ABSTRACT:

This document describes the design, implementation and possible applications of Real Time Linux and the implementation of the cloud in the Linux operating system. The term "cloud" is analyzed in "Internet". Cloud computing is Internet-based computing where shared virtual servers provide software, infrastructure, platform, devices and other resources. In this system, a standard time-sharing operating system and a real-time executive are run on the same computer. It supports real-time operating system functionality through the kernel patch in real time, and would be useful for developers who want to build the Linux Embedded system in real time at a low cost. Many agencies store the most important data in the cloud storage, gradually abandoning expensive servers and backup systems. Moreover, even if the telephone systems are moving to the cloud. Many companies still use powerful computers to process data. But soon the clouds download many computers and provide access to data wherever there is access to the Internet. In recent years, the Linux operating system is widely applied in the integrated field, such as industrial automation, consumer electronics, communications, aviation control, military equipment, defense system, nuclear power plant, smart appliances, etc. For its open source advantages, it is compatible with a variety of hardware platforms, Linux has strongly attracted most developers to study. If Linux open source supports real time, the real-time operating system can be easily realized at low cost. This paper focuses upon the automation of customer application right from environment provisioning to application deployment. In this paper the whole architecture of automated application deployment assembled using Ansible as the orchestration engine.
INTRODUCTION:

Cloud computing is a modality to provide computer facilities over the Internet. Cloud computing provides a three-delivery model: a) SaaS - Software as Service uses software applications in the cloud, b) PaaS - Platform as a service implements the applications created by the user in a cloud, c) IaaS - Infrastructure as a service which is used to provide storage and computing resources for rent. This three-delivery model supports on-demand services, extensive network access, location independence, fast scalability, pay-per-use. With the increasing use of high-speed Internet technologies over the past few years, the concept of cloud computing has become more popular. In cloud computing, users work with web-based storage and software instead of local ones. These applications are accessible through a browser and they look and act like desktop programs. A server in the cloud is a process that runs on a hypervisor. This hypervisor software allows a single computer to look like a dozen or more computers. A single server in the cloud is just one of these processes. The user looks like a server, but it is not his own collection of hardware. Cloud servers can scale up and down with minimal effort, using only software. You can increase or decrease RAM, hard drive space and even CPU power dynamically using a software interface. The cloud server does not require maintenance once the automatic backups are configured. The servers in the cloud are the same as the virtual servers and, no doubt, it is easier to move between the data centers. Red Hat Cluster Suite (RHCS) is an integrated set of software components that can be deployed in a variety of configurations to meet your performance, high availability, load balancing, scalability, file sharing, and economy needs. Cluster infrastructure: provides key functions for the nodes to work together as a cluster: configuration file management, membership management, lock management and fencing. High availability service management: provides service failover from one cluster node to another in case a node stops working. Cluster Management Tools: Configuration and Administration Tools to Configure, Configure, and Manage a Red Hat Cluster. The tools are used with Cluster Infrastructure components, High Availability components and Service Management, and storage. Linux Virtual Server (LVS) - Routing software that provides IP load balancing. LVS works on a pair of redundant servers that distribute client requests evenly to the real servers behind LVS servers. The Red Hat Cluster Suite cluster infrastructure
provides the basic functions for a group of computers (called nodes or members) that work together as a cluster. Once a cluster is formed using the cluster infrastructure, you can use other Red Hat Cluster Suite components to meet cluster needs (for example, configure a cluster to share files on a GFS file system or configure service failover). In this paper, we suggest a holistic approach to cloud computing that transcends the boundaries of individual machines, the cluster infrastructure that provides the basic functions for a group of computers to work together as a method to produce patches and clusters for Real kernel patches. At one time can build Linux environment integrated into Linux at low cost. Our goal is to provide a uniform abstraction of the cloud operating system that adheres to the conventions of consolidated operating systems. Ansible is a very powerful open source automation language. What makes it unique from other management tools is that it is also a distribution and orchestration tool. In many respects, aiming to provide large productivity gains to a wide range of automation challenges. While Ansible offers more productive drop-in replacements for many basic features in other automation solutions, it also tries to solve other important unresolved IT challenges.

