Redbooks
ibm.com/redbooks

Always On
Considerations When Adopting the Cloud

Bertrand Portier

Edward Calusinski

In partnership with
IBM **Academy of Technology**

IBM

Redguide

Executive overview

As organizations look to differentiate in the marketplace, they are increasingly shifting their focus to the cloud as a preferred model for developing and hosting their solutions. The use of application programming interfaces (APIs) in the API economy is accelerating the rate by which organizations can deliver their solutions.

Although the business value of the use of cloud computing and the API economy can be understood, the operational requirements to realize the business value are not understood; specifically, the operational expectation of the cloud being Always On.

We assert that organizations hold an implicit expectation that services that are delivered via the cloud are always available because that is what they believe the cloud offers. Although we agree that cloud computing can enable an Always On solution, it is only with the explicit understanding that achieving Always On is a function of design and practice, and not something that is inherited.

In this IBM® Redguide™ publication, we explore the contemporary business need for Always On, the evolution of key organizational roles and responsibilities that are necessary to achieve Always On in the cloud, and a set of key capabilities that are required to ensure that Always On can be realized with cloud solutions.

© Copyright IBM Corp. 2015. All rights reserved.

Introduction

As we canvas the marketplace and examine the motivations of companies that are adopting cloud computing, we find that our research is consistent with the commonly cited reasons that are often published in trade magazines and digital publications. Companies look to cloud to provide an increased speed of information technology (IT) service delivery, mature business solutions (often in the form of Software as a Service), increased adoption of new technology innovation, and the ability to procure based on a consumption model.

Organizations see value and are motivated by the variable nature of cloud in cost, features, usage, and service level. This variability is appealing as it offers the tenants of the cloud the option to pay for only what they need without bearing the full cost burden that they otherwise need to when building their own solution in-house.

With the ultra competitive business environment today, business survival is predicated on the ability to rapidly use the cloud for traditional systems and new systems of engagement. There is a strong alignment of the business need to use cloud and the value proposition that cloud has to offer.

It seems obvious why organizations are moving quickly to use the cloud. However, inherent in the motivation to use the cloud is the expectation that the cloud is always available, or as we call it Always On. If business survival were predicated on the use of the cloud, it seems only natural to expect that the cloud be Always On.

Unfortunately, there is a significant expectation gap in the market. Cloud service providers (CSPs) might claim 100% uptime in their service level agreements (SLAs); however, although in certain circumstances it might be true that a CSP is providing 100% uptime of their services, it is often a special case scenario or a limited set of functions in their environment.

The reality is that none of the cloud service providers today offer and guarantee 100% uptime of their full set of services. Planned and unplanned outages can occur at any layer of the cloud stack. The Infrastructure as a Service (IaaS) layer, Platform as a Service (PaaS) layer, or Software as a Service (SaaS) layer are all susceptible to failure. Although CSPs might provide their services at higher degrees of service, the burden of ensuring that a solution is Always On falls on the organizations who use the cloud services, not the CSPs.

With the burden of ensuring Always On falling on the solution owner, it is critical that the solution owner ensures that the operations and procedures of the CSP and the solution owner support Always On practices and principles. Establishing a shared responsibility model between the solution owner and the CSPs helps ensure that the proper controls and procedures are in place to handle unexpected events, ensure that the affect of design changes are well-understood, and that the expected qualities of service levels are being realized.

In this paper, we explore the business drivers that facilitate the need to have an Always On environment in the cloud, the enterprise roles and responsibilities that are needed for achieving Always On in the cloud, and the set of technical capabilities that, when instituted, enable an organization to achieve Always On in the cloud.

Business survival requires Always On

We define Always On as a design style for achieving solutions that are responsive and usable by the user at anytime; that is, 100% uptime of the system as perceived by the user. Within Always On, we identify the following foundational principles:

- The ability to withstand component and application failures transparently.
- The ability to withstand catastrophic events transparently.
- The ability to introduce change non-disruptively.

