebLogic Server is Java Enterprise Edition Application Server hosting Enterprise Applications. Oracle WebLogic Server is the application server for developing and deploying applications across conventional and cloud environments. It provides a mission critical cloud platform for applications requiring performance, scalability and reliability. It handles all the operations including front end, backend and database processes for deploying applications and Fusion Middleware components. It also offers security realms for authentication and authorization.

1.5.1 WebLogic Server Characteristics

- WebLogic server as java code itself, requires java installation as prerequisite for its installation.
- WebLogic server can be simply installed through command line by downloading a WebLogic server jar and typing the following command :
  java -jar <WebLogic jar_ path>
- GUI mode of WebLogic Server installation screens guides through whole installation of WebLogic server. WEBLOGIC_HOME directory and path to JAVA_HOME are the inputs to be given. Rest of the inputs are taken from system.
- Various WebLogic Server packages Utilities like WebLogic JDBC features, Remote Method Invocation (RMI), Java Server Pages (JSP) are installed along with WebLogic Server.
- WebLogic server can be simply launch by using "startweblogic.cmd" and "startweblogic.sh" on Windows and Linux respectively can be used to start the WebLogic domain built on the WebLogic server. Environment variables needed to start the server are set by these scripts.

1.5.2 WebLogic Server Architecture

Following are the WebLogic architecture tiers:

1. Client Tier:
    - It contains programs that are executed by users and web browsers.
    - Standard interfaces are used to take up Web Logic services.
    - Uses HTTP protocol to take content from WebLogic Server.
    - Ensures facilities like clustering and security for its clients.
2. WebLogic Server Tier:
    - Reliability, scalability and high performance are necessary for applications that uses this middle-tier architecture and these parameters are provided by these servers.
    - WebLogic servers can be added to the cluster if load increases.
    - It also ensures failure transparency.
3. Database Tier:
    - This tier basically contains the data needed by clients via WebLogic Server.
    - A desired number of database connections are opened using JDBC connection pool. Hence, now it is one connection per server instead of one connection per application. Same connection is used by all applications on that server.

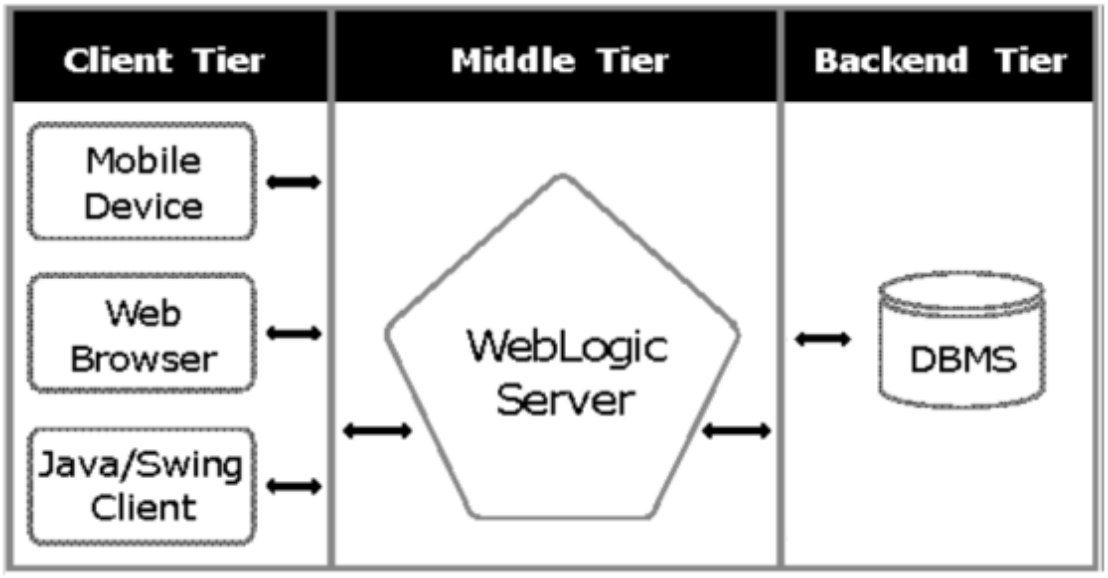

Figure 1.1 Architecture of WebLogic Server

1.6 WebLogic Server Domains

A Domain is a collection of WebLogic Server resources. Applications are installed within the WebLogic server domains.

1.6.1 WebLogic Domain basics

- Once WebLogic server installation gets completed, a domain is created.
- "Domain is defined as a cluster of WebLogic software pieces that are grouped together to make it logically easy to administer and use the services of WebLogic Server".
- "Config.sh" script is used to create a basic Domain. Before running the script certain environment variables have to be defined.
- After running the script on VNC session, some input screen appears for which the following inputs are to be given: Name of domain, path of domain, Template used, data sources, User name and password, Managed Servers, creating machines, system components, ports used.

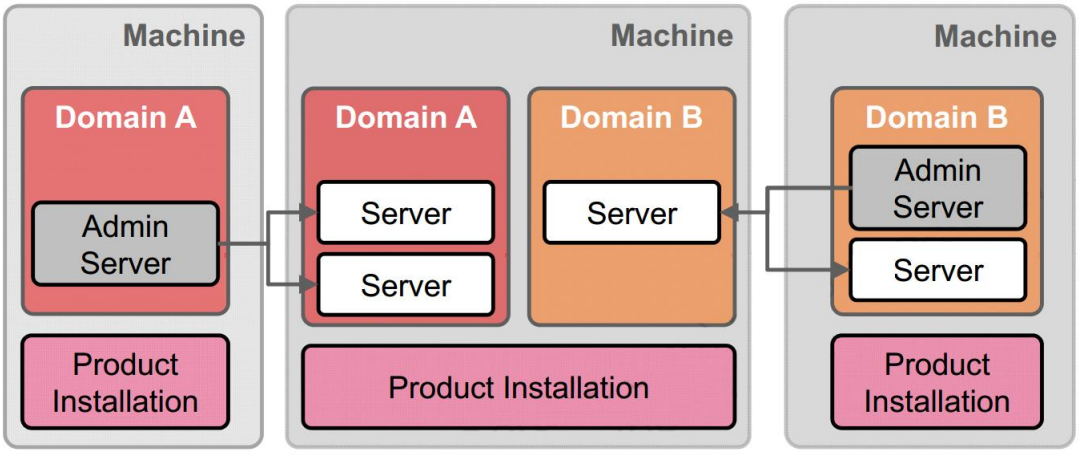

Figure 1.2 Architecture of a basic WebLogic domain

A WebLogic domain normally consists of an "Admin Server" and one or more “Managed servers” where the prime task of an Admin server is to take care of the "managed servers" and set of managed servers together form a “cluster”.

1.6.2 Admin Server and Managed Server

Admin and managed servers communicate as follows:

- An Admin Server is managed by an Admin Console.

- As soon as the domain gets created, the link to the particular admin console appears.
- The user has to enter the login credentials to login into an admin while creating a domain and these details have to match the ones given during manual domain creation.
- Status of all created Managed Servers can be seen in the admin console. Possible status may be idle, running or shutdown.
- Manage servers can be started once admin server is in "running" state.
- Once managed servers comes to "running" state, application becomes alive.
- "startWeblogic.sh" script under domain home directory can be used to bring the Admin server to running state.

1.6.3 Managed Server Components

- An application deployed on a managed server connects to database using “DataSource”.
- JDBC database connections are configured during the creation of a DataSource while creating WebLogic domain or can be done manually later using WebLogic console.

1.6.4 DataSource in WebLogic domain

- A "DataSource" is required if a domain has to connect to a database.
- A Data Source is created during the creation of a basic domain.
- Navigate to "Services -> Data source ->Create a new Generic data source".
- Fields such as "URL of database”, password of database, schema required and "Test the required connection" are to be filled.
- "Save" the data source once connection is successful.

1.7 Oracle Internet Directory

Oracle Internet Directory is an LDAP directory which uses an Oracle Database for storage. Lightweight Directory Access Protocol (LDAP) is used by clients to communicate with a directory server. Below figure shows OID overview.

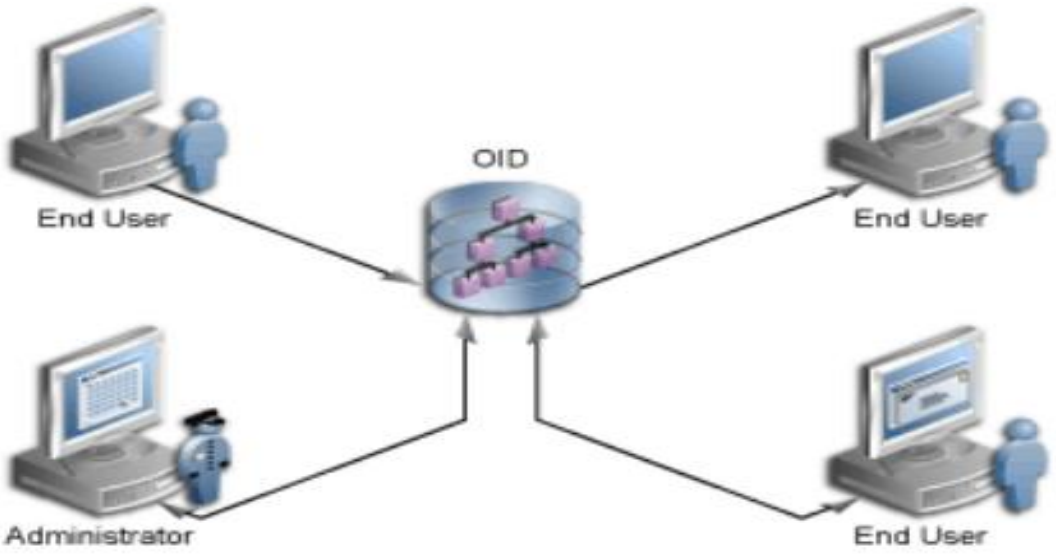

Figure 1.3 OID overview

Specific types of directory data can be managed by administrators and users. Applications that derive data from the directory has this information is immediately available.

