

Building DevOps on Amazon Web Services (AWS)



Abstract

At its core, DevOps makes delivery of applications more efficient. Amazon Web Services (AWS) has the platform and services to recognize a code change and automate delivery of that change from development, through the support environments, to production. However, delivery of code is just one aspect of DevOps.

IBM extends the DevOps definition, making it an enterprise capability that enables organizations to seize market opportunities and reduce time to customer feedback. IBM's main objectives are speeding continuous innovation of ideas, enabling continuous delivery of those innovations, and providing meaningful feedback for continuous learning, thereby putting all the emphasis on deciding what code to change.

IBM extends DevOps to include all stakeholders in an organization who develop, operate or benefit from businesses systems. DevOps enables design thinking, which focuses on user outcomes, restless reinvention, and empowering teams to act. In addition, DevOps enables lean and agile methodologies, which guide teams to deliver in smaller increments and get early feedback. These approaches improve the content and quality of the changes in the application delivery lifecycle.

IBM provides an engineering approach to implementing DevOps on AWS on an existing portfolio of applications. Through a discovery workshop, we will analyze your application delivery lifecycle, identify areas for improvement, and then execute proof points on preselected applications. Based on those proof points, we will help you learn and move forward to onboard new applications, while also monitoring and measuring impact.

The IBM point of view

This paper will provide IBM's point of view on DevOps, and how deploying on cloud can help you make the most of it. It will discuss practical approaches while focusing on AWS cloud offerings. It is the latest in our series of papers highlighting the partnership between IBM and AWS to help our joint customers achieve cloud success.

IBM defines **DevOps** as an enterprise capability that enables organizations to seize market opportunities and reduce time to customer feedback, and has three main business objectives:

1. Speeding **continuous innovation** of ideas by enabling collaborative development and testing across the value chain
2. Enabling **continuous delivery** of these innovations by automating software delivery processes and eliminating waste, while also helping to meet regulatory concerns
3. Providing a feedback loop for **continuous learning** from customers by monitoring and optimizing software-driven innovation

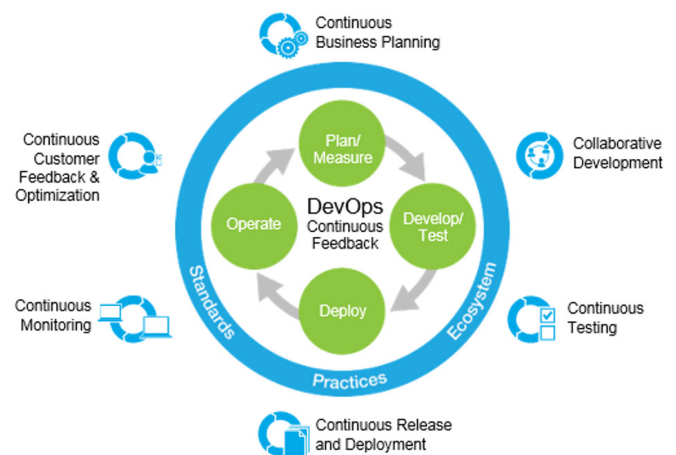


Figure 1. The DevOps continuous feedback loop

DevOps enables process and technology

Process

DevOps works with agile, lean, and design thinking to drive the loop of continuous delivery, feedback, and innovation. As we will see when discussing technology, AWS' rapid deployment and data collection feed and improve this cycle.

	Speeding continuous innovation of ideas	Enabling continuous delivery of these innovations	Providing a feedback loop for continuous learning
Agile <ul style="list-style-type: none"> Fast feedback cycles through early customer involvement 	✓	✓	
Lean <ul style="list-style-type: none"> Value stream mapping Eliminate waste 	✓		✓
Design thinking <ul style="list-style-type: none"> Focus on delivering a delightful user experience 	✓		✓
Cloud operation <ul style="list-style-type: none"> Quick and flexible management of development, test and production environments Resilient and scalable 		✓	
Automation <ul style="list-style-type: none"> Removing the silos between development and IT operations Treat infrastructure as code Continuous delivery of changes 		✓	
Application Analytics <ul style="list-style-type: none"> Real-time insight on problems in production Insight on application usage 			✓

Table 1. The role of process and technology in DevOps

“Lean and agile thinking guides teams to deliver in smaller increments and get early feedback. As a result, teams reduce cycle time by focusing only on those activities that maximize value based on feedback. Wasted effort is identified and eliminated, enabling teams to spend time on value-add activities, such as innovation and quality improvements.”

