International Journal of Engineering & Technology, 7 (2.33) (2018) 637-641

International Journal of Engineering & Technology

Website: www.sciencepubco.com/index.php/IJET

Research paper

Design of devops solution for managing multi cloud distributed environment

Baskaran Jambunathan [1] *, Dr. Y. Kalpana [2]

[1] *Research Scholar, VISTAS, Chennai, India*
[2] *Department of Computer Science, VISTAS, Chennai, India*
**Corresponding author E-mail: ykalpanaravi@gmail.com*

Abstract

DevOps is the emerging field in the area of IT infrastructure management and cloud automation. Organization are spending time and money in terms of continuous improvement and optimization in their infrastructure management and finding to avenues for efficiency and produc-tivity improvement, reduce redundancy and optimise cost in the area of development and deployment of their enterprise application. More importantly, organisation have multiple environment to manage and hence need to have a mechanism to continuously monitor and optimize operational task and improve collaboration with various teams across organization. In addition, many companies are migrating or migrated on to cloud, some are having multiple cloud environments across regions, and hence managing it is even more difficult and need to have a mechanism to address the key challenges and risk in managing devOps in multicloud. . In this article, we would like to discuss the key challenges and how it can be addressed in a multi cloud environment and proposing an integrated solution addressing all challenges with right set of tools and framework design which enables devops in multi cloud efficiently.

Keywords: *Cluster Management; Docker; Flocker; Fabric8; Jenkins; Kubernetes; Rancher; Micro Services; Wrecker*

1. Introduction

Organisations across the globe are primarily focussing on improving their operational efficiency and want to minimise their IT spent on various areas in their cloud transformation journey. DevOps is normally referred as a a set of industry operational practices that mainly focus on collaboration and communication of both software developers and information technology (IT) professionals in order to automate the process of software delivery and infrastructure changes. It aims at provide a platform and establishing a culture in making the environment where building, testing, and releasing software can happen rapidly, frequently, and more reliably. Devops brings improvement and standardization in both development and operational process, brings these two teams together and making room for collaboration and communication between them. It facilitate respective team to focus only on their job and not to worry too much on the other areas and minimize the dependencies to a large extend.

Fig. 1: Devops Process.

DevOps is not about combining traditional responsibilities into a single team. But also devops is a set of guiding principles and methodology which promotes software excellence throughout its SDLC lifecycle.
DevOps is focused on the following key objectives:
Improves the collaboration and communication between Developers and operational engineers and breaking down the siloed way of working.
Improve the performance through constant feedback from customer and accelerate time to market with customer
Focus on quality of software delivery though continuous improvement, integration and deployment.
Institutionalise experimentation, adapt to failure and realise the mistakes and provide room to rectify quickly.
Provide enough space to automate traditional operational and development responsibilities.
Help in continuous monitoring and help in maturing the environment through optimization and automation.
DevOps and Agile process are comparable. Many people think that Agile and DevOps are similar, but they are actually similar in some sense and differ in many ways. Agile process helps organization to bring the culture of representing a change in thinking and practice that should lead to organizational change, devops brings more emphasis on implementing organizational changes in practice to achieve its goals. Agile is all about processes (like Scrum and Kanban) and devops is all about technical practices like CI, CD, test automation and Infrastructure Automation. Agile teams, in general, have used automated build, test automation, continuous integration and continuous delivery as part of software development and delivery process to a large extend. Hence we can consider devops as not a separate concept but a mere extension of agile to include operations as well in the definition of cross-functional

agile team, collaborate together and work as ONE team with an objective to delivery software fully to the customer.