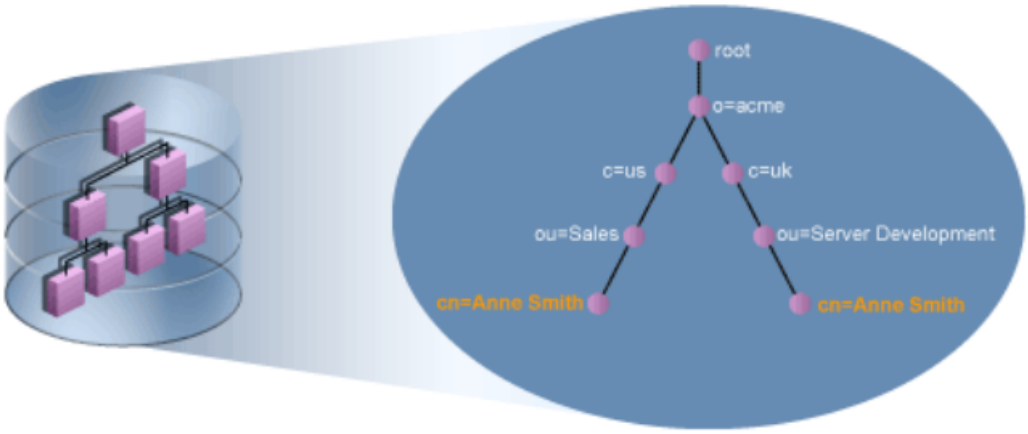

Figure 1.4 LDAP Directory Information Tree

Directory information tree (DIT) is a hierarchy in which entries in an LDAP directory are arranged. The distinguished name tells where exactly the entry resides in the directory hierarchy. The graphic shows a DIT with two users named Anne Smith. The DN for the Anne Smith on the left is:

cn=Anne Smith, ou=Sales, c=us

The DN for the Anne Smith on the right is:

cn=Anne Smith, ou=Server Development, c=uk

In OID we can store Password policies, connection information etc.

Use of ODSM in OID

Configuring the structure of the directory is enabled by Oracle Directory Services Manager. ODSM is the interface used to manage entries, schema, security, adapters, extensions, and other directory features.

1.8 Oracle Data Integration

Oracle Data Integration is a fully unified solution for building, deploying, and managing real-time data-centric architectures in an SOA, BI, and data warehouse environment. Also, it is used to combine all the elements of data integration such as real-time data movement, transformation, synchronization, data management, data quality and data services to ensure accuracy and consistency of information across complex systems.

Repository Creation Utility (RCU) Overview

The existence of schemas in a database is a prerequisite to installation is required in most of the Oracle Fusion Middleware components. Repository Creation Utility (RCU) is used to create these schemas and load into the database. The Repository Creation Utility (RCU) is only available for Intel-based platforms (Windows and Linux) platforms.

1.9 Automation

Automation in any system removes the necessity for direct human control of certain processes. Automation is treating the infrastructure as code by programmatically provisioning and configuring components.

Automating infrastructure is generally done by ‘shell scripting’. Shell scripting has its own disadvantages so Oracle is adopting ‘Chef’ Automation Platform to that transform infrastructure into code.

The Chef code is easier to manage, distribute. Chef code can be run number of times on the nodes to converge to the desired state if any changes are made to the node then

rerunning the chef automation will bring it back to the defined state implementing only changes which are required this property of preserving the state of the node in Chef is called ‘idempotence’. If automation fails it can handle it by rolling back the changes to the previous point.

Chef works on a three-tier client server model. On the workstation users develops and test chef code. Chef server stores cookbooks, the policies that are applied to nodes. Chef client nodes are the machines that are managed by Chef. The Chef client is installed on each node and is used to configure the node to its desired state.

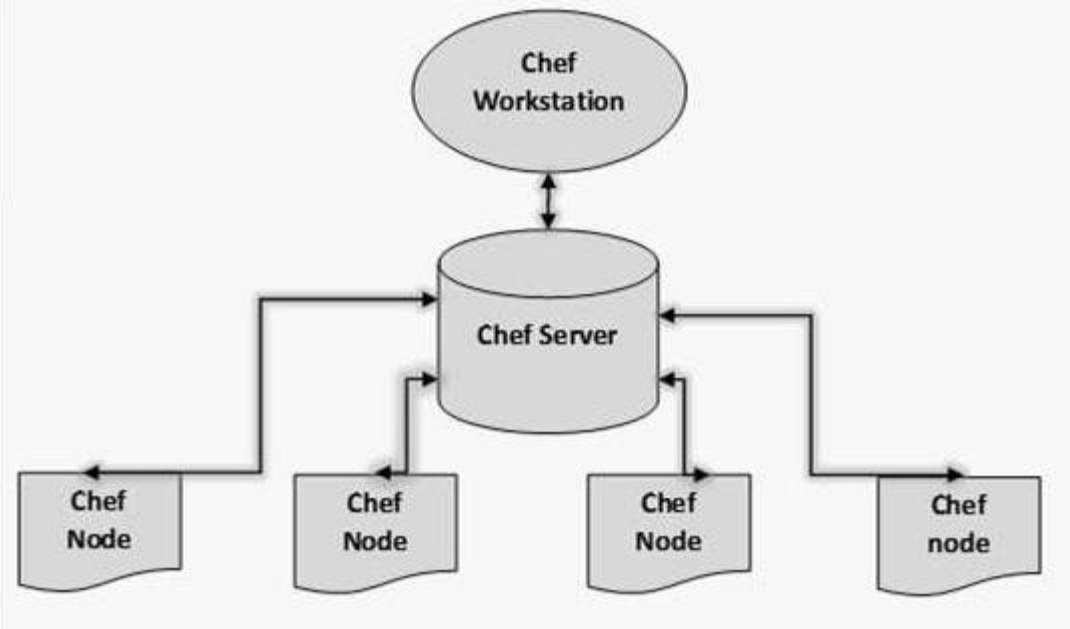

Figure 1.5 three-tier client server model of Chef

Chapter 2

Literature Review

This literature survey focuses on research done in the field of DevOps. Most of the work proves the requirement of the DevOps in the Organizations and importance of DevOps Lifecycle, mainly focusing on providing infrastructure as a code.

2.1 Literature Survey

In this work [1], the author proposes the importance of the DevOps in the Organizations, based on the data collected by various surveys measuring the impact of major factors to practice DevOps including culture followed, automation, sharing information. Paper concluded that all these factors have directly proportional impact on the improvement of software quality. Author also says that to improve software quality companies should follow DevOps Aligned approach. The automation of tasks plays an important role in improving software quality and maintaining standards. The return of investment should be considered before implementing the automation in the company. Also the working culture is very important, the development and operations team should work together on the current problem and should share responsibility.

In this work [2], the author has discussed one of the tactics to speed up the product's development and deployment lifecycle. The tactic discussed here is to provide infrastructure as a code. The TOSCA (Topology and Orchestration Specification for Cloud Applications) is one of the industrial standards supporting Infrastructure as a code. Infrastructure provision includes to provide the required physical or virtual machines on which applications run, to install and configure the required middleware and software on the machines, to instantiate and run the required services for the software to be operated. Standard language and procedures are followed to provide source code as scripts, automation, configuration code, etc. Author has proposed various advantages of infrastructure-as-code some of which are re-usable service templates, consistency, cost efficiency, etc.

In this work [3], the author has understood PaaS in cloud computing to be closer to the DevOps. Cloud computing spans every area of computer science so need full stack

development which deals with the configuration, deployment and performance of Hardware, OS, Container, Database, TCP/IP, firewall, etc. which provide the functions like coding/testing, VM/Container monitoring, resource monitoring, concurrency, scheduling, concurrency, logging, etc. Author has described the features of state-of-the-art solutions available in the market. No PaaS solution fulfill all the requirements, if we increase usability and reliability then the performance decreases, VMs and contains cannot act as real machines, containers are more efficient in resource utilization, VMs have better compatibility and isolation, startup companies prefer to go for Public clouds, big well established organization prefer to go for private clouds.

In this work [4], author proposes that earlier the applications used to be monolithic but with the advent of internet facing applications which require relational databases, servers, load balancing, disaster recovery using various third party libraries and online services, we should adopt the configuration-management-tool to set up and configure the IT infrastructure. Configuration management tools like Chef and Puppet allow to control and automate infrastructure nodes by managing hosts, database, users, services, configurations, storage, security, etc. Using configuration management tools we can write rules and policies according to which the node need to converge. The code for rules and steps are to be written in such a way that the dependencies are taken care of by keeping in view the pre conditions and post conditions are satisfied. The code should be written in the form of modules so that it can be managed easily and it also hides the irrelevant information. The system can be put in the version control system to collaborate, upgrade and manage code. This way of managing the infrastructure is called providing “Infrastructure as Code”.

In this work [5] author has taken the example of an Australian company called wotif group which was able to reduce the time for their software release from weeks to hours by defining the deployment standards for development and operations teams. They were facing the problem due to monolithic code and number of micro services been added to the architecture. To address this issue a team was formed for collaboration among the member from development, operations, testing, architecture, and management. Discussions and meeting were encouraged to share problems, propose solutions, fresh out ideas, discussing preferences and requirements. They started using Atlassian’s Confluence software where they started collecting,

documenting and sharing ideas which formed lightweight application deployment standards. Various targets including log file format and location, initialization script location and invocation, configuration file location, packaging conventions, and SSL configuration and certificate packaging, etc. were discovered and common standards were decided and followed. Later on the automated scripts were developed to verify the standards compliance to verify the system and applications look as expected before and after deployment. The deployment process was also automated. Some rules were followed to keep the process simple like changes should be independent, prevent manual testing, application should comply with the latest standards, etc.

