DZone

THE DZONE GUIDE TO
CLOUD DEVELOPMENT

2015 EDITION

BROUGHT TO YOU IN PARTNERSHIP WITH

New Relic®

redhat.

stackato® by ActiveState®

Dear Reader,

"Cloud computing is an evolving paradigm." - The NIST Definition of Cloud Computing 1.2. September 2011.

No kidding. Even NIST can't avoid thinking of cloud computing as something nebulous (etymologically, of course). The metaphor won't break character.

But the cloud metaphor isn't precise enough. Yes, physical locations of storage, computation, and delivery vary happily nowadays. Sure, Turing-drops cohere into variously-formed services, services precipitate elastically into ever-shifting applications, and *aas-backed applications deliver previously unimaginable deluges of user value at lightning speed (presumably thundering all the while). Is "anything as a service" what makes cloud so exciting?

No—"*aaS" doesn't capture the bleeding edge. Everybody knows about web services nowadays—even specific providers, still most notably Amazon, but now who knows what else (well, we do—check out the Solutions Directory at the end of this guide). And the concept of computing services is as old as the operating system.

Nor do NIST's five essential characteristics, three service models, and three deployment models really capture why "cloud" is neither entirely precise nor an entirely meaningless buzzword. We're stuck somewhere between arbitrarily bundled sets of properties and a hazy non-definition. We need more concretes from developers and deeper conceptual penetration from researchers.

Toward the abstract: we've included some (we hope) original thoughts on the present and future of cloud development. Location-indifference is bourgeoning at deeper infrastructure levels as well: named data networking, content-centric networking, and information-centric networking concepts—making content the primary addressable, not the physical endpoints—are at the heart of future Internet research in North America, Japan, and Europe.

Toward the concretes: we've done some original research, crunched some numbers, and cross-checked a few more. The 2015 edition of the *Guide to Cloud Development* is our second second edition (the *Guide to Continuous Delivery* was our first), so we're starting to see the inklings of some maybe-new trends in cloud adoption by software developers.

I hope you'll find this guide as interesting as I do.

JOHN ESPOSITO
EDITOR-IN-CHIEF, DZONE RESEARCH
RESEARCH@DZONE.COM

Table of Contents

3 SUMMARY & KEY TAKEAWAYS

4 KEY RESEARCH FINDINGS

6 DEVELOPING SOFTWARE FOR THE CLOUD: THE DHARMA PRINCIPLES
BY DANIEL BRYANT

12 7 RULES FOR HYBRID CLOUD ARCHITECTURES
BY JP MORGENTHAL

16 THE CLOUD SERVICES LANDSCAPE
INFOGRAPHIC

18 WHY RUN YOUR MICROSERVICES ON A PaaS
BY CHRIS HADDAD

21 DIVING DEEPER INTO CLOUD DEVELOPMENT

24 FORECASTING THE NEXT YEAR IN CLOUD
BY BERNARD GOLDEN

26 QUESTIONS TO ASK ABOUT SECURITY IN THE CLOUD
BY JOHN WETHERILL

27 SOLUTIONS DIRECTORY

30 DIVING DEEPER INTO CLOUD DEVELOPMENT PRODUCTS

31 GLOSSARY

Credits

EDITORIAL

John Esposito
research@dzone.com
EDITOR-IN-CHIEF

Jayashree Gopal
DIRECTOR OF RESEARCH

Mitch Pronschinske
SR. RESEARCH ANALYST

Benjamin Ball
RESEARCH ANALYST

Matt Werner
MARKET RESEARCHER

John Walter
CONTENT CURATOR

Ryan Spain
CONTENT CURATOR

BUSINESS

Rick Ross
CEO

Matt Schmidt
PRESIDENT & CTO

Kellet Atkinson
GENERAL MANAGER

Alex Crafts
sales@dzone.com
VP OF SALES

Matt O'Brian
DIRECTOR OF BUSINESS DEVELOPMENT

Chelsea Bosworth
MARKETING ASSOCIATE

Chris Smith
PRODUCTION ADVISOR

Brandon Rosser
CUSTOMER SUCCESS ADVOCATE

Jillian Poore
SALES ASSOCIATE

Jim Howard
SALES ASSOCIATE

ART

Ashley Slate
DESIGN DIRECTOR

Special thanks to our topic experts Chris Haddad, Daniel Bryant, Bernard Golden, JP Morgenthal, Christian Posta, Nati Shalom, and our trusted DZone Most Valuable Bloggers for all their help and feedback in making this report a great success.

WANT YOUR SOLUTION TO BE FEATURED IN THIS OR COMING GUIDES?
Please contact research@dzone.com for submission information.

LIKE TO CONTRIBUTE CONTENT TO COMING GUIDES?
Please contact research@dzone.com for consideration.

INTERESTED IN BECOMING A DZONE RESEARCH PARTNER?
Please contact sales@dzone.com for information.

Summary and Key Takeaways

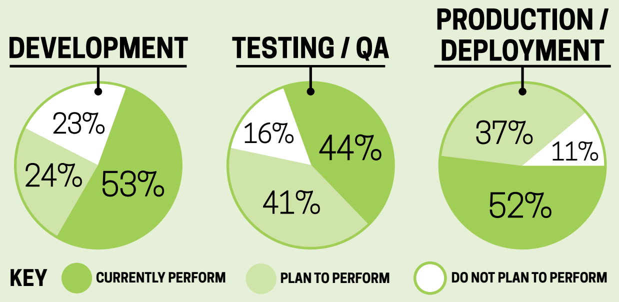
CLOUD TECHNOLOGIES ARE NOTHING NEW TO THE SOFTWARE development world—but increasingly, teams are discovering the value of cloud development environments. Cloud technologies are rapidly evolving, and cloud services have moved from the backroom IT department to the front of complex and successful businesses. And it's not just the technology itself that's maturing. Adoption of cloud computing is constantly bringing new teams, companies, and services into the fold. The DZone *2015 Guide to Cloud Development* looks at not only what your options are for cloud providers and services, but what IT professionals are accomplishing with these changing technologies. The resources in this guide include:

- A directory of the best solutions for PaaS, iPaaS, IaaS, and MBaaS.
- Articles from topic experts and industry luminaries.
- Security questions to ask your cloud provider.
- A visual classification of cloud service providers by layer and use.
- Analysis of patterns and usage based on research collected from 600+ IT professionals.

STRONG ADOPTION IN TESTING, DEPLOYMENT, AND PRODUCTION ENVIRONMENTS

One of the biggest changes in this year's survey results from last year's has been the growth of Testing/QA and Production/Deployment in the cloud. Our results show that 55% of respondents are performing Testing/QA in the cloud (11% increase), and 62% are performing Production/Deployment (10% increase). And this growth isn't just production and testing environments catching up to development—Production/Deployment is now more likely to be performed on a cloud platform than Development. This shift could mean that last year's respondents who said they were "Planning to Perform" testing and deployment on a cloud platform actually kept their word.

01. SOFTWARE LIFECYCLE CLOUD USAGE



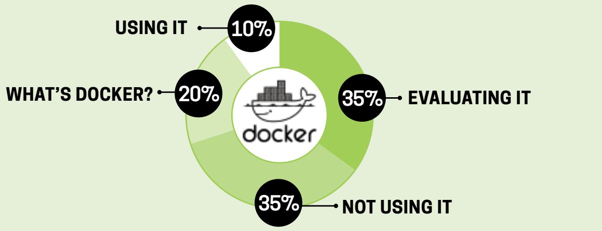
NOT THAT MANY ACTUALLY HAVE SECURITY ISSUES

For the second year in a row our survey respondents said security in the cloud was their number one concern when picking a provider. When asked what issues they expect to experience with a cloud platform, 75% of respondents expected security issues, while only 29% of respondents actually experienced them. This disparity is especially large when considering that security was the issue that the least amount of respondents currently deal with, falling behind reliability issues (33%) and issues managing multi-cloud environments (35%).

THEY'RE PROBABLY CONSIDERING DOCKER CONTAINERS

Docker has changed the way that many IT professionals prepare their applications for production environments, but is anyone actually using Docker in their organization? It's a fair question—Docker has barely been around for a year and a half, and stable enough to be production-ready for much less than that. We found that 10% of respondents are currently using Docker or container technologies in their organization, and 35% are evaluating it for usage. Of those who are using or considering Docker, 89% intend to use it in Development, 66% for QA/Testing, and 61% in Production. Even 10% of users is significant adoption for such a new technology.

02. DO YOU CURRENTLY USE DOCKER OR CONTAINER TECHNOLOGIES IN YOUR ORGANIZATION?



Key Research Findings

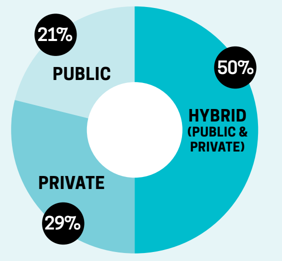
More than 600 IT professionals responded to DZone's 2015 Cloud Development Survey. Here are the demographics for this survey:

- **Developers and Engineers made up 69% of the total respondents.**
- **60% of respondents come from large organizations (100 or more employees) and 40% come from small organizations (under 100 employees).**
- **The majority of respondents are headquartered in Europe (48%) or the US (28%).**

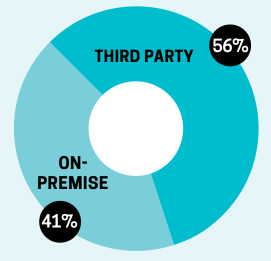
BIGGER TEAMS WANT MORE CONTROL OF THE CLOUD

Half of all respondents (50%) see hybrid cloud as their ideal platform, while private (29%) leads slightly ahead of public offerings (21%). When asked about cloud hosting types, respondents preferred third party (56%) over on premise (41%). When comparing these results with the team size of survey respondents, we found that larger teams preferred an on premise and private cloud, whereas smaller teams preferred third party and public options. Below the 100 employee mark, respondents more likely to select public and third party—and they chose on premise and private above the 100 employee mark.

01. WHICH CLOUD PLATFORM BEST FITS YOUR COMPANY'S NEEDS?



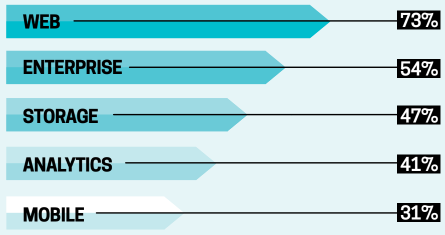
02. TYPE OF CLOUD HOSTING USED



STORAGE AND ANALYTICS ARE ON THE RISE

Respondents are most likely to deploy web applications (73%) and enterprise applications (54%) in a cloud environment, and those numbers have remained consistent in the last year. Mobile application deployment also remained firm at 31%. More interesting is the increase in deployment of storage (47%), analytics (41%), and batch processing applications (36%). All of these applications showed noticeable growth, with storage applications up 5%, and both analytics and batch processing up 8%. This growth seems to correspond to an increase in Big Data applications and practices, since most of these application types are involved in either storing, processing, or analyzing large amounts of data, and benefit from the elasticity of cloud infrastructure.

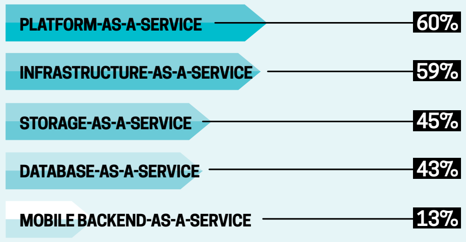
03. WHAT TYPES OF APPS HAVE YOU DEPLOYED IN A CLOUD ENVIRONMENT?