As we discuss the design style of Always On with clients, we often hear objections or questions about the business and technical viability of trying to achieve Always On. Each organization must assess the business benefit versus the organizational and technical cost that is associated with achieving Always On based on their unique context. Historically, many clients conclude that Always On is too costly for all but the most mission-critical workloads and that their business often does not merit such stringent requirements. Although that ascertion might have been true in the past, we assert that in today's digital world, what was once an edge case need for Always On is now rapidly approaching a mainstream mandate.

To help illustrate this rapid move to mainstream, we start with an API economy example. For more information about the API economy, see *The Power of the API Economy: Stimulate Innovation, Increase Productivity, Develop New Channels and Reach New Markets*, REDP-5096.

Companies use the API economy to extend their product's reach by making their services available via standard APIs. By doing so, they enable the developer community to integrate their services into newly developed applications. Often, these applications are mobile applications or some type of system of engagement application.

In many cases, the APIs are supported by cloud-based infrastructures, which allows the developer to more rapidly compose their new application. Contemporary startups, such as Uber, disrupt their industries by using the API economy to enable their innovative business model. Traditional enterprises see the effect of the startups and are also beginning to use the API economy as a vehicle to extend their reach and establish new partnerships.

Citi's mobile challenge[1] is a great example of a traditional organization fully embracing the notion of the API economy. As organizations publish their services via APIs or as organizations rely on third-party APIs for running their own business, the need for providing an Always On experience becomes more essential.

In the API economy, the primary attraction to use an API is its functional capability. However, there is an underlying implicit expectation that the API is available for use when it is needed. After all, if the business model is predicated on the ability to use an API, it is obvious that the consumer of the service can expect it to be available. However, if this underlying expectation cannot be met, organizations look for alternative API solutions to protect their business interests.

For example, consider a startup that is creating a mobile application for the home buying experience. With this application, buyers can compare houses, get historical data, see whether they are pre-qualified for a loan, and even see how their furniture can look in the new home. This application uses the user's preferences, financial information, and risk tolerance levels to recommend the optimal mortgage for the home buyers unique situation (for example, 30-year fixed, 3-year adjustable, or 5-year balloon).

[1] http://www.citimobilechallenge.com

To achieve this recommendation, the application might use APIs to collect information about the user's finances, aggregate information from several lending institutions about eligibility requirements, interest rates, and the fees for their mortgage products. At the core of this startup's business model is the ability to integrate into a system of third-party market participants to complete the process.

The startup chooses a mobile application because it provides an "in context" user experience. The users experience the application at the point of greatest effect, at the heart of the home shopping experience, right in the home they are considering to purchase. However, this type of experience is time-sensitive. To maximize the value, the application must work at the moment when the user is in the home, dreaming about their potential future in that home. If the application (and the APIs that are central to its functions) is not working, the value is lost and the affect to the business is felt.

By using the API economy, the application can use multiple API providers for similar functions, which limits their risk on a sole provider of a service. This multi-provider model puts pressure on the API provider to ensure that their services are Always On, otherwise suffer the consequences of being disintermediated and the business effect of not participating in the transactions. For the application and the API providers, Always On is not only a nice feature, it is an essential requirement.

This example shows new business models where innovators from one company use other companies' APIs, which are hosted on the cloud, to provide services to their customers. The importance of the user experience makes it so that companies who fail to keep their APIs up and running all of the time are ignored and not part of the new economy. To some extent, companies that fail to update and rapidly innovate their own APIs while keeping the availability promise are left behind as well.

As we highlighted the importance of organizations becoming more aware of the need for an Always On design approach for their cloud solutions, we shift our focus to explore how essential organizational roles and responsibilities must evolve to better accommodate this style of design.