In this work [6], author has presented the analysis of the design and implementation of the Enterprise Application Server. It has mentioned three tier architecture for the application: the web interface tier, the logical tier, and the database tier. The important characteristics include the repeatability, scalability, comparability, etc. For database tier Oracle Database is used, for Middle Logical Tier Oracle WebLogic Server 10g is used, and web interface comprise of the Actual application interface with user can be implemented in any technology like forms and ADF.

In this work [7], author has proposed that for implementing DevOps only the cultural and organizational change are not enough, there is need of end-to-end automation of the processes. There are many heterogeneous automation approaches available in the community which make hard to integrate it with cloud. TOSCA (Topology and Orchestration Specification for Cloud Applications) standard should be followed to enable seamless orchestration of artifacts in the cloud environment. For automation reusable artifacts like scripts, modules, and templates are publicly available to be used. Several aspects to be considered while choosing artifacts are level of dependencies, virtualization, infrastructure or application dependency, etc. Chef is a configuration management framework which provides a domain-specific language based on Ruby. It used to define configurations of resources such as VMs. Recipes contains configuration definitions for nodes. Multiple recipes are bundled in cookbooks.

In this work [8], author describe the features of WebLogic server and said that it is best for deploying the mission critical applications. Transactions as processed reliably and with high performance because it leverage the features of new Oracle Database

features. Advance garbage collection has reduces pause time and has lower latency. It has all new development feature available with java. It is available as a lightweight zip installer. Performance can be easily monitored. Its web tier helps in easy integration with the web services.

In this work [9], author describes the Single Sign On architecture which use cookie based method for managing users and passwords. It supports the access to the legacy application via middle tier. It is simple to configure and administer. Policies can be configured for authentication, authorization and auditing. Cookie is sent to the client as part of HTTP response header which include parameters like name, value, expiry, URL, domain, etc. The author has also mentioned the mechanism that how the centralized login server approach can be used to authenticate the user and describe the flow as how the cookies are exchanged on that. Single Sign On can be chosen as the authentication approach due to its centralized management which can be easily deployed and good performance.

2.2 Gaps in Literature

Based on the Literature Review it has been identified that the Traditional Way of working in which the Development and Operations Teams used to be separate the release and deployment of the code used to be slow process due to inconsistency of the process followed and infrastructure used by each teams.

For a user to use various products of the company separate logins were required for each application which was a major inconvenience, so Oracle has come up with the solution by designing the Single Sign On Environment for Authentication and Authorization in which the user need not to go through authentication separately for each application.

Manual Environment setup is time consuming and error prone task so many tasks are automated. Shell scripts are used which are tedious to write and manage. Moreover if automation fails the changes made till then cannot be revert back and clean up of the server need to be done. Chef is the new technology to provide Infrastructure as a code and has advantage over the drawbacks of shell scripting so Oracle is shifting to Chef for the environment automation.

Chapter 3

Problem Statement

3.1 Problem Definition

IT Industry has developed a huge number of products, mainly in Retail Stores to perform different manual tasks such as ordering store items etc. After the development and testing phase of these products to a consistent level comes the task of deployment of the products on the customer's site. Product deployment and implementation is an extensive part which is performed in order to expand the business in an effective manner. For deployment of the product the infrastructure in the form of Environment need to be provided to the Development, Quality Assurance and Production teams. These Days due to following of agile practices and implementation of DevOps the time to delivery of the infrastructure need to be reduced and at the same time the quality should not be compromised.

The system of Signing in to various applications is inefficient as separate logins are required for each of them. The design of the environment should be such that user should be able to login through a single portal which is valid for all the application.

The present manual system is time consuming, tedious, error prone, inefficient at resource utilization and costly. So to address this problem need of shift from manual to automated system is necessary. Automated systems are fast, standardized, error free, cost effective, anyone can use it and is efficient at resource utilization.

Deployments should be treated as part of a development workflow, not as an afterthought. Details of the staging environment, the server IP address, the database version, the environment variables to be set, etc. must all be noted and verified accordingly. As a result of which "Automation" comes into picture.

While there is no doubt that human intervention cannot be completely eliminated from computer-based systems, automation can help during various phases of environment Creation in order to reduce problems in the finished Environment.

Environment creation tasks are performed so as to maximize profits, expand the business, manage, consolidate, and use all the existing resources in proactive fashion. The aim of this project is to conduct a qualitative product deployments and implementing them on specific hardware and architectural platforms, by following the respective install guide and provide feedback to make the entire procedure easy and error-free, even for those who are not familiar with the technologies. Automation tool helps to increase the productivity, allowing the employees to focus on other important business operations. Developing an automation tool involves the writing of scripts, basically involves the automation of a manual process. Automation should be adopted as a prime strategy where human beings are likely to make more mistakes as compared to machines.

3.2 Objective

The main objectives of this work are listed below:

- To design Single Sign On Environment for application development followed by deployment.
- To perform qualitative analysis of manual builds as well as automated builds.
- To verify and validate advantages rendered by automated environment over manual environment.

Chapter 4

Design & Qualitative Assessment

4.1 Design of Oracle Single Sign-On (SSO)

Oracle Single Sign-On is architecture designed for providing authentication and authorization to the enterprise applications. It uses cookie based method for managing logged in user. Policies can be configured for authentication, authorization and auditing. Cookie is sent to the client as part of HTTP response header which includes parameters like name, value, expiry, URL, domain, etc.

In SSO we use many Oracle Middleware products which are configured and set up to interact with each other for this mechanism to work. The Middleware Products used are:

1. OHS – Oracle HTTP Server is webserver component for Middleware which provide a listener for WebLogic Server to host web applications online. HTTP listener, to handle incoming requests and route them to the appropriate processing utility [10].
2. OAM – Oracle Access Manager is Identity Management solution. It consists of the Access System, and the Identity System. The Access System secures applications by providing centralized authentication, authorization and auditing to enable single sign-on. The Identity System manages information about individuals, groups and organizations. [11]
3. OID – Oracle Internet Directory is an online directory is a specialized database that stores and retrieves collections of information about objects. The information can represent any resources that require management, for example Employee names, titles, and security credentials [12]. The information in the directory is available to single sign-on solution client. Clients communicate with a directory server using Lightweight Directory Access Protocol (LDAP). Oracle Internet Directory is an LDAP directory that uses an Oracle Database for storage.

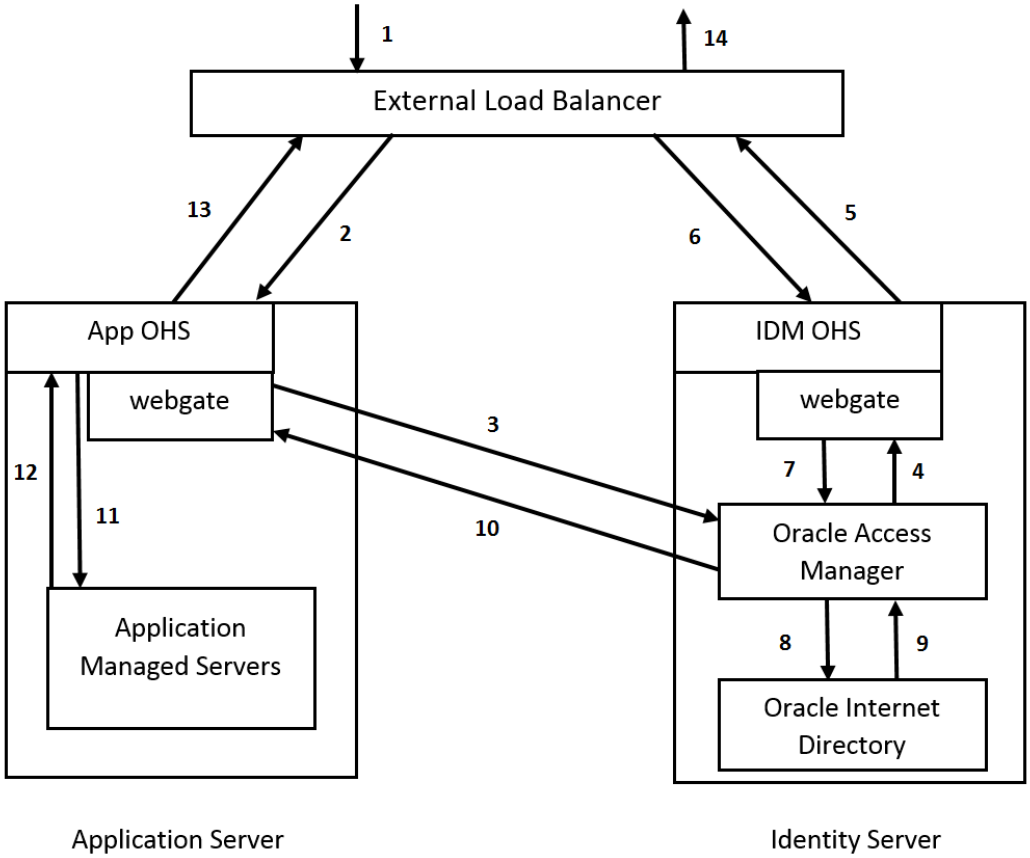

Figure 4.1 Single Sign-On flow

The flow of Steps in new login for SSO is described referring the numbering in the above diagram:

1. User request URL at External Load Balancer (ELB).
2. External Load Balancer redirects the request to Application Host Oracle HTTP Server (OHS).
3. Application Host OHS checks app cookie. Since it is new login it is not present. The request is redirected to OAM, configured in web gate.
4. OAM prompts user for username/password through SSO Login Page via Identity Management Host OHS.
5. Identity OHS display the user the Login Page to enter username and password.
6. Username and password from user is sent to OAM via OHS.
7. OHS forward username and password to OAM.
8. OAM checks the user's validity in OID.
9. Once the user is valid, OID provides OAM with the information about User ID, Groups, etc.