PAAS AND IAAS ARE FIGHTING FOR MAJORITY

When asked which cloud-based services they use in software production, respondents were most likely to answer Platform-as-a-Service (PaaS) and Infrastructure-as-a-Service (IaaS), which comes as no surprise. The service to see the most growth was Infrastructure-as-a-Service, which grew by 7% since last year's survey results. It's interesting to note that the prominence of Storage-as-a-Service (45%) and Database-as-a-Service (43%) likely relates to the impact of Big Data technologies on cloud environments.

04. WHICH CLOUD SERVICES DO YOU USE IN PRODUCTION?



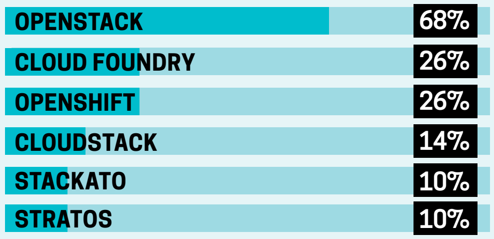
05. TOP CLOUD SECURITY FEATURES

1
DATA ENCRYPTION
2
DATA LOCATION
3
AUTHEN-TICATION OPTIONS
4
FIREWALLED SERVERS
5
DEDICATED SERVERS

TAKING A CLOSER LOOK AT CLOUD SECURITY

For the second year in a row, respondents chose security as the number one factor when choosing a cloud provider by 20% more than the second most important factor. Since our survey audience has repeatedly shown interest in security, we asked about the cloud security factors they value when choosing a provider. Data encryption and data location took the top spots, followed by authentication options, firewalled servers, and dedicated servers. Cloud security is often focused on the security of data and where it's kept, and this data indicates that respondents see a lot of value in such precautions.

06. OPEN SOURCE CLOUD PRODUCT USED IN BUSINESS APPLICATIONS



OPEN SOURCE COMMANDS A LARGE SHARE OF THE USERS

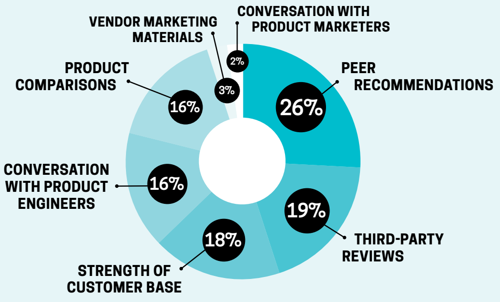
31% of all survey respondents indicated that they have used OpenStack for a business application, and that includes the 55% of our audience that hasn't used an open source solution. When focusing only on respondents that have used an open source cloud product for a business application, those numbers go way up. Users of open source products by far preferred OpenStack (68%) followed by OpenShift (26%) and Cloud Foundry (26%) in a tie. It's interesting to note that OpenStack usage in our audience increased 21% in the last year, and may be drawing away users from other solutions. Those who responded as having used open source products were also considerably more likely to have used a free trial product (65% of respondents).

OpenStack adoption increased 21% among open source cloud users

PEER AND PRODUCT REVIEWS SELL PRODUCTS

When asked what resources they rely on when choosing a cloud product, peer recommendations were the most relied on, and conversations with product marketers had the least impact. By looking at a breakdown of resources that they strongly relied on, respondents valued peer recommendations the most (26%), followed by third party reviews (19%) and strength of customer base (18%). The data suggests that IT professionals value products that they know their peers and the community are using, and which have ample documentation through third-party sources. What doesn't click with IT professionals are materials they think have a vendor marketing bias.

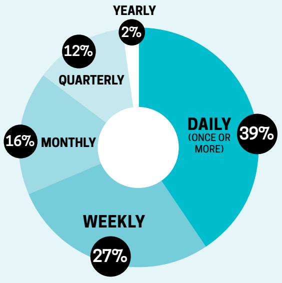
07. WHAT RESOURCES DO YOU STRONGLY RELY ON WHEN CHOOSING A CLOUD PRODUCT?



MOST CLOUD USERS DEPLOY CODE WEEKLY OR DAILY

When asked about the speed at which they deploy code, 39% of survey respondents said they deploy code once or more a day, 27% deployed weekly, 16% monthly, and so on. The results show that most cloud users are moving towards daily and on-demand deployment on the whole. When you break down respondents by company size, the interesting result is that there's very little difference in deployment frequency. 24.9% of small companies (<100 employees) deploy on-demand and 14.7% daily, whereas 25.1% of large companies (>100 employees) deploy on-demand and 12.9% daily.

08. HOW OFTEN DOES YOUR ORGANIZATION DEPLOY CODE?



DEVELOPING SOFTWARE FOR THE CLOUD:

The DHARMA Principles

BY DANIEL BRYANT

QUICK VIEW

01

The six DHARMA principles for cloud development: **D**ocumented, **H**ighly cohesive/loosely coupled, **A**utomated from commit to cloud, **R**esource aware, **M**onitored thoroughly, **A**ntifragile

02

Always document architecture and deployment topologies, utilizing software design principles, creating a comprehensive build pipeline, implementing comprehensive monitoring, and building antifragile systems

03

The benefits of these principles make cloud environments attractive for many software applications

Everyone is talking about moving to the cloud these days, regardless of whether the business actually focuses on software development. We software creators are constantly being challenged by the leading "cloud native" companies, such as Netflix and Amazon, to take better advantage of the unique environment provided by the cloud. Whether an application is being migrated to the cloud (a "lift and shift"), built as a greenfield project, or developed as hybrid migration and enhancement, cloud developers face a common set of challenges.

This article aims to accomplish two things: first, to identify and discuss these challenges; and second, to offer advice on how to overcome them.

FLYING HIGH (AND LOW) IN THE CLOUD - CHANGING THE GAME

If done correctly, cloud-based deployments are a game-changer, and can enable extremely rapid deployment of software that can be scaled elastically to meet demand. However, this flexibility is countered by several constraints placed on the architecture and design of software. Common problems experienced when building software for the cloud can be broadly grouped into the following categories:

- Difficulty in communicating the new design and deployment topology that results from embracing the cloud
- Problems with creating an architecture that works in harmony with (and leverages) cloud infrastructure
- Lack of testing in a cloud-based environment
- Lack of understanding of the underlying cloud fabric and its properties
- Limited strategy for application and platform monitoring
- Difficulty in designing for bizarre (and partial) cloud infrastructure failure modes

The remainder of this article will discuss these issues in greater depth, and also provide strategies for overcoming or mitigating the accompanying challenges.

CORE DESIGN PRINCIPLES - INSPIRATION FROM COSMIC LAW AND ORDER

This article introduces the cloud DHARMA development principles, which have been designed to be used in much the same way as the SOLID software development principles. The word dharma can be found in Hindu and Buddhist texts, and signifies behaviors that are considered to be in accord with the order that makes life and the universe possible. Although this underlying definition may also be relevant, the word has been chosen primarily as a mnemonic to represent the following cloud development guidelines:

- **D**ocumented (just enough)
- **H**ighly cohesive/loosely coupled (all the way down)
- **A**utomated from commit to cloud
- **R**esource aware
- **M**onitored thoroughly
- **A**ntifragile

DOCUMENTATION - NOT JUST FOR WATERFALL PROJECTS

Documentation is often shunned in software projects that are run using an agile methodology, despite many prominent members of the community, including Simon Brown and Bob Martin, constantly attempting to reiterate the benefits of "just enough" documentation. Whether a greenfield software project is being designed for a cloud-native deployment, or an existing application is being "lifted and shifted," the current trend is to split large code bases into smaller microservices. This, combined with the unique opportunities and challenges presented by a cloud environment, should convince architects, developers, and operators that lightweight documentation is essential in order to communicate our design and intentions.

The key purpose of documentation in a cloud-based project is to provide a map of the application and deployment territory. Architectural views should be captured in diagrams, especially if the application has been adapted architecturally to take advantage of cloud features, such as elastic scaling or multi-region failover, as these features necessarily impact the runtime operation of a system. Simon Brown's C4 model of architecture provides a great start in learning how to create such diagrams.

The creation of a physical infrastructure diagram is also essential. This should highlight operational features that are important to the ops, dev, and QA teams. For example, private subnets, firewalls, load-balancers, replicated data stores, and service clustering all have a large impact on the design and configuration of software.

Other essential documentation for a cloud application includes a list of software components, their purpose (and contract), and initialization instructions. Any discrepancies between the approach used to run services in a local development environment in comparison with the production cloud stack should be clearly noted. Brief documented highlights of service state and caching are also often useful, and provide essential information during configuration and debugging sessions. All of this information can typically be summarized in a code repository README file, and suggested section headings for this documentation can be found below:

- Component description responsibilities
- Component initialization instructions
- Profiles available (i.e., modes of operation)
- Component external interactions (i.e., collaborations)
- State characteristics (i.e., stateless, mandatory sticky-sessions)
- Data store and cache interactions (in/out of process, eviction policy, etc.)
- API documentation (e.g. Swagger, Thrift IDL, etc.)
- Failure modes of component
- Developer highlights (i.e. classes of interest)
- Decision log (core architectural changes)

HIGHLY COHESIVE/LOOSELY COUPLED - GOOD ARCHITECTURE ALL THE WAY DOWN

Creating an appropriate architecture that supports the functioning and evolution of a software application is essential. regardless of the deployment environment, but is especially important when building software that will take advantage of the flexible (but volatile) cloud fabric. Anyone performing a "lift and shift" will potentially have less options for establishing a good architecture, but an analysis should at least be performed, and any areas of the system that may cause friction with a cloud environment should be noted. For example, the volatile nature of the cloud means that services that are highly coupled to a dedicated IP address can cause problems, as can services that are not capable of being clustered or surviving a restart.

A good architecture is evident "all the way down" a software system, and in a cloud application this will most likely include the public-facing API, services (and internal APIs), components, and the code itself. The core measures of cohesion and coupling can be applied at every level. It is difficult to escape the current trend toward implementing a system by decomposing functionality into small services, and architectural guidelines such as those presented at 12factor.net and the book Building Microservices are well worth understanding. Martin Abbott and Michael Fisher's Scalability Rules is also essential reading for any cloud developer.

AUTOMATED FROM COMMIT TO CLOUD - THE PIPELINE TO PRODUCTION

The creation of a build pipeline that supports continuous delivery is beneficial for many software projects. This is especially important when developing applications for the cloud, as typically developers will create software using an environment that has radically different hardware and operating system configuration than the production stack. The goal of a build pipeline is to take code committed to a version control systems such as Git, create the corresponding build artifacts and provision the required infrastructure, and exercise (or "torture") the resulting applications. Only artifacts that survive their journey through the complete build pipeline are flagged as being suitable for production

continued

deployment. A new version of the software emerging from the pipeline should ideally add value to end-users, and must definitely not introduce any regressions in existing functional or cross-functional requirements. Jez Humble and Dave Farley's Continuous Delivery book is an essential reference for learning more about the benefits and implementation of a build pipeline.

A build pipeline for a cloud project must push the modified application into a cloud environment as soon as possible. This cannot be emphasised enough! It is not uncommon for developers to be creating software on an ultra-powerful laptop running Mac OS X, but the corresponding production infrastructure is a micro-powered cloud instance running Debian. It should go without saying that performance (and potentially even functionality) will be very different across these two environments. Accordingly, running automated acceptance and performance tests against a cloud-based deployment of the application is essential.

Cloud infrastructure and compute resources are often defined with provisioning/configuration management utilities (such as Chef, Ansible, Puppet), and the resulting artifacts may be packaged as images or containers (with Packer or Docker). This infrastructure as code should also be tested as part of the build pipeline. Vendor-specific testing tools, such as Chef's Foodcritic and Puppet's rspec-puppet, are the operational code equivalent of unit and integration testing, and tools such as ServerSpec can be used for acceptance testing infrastructure deployed into a cloud environment. It is also beneficial at the infrastructural testing phase of the build pipeline to follow the example of simulate failure modes that can be experienced in a heavily virtualized and networked environment.

