

# Alibaba Cloud DevOps Solution

## Bringing the Best of Cloud and DevOps to You



# Contents

01	The DevOps Handshake	03
02	Challenges of Traditional Software Development Models	04
03	What Makes DevOps the Preferred Software Development Mode	06
04	DevOps Best Practices for Your Business Growth	08
05	Trending DevOps Tools	10
5.1	Container	10
5.1.1	Kubernetes	10
5.1.2	Features of Kubernetes	11
5.1.3	Alibaba Cloud and Kubernetes	12
5.1.4	Business Scenario	13
5.2	Automation Using DevOps	15
5.2.1	Terraform	16
5.2.1.1	Features of Terraform	17
5.2.1.2	Business Scenario	17
5.2.2	Packer	19
5.2.2.1	Features of Packer	19
5.2.2.2	Business Scenario	20
5.2.2.3	Using Packer and Terraform with Alibaba Cloud	21
06	Alibaba Cloud - Offering the Best of Cloud and DevOps	22
07	References	23

# 01 The DevOps Handshake

For years, IT organizations have maintained development and operations teams as separate entities. Despite having the same organizational goals, these parallel functioning teams often would be at odds with each other. Organizations realized the need to create a new and merged functioning model which inspired the creation of the DevOps methodology.

DevOps is a model that goes beyond simple implementation of agile principles to manage the infrastructure. John Willis and Damon Edwards defined DevOps using the term **CAMS: Culture, Automation, Measurement, and Sharing**. DevOps seeks to promote collaboration between the development and operations teams. It aims to bring them closer in multiple aspects by eliminating barriers caused by physical location, organizational functionality, and business goals. By blurring the boundaries between conventional developer and operations roles, DevOps can minimize gaps and silos. This facilitates business continuity, accountability, and results across groups. DevOps agile methodology also helps businesses cope with changing requirements quickly and easily. In short, DevOps is a collaboration between development and operations that focus on enhancing the efficiency and innovation of delivering products to organizations and customers.

Through this white paper, CTOs, developers, and operation managers will get an initial insight into the benefits and significance of DevOps. It then dives into the main features of the open sourced container services and automation tools, including Kubernetes, Terraform, and Packer. The subsequent discussion also offers business case studies to help readers understand the implementation of the DevOps solution.



## 02 Challenges of Traditional Software Development Models

Traditional software development models include Rational Unified Process, the V model, and the Waterfall model. Modern processes, particularly the Agile method, are significantly replacing existing traditional development models. A primary reason for this gradual change is the challenges that organizations face when implementing traditional software development models, including:

### **Manual Mediation:**

A major challenge/limitation for traditional software development models is the need for manual interference. Such interventions often lead to non-repeatable procedures and introduce human errors. Manual intervention is also a hindrance to agility especially when it comes to testing and deployment. A manually performed testing makes continuous delivery and continuous integration impossible. Furthermore, manual testing increases the probability of producing defects and engaging in unplanned work. Deployments performed manually always contain the risk of deployment failures, resulting in unreliable and ill-planned tasks.

### **Inconsistent Environment:**

Organizations often function in irregular and non-uniform environments, creating complex configurations for development, testing, and production. Teams often waste days and weeks fixing bugs that are caused by inconsistent environment, wasting resources and time.

### **Restricted Monitoring:**

A significant advantage of DevOps is the possibility of continuous and efficient monitoring, which is not possible in traditional development models. DevOps can trigger custom alarms and several monitoring alerts, enabling users to utilize resources more efficiently. However, traditional models do not have such features. They are solely dependent on manual checks performed by developers. This process introduces errors and delays the delivery or roll out time of products.

### **Lack of Shared Ownership:**

Traditional software development models lack the concept of shared ownership, which create communication problems in organizations. This void was one of the primary reasons for the emergence of DevOps. In a traditional model, the development and the operations teams do not function as a single unit. Each team operates independently and is responsible for their respective tasks.

**Mismatch between Expectation and Reality:**

A major criticism of the traditional method is that the models do not consider real-world constraints. Over time, leading developers and tech experts have witnessed the failure of this notion as they often face roadblocks after initiating implementation. Designs that appear feasible on paper are often economically infeasible or complicated in practice.

**Obsolete Testing Practices:**

Traditional software development models comprise of outdated testing practices. The lack of continuous integration means that developers manually detect bugs and follow it up with a manual deployment. The developers would repeat this process until the teams agree on a usable solution, and then push the software to production. Every release on similar grounds would mean the introduction of more technical debt to the system, resulting in unplanned and unreliable work.

**Rigidity:**

Traditional methods of development were infamous for their rigidity when responding to change management. Developers often had to bring the process to much earlier stages to incorporate any desired modification or edits. The rigidity of conventional methods makes them unsuitable for an Agile approach of development.



## 03 What Makes DevOps the Preferred Software Development Model

With every passing day, organizations are adopting DevOps as their preferred software development model. RightScale's annual [State of the Cloud survey](#) revealed that DevOps adoption has risen from 54% in 2013 to 78% in 2017. Puppet's [2017 State of DevOps Report](#) revealed that by 2020, organizations are likely to displace CIOs that fail to adapt and transform their organizations to DevOps. The trend of adopting DevOps methodology is set to increase significantly as enterprises start to understand the advantages of DevOps. Below are some key benefits that organizations receive because of DevOps.

### **Convenient Automation:**

The DevOps model includes several functions such as OS patching, setting up CI – CD to automate deployments, report generation, and testing. The automation process promoted by DevOps involves creating builds, running test cases, and generating reports. Automated processes are error free, more reliable, efficient, and robust, leading to a reduction in delivery time.

### **End-to-End Monitoring:**

A crucial aspect of the DevOps model is the requirement of an end-to-end monitoring. Complete monitoring helps users to understand the overall application performance. The visibility across the application stack further improves collaboration and efficiency. Additionally, this helps enterprises quickly identify issues or complications within their infrastructure and software.