2. Key focus area of develops

DevOps consist of key functional areas. We would like to introduce those key concepts in the following section.
Continuous Integration (CI): It is the task that developers integrate the code into the shared repository continuously multiple times in a day Each check-in is then verified by an automated build, allowing teams to detect problems early.
Continuous delivery It is a software development practices designed to ensure that code can be rapidly and safely deployed on to production by delivering every small change to a production-like environment and ensuring business applications and services function as expected through rigorous automated testing.
Continuous Deployment (CD): It is a strategy for software releases wherein any commit that passes the automated testing phase is automatically released into the production deployment.
Build and Release management: In any typical organization, multiple environment are available for software development process like development, testing, production etc. and hence multiple build and release cycle has to be made available and should be automated.
Configuration management: Configuration management gives us the consistency across environments, which we need for continuous delivery. Since multiple release and code repository have to be maintained and managed for multiple releases, any issue or problem, we need to revert or rollback to previous versions and hence it is essential.
Orchestration: In any multiple environment management, lots of tasks are performed at different places and needs to be co-ordinated, collaborated, communicated with different workflows and needs to be managed effectively. Hence a good orchestration mechanism is essential to co-ordinate, manages, executed as a single task across different environment by managing dependency, and does proper notification.
Configuration as a code: It is the discipline of thinking of configuration elements of a software system as you would think of code. It allows the entire configuration to store as source code. It enables us to collaborate with operations on the application environments to ensure that they have correct configurations. It allows continuous deployment and prevents continuous drifts.

3. Devops key objectives

From our analysis and understanding, we found that there are certain key objectives, to meet in designing devops solution. Some of them are listed below.

- Focussed mainly Application and end user perspective.
- Improving collaboration between Stakeholders in the project.
- Focussed on to improve the system performance over all.
- Improve development speed
- Increase Service orientation
- Improve automation and increase productivity
- Monitoring everything and take corrective action.

Traditionally devops solution are tried and implemented by the project team in their on -premise environment. Since the inception of cloud and companies started migrating on to cloud, cloud service providers started providing features, API, services etc. in their cloud and facilitating project team to use the devops in cloud effectively.
Companies objective are more towards productivity improvement and cost reduction and wanted to optimize their operational task and more focussed on automation. By improving collaboration between different groups participating in any project, reduces the redundancy and assumptions and improve the throughput of their task. Also the teams involved need to know how the system works as a whole and how the individual components impact the performance. This requires transparency, visibility, consistent across tools and teamwork. Every team member need to understand the impact of the performance on cost and need to have a system which is very effective. In addition, we need to have a process which accelerate the innovation and learning and improve the development speed.
Services availability is the key for system productivity and if any of the services are not available, it has to be identified and replaced immediately without impacting rest of the system. System should be scalable and available any time to the user. Organization realise that the manual error needs to be reduced and repetition to be avoided as much as we can and focus more on automation in the area where it is possible.
Automation enables speed and consistency and allows the team to focus where it matters. Hence we need to automate the process like build, release, deployment, configuration, provisioning etc. and minimise the repetition and error.
Another key area is monitoring. It is a discipline to be followed and provides intelligence and brings all the team members in the same page. If we cannot measure, we cannot improve. DevOps facilitate the feedback mechanism so that the problems are identified quickly and rectified. Cloud service provider's realised all these aspects and enable it in their platform for the users to make use of it and make organization to take benefit out of the same.

4. Devops in cloud

DevOps process is applicable in on premise, virtual and cloud environment. Most of the cloud service providers enabled devops process addressing the functionality explained in the previous section as part of their services. Cloud computing, whether inside the firewall or purchased from a service provider, is essential to success with devops. Nearly every major cloud provider offers a set of platform as a service (PaaS)[1] tools that are fine-tuned to their environment. Following are the key criteria for choosing devOps tools in cloud.

- Compatibility with public cloud services.
- Support for both network and IT (i.e., server, storage) resources.
- Link between deployment and monitoring/management tools

With devops services, cloud service providers enable the organization to manage, automate and improve efficiency of operational process. DevOps provides the following core capabilities that can help manage cloud-based computing like infrastructure as code, provisioning servers, automated application deployment, knowledge sharing, managing the full application lifecycle management, continuous integration and delivery, continuous QA and testing etc. Hence devops is integral part of Cloud services to manage both infrastructure and application when we deal with multiple environments

5. Decops in multicloud

Now a days, it is more evident that many organizations does not want to stick with single cloud service providers to avoid vendor lock-in and wanted to have two or more cloud services providers and wanted to have a truly distributed cloud environment. Organizations are looking for flexibility in moving from one cloud service provider to another and wanted to manage different workload across different cloud and also have different environment in different cloud service platform. The most common practice would be to have their dev/test in private cloud and have their production in public cloud and to have a hybrid model. It introduces complexity, heterogeneity and various other problems that come with the use of different clouds in managing development and deployment process and difficult to collaborate, orchestrate, and automate the process across clouds.