RESOURCE AWARE - NOISY NEIGHBOURS AND MECHANICAL SYMPATHY

A core change when moving to develop software for the cloud is the fact that practically everything within this environment is virtualized, and the vast majority of communication occurs over a network. At first glance this may appear to be an obvious statement, but developers are often used to creating software on computing infrastructure that is not multi-tenant. When software is running on compute resource that is not shared, there are no "noisy neighbours" that are attempting to compete for contended physical resources, such as the CPU, memory, and disk access. In addition, traditional applications typically communicate to a database over dedicated low-latency network channels, but in the cloud this channel may be shared by many other services. In high availability configurations, it is not uncommon for a database to be running in a different data center than the machine making a request (although this should be avoided), and this only adds to the communication overhead.

Werner Vogels, CTO of Amazon, is famous for stating that "everything fails all the time in the cloud." The flexibility and cost benefits of using virtualized commodity hardware within a public cloud has a clear trade-off—every infrastructure resource must be treated as ephemeral and volatile. The challenges introduced by using cloud fabric must be countered by cultivating "mechanical sympathy," or put another way, developing an understanding of the hardware fabric onto which you are deploying applications. Key skills that every developer, QA specialist, and operator must develop when deploying applications to the cloud include:

- Deep understanding of virtualization—Hypervisors, steal time and resource contention
- Good comprehension of computer networking—TCP/IP, DNS, and the OSI model
- Good knowledge of caching—Reverse proxies, distributed caches, and CDNs
- Expert Linux skills—Including diagnostic tools like top, vmstat and tcpdump

A good architecture is evident "all the way down" a software system.

MONITORED THOROUGHLY - IF IT MOVES, GRAPH IT...

Potentially the biggest operational challenge with moving an application to the cloud is the new monitoring infrastructure required. Although a good monitoring strategy and implementation is beneficial in any environment, it is essential with a volatile, multi-tenant and virtualised fabric such as that offered by a public cloud. Basic operational details about each compute instance should be monitored, for example utilizing collectd or Munin to collect CPU usage, memory statistic and disk performance. This data can then be shipped to time-series datastores such as InfluxDB or OpenTSDB, graphed by tools such as Grafana or Cacti, and used for on-call alerting by applications such as Nagios or Zabbix. Data stores and middleware should also be monitored, and modern applications typically expose these metrics out of the box via a series of mechanisms. For example, MySQL exposes a proprietary API, Solr exposes statistics via JMX, and RabbitMQ exposes data via an HTTP interface.

All software applications running on these compute resources should also be monitored, whether they have been developed in-house or not. Key metrics for an application can be exposed using frameworks such as Codahale's Metrics or StatsD. These frameworks allow developers to specify status flags, counters, and gauges in code, which can emit information such the current service health status, transaction throughput, or cache statistics. If you are operating in a microservice environment, then distributed tracing is also essential in order to learn how each ingress request is handled as it passes through your application stack. Tools such as Twitter's Zipkin or AppDynamics Application Performance Monitor are prime candidates for this. Finally, centralized logging is also essential in order to avoid having to manually SSH into multiple locations, and the ELK stack of Elasticsearch, Logstash and Kibana is becoming a de facto standard for achieving this goal (and can easily be experienced via a Vagrant box).

ANTIFRAGILE - ROBUST IS NOT THE OPPOSITE OF FRAGILE

The fabric of the cloud provides a set of unique challenges in terms of increased contention, volatility, and transience. Software must clearly be designed to handle the "fragile" nature of the underlying infrastructure. The initial design of a system to counter fragility often leads to the creation of a robust system, which can typically withstand heavy load and recover from failure. However, the methods by which a robust system achieves these properties may not be optimal for the post-Web 2.0 generation of users, who expect applications to be constantly available and highly responsive.

A robust web server that simply refuses any connection over a preset maximum limit will provide a good experience for those already connected, but what about the additional users who can't connect? The same can be said from an e-commerce site where failure in the recommendation component automatically shuts down the entire site—a user will never receive an incorrect recommendation; but is this really the correct response to the problem that the business would expect (or want)?

Applications must go further than being robust, they must be "antifragile." As a foundation for antifragility, cloud-native software must be created using fault-tolerant design patterns such as timeouts, retries, bulkheads, and circuit breakers. Michael Nygard's book "Release It!" provides a great overview of all of these concepts.

The elastic nature of compute resources truly allows antifragile behavior. Software can be designed to take advantage of this elasticity by rapidly scaling compute power to meet increased demand or reduce costs. The vast majority of cloud vendor APIs and SDKs make this operation trivial, and many vendors provide an automated approach to handling this common use case. Additional tools for creating an antifragile application include using message queues, such as RabbitMQ or Kafka, which enable asynchronous communication and the buffering of events, and the introduction of eventual consistency into datastores, for example using Cassandra or Riak.

Software must clearly be designed to handle the "fragile" nature of the underlying infrastructure.

As a first step towards understanding the principle of antifragility, it is well worth consulting the classic texts on distributed computing principles, such as the eight fallacies of distributed computing and notes on distributed systems for young bloods. Netflix is the poster child of antifragility within software, and every cloud developer should take a tour of their public Github account, if only to take inspiration from the plethora of solutions provided.

SUMMARY - THE DHARMA CHECKLIST

This article has attempted to identify the common challenges experienced when developing and deploying software applications to the cloud. The cloud DHARMA principles are designed to act as a checklist when designing and implementing software that will be deployed onto a cloud environment. The principles state that attention must be focused on documenting architecture and deployment topologies, utilizing good software design principles "all the way down" the stack, creating a comprehensive build pipeline, increasing awareness of the underlying fabric of the cloud, implementing a comprehensive monitoring strategy, and building antifragile systems. Although there are many challenges encountered when developing applications for the cloud, the benefits make this an attractive environment for many software applications. Simply remember that the cloud is a completely different animal than a traditional datacenter.

The content of this article was originally presented at JavaOne 2014, and the recording can be found on Parley's.

DANIEL BRYANT (@danielbryantuk) is a Principal Consultant at OpenCredo, a software consultancy and delivery company dedicated to helping clients deliver better software faster. Currently, Daniel specializes in continuous delivery, DevOps methodologies, and leveraging container-based platforms such as Docker, Mesos and Kubernetes. He is also a leader within the London Java Community, and a contributor to the Java Community Process (JCP).

BE AGILE,
BE FLEXIBLE.
ACCELERATE TO MARKET.

Don't compromise on performance.
Bring your Java EE apps to the cloud, fully supported.

Learn how with Red Hat® JBoss® Enterprise Application Platform for xPaaS.

redhat.com/accelerate

redhat.

5 Critical Questions to Ask Before Selecting a PaaS

When you're constrained by legacy applications in traditional IT environments, it's hard to keep up with fast-changing markets and rapidly deliver new products and services.

Some organizations turn to DevOps and a bimodal IT structure to increase agility while maintaining quality, reliability, and security. They also seek technology to quickly build high-value, web-scale, enterprise performance applications.

One such technology—Platform-as-a-Service (PaaS)—promises developers the ability to produce new code fast while giving operations control over production performance.

Before choosing a PaaS, ask yourself these 5 questions:

1. **Will it provide agility and a low-risk way to experiment and find our competitive edge?** Choose a PaaS that offers developers a choice of tools, frameworks, languages, and the flexibility to innovate.

2. **Will it provide immediate and long-term performance, reliability, and security?** Most enterprise applications need more than just web platform-level support as your critical applications mature.

> YOUR APPLICATIONS MAY GO THROUGH FREQUENT CHANGES, BUT YOUR PAAS TECHNOLOGY SHOULD NOT

3. **How transferable are our developers' existing skills, expertise, and code?**

4. **Does it provide value beyond deploying applications?** Choose a single platform that supports middleware services such as the application platform, developer tools, integration services, and mobile support.

5. **Does the vendor have a reputation for agile, flexible solutions, and predictable costs?** Your applications may go through frequent changes, but your PaaS technology should not.

WRITTEN BY CHRISTINA WONG
PRINCIPAL PRODUCT MARKETING MANAGER, RED HAT JBOSS MIDDLEWARE, RED HAT

Red Hat JBoss Enterprise Application Platform for xPaaS by Red Hat

PaaS

redhat.

JBoss EAP for xPaaS is the only Java EE 6 full platform certified application platform offering commercially supported for deployment in private and public PaaS environments.

SERVICES PROVIDED

- Virtual Machines
- Routing, Queueing, and Scheduling
- Application Containers
- aPaaS
- DBaaS

HOSTING OPTIONS

- Private by provider
- Private on-premise
- Hybrid

SECURITY OPTIONS

- Keypair-based authentication
- Configurable role-based permissions
- SSH connections
- SSL connections
- SAML
- Oauth

LANGUAGES

JAVA

CUSTOMERS

- Cigna
- U.S. Department of Defense
- CenturyLink
- New Zealand Department of Internal Affairs
- EZ Systems
- Ice.com
- Fonecta

CASE STUDY

French insurance broker Filhet-Allard was using highly disparate and independent IT systems, which made it difficult to deliver new applications to customers. To bring greater agility, integration, and connectivity, they decided to use a Software-as-a-Service (SaaS) model to deliver their business applications. For both the performance and business benefits, they chose a Red Hat stack that includes Red Hat Enterprise Linux, Red Hat JBoss Enterprise Application Platform, and other middleware products. Filhet-Allard is now able to quickly and seamlessly deliver white-label business applications to their customers. Setting up the front office platform in a private cloud led to greater standardization, which helped make provisioning and delivering development environments simple—and less expensive.

BLOG planet.jboss.org | TWITTER @jboss | WEBSITE redhat.com/jboss

7 Rules for Hybrid Cloud Architectures

QUICK VIEW

01
In the next couple years, research experts predict that hybrid cloud adoption rates will greatly increase.

02
Consider your operational footprint, network operations, data processing efficiency, network security, application performance, cloud provisioning, and testing.

BY JP MORGENTHAL

Research firm IDC predicts that by 2016 more than 65% of enterprise IT will commit to hybrid cloud technologies. IDC is not alone in predicting a high rate of enterprise adoption for hybrid cloud computing, Gartner Group also has predicted that 50% of enterprises will have hybrid cloud deployments by 2017, and DZone's 2015 Cloud Development Survey revealed that 50% of their audience is currently using hybrid cloud technologies.

Hybrid cloud architecture is often described as 'private and public clouds sharing resources'. But in reality, many hybrid architectures merely leverage public cloud resources in tandem with privately-hosted applications. For example, a business might capture, aggregate and analyze data from multiple external sources on a public cloud and then pass those results to an application running in a privately-hosted environment. Or a business might put its public web presence in a public cloud but keep the data for that application in a privately-hosted environment.

This public/private cooperation introduces potential complexities. This article will present seven rules to consider when adopting hybrid cloud architectures.

RULE 1

YOU ARE EXTENDING YOUR OPERATIONAL FOOTPRINT

One of the most critical things to keep in mind when deploying a hybrid cloud architecture is that you are extending your operational footprint. While this may seem obvious, it may have a real impact on IT operations and development organizations. Consider the following questions:

- Which team is responsible for the components that are running in the public cloud?
- Is your IT operations team prepared to manage another platform?
- Can your current monitoring and operations tools be used with the public cloud provider?
- How is this architecture going to impact calls to the service desk?
- What is the plan for network interruption between public cloud and data center?

As you can see, hybrid architectures may require you to hire individuals that have the right skills to operate on the public cloud platform selected. Likewise, it may require acquisition of new monitoring tools if the current tools do not support the cloud platform selected. Some organizations may choose to use a "NoOps" style of approach in which the development team is responsible for the application components that operate on the public cloud — an approach that removes the burden from IT operations, but leads to the need to define new operational processes for the business.