LOGICAL ARCHITECTURE OF CLOUD OS:

Figure 1 represents the proposed cloud operating system with different levels and components in each layer is also specified in the respective levels. The Cloud object can be defined as a set of Local operating system processes that run on a single node, which intertwine with each other and are assigned locally. A random identifier of appropriate length is used to minimize the risk of ID collision throughout the system. A process in the cloud (CP) is a collection of objects in the cloud that implement the same, that is, similar to the distributed application. Here we refer to the small number of PCs that regulate physical allocation, access control, accounting and measurement of resources such as the central cloud space. CPs that are executed in user space directly by users and by these CPs are called User Applications. Cloud libraries are the CPs normally used by applications and other libraries. Applications can interact with libraries and kernel CPs through the network through a series of standard interfaces called Cloud SystemCalls.
The Cluster Configuration System (CCS) managed the cluster configuration and provides configuration information to other cluster components in a Red Hat cluster. CCS runs on each node in the cluster and ensures that the cluster configuration file in each cluster is updated.

Figure 1: Logical architecture of cloud OS

For example, if a cluster system administrator updates the configuration file on node A, CCS propagates the update from the node to the other nodes in the cluster. Cluster Name: Displays the name of the cluster, the cluster configuration, and the basic time properties used when the cluster is clustered or clustered. Cluster: shows each node in the cluster, specifying the node name, the node ID, the quorum classification number and the fencing method for that node. Fence device: shows nearby devices in the cluster. The parameters vary according to the type of fencing device. For example, the cluster configuration defines the name of the power driver, its IP address, login and password. Managed resources: shows the resources needed to create cluster services. Managed resources include the definition of domains, resources and failover services. Together with the managed resources, it defines the cluster services and the failover behavior of the cluster services.

Figure 2: Cluster Overview

**FEATURES PROVIDED BY THE CLOUD USER SPACE:**

To fully exploit the potential of a general-purpose cloud operating system, developers must have access to a number of standard modules to meet the common requirements of widely deployed applications. As a general principle, cloud libraries provided by the cloud operating system should allow developers to control the required level of data replication, consistency and
availability, and also how error handling is performed when they are not met. The application developer can focus on the specific properties of the application, knowing that the system will do everything possible to meet the established requirements. For example, when an application requires high availability and can handle temporary inconsistencies, the library can provide support for final consistency rather than greater consistency.

AUTOMATED PROVISIONING USING ANSIBLE:

Because the cloud infrastructure is a very large environment consisting of big data, different architectures and more, the information technology industry has understood the need for reliable code for the implementation of customer needs. The reliability of the code is considered an important problem since the final destination environment changes dynamically and if the code dissolves into such a high gaming environment, customer expectations will not meet and will be disappointed. A DevOps team intends to provide the application through high-end code. An instrument is necessary for configuration management together with continuous delivery. For this requirement, Ansible is used as an information technology automation engine that automates cloud provisioning. The main reason why Ansible is our configuration management tool is the fact that it is designed for multi-layered implementation. Ansible represents the information technology infrastructure by telling how servers interact rather than a single server at
A key concept of Ansible is the use of inventory files that is a single point of truth. In addition, the cloud infrastructure is made up of virtual machines that come and go very often, so keeping a single inventory file in an environment with such dynamic provisioning is cumbersome. But Ansible has all the features to handle such situations right from dynamic inventory files to full automation modules. All the modules require the famous boto library. Since Ansible is written in python, it makes it extremely portable and flexible. In this work, limit will be set for deployment to provisioning of the license administration system component only; before this the overview of the deployed services and dependencies for license administration system have initialized. The typical production deployment should contain redundant services, load balancing application, clustered back-end databases and deployment performance for centralized repository. The additional support repository includes a central repository of script, tools and application along with monitoring at system and service level. In this new architecture Ansible playbooks are responsible for replace the script for idempotency and much more. To start with the deployment process, Ansible plays important roles. Roles can be understood as include directives for automation purpose. In this scenario, a number of roles are written which deploy our application in cloud environment. To start the deployment of databases of different components in one single instance i.e. the databases of index service; license administration system and usage recorder are on one single EC2 instance. The first role is the database role which provisions the database over the instance. As a prerequisite the sql database is install using the sql role and then starts up with our application role. After this script gets copied on the target machine, send the specific customer reference id script to the target. Right after the completion of this task check if a database already exists in the instance and if it does make sure it gets deleted by using the absent tag. Then move on to create the user for our new database and create the application specific database with the requested database name. Once this is accomplished, the task to run sql scripts against database get executed and also the conditional cloud deployment request is also taken care off in the playbook.
Cluster Configuration and Management:

1. Setting Up Hardware

Setting up hardware consists of connecting cluster nodes to other hardware required to run a Red Hat Cluster. The amount and type of hardware varies according to the purpose and availability requirements of the cluster. Typically, an enterprise-level cluster requires the following type of hardware. Cluster nodes — Computers that are capable of running Red Hat Enterprise Linux 5 software, with at least 1GB of RAM. The maximum number of nodes supported in a Red Hat Cluster is 16. Ethernet switch or hub for public network - This is required for client access to the cluster. Ethernet switch or hub for private network - This is required for communication among the cluster nodes and other cluster hardware such as network power switches and Fibre Channel switches. Network power switch - A network power switch is recommended to perform fencing in an enterprise-level cluster. Fibre Channel switch — A Fibre Channel switch provides access to Fibre Channel storage. Other options are available for storage according to the type of storage interface; for example, iSCSI or GNBD. A Fibre Channel switch can be configured to perform fencing. Storage — Some type of storage is required for a cluster. The type required depends on the purpose of the cluster.

2. Installing Red Hat Cluster software

To install the Red Hat Cluster software, you must have rights to the software. If you use the Conga configuration GUI, you can allow the cluster software to be installed. If you use other tools to configure the cluster, protect and install the software as you would with Red Hat Enterprise Linux software. The following cluster configuration tools are available with Red Hat Cluster: Conga: This is a complete user interface for installing, configuring and managing Clusters, computers and Red Hat storage connected to clusters and computers. System-config-cluster: this is a user interface for configuring and managing a Red Hat cluster. Command-line tools: a set of command-line tools for configuring and managing a Red Hat cluster.

3. Configuring Red Hat Cluster Software

Configuring Red Hat Cluster software consists of using configuration tools to specify the relationship among the cluster components. The cluster nodes are connected to one or more fencing devices. Nodes can be grouped into a failover domain for a cluster service. The services
CONCLUSION:

In this document, we propose a server system in the cloud based on Linux to control applications through cloud computing. Introducing the cloud operating system, Cloud OS, to make cloud computing more powerful. Cloud OS aims to provide an expressive set of resource management options and metrics to applications to facilitate cloud programming. Therefore, the operating system in the cloud provides fast, flexible and secure access to the computer and network environment. In addition, we have enabled the grouping in the servers. Normally, the main way to communicate the state of the node is through a network device (commonly Ethernet), although in the case of a possible network failure, the quorum can be decided through secondary methods such as shared storage or multicast. Software services, file systems and network status can be monitored and controlled by the cluster set, services and resources can be transferred to other nodes of the network in case of failure. In addition, we have suggested the method to build embedded Linux in real time and produce the patch by applying a patch in real time. We hope it will be possible to replace the existing high price of commercial RTOS, and it will be useful for developers who want to build the Linux Embedded system in real time with low cost. This will lead to changes in the production cycle and will improve links in the various components needed to create the final product. With the advent of cloud computing, new dimensions of automation deployment are coming into the picture. The features offered by cloud providers are becoming more vivid as technology progresses. More number of customers demand cloud deployment of requested application for increased flexibility. This paper presented the new way of deploying applications on cloud using automated provisioning. there is a use of Ansible, a 4th generation configuration management tool for automated provisioning. It is shown how flexibility of Ansible makes it easy to automate the whole procedure of deployment of applications.
REFERENCES:

1. Vinoth Kumar G, Ashish Kumar Pillai “A Modified Long Lasting Pacemaker Using Piezoelectric Nanowires For Treating Cardiac Diseases” international journal of pure and applied mathematics, Volme 118 (20), Page no.709-717, ISSN: 0974-2115


3. Dehradun, India, 4-5 September 2015
Cloud on Linux Operating System :
Pritee Gavhane, Pratibha Kurumkar, Madhuri Jagtap, Dharti Mahajan

Journal No : ISSN 2320–088x