Enterprise roles and responsibilities must evolve to achieve Always On in the cloud

Many organizations are not suited to achieve high levels of resiliency in an operating environment that includes the cloud. In fact, enterprise roles and responsibilities (as they are defined today) tend to be an impediment to achieving Always On while using the cloud. Unless these roles and responsibilities evolve, organizations derive only limited benefits from the cloud or experience short-lived successes until the next outage.

Enterprises that operated for years have clearly defined roles and adhere to proven practices. For example, a process exists for promoting a code change from the staging environment to the production environment. The developer, system administrator, and other actors that are involved each have experience in running their steps in this process.

Also, as many steps as possible are automated to reduce the risk of human errors occurring. As a result, the organization as a whole trusts the "Promote code change from staging to production" process and many other processes. In fact, the maturation of these practices allowed organizations to achieve a respectable level of resiliency, sometimes close to Always On. However, most organizations achieved this level of maturity in a traditional on-premises operating environment.

As shown in analysts reports[2], organizations successfully adopted the cloud and achieved cost reduction and improved time to value. Despite these specific successes, enterprise-wide cloud adoption is a blurry vision that might not be realized for most organizations. Privacy and security are the most often cited primary concerns. For organizations who truly care about their customers experience, availability is also a critical concern and an impediment to scaling cloud adoption. Executives do not trust the cloud to achieve the same level of resiliency they are used to in their traditional on-premises environment. There are too many unknowns and lack of trust when it comes to evolving proven practices and processes to an environment that involves the cloud.

Let us use the same “promote code change from staging to production” process example, but this time with the staging environment hosted in an off-premise PaaS and the production environment hosted in the traditional on-premises environment. With this evolved environment comes several unknowns. For example, what is the role of the system administrator regarding the off-premise PaaS? How will the executable code be transferred from off-premise to on-premises? Who is responsible if the transfer fails?

To better understand the challenges decision makers face, we take a step back and look at three important IT executive roles in the context of Always On and cloud adoption: vice president (VP) of IT operations, VP of architecture, and VP of development.

With today’s customers demanding better experiences more quickly, the VP of IT operations is faced with the need to support enterprise practices that are becoming more agile. This VP has not had much time to review cloud offerings because of the daily pressures to keep the traditional on-premises systems running.

The VP of architecture closely follows and evaluates industry innovations and helps their organization adopt the relevant innovations. The cloud is one of these innovations their business users, Business Partners, and vendors talk about regularly. This VP works with the rest of the organization to establish a decision framework for cloud-based offerings.

The VP of development is collaborating with the VP of IT operations in the context of the enterprise DevOps initiative. They are faced with releasing more features more quickly, and at a higher-quality level. They find attractive the ability for their team to use the cloud to rapidly self-provision development and test environments.

Based on their roles and associated responsibilities, each of the three VPs look at things through a different lens. They each have a set of core principles that they refer to when making strategic and tactical decisions, including cloud and resiliency-related decisions. For example, the VP of architecture, through their innovation lens, believes that the organization needs to shift from “building” and “running” systems to becoming a true business solution innovator. They believe in a cloud-first approach where any new workload or application runs in an off-premise cloud by default, unless there is a good reason for it not to.

However, the VP of IT operations is worried about the disruptions, outages, or breakages the cloud can cause when they look at their situation through their risk avoidance lens. When the VP of architecture talks to them about adopting the cloud, they tells them that as the VP of IT operations, they are still responsible for the availability of systems and applications. Until that responsibility changes, they primarily use their own proven and trusted practices and systems, which are not cloud-based.

Each of the VPs have their own point of view about the cloud and how to make decisions about it. We refer to these conflicts as value conflicts, which are an impediment to enterprise-wide cloud adoption.

---

[2] Forrester Q4 2013

One way to alleviate these value conflicts is for each of the roles to make trade-offs. For example, the VP of IT operations might allow for the staging environment to be hosted on an off-premise PaaS environment. In exchange, the VP of development must institutionalize practices to make the code their department produces more resilient in a cloud environment. Let us look at this issue more closely.