### **Business Agility:**

DevOps helps organizations to classify silos and focus on feedback and collaboration to move promptly across multiple developmental stages. Daniel Newman, principal analyst at Futurum Research and CEO of Broadsuite Media, [states that](#) large organizations use DevOps to minimize the fat when it comes to organizational responsiveness. Moreover, DevOps helps organizations of all sizes to create shorter iterative processes, which enable them to innovate and handle issues faster.

### **Work Routine Efficiency:**

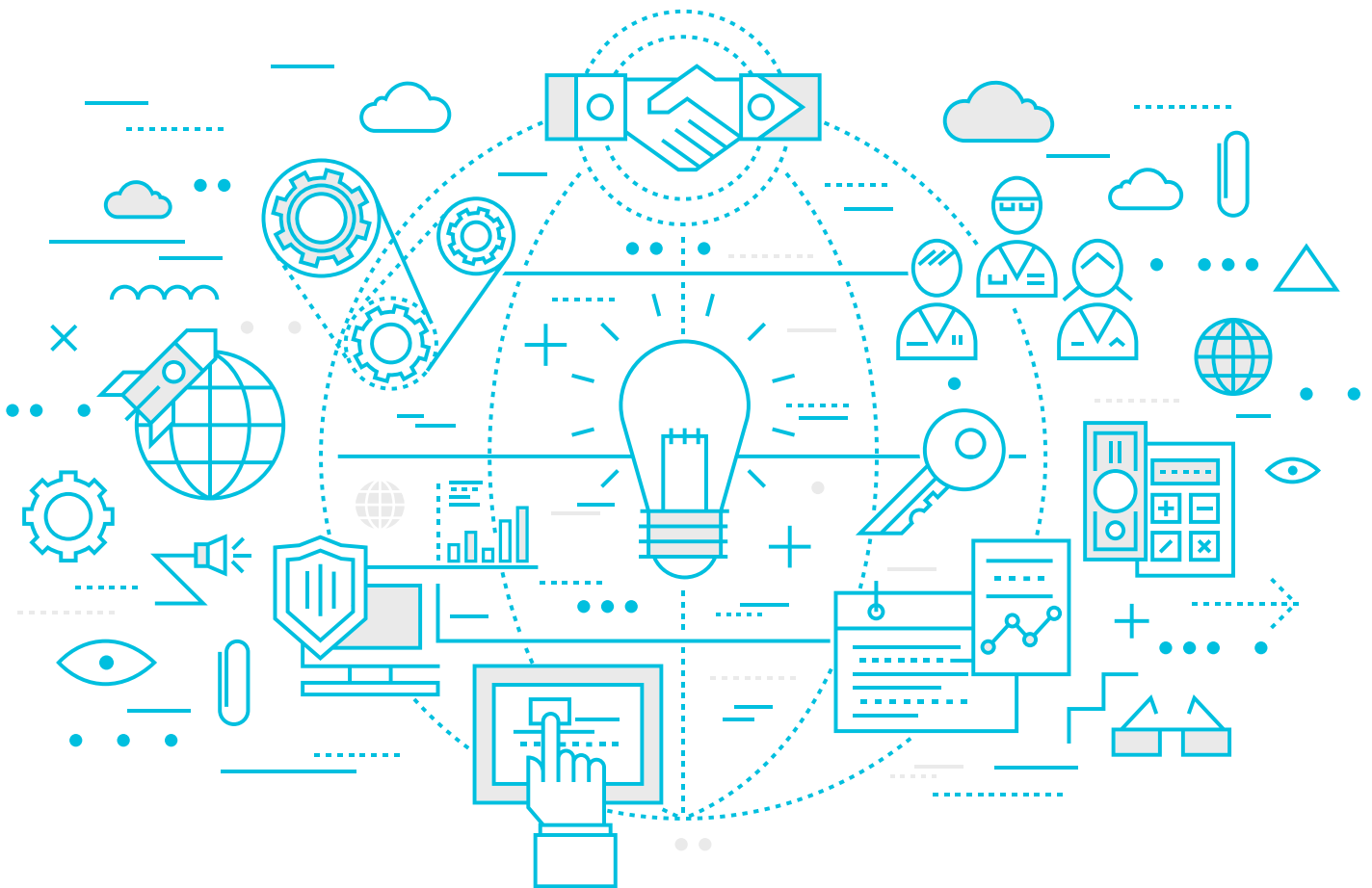
A typical IT environment witnesses enormous waste in resources from waiting and solving the same problem repeatedly. It thus becomes imperative for organizations to eliminate unwanted repetitions and utilize their effort in adding value to their organization. DevOps facilitates automation of deployments and standardization of production environments. Implementing the DevOps model creates a predictable and controllable deployment, liberating teams from mundane tasks and increasing their effectiveness at work.

**Automation by Orchestration:**

DevOps provides comprehensive control and coordination in automation covering the entire hierarchy in the infrastructure. In other words, orchestration is a specialized manner of automation. Users can carry out orchestration using tools such as Chef, Ansible, and Puppet. These tools come with their own distinctly defined standards and are compatible with leading cloud providers.

**Deliver Prompt Deployment:**

DevOps facilitates the solving of infrastructural issues with the latest tools. It does this by creating custom logic and its writing capabilities. Furthermore, DevOps assists in automating the entire process using single-click build tools. These tools are compatible with the cloud services and make deployment more efficient.



## 04 DevOps Best Practices for Your Business Growth

Users can make the most out of the numerous advantages of DevOps by implementing DevOps best practices. These best practices aim to ensure that organizations use a software development methodology that is agile, promoting coordination between development and operations teams, and resulting in a better end-user experience.

### **Dismantling IT Silos:**

IT organizations have been traditionally segmented into disciplinary silos. In such setups, software development follows an assembly line approach, with one department doing only one set of task. Such separation limits active collaboration and leads to delay in cooperative tasks and delivery. Therefore, it is imperative for the top-level management of an organization to break down the functional silos between teams.