– Agile for Dummies, 2nd IBM Limited Edition, ibm.biz

IBM’s DevOps approach applies these thinking principles to all stakeholders in an organization that develops, operates or benefits from the business’ software systems, including customers, suppliers, and partners. By extending lean principles across the entire software supply chain, DevOps capabilities can improve productivity through accelerated customer feedback cycles, unified measurements and collaboration across an enterprise, and reduced overhead, duplication and rework.

Design thinking principles include:

- Focus on user outcomes, and drive business by helping customers achieve their goals
- Restless reinvention: stay essential by treating everything as a prototype
- Move faster by empowering diverse teams to be proactive

Design thinking provides a complementary set of principles and practices that fits very well with a DevOps approach. In the traditional model, developers are often the furthest removed from the customers. Design thinking reverses this by allowing developers to respond directly to customer feedback. A DevOps team that applies design thinking will focus on achieving their customers’ goals, delivering a quickly expanding minimum viable product based on customer feedback, and empowering team members to fail until they succeed.

Technology

Cloud operations

AWS operates in a **shared responsibility model** with the customers for security, compliance and IT controls. For infrastructure as a service (IaaS) services like Amazon EC2, AWS operates, manages and controls the components from the host operating system and virtualization layer down to the physical security of the facilities in which the service operates. The customer assumes responsibility and management of the guest operating system (including updates and security patches), other associated application software, as well as the configuration of the AWS-provided security group firewall as shown in Figure 2.

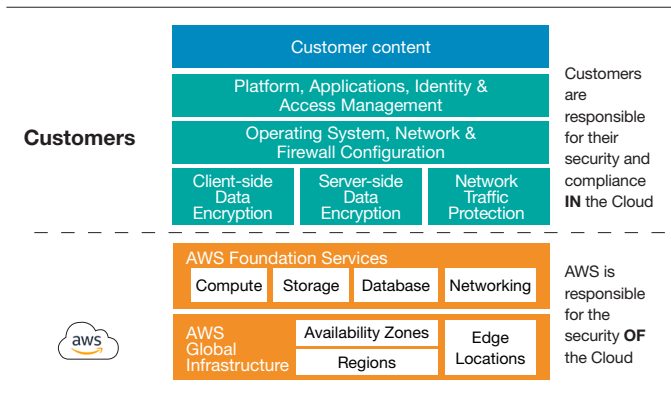


Figure 2. AWS shared responsibility model

AWS manages the infrastructure and, depending on the service, may also manage further up the technology stack. Figure 3 shows customer responsibilities for different types of services. AWS provides on-demand delivery of compute, storage, databases, networking and other IT resources. You retain control of how you choose to manage and use these resources.

Infrastructure Services	Container Services	Abstract Services
Data	Data	Data
Customer IAM	Customer IAM	AWS IAM
AWS IAM	AWS IAM	
Applications	Firewall	
Operating System		
Networking/Firewall		

Figure 3. Customer responsibilities vary with type of services used

AWS provides a white paper describing the five pillars of a **well-architected framework**—security, reliability, performance efficiency, cost optimization, and operational excellence—and a set of best practices that align to those pillars. When designing your virtual data center, building and deploying applications in that data center, and managing those applications, you need to be cognizant of AWS’ best practices. To read the white paper, visit <https://aws.amazon.com/whitepapers>.

Properly leveraging AWS and third-party services can automate cloud operations, as well as application development lifecycle management. Support environments can be built as needed, and just as easily released. The number of code lines is no longer limited by the number of support environments.

Managing your virtual data center 24/7/365 and maintaining an experienced staff trained in the latest tools around operations requires an investment for work that does not necessarily provide a competitive difference. You can look to managed service providers to perform this work while your staff focuses on the portion of operations that adds value to your business.

Automation

Continuous integration is a DevOps practice where developers continuously commit their code changes into a source repository. Then, at regular intervals, the system will conduct an automated build, deploy, and test. **Continuous delivery** expands on continuous integration by automatically deploying code changes through the support environments, pausing for approval before going to production. **Continuous deployment** does not wait for approval; rather, it goes to production automatically after successfully passing through automated testing in the support environments.

You can use automation technology to build a target support environment that includes application, database and test servers, install and configure middleware and applications, and then execute automated testing. When the testing is completed, the environment can be released. AWS enables this level of automation through its tool sets and pay-as-you-go pricing model.

AWS also offers a set of code services that provide tools to developers to implement automation. Figure 3 shows these services and how they work together.