The key takeaway here is that the decision to go hybrid may seem simple because of the nature of on-demand computing, but should not be taken lightly by the business.

RULE 2

DON'T SHARE YOUR NETWORK WITH YOUTUBE AND FACEBOOK

Make sure you work with your network operations team to identify networking requirements surrounding your hybrid applications. Sometimes the only collaboration that occurs is that development requests that a port be opened to the public Internet so that the privately-hosted application can receive requests from public cloud applications.

If network operations is not informed about the need for the connections to and from your public cloud to have a guaranteed quality-of-service, it is possible that your application traffic will be sharing bandwidth with fellow employees watching cat videos on Facebook. Your network operations team needs to be an integral part of the team deploying your hybrid cloud application so that they can focus on appropriate connectivity and traffic shaping.

RULE 3

AVOID REVERSE DATA GRAVITY

Simply stated, "data gravity" is the theory that processes should migrate to the data given the relative mass of each. That is, processes are fairly lightweight to move when you consider that they operate on terabytes or petabytes of data. However, hybrid cloud architectures sometimes suffer from "reverse data gravity" because using a hybrid cloud means keeping (some) data on privately-hosted environments. That is, the processes hosted in the public cloud draw data from the private side. This is an architectural choice that limits risk, but at some point the business needs to identify what is an acceptable quantity of data that can be moved before reaching a point of diminishing returns. Further impacting reverse data gravity is the choice to pull data through services or directly connect to data sources on the private side. The latter will most likely result in higher quantities of data being transferred.

Additionally, since you now have application components running on the public cloud, data is being generated on both public and private clouds. Specifically, logs are being generated by activities on the public cloud — which means new data to be managed and analyzed.

RULE 4
TRUST ONLY GOES SO FAR

This is also known as the Captain Obvious rule. But the truth is that sometimes businesses make poor choices in favor of simplicity. For example, some businesses have actually stored private keys on the private cloud that enable connectivity back to the privately-hosted environment. This opens up significant risk of a breach of the private environment. If keys are needed, they should be stored encrypted in a third-party repository and pulled by the process when needed and erased when no longer required. This separation of concerns significantly increases the attack surface.

Here are some additional best practice recommendations:

- Limit the number of endpoints that can establish secured connections with the data center
- Limit access from Internet-enabled servers in the public cloud
- Traffic leaving secured areas on public cloud should go through Network Address

Note that the NAT server offers a higher degree of security by limiting access into private network areas on the public cloud. But this is also a traffic bottleneck and single point of failure, so high availability and load balancing solutions may be required to obtain the necessary performance.

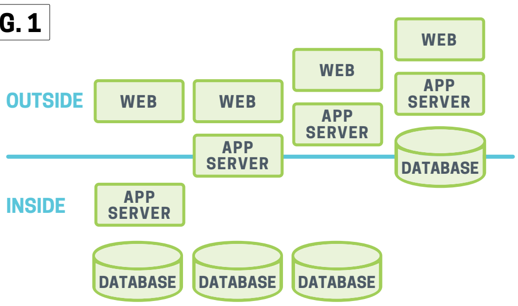
RULE 5
APPLICATION REDESIGN MAY OFFER BETTER PERFORMANCE

If we consider the traditional three-tier web application as a good model for hybrid architectures, there is a natural inclination to have the web interface on the public side and the application server and database on the private side. However, this architecture may not offer the best economics or performance for the application.

Sometimes it is advantageous to redesign the application to take better advantage of public cloud services while still offering the benefits of hybrid cloud architecture. As depicted in Figure 1: moving left to right, we can see various parts of the application start to shift to the public cloud. The first transition illustrates splitting up the application server so that some services may be running local to the web server while others are running local to the database, which is a good trade-off if there is a division of compute functions and data management functions.

The far right example represents moving everything but the data onto the public side, but perhaps using caching techniques or NoSQL databases to store data temporarily so that compute functions can respond more effectively. These modifications have the effect of reducing latency, being more responsive to user interactions, and limiting the amount of data that needs to be transferred back to the private side.

FIG. 1



RULE 6
DON'T TREAT PUBLIC CLOUD LIKE ANOTHER DATA CENTER

It's human nature to revert to what we know when in unchartered or unfamiliar territories. For operations personnel, even if they have training on the cloud platform, they are going to be most familiar with how they've run things for the past few years of their careers. This means overprovisioning, which in cloud speak means subscribing to more virtual machines and storage than you really need. It could also mean that they choose to deploy and manage all the software components versus using services from the public cloud provider, such as relational database and queuing services.

The value of the public cloud is more than just availability of resources on demand. Public clouds also let you obtain these resources at a good economic value. Taking advantage of things like discount policies for reservations is a better alternative than acquiring everything on-demand. Using tools and services that provide high availability and failover options can be much more effective, and less costly, than building in these capabilities manually. Since these options don't exist in the privately-hosted environment, operations is often forced into a bimodal situation requiring different approaches to managing resources. For many, the duality can be disconcerting and disorienting.

RULE 7
TEST, TEST & THEN, TEST AGAIN

Testing your hybrid cloud architecture requires an understanding of common issues for building distributed applications. Networks have gotten so reliable that sometimes we take for granted that our packets will arrive at their destination. However, as reliable as networks have gotten, transmission errors still occur due to hardware failures or increased traffic. The key is to develop the appropriate resiliency into your application in face of these failures.

Application developers have spent years dealing with reliability issues within in the data center, which can rely on fiber optic connections and high-speed connectivity. When we move to a hybrid architecture, we often are communicating over slower connections and sharing bandwidth with other applications. Hence, we need some additional testing to identify how our application will respond.

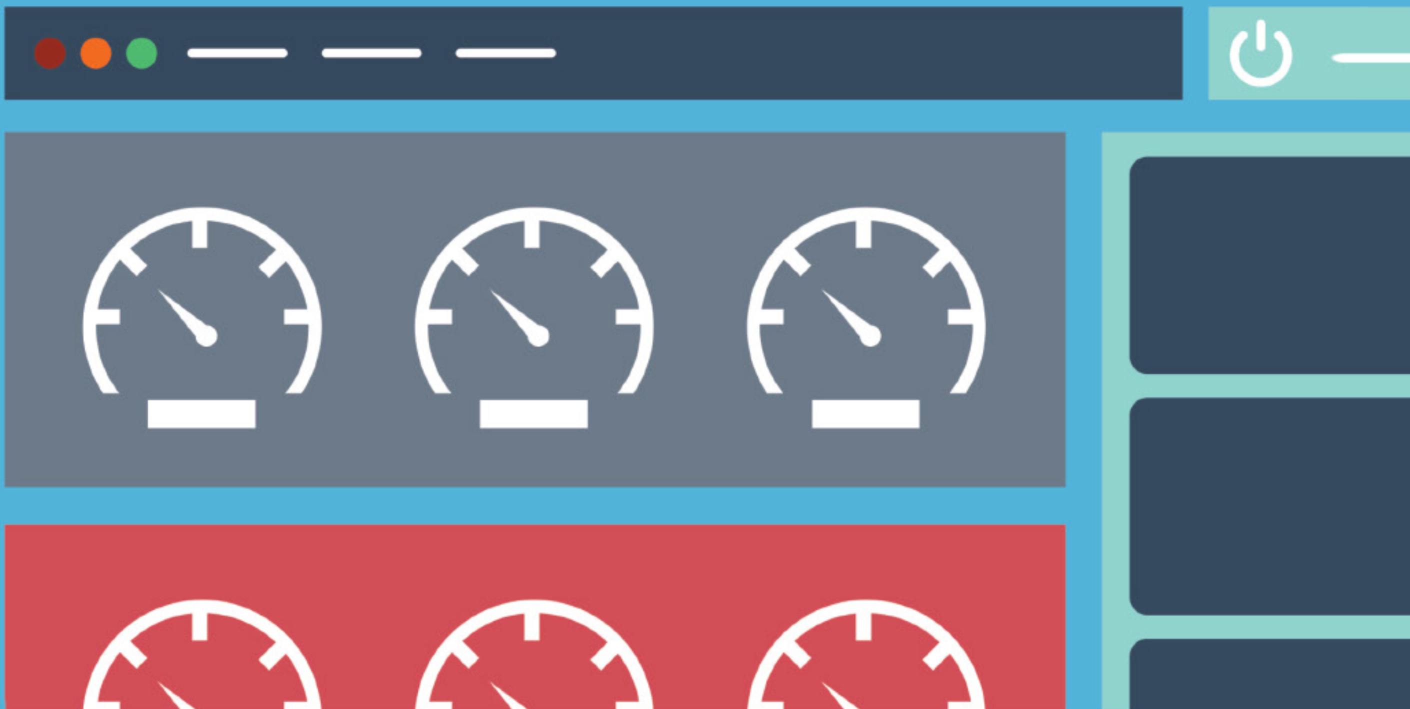
Here's a list of additional tests we recommend for hybrid applications:

- Network failure testing
- Increased latency testing
- VM server failure testing
- Invalid message testing (for services)
- Authorization testing
- Authentication testing

CONCLUSIONS

As I like to say: "Hybrid cloud architectures are easy ... until they're not." A lot of hidden complexities are involved in managing an enterprise-scale, robust, hybrid cloud deployment. Methodologies for building applications that operate in silos within a data center and those that cross geographic boundaries over common Wide-Area Networks cannot be expected to behave identically nor can they be operated in the same manner. Following the advice put forth in these seven rules, I hope, will help you deliver resilient hybrid applications.

J.P. MORGENTHAL is an internationally renowned thought leader in the areas of IT transformation, modernization, and cloud computing. J.P. has served in executive roles within major software companies and startups. His expertise includes strategy, architecture, application development, infrastructure and operations, cloud computing, DevOps, and integration. He advises C-level executives on the best ways to use technology to derive business value.



Going Rogue with PaaS:

Bringing Shadow IT into the Light

I recently suggested that it's okay to install PaaS on a couple of PCs, and run them "under the desk" for cloud development. This provoked a comment, "You're not endorsing Shadow IT, are you?" Well, in fact I am.

A common practice is configuring a PC with a bunch of memory then putting it under a desk and making it available to the whole team for spinning up VMs. There are good reasons why this is so common. In many environments, if a team needs a cluster of VMs, they must submit multiple service tickets, then wait days, weeks, or even months before the new VMs are available. Restrictions on networks, software, and behaviors might prohibit the cluster entirely. Rogue IT is just so darned simple in comparison: self-service, on-demand, and elastically scalable resources.

There are several reasons why the PC under the desk is not encouraged by IT. If the person who built it were to vanish, it would take time and effort to figure out how it was configured. Then, after the development period is over, the process to move whatever was built on it to IT-supported servers is unclear.

These issues can be resolved using Platform-as-a-Service (PaaS). With PaaS installed on the PC, there's less risk if the person leaves, since it would take under an hour to rebuild it all again. The PaaS on the PC is configured exactly the same as the PaaS running in the data center or cloud, so there is no unsanctioned software, and moving software from the PC PaaS to the cloud PaaS is trivial.

> **IT'S EASY TO PROVE OUT PAAS IN A SHADOW IT FRAMEWORK AND THEN MOVE IT INTO A MORE STRUCTURED USE ONCE ITS VALUE IS EVIDENT.**

It could be argued that most IT innovation today is happening in the shadows of IT. It's best to embrace it, and PaaS is the perfect enabler. It's easy to prove out PaaS in a shadow IT framework and then move it into a more structured use once its value is evident, as it surely will be.

Read more of John's blog posts.

WRITTEN BY JOHN WETHERILL
TECHNOLOGY EVANGELIST,
ACTIVESTATE

Stackato by ActiveState

PaaS

Stackato is a secure, stable and commercially supported PaaS built on Cloud Foundry, Docker and other proven open source technologies.

