PERFORMANCE TESTING AND APM IN THE AWS CLOUD

APPDYNAMICS

APPDYNAMICS

INTRODUCTION

This eBook discusses the history of application performance testing, and with that backdrop, offers advice on combining performance testing and Application Performance Monitoring (APM) in the AWS cloud.

CHAPTER 1: PERFORMANCE TESTING AND APM BEFORE THE CLOUD

PERFORMANCE TEST AUTOMATION

Performance testing has always been about ensuring the scalability of a software application. Until the arrival of the first performance test automation solutions in the late 90's, performance testing was a manual process that was difficult, if not impossible, to test in a consistent and reliable fashion.

The arrival of these new tool sets suddenly allowed software testers to turn discrete user actions into scripts that could be combined and replayed as test scenarios. Solving the consistency and reliability challenge, software testers could now repeat the same test on demand while reinforcing and imposing some new requirements:

LOAD INJECTION

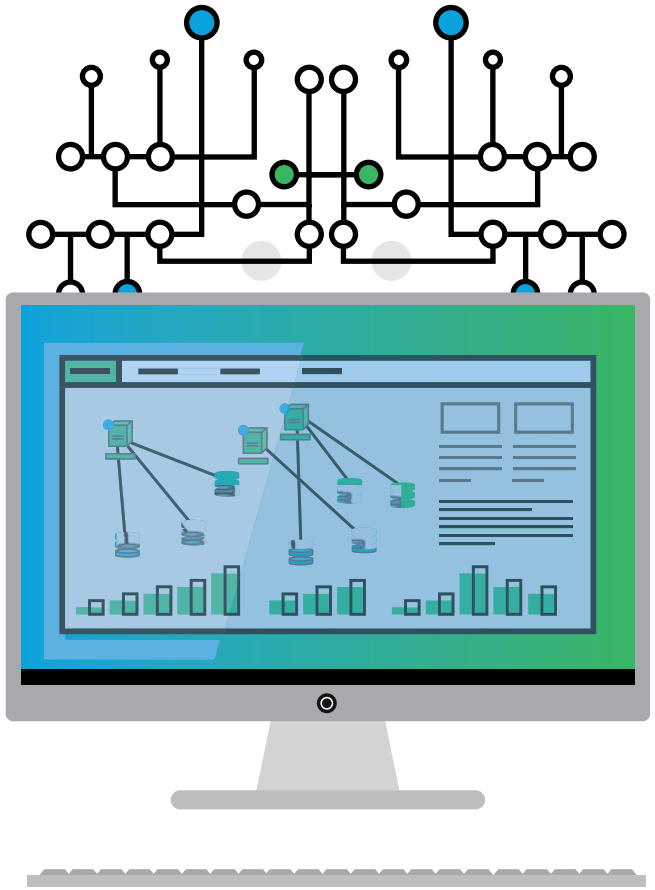
For each performance test, you needed to provision sufficient hardware to act as load injectors. A small test simulating the load of say 200 end users could be easily accommodated by a few PC's. This solution, however, was unable to rapidly scale to meet the test requirements of the enterprise – potentially escalating to hundreds or even thousands of PC's or servers.

TECHNOLOGY COULD ALSO IMPOSE FURTHER CONSTRAINTS (E.G CITRIX AND SAPGUI), LIMITING THE NUMBER OF VIRTUAL USERS THAT COULD BE GENERATED FROM A GIVEN HARDWARE PLATFORM REGARDLESS OF SPECIFICATION REQUIRING EVEN MORE INJECTION RESOURCE.

TEST ENVIRONMENTS

As performance test automation became a reality, the need to create realistic performance test environments became significantly more important. The ideal environment for end-to-end testing has always been an exact or close copy of production. However even today this is often difficult to achieve for many businesses. The cost of provisioning and maintaining such an environment is simply unsustainable, as both requirements rely on the availability of physical hardware provisioning.

Today, hardware has never been cheaper with enterprises leveraging it to support virtual server environments. However, 10-15 years ago this was not the case, the business cost of provisioning sufficient hardware for load injectors and performance test environments was exorbitant.



Even if cost was not an issue, simply sourcing enough hardware with the correct specification at short notice was very challenging.

APPEARANCE OF APM