10. OAM then reconstructs a HTTP cookie with user authentication information and send the cookie to Application Host OHS.
11. Application OHS redirects the request the request along with cookie to managed server on which application is running.
12. Application authorizes the user and provides the response to Application Host OHS.
13. Finally, Application Host OHS redirects the application to external lode balancer and then to user along with the cookie.

4.2 Qualitative Analysis of Environment Build

Build of any environment is carried out by the resources in a step by step manner. For setting up of environment we have to set certain number of servers which all work together to provide environment for an enterprise application to run. Primarily the Database server, SFTP Server, Identity Management server and Application server are set up to build an environment. In this work flow of common tasks for setting up each server by analyzing the manual and automated build is described. Some portion of the manual installation related to WebLogic server installation, domain creation for Application installation is explained.

4.3 Tech stack

The most initial step is to find out the components of the environment. Any manual build starts from the tech stack. Tech stack is the list of components to be included in a particular environment.

Following is the Tech Stack used in this work:

- Oracle Enterprise Linux (OEL 6)
- Oracle DB 12.1.0.2
- Oracle Client 12.1.0.2
- Oracle JDK 1.8
- WebLogic Server12.2.1.3
- Oracle Repository Creation Utility (RCU) 11.1.1.9
- Oracle HTTP Server 12.2.1.3 for Application server
- Oracle HTTP Server 11.1.1.9 for Identity Management Server
- Oracle Access Manager (OAM) 11.1.2.3

- Oracle Internet Directory(OID) 11.1.1.9
- Desktop web browser (Chrome, Mozilla Firefox, Internet Explorer)

4.4 Manual Build

Following are the logic and steps followed for manually building the environment:

- Pre Installation Tasks - This includes tasks like checking hardware and software requirements and installing the components required.
- Database Installation - This includes database schema creation for product which will be used during deployment.
- If successful then install Oracle Fusion Middleware - This includes installation of WebLogic base.
- Create Domain if not already present - This includes creation of domain related to that particular product.
- OID Installation - This installs oracle internet directory.
- OBIEE Installation - This includes oracle business intelligence enterprise edition installation.
- RCU Installation - Repository creation utility is installed and MDS schemas are created.
- Application Installation - Application is installed.
- Launch Application URL - Application launching is tested.
- If successful then Create users and groups - The users used for logging into application are created.
- Login into the application - Application login is tested

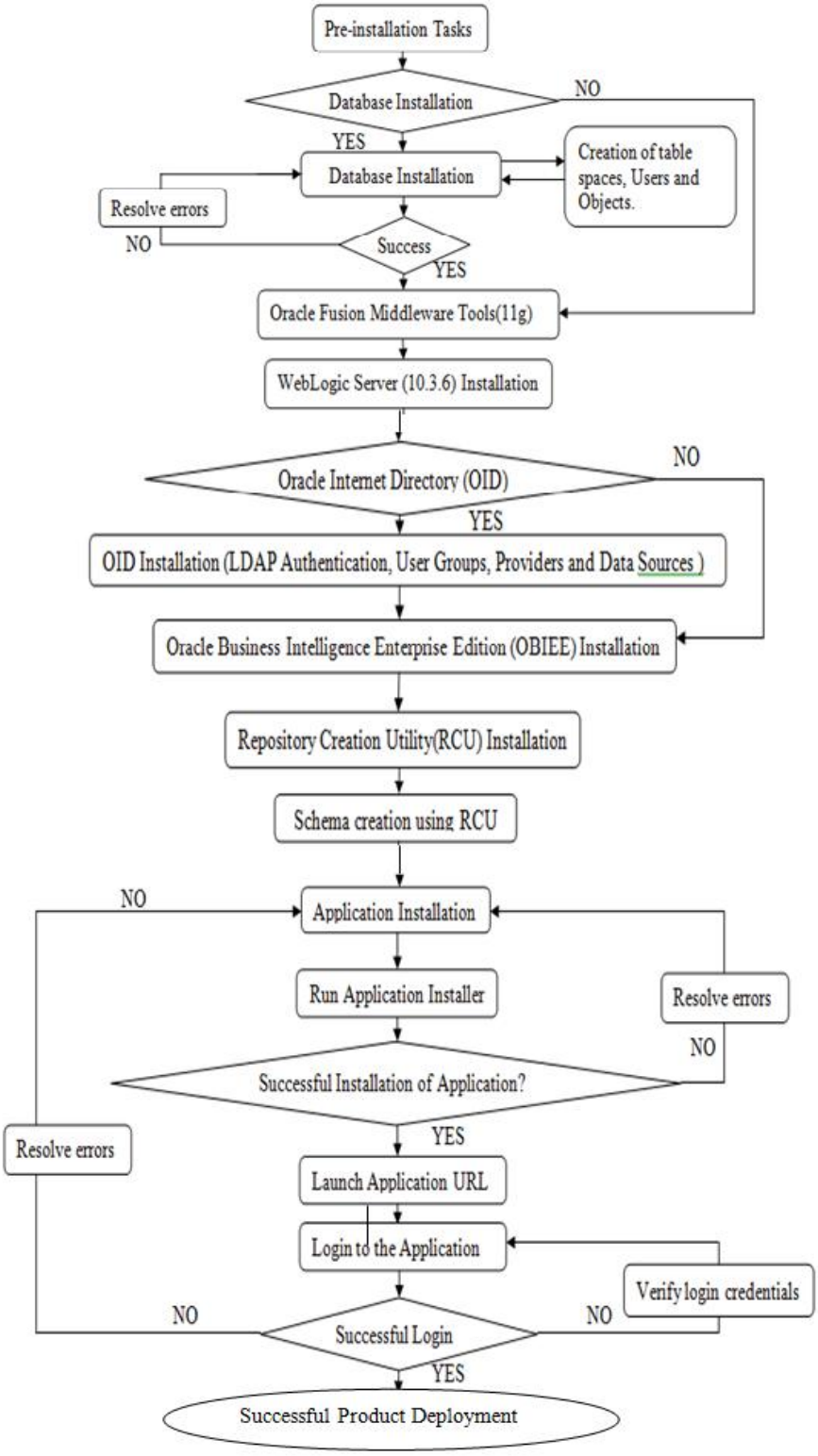

Figure 4.2 Flow diagram for logics of application installation

4.5 Automated Build

In this work the high level view of automation is analyzed and described that how each module is called to set up the complete environment. The proper standards are followed while writing the automation code for the infrastructure.

High Level Modules

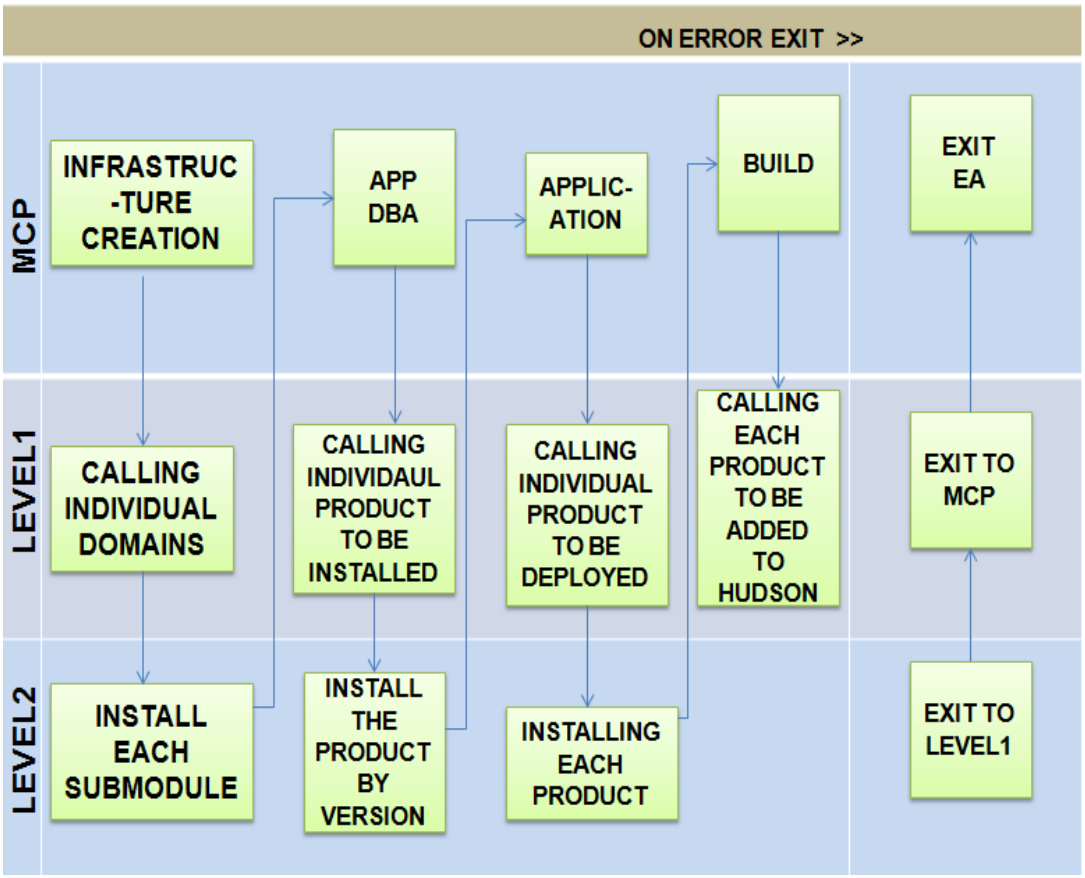

Figure 4.3 Diagram for Environment Automation high level modules

Description

MCP: MCP stands for master control program, which is the driver for calling all the other modules in the Environment automation. The Master Control program reads the response file, and gets the Environment Variable values. Based on the values the master control program calls OS master program which sets up the servers (OS Module), calls DB install and create programs to Install oracle database binaries and creates Databases (DB Module), calls APP dba master program that creates schemas and installs schema objects in Database and configures the apps (APP DBA Module),

calls Application master program that installs WebLogic, reports etc. and finally calls the Build module. If any of the modules fail the MCP will stop and exit.