SERVICES PROVIDED

- Virtual Machines
- Routing, Queueing, and Scheduling
- Application Containers
- aPaaS
- DBaaS

HOSTING OPTIONS

- Private by provider
- Private on-premise
- Hybrid

SECURITY OPTIONS

- Keypair-based authentication
- Configurable role-based permissions
- SSH connections
- SSL connections
- SAML
- Oauth

CUSTOMERS

- ExactTarget (a Salesforce.com company)
- Dirk Rossmann GmbH
- Mozilla
- Angie's List
- HP
- Cisco
- MTN Communications
- Nelnet

LANGUAGES

JAVA RUBY PYTHON PERL PHP NODE.JS GO GROOVY

CASE STUDY

Service-Flow is a leading provider of SaaS for IT service integration, trusted by financial services, insurance and media companies. To enable rapid deployment of Java applications without having to manage their own servers, Service-Flow decided to run their service on a public PaaS. However, they soon moved to Stackato, a private PaaS supporting the cloud of their choice, AWS, in order to gain more control of the underlying resources. With Stackato, Service-Flow achieves 400% faster deployment cycles using microservices and continuous delivery, portability to any cloud, high availability using Stackato's blue-green deployment capabilities, and deploys software updates every day, often several times per day.

BLOG blog.activestate.com

TWITTER @activestate

WEBSITE stackato.com

THE CLOUD SERVICES LANDSCAPE

CLASSIFIED BY LAYER AND USE

The great thing about *aaS: there is (or will be) a well-managed service for pretty much everything. The frustrating thing about *aaS: it's hard to tell which particular services play how nice with which other services, at which layers, for which purposes. So we've painted a nice little picture to help developers navigate the cloud services soup.

Major service providers are listed below, classified along two dimensions: layer and use.
(The majority of providers could not fit into this graphic. For many more see the Solutions Directory at the end of this Guide.)

	COMPUTING	COMMUNICATION	STORAGE
MODEL-DRIVEN LAYER	**MODEL-DRIVEN PaaS** mx mendix the app platform salesforce PROGRESS	**MODEL-DRIVEN iPaaS** Microsoft Azure DELL Boomi mx mendix the app platform MuleSoft	**BUSINESS ANALYTICS PaaS** birst GoodData ORACLE MicroStrategy
PAAS LAYER	**aPaaS & LANGUAGE RUNTIMES** Engine Yard CLOUD FOUNDRY OPENSHIFT by Red Hat stackato by ActiveState salesforce heroku	**iPaaS, EsBaaS, MESSAGING** amazon web services DELL Boomi MuleSoft	**DBaaS** VERTICA SAP ObjectRocket ORACLE lunacloud
FOUNDATIONAL LAYER	**APPLICATION CONTAINERS** docker heroku CLOUD FOUNDRY OPENSHIFT by Red Hat WSO2	**ROUTING, MESSAGING, SCHEDULING** informatica snapLogic apache Stratos	**OBJECT STORAGE** lunacloud Parse HP Helion amazon web services Joyent
IAAS LAYER	**VIRTUAL MACHINES** amazon web services openstack Google Compute Engine Microsoft Azure DigitalOcean	**SDN** nicira OPEN DAYLIGHT JUNIPER NETWORKS CISCO	**BLOCK STORAGE** virtustream PROFITBRICKS The IaaS-Company rackspace HP Helion

Why Run Your Microservices on a PaaS

BY CHRIS HADDAD

QUICK VIEW

01

The microservice approach creates a more complex, loosely coupled, and dynamic environment that distributes business capabilities all over the network.

02

Evolve your monolithic design by applying service-oriented principles in conjunction with domain-driven design techniques and dynamic runtime application composition.

03

Choose a PaaS environment that can automatically deploy, provision, and link full-stack microservices.

Micrososervices can be understood from two angles. First, the differential: teams that take a microservice design approach divide business solutions into distinct, full-stack business services owned by autonomous teams. Second, the integral: microservice-based applications weave multiple atomic microservices into holistic user experiences.

Unfortunately, traditional application delivery models and traditional middleware infrastructure do not address microservice-specific demands for on-demand provisioning, dynamic composition, and service level management. On the other hand, the Platform-as-a-Service (PaaS) model addresses these demands perfectly. Running microservices on a PaaS fabric decreases solution fragility, reduces operational burden, and enhances developer productivity.

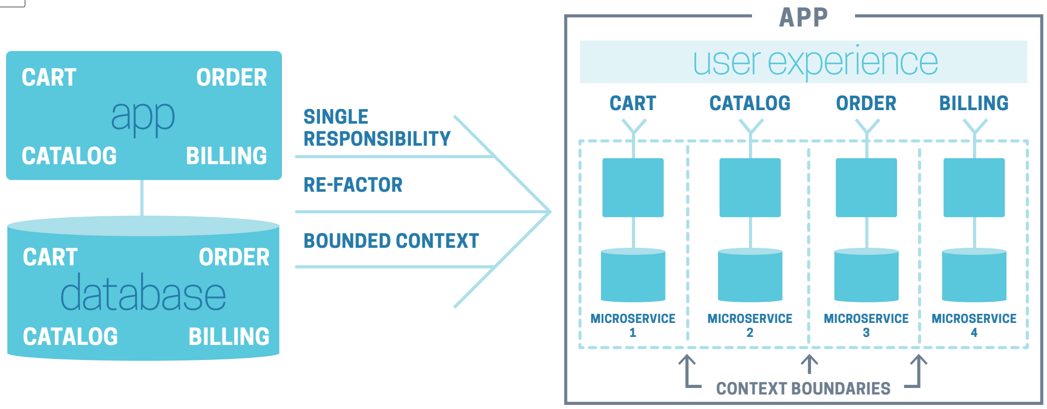
To understand why, we'll first review how microservices separate concerns from both business and object-oriented design perspectives. Second, we'll consider how microservice-based design can complicate deployment as applications scale dynamically. Third, we'll focus on how a PaaS environment helps to solve many of the problems both addressed and introduced by microservices-based architectures — in other words, why PaaS and microservices are a match made in heaven.

MICROSERVICES: SEPARATING CONCERNS BY BUSINESS SOLUTION

A microservice approach decomposes monolithic applications according to the single responsibility pattern. In a microservice solution, each microservice interface delivers discrete business capabilities (e.g. customer profile, product catalogue, inventory, order, billing, fulfillment) within a well-defined, bounded context. The atomic microservice interfaces reside on separate and distinct full-stack application platforms that contain separate database storage, integration flows, and web application hosting. By separating concerns onto separate full-stack platforms and not sharing database instances or web application hosts across services, every team is free to choose different runtime languages and frameworks for its own microservice. Also, every team is free to evolve its data schemas, application frameworks, and business logic without impacting other teams.

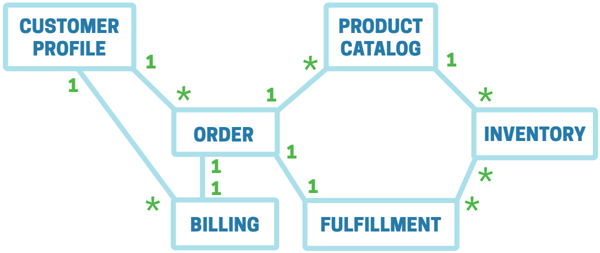
Because microservices are a relatively new design approach, many development teams may have the misconception that creating a microservice-based solution requires simply deploying small web services in containers. But this doesn't cut quite deep enough. The correct approach is to evolve your monolithic design by applying service-oriented principles (i.e. encapsulation, loose coupling, separation of concerns) in conjunction with domain-driven design techniques and dynamic runtime application composition.

FIG. 1



For example, in a typical ecommerce scenario, a development team applies the bounded context pattern and single responsibility pattern to refactor a monolithic application into units distinguished by business capability (see Figure 2). By creating a user experience from loosely coupled services instead of tightly coupled native-language business objects, teams have more independence to develop, evolve, and deploy each business capability separately. Obviously, the microservice design approach works best for (a) greenfield projects or (b) modernization efforts where teams focus on refactoring monolithic application assets.

FIG. 2



THE MICROSERVICE EXECUTION TRAP

Although a microservice approach decouples development dependencies and speeds up development iterations, microservices also create a challenging environment for high-performance scaling and reliable runtime execution. More complex, loosely coupled, and dynamic environments distribute business capabilities over the entire network. Even a task as simple as responding to a single web application page request may spread out across several microservice instances residing on a distributed network topology. Martin Fowler and Stefan Tilkov (both microservice proponents) warn teams that successfully implementing a microservice approach requires choosing platforms that decrease solution fragility and reduce operational burdens.

WHAT PLATFORM-AS-A-SERVICE OFFERS

Platform-as-a-Service environments reduce microservice operational burdens when infrastructure-as-code and declarative policies are used to eliminate all manual actions and increase runtime quality of service (i.e. reliability, availability, scalability, and performance). The appropriate PaaS environment will automatically deploy, provision, and link full-stack microservices.

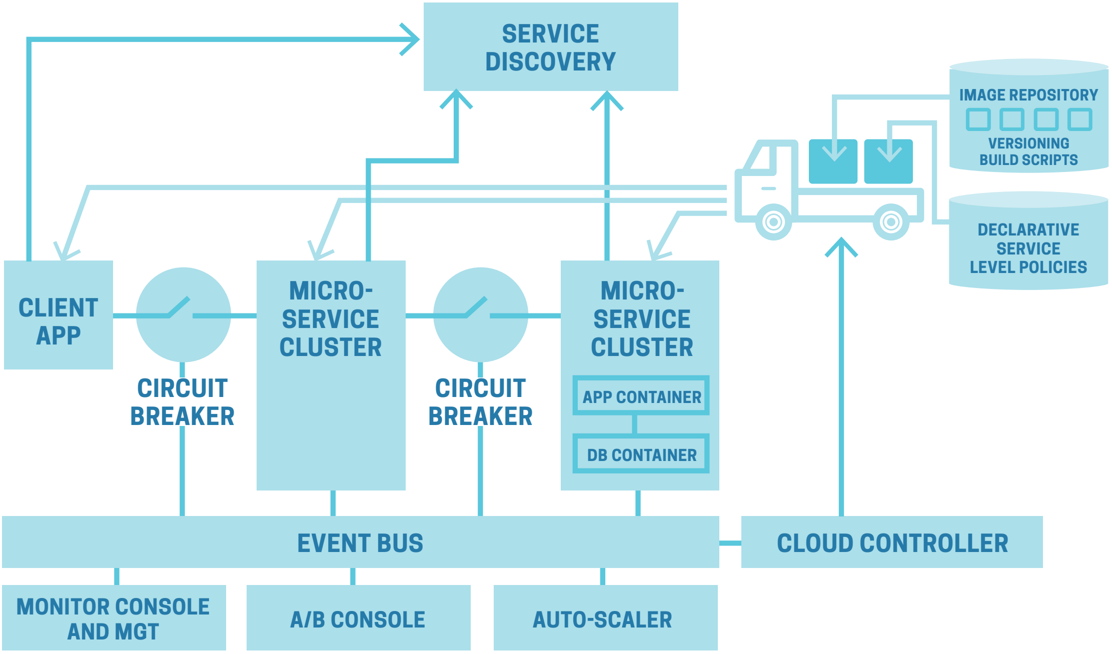
In a microservice architecture, teams want to rapidly release new versions and perform A/B testing across versions. When teams define instance dependencies, scaling properties, and security policies as PaaS metadata or code scripts, the runtime fabric can reduce manual effort and increase release confidence. With a DevOps-friendly PaaS, the team can experiment with new service versions and safely rollback to a prior stable release if a problem arises. Because microservices are full-stack silos that can be composed of multiple server instances (e.g. web server, database, load balancer, integration server), a PaaS can reduce deployment complexity by automatically spinning up and linking all instances. Linking may require discovering instance locations, dynamically initializing network routes, and auto-configuring connection strings based on service version or tenant.