Moreover, developing applications for multi-clouds impacts established enterprise procedures and business models.
Organizations want to use and deploy different pieces using the best infrastructure across multiple and largely incompatible cloud platforms. It's always the questions in our mind that whether it is possible to manage and operate cloud services in multiple clouds, taking care of the heavy work with automated migration of application's components to better venues according to our specifications.

5.1. Key challenges in multicloud

In our understanding following are the key challenges we could see when we manage multicloud environment and need to have a good framework / tool to manage devops process across different platform. Some of the challenges are as follows
Provisioning and configuration of different workloads in both infrastructure and application side across different cloud services.
Have a common build and release management platform and tools to manage the process and automate them to minimize issues and risks.
Ability to deploy in multiple target environments and still take the advantage of native cloud capabilities. Changes needs to continuously integrated, deployed and tested in different cloud platform.
Application developed needs to be distributed and communicated to each other effectively to complete the task. Also we need to have applications which are scalable, collaborative, lightweight, portable and heterogeneous in nature.
Target cloud environment tend to undergo rapid changes and the devops tools are expected to adapt to those changes automatically.
Security and governance are part of process and specific to cloud platform and hence devops process has to address these governance process mentation in the cloud platform
All services and components needs to be monitored, managed and changes are reported effectively to address the operation and business needs
Developers' needs to address these challenges while managing the multicloud environment in logging, reporting, testing, deployment etc., and selection of right set of process and tools is the key for this management. In the following sections we would like to address those set of process and tools which we experimented in meeting this objectives.

5.2. Devops design for multicloud

Managing devops process in multicloud environment, as discussed earlier, has to address many challenges. We tried and experimented multiples tools, process and framework to address the above challenges and recommending the right choice for building complete end to end devops solution for managing distributed application in a multicloud environment. Below diagram gives high level overview of our key are of consideration.

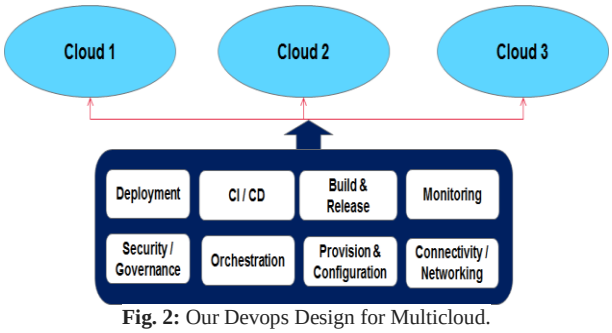

Fig. 2: Our Devops Design for Multicloud.

Portability across multicloud: We used Docker [1] container as a basic unit of application packaging and deployment across multicloud. Since it is light weight, portable, language neutral and scalable across platform, we used this as a unit of deployment.

