

Review on Cloud Automation Tools

Faraz Hussain

Department of Electrical and Electronics, RV College of Engineering, Bangalore, India

Key words: Cloud, automation tools, virtual machine, bunches

Corresponding Author:

Faraz Hussain

Department of Electrical and Electronics, RV College of Engineering, Bangalore, India

Page No.: 207-211

Volume: 19, Issue 9, 2020

ISSN: 1682-3915

Asian Journal of Information Technology

Copy Right: Medwell Publications

Abstract: Traditionally, the outstanding burdens were conveyed and overseen utilizing manual procedure. It was tedious due to the redundant assignments acted regarding scaling, designing and provisioning of the bunches, virtual machines. The cloud robotization devices help us to speed the procedure. There are different cloud robotization instruments while there is no single apparatus reasonable for each circumstance. This study sums up the advantages offered by different cloud robotization apparatuses.

INTRODUCTION

Cloud computerization is a product arrangement which empowers the designers and the IT group to introduce, arrange and deal with the distributed computing administrations. Hence, permits organizations to pick the perfect measure of assets required for distributed computing. Offering types of assistance on request is the fundamental point of distributed computing. In any case, as a general rule, somebody needs to make them, continue observing them ceaselessly and erase them when they are not at this point required. This can require a gigantic manual exertion.

Cloud robotization to a great extent rotates around the Infrastructure as a code. Cloud computerization procedures and instruments utilize the asset pools from the cloud to make basic design things, for example, Virtual machines, virtual private systems and holders. Examples can be made and conveyed utilizing these design things^[1]. For instance, a particular number of compartments can be made utilizing a cloud mechanization format which can be utilized for a microservices application. Also,

used for connecting a storage and a database, virtual network configuration and creating load balancers^[2].

Aside from the sending cloud robotization can be utilized for remaining task at hand administration and screen the presentation of utilization and outstanding task at hand.

CLOUD AUTOMATION TOOLS

AWS cloud formation: The Amazon web services cloud formation instruments give executives and engineers a straightforward strategy to manufacture set of related assets, flexibly and update them in sorted out and unsurprising way. Cloud formation offers test layouts or we can assemble our formats to speak to AWS instruments, pertinent conditions our gadget. Endless supply of the AWS instruments, we can change and update them in an oversight and predictable way, viably including adaptation control for AWS foundation.

Features

Dependency management: During stack the executives conduct AWS cloud formation consequently handles

conditions between our assets. We don't need to stress over indicating in which request the asset is produced, adjusted or erased. The suitable activities to be performed for stack activities are dictated by the cloud formation for every asset.

Authoring with familiar programming: The cloud development kit of the AWS enables user to define applications using familiar programming languages like Typescript, Python, Java and NET. Additionally, it enables us to provide our infrastructure using AWS cloud formation directly from our IDE.

Authoring with JSON/YAML: Using AWS cloud formation an entire network can be modelled in text. YAML or JSON file are used to define resources required for configuring or building AWS.

Safety controls: The provisioning and refreshing an AWS framework are mechanized by AWS cloud formation in a made sure about way. Rollback Triggers can be utilized to indicate the cloud watch We can utilize Rollback Triggers to determine the cloud watch caution observed by the cloud formation and utilized for the presentation of slack and supplant process. In the event that any of the cautions are penetrated, the entire stack activity is gone before back to send state by cloud formation.

Managing of cross-region cross-account: AWS stack sets allows us to provide collection of tools provided by AWS with a single cloud formation template across multiple accounts and regions. Stack sets guarantees that multiple accounts and regions stacks are automatically and safely supplied, changed or removed.

Terraform: An apparatus made by HashiCorp, helps in provisioning the foundation as code. HashiCorp design Language is utilized to arrangement a datacentre foundation. Utilized an instrument for forming, constructing and changing framework productively. Many existing specialist co-ops are overseen by Terraform with the assistance of custom in-house arrangements.

Features

Infrastructure as code: The infrastructure can be described using high-level which is configurable and reusable. It allows us to create a blue-print of infrastructure which can be versioned too.

Execution plans: Terraform provides as step named as planning step which allows user to know the complete details of execution when the apply is called.

Resource graph: The resources which can created or modified independently are parallelized in Terraform by building graphs. This help the Terraform to build the infrastructure faster and efficiently.