### **Implementing Automated Testing:**

One of the most crucial aspects of DevOps is to implement an automated form of testing. Users should combine Test automation with Test-Driven Development (TDD) and Behavior-Driven Development (BDD) for optimal results. Automation is achievable by identifying scenarios and test cases, choosing an appropriate automation tool, establishing the test environment, running test cases, and ultimately analyzing results. Agile teams conduct automated testing teams in multiple batches, in which teams fix glitches as soon as they are spotted.

### **Using Integrated Configuration Management:**

To maximize the benefits of DevOps, organizations should implement an integrated configuration management approach to software development. By doing so, development teams apply configuration management at the solution level while not forgetting about production configuration issues among the organization and the solution. In a DevOps environment, it is imperative for developers to be enterprise-aware and have the larger picture in mind. Integrated configuration management enables operations teams to comprehend the potential impact of a new release, making it simpler plan the release of a product.

### **Deploying Continuous Integration:**

The DevOps deployment process is an amalgamation of several sub-processes, which includes versioning, code development, testing, deployment, and multiple post-deployment tasks. DevOps aims at automating these tasks using DevOps tools such as Bamboo and Jenkins to reduce manual intervention. Organizations must integrate changes continuously rather than periodically to ensure that they can make most out of DevOps.



### Ensuring Continuous Delivery:

Continuous delivery is a software development model where organizations create software in smaller batches so that it is possible to release the software to production at any time. [Martin Fowler](#) of ThoughtWorks further explains that teams can achieve continuous delivery when:

- o The software is deployable throughout its lifecycle.
- o The development team ensures that the software is deployable by working on new features.
- o Maintaining production readiness of the system despite making changes to it.
- o Teams can perform push-button deployments of any version of any environment on demand.

Teams can be in the state of continuous delivery by integrating the software, creating executables, and running automated tests to detect any issues. Organizations can benefit from continuous delivery through minimal deployment risks, trackable progress, and opportunity to receive user feedback.

### Dedicated Monitoring:

When it comes to DevOps, monitoring the infrastructure becomes even more important. Load time tracking, query logs, and other crucial details can prove to be helpful in optimizing application performance. A bug from a deployment can be disastrous for the entire application and needs application checks. The concerned teams can make use of online tools such as Geckoboard to ensure proactive monitoring. It combines several metrics related to infrastructure, analytics, application, sales, and graphs for numerous tools. An important aspect that monitoring teams must keep in mind is to set up relevant alerts while monitoring to enable the prompt management of events such as unresponsive applications.

### Data-driven Approach:

Teams involved in the DevOps process need to ensure that their prime focus is on performance. Analysis of factual data can prove to be the deciding factor in achieving this. Team members must also diligently share application graphs, usage patterns, and any other relevant data with other team members to create a sense of unity in direction and clarity in targets. Furthermore, the inclusion of testing, scalability, and deployment will help further simplify the entire development process. It is imperative for the development and operations team to work in unison to create and improve application performance, utilizing a repeatable, cohesive, and continuously refined approach.

### Team Culture:

DevOps is fundamentally a different way of functioning and working, and is much more complicated than simply implementing new tools. [Gartner's prediction](#) further reiterates the importance of a robust and a dynamic team composition in DevOps. The report predicts that by 2018, 90% of I&O organizations attempting to utilize DevOps without categorically addressing their cultural foundations will fail.

## 05 Trending DevOps Tools

The basis of DevOps is the usability and functionality of two essential features and their respective tools. DevOps' success significantly depends on the use of containers and the automation process along with its tools. This white paper will discuss how businesses can make the most out of DevOps with containers and automation.

### 5.1 Container:

Containers refer to solutions that help in running the application or software when moving across multiple computing environments with the same kernel type. This movement could vary from staging to the production environment, or from physical machine towards a virtual machine in a cloud. Furthermore, containers help in minimizing conflicts among teams functioning on different software on the same infrastructure by isolating the software from its surroundings. Each containerized application runs on a single operating system and shares kernel with other containers, making it lightweight. Along with this, containers also enable microservices; an architecture popularly used when it comes to DevOps. Microservices refer to disassembled small parts of an application that replace the large monolithic application.



#### Benefits of Container Services:

##### **Facilitates Business Agility:**

Through containers, development and deployment processes become much simpler for users. This enables organizations to launch new software and applications at a much faster rate, increasing the agility of the organization.

##### **Accelerates Development:**

Containers allow users to dismantle an application into small and discrete elements called microservices. This minimizes the cycle time between creation, testing, deployment, and integration.

##### **Optimize Cloud Utilization:**

With organizations adopting cloud technology more than ever before, containers help in utilizing cloud services to the fullest by enhancing the cloud environment to deploy several containerized applications onto a single cloud instance.

##### **Promotes Service Oriented Architecture:**

Containerization ensures every individual container runs a single application or process. By doing users can enhance the entire development model's quality

### Lightweight and Easy to Use:

Containers are extremely lightweight. This feature enables managing application components with ease. Moreover, it is possible to deploy them on multiple OS and VM.

One of the most popular container services used by developers across industries is Kubernetes. Below is a detailed explanation of Kubernetes, its features, and its use.

#### 5.1.1 Kubernetes

Kubernetes is an open-source system that helps developers to facilitate automated deployment, scalability, and the management of containerized applications. Additionally, it helps users to group containers that make up an application into logical units for ease of discovery and management. For organizations, Kubernetes provides the liberty to take advantage of infrastructures including private, public, or hybrid cloud as it is an open source system. Over time, organizations are finding Kubernetes as the most preferred container tool. A recent [survey](#) carried out by 451 Research further reiterates this by stating that 71% of the respondents, which included 200 large organizations across verticals, said that they were using Kubernetes.

Furthermore, Kubernetes' capability to function on public and private cloud adds to its popularity. In the same survey conducted by 451 Research, 57% of IT decision-makers shared that they prefer hosting containers on Containers as a Service (CaaS) and on-premise CaaS.