Application design for multicloud: We used microservices[2] as a model for application design for various services and functions, which can be ported with dockers and can be deployed across host and cloud using various tools.
Multicloud Networking: Initially, to start with we would like to ensure that the cloud platform in consideration for multi cloud needs to be connected and integrated and application deployed needs to communicate to each other. Overlay network [3] is the concept used here for containers to communicate to each other. It is a method of using software to create layers of network abstraction that can be used to run multiple separate, discrete virtualized network layers on top of the physical network, often providing new applications or security benefits. We used a tool called Flanne l [3] for setting up the overlay network across cloud.
Docker Engine [1] supports multi-host networking out-of-the-box through the overlay network driver. . The Docker multi-host networking can also be extended to hosts running on different clouds. This would enable containers running in multiple cloud to communicate with each other seamlessly. For example in a typical multi-tier application, the app tier can run in AWS cloud, the Database tier can run in Azure and web tier can run in Openstack. These three can be made to communicate to each other through overlay network. We implemented overlay network to communicate across multiple clouds.
Provisioning and Configuration: We used puppet based like scripting for provisioning the containers (dockers) across different cloud in consideration and used Ansible script for deployment across cloud environment.
MultiCloud Orchestration: There are many open source tools available in management, orchestrate and automate process in multicloud management like Docker Swarm, Apache Mesos, Google's Kubernetes etc. Among this, Kubernetes is widely accepted tools for managing multicloud environment and are integrated with many cloud services and PaaS platforms available in the market. It has ability to scale to any production requirement and has good scalability, security, load balancing and support to Dockers and Pods which are widely accepted industry standard in true distributed application development.
We used a open source tool Rancher for setting up the kubernetes cluster. Rancher natively supports Kubernetes, making it simple and painless for users to deploy Kubernetes clusters. It automatically sets up the Kubernetes environment, deploying etcd, the API server, and all of the services necessary for a reliable deployment.
Kubernetes has in-built features for monitoring, Load balancing, scheduling, workflow management auto-scaling, self-healing, service discovery, lifecycle and health management of different cluster of Containers / VMs distributed across different cloud platform. We can configure the cluster spread across different cloud and ask Kubernetes to manage the cluster which has different applications, data storage and workloads which are truly distributed. Kubernetes is a powerful container orchestration tool that makes it possible to reliably deploy and operate containerized applications at any scale.
CI / CD in multicloud: In general when we develop and deploy application on to multiple clouds, managed by different teams with different languages and platforms, it is difficult to co-ordinate and manages the development and deployment with Continuous integration of changes. Jenkins is a popular Continuous Integration tool and kubernetes is for orchestration engine. We propose a open source tool fabric8 which is microservices based platform integrated with Docker, Kubernetes and Jenkins. With Fabric8, it makes it easy to create microservices, build, test and deploy them via continuous delivery pipelines then run and manage them with Continuous Improvement effectively. This addresses distributive application development and deployment effectively.
We used a tool called Wrecker, a Docker-native CI/CD Automation platform for Kubernetes & Microservice. It works with complex microservice architecture, utilise continuous integration and deployment, natively integrate with kubernetes in cloud and dock-

er native platform. Wercker automates deployment workflows so you can focus on building great applications.
Build and Release management: We used Jenkins for continuous build and release management. We configured the Jenkins server and integrated this with multiple cloud environments. Whenever the code changes happen, based on the cloud configured for a particular application service component, continuous deployment happens and changes are integrated effectively.

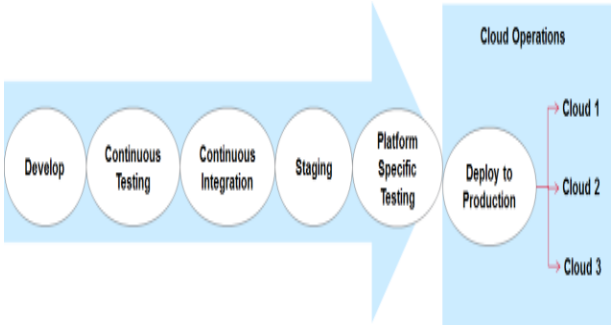

Fig. 3: Multicloud Release Management.