When aiming for Always On in the cloud, the VP of development faces the challenge of shifting from releasing large sets of changes infrequently to releasing small sets of changes frequently. This change allows for improved time-to-value, reduced complexity, and smaller effect in case of failures.

To achieve this improvement, developers must change the way that they design, implement, and test applications. One aspect of this process is dependencies. In a traditional environment, developers rely on the availability of "system-wide" packages to use in their code. In a cloud-based environment, developers cannot rely on system-wide dependencies because they have less control over how their application scales and where it runs. In fact, cloud developers must explicitly declare individual dependencies[3].

Developers want to release more features more quickly. Speed is one of their core principles. For developers to update their dependency practice, they must take a step back instead of moving at full speed. However, if the VP of development compromises on speed and improves their group's dependency practice, it reduces the risk of outages. With this reduced risk, the VP of IT operations is then in a better position to accept responsibilities for the availability of the off-premises PaaS staging environment.

The cloud staging environment example illustrates the challenges organizations face when considering resiliency and the cloud. To summarize, unknowns and value conflicts exist between the different roles in the enterprise. To alleviate these challenges, organizations must evolve the definition of roles, responsibilities, and practices to make the cloud an acceptable evolution for the enterprise.

After organizations get past these initial internal hurdles, there are critical issues to consider. For example, organizations must have a clear understanding of their responsibilities versus those of their CSPs. After that understanding is in place, the next step is to start experimenting. This experimentation allows organizations to build success stories about cloud adoption and resiliency. These success stories lead to increased confidence and broader adoption. Then, the cloud becomes a critical enabler for Always On, which provides automation, speed of execution, diversification, and geographic distribution.

The examples that are described in this section illustrate the collaborative nature of today's workplace. People, teams, and companies no longer work in isolation and are becoming part of a wider system in which everyone plays a role to keep the Always On promise. This emphasis on collaboration accentuates the need for the practice of architecture, which helps communicate and document each party's needs, constraints, and responsibilities.

When adopting a cloud environment, the role of the architect evolves and the decisions they make are different from the ones they made in a traditional on-premises environment. For example, the architect has no choice about the type of systems (processor, memory) to use when using an IaaS provider. However, the architect now must make decisions about how the data is transferred between the company's data center and the cloud provider's data center. When aiming to achieve Always On in the cloud, we believe that architects must consider the 10 capabilities that are described in the next section.

[3] This and other practices are described in The Twelve-Factor App, a comprehensive methodology for developing software, which runs as a service.

Achieving Always On in the cloud requires specific technical capabilities

Always On means achieving 100% uptime, or the ability to prevent outages as perceived by the user. Always On lies in the ability to withstand component and application failures, withstand catastrophes transparently, and the ability to introduce changes non-disruptively. Although many organizations instituted strong practices to strive to achieve Always On in their own data center environments, how well does this translate to an off premise cloud environment? How can an organization institute these practices in an off-premise cloud environment when they do not have control of the full application stack? We assert that it can be done, but it is essential to understand the key capabilities that are necessary to achieve Always On when working with off premise cloud providers.

In this section, we highlight a set of necessary key practices and capabilities to consider when you are attempting to achieve Always On in an off-premise cloud environment.

Eliminating single points of failure

Eliminating single points of failure might be one of the most obvious practices when dealing with your private data center where you have complete visibility to the application and the supporting infrastructure. However, what level of visibility do you have with your cloud providers to answer the same question? As part of this issue, we look at geographic dispersion and partition tolerance.

Geographic dispersion

Ensuring that your applications can operate over a set of geographically dispersed facilities is necessary to withstand catastrophic events transparently. In a hurricane, fire, flood, power outage, public unrest, or significant human errors, can your applications continue to run when a catastrophic event occurs? What mechanisms does your cloud provider offer to aid in ensuring your applications can run in multiple facilities that are far enough apart to not be affected by the same event?