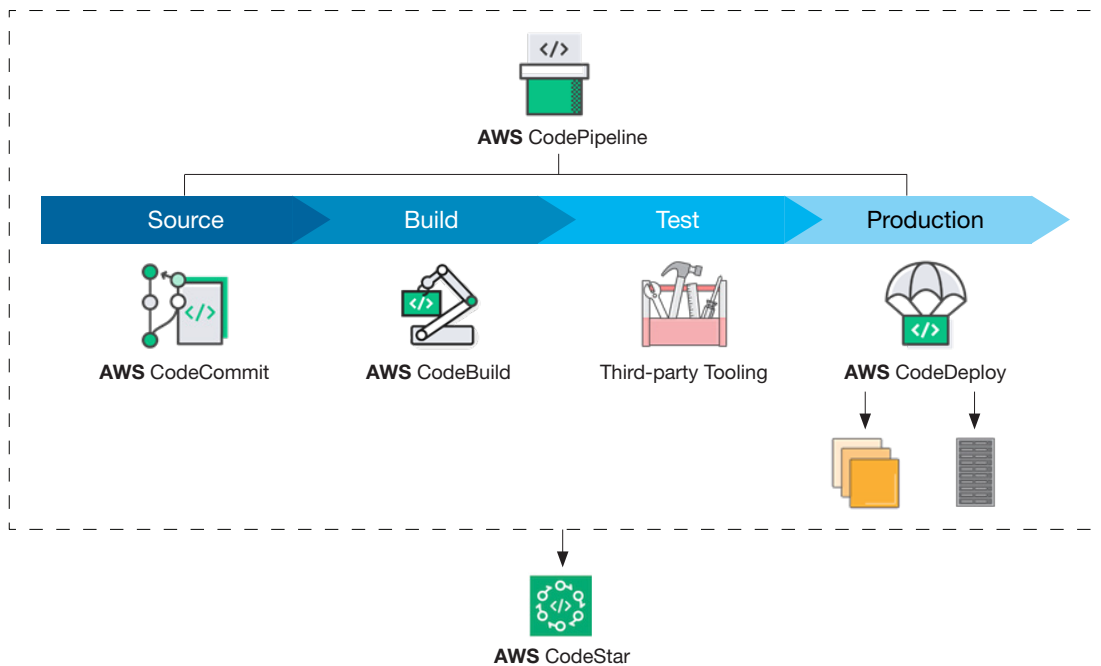


Figure 4. AWS code services

AWS CodeCommit is a fully-managed source control service that makes it easy for companies to host secure and highly scalable private Git repositories. **AWS CodeBuild** is a fully managed build service that compiles source code, runs tests, and produces software packages that are ready to deploy. **AWS CodeDeploy** is a service that automates code deployments to any instance, including Amazon EC2 instances and servers running on-premises. **AWS CodePipeline** is a continuous integration and continuous delivery service for fast and reliable application and infrastructure updates. **AWS CodePipeline** builds, tests and deploys your code every time there is a code change, based on the release process models you define. **AWS CodeStar** enables you to quickly develop, build and deploy applications on AWS. **AWS CodeStar** provides a unified user interface, enabling you to easily manage your software development activities in one place. **AWS X-Ray** is a distributed tracing service that helps developers analyze and debug distributed applications, and understand how their application and its underlying services are performing to identify and troubleshoot the root cause of performance issues and errors.

Application analytics

Understanding how an application is being used is valuable to both the business and technology teams. **Amazon CloudWatch** is a monitoring service for cloud resources and applications that run on AWS. You can use **Amazon CloudWatch** to collect and track metrics, collect and monitor log files, and automatically react to changes. The service can also be used to gain system-wide visibility into resource utilization, application performance, and operational health.

Today's analytics capabilities can go far beyond traditional monitoring. For instance, data can be captured about how customers interact with applications. The section on continuous learning later in the paper will touch on some of the tools available to help you gain insight into how your customers are interacting with your applications.

Continuous innovation

Businesses are under tremendous pressure to create new value for their customers through innovation. However, they are finding that traditional approaches to software development and delivery are not sufficient to deliver the business innovation their customers expect. Manual development processes are error-prone, wasteful, and known to cause significant delays. Through proper application of new technology and the principles of continuous innovation, businesses can eliminate these manual tasks, and start delivering value like never before.

Continuous innovation means continuously developing new ideas into innovative software, which in turn, can continuously improve the value delivered to customers. IBM believes that DevOps is one of the primary means for achieving this sustained innovation.