##### 5.1.1.2 Features

To reiterate, a major reason for Kubernetes being so popular is its heavy support for a wide variety of features, such as:

#### Automatic Binpacking:

Kubernetes allows users to automatically place containers based on the requirement and limitation of the resource. It achieves this without compromising the availability of the resource. Additionally, users can mix best effort workloads with critical loads to enhance utilization and make the most of the resource.

#### Automated Rollbacks and Rollouts:

A feature that makes Kubernetes one of the most preferred container services is its ability to roll out changes to applications and its configuration regularly. It does this while simultaneously monitoring application health status to ensure it does not eliminate all the instances at one go. In case of an error, Kubernetes can undo the change.

**Self-Healing:**

In instances when containers fail, Kubernetes allows users to restart the container, reschedule, and replace it when the nodes are not responding. For containers that do not pass the user-defined health check, Kubernetes destroys them instantly, leaving no room for any errors or issues.

**Load Balancing and Service Discovery:**

Kubernetes allocates containers their own IP addresses and a single DNS name for a set of containers, thus ensuring smooth load balancing across them. Furthermore, users need not worry about modifying their application to use an unaccustomed service discovery mechanism.

**5.1.1.3 Alibaba Cloud and Kubernetes**

To facilitate the use of Kubernetes to manage container applications in Alibaba Cloud, Alibaba Cloud Container Service provides support for Kubernetes clusters. Using the application deployment capability of Alibaba Cloud Resource Orchestration Service (ROS), users can create a highly available and secure Kubernetes cluster with one click by using ROS templates.

The Kubernetes cluster consolidates Alibaba Cloud's storage, network, virtualization, and security capabilities to provide a high-performance application management that simplifies cluster creation and expansion.

Kubernetes deployed on Alibaba Cloud facilitates deployment, expansion, and management of containerized applications. It further focuses on containerized management and application development, and comes with the following features:



Elastic expansion and self-reparation



Service discovery and server load balancing



Service publication and rollback



Secrets and configuration management

#### 5.1.1.4 Business Scenario

##### **Requirement:**

A leisure and entertainment company based out of UAE wants to create a destination centric web and mobile application capable of functioning under the OTA model. Furthermore, it seeks to enhance its travel platform's ecosystem for product and sales management, revenue management, inventory management, content management, and other capabilities for both B2C and B2B.

##### **Challenges:**

##### **Requirement of a Completely-Managed Support Implementation:**

Like all organizations, it is imperative for the organization to ensure that the support system would be accessible across all of the customer interactions.

##### **Avoiding Infrastructure Lock-In:**

The organization's primary concern was if their work would end up locking them up in their existing environment. Such a possibility exists even with an infrastructure built on standard tools and frameworks.

##### **Need for a PaaS-like Solution:**

A major requirement for the organization is a reliable model for running applications without the hassle of being responsible for the hardware and software infrastructure maintenance. Adoption of PaaS solutions ensures simplicity, scalability, and reliability.

##### **Multi-Region Support for Backup:**

The organization wants to eliminate the risk of a single point of failure through data replication available across geographies.

##### **Rolling Upgrades:**

The organization wanted applications to be available 24x7, which requires developers to deploy new versions of the applications several times a day.

##### **Running Containerized Application:**

Running containerized application is an extremely volatile and complex process. To tackle this, the organization requires a tool that could smoothly run containerized workflows as they shift from development to production.

### Why Kubernetes Is the Preferred Tool

Kubernetes fulfills all the organization's requirements for building large clusters that run complex applications. The use of Kubernetes would also ensure the avoidance of infrastructure lock-in by its support for scalability and rolling upgrades. Additionally, it provides users a framework to establish the rules of interaction between applications.

Kubernetes also offers a uniform platform for application deployment relieving the organization from the burden of taking care of these tasks manually. Moreover, the unit of work presented to the user is at the "service" level. Kubernetes equips users with the ability to gather detailed data over time and use that data to spot trends that help in indicating if the system is moving towards a failure.

Furthermore, it helps users define demarcation lines for moving into production. This clear distinction ensures disaster recovery for the users and business continuity for the organization. Kubernetes also provides seamless update and upgrade of the development solution along with multiple functional improvements on top of the clustered infrastructure.