Instructions for running the Automation

A master wrapper script is executed which takes certain number of inputs to set up variables and calling the appropriate chef command to run the specific module.

The attributes passed to the wrapper script while running the wrapper script include modules to be called, branch to take code from, environment type, environment file to be used. In addition to these attributes various flags can also be set to tell whether to provide database host information, identity host information, to encrypt credentials, whether the environment is clustered, etc.

After getting this information wrapper script calls the appropriate chef solo command which in turn asks for final inputs which include App host internal load balancer, external load balancer, database host name, identity host name, environment build name, seed value, environment type, system credentials, username and passwords, etc. Flags are also set for various options.

The above inputs are updated by the script in environment file and config files based on which automation proceeds. Credentials used can be found in data-bags folder and the log files generated can be found in respective log folder.

4.6 Software, Tools, and Technologies Used

- Windows 7,8
- Oracle Enterprise Linux
- Putty
- MobaXterm
- Remote Desktop
- Web Browser
- VNC viewer
- Oracle Java Runtime Environment
- Oracle fusion middleware
- Oracle WebLogic Server
- sqlplus

- SQL Developer
- Chef
- Shell Scripting
- Confluence
- Jira

Chapter 5

Results & Discussions

5.1 WebLogic Installation

This section presents the screen shots for comparative study between manual deployment and automated build of an environment. Follow by the statistics comparing the time consumption by the manual build and time consumed by the automation scripts of the same tasks.

5.1.1 Manual Installation

Manual installation of WebLogic takes place through a series of steps and installer screens when it is done manually. Steps along with the starting screen and ending screens of installer containing the summary of the installation are presented.

- Java installation is prerequisite for the WebLogic to be installed in JAVA_HOME. Path of JAVA_HOME is exported and added to the system $PATH
- For accessing graphical user interface DISPLAY variable is exported.
- The jar file for the WebLogic installer is placed and is run to launch the user interface screens. Different inputs are fed into the successive screens as per requirement. Then finally on clicking install, the WebLogic base is created on that machine.

Below are the screen shots for comparative study between manual deployment and automated build of an environment.

The successive screens look like the ones described below:

```
msp32302:[] /u00/webadmin/media> export DISPLAY=10.141.30.200:0.0
msp32302:[] /u00/webadmin/media> export JAVA_HOME=/u00/webadmin/products/jdk_java/
msp32302:[] /u00/webadmin/media> export PATH=$JAVA_HOME/bin:$PATH
msp32302:[] /u00/webadmin/media> java -jar fmw_12.1.3.0.0_infrastructure.jar
```

Figure 5.1 Starting WebLogic installation using command prompt manually

Figure 5.2 Welcome screen while installing WebLogic manually

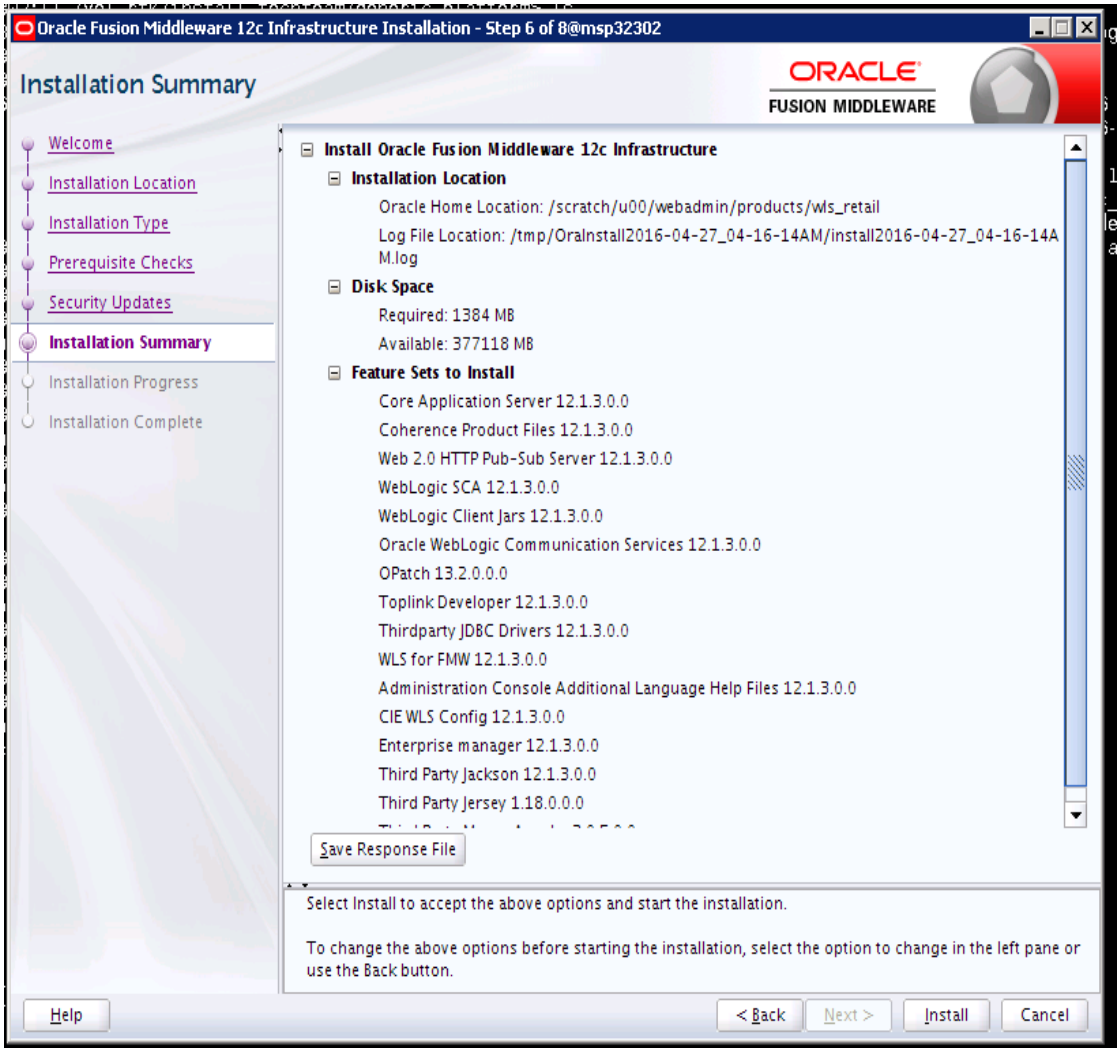

Figure 5.3 Installation summary for WebLogic in manual installation

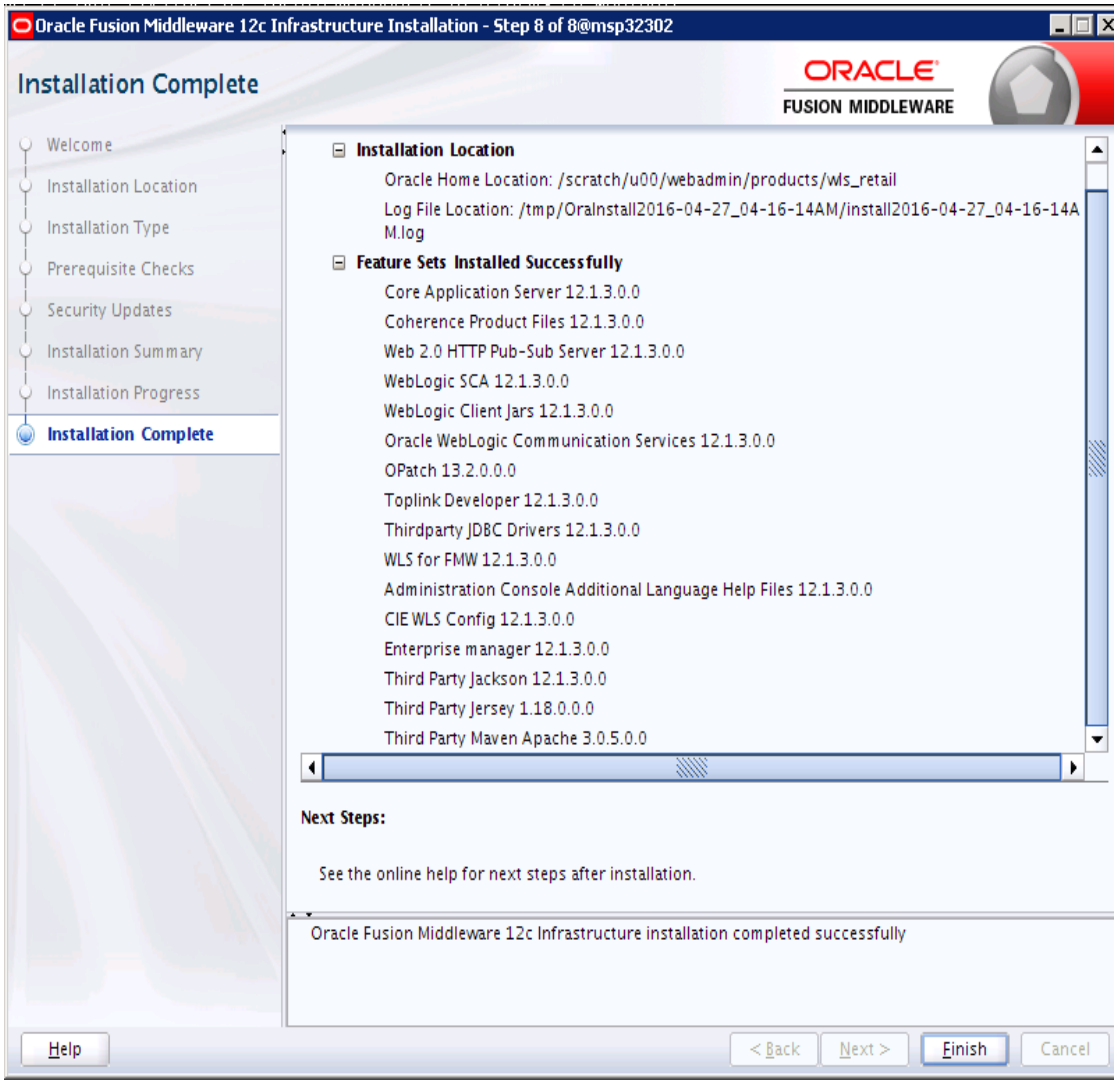

Figure 5.4 Final screen showing the completion of WebLogic manual installation

```
msp32302:[] /u00/webadmin/products> ls -ltr
total 16
drwxrwxrwx  8 webadmin dba 4096 Feb  9 02:02 jdk_java
drwxrwxrwx 11 webadmin dba 4096 Feb  9 02:17 wls_forms
drwxrwxrwx 11 webadmin dba 4096 Apr 27 01:44 wls_retail_bak
drwxr-x--- 10 webadmin dba 4096 Apr 27 04:24 wls_retail
```