Partition tolerance

Partition tolerance is the ability for the solution to continue to operate despite arbitrary message losses or failures of part of the solution. For example, if there is a network failure between two sites (data centers), each of the sites continues to operate independently. Supporting partition tolerance involves making trade-offs against supporting data consistency. In our example, the data is no longer consistent between the two sites. Brewers CAP theorem states that a distributed system can guarantee only two out of three features: data consistency, availability, and partition tolerance. What level of partition tolerance does your CSP provide within one of their data centers, across their data centers, or across theirs and others' data centers?

Testing and experiencing failures

Practice makes perfect. Therefore, it is essential to plan for and test for failures. This practice should include catastrophic failure testing. Many organizations have disaster recovery (DR) plans and procedures, but do not institute regular testing and running of their DR processes. With Always On, we remove the need for separate DR processes and facilities, as the principles of DR are a part of the Always On model.

We encourage getting as much experience as possible with failure scenarios (components, application, systems, or environmental). For example, Netflix famously uses the chaos monkey[4] to break elements of the environment to observe how the rest of the environment reacts to these disruptions.

Load balancing

Load balancing is the ability to route application traffic. The following types of load balancing are available:

- Global

  This type of load balancing is the ability to route application traffic to multiple geographically separated cloud instances and balance workload requests to accommodate catastrophic failures. This capability is critical to use geographic dispersion of workload. It might be necessary to examine the applications to determine whether any data or application logic localities are assumed in the design of the application. If the applications are designed to expect data to be in a fixed location, global load balancing can be difficult to realize. In what way does your cloud service provider offer you services and capabilities that support the use of global load balancing?

- Local

  This type of load balancing is the ability to route application workload traffic to multiple instances of the application within a local cloud. This capability allows the application to withstand component failures transparently and help ensure higher availability. Does your cloud provider offer this service and can your application use this capability?

Data replication

Data replication is the ability to write data to separate systems. The following types of data replication are available:

- Synchronous

  This replication is the ability to write to multiple independent storage systems simultaneously, which ensures that data is present in multiple locations where multiple instances of the application can access the same data while ensuring data consistency. This capability is necessary for workloads with absolute data consistency requirements. With synchronous data replication, there is a limit on the distance between storage systems before you run into significant performance issues. Can your cloud service provider support synchronous or dual writes without affecting performance and the user experience?

- Asynchronous

  This replication is the ability to provide eventual consistency of data between two separate systems where the data that is written in one system is asynchronously written in a secondary system (eventually). This capability is necessary to cross the long distances between cities where synchronous data replication in not viable. In what way does your cloud provider allow you to replicate data asynchronously thus achieving eventual data consistency?

[4] http://techblog.netflix.com/2012/07/chaos-monkey-released-into-wild.html

Visibility and control

The ability to have visibility into the infrastructure environment and have a mechanism to take action when conditions indicate that a failure occurs or is likely to occur. Having the ability to gain an understanding of root cause and eventually predict outages before they occur is a key capability for achieving Always On. In what way does your cloud provider allow you to monitor your end-to-end solution? What monitoring platform or interfaces (APIs) does your cloud platform provide to gain this visibility? What mechanism does your cloud provider offer for alerting if specific thresholds are met?

Automated remediation

Automated remediation is the ability to autonomously run a set of actions that automatically change the configuration of the environment to ensure continuous operations in response to an anticipated or actual event. To achieve “5 9's” of availability (99.999% availability) means maintaining 27 seconds or less of downtime per month. Service levels such as this level do not allow for human intervention if there is a failure. Integration of automation into the cloud environment is essential. What automations can be implemented in the cloud environment to immediately take action if there is an issue before a component failure or another glitch has a negative effect on the customer experience?