### Solution Architecture

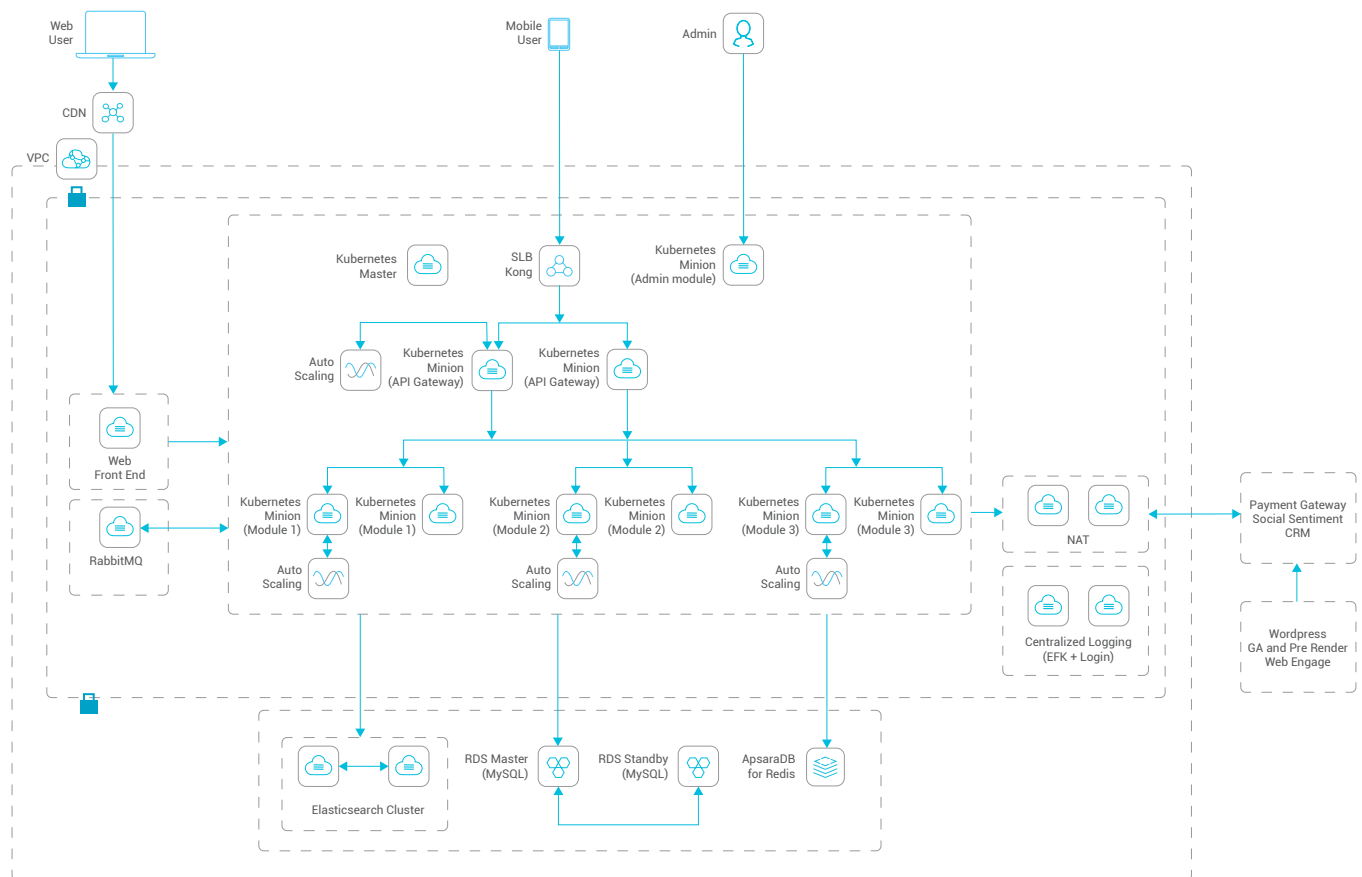


Figure 1

### Architecture Description:

Alibaba Cloud implemented a public facing Server Load Balancer that automatically distributed incoming HTTP requests to Elastic Compute Service (ECS) web servers running Kubernetes pods. This is possible using shared storage/network, and a specification for running the containers.

Additionally, Server Load Balancer helped minimize the response time and enhance the user experience through its effective routing of requests. The web/app servers deployed on ECS instances handled business logic effectively. This process delivered a flexible architecture conducive for a startup as ECS works as per real-time demands of an application.

ApsaraDB for RDS helped to store automatic data backups to avoid redundancy and segregated the data to achieve scalability. This helped in saving time and significantly optimizing resource usage. ApsaraDB for RDS was integral in ensuring data security and high level of built-in security, including resource and access management at a network level.

Furthermore, Content Delivery Network (CDN) delivered static content to a global network of edge locations, resulting in access to users across the globe to MewMe platform with better user experience. It also helped the organization to effectively shorten their website response time to milliseconds, ensuring smooth image browsing, and handling large volumes of traffic smoothly. Alibaba Cloud's reliability, availability, and simplified maintenance helped the organization to reduce infrastructure costs significantly, allowing them to focus more on application development, innovation, and user experience.

### Products Used:

Auto Scaling, Elastic Compute Service (ECS), Server Load Balancer, Virtual Private Cloud (VPC), ApsaraDB for RDS (Relational Database System), ApsaraDB for Redis, CDN (Content Delivery Network).

## 5.2 Automation Using DevOps

With Continuous Integration and Continuous Deployment being core features of DevOps, it is clear that automation is a significant contributor to the entire DevOps model. Automation aims to not only enhance the software development mechanism but also fill in the loopholes created by manual efforts in the software development model. Organizations can adopt automation to tackle frequent regression testing iterations and seek to pace up the delivery process.

Additionally, developers will find automation a blessing when working on microservices architecture or when working on exceptionally large projects.



### Benefits of Automation:

Automation has transformed the manner in which developers function, deliver, and go about things. The benefits of automation can help developers understand the impact and significance of the entire mechanism:



**Paced up Releases:**

A deployment process carried out by an automated mechanism witnesses minimal overhead. This makes it possible for organizations to easily repeat such processes, allowing developers to pace up their releases. Frequent releases subsequently promote a more agile software development mechanism.

**Deployments with Minimal Errors:**

As manual deployments are based on human judgment and calculation, there is always a possibility of human error. During a manual deployment, developers may overlook a crucial step or a critical fault. In fully automated deployments, such possibilities do not exist because the deployments are independent of human intervention. Upon configuration, the deployment process remains constant for every time the developer initiates a release, eliminating any possibility of errors.

**Reduced Development Time:**

Manual deployment process is a time-consuming task. It requires a developer to invest long hours, which could be spent on tasks that are more meaningful. When it comes to automated deployments, the entire process occurs in an extremely short span of time. Validation occurs behind the scenes and does not require the engagement of the developer. This minimizes the time for development, allowing developers to do more impactful tasks.

**Ease in New Releases:**

Automated deployments allow target environment and machines to change easily, despite the underlying release being permanent. In situations where teams need to create new installations, automated deployment ensures that the overhead involved is at a bare minimum. It only requires configuring existing setup and allowing automation to do the rest.

In this white paper, we will be discussing two automated tools – Terraform and Packer.

### 5.2.1. Terraform

Terraform is an open source tool that organizes APIs into analytical configuration files. These files are sharable among different DevOps team members and can be treated as code, modified, reviewed, and versioned. Additionally, it facilitates creation, edition, and versioning infrastructure efficiently and safely. Provided by [HashiCorp](#), Terraform helps users create a solution for codifying the creation of the complete data center. It generates an execution plan that outlines the plan to obtain the desired state. Furthermore, it can determine the changes made and create applicable incremental execution plans.

With the ability to manage low-level components such as storage, compute instances, and networking, Terraform has its own distinct language called HashiCorp Configuration Language (HCL). The language facilitates a balance between human readable and editable along with being machine-friendly.