Figure 5.5 wls_retail is created as WebLogic home after manual WebLogic installation

5.1.2 Automation

Screen shots for automated install of WebLogic:

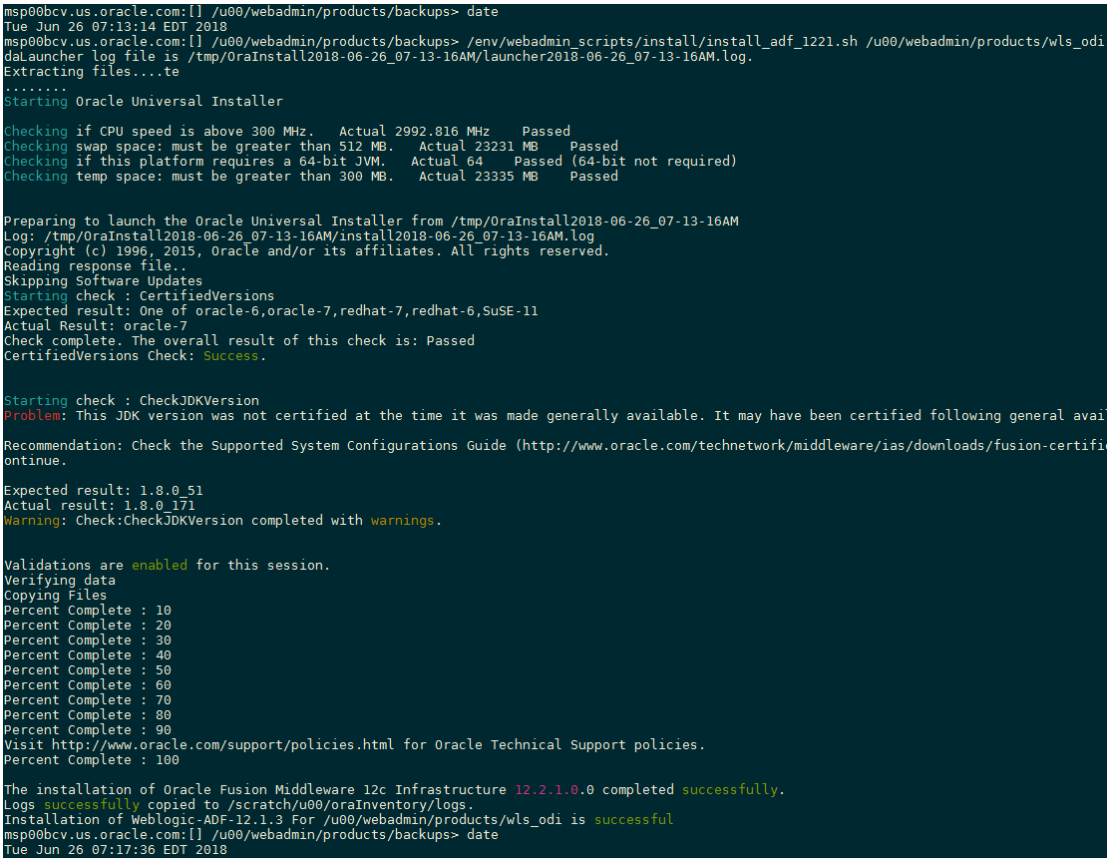

Figure 5.6 Starting time and finishing time of automated WebLogic installation

5.2 Repository creation utility

5.2.1 Manual creation

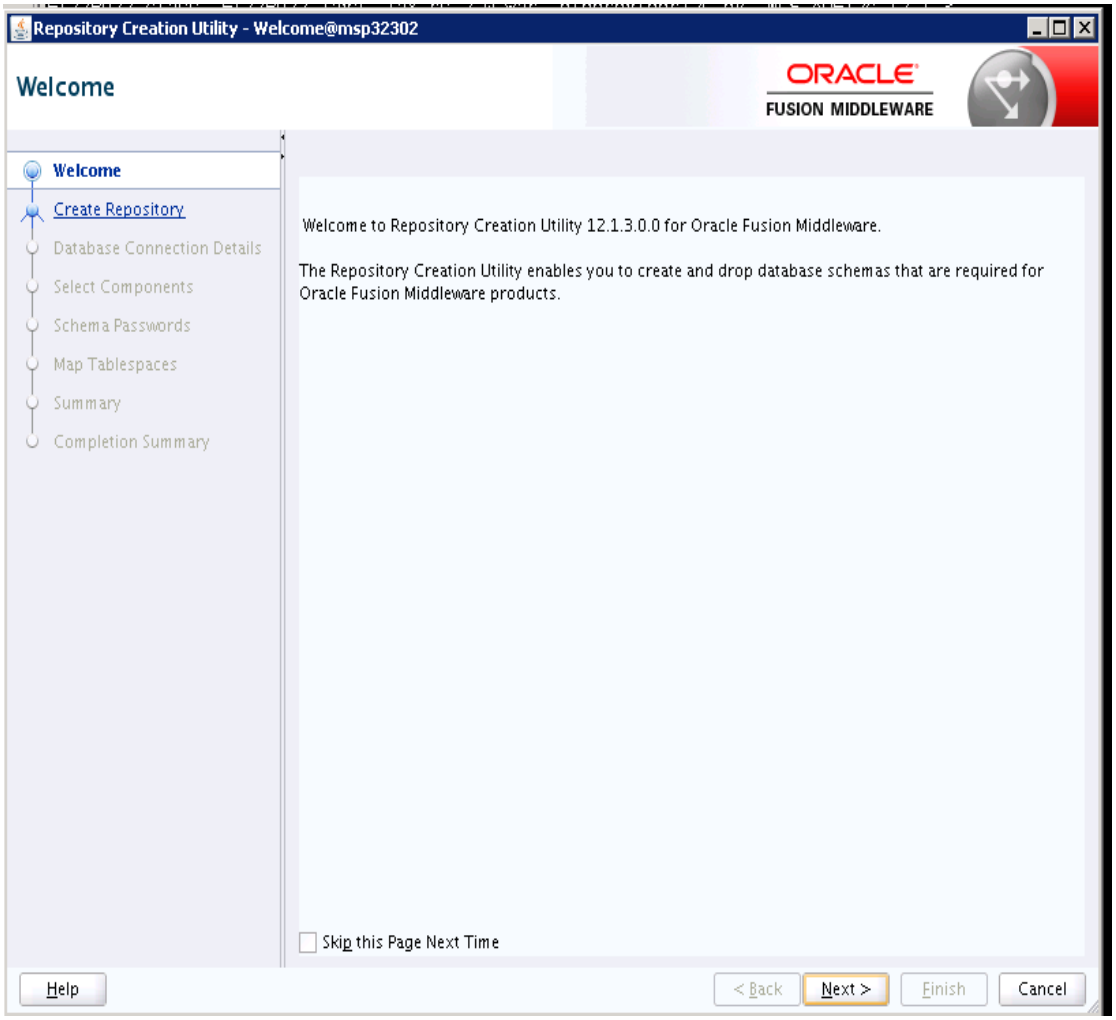

Figure 5.7 Welcome GUI screen for RCU creation when done manually

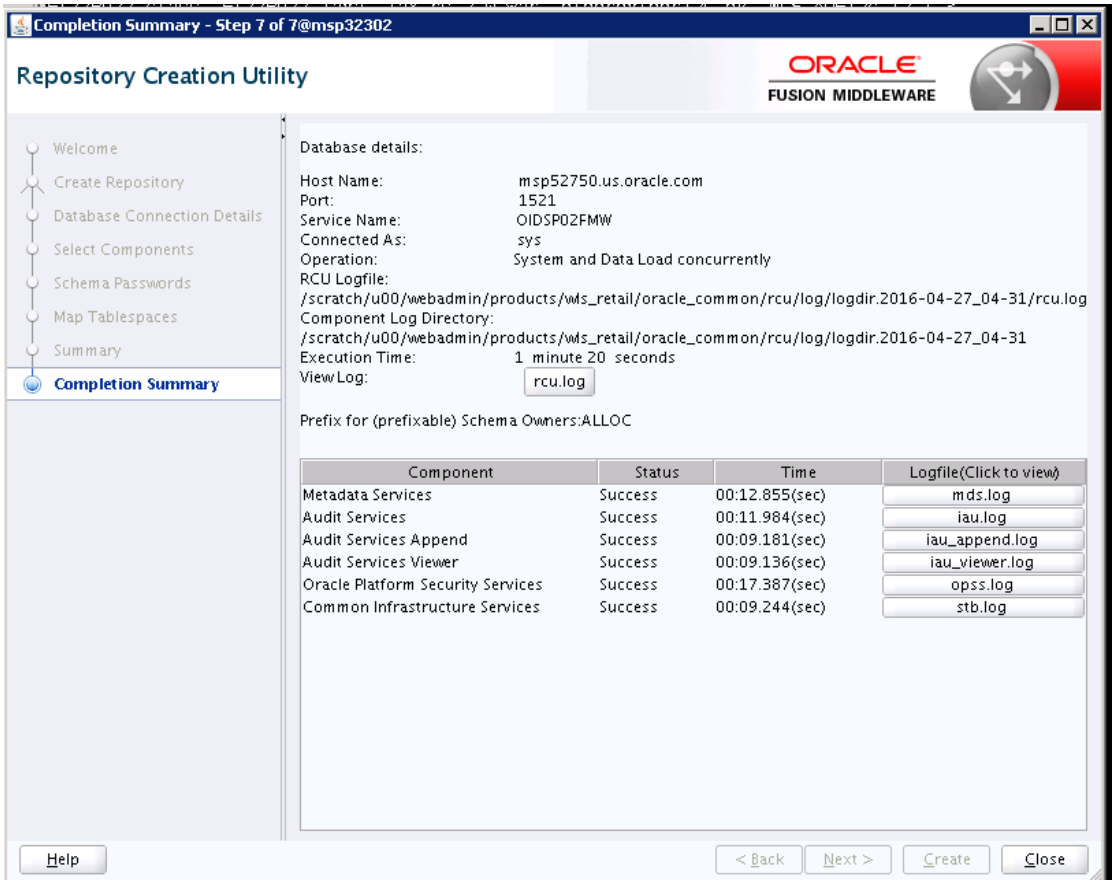

Figure 5.8 Final finishing screen during manual creation of RCU

5.3 Domain creation and configuration

5.3.1 Manual Creation

A Domain is a collection of WebLogic Server resources. Actual Applications are installed within the WebLogic server domains.

For accessing graphical user interface DISPLAY variable is exported.

For the Domain creation $WEBLOGIC_HOME/oracle_common/bin/config.sh script is invoked to call GUI. Various inputs like Name of domain, path of domain, Template used, data sources, User name and password, Managed Servers, creating machines, system components, ports used are given in the successive screen of GUI out of which some of the configuration screens along with the starting and the end summary screen are shown here.

Figure 5.9 Welcome screen during manual creation of domain.

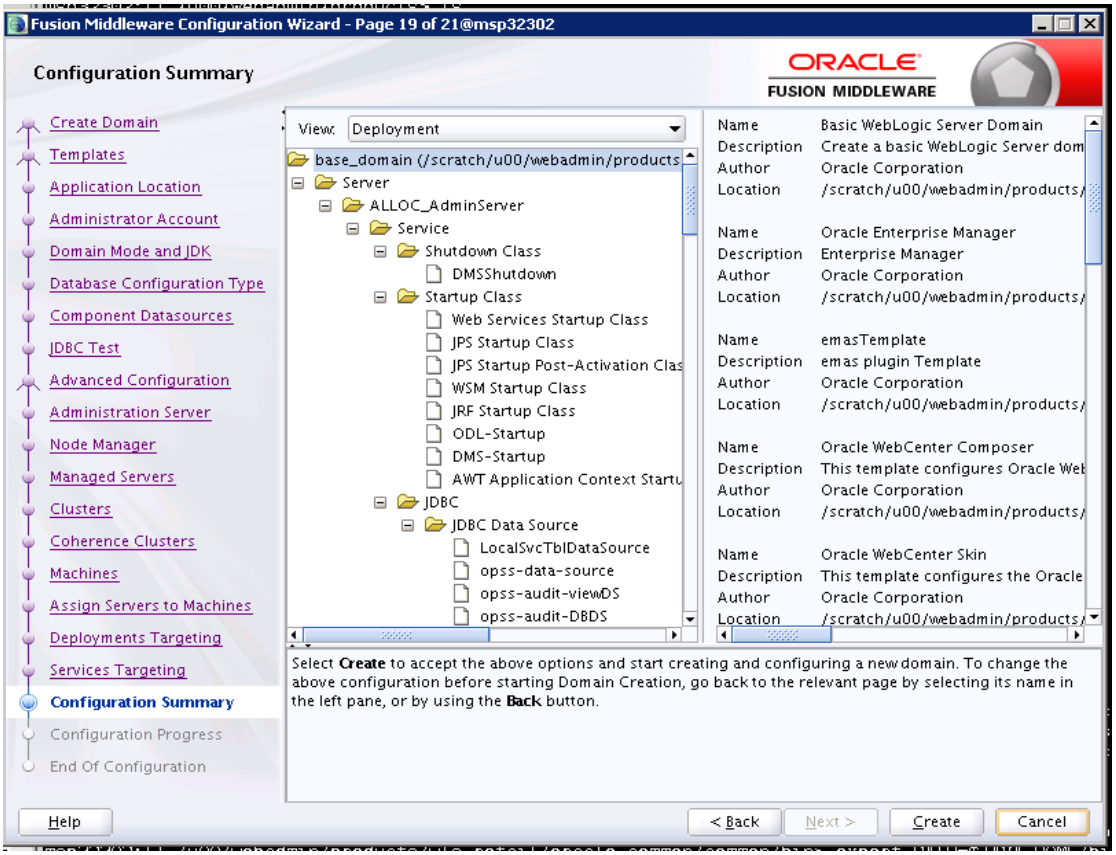

Figure 5.10 Final screen during manual creation of domain.

```
[1] +  Done                    nohup ./startWebLogic.sh &
msp32302:[] /u00/webadmin/config/domains/wls_retail/ALLOCDomain/bin> cd ../