Upgrades, roll-backs, and A/B testing

Often overlooked is the notion of introducing change non-disruptively. Non-disruptive change is a challenge for most organizations because upgrades often involve planned or unplanned outages. The introduction of cloud computing does not alleviate this challenge.

Upgrades

How much visibility do you have into your cloud service provider’s planned outages or upgrade schedule? When the cloud service provider upgrades, do your customizations upgrade automatically? What mechanism do you have to introduce changes to your cloud applications non-disruptively? For more information about the considerations and required capabilities to enable nondisruptive change, see *Achieving Always On: Introducing changes without breaking the availability promise*, which is available at this website:

http://www.ibm.com/developerworks/library/d-achieving-always-on/index.html

Fast roll-back

Fast roll-back is an operational technique that allows for previous versions of application code or operational changes to be reverted to a known good state quickly. If a faulty application is deployed, how quickly can your cloud service provider roll-back to the previous version? Do they offer gradual deployment of applications across the cloud instance (eventual application version consistency)?

A/B testing

A/B testing is an application testing technique where a new version of the application (version B) is deployed to production to a small subset of users while the previous version (version A) is still handling most of the workload. This technique is used to help minimize the effect of code releases and provide early insight into issues that might cause a negative event. Although this technique is often used at the application level, this technique easily can be applied at the PaaS or IaaS level. In what way does your cloud provider help you practice A/B testing?

Cloud connectivity and integration

By definition, the cloud is not one environment only, but a collection of many environments that are potentially provided by multiple cloud service providers or in your on-premises data center. As such, organizations must ensure stable inter-connectivity and integration between cloud instances to benefit from the cloud style of computing.

Your cloud is a combination of geographically dispersed operating environments with various levels of isolation. Security, gateways, network links, and firewalls all must be constantly monitored and managed to ensure the web of inter-connections are up and secure. What tooling and insight does your cloud provider offer to allow you to monitor and manage the interconnections?

Workload portability

Workload portability is the ability to move applications between infrastructure instances in an automated manner. Workload portability of applications in your cloud environment is key for Always On. The use of automation to move workloads between cloud instances allows for infrastructure changes that require recycling of the infrastructure to not affect the applications. If the applications are not portable or require significant hand holding to move, the applications are susceptible to the effect of infrastructure changes. There is another side benefit of workload portability in that it helps reduce cloud vendor lock in. The easier it is to move applications, the easier it is to change cloud vendors.

Continuous delivery

Continuous delivery is a software development approach where teams deliver software in small increments and ensure that the software can be reliably released at any time. This approach moves away from the event of a significant release date, which is often littered with issues that affect system availability. Continuous delivery by nature removes the burden of the significant planned outage model by incrementally developing code, releasing the code, and gradual changes to the operating environment are all performed in smaller more manageable chucks.

The cloud is often used in the continuous delivery model because of the inherent automated practices most cloud service providers offer to their tenets. However, how often does your cloud provider deliver major or minor changes? What is the effect this has on your continuous delivery model?

In this section, we described 10 capabilities that are critical to achieving Always On. We raised concerns about how an organization might achieve these capabilities when the cloud is used. Our goal was to provide a framework for the reader to consider these important unknowns and have a discussion with their cloud service providers. In future articles, we plan to provide specific guidance on how to address these challenges.

Summary

In this paper, we highlighted the importance of building Always On solutions for the cloud. There is little doubt that traditional organizations are continuing to adopt cloud computing for more of their critical workloads. The cloud is the dominant option for startups and we outlined the importance of the API economy in aiding the development of core business models between newer and more established organizations.

There is strong evidence this trend will continue. With a stronger dependency on cloud computing as the foundation of the business comes the realization of the importance of the qualities of service necessary to maintain a vibrant and durable infrastructure and application development model for the applications that are running in the cloud. In particular, the necessary capabilities to create an Always On cloud environment at the infrastructure, operations, and application layers of the solution.