### 5.2.1.1 Features of Terraform

The automation DevOps tools come with several features that help in managing popular and existing service providers along with custom in-house solutions. Its features make Terraform to compose and combine several service providers' resources in an agnostic and declarative manner. Below is a brief description of its features:

#### **Infrastructure as Code (IaC):**

It refers to using a high-level configuration syntax to provision infrastructure instead of defining physical hardware configurations. Using such syntaxes enable the versioning of the data center and allows its creation to proceed as that of any other code. Terraform also lets users to reuse and share the infrastructure over time and with other team members.

#### **Execution Plans:**

A significant feature of Terraform is the inclusion of a "planning" step. This step generates an execution plan that reveals to the user Terraform's action when users apply it. Such revelations ensure that the user is aware of the manipulation of the infrastructure by Terraform.

#### **Resource Graph:**

Terraform creates a graph for all the resources and freezes the creation and edition of any non-dependent resources. This results in the creation of an efficient infrastructure that allows users to receive insights into the dependencies in their infrastructure.

#### **Facilitate Automated Change:**


Terraform supports the application of complex change sets on the infrastructure keeping human interaction to the minimum. With distinct execution plans and resource graphs, users can track the change implemented by Terraform and the order of change. This automated mechanism of changes requires no human intervention and lowers the possibility of human error.

### 5.2.1.2 Business Scenario

#### **Requirement:**

A leading direct broadcast satellite television provider has set up a new environment for load testing. However, it does not want to waste resources and time by doing it manually. They want an automated tool that would help them to provide all the resources.

**Challenge:**

-  Facilitate integration with cloud providers
-  Automate creation of new environment
-  Provision resources automatically

**Why Terraform is the Preferred Tool**

A major challenge for the organization was to set up the environment for loading without spending too much time on it. Without Terraform, this would have been a manual and time-consuming process. Furthermore, they would have to provision resources whenever they had to create a new environment. The organization did not want to use a tool that was complex to implement and would require an extensive skill set and knowledge.

The developers in the organization decided to opt for Terraform as it offered them the feature of provisioning all the resources with great ease and convenience. Developers were also able to destroy the provisioned resources when they did not require the resource to exist.

**Solution Architecture**

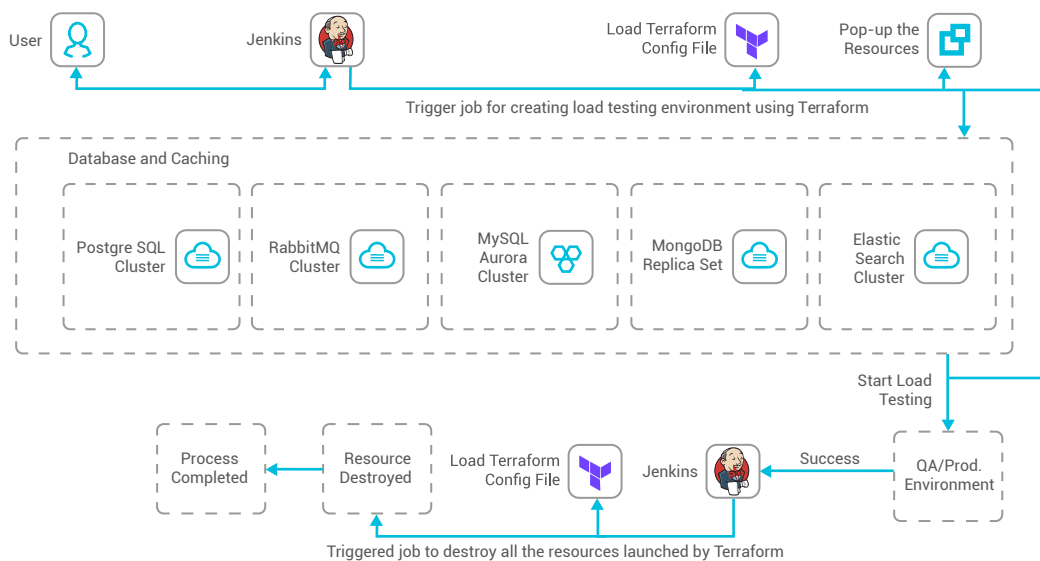


Figure 2

### Architecture Description:

To load the environments, developers used PostgreSQL cluster, RabbitMQ cluster, MongoDB replica, and Elasticsearch cluster whenever load testing required these resources. As such, the developers deployed Terraform ensure proper provisioning of these resources. To do this, the developers created a Terraform template for each resource, a process followed by running the resources. On the completion of running these resources, the developers ran chef recipes to ensure management of the configuration.

### Products Used:

PostgreSQL, RabbitMQ, MongoDB and Elasticsearch

## 5.2.2 Packer

For organizations that want a tool capable of making the automated creation of machine image possible, HashiCorp offers an automation tool called Packer. It does this by adopting modern configuration management by facilitating users to use automated scripts for installing and configuring the software within the Packer-made images. Packer stands out from other existing DevOps automation tools with its ability to catapult machine images into the modern age, opening multiple potentials and initiating new opportunities. It does this by promoting users to use a framework such as Chef or Puppet to install and configure the software within their Packer-made images.

### 5.2.2.1 Features of Packer

#### Prompt Infrastructure Deployment:

Packer images allows users to launch completely configured and provisioned machines in seconds, as compared to previous tools that would take several minutes or even hours. This benefits the production as well as the development team since Packer can launch development virtual machines instantly, without waiting for prolonged provisioning time.

#### Multi-provider Portability:

Packer creates identical images for multiple platforms capable of running on the private cloud and development in desktop virtualization solutions. Each environment runs an identical machine image, giving the user benefit of portability.

**Enhanced Stability:** Packer provides users with enhanced stability. It allows installation and configuration of all the software for a machine during image formation. Users can instantly trace and resolve any bugs that may appear.