Change automation: Terraform uses the execution plan and the resource graph to make a complex change. Thus, allowing a complex modification with minimum human errors and interaction.

How Terraform works: Terraform design which is module based, permits the engineers to expand Terraform by either composing new modules or adjust the current one. Terraform has two principle parts: Core and Plugins.

Terraform core: GO programming-based order line apparatus. It helps in making foundation as code, asset state the executives, asset chart development, execution of plan and correspondence over RPC with modules.

Terraform plugins: Conjured by the Terraform centre over Remote Procedure call as doubles which are executable. Terraform modules are utilized for confirmation, characterizing assets and make API calls required for libraries introduction.

Microsoft azure automation: Microsoft Azure is an incredible instrument for process robotization. It streamlines cloud the executives and computerization. Purplish blue enhances and broadens existing speculation and aptitudes of reconciliation, organization and mechanization by offering the accompanying highlights and advantages.

Features: Upgrade and expand existing IT speculation and aptitudes by incorporating your current frameworks with different reconciliation modules. This empowers quick reconciliation with different frameworks. Gives adaptable and dependable arrangements by the coordination of procedure outstanding tasks at hand according to your need.

Improves administration unwavering quality along your numerous LOBs, offices, frameworks and instruments. This breaks 'working in storehouses'.

Improves consistency with lower costs: Azure Automation permits you to concentrate on business-basic work that includes esteem while diminishing mistakes and expenses. Microsoft Azure permits undertakings to improve cloud the executives by evacuating tedious and repeating errands, sparing you both time and cash.

It additionally, improves productivity and dependability as reconciliation with any assistance is made conceivable by Windows Power Shell contents and an exceptionally accessible Engine.

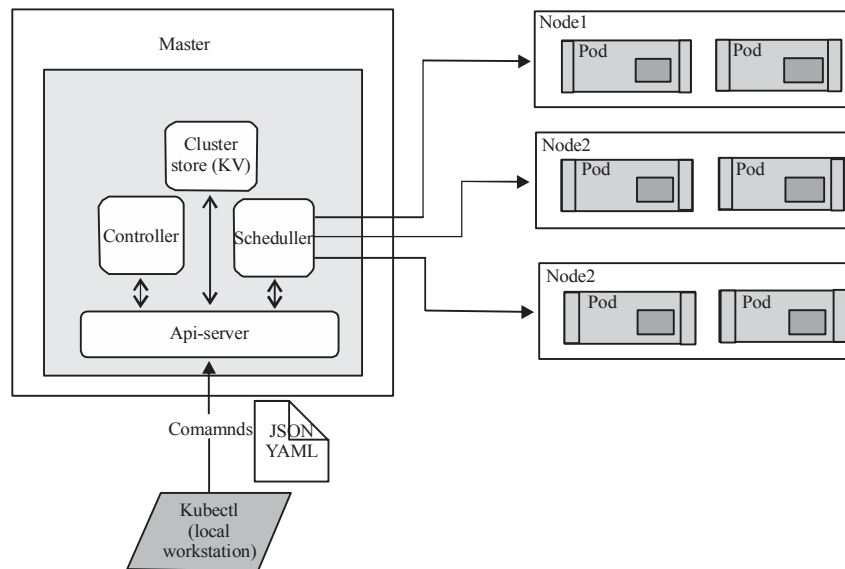


Fig. 1: Kubernetes architecture

Benefits

Cost efficient: You can totally move away from those daily practice and tedious cloud the board assignments which are frequently mistake inclined. This gives you additional time on work that enhances your business. With decreased blunders and expanded effectiveness, Windows Azure can likewise assist you with lessening operational expenses.

Upgrades existing work processes and makes your own work processes: Azure computerization permits you to use the current work processes or structure your own work processes. This component causes you to make, send, screen and keep up Azure assets alongside outsider applications. Use your current runbooks or make your own whichever best suits you.

Incorporate with different administrations: Azure Automation flawlessly works with sites, VMs, servers, stockpiling, and other prevalently utilized Azure administrations. It can likewise be utilized with any help offering or open web APIs. Conveys quicker and dependable assistance: Azure productively handles frameworks, devices and offices that permit you to convey benefits quicker and all the more proficiently.