In its conventional sense, DevOps refers to a closer collaboration between development and operations teams, and the integration of associated processes and tooling. In IBM's point of view, DevOps is much more than that. We believe that DevOps should encompass collaboration among all stakeholders—not just between development and operations, but also among lines of business, suppliers involved in software delivery, and customers themselves. In this expanded definition, DevOps includes business governance practices around security and compliance, and all aspects of the delivery process, such as multi-sourcing.

Continuous delivery

The main goal of DevOps is to make delivery more efficient. Support environment availability and configuration is a roadblock that often interferes with achieving this goal. It is important to ensure that the support environment matches the production environment, as a mismatch can introduce significant quality issues. Additionally, changes to complex systems—even when componentized—can have unexpected results. Requirements, written or verbal, can be misinterpreted. Automating functional and non-functional testing, along with early feedback by stakeholders, is critical to maintaining quality. Deploying DevOps on AWS can help address these problems.

Let us start with a working definition of Infrastructure as Code (IaC): the process of managing and provisioning computing infrastructure and its configuration through machine-processable definition files, rather than the use of interactive configuration tools.

AWS' machine-processable definition files are AWS CloudFormation templates. The templates access the same API as the AWS Console and the AWS Command Line Interface (CLI). The templates are JSON or YAML formatted text files that should be placed under normal source control. They are also parameterized, allowing the environments to differ in a controlled way. As an example, a dev environment could use a smaller Amazon EC2 instance (virtual server) than a performance or production environment.

Now, let us walk through the diagram in Figure 5 as an example of a code change being deployed to production. The environment being built on demand could be far more complex than in this example where there is a single application connecting to a data store. A developer is going to make a change to the application code.

The developer commits a change to the source control repository, AWS CodeCommit.

AWS CodePipeline detects the change, and rebuilds the components affected by the change.

AWS CodePipeline creates the dev environment using a parameterized AWS CloudFormation template. The template would also be stored in AWS CodeCommit.

AWS CodeDeploy is instructed by CodePipeline to install and configure the applications in the dev environment.

AWS CodePipeline triggers automation testing of the dev environment.

AWS CodePipeline repeats the process for the QA environment and then pauses for user acceptance testing.

After approval, AWS CodePipeline uses the parameterized CloudFormation templates to build the new production compute environment, and AWS CodeDeploy to deploy the applications.

AWS CodePipeline triggers a brief automation test, and if the environment passes, switches the DNS server to the new compute servers. This is a release technique called blue-green deployment.

The AWS services described in this section are designed to work with standard industry software. For example, AWS CodeCommit is a Git repository. The service at the core of AWS' Infrastructure-as-Code strategy is CloudFormation, which can use configuration management tools such as Chef, Puppet, or Ansible.

See the glossary at the end of this paper for a more detailed description of these services.

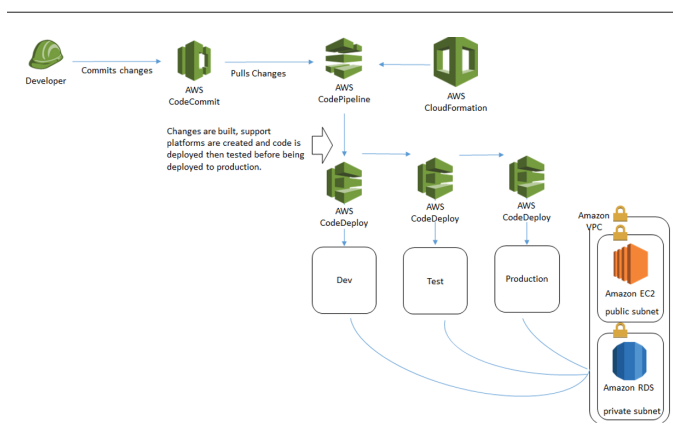


Figure 5. Deploying a code change in AWS

Continuous learning

As mentioned previously, understanding how an application is being used is valuable to both business and technology teams. In addition to Amazon CloudWatch, there are a variety of analytics solutions available to help you better understand how customers use your applications.

IBM Digital Analytics, formerly Coremetrics Web Analytics, is a platform for near real-time digital analytics, data monitoring, and comparative benchmarking. The solution allows you to track and analyze visitor behavior over time, across multiple touchpoints and channels, and deliver more personalized, relevant and effective information. It also allows you to optimize your web, mobile and social channels by monitoring critical data and key performance indicators in near real-time. With IBM Digital Analytics, you can uncover growth opportunities and areas for improvement.