#### Advanced Testability:

Packer facilitates superior testability by allowing prompt smoke testing once building the machine image is complete. A positive result in the testing would mean that any other machines image will function properly.

### 5.2.2.2 Business Scenario

**Requirement:** A leading organization specializing in providing customizable images requires an automated development model that ensured images are independent of the infrastructure.

After a lot of research, the DevOps manager decides to use Packer to cater to the organization's various needs. The pre-existing machine images were too tedious to create and manage. Packer, on the other hand, handled this with great ease and speed, facilitating agility of operations and the business.

#### Why Packer is the Preferred Tool

Packer is an easy to use automation DevOps tool that automates the creation of any machine image. It adopts modern configuration management methods by encouraging users to deploy frameworks such as Chef or Puppet for installing and configuring the software within the organization's Packer-made images. The organization in discussion used Packer to create multiple images at the same time with the same configuration. This helped their customers to possess images of the same configuration in all environments.

The Packer deployment initiates by creating an instance and then follows it up by configuring it 'live' via SSH. Contrary to pre-existing tools, Packer does not need to unpack and repack the system image.

#### Solution Architecture

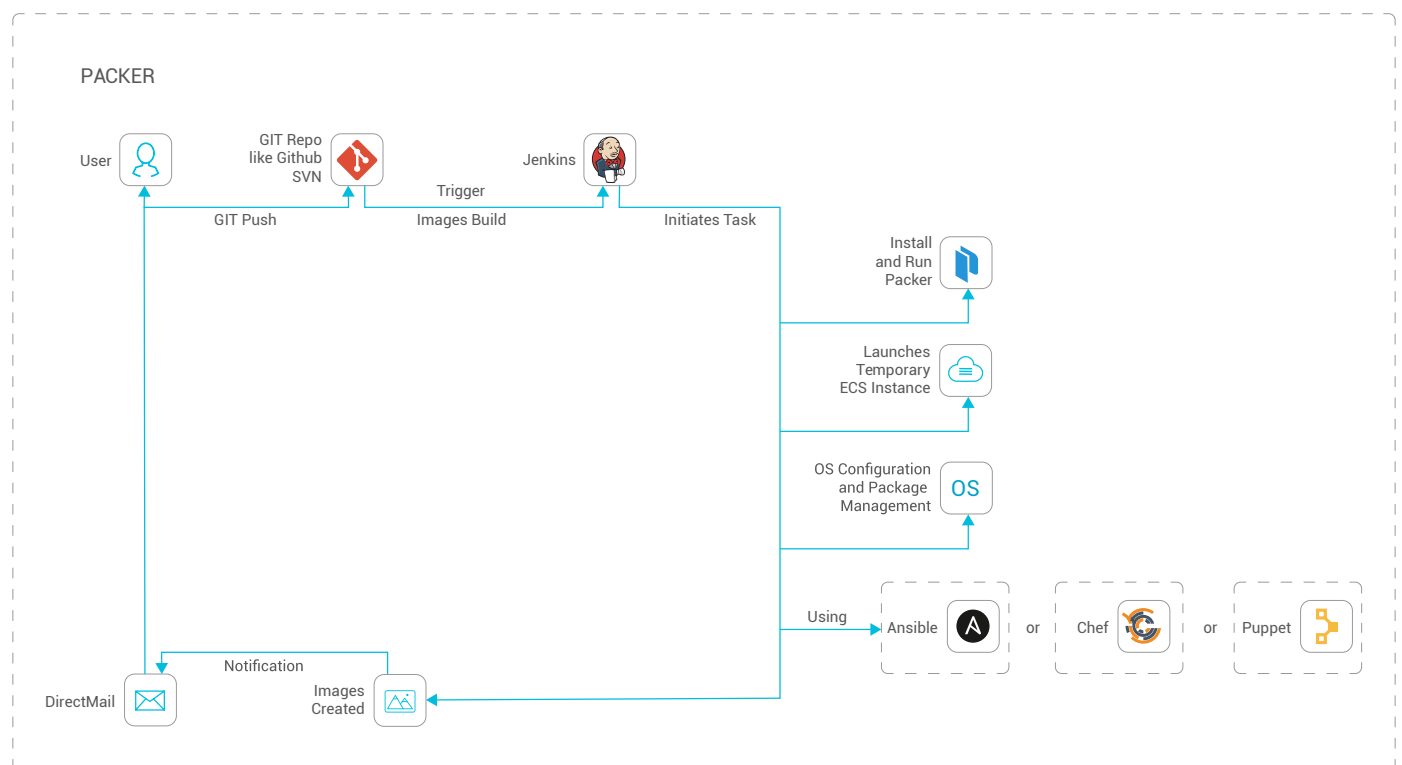


Figure 3

**Solution Description:**

To begin with, the system pushes the Packer file in the Git Repository. Git then triggers Jenkins, which is deployed on ECS instances or an Image Build task. Jenkins is used to test a code and build it. Jenkins then runs Packer, which in turn launches an ECS instance using the base image provided in the packer config file. It then installs any packages as per the config file and creates images based on the configuration provided in the Packer config file. Finally, it notifies the user through direct mail service of Alibaba.

**5.2.2.3 Using Packer and Terraform with Alibaba Cloud**

Alibaba Cloud provides supports Packer and Terraform for core packaging and infrastructure provisioning. These tools allow users to swiftly deploy their infrastructure and application on Alibaba Cloud. Enterprise business's rapid iteration of applications and infrastructure along with continuous development ensure enhanced operations and minimize maintenance costs. Furthermore, Alibaba Cloud provides a set of flexible services designed to help customers to rapidly and reliably build and deliver products using Alibaba Cloud and DevOps practices.

With the support of Terraform and Packer, Alibaba Cloud customers can possess impactful workflows to manage their global infrastructure. Subsequently, users can save time and focus on delivering business-critical needs.

Packer users can easily build and configure customized images on Alibaba Cloud using the same workflow and configuration as used for managing images on other platforms.