Platform specific Testing: In multicloud environment, we need to remember that we are taking the same code to different cloud environment. Hence it is essential to consider platform specific testing to ensure that all aspects of the code are testing in staging environment before moving on to production. The identified tool/process checks the application for issues that it may have when using some of the native features for each cloud platform. The automated testing process will help in identifying the platform specific issues and proactively identify and fix those issues. A good system should automate the correction of issues where it makes sense, and if they cannot be corrected, they should be returned to the developer for manual correction. Once that's complete, the application needs to go through the platform-specific testing step again.
Data persistence: Flocker is an agent-based cluster framework that works on top of docker and docker-services to manage and orchestrate docker based containers while providing data volume management services allowing for data persistence and portability. Additionally, the Flocker framework provides transparency by allowing access to container resources regardless of the host where the container may actively reside. Flocker provides an API and CLI for managing containers and data volumes, and works with multiple storage solutions. The pluggable nature of flocker is designed to work with any storage system, delivering the ability to integrate dockerized applications with existing storage backends. Any company or community that wants its storage to work with Docker can easily write a driver for flocker. We used flocker framework for managing data persistence across cloud when we move application from one cloud to another. This address the stateless nature of docker container and supplement the data cache issue with persistence.
Rolling Updates with Kubernetes: Rolling updates are the concept used for frequently delivering updates to applications in a continuous delivery model. A rolling deployment is used to reduce application downtime and unforeseen consequences or errors in software updates. In CI/CD through dockers we use rolling update methods to deploy changes on to dockerized environment. In a kubernetes clusters, deployment of application through docker container are achieved through rolling updates. With zero down time, kubernetes provides deployment support through rolling updates. The history of deployment is maintained in the registry and hence it is easy to roll back the deployment as well.
Hence in a typical multicloud environment, the applications are distributed and any changes in application through continuous delivery and deployment is achieved through rolling updates as kubernetes provides complete support to rolling updates.
Monitoring, management and reporting: We used a set of monitoring tools for monitoring containers deployed across various cloud platforms. We analysed multiple tools like dynatrace, cAdviser, weavescope, sematext etc and finalized dynatrace as a effective tool for monitoring and reporting as it has the ability to diagnose the dependency with different components and able to drill down the root cause of the problem. It provides a complete reporting feature which helps user to analyse various problems and derive a solution to address the issues identified.
Security and Governance: We found Docker data center (DC/OS) as a tool that can be used for managing security and governance across cloud. When we use docker ecosystem, Docker datacentre provides effective security and governance feature when managing multicloud environment which are integrated and configured with DCOS.

6. Integrated solution design and recommendation

DevOps in multicloud is a challenge. Managing and optimizing multiple cloud environment and administration of different environment is even more critical. In our proposed framework, which is tested and validated, we have identified different tools for various devOps activities. The functionality of each tool was discussed earlier in sections.
The Key aspect of our framework is that, we have a set of tools for multicloud platform management, where each one of them is meant for a particular aspect of DevOps. We looked at the functions like cluster management setup, multi cloud networking, CI/CD, orchestration and micro service deployment for distributed applications across different cloud. AWS, Azure and Google are the three different cloud considered for deployment and management.
We used Kubernetes for cluster management, although Docker swarm and Apache Mesos can also be considered for cluster management. Using Rancher, we setup the Kubernetes cluster and deployed in multi cloud environment. Rancher helps setting up and deployment of cluster in multicloud effortlessly. We used Flannel for setting up the overlay network across multiple clouds for easy docker deployment and bring up a multi-node kubernetes cluster with an overlay network. An overlay is necessary to fulfill the networking requirements for a fully functional kubernetes cluster. Having done the networking and cluster management, we used, wrecker for setting up the CI/CD across multiple cloud for continuous integration and deployment.

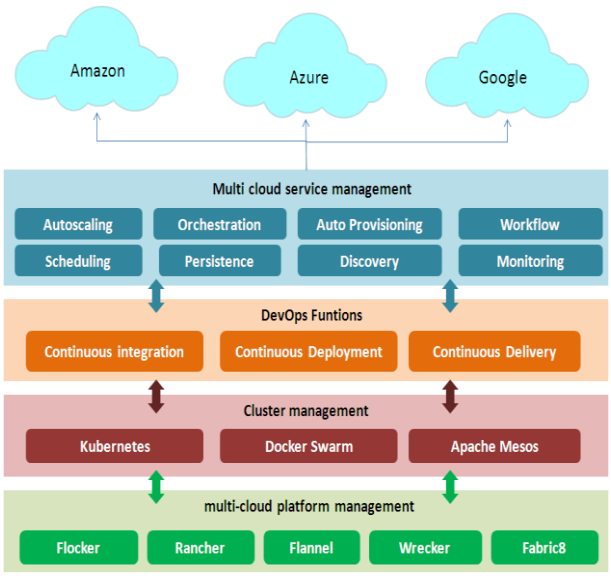

Fig. 5: Multicloud DevOps Framework.