IBM Tealeaf® is a family of products to improve visitor interactions with customer experience management solutions. For example, IBM Tealeaf CX provides visibility into web and mobile browsers, capturing data down to the individual session level, and analyzes the data to uncover trends and valuable insights. The solution can help you discover unexpected customer pathways through your applications, and areas where customers tend to struggle the most.

IBM's approach to DevOps in AWS

IBM adapted its methodology to provide an engineered approach to implementing DevOps on AWS.

Discovery workshop

During a discovery workshop, IBM will help your team perform an assessment of your application delivery lifecycle management (ADLM) through value stream mapping. This is a lean strategy for analyzing the current “as is” state and designing the future “to be” state of ADLM.

The mapping looks at the full lifecycle documenting each step, milestone, and gate. Data about the effort in man hours and duration, as well as value, is considered. In the end, the discovery workshop looks to address bottlenecks in your pipeline. Some common examples include:

- Replacing ticket-based environment provisioning with cloud-hosted self-service
- Replacing weekly hands-on deployments to an integrated environment with more automated daily deployments
- “Shift left” integration testing to match more frequent deployments, while focusing on increasing automation testing

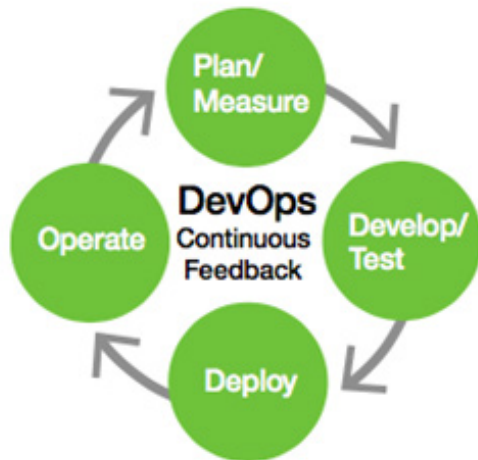


Figure 6. DevOps continuous feedback loop

In the discovery workshop, you will review your existing application portfolio for automation readiness, and then select applications that can provide quick wins.

Proof point

Following the discovery workshop, IBM will help you gather baseline as-is metrics for environment management and application code delivery. Then, we will execute proof points on select applications by testing proposed remediation of bottlenecks such as automation of environment creation and code delivery. Finally, we will help you capture DevOps-enabled metrics, and then analyze.

Optimize and expand

Just as we apply DevOps principles to application development, we will be applying them here as well. Our goal at this stage is to help you learn from the proof points, onboard the next set of applications, and continue to monitor and measure impact.

For more information

To learn more about IBM Cloud Migration Services, visit us at ibm.com/cloud-computing/services/cloud-migration, or contact your IBM representative

Glossary

AWS CloudFormation

AWS CloudFormation gives developers and systems administrators an easy way to create and manage a collection of related AWS resources, provisioning and updating them in an orderly and predictable fashion. You can use AWS CloudFormation's templates to describe the AWS resources, and any associated dependencies or runtime parameters, required to run your application. You can also visualize your templates as diagrams and edit them using a drag-and-drop interface with the AWS CloudFormation Designer.

This service is at the core of AWS' Infrastructure-as-Code strategy. When deploying AmazonEC2 instances, you can build a custom Amazon Machine Image (AMI), run user data scripts to build out the server, or use configuration management tools such as Chef, Puppet, or Ansible. These are third-party tools that integrate with AWS CloudFormation to help automate provisioning, managing, patching, and configuration management of the cloud infrastructure.

<https://aws.amazon.com/cloudformation/>

Amazon CloudWatch

Amazon CloudWatch is a monitoring service for AWS cloud resources and the applications you run on AWS. You can use Amazon CloudWatch to collect and track metrics, collect and monitor log files, set alarms, and automatically react to changes in your AWS resources. Amazon CloudWatch can monitor AWS resources such as Amazon EC2 instances, Amazon DynamoDB tables, and Amazon RDS DB instances, as well as custom metrics generated by your applications and services, and any log files your applications generate. You can use Amazon CloudWatch to gain system-wide visibility into resource utilization, application performance, and operational health. You can use these insights to react and keep your application running smoothly.

<https://aws.amazon.com/cloudwatch/>

AWS CodeCommit

AWS CodeCommit is a fully-managed source control service that makes it easy for companies to host secure and highly scalable private Git repositories.