Similarly, Terraform users can provision compute, network, and storage resources on Alibaba Cloud utilizing similar workflow and configuration as they would, when managing infrastructure on other clouds.

## 06 Alibaba Cloud - Offering the Best of Cloud and DevOps

There are no doubts that businesses can benefit from the combination of cloud technology and DevOps. [Bernard Golden](#), one of cloud computing's pre-eminent thought leaders, believes that Agile development and DevOps both require immediate infrastructure availability, and the cloud caters to this in the most quintessential form. Hosting DevOps on the cloud can help organizations to evolve from a reactive approach to a more proactive approach.

To cater for the increasing demand for automation in infrastructure provisioning, many cloud providers provide dedicated services meant to help the DevOps team to spawn an entirely new environment, which can be the replica of the existing one. With DevOps on the cloud, organizations can have the assurance of achieving delivery through continuous integration, scalability, testing, and deployment.

Below are ways in which one can utilize DevOps optimally on the cloud.

### **Increased Scalability:**

Organizations opt to use cloud computing with their DevOps model because of the scalability offered by cloud. Increased scalability facilitates increased capacity with a click of a button. Coupling it with DevOps, the applications automatically integrate scalability within themselves. Moreover, it achieves all this while minimizing the overall infrastructure cost.

### **Efficient Recovery Mechanisms:**

When it comes to cloud, backup restoration is much easier as compared to data centers. Cloud providers expose their command line API, which helps users to perform several tasks in an automated manner. Additionally, users can also integrate these scripts with continuous integration tools. Using command line, users can configure automated backups, automate restore of the most recent backup, and delete old backups.

### **Eliminate Downtime:**

With cloud-based continuous operations, users can eliminate downtime of their system. The implementation of an automated development model will allow developers to create stateless applications. These applications help in enhancing business reliability and customer satisfaction.

With several success stories of transforming the way organizations function, develop, and deliver, Alibaba Cloud is the one stop solution for organizations to evolve towards the DevOps model. Learn more about Alibaba Cloud by visiting [www.alibabacloud.com](http://www.alibabacloud.com).

## References

1. <https://www.atlassian.com/devops>
2. <https://www.mongodb.com/what-is-devops>
3. [http://www.webopedia.com/TERM/D/devops\\_development\\_operations.html](http://www.webopedia.com/TERM/D/devops_development_operations.html)
4. <http://www.tothenew.com/devops-automation-consulting>
5. <http://www.tothenew.com/blog/10-mistakes-to-avoid-while-implementing-devops/>
6. <http://www.tothenew.com/blog/6-barriers-to-devops-and-how-to-avoid-them/>
7. <http://www.tothenew.com/blog/10-key-devops-practices-to-improve-it-efficiency/>
8. <https://eternalsunshineoftheismind.wordpress.com/2013/03/10/problems-with-traditional-methods-of-software-development/>
9. <https://techbeacon.com/devops-automation-4-simple-steps-prioritizing>
10. <https://dzone.com/articles/what-is-devops-and-how-automation-helps-achieve-itdevops>
11. <https://www.alibabacloud.com/help/doc-detail/53751.htm>
12. <https://www.packer.io/docs/commands/index.html>
13. <https://dzone.com/articles/microservices-with-kubernetes-and-docker>
14. <https://kubernetes.io/docs/concepts/overview/what-is-kubernetes/>
15. <http://www.infoworld.com/article/3118345/cloud-computing/why-kubernetes-is-winning-the-container-war.html>
16. <https://www.ubuntu.com/containers/kubernetes>
17. <https://www.ubuntu.com/containers/kubernetes>
18. <https://www.alibabacloud.com/forum/read-830>
19. <https://www.terraform.io/devops>
20. <https://www.packer.io/>
21. 2017-state-of-devops-report.pdfoperations.html
22. <https://newrelic.com/devops/benefits-of-devops>
23. <https://betanews.com/2016/11/25/devops-benefits/>
24. <https://www.forbes.com/sites/mikekavis/2014/12/18/11-common-devops-bottlenecks/#621dcd2a7737>
25. <https://www.forbes.com/sites/danielnewman/2017/04/18/agility-is-the-key-to-accelerating-digital-transformation/#1b8de5397277>
26. <https://www.futurum.xyz/devops-the-key-to-your-companys-success-in-2017/>
27. <https://www.rightscale.com/press-releases/rightscale-2017-state-of-the-cloud-report-uncovers-cloud-adoption-trends>
28. <https://dzone.com/articles/what-is-continuous-integration>
29. <http://www.drdoobs.com/architecture-and-design/top-10-practices-for-effective-devops/240149363>
30. <https://martinfowler.com/bliki/ContinuousDelivery.html>
31. <https://www.digitalocean.com/community/tutorials/an-introduction-to-kubernetes#what-is-kubernetes>
32. <http://www.zdnet.com/article/kubernetes-leads-container-orchestration/>
33. hybrid-cloud-drives-growing-container-production-use-and-disruption-451-research-report.pdf
34. <http://www.infoworld.com/article/3173266/containers/4-reasons-you-should-use-kubernetes.html>
35. <https://www.infoq.com/news/2014/08/terraform>
36. <https://www.packer.io/intro/why.html>
37. <http://www.devopsdigest.com/devops-advantages-3>
38. <http://www.tothenew.com/blog/infographic-8-key-benefits-of-using-container-technology/>
39. <http://www.tothenew.com/blog/why-cios-should-adopt-containers-technology/>
40. <http://www.decisiondesign.com/approach/methodology-problem/>
41. <https://dzone.com/articles/jez-humble-why-software>
42. <https://www.red-gate.com/blog/database-lifecycle-management/5-big-benefits-automated-deployment>
43. /Why%20CIOs%20Should%20Consider%20Using%20DevOps%20with%20Cloud.pdf
44. <https://insights.hpe.com/articles/why-cloud-makes-agile-and-devops-more-important-1701.html>
45. <https://www.forbes.com/sites/forbestechcouncil/2017/07/21/the-relationship-between-the-cloud-and-devops/2/#4783fa5c3428>