We also highlighted how the key roles in the development, operations, and architecture departments must evolve and converge to accommodate the unique aspects that are essential to achieving Always On in the cloud. In particular, the roles and responsibilities must be shared and understood by the solution owners and the CSPs. We also emphasized that with the sharing of responsibilities, value conflicts can emerge between competing values (speed, qualities of service, and costs) and that it is critical to compromise as the value conflicts emerge.

We concluded by defining the necessary set of capabilities that are required to achieve Always On in the cloud and to help ensure that there is a resilient and vibrant operating environment for the cloud. We believe that by using these capabilities and working with your cloud service provider to help you realize these capabilities, you and your organization can be well on your way to achieving Always On.

Other resources for more information

For more information, see the following resources:

- *The Power of the API Economy: Stimulate Innovation, Increase Productivity, Develop New Channels and Reach New Markets*, REDP-5096:

  http://www.redbooks.ibm.com/abstracts/redp5096.html?Open
- *Always On: Business Considerations for Continuous Availability*, REDP-5090:

  http://www.redbooks.ibm.com/abstracts/redp5090.html?Open
- *Always On: Assess, Design, Implement, and Manage Continuous Availability*, REDP-5109:

  http://www.redbooks.ibm.com/abstracts/redp5109.html?Open
- *Achieving Always On: Introducing changes without breaking the availability promise:*

  http://www.ibm.com/developerworks/library/d-achieving-always-on/index.html
- IBM Bluemix:

  http://bluemix.net
- *Using the cloud to improve business resilience:*

  http://www.ibm.com/common/ssi/cgi-bin/ssialias?infotype=SA&subtype=WH&htmlfid=BUW03026USEN
- *Top 9 rules for cloud applications:*

  http://www.ibm.com/developerworks/websphere/techjournal/1404_brown/1404_brown.html

Authors

This guide was produced by the following team of specialists:

Bertrand Portier is an IBM Executive IT Architect. He has written extensively about Always On, distributed computing, modeling, service orientation, and business process management. He holds a Master of Science degree in Computer Engineering. He lives in Austin, Texas.

Edward Calusinski is an IBM Distinguished Engineer in North America. He has been with IBM for 20 years, serving clients in the field by helping them adopt technologies to address their business challenges. He holds a Bachelors degree in Metalurgy from Lewis University and a Masters degree in Computer Science from the Illinois Institute of Technology. He is an expert in Enterprise Architectures, application development, and distributed computing. He has written extensively about various IT Architectures design styles and the application of architectural patterns.

We acknowledge and thank the following people who helped contribute their time and expertise to this guide:

- George Galambos
- Jeff Calusinski
- Rashik Parmar
- John Heap
- Jonathan Collins
- Chris Molloy
- Herbie Pearthree
- Scott Simmons
- Jaylani Sharif
- Tony Carrato

Now you can become a published author, too!

Here's an opportunity to spotlight your skills, grow your career, and become a published author—all at the same time! Join an ITSO residency project and help write a book in your area of expertise, while honing your experience by using leading-edge technologies. Your efforts help to increase product acceptance and customer satisfaction, as you expand your network of technical contacts and relationships. Residencies run 2 - 6 weeks in length and you can participate in person or as a remote resident working from your home base.

Find out more about the residency program, browse the residency index, and apply online at this website:

ibm.com/redbooks/residencies.html

Stay connected to IBM Redbooks

For more information about IBM Redbooks, see the following resources:

- Find us on Facebook:

  http://www.facebook.com/IBMRedbooks

- Follow us on Twitter:

  http://twitter.com/ibmredbooks

- Look for us on LinkedIn:

  http://www.linkedin.com/groups?home=&gid=2130806

- Explore new Redbooks publications, residencies, and workshops with the IBM Redbooks weekly newsletter:

  https://www.redbooks.ibm.com/Redbooks.nsf/subscribe?OpenForm

- Stay current on recent Redbooks publications with RSS Feeds:

  http://www.redbooks.ibm.com/rss.html

Notices

This information was developed for products and services offered in the U.S.A.