<https://aws.amazon.com/codecommit/>

AWS CodeDeploy

AWS CodeDeploy is a service that automates application deployments to any instance, including Amazon EC2 instances and instances running on premises. AWS CodeDeploy allows you to launch and track the status of your application deployments through the AWS Console or CLI. AWS CodeDeploy is platform and language agnostic and works with any application. AWS CodeDeploy can also integrate with your existing software release process or continuous delivery toolchain (such as Jenkins).

<https://aws.amazon.com/codedeploy/>

AWS CodePipeline

AWS CodePipeline creates continuous delivery pipelines that track code changes from sources such as AWS CodeCommit, Amazon Simple Storage Service (S3), or GitHub. You can design your development workflow for checking in code, building the code, deploying your application into staging, testing it, and releasing it to production. You can use AWS CodePipeline as an end-to-end solution. With AWS CodePipeline, you can rapidly deliver features and updates with high quality through the automation of your build, test and release process. You can use AWS CodePipeline to automate the release of your Chef cookbooks and application code to AWS OpsWorks.

<http://docs.aws.amazon.com/opsworks/latest/userguide/other-services-cp.html>

<https://aws.amazon.com/codepipeline/>

AWS Config

AWS Config is a fully managed service that provides you with an AWS resource inventory, configuration history, and configuration change notifications to enable security and governance. AWS Config Rules enables you to create rules that automatically check the configuration of AWS resources recorded by AWS Config.

<https://aws.amazon.com/config/>

AWS Elastic Beanstalk

AWS Elastic Beanstalk is an easy-to-use service for deploying and scaling web applications and services developed with Java, .NET, PHP, Node.js, Python, Ruby, Go, and Docker on familiar servers such as Apache, Nginx, Passenger, and IIS.

You can simply upload your code and AWS Elastic Beanstalk automatically handles the deployment, from capacity

provisioning, load balancing, auto-scaling to application health monitoring. At the same time, you retain full control over the AWS resources powering your application, and can access the underlying resources at any time.

<https://aws.amazon.com/elasticbeanstalk/>

AWS OpsWorks

AWS OpsWorks is a configuration management service that helps you configure and operate applications of all shapes and sizes using Chef. You can define the application's architecture and the specification of each component, including package installation, software configuration and resources such as storage. Start from templates for common technologies like application servers and databases, or build your own to perform any task that can be scripted. AWS OpsWorks includes automation to scale your application based on time or load, and dynamic configuration to orchestrate changes as your environment scales.

<https://aws.amazon.com/opsworks/>

Blue-green deployment

Blue-green deployment is a DevOps deployment practice that uses domain name services (DNS) to make application deployments. The strategy involves starting with an existing (blue) environment while testing a new (green) one. When the new environment has passed all the necessary tests and is ready to go live, you simply redirect traffic from the old environment to the new one via DNS.

About the authors

Daniel (Dan) Carr

Dan is the Global AWS Practice Leader at IBM Global Business Services. He is focused on delivering cloud transformations for IBM's largest enterprise customers, where he brings extensive experience and expertise in enterprise architecture, DevOps, and cloud migrations.

KD Singh

KD is a Global Lead Partner Solution Architect with Amazon Web Services. He brings his expertise in cloud migrations, enterprise architecture, DevOps, and big data analytics to help AWS strategic partners build cloud offerings and implement digital transformations using the cloud.

Endnote

¹ *Witting, Andreas and Michael Wittig. Amazon Web Services in Action, p. 93. Shelter Island, New York: Manning Publications, 2016.*



© Copyright IBM Corporation 2018

IBM Corporation
New Orchard Road
Armonk, NY 10504

Produced in the United States of America
January 2018

IBM, the IBM logo, ibm.com, and Tealeaf are trademarks of International Business Machines Corp., registered in many jurisdictions worldwide. Other product and service names might be trademarks of IBM or other companies. A current list of IBM trademarks is available on the web at “Copyright and trademark information” at ibm.com/legal/copytrade.shtml

This document is current as of the initial date of publication and may be changed by IBM at any time. Not all offerings are available in every country in which IBM operates.

THE INFORMATION IN THIS DOCUMENT IS PROVIDED “AS IS” WITHOUT ANY WARRANTY, EXPRESS OR IMPLIED, INCLUDING WITHOUT ANY WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND ANY WARRANTY OR CONDITION OF NON-INFRINGEMENT. IBM products are warranted according to the terms and conditions of the agreements under which they are provided.



Please Recycle



Advanced
Consulting
Partner