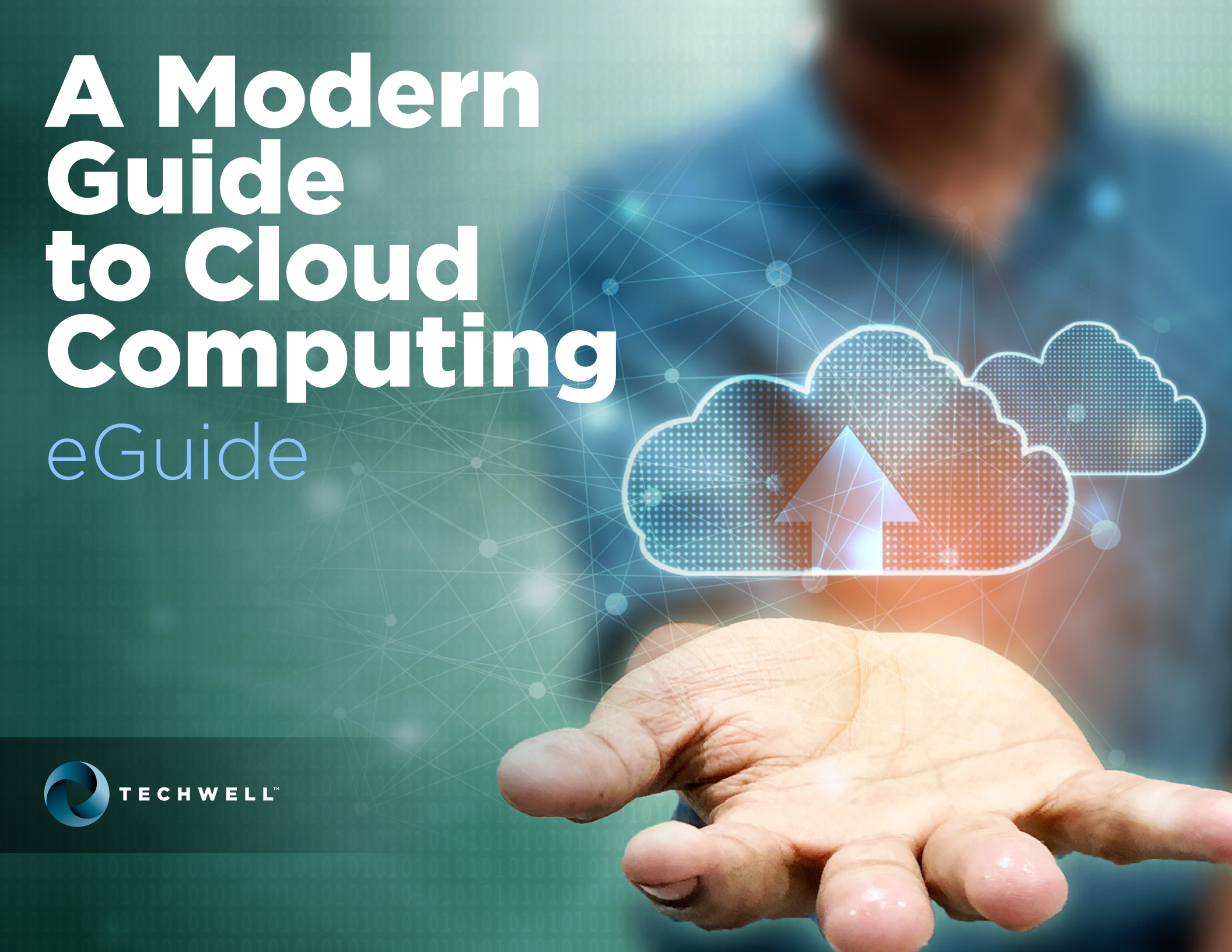
A Modern Guide to Cloud Computing

eGuide



TECHWELL™

3
The Pros and Cons of a Serverless DevOps Solution

5
Migrating to the Cloud: Which Model Is Right for You?

6
6 Major Challenges of Cloud Computing

8
The Advantages of Serverless Cloud Providers

10
Ensure That Your Current Cloud Solution Will Stand the Test of Time

11
Lessons Learned from an Enterprise Government Cutover to the Cloud

14
Quality Checks to Address Before, During, and After Cloud Migration

15
Serverless Technology and Integration with DevOps: An Interview with Glenn Buckholz

17
Additional Resources

Organizations embarking on a digital transformation need to decide how they want to handle their data and what infrastructure makes sense for their goals. Cloud computing is a popular choice, but it has changed a lot over the years, and there are more options than ever before. This guide to modern cloud computing details its advantages and challenges; how to orchestrate your cloud migration; public, private, and hybrid cloud options; and how to choose the right cloud services for your business.

In this Cloud Computing eGuide

The Pros and Cons of a Serverless DevOps Solution

The dream of any product owner is fully customizable production software without the expense of the hardware it rests upon. While not completely free of infrastructure, serverless infrastructure significantly reduces overhead costs by abstracting away physical hosting, physical security, server maintenance, and OS patching. Here's what you need to know to decide if serverless infrastructure is right for you.

Migrating to the Cloud: Which Model Is Right for You?

Cloud computing is a relatively recent trend, and several organizations have opted to migrate their services and data to the cloud. Which of the cloud computing models available is right for which situation? Let's look at the three options—public, private, and hybrid—and discuss when it's a good idea to use each one.

6 Major Challenges of Cloud Computing

Companies of all sizes depend on cloud computing to store important data. However, significant factors such as cost, reliability, and security must not be overlooked. Here are six common challenges you should consider—and develop plans to mitigate—before implementing cloud computing technology.

The Advantages of Serverless Cloud Providers

Most cloud providers have server-based computing services. But that requires servers to be provisioned and administered, and servers have a fixed capacity to operate within. A new DevOps trend is to go serverless—however, this doesn't mean no servers are used at all. Learn more about this model and its advantages now.

Ensure That Your Current Cloud Solution Will Stand the Test of Time

It's still early in the lifecycle of cloud adoption. This means certain cloud vendors and technologies will fall by the wayside as adoption takes on critical mass. How, then, do you future-proof your cloud solution to make sure you don't make a decision that you'll regret later? Here are three ideas to consider.

Lessons Learned from an Enterprise Government Cutover to the Cloud

As more and more organizations move from datacenters to the cloud, there are some traps that could plague your cloud migrations—and those traps don't differ for cloud migrations inside the U.S. government. Cloud cutover challenges range from configuration management downfalls, to communications failures, to delayed security involvement. Here's what you need to know about migrating to the cloud on a U.S. government project.

Quality Checks to Address Before, During, and After Cloud Migration

While it is a no-brainer that most organizations have either migrated to or are considering a move to the cloud, the stakes for cloud providers and consumers are quite high. Mukesh Sharma details some quality checks to address before, during, and after cloud migration to ensure a smooth transition.

Serverless Technology and Integration with DevOps: An Interview with Glenn Buckholz

Glenn Buckholz, a technical manager at Coveros Inc., discusses methods to gain an understanding of serverless technology, the motivation behind businesses moving to serverless technology, and how a serverless infrastructure changes your testing strategy and bug reports.

Additional Resources

3 The Pros and Cons of a Serverless DevOps Solution
5 Migrating to the Cloud: Which Model Is Right for You?
6 6 Major Challenges of Cloud Computing
8 The Advantages of Serverless Cloud Providers
10 Ensure That Your Current Cloud Solution Will Stand the Test of Time
11 Lessons Learned from an Enterprise Government Cutover to the Cloud
14 Quality Checks to Address Before, During, and After Cloud Migration
15 Serverless Technology and Integration with DevOps: An Interview with Glenn Buckholz
17 Additional Resources

The Pros and Cons of a Serverless DevOps Solution

By Glenn Buckholz

The dream of any product owner is fully customizable production software without the expense of paying for the hardware it rests upon. While the cloud and infrastructure as a service (IaaS) partially deliver on this promise, a completely serverless infrastructure would be much closer to this dream. From a product owner's perspective, the possibility of deploying a public-facing application without investing precious dollars and time on infrastructure is tantalizing.

While not completely free of infrastructure, serverless infrastructure significantly reduces overhead costs by abstracting away physical hosting, physical security, server maintenance, and OS patching. Additionally, the flexibility of a serverless implementation is much greater than physical infrastructure, or even IaaS, as you pay only for what you use—at hundred-millisecond increments, on some platforms.

So, what exactly is serverless infrastructure? The answer varies slightly from provider to provider, but here is a short explanation. "Serverless" does not mean the absence of servers; it refers to the fact that the application owner does not need to know much about the underlying servers their software is running on. The serverless infrastructure provider abstracts away all the details about the infrastructure so your team can focus its efforts solely on delivery of new functionality instead of server maintenance.

For instance, a typical static web server is replaced by a blob hosting service, a database instance is replaced by a hosting providers database service, user management is handled by the provider's single sign-on (SSO) and login solution, etc. All your DevOps team has to do is upload your application and content.

You do not need to worry about server patches, upgrading environments, reboots, adding additional resources to handle more load, spinning up additional needed services, or dealing with hardware failures.

Serverless infrastructure also gives us other technical and business advantages. First, you don't need to hire twenty-four-hour on-call

3
The Pros and Cons of a Serverless DevOps Solution

5
Migrating to the Cloud: Which Model Is Right for You?

6
6 Major Challenges of Cloud Computing

8
The Advantages of Serverless Cloud Providers

10
Ensure That Your Current Cloud Solution Will Stand the Test of Time

11
Lessons Learned from an Enterprise Government Cutover to the Cloud

14
Quality Checks to Address Before, During, and After Cloud Migration

15
Serverless Technology and Integration with DevOps: An Interview with Glenn Buckholz

17
Additional Resources

operations support just to keep the infrastructure running. This shrinks your technology footprint, meaning you can focus more resources on delivery and your DevOps team.

Serverless infrastructure does not eliminate the need for security experts, but it does allow you to trust your cloud provider for a large portion of your security needs. The cloud provider filling this role lets you hire only what you need to secure your application code and configure the cloud services properly.

Similarly, serverless infrastructure will not eliminate the need for cloud engineers, but it reduces the number you need so that they are only managing the abstract cloud services in relation to your application, not a myriad of infrastructure responsibilities that you used to need, such as auditing, monitoring, and patch compliance.

Overall, the main advantage is that you can hire fewer specialists not related to your application domain and let your provider handle most aspects of your infrastructure, allowing your team to focus on tasks directly related to feature delivery.

What does not go away with a serverless infrastructure is the need for proper automated testing, continuous integration (CI), continuous delivery (CD), and a DevOps mindset. Giving a developer unfettered access to production code almost never ends well. While they may fix whatever immediate issue is at hand, that easy-access cloud editor that can change production code will also allow defects to creep in. But there are tools that can help eradicate defects during delivery by gating code with automated static analysis, unit tests, and even functional UI tests.

> *Serverless infrastructure does not eliminate the need for security experts, but it does allow you to trust your cloud provider for a large portion of your security needs.*

Additionally, by committing to a CI/CD paradigm, the code is going to reside in an appropriate source code management tool, creating an audit trail of who changed what and when. By properly integrating and testing your application during the delivery process, you can decrease the number of production issues and allow your software developers to focus on new functionality.

However, there are also some disadvantages to using serverless technology. You are ceding two very important things to your cloud provider: choice and control. While you no longer need to worry about the underlying details of your application servers, you are very limited in how you access your application and server setting.

You are only going to be able to use technologies that the cloud provider you have selected offers, which can lead to vendor lock-in. Currently, serverless offerings are not compatible between providers, so if you are not satisfied with a particular serverless infrastructure provider, you are out of luck because nothing you have will work as is on another provider's environment.

Serverless infrastructure is a way to absolve a project from the responsibility of managing hardware while simplifying infrastructure management, but it does not give product owners a free pass to ignore development and delivery best practices. Strong CI/CD automation and a DevOps mindset can help amplify the value of serverless infrastructure without significantly slowing down new feature implementation. After all, the technology is "serverless," not "peopleless."

3
The Pros and Cons of a Serverless DevOps Solution

5
Migrating to the Cloud: Which Model Is Right for You?

6
6 Major Challenges of Cloud Computing

8
The Advantages of Serverless Cloud Providers

10
Ensure That Your Current Cloud Solution Will Stand the Test of Time

11
Lessons Learned from an Enterprise Government Cutover to the Cloud

14
Quality Checks to Address Before, During, and After Cloud Migration

15
Serverless Technology and Integration with DevOps: An Interview with Glenn Buckholz

17
Additional Resources

Migrating to the Cloud: Which Model Is Right for You?

By Deepak Vohra

Cloud computing is a relatively recent trend, and several organizations have opted to migrate their databases and services to the cloud. There are different cloud computing models available—which is right for which situation? Let's look at the three options.

Public Cloud

Public cloud is the most common cloud computing model and is composed of shared infrastructure and resources hosted in a public domain. Typically, public cloud service providers host the infrastructure at multiple data centers and make their services available over the internet.

Consequently, public cloud does not take network security into consideration. Choose the public cloud if your applications and services are to be accessible to anyone with access to the internet.

Private Cloud

A private cloud hosts the infrastructure and services on a private network that is accessible only to those who have been explicitly granted access. The architecture of a private cloud is no different than that of a public cloud, except that it is hosted over a secure network, such as a data center for a single organization. This increased security may involve additional configuration.

Most cloud service providers provide an option of a virtual private cloud (VPC), or an isolated section of the virtual network in which a user has complete administrative control over the networking environment, including the choice of public IP addresses, routing tables, and subnets. A user may customize the network to provision some of the resources in public subnets and others in private subnets. Access to services may be regulated with network access control lists.

Hybrid Cloud

A hybrid cloud is a mixture of public cloud and private cloud models, where an organization-based data center is combined with public cloud hosted services. The hardware and network at the local data comprise the private cloud, and the hardware hosted on a cloud network comprises the public cloud.

Some public cloud service providers provide services and hosts dedicated to a single user account, isolated from servers and services for other users at the hardware level. In effect, a dedicated service or server is equivalent to a service or server hosted at a private data center, except that it is hosted by a public cloud service provider.

You could want to use a hybrid cloud if you have data that needs to be stored in a secure environment, such as user credentials that should be stored in a private cloud but services and applications that need to be available publicly.

An organization also may extend its resources and infrastructure capacity by using a public cloud. Extending to the public cloud infrastructure could be integrated into the design model of an organization as a permanent feature, or it could be for temporary scaling of resources when the load on a private cloud exceeds capacity.

If you want to migrate to the cloud, use a private cloud if isolation of the network is a priority. Use a public cloud for publicly hosted services. And use a hybrid cloud if some of the services and data need to be secure and isolated, while other services and data need to be accessible to the public.

3
The Pros and Cons of a Serverless DevOps Solution

5
Migrating to the Cloud: Which Model Is Right for You?

6
6 Major Challenges of Cloud Computing

8
The Advantages of Serverless Cloud Providers

10
Ensure That Your Current Cloud Solution Will Stand the Test of Time

11
Lessons Learned from an Enterprise Government Cutover to the Cloud

14
Quality Checks to Address Before, During, and After Cloud Migration

15
Serverless Technology and Integration with DevOps: An Interview with Glenn Buckholz

17
Additional Resources

6 Major Challenges of Cloud Computing

By Ray Parker

Cloud computing is used for enabling global access to mutual pools of resources such as services, apps, data, servers, and computer networks. It is done on either a third-party server located in a data center or a privately owned cloud. This makes data-accessing contrivances more reliable and efficient, with nominal administration effort.

Because cloud technology depends on the allocation of resources to attain consistency and economy of scale, similar to a utility, it is also fairly cost-effective, making it the choice for many small businesses and firms.

But there are also many challenges involved in cloud computing, and if you're not prepared to deal with them, you won't realize the benefits. Here are six common challenges you must consider before implementing cloud computing technology.

1. Cost

Cloud computing itself is affordable, but tuning the platform according to the company's needs can be expensive. Furthermore, the expense of transferring the data to public clouds can prove to be a problem for short-lived and small-scale projects.

3
The Pros and Cons of a Serverless DevOps Solution

5
Migrating to the Cloud: Which Model Is Right for You?

6
6 Major Challenges of Cloud Computing

8
The Advantages of Serverless Cloud Providers

10
Ensure That Your Current Cloud Solution Will Stand the Test of Time

11
Lessons Learned from an Enterprise Government Cutover to the Cloud

14
Quality Checks to Address Before, During, and After Cloud Migration

15
Serverless Technology and Integration with DevOps: An Interview with Glenn Buckholz

17
Additional Resources

Companies can save some money on system maintenance, management, and acquisitions. But they also have to invest in additional bandwidth, and the absence of routine control in an infinitely scalable computing platform can increase costs.

2. Service Provider Reliability

The capacity and capability of a technical service provider are as important as price. The service provider must be available when you need them. The main concern should be the service provider's sustainability and reputation. Make sure you comprehend the techniques via which a provider observes its services and defends dependability claims.

3. Downtime

Downtime is a significant shortcoming of cloud technology. No seller can promise a platform that is free of possible downtime. Cloud technology makes small companies reliant on their connectivity, so companies with an untrustworthy internet connection probably want to think twice before adopting cloud computing.

4. Password Security

Industrious password supervision plays a vital role in cloud security. However, the more people you have accessing your cloud account, the less secure it is. Anybody aware of your passwords will be able to access the information you store there.

Businesses should employ multi-factor authentication and make sure that passwords are protected and altered regularly, particularly when staff members leave. Access rights related to passwords and usernames should only be allocated to those who require them.

5. Data privacy

Sensitive and personal information that is kept in the cloud should be defined as being for internal use only, not to be shared with third parties. Businesses must have a plan to securely and efficiently manage the data they gather.

Cloud computing is a good solution for many businesses, but it's important to know what you're getting into. Having plans to address these six prominent challenges first will help ensure a successful experience.

6. Vendor lock-in

Entering a cloud computing agreement is easier than leaving it. "Vendor lock-in" happens when altering providers is either excessively expensive or just not possible. It could be that the service is nonstandard or that there is no viable vendor substitute.

It comes down to buyer carefulness. Guarantee the services you involve are typical and transportable to other providers, and above all, understand the requirements.

Cloud computing is a good solution for many businesses, but it's important to know what you're getting into. Having plans to address these six prominent challenges first will help ensure a successful experience.

3
The Pros and Cons of a Serverless DevOps Solution

5
Migrating to the Cloud: Which Model Is Right for You?

6
6 Major Challenges of Cloud Computing

8
The Advantages of Serverless Cloud Providers

10
Ensure That Your Current Cloud Solution Will Stand the Test of Time

11
Lessons Learned from an Enterprise Government Cutover to the Cloud

14
Quality Checks to Address Before, During, and After Cloud Migration

15
Serverless Technology and Integration with DevOps: An Interview with Glenn Buckholz

17
Additional Resources

The Advantages of Serverless Cloud Providers

By Deepak Vohra

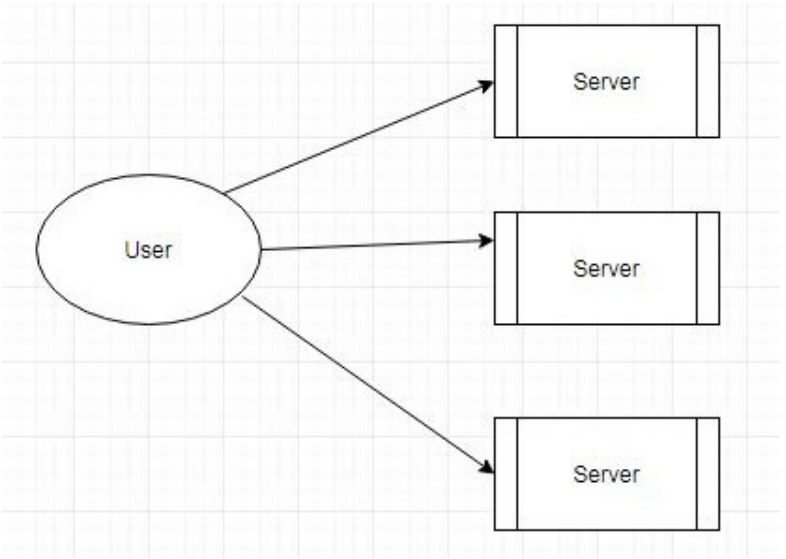
Most cloud providers have server-based computing services. But that requires servers to be provisioned and administered. Servers have a fixed capacity, and computation has to be performed within that capacity, which could lead to unused resources or insufficient resources.

A new DevOps trend is to go serverless. This doesn't mean no servers are used at all; it just means that a programming model runs and tests code, without the platform using servers directly.

This serverless model has several benefits:

- Not having to provision servers
- Not having to connect to servers with a client such as SSH or a remote desktop
- Continuous scaling of computational capacity
- No server administration
- Code runs when required, so there are no unused compute resources
- Extra computation capacity is available if needed

With typical server-based cloud computing, a user provisions the servers directly, connects to a server, and runs software, as illustrated in figure 1.



3
The Pros and Cons of a Serverless DevOps Solution

5
Migrating to the Cloud: Which Model Is Right for You?

6
6 Major Challenges of Cloud Computing

8
The Advantages of Serverless Cloud Providers

10
Ensure That Your Current Cloud Solution Will Stand the Test of Time

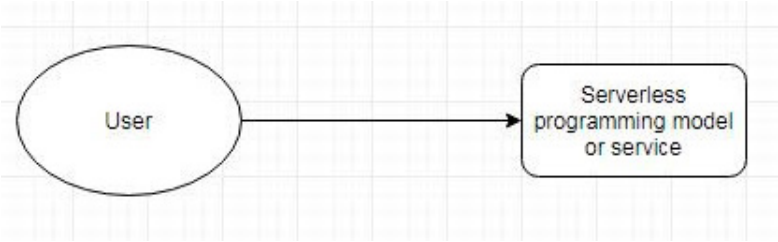
11
Lessons Learned from an Enterprise Government Cutover to the Cloud

14
Quality Checks to Address Before, During, and After Cloud Migration

15
Serverless Technology and Integration with DevOps: An Interview with Glenn Buckholz

17
Additional Resources

With a serverless model, a user connects to a programming model or service to perform computation without ever having to provision, connect to, or manage a server, as illustrated in figure 2.



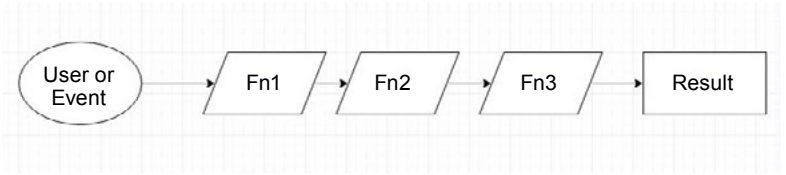
Function as a service, or FaaS, is the most common serverless architecture. FaaS models are most commonly cloud services on which users test and run application code without having to provision the infrastructure needed to run the application. An example could be a data processing application that makes use of a FaaS model, either for a short while or as a long-running application.

This could be a typical use of a FaaS model:

1. User creates and deploys functions, including function code, to a platform
2. User updates or modifies function configuration as needed
3. User invokes the function and gets a result

A function could be invoked directly via a command or indirectly as a response to an event. A developer still needs to develop the functional logic (also called "actions") to be invoked.

A single function may be insufficient to perform all the necessary computation. Multiple functions could be linked to channel the output of one function as the input to another, to create a function composition. Figure 3 illustrates a composition in which three functions are used to generate the result.



The serverless model of computing makes more efficient use of resources while minimizing or totally removing user input for provisioning and managing the infrastructure needed to run the computation. Consequently, this model is becoming popular in the streamlined new world of DevOps.

> *With a serverless model, a user connects to a programming model or service to perform computation without ever having to provision, connect to, or manage a server.*

3
The Pros and Cons of a Serverless DevOps Solution

5
Migrating to the Cloud: Which Model Is Right for You?

6
6 Major Challenges of Cloud Computing

8
The Advantages of Serverless Cloud Providers

10
Ensure That Your Current Cloud Solution Will Stand the Test of Time

11
Lessons Learned from an Enterprise Government Cutover to the Cloud

14
Quality Checks to Address Before, During, and After Cloud Migration

15
Serverless Technology and Integration with DevOps: An Interview with Glenn Buckholz

17
Additional Resources

Ensure That Your Current Cloud Solution Will Stand the Test of Time

By Pete Johnson

It might not feel like it, but it's still early in the lifecycle of cloud adoption. This means certain cloud vendors and technologies will fall by the wayside as adoption takes on critical mass. How, then, do you future-proof your cloud solution to make sure you don't make a decision that you'll regret later?

Here are three ideas to consider.

Careful Use of Higher-Order Services

When public cloud services began, their offerings were limited to the three basics: compute, network, and storage. As time wore on, they all began to create higher-order services like load balancing, databases, and message queues.

Ultimately, those are all about selling more compute, network, and storage under the hood of those interfaces. However, they make it easier for the application developer to narrow the focus of original engineering to the business problem at hand instead of managing a load balancer, database, or message queue. The catch-22 of application architectures in the cloud era is that using higher-order services accelerates development substantially but increases the possibility of lock-in.

Carefully relying on services that have standard interfaces—like SQL, whose query language is the same across implementations—makes their use more portable and reduces the lock-in concern. There will be times where the time-to-market gains will be too good to pass up, but when using these higher-order services, have a remediation plan for getting out if you need to.

Loosely Coupled Data Chunks

A recent IDC study found that 73 percent of surveyed cloud adopters have a hybrid strategy, meaning different applications running on different clouds based on what fits best where. Not only is it a great future-proofing strategy to have diversified investments across multiple cloud vendors, but it implies that data should no longer be thought of as a behemoth that sits in a single place.

Instead, data is more loosely coupled into chunks that can be spread around to multiple clouds, allowing the applications that need it to deduce access latency. A nice side effect of this is that it reduces the data gravity issue. If you need to move data from one cloud to another, the chunks are smaller and more easily migrated.

Cloud Management Platforms

Cloud management platforms (CMPs) are a new class of product that can take a lot of the management overhead away from a hybrid cloud strategy. This makes it easier to deploy applications across multiple clouds by providing a single pane of glass from which to view them all. Such tools typically also provide application migration assistance, benchmarking, governance, and other features that make it easier to diversify cloud investment.

As cloud adoption reaches the masses, most companies are utilizing a hybrid cloud strategy that spreads their bets across multiple vendors so that their investment is future-proofed. Careful use of higher-order services from those cloud vendors, spreading data out to multiple places and in smaller pieces, and relying on the latest tooling in the form of CMPs can go a long way toward protecting that investment.

3
The Pros and Cons of a Serverless DevOps Solution

5
Migrating to the Cloud: Which Model Is Right for You?

6
6 Major Challenges of Cloud Computing

8
The Advantages of Serverless Cloud Providers

10
Ensure That Your Current Cloud Solution Will Stand the Test of Time

11
Lessons Learned from an Enterprise Government Cutover to the Cloud

14
Quality Checks to Address Before, During, and After Cloud Migration

15
Serverless Technology and Integration with DevOps: An Interview with Glenn Buckholz

17
Additional Resources

Lessons Learned from an Enterprise Government Cutover to the Cloud

By Jason Moskowitz

On a recent government project, the DevOps team I was leading was tasked with spearheading the migration of our applications from a datacenter to the cloud. It did not go well. It almost fell into many traps that I imagine plague many cloud migrations, ranging from configuration management downfalls, to communications failures and delayed security involvement. In the blog below, I will go over some of the troubles that plagued our migration.

Background and Configuration Management

Our team was originally brought on to create a CICD pipeline for the currently on-premise system. We were also tasked with helping a separate Cloud team with setting up a similar CICD pipeline in their new cloud pipeline. We chose to create all of our infrastructure and configuration management with Chef, so that when the cloud was ready for it, they could just run our recipes and could easily recreate the same pipeline we had already been using. In reality, the cloud team could never get chef approved for use in the cloud, so everything we built, they rebuilt with Powershell scripts, and lengthy build documents.

HINDSIGHT RESOLUTION

When migrating to the cloud, or really migrating your system anywhere, I would ALWAYS recommend to be using a configuration management tool like Chef, Puppet, or Ansible to maintain consistency in your pipeline. In our case, if the cloud team had put more effort into getting Chef approved, they could have saved months of their time that they spent recreating the work that we had already done in Chef. Short of that, having a better understanding of our limitations in tooling would have been helpful to reduce rework.

3
The Pros and Cons of a Serverless DevOps Solution

5
Migrating to the Cloud: Which Model Is Right for You?

6
6 Major Challenges of Cloud Computing

8
The Advantages of Serverless Cloud Providers

10
Ensure That Your Current Cloud Solution Will Stand the Test of Time

11
Lessons Learned from an Enterprise Government Cutover to the Cloud

14
Quality Checks to Address Before, During, and After Cloud Migration

15
Serverless Technology and Integration with DevOps: An Interview with Glenn Buckholz

17
Additional Resources

Communication Failures

Once the pipeline and infrastructure in the cloud had been created, it was time to demo it to high level agency management. The cloud team created a very good demo. It showed an application go through a pipeline, in a simple way, explained many of the steps in ways that many of the less than technical management could follow, and showed how deployments could work. Unfortunately, this demo was mostly smoke and mirrors; the cloud team built their own, very basic example application, and created their own pipeline process to push the application through. So, basically, they proved that a made up app can be pushed through a pipeline that didn't matter. This wasn't the way our AppDev Teams were going to use the pipeline, and many of the plugins and tools that were going to be needed were still not installed.

> *When designing anything, security needs to be involved from day one. It is always better to think about security at architecture time, rather than trying to squeeze it on top.*

The government loved the demo. Even though they were aware it was only a demo application, I believe that they were left with the impression that the pipeline was more ready than it actually was: that once the cloud received an ATO (Authority to operate), all of the applications could easily be migrated over. This was far from the truth, not only because of the reasons stated above, but also because the security compliance team hadn't been involved yet.

HINDSIGHT RESOLUTION

There are two real problems here, a failure to communicate with the AppDev teams, and a failure to communicate with management. The first issue, was that, in creating a new application and pipeline(even though they are using the same tools suite like Jenkins, Sonar, fortift etc.) without working closely with the application development teams that already had an established pipeline made the demo ineffective; we were demoing tools and apps that would never actually be used. Now, the simplest solution would have been to use the configuration management tool (in this case, run the Chef cookbooks we had already used to create the pipeline on premise). Had this not been an option, the cloud team should have understood that a pipeline is more than just a tool stack of applications like Jenkins, Fortify and Sonar. When they re-created a pipeline there are many questions they needed to have asked and resolved, including: are they using pipeline as code or a traditional freestyle Jenkins job, what are the plugins that Jenkins and the other tools needed, what are the technologies contained in the applications, how are things compiled, what are the gates you have set up, how are applications being deployed, and more.

Delayed Security/Compliance Involvement

The security team took several weeks to scan through the pipeline. It was known that this was going to be a destructive scan, so no matter what, the cloud team knew they were going to have to rebuild a large portion of their infrastructure, but, on top of that, security came back with a LOT of things that needed to change. From complicated network fixes, to updated authentication methods within the pipeline applications themselves, these remediations took months worth of work. They were forced to go back to the drawing board and re-do a lot of the work they had initially done to set up the environment in the first place.

HINDSIGHT RESOLUTION

When designing anything, security needs to be involved from day one. It is ALWAYS better to think about security at architecture time, rather than trying to squeeze it on top.

3
The Pros and Cons of a Serverless DevOps Solution

5
Migrating to the Cloud: Which Model Is Right for You?

6
6 Major Challenges of Cloud Computing

8
The Advantages of Serverless Cloud Providers

10
Ensure That Your Current Cloud Solution Will Stand the Test of Time

11
Lessons Learned from an Enterprise Government Cutover to the Cloud

14
Quality Checks to Address Before, During, and After Cloud Migration

15
Serverless Technology and Integration with DevOps: An Interview with Glenn Buckholz

17
Additional Resources

Backup and Cutover

The cloud team and our team eventually worked through all of the issues, got an ATO, and we were able to demo the simplest application we had, to successfully go through the cloud pipeline. The pipeline was not ready to run any other applications, and several technologies were still not installed that many of our other apps would need. (.Net was available but we still couldn't migrate any code that used Java or Node and some of our functional and user experience testing tools weren't available yet). But because the government had 1 successful demo, and communication was still an issue, the high level agency management decided to completely shut down access to the on-premise dev environment, with one month notice. The government direction that trickled down to our contract management(a week later), was something along the lines of: "No need to tell developers that dev is going away, we don't want to worry them, we will just be replicating your dev environment for you, no need to do anything." Fun fact, they didn't replicate or backup anything for us.

With around 2 and a half weeks left, we had to make a plan to backup all of the CICD pipeline things that we couldn't lose when they turned off the environment, but they had given us nowhere to put anything. First, we reached out to the cloud team and were able to move our code repositories over to their TFS server. Then we had to figure out where to backup all of the historical data we would need to keep, focusing mostly on the Sonar and Fortify databases, and the Nexus Repository manager(which holds all of the artifacts we had ever built). With the short deadline, and not being given a place to put anything, we had to get creative in finding places to store everything, and it was a mess.

HINDSIGHT RESOLUTION

While having a hard cutover date isn't a bad idea, the government should have confirmed that everything was ready to handle the cutover on the cloud side. On top of that, a sufficient place to backup any historical data from the pipeline, is a must. That way a real backup plan could be created, and if the actual migration puts the developers in a limbo state where on-premise is gone, but the cloud isn't quite ready, we will be able to recover eventually.

In Conclusion

While I could go on forever about all the downfalls of the cutover, and how to avoid them, I think you should take 4 key points with you when migrating to the cloud.

1. Use a configuration management tool, it just makes everything easier.
2. Communication is your friend, make sure everyone is on the same page, and when giving a demo of your progress, be clear what is still needed to be done.
3. Get security involved early and often, its always easiest to create things in a secure way than it is to have to go back and try to wiggle security on top of what you have, or in many cases, force you to re-do things you thought were done.
4. Come up with a real cutover and backup plan. Make sure the teams have ample time, space, and access to migrate and backup all their data and tools into the new cloud environment before cutting access to your on-premise environment.

3
The Pros and Cons of a Serverless DevOps Solution

5
Migrating to the Cloud: Which Model Is Right for You?

6
6 Major Challenges of Cloud Computing

8
The Advantages of Serverless Cloud Providers

10
Ensure That Your Current Cloud Solution Will Stand the Test of Time

11
Lessons Learned from an Enterprise Government Cutover to the Cloud

14
Quality Checks to Address Before, During, and After Cloud Migration

15
Serverless Technology and Integration with DevOps: An Interview with Glenn Buckholz

17
Additional Resources

Quality Checks to Address Before, During, and After Cloud Migration

By Mukesh Sharma

Cloud computing has become an inevitable solution in today's world. While it is a no-brainer that most organizations have either migrated to or are considering a move to the cloud, the stakes for cloud providers and consumers are quite high. These facts make the cloud computing world very nimble and dynamic on the business and implementation sides. For example, there continues to be a strong competition among providers, with each trying to build an edge over the other in every way possible.

Gartner's latest report continues to rank Amazon AWS and Microsoft Azure in a close race for first and second positions, while Google is still a distant third. Google, on the other hand, is weighing in all options, with a strong acquisition strategy to push its way up the rankings. While Amazon and Microsoft's positioning are clearly above the rest, this does not give them complete market monopoly—they continue to keep innovating and executing to retain their leaderboard positions.

From an adopter's standpoint, the cloud has immense value to bring to the table, but a cloud migration effort is no small feat. From reliability and quality standpoints, a lot has to be planned to ensure a smooth transition covering the before, during, and after migration phases, and the challenges to mitigate are non-negligible. Some of the core quality checks to address include:

Before Migration: requirements are mapped out over a proof of concept setup; test cases to test for smoke and regression are identified; component migration priority is determined; balance between the manual and automated test suites is finalized; known application issues are kept handy; integration checkpoints are finalized, live environment is benchmarked for performance; backup plan for downtime is chalked out; among others

From reliability and quality standpoints, a lot has to be planned to ensure a smooth transition covering the before, during, and after migration phases.

During Migration: sanity is tested on the proof of concept environment and the server load is gradually increased; identified tests are gradually and incrementally tested for in phases as and when migration happens; a roll back plan is in place; live data is collected as required during the migration

After Migration: once the cut off date is identified and completed, all identified tests are run; data before and after migration is compared and signed off; reliability of data, uptime, and integration of modules are all monitored on an ongoing basis; service level agreements guaranteed by the cloud service provider

The value from moving to the cloud needs no selling; however, the entire cloud business is a very competitive space that requires close attention and involvement of all entities, whether you are a cloud provider or a consumer.

3
The Pros and Cons of a Serverless DevOps Solution

5
Migrating to the Cloud: Which Model Is Right for You?

6
6 Major Challenges of Cloud Computing

8
The Advantages of Serverless Cloud Providers

10
Ensure That Your Current Cloud Solution Will Stand the Test of Time

11
Lessons Learned from an Enterprise Government Cutover to the Cloud

14
Quality Checks to Address Before, During, and After Cloud Migration

15
Serverless Technology and Integration with DevOps: An Interview with Glenn Buckholz

17
Additional Resources

Serverless Technology and Integration with DevOps:

An Interview with Glenn Buckholz

By Jennifer Bonine

Glenn Buckholz, a technical manager at Coveros Inc., discusses methods to gain an understanding of serverless technology, the motivation behind businesses moving to serverless technology, and how a serverless infrastructure changes your testing strategy and bug reports.

Glenn Buckholz

We're seeing a trend where more and more companies are moving towards what we call serverless technology. Now, it's a bit of a misnomer. There are actually servers. That's one of the things that go into the talk: how does a serverless infrastructure change my testing strategy? How does a serverless infrastructure change bug reports? How does serverless infrastructure change what I as a tester have to know so that I can, you know, deliver and help make sure the quality product is coming out? What are the motivations behind businesses moving to serverless technology because not everything fits the serverless mold? So you want to be able to recognize whether or not in your particular niche, if people are going to be looking at serverless to replace what they're currently doing.

So ultimately, everything revolves around money. However, that's a bit too broad. Specifically, what service serverless technology allows you to do is it allows you to limit the number of staff you need to maintain an active and running application. Now, this is not for free. You do have to cede some freedom, you don't have as much control over what you're doing as if you had everything in house. But for most people, that fits their business case. So you can leave things like a good portion of security and a good portion of maintenance and updates and security patching and OS patching all to the cloud vendor. And those are no longer responsibilities, you have to take on yourself, so that if you can do the majority of your work on a serverless infrastructure, you can shift those roles more towards development and testing to make sure that you're delivering features faster and that they're of high quality.

3 The Pros and Cons of a Serverless DevOps Solution

5 Migrating to the Cloud: Which Model Is Right for You?

6 6 Major Challenges of Cloud Computing

8 The Advantages of Serverless Cloud Providers

10 Ensure That Your Current Cloud Solution Will Stand the Test of Time

11 Lessons Learned from an Enterprise Government Cutover to the Cloud

14 Quality Checks to Address Before, During, and After Cloud Migration

15 Serverless Technology and Integration with DevOps: An Interview with Glenn Buckholz

17 Additional Resources

As you're transitioning from a more traditional architecture to a serverless architecture, any tips or techniques for people, if they say "I'm doing this right now. We're actually starting that transition," and what they need to look out for as opposed to "I'm transitioned, I'm living in this world, what changes day to day for me?"

Jennifer Bonine

Glenn Buckholz

So, there are a couple of things here. If you fully embrace the DevOps mindset, which not everybody has, you should probably think about moving more in that direction. The reason being, if you have a very qualified DevOps team, you don't have to worry about changing skill sets: these people will be able to handle and deal with the technology, because it looks very similar to a lot of the things that they're already doing. So that would be one thing is that you should value your DevOps team, because they're going to be instrumental in easily transitioning you over to a serverless world. Then, the second thing is be cognizant of what the limitations of your cloud providers serverless technologies are. It's not a fit for every single application out there. For the ones it is, it's a great advantage. But for the ones it isn't, you need to know up front, so that you don't have a false start where you get 80% of the way, and you're like, "Oh, well, it can't do this one thing" and that one thing is business critical, so you have to scrap the whole effort.

If you haven't embraced microservices as a way that you're structuring your applications and how you're operating, are those precursors or things people need to step back and make sure they do first before this is a fit?

Jennifer Bonine

Glenn Buckholz

DevOps helps with process and people. You're going to have a much easier time understanding the technology internally if you've already gone in that direction in terms of your people, but it's not necessarily a requirement. However, serverless technology does require that you break down your application into tiny bits, and if you haven't already taken that as a design consideration in your architecture, then it's something that you are going to have to do as a prerequisite.

READ THE FULL INTERVIEW

3
The Pros and Cons of a Serverless DevOps Solution

5
Migrating to the Cloud: Which Model Is Right for You?

6
6 Major Challenges of Cloud Computing

8
The Advantages of Serverless Cloud Providers

10
Ensure That Your Current Cloud Solution Will Stand the Test of Time

11
Lessons Learned from an Enterprise Government Cutover to the Cloud

14
Quality Checks to Address Before, During, and After Cloud Migration

15
Serverless Technology and Integration with DevOps: An Interview with Glenn Buckholz

17
Additional Resources

Additional Resources

MORE INFORMATION FOR SOFTWARE PROFESSIONALS

TECHWELL™

TechWell provides continual improvement and learning resources for software professionals. With conferences, training, web seminars, and more, we have lots of options to help you keep up with what is happening in the software development industry. TechWell Insights publishes the latest stories on the cloud, testing, agile, development, and more from software industry professionals. Subscribe to the TechWell Insights weekly newsletter and get the most recent and relevant articles and information directly to your inbox.

JOIN HERE

NARROW YOUR SEARCH TO A SPECIFIC TYPE OF RESOURCE:

StickyMinds Articles

StickyMinds articles cover a wide range of software testing topics, including testing in DevOps, continuous testing, test automation, agile testing, test process improvement, and much more. **Click here** to read articles about testing in the cloud on StickyMinds.

Interviews

Each year, TechWell interviews dozens of software professionals, including well-known thought leaders, seasoned practitioners, and respected conference speakers. **Click here** to read, listen to, and watch interviews with testing experts about cloud computing.

TechWell Insights

Read stories about cloud development, virtualization and containerization, testing, agile, teams, and more, all written by software industry professionals. New stories are added each weekday, so **click here** to read the latest—or sign up to receive the weekly newsletter roundup of the newest stories.

TechWell Conference Presentations

Couldn't make it to a TechWell conference to sharpen your testing knowledge? TechWell conference presentations are available to StickyMinds members for free soon after conferences end. **Click here** to join StickyMinds and access conference presentations related to all aspects of the cloud.

TechWell Conferences

TechWell conferences feature keynote presentations, tutorials, and classes covering cloud computing, testing, agile, DevOps, test automation, agile testing, people management, and more. Learn from experts in the field and network with your peers to get the most immersive conference experience possible. **Click here** to learn more.

The TechWell Hub

Join the active and engaging TechWell Hub community on Slack to keep in touch with what's happening in the industry and ask questions about your day-to-day challenges. The Hub offers a safe space to get real-time answers from other software professionals in your shoes, and even experts in the field! **Click here** to join for free.

coveros

Accelerate Delivery

Our partner, Coveros, offers training, coaching, and consulting to help organizations implement agile and DevOps solutions in data centers, on-premise, or in the cloud. Whether you're just getting started with the cloud, building on existing AWS skills, or preparing your team for an enterprise transformation, their AWS Training and Certification courses can help you be more effective and do more. Additionally, Coveros offers courses in agile development, DevOps, automation, security, development and testing tools, and fundamental and specialized software testing topics—all of which include best practices taught by industry leaders.

DevOps courses | **AWS courses** | **DevOps Engineering courses** | **DevSecOps courses**