A traditional application will compose business functions and user experience by statically linking class files and shared object libraries. In contrast, microservice-based applications use service composition to connect available microservices endpoints and realize a fully functional application. While many microservice proponents promote microservice-based interactions by “smart endpoints through dumb pipes, ‘ effective service composition requires smart infrastructure building blocks to bootstrap and maintain connections between services and consumers.

The right PaaS solves these problems. Infrastructure building blocks will register service endpoint locations,

continued

FIG. 3 SMART ENDPOINTS AND SMART SERVICE FABRIC



associate metadata and policies, connect clients, circuit break around failures, correlate inter-service calls, and load balance traffic. A microservice-friendly PaaS will provide service registries, metadata services, discovery services, and service virtualization gateways. In the pipe, circuit breakers will automatically route traffic on failover or overload.

Smart endpoint code will dynamically connect with microservices based on discovery service responses and negotiated quality of service parameters. Rather than being hard-coded to a specific service hostname and URI, endpoint code will query for microservice location based on security assurances, performance guarantees, traffic load, service version, client tenancy, or business domain. When services are unavailable or underperform, smart endpoints will follow the tolerant reader pattern and gracefully degrade experience or proactively recover. A few recovery options include reading from local caches or circuit tripping to backup service endpoints. In conjunction with smart endpoint actions, a smart PaaS will spin up new microservice endpoints and full-stack instances based on service level management metrics. By following microservice architecture best practices, teams create anti-fragile applications that not only withstand a shock, but also improve performance and quality of service when stressed or experiencing failures. To drive this non-intuitive behavior, the underlying platform environment must be ready to scale, repair, and reconnect services. PaaS service level management components will create more resilient and anti-fragile microservices by monitoring performance, elastically provisioning instances, and dynamically re-routing traffic.

Scaling an anti-fragile microservice is more difficult than scaling a web application. The PaaS should distribute microservice instances across multiple availability zones and dynamically adjust traffic to reduce latency and response time. Because transient microservice instances will rapidly start, stop, and change location, the service management layer must be completely automated and integrated with routing services.

A PaaS environment will deliver the service level management, dynamic service composition, circuit breakers, and on-demand provisioning functions required to overcome the complexity inherent within a distributed microservice-based application architecture. Running microservices on a PaaS fabric will decrease solution fragility, reduce operational burden, and enhance developer productivity. If you are pursuing a microservice design approach, make sure you choose a microservice-friendly PaaS.

CHRIS HADDAD A cloudy PaaS and SaaS architect since the turn of the century, Chris is an Apache Stratos and cloud architecture aficionado who has worked with clients to deploy innovative ecosystem PaaS environments. Chris has led many successful startup software teams who operated as the provider of choice. He has advised Fortune 500 / Global 2000 organizations on software strategies, roadmaps, and best practices.

diving deeper

INTO CLOUD DEVELOPMENT

top 10 #cloud twitter feeds TO FOLLOW RIGHT AWAY

@DAVIDLINTHICUM

@SAMCHARRINGTON

@RANDYBIAS

@AARONDELP

@CHRISTIANVE

@JEFFBARR

@RUV

@BGRACELY

@MARTENMICKOS

@WERNER

cloud zones LEARN MORE & ENGAGE YOUR PEERS IN OUR CLOUD-RELATED TOPIC PORTALS

Cloud Zone

dzone.com/mz/cloud

The Cloud Zone covers the host of providers and utilities that make cloud computing possible and push the limits (and savings) with which we can deploy, store, and host applications in a flexible, elastic manner. This Zone focuses on PaaS, infrastructures, security, scalability, and hosting servers.

DevOps Zone

dzone.com/mz/devops

DevOps is a cultural movement, supported by exciting new tools, that is aimed at encouraging close cooperation within cross-disciplinary teams of developers and IT operations/system admins. The DevOps Zone is your hot spot for news and resources about Continuous Delivery, Puppet, Chef, Jenkins, and much more.

Big Data Zone

dzone.com/mz/big-data

The Big Data/Analytics Zone is a prime resource and community for Big Data professionals of all types. We're on top of all the best tips and news for Hadoop, R, and data visualization technologies. Not only that, but we also give you advice from data science experts on how to understand and present that data.

top cloud refcardz

Getting Started with Cloud Computing

bit.ly/DZ-CLOUD101

Cloud Foundry

http://bit.ly/DZ-CloudFound

Open Stack

bit.ly/DZ-OpenStack

Scalability & High Availabilty

bit.ly/DZ-Scalability

top cloud websites

Cloud Tweaks

cloudtweaks.com

Cloudscaling

cloudscaling.com/blog

All Things Distributed

allthingsdistributed.com

top cloud podcasts

The Cloudcast

thecloudcast.net

Cloud Computing Weekly

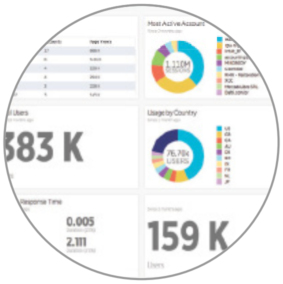
bit.ly/cloudcw

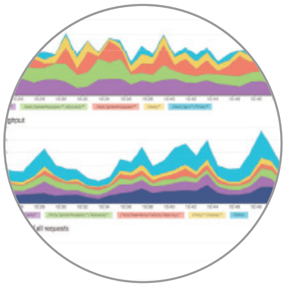
Exponent

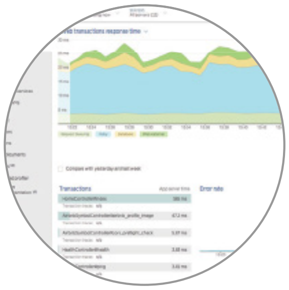
exponent.fm

100% visibility in the cloud

Open a window on your software performance.







Measure app performance

Before, during, and after migration.

Full stack visibility

Across your entire hybrid, public, private, single or multiple cloud environment.

Developer insights

Quickly find production problems in the exact line of code.

Business success

Drive accountability and stay focused on your digital future.

The New Relic Software Analytics Platform gives you all the information you need to manage your cloud-based apps.

New Relic®

Change is Constant in the Age of the Cloud: How Will You Keep Up?

There's no going around it: every consumer is now a digital consumer. And for companies striving to deliver personalized and consistent digital experiences across web and mobile, there's no choice but to embrace the fact that they're now in the software business.

Being in the software business, however, requires entirely new ways of thinking and running the organization. Success requires tight coordination and constant feedback loops between the following core disciplines:

- **Application performance.** With the rise of cloud computing, containerization, microservices, and DevOps, modern software architectures have grown incredibly powerful—and incredibly dynamic. In order to keep up with change, companies need the agility to manage distributed, service-oriented environments and ensure their applications are running optimally.
- **Customer experience.** With the customer's primary interface now being digital, companies need to leverage new technologies to meet constantly evolving needs and engagement tactics. As businesses gain a better understanding of customer behavior, they need a way to make sure the experience they're delivering is a positive one.
- **Business success.** Customer-facing applications are generating a sea of data and critical business insights. Companies need to understand who is buying what to provide appealing promotions and bundling options. They need to proactively analyze users' activity and use those findings to innovate their products further.

All of the data needed to excel in these disciplines is readily available, but it has to be collected, stored, and analyzed in a consistent manner across developers, operations teams, and product owners. Data can't be siloed across disparate tools. Companies need a single, all-encompassing platform to break down complexity and provide a unified, end-to-end view. With a real-time software analytics platform, companies can better manage their cloud-based applications and improve their business.

WRITTEN BY BHARATH GOWDA
DIRECTOR OF MARKET LEADERSHIP,
NEW RELIC

New Relic Product Suite by New Relic

Shift, re-fit, or modernize. No matter your approach, New Relic gives you the data you need to prove your cloud migration is a success, with 100% visibility for your entire stack.

FEATURES

- Reduce failure risks and fix errors fast via real-time app monitoring
- Measure the impact of code changes on server performance
- Prove fixes in both re-architected and new applications
- Prove fixes in both re-architected and new applications

CLOUD INTEGRATIONS

- Amazon Web Services
- Microsoft Azure

CONTINUOUS DELIVERY INTEGRATIONS

- Puppet
- Jenkins
- Chef
- Jira
- PagerDuty
- Ansible
- SaltStack
- Codeship
- BigPanda
- dploy

CUSTOMERS

- Bleacher Report
- Miniclip
- Microsoft
- Nike
- Groupon
- NBC
- Rdio
- Sony

CASE STUDY

Zendesk's goal is to fully democratize the help desk, making it possible for any company small or large to provide superior customer service while also empowering their customers. Ensuring that customers trying out the service have an excellent user experience is critical. Not only must it be easy and intuitive, but it is also equally important that the system is always available and performance is consistently stellar in every component from the front end to the back end. With more than 65 million end users interacting with the system, the service has to be accessible from anywhere, from any device all the time. Zendesk has a published SLA of 99.9% uptime/month, often achieving 100% uptime. Great visibility into performance issues and the ability to quickly find and resolve them is essential. New Relic gave the team an easy-to-use, intuitive tool to help them pinpoint and resolve issues quickly, rapidly becoming the backbone of Zendesk's performance monitoring strategy. Helpdesks are critical to any company's operation. The team understands all too well that any Zendesk performance issue or outage causes their customers to suffer, so preventing issues is their first concern. The second is resolving them quickly if they do occur.

BLOG blog.newrelic.com

TWITTER @newrelic

WEBSITE newrelic.com/insights

Forecasting the Next Year in Cloud

BY BERNARD GOLDEN

QUICK VIEW

01
2015 will see a continuation of the curve of accelerating technology change.

02
IT is moving out of the back office and into the front lines of business today.

03
Enterprise cloud computing adoption will accelerate.

04
There is a need for new cloud-based application architectures.

05
A skill-shortage among developers is creating an industry bottleneck.

Cloud computing has taken the IT world by storm. Less than a decade old, cloud computing is now firmly placed as the de facto infrastructure for all new applications. Given its success, it's tempting to think that its revolutionary impact is complete, and future innovation will come from a different direction. That thought, while understandable, is incorrect. When it comes to the cloud, as the saying goes, "you ain't seen nothin' yet." Here's why.

ACCELERATING TECHNOLOGY CHANGE

The notion that things are changing faster and faster is widely shared — so much so that it can be considered a sort of a cliche. But, just as cliches often capture a general truth, so too does the notion that our society and economy are moving faster than ever before.

The reason is straightforward: we are experiencing "accelerating technology change." The key to that statement is the first word: accelerating. Computer processing is infusing more and more elements of our business and personal lives. As processing form factors shrink via Moore's Law, computing moves into new areas and devices.

Less than a decade ago, the mobile phone went from a talking device to a computing device, with integrated GPS, accelerometer, time, photography and video, and IP communication, unleashing an incredible wave of innovation via the app revolution. This year, we will see the arrival of wearables with the initial launch of smartwatches. Too many people make the common mistake of viewing a new technology platform as an inferior form of the existing platform, so many initial reviews of smartwatches criticize them as being inadequate versions of smartphones. If history tells us anything, clever inventors and entrepreneurs will take the new platform as a launchpad for totally new applications and use cases, so look for interesting wearable innovation over this and coming years.

And here's the thing: it doesn't stop with wearables. The reason technology is accelerating is that each doubling of Moore's Law delivers vastly increased capability—instead of the 240 million to 480 million transistor per chip increase of a decade ago, today it's from to 4 billion to 8 billion—in other words, each of today's doubling delivers 10 times the total amount of processing power available a decade ago. So, we will continue on the curve of accelerating technology change, and this will underlay the torrid pace of IT innovation 2015 will bring.