msp32302:[] /u00/webadmin/config/domains/wls_retail/ALLOCDomain> cd ..
msp32302:[] /u00/webadmin/config/domains/wls_retail> cd ALLOCDomain/
msp32302:[] /u00/webadmin/config/domains/wls_retail/ALLOCDomain> ls
autodeploy  config       derbyShutdown.log     generated_classes  logs         securi
bin         console-ext  edit.lok              init-info          nodemanager  serve
common      derby.log    fileRealm.properties  lib                pending      shutd
msp32302:[] /u00/webadmin/config/domains/wls_retail/ALLOCDomain> cd servers/
msp32302:[] /u00/webadmin/config/domains/wls_retail/ALLOCDomain/servers> cd ALLOC_Ad
msp32302:[] /u00/webadmin/config/domains/wls_retail/ALLOCDomain/servers/ALLOC_AdminS
adr  cache  data  logs  security  sysman  tmp
msp32302:[] /u00/webadmin/config/domains/wls_retail/ALLOCDomain/servers/ALLOC_AdminS
msp32302:[] /u00/webadmin/config/domains/wls_retail/ALLOCDomain/servers/ALLOC_AdminS
boot.properties
msp32302:[] /u00/webadmin/config/domains/wls_retail/ALLOCDomain/servers/ALLOC_AdminS
msp32302:[] /u00/webadmin/config/domains/wls_retail/ALLOCDomain/servers/ALLOC_AdminS
msp32302:[] /u00/webadmin/config/domains/wls_retail/ALLOCDomain/servers> cd ..
msp32302:[] /u00/webadmin/config/domains/wls_retail/ALLOCDomain> ./startWebLogic.sh
```

Figure 5.11 Manually starting Domain screenshots

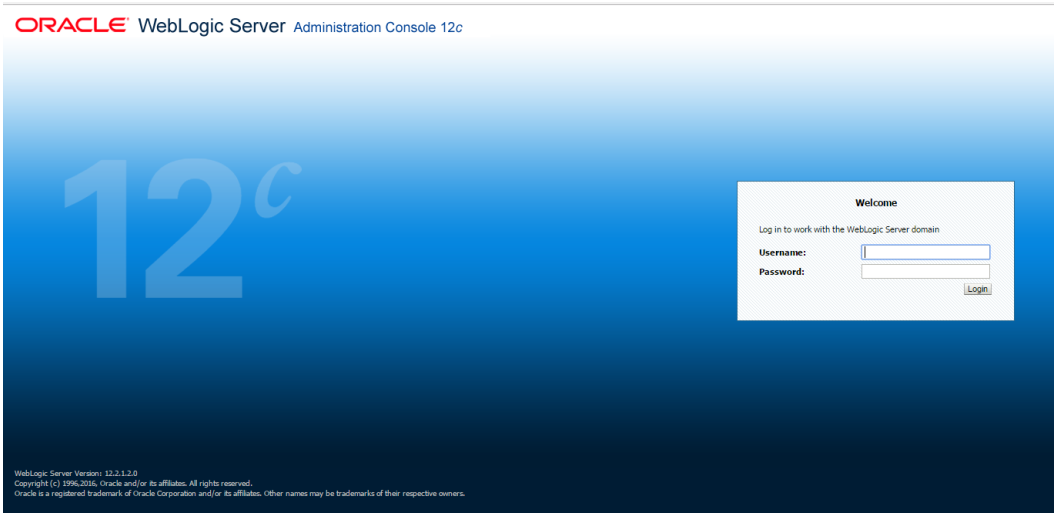

Figure 5.12 After the creation of Domain we can access the WebLogic Admin Console

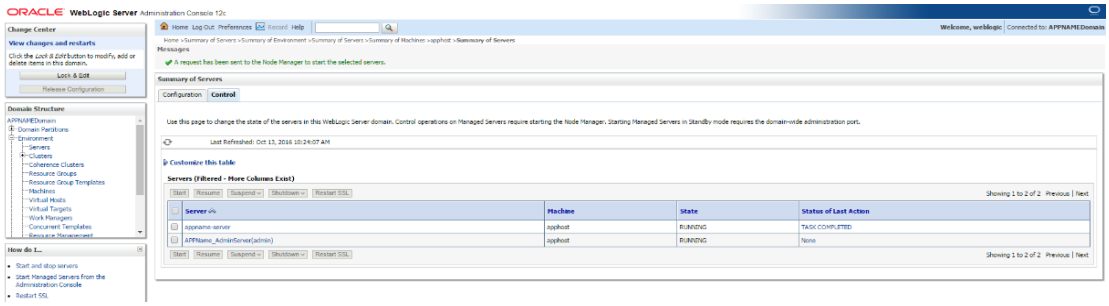

Figure 5.13 After logging in to console we can manually start the managed servers