Kubernetes: Kubernetes is a containerized programming running and overseeing through a network of machines at its centre stage. It has been worked with strategies that permit consistency, adaptability and high accessibility for the life of containerized applications and administrations. You will choose how your applications work and how they will speak with different applications or the outside

world as a Kubernetes client. You can refresh or uninstall your administrations, update smoothly and move traffic between different adaptations of your applications to check usefulness or rollback issues. Kubernetes gives crude interfaces and stage composable that empower elevated levels of adaptability, force and certainty to characterize and deal with your application (Fig. 1).

Architecture: It is important to know how Kubernetes can have these functionalities, how it is structured and organized at a high level. Kubernetes can be regarded as a layer-built network with the complexity at the lower levels being resumed by any higher layer^[3].

Master node: The master node is the first and most critical part for cluster management of Kubernetes. An administrative gateway for activities of all kinds. In the cluster, fault tolerance can be monitored by multiple master nodes. There are many components to the master node such as API Server, Scheduler, Controller Manager and ETCD.

API SERVER: ACTS AS ENTRY POINT FOR ALL THE COMMANDS (REST) USED IN CLUSTER

Scheduler: The programmer running node functions. Stores information about the usage of resources by each node. The company shall divide the workload. It also lets you monitor how cluster nodes use the working load. It helps you to position the workload on the available resources and to accept the workload.

Master/slave node: These are the worker nodes that play an important role in providing the necessary services for networking and collaboration between containers, allowing you to allocate resources to scheduled containers. worker nodes are also important.

Kubelet: Gets an API server Pod setup and ensures the containers listed are up and running. Docker container: These containers run on worker's nodes that operates on the Kube-proxy (the kube-proxy helps in balancing the load and network proxy for the output of a single working node) pods that are configured.

Kubernetes basically ties numerous individual virtual as well as physical machines in bunch that is connected to every server through a typical system. This bunch is the physical system for arranging all the modules, capacities and outstanding burdens of Kubernetes.

In the Kubernetes biological system, the machines in each bunch have a specific capacity. The ace machine goes about as a machine (or as a little network in profoundly open establishments). Programming interface is given to clients and clients, security minds different servers, how best to scatter and representative work (booking) and arrange coordination with different parts as the cerebrum for the bunch. The server fills in as an entryway and mind. The ace server fills in as the primary concern of contact with the bunch and predominantly bolsters the unified rationale gave by Anonymous^[4].

Certain PCs in the bunch are known as hubs: servers with neighbourhood and outside assets for remaining burden adequacy and activity. The product and administrations are made to run in holders to help with disconnection, adaptability and the board so hub is fitted with the runtime of its compartment (for example Docker). The holders are either made or annihilated dependent on the directions got by the hub from the ace. System rules are changed by traffic courses and travel.

Puppet: A publicly released design the executive's apparatus, utilized for private, open and half and half mists. It gives its own arrangement language Puppet DSL (Domain-explicit language). The framework arrangements and foundation as code are characterized utilizing a DSL. Manikin venture arranges the errand based multi-gadget the board and order execution. It gives the GUI comfort to arrange and deal with all the sent cloud machines^[5].

Architecture: Puppet is based on master-slave architecture. The client and server are interconnected by the secure socket layer. The puppet architecture has following components^[6] (Fig. 2).

Puppet master: Puppet master is a Linux based system which handles all the configuration related process in the form of puppet codes. The SSL certificates are checked and marked by the master.

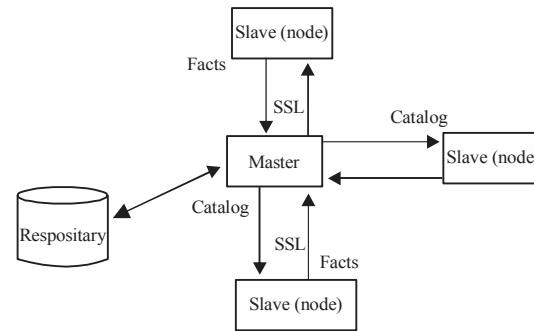


Fig. 2: Simple puppet architecture

Puppet slave: Puppet slave are working systems used by the client. Puppet master maintains and manages the slave. The Puppet agent daemon service runs inside the slave.

Repository: Repository stores the node and server related configuration. Puppet keeps the official bundle archives of operating systems. Puppet collections help in gathering the majority of software required for the utilitarian Puppet deployment.