IT: FROM BACK OFFICE TO FRONT LINE OF THE BUSINESS

Traditionally, IT has automated repetitive business functions and reduced their cost. For example, in the early days of computing, companies used IT to move payroll processing from handwritten paycards to automated systems. This automation process migrated many aspects of business to software systems: invoicing, inventory tracking, document preparation and the like. Consequently, IT became viewed primarily as a cost reduction mechanism, and was measured according to how cheaply it could perform its tasks. In the parlance of organizational theory, companies considered IT as a cost center and part of corporate overhead.

Today that is all changing. IT is no longer just a way to perform back office functions less expensively than having humans do the same job. With the accelerating technology change described above, we increasingly execute business and personal activities via software-based systems. As the phrase goes, bits are replacing atoms.

To offer an example, just a few years ago if you visited a foreign city, you bought a tourist guide and a map. Getting around required figuring out the local transit system and how to buy tickets, often from a not-very-helpful employee. Today, a mobile phone is a magic portal that simplifies all manner of travel challenges. Google Maps helps you discern where to go, and TripAdvisor can tell you everything you need to know about quality restaurants and hotels. Transit systems have apps to offer tickets with information in many languages, and scanners to read electronically-delivered tickets. And if you want to communicate with someone who only speaks the local language, Google Translate facilitates conversation with a real-time translation.

Every one of the services accessed via this magic portal is a software application exposed through the mobile device. Using these applications is so much easier and, from the application provider's perspective, so much more efficient, that online interaction is quickly becoming the default way we live and work. The net effect of this is that IT is moving out of the back office and into the front lines of business today.

INCREASED ADOPTION OF CLOUD COMPUTING

The shift of IT from back office functions to the front line of business carries an important implication in terms of the resources required to operate these new customer-facing applications. Traditional back office applications are quite predictable—stable user populations, highly consistent usage loads, static topologies and infrastructure, and so on.

The new breed of front line applications are nothing like traditional back office systems. They are highly unpredictable in every dimension—so traditional data center environments aren't a good fit to operate them. The shift of how we do business means that companies need new computing environments that are flexible, hugely scalable, rapidly reconfigurable, and highly elastic—able to add or subtract application resources quickly and easily. In a phrase, companies need cloud computing.

The decade-long shift of the nature of IT almost exactly tracks the growth curve of cloud computing. And one can only expect to see cloud usage continue to explode, given the dynamics explored in the first section of this piece. The scale of this adoption is a subject of some controversy. Many industry observers have fairly modest expectations of cloud adoption, seeing cloud computing as an incremental addition to what will remain the preponderance of infrastructure usage, traditional back office applications.

However, this viewpoint is almost certainly wrong. Every platform shift in the history of IT has brought at least an order of magnitude increase in usage, and one should expect to see this repeated, at least, with cloud computing. With the ongoing pattern shift of atoms to bits, the growth of cloud computing could be significantly larger than even the ten times associated with an order of magnitude.

NEW APPLICATION ARCHITECTURES

One critical restraint in the shift to cloud-based applications, with their highly erratic load patterns and service mashup functionality, is the inadequacy of traditional application architectures. The monolithic three-tier application architecture, based on hand-installed software components and hand-configured connectivity, can't meet the needs of modern application operations.

When an application is composed of dozens or hundreds of different kinds of functions (e.g., a restaurant review app that draws in GPS-based restaurant lists, cloud storage-based photos, user-submitted reviews, and provides a reservation transaction capability), the many component parts update at different frequencies, while the end application itself may change frequently as new functionality or bugfixes roll out on their own schedule.

Trapping all of that complex, frequently changing software system in a monolithic, hard-to-build, hard-to-configure, application delivery package is a recipe for dissatisfaction, if not disaster. The new front line application world expects frequent and low-effort change, and traditional application architectures can't deliver it.

Consequently, one can expect to see new application architectures come to the fore. The new microservices architecture, in which many independently operating API-exposing mini-applications are assembled into a larger aggregation comprising a single end user-facing application, is the new application architecture paradigm. This paradigm, pioneered by Netflix and other "webscale" Internet-based offerings, is rapidly migrating into enterprises, which, after all, are increasingly coming to resemble the webscale offerings of online enterprises.

One challenge of these new architectures is that they require new skills on the part of application developers and operators. Moreover, they require new toolsets for both application lifecycle management and operational monitoring and management. Enterprise IT is likely to see a new set of companies added to their approved vendor lists, as legacy providers struggle to update their tools to manage these new application topologies, and new, innovative companies deliver products suitable for the new world of IT.

> IT is no longer just a way to perform back office functions less expensively than having humans do the same job."

The need for new application architectures is a second-order aspect of the cloud computing revolution, and one not widely perceived. Look to this year as one in which the constraint of traditional application architectures becomes more widely comprehended.

THE BATTLE FOR SKILLS

Just as legacy infrastructure and architectures are poorly suited for the new world of applications, so too are legacy IT skills. Not only are many IT personnel inadequately skilled for the new demands of cloud computing, many resist the need to develop new talents at all, decrying cloud computing as just a flash-in-the-pan or as something suited for companies outside of their own.

They're wrong.

The underlying shift of technology requires new skills. Even more important, the changing nature of business means a skill shortage doesn't just imply delayed upgrades to the existing email system—it means being unprepared to do business they way customers want and competitors deliver. And that's something no company can afford.

As companies come to recognize how critical building new applications that are designed and operated for cloud environments is, one can expect a vicious battle for skill acquisition. In essence, developers are now the resource in shortest supply and this shortage represents a bottleneck in enabling companies to do business in the 21st century way.

CONCLUSION

The long-heralded Information Age, which was predicted to supplant the Industrial Age, has now arrived in full force. As technology functionality and adoption continue to accelerate, IT capability will rapidly come to represent a key corporate resource. Those companies that recognize this fact will be well-prepared for this year's dynamic environment, while those that overlook it — or worse, deny it — will find themselves lagging further and further behind the leaders in their industries.

BERNARD GOLDEN Named in Wired.com as one of the ten most influential people in cloud computing, Bernard advises organizations that leverage his expertise to accelerate their success. He is the author/co-author of four books, including *Virtualization for Dummies* and *Amazon Web Services for Dummies*. Bernard is VP of Strategy at ActiveState; he is also the cloud computing advisor for CIO Magazine, and a highly regarded speaker and presenter at conferences throughout the world.

Questions to Ask About Security in the Cloud

BY JOHN WETHERILL TECHNOLOGY EVANGELIST, ACTIVESTATE

ON PREM/PRIVATE CLOUD: IS THE DATACENTER SUPPLY CHAIN SECURE?

- Manufacturing
- Delivery
- Assembly
- Configuration

ON PREM/PRIVATE CLOUD

- Are servers caged and locked?
- Who has the key?
- Who has access?
- Is the datacenter perimeter secured and tracked?

WHAT'S YOUR SHADOW IT POLICY?

- How is it enforced?
- How is it communicated?
- What is at risk in event of violation?

IS THERE A BYOD POLICY?

- How is it enforced?
- How is it communicated?
- Can you automatically detect data leakage?

EDUCATION & COMMUNICATION

- Do you understand OWASP Top 10? [1]
- Do you have your ear to the ground for new vulnerabilities dispatches?
- Are you in contact with your peers to stay on top of their breaches and practices?
- Do you have publicity plans in place to inform users, customers, shareholders, and public in case of breach?
- Is a brand management process in place to mitigate potential damage?

PIPELINE

- Do you have threat detection tools in place?
- Do you test this process under heavy load?
- Are you performing static code analysis to detect OWASP vulnerabilities?
- Are they integrated with your CI/CD system?
- Are you focused on security from day one of the SDLC?
- Are you rigorously practicing change tracking? [2]

SECURITY AUDIT

- Is a full end-to-end security audit process in place?
- Can it be performed in real time?
- Does it target all high-impact and regulatory security threats?

PATCHING

- What's the process to patch security updates?
- Do you vet and test patches before they're applied?

TESTING

- Do you test for vulnerabilities?
- Do you test at scale and heavy load?
- Do you deliberately instigate security failures?

MONITORING & LOGGING

- Are you able to integrate real-time logs with incidence events?

RESILIENCY, RECOVERY, REMEDIATION

- Are you able to instantly tear down, snapshot (for analysis) then reprovision compromised systems?

AWARENESS

- How are you tracking bleeding edge tools, technologies, practices?
- Do you attend security conferences like BlackHat, DEFCon, RAS, CSA?

FIRST RESPONDERS

- Have you developed a breach response process?
- Is a first responder team in place?
- Are they trained?
- Are they adapting to new vulnerabilities as they appear?
- Do they stand by, 100% ready to respond, at all critical times?

STAFFING

- Is there restricted developer access to sensitive data?
- Have you identified trusted circles inside and outside the organization, with precisely defined levels of access?

THREAT DETECTION & ANALYSIS

- Are you able to detect anomalous activity?
- Is a process in place to collect, normalize, correlate incoming security threat data?
- Can you distill/reduce/prioritize events into a shortlist worthy of investigation?
- Can you identify highly relevant events with real business impact?
- Are you tracking malicious activity over time to identify trends?

REGULATIONS

- Are you on top of current security regulations and compliance requirements?
- Do you keep up to speed on evolving requirements?
- Are you tracking mandates for privacy and anonymity?

[1] https://www.owasp.org/index.php/Top_10_2013-Top_10

[2] http://techblog.netflix.com/2014/06/announcing-security-monkey-aws-security.html

Solutions Directory

This directory of cloud platforms, tools, and services provides comprehensive, factual comparisons of data gathered from third-party sources and the tool creators' organizations. Solutions are selected for inclusion in the directory based on several impartial criteria, including solution maturity, technical innovativeness, relevance, and data availability.

For a more structured classification of a subset of these solutions by layer and case, see graphic on pages 16-17.

IaaS

PRODUCT	CLASSIFICATION	PUBLIC/PRIVATE	FREE TRIAL	WEBSITE
Amazon EC2	Virtual Machines	Public	750 hours, Limited by storage	aws.amazon.com
Apache CloudStack	Virtual Machines	Public, Private by provider	Open Source	cloudstack.apache.org
CenturyLink Cloud	Virtual Machines, App Containers	Public, Private by provider, Hybrid	30 days	centurylinkcloud.com
Datapipe	Virtual Machines	Private by provider	None	datapipe.com
Digital Ocean	Virtual Machines	Public	None	digitalocean.com
EMC Cloudscaling Elastic Cloud	Virtual Machines	Public, Private by provider, Hybrid	None	cloudscaling.com
Google Compute Engine	Virtual Machines	Public	None	cloud.google.com
Hosting.com	Virtual Machines	Any	None	hosting.com
HP Helion Public Cloud	Virtual Machines, iPaaS, DBaaS	Public	Limted by storage	hp.com
IBM Softlayer	Virtual Machines	Any	30 days	ibm.com
Internap	Virtual Machines	Public, Private by provider, Hybrid	None	internap.com

IaaS *cntd.*

PRODUCT	CLASSIFICATION	PUBLIC/PRIVATE	FREE TRIAL	WEBSITE
Joyent	Virtual Machines	Public, Private on-premise, Hybrid	Free tier available	joyent.com
Linode	Virtual Machines	Public, Private by provider	None	linode.com
NaviSIte	Virtual Machines	Public, Private on-premise	None	navisite.com
OpenStack	Virtual Machines	Public, Private on-premise	Open Source	openstack.com
Profitbricks	Virtual Machines	Public, Private by provider	14 days	profitbricks.com
Rackspace Open Cloud	Virtual Machines, DBaaS	Any	Free tier available	rackspace.com
Red Hat Enterprise Linux Openstack	Virtual Machines, App Containers, DBaaS	Public, Private on-premise	90 days	redhat.com
Sungard	Virtual Machines	Public, Private by provider	None	sungardas.com
Verizon Terremark	Virtual Machines	Any	None	verizon.com
Virtustream Cloud IaaS	Virtual Machines	Private by provider, Hybrid	None	virtustream.com
Windstream	Virtual Machines	Public, Private on-premise, Hybrid	None	windstream.com

PaaS

PRODUCT	CLASSIFICATION	PUBLIC/PRIVATE	FREE TRIAL	WEBSITE
ActiveState Stackato	Virtual Machines, App Containers, aPaaS, DBaaS	Private by provider, Private on-premise, Hybrid	Limited by storage	stackato.com
Apprenda	App Containers, aPaaS, DBaaS, MBaaS	Private on-premise, Hybrid	Limited by storage	apprenda.com
AWS Elastic Beanstalk	Application Containers, aPaaS	Public	Limited by storage	aws.amazon.com
CenturyLink AppFog	App Containers, aPaaS, DBaaS	Public	7 days	centurylinkcloud.com
Clever Cloud	App Containers, aPaaS, DBaaS	Any	No free trial	clever-cloud.com
cloudControl	App Containers, aPaaS, DBaaS	Public, Private on-premise	14 days	cloudcontrol.com
Engine Yard	App Containers, aPaaS	Public	500 hours	engineyard.com
FatFractal	Virtual Machines, App Containers, aPaaS, DBaaS, MBaaS	Public, Private on-premise, Hybrid	30 days	fatfractal.com
Google App Engine	App Containers, aPaaS	Public	None	cloud.google.com
Heroku	App Containers, aPaaS, DBaaS	Public	Limited by storage and 8 instances	heroku.com

PaaS *cntd.*

PRODUCT	CLASSIFICATION	PUBLIC/PRIVATE	FREE TRIAL	WEBSITE
Jelastic	Virtual Machines, App Containers, aPaaS, iPaaS	Any	Two weeks with hosting partner	jelastic.com
Lunacloud	Virtual Machines, App Containers, iPaaS, DBaaS, Model-Driven PaaS	Public	None	lunacloud.com
Mendix App Platform	aPaaS, DBaaS, Model-Driven PaaS	Any	Free tier available	mendix.com
Microsoft Azure	Virtual Machines, App Containers, aPaaS	Public	30 days	azure.microsoft.com
Outsystems Platform	aPaaS, iPaaS, DBaaS, MBaaS, Model-Driven PaaS	Any	30 days	outsystems.com
Pivotal CF	App Containers, aPaaS, DBaaS	Public, Private on-premise	90 days	pivotal.com
Red Hat JBoss Enterprise Application Platform for xPaaS	App Containers, aPaaS	Any	Open Source	openshift.com
Salesforce1	App Containers, aPaaS, iPaaS, DBaaS, Model-Driven PaaS	Public	Limited by storage	salesforce.com
SAP HANA	Virtual Machines, App Containers, aPaaS, iPaaS, DBaaS, MBaaS	Public	30 days	sap.com
Progress Rollbase	aPaaS, Model-Driven PaaS	Any	30 days	progress.com
WorkXpress PaaS	aPaaS, DBaaS, MBaaS	Public, Private on-premise, Private by-provider	30 days	workxpress.com
WSO2 App Cloud	Virtual Machines, App Containers, aPaaS, DBaaS, MBaaS	Public, Private on-premise, Hybrid	Free tier available	wso2.com

MBaaS

PRODUCT	CLASSIFICATION	PUBLIC/PRIVATE	FREE TRIAL	WEBSITE
Anypresence	aPaaS, mBaaS	Public, Private by provider, Private on-premise	Pilot program available	anypresence.com
Baasbox	mBaaS	Public, Private on-premise	30 days	baasbox.com
Backendless	Virtual Machines, App Containers, aPaaS, DBaaS, MBaaS	Any	Free tier available	backendless.com
Facebook Parse	aPaaS, mBaaS	Public	Free tier available	parse.com
Kii	mBaaS	Any	Free tier available	kii.com
Kinvey	mBaaS	Private by-provider, Private on-premise	Limited by storage	kinvey.com
Kony MobileFabric	aPaaS, iPaaS, mBaaS	Public, Private on-premise, Hybrid	90 days	kony.com
Kumulos	mBaaS	Private by-provider	Free until app is deployed	kumulos.com

diving deeper

INTO FEATURED CLOUD PRODUCTS

Looking for more information on individual cloud solutions providers? Eight of our partners have shared additional details about their offerings, and we've summarized this data below.

If you'd like to share data about these or other related solutions, please email us at research@dzone.com.

Bluemix by IBM

LANGUAGES

JAVA RUBY PYTHON PERL PHP NODE.JS GROOVY

INFRASTRUCTURE USED

IBM Infrastructure

SECURITY OPTIONS

- Keypair-based authentication
- 2-factor authentication
- SSH connections
- SSL connections
- SAML
- Oauth

Red Hat JBoss Enterprise Application Platform for xPaaS by Red Hat

LANGUAGES

JAVA

INFRASTRUCTURE USED

Customer's choice

SECURITY OPTIONS

- Keypair-based authentication
- Configurable role-based permissions
- SSL connections
- SAML

Jelastic by Jelastic

LANGUAGES

JAVA C# RUBY PYTHON PHP NODE.JS

INFRASTRUCTURE USED

Customer's choice

SECURITY OPTIONS

- Keypair-based authentication
- Configurable role-based permissions
- SSH connections
- SSL connections
- File-level encryption

OutSystems Platform by OutSystems

LANGUAGES

JAVA C# C/C++

INFRASTRUCTURE USED

Amazon EC2

SECURITY OPTIONS

- Keypair-based authentication
- 2-factor authentication
- Configurable role-based permissions
- SSH connections
- SSL connections
- File-level encryption
- SAML
- Oauth

Rollbase by Progress Software

LANGUAGES

JAVASCRIPT NODE.JS

INFRASTRUCTURE USED

Customer's choice

SECURITY OPTIONS

- Configurable role-based permissions
- SSL connections
- OAuth

Stackato by ActiveState

LANGUAGES

JAVA RUBY PYTHON PERL PHP NODE.JS GO GROOVY

INFRASTRUCTURE USED

Customer's choice

SECURITY OPTIONS

- Keypair-based authentication
- Configurable role-based permissions
- SSH connections
- SSL connections
- SAML
- Oauth

Pivotal CF by Pivotal

LANGUAGES

JAVA RUBY NODE.JS GOLANG JAVASCRIPT

INFRASTRUCTURE USED

Amazon EC2,
OpenStack, or vSphere

SECURITY OPTIONS

- SSL
- SAML
- OAuth

WSO2 App Cloud by WSO2

LANGUAGES

JAVA PHP NODE.JS

INFRASTRUCTURE USED

Amazon EC2 or
Customer's choice

SECURITY OPTIONS

- SSL Connections
- Oauth

glossary

APPLICATION CONTAINER An isolated package of application components and dependencies that is infrastructure-agnostic.

APPLICATION PLATFORM-AS-A-SERVICE (APAAS) A cloud service that offers development and deployment environments for application services.

BLOCK STORAGE A form of data storage virtualization that decouples storage volumes from their hardware, allowing more advanced user management.

CLOUD BROKER An entity that serves as a connection between cloud customers and cloud service providers.

CLOUD BURSTING A technique where a hybrid cloud provides extra resources on an as-needed basis to private clouds.

CLOUD CONTENT DELIVERY NETWORK (CDN) A CDN on the cloud. It consists of geographically distributed server networks that improve web content delivery performance.

CLOUD COMPUTING A model where infrastructure, applications, and business processes can be delivered as an on-demand service over a network.

CLOUD MANAGEMENT PLATFORM (CMP) An integrated suite of cloud management products that provide self-service interfaces, system images, metering and billing systems, and workload optimization features.

CLOUD SERVERS Virtualized servers with additional capabilities and a web interface or API for self-service provisioning.

CLOUD STORAGE A data storage service where customers transfer their data over the internet or another network to an offsite storage system maintained by a third party.

CONTAINERIZATION A server virtualization method where the kernel of an operating system allows for multiple isolated user space instances, instead of just one.

DATABASE-AS-A-SERVICE (DBAAS) Delivers the entire database management system (DBMS) as a cloud service to the customer.

DEVOPS An IT organizational methodology where all teams in the organization, especially development teams and operations teams, collaborate and implement technology to increase software production agility and achieve business goals.

FEDERATION When a cloud provider or cloud broker merges data across multiple cloud networks.

HIGH AVAILABILITY Refers to systems that are resistant to failure and engineered to operate continuously without failure for long periods of time.

HYBRID CLOUD An environment that combines public and private cloud infrastructure.

INFRASTRUCTURE-AS-A-SERVICE (IAAS) A service model that delivers self-service computing, networking and storage resources on-demand over a network.

INTEGRATION PLATFORM-AS-A-SERVICE (IPAAS) A broker utility component in a cloud platform that handles communication between applications.

LOCK-IN When a client has difficulty moving from one cloud vendor to another due to non-standardized APIs, data structures, and service models.

MOBILE BACKEND-AS-A-SERVICE (MBAAS) Connects mobile applications to cloud databases while providing user management, push notifications, and social integrations.

MODEL-DRIVEN PAAS A PaaS architecture that adds support for simplified programming languages that can be used by less-technical business engineers.

MULTI-TENANCY A system attribute where a single instance of an application serves multiple client organizations (tenants).

NETWORKING-AS-A-SERVICE Provides computing and connectivity resources that allow network connections and inter-cloud connections.

OBJECT STORAGE A cloud-based storage service containing data components that are ready to be retrieved and manipulated on-demand in an application.

PLATFORM-AS-A-SERVICE (PAAS) A cloud service model that provides the hardware, operating systems, storage, or network capacity needed to develop and run applications.

PRIVATE CLOUD A cloud data center that can be hosted on-premise, behind the company's firewall, or by a third party in a reserved, secure space (e.g. virtual private cloud).

PUBLIC CLOUD Cloud services that are offered over the Internet and available to the general public.

SERVICE LEVEL AGREEMENT (SLA) A contract that specifies the consumer's IT requirements and the provider's commitment to them.

SOFTWARE-AS-A-SERVICE (SAAS) An application provided over a network by the vendor with no installation required.

SOFTWARE-DEFINED NETWORKING (SDN) Software Defined Networking is an approach to networking (enabled by a commercial or in-house hardware solution) that decouples switching and other network handling processes from the hardware and instead allows these processes to be controlled completely by a software application called a centralized controller. This provides more control over network traffic flow and allows the organization to buy less expensive network switches.

VIRTUAL MACHINE A software emulation of a physical computing resource that can be modified independent of the hardware attributes.

VIRTUAL PRIVATE CLOUD (VPC) A private cloud that is hosted from a third party's data center.offered to only authorized users, not the general public.

DZone Research Guides

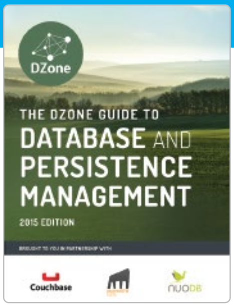
> Better, more concrete, and closer to reality than reports from Gartner and Forrester...

ROBERT ZAKRZEWSKI,
DZONE MEMBER

DZONE GUIDE TO Continuous Delivery

Understand the role of DevOps, automation, testing, and other best practices that allow Continuous Delivery adoption.

DOWNLOAD



DZONE GUIDE TO Databases

Explore effective database performance strategies and resolve common database-related obstacles.

DOWNLOAD

DZONE GUIDE TO Developer Programs

Get a full analysis of developer program trends and learn how you can benefit from joining the right one.

DOWNLOAD

UPCOMING IN 2015

Performance & Monitoring
Java Ecosystem

Software Quality
Internet of Things

Mobile Development
Big Data

Get them all for free on DZone at dzone.com/research