Our application developed is using microservices, developed in Java using Spring Cloud design patterns. Based on the functionality of these services, different services are deployed in different cloud using Frabic8, tool for micoservice deployment. Fabric8

helps in develop, test and deploy microservice with CI/CD in multicloud environment.

Orchestration of services and managing workflow in multicloud is the key and we used Flocker for orchestration of Docker container across multiple cloud along with data persistence across cloud. Application management and data management are addressed using these tools in a multicloud environment. Fabric8 has the inbuilt Jenkins for managing CI/CD activities. Since Containers are the key for porting, managing across different cloud and micorservices are used for distributed application management, each application services uses data which needs to be managed and persisted effectively, tools like Flocker is of great use in a kubernetes environment. All functionality of DevOps services needed are addressed with these combinations and were able to demonstrate the development and deployment along with continuous integration and delivery in multicloud environment.

Enterprise are looking for solutions to manage their cloud environment where applications and workload are distributed, and application needs changes to meet the business objectives regularly. Hence addressing devops in multicloud is the need of hour and we felt that this framework address most of the concerns in managing devops across multiple cloud. We are further exploring options in making this approach more robust to address security, staleful management, and performance and service discovery aspects of applications working in distributed application deployment using microservices. As part of our research, we are still exploring more robust design to meet the challenges in devops in multicloud and provide a more stable and effective solution in future.

7. Conclusion

Enterprise are looking for solutions to manage their cloud environment where applications and workload are distributed, and application needs changes to meet the business objectives regularly. Hence addressing devops in multicloud is the need of hour and we felt that this framework address most of the concerns in managing devops across multiple cloud. We are further exploring options in making this approach more robust to address security, staleful management, and performance and service discovery aspects of applications working in distributed application deployment using microservices. As part of our research, we are still exploring more robust design to meet the challenges in devops in multicloud and provide a more stable and effective solution in future.

References

[1] Baskaran Jambunathan and Dr Kalpana Yoganathan ” Multi Cloud Deployment with Containers”, International Journal of Engineering and Technology (IJET) - e-ISSN : 0975-4024.

[2] Baskaran Jambunathan and Dr Kalpana Yoganathan ,’Microservice Design for Container based Multi-cloud Deployment”, Jour of Adv Research in Dynamical & Control Systems, 04-Special Issue, June 2017.

[3] Hui Kang, Michael Le , Shu Tao,” Container and Microservice Driven Design for Cloud Infrastructure DevOps” - 2016 IEEE International Conference on Cloud Engineering (IC2E), 4-8 April 2016.

[4] Armin Balalaie, Abbas Heydarnoori - Sharif University of Technology, “Microservices Architecture Enables DevOps: Migration to a Cloud-Native Architecture” IEEE Software (Volume: 33, Issue: three, May-June 2016), Page 42 – 52.

[5] Fauci AS, Braunwald E, Kasper DL & Hauser SL (2008), Principles of Harrison’s Internal Medicine, Vol. 9, 17thedn. *McGraw-Hill*, New York, NY, pp.2275–2304.

[6] Kim HS & Jeong HS (2007), a nurse short message service by cellular phone in type-2 diabetic patients for six months. *Journal of Clinical Nursing* 16, 1082–1087.

[7] Lee JR, Kim SA, Yoo JW & Kang YK (2007), The present status of diabetes education and the role recognition as a diabetes educator of nurses in korea. Diabetes Research and Clinical Practice 77, 199–204.

[8] McMahon GT, Gomes HE, Hohne SH, Hu TM, Levine BA & Conlin PR (2005), Web-based care management in patients with poorly controlled diabetes. *Diabetes Care* 28, 1624–1629.

[9] Thakurdesai PA, Kole PL & Pareek RP (2004), Evaluation of the quality and contents of diabetes mellitus patient education on Internet. *Patient Education and Counseling* 53, 309–313.