IBM may not offer the products, services, or features discussed in this document in other countries. Consult your local IBM representative for information on the products and services currently available in your area. Any reference to an IBM product, program, or service is not intended to state or imply that only that IBM product, program, or service may be used. Any functionally equivalent product, program, or service that does not infringe any IBM intellectual property right may be used instead. However, it is the user's responsibility to evaluate and verify the operation of any non-IBM product, program, or service.

IBM may have patents or pending patent applications covering subject matter described in this document. The furnishing of this document does not grant you any license to these patents. You can send license inquiries, in writing, to:
IBM Director of Licensing, IBM Corporation, North Castle Drive, Armonk, NY 10504-1785 U.S.A.

The following paragraph does not apply to the United Kingdom or any other country where such provisions are inconsistent with local law: INTERNATIONAL BUSINESS MACHINES CORPORATION PROVIDES THIS PUBLICATION "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Some states do not allow disclaimer of express or implied warranties in certain transactions, therefore, this statement may not apply to you.

This information could include technical inaccuracies or typographical errors. Changes are periodically made to the information herein; these changes will be incorporated in new editions of the publication. IBM may make improvements and/or changes in the product(s) and/or the program(s) described in this publication at any time without notice.

Any references in this information to non-IBM Web sites are provided for convenience only and do not in any manner serve as an endorsement of those Web sites. The materials at those Web sites are not part of the materials for this IBM product and use of those Web sites is at your own risk.

IBM may use or distribute any of the information you supply in any way it believes appropriate without incurring any obligation to you.

Any performance data contained herein was determined in a controlled environment. Therefore, the results obtained in other operating environments may vary significantly. Some measurements may have been made on development-level systems and there is no guarantee that these measurements will be the same on generally available systems. Furthermore, some measurements may have been estimated through extrapolation. Actual results may vary. Users of this document should verify the applicable data for their specific environment.

Information concerning non-IBM products was obtained from the suppliers of those products, their published announcements or other publicly available sources. IBM has not tested those products and cannot confirm the accuracy of performance, compatibility or any other claims related to non-IBM products. Questions on the capabilities of non-IBM products should be addressed to the suppliers of those products.

This information contains examples of data and reports used in daily business operations. To illustrate them as completely as possible, the examples include the names of individuals, companies, brands, and products. All of these names are fictitious and any similarity to the names and addresses used by an actual business enterprise is entirely coincidental.

COPYRIGHT LICENSE:

This information contains sample application programs in source language, which illustrate programming techniques on various operating platforms. You may copy, modify, and distribute these sample programs in any form without payment to IBM, for the purposes of developing, using, marketing or distributing application programs conforming to the application programming interface for the operating platform for which the sample programs are written. These examples have not been thoroughly tested under all conditions. IBM, therefore, cannot guarantee or imply reliability, serviceability, or function of these programs.

© Copyright IBM Corp. 2015. All rights reserved.

This document, REDP-5297-00, was created or updated on October 13, 2015.

Trademarks

IBM, the IBM logo, and ibm.com are trademarks or registered trademarks of International Business Machines Corporation in the United States, other countries, or both. These and other IBM trademarked terms are marked on their first occurrence in this information with the appropriate symbol (® or ™), indicating US registered or common law trademarks owned by IBM at the time this information was published. Such trademarks may also be registered or common law trademarks in other countries. A current list of IBM trademarks is available on the Web at http://www.ibm.com/legal/copytrade.shtml

The following terms are trademarks of the International Business Machines Corporation in the United States, other countries, or both:

Redbooks (logo) ®
Bluemix™
IBM®
Redbooks®
Redguide™

Other company, product, or service names may be trademarks or service marks of others.

REDP-5297-00

ISBN 0738454567

Printed in U.S.A.

Get connected