5.3.2 Automation

```
09:19:06 05/16/18 Installing ALLOCDomain
09:19:06 05/16/18 Dropping RCU Schemas for ALLOCDomain if already present
                  Dropping RCU Schemas for ALLOCDomain: File:/vol.rtk/automation/FY12/ea_reorg/apps_infra/16.0.2/log/RCUDROP.2018.05.16_09.19.06.log
09:19:12 05/16/18 Creating RCU Schemas for ALLOCDomain
                  Creating RCU Schemas for ALLOCDomain: /vol.rtk/automation/FY12/ea_reorg/apps_infra/16.0.2/log/RCUCREAT.2018.05.16_09.19.12.log
                  RCU Schema created sucessfully for ALLOCDomain
                  Deleting Domain Directories for ALLOCDomain if it already exists
09:19:45 05/16/18 ALLOCDomain Creation is in progress Please Check /vol.rtk/automation/FY12/ea_reorg/apps_infra/16.0.2/log/16May2018_0919.log for logs
                  ALLOCDomain Created Check /vol.rtk/automation/FY12/ea_reorg/apps_infra/16.0.2/log/16May2018_0919.log for logs
                  Extending ALLOCDomain with ADF Template Check /vol.rtk/automation/FY12/ea_reorg/apps_infra/16.0.2/log/16May2018_0919.log for logs
                  ALLOCDomain extended with ADF Template Check /vol.rtk/automation/FY12/ea_reorg/apps_infra/16.0.2/log/16May2018_0919.log for logs
                  Adding alloc-server to ALLOCDomain Check /vol.rtk/automation/FY12/ea_reorg/apps_infra/16.0.2/log/16May2018_0919.log for logs
                  alloc-server added to ALLOCDomain Check /vol.rtk/automation/FY12/ea_reorg/apps_infra/16.0.2/log/16May2018_0919.log for logs
09:21:16 05/16/18 Creating the boot.properties file for the server start ups
                  Directory /u00/webadmin/config/domains/wls_retail/ALLOCDomain/servers/ALLOC_AdminServer/security doesnot Exists
                  Creating the Directory /u00/webadmin/config/domains/wls_retail/ALLOCDomain/servers/ALLOC_AdminServer/security ..
                  Directory /u00/webadmin/config/domains/wls_retail/ALLOCDomain/servers/alloc-server/security doesnot Exists
                  Creating the Directory /u00/webadmin/config/domains/wls_retail/ALLOCDomain/servers/alloc-server/security ..
09:21:17 05/16/18 Starting the Servers of /u00/webadmin/config/domains/wls_retail/ALLOCDomain
                  Starting ALLOC_AdminServer for /u00/webadmin/config/domains/wls_retail/ALLOCDomain:OutFile: /vol.rtk/automation/FY12/ea_reorg/apps_in
fra/16.0.2/log/ALLOC_AdminServer.2018.05.16_09.21.17.out
                  Starting alloc-server for /u00/webadmin/config/domains/wls_retail/ALLOCDomain:OutFile: /vol.rtk/automation/FY12/ea_reorg/apps_infra/1
6.0.2/log/alloc-server.2018.05.16_09.23.17.out
                  Starting Node Manager for /u00/webadmin/config/domains/wls_retail/ALLOCDomain:OutFile: /vol.rtk/automation/FY12/ea_reorg/apps_infra/1
6.0.2/log/NM_STARTUP.2018.05.16_09.24.17.out
09:25:17 05/16/18 /u00/webadmin/config/domains/wls_retail/ALLOCDomain creation completed successfully
```

Figure 5.14 Screenshot Showing the starting and ending time for Domain creation

5.4 Results

The following table represents the statistics related to the time consumption in manual build versus the time consumption after using automation scripts for the same tasks.

Table 5.1 Time consumption in manual build vs after using automation scripts

Description	**Manual Build**	**Automation**
Time consumption for WebLogic Server Installation.	15 min	4 min
Time consumption for the Domain Creation which include repository creation using RCU, domain creation, starting Admin Server and Node Manager.	55 min	6 min

Considering the statistics in above table the time consumption in WebLogic installation reduced to 4 min with automation from 15 min when it was done manually. Similarly, the time for Domain creation task which include the various steps starting from repositories creation, domain creation, starting the WebLogic server and node manager dropped from 55 min to 6 min. Similarly various individual tasks are automated which lead to huge reduction in the overall time consumption to set up the environment.

5.5 Issues with Manual Build of Environment

- Time Consuming - It takes approximately 2 to 3 weeks to completely build an environment.
- More Resources Required - A number of resource personnel are required to build an environment.
- Error Prone - Since it is done manually there are chances of more manual errors.
- Fixing errors is comparatively time taking - This is because building an environment itself takes around 2 weeks and if there is an error at some stage which was missed out then it again takes double amount of time to go back and fix that error and again continue with further steps.

5.6 Advantages of Automation

The major advantages associated with Environment Automation are:

- FAST - As it is an automatic process it will definitely be faster as compared to manual build.
- BETTER RESOURCE UTILIZATION - As the procedure is being automated there will be very less requirement of human intervention. So the resources can continue with some other important company works and thus this will help in better resource utilization.
- QUALITY ASSURANCE - As a standardized process is being used and as it is automatic there will be fewer chances of manual errors and the product will be of better quality.
- SYNERGY BETWEEN COMPONENTS - A standard procedure is there which is used for automation. So it will remain same even if it is being done by different persons.

Chapter 6

Conclusion and Future Scope

6.1 Conclusion

Environment Creation is always a lengthy and time consuming process. All the steps are correctly documented, to ensure that the quality and standards of the Environment.

The work done here has resulted in successful Design of the Single Sign On Environment for managing authentication and authorization for Users accessing the web application.

The qualitative analysis gives the detailed overview of the flow of work for manual build as well as automated build of the environment.

By validating and verifying both the manual and automated build of environment it is concluded that using automation is advantageous over doing manual build. Automation reduces the time to build the environment to great extent. The quality of the environment creation increase as the chances of introduction of human errors is reduced. There is synergy between the components as it is automatic procedure which turns out the same results even if the work is done by different people.

This has help a lot to the organization to reduce workload and utilize the resource potential in other tasks as the time consumption in building an environment has reduced a lot.

6.2 Future Scope

This idea may also serve as a base for building of many other new automation projects. Many post installation tasks are still required and the identification of the tasks which can be automated should be find out and automated so that maximum level of automation can be achieved.

REFERENCES

[1] Perera, P., Silva, R. and Perera, I., 2017, September. Improve software quality through practicing DevOps. In Advances in ICT for Emerging Regions (ICTer), 2017 Seventeenth International Conference on (pp. 1-6). IEEE.

[2] Artac, M., Borovsak, T., Nitto, E.D., Guerriero, M., & Tamburri, D.A. (2017). DevOps: Introducing Infrastructure-as-Code. 2017 IEEE/ACM 39th International Conference on Software Engineering Companion (ICSE-C), 497-498.

[3] Li, Z., Zhang, Y. and Liu, Y., 2017. Towards a full-stack devops environment (platform-as-a-service) for cloud-hosted applications. Tsinghua Science and Technology, 22(01), pp.1-9.

[4] Diomidis, S., Don't Install Software by Hand, Tools of The Trade, Athens University of Economics and Business. [online]. Available:https://www.spinellis.gr/pubs/jrnl/2005-IEEESW-TotT/html/v29n4.pdf [Accessed 11 3 2018]

[5] Callanan, M. and Spillane, A., 2016. DevOps: making it easy to do the right thing. IEEE Software, 33(3), pp.53-59.

[6] Chen, Y., Wu, J., Xie, X. and Li, X., 2013, October. Design and analysis of application benchmark for enterprise server evaluation. In TENCON 2013-2013 IEEE Region 10 Conference (31194) (pp. 1-4). IEEE.

[7] Wettinger, J., Breitenbücher, U. and Leymann, F., 2014, December. Standards-based DevOps automation and integration using TOSCA. In Utility and Cloud Computing (UCC), 2014 IEEE/ACM 7th International Conference on (pp. 59-68). IEEE.

[8] "Oracle WebLogic Server" [online]. Available: http://www.oracle.com/us/products/middleware/application-server/oracle-weblogic-server-ds-1391360.pdf [Accessed 15 3 2018]

[9] Samar, V., 1999. Single sign-on using cookies for Web applications. In Enabling Technologies: Infrastructure for Collaborative Enterprises, 1999.(WET ICE'99) Proceedings. IEEE 8th International Workshops on (pp. 158-163). IEEE.

[10] “Fusion Middleware Administrator's Guide for Oracle HTTP Server” [online]. Available:https://docs.oracle.com/cd/E28280_01/web.1111/e10144/intro_ohs.htm#HSADM101 [Accessed 10 4 2018]

[11] “Oracle Access Manager” [online]. Avaiable:http://www.oracle.com/technetwork/testcontent/access-manager-wp-10gr3-131015.pdf [Accessed 15 4 2018]

[12] “Learn the Basics: Oracle Internet Directory” [online].

Available:https://docs.oracle.com/cd/E15523_01/oid.1111/e10036/basics_02_oid_what.htm [Accessed 15 4 2018]

[13] “Introduction to the Multitenant Architecture” [online]. Available: https://docs.oracle.com/database/121/CNCPT/cdbovrvw.htm#CNCPT89234 [Accessed 18 4 2018]

[14] “Getting Started With WebLogic Server” [online]. Available:https://docs.oracle.com/cd/E14571_01/web.1111/e14529/getting_started.htm#INRMP109 [Accessed 20 4 2018]

[15] “Introduction to Single Sign-On with Access Manager” [online]. Available: https://docs.oracle.com/cd/E27559_01/admin.1112/e27239/sso.htm#AIAAG296 [Accessed 25 4 2018]