Catalog: The compiled format of configuration and manifest files written in Puppet are called as catalog. It defines the state and dependency data for all the assets that ought to be overseen by hub in a specific request.

Facts: The facts are the key-value pair that contain the information about the node and master machine. Facts are used for determining the state of any slave as facts represent the client states such as operating systems, IP, network interface.

Features

Idempotency: Puppet supports idempotency, thus same set of configurations can be run multiple times on the same machine. Puppet basically checks for the current status of the target machine and makes changes only if there is change in configuration.

Cross-platform: Puppet helps in configuring the system. Implementation details is not taken into consideration as it is handled with the help of resource abstraction layer.

Salt stack: A cloud mechanization apparatus that utilizes the Infrastructure as code for design and arrangement computerization. It is open-source and python-based programming utilized for remote execution, arrangement the executives and cloud control. Salt backings numerous cloud suppliers, for example, Azure, AWS, OpenStack, IBM Cloud and VMware. Salt Stack arrangements server and foundation with assistance of focal storehouse.

Architecture: Salt stack has highly modular design configured to work with multiple servers ranging from

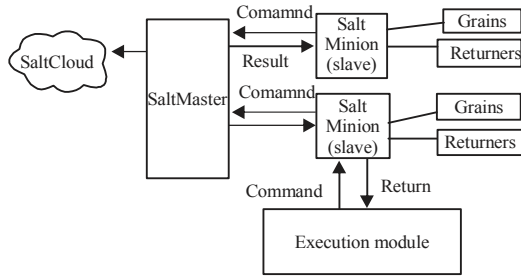


Fig. 3: Simple architecture of SaltStack

network system in local to data centers deployment. It has a simple client-server model with multiple daemons working in co-ordination. Salt architecture composes of following components:

Salt master: A master daemon that sends various commands and configuration to slave daemons.

Salt minions: A slave daemon that receives commands and configuration from the master daemon.

Execution: Monitoring in real time using the adhoc and module commands executed against the slave daemons.

Formulas: These are the states used for various tasks such as starting a service, monitoring permissions and installing packages.

Grains: System used for detecting various information and storing in RAM.

Salt cloud: Cloud hosts are monitored by the salt cloud

Salt SSH: Used to SSH on systems and execute various commands.

Runners: Applications on the master end used using the run command of salt (Fig. 3).

Features

Scalable and fault tolerant: Salt stack has high fault tolerance, it can connect to multiple masters at once and used YAML to configure for all the masters at once. Salt master can handle around ten thousand minions.

Python API: Salt is a python based and provides a modular, extensible programming interface to configure and monitor applications.

Authentication: Salt provides and secure SSH key pairs used for authentication.

Execution model: Salt provides tool which can run commands in remote systems parallelly.

CONCLUSION

This study discussed the highlights, engineering for every one of the cloud computerization devices, likewise characterizes the significance of same. The future work includes the making of foundation in every one of the instruments talked about and send them. Additionally, investigate the exhibition, dependability and adaptability of the sent framework.

REFERENCES

01. Zhang, R., Y. Shang and S. Zhang, 2014. An automatic deployment mechanism on cloud computing platform. Proceedings of the 2014 IEEE 6th International Conference on Cloud Computing Technology and Science, December 15-18, 2014, IEEE, Singapore, pp: 511-518.
02. Callanan, S., D. O'Shea and E. O'Regan, 2016. Automated environment migration to the cloud. Proceedings of the 2016 27th Irish Signals and Systems Conference (ISSC), June 21-22, 2016, IEEE, Londonderry, UK., pp: 1-6.
03. Vayghan, L.A., M.A. Saied, M. Toeroe and F. Khendek, 2018. Deploying microservice based applications with Kubernetes: Experiments and lessons learned. Proceedings of the 2018 IEEE 11th International Conference on Cloud Computing (CLOUD), July 2-7, 2018, IEEE, San Francisco, California, pp: 970-973.
04. Anonymous, 2020. Kubernetes documentation. The Linux Foundation, San Francisco, California.
05. Hintsch, J., C. Gorling and K. Turowski, 2015. Modularization of software as a service products: A case study of the configuration management tool puppet. Proceedings of the 2015 International Conference on Enterprise Systems (ES), October 14-15, 2015, IEEE, Basel, Switzerland, pp: 184-191.
06. Anonymous, 2018. New features. Puppet Software Company, Portland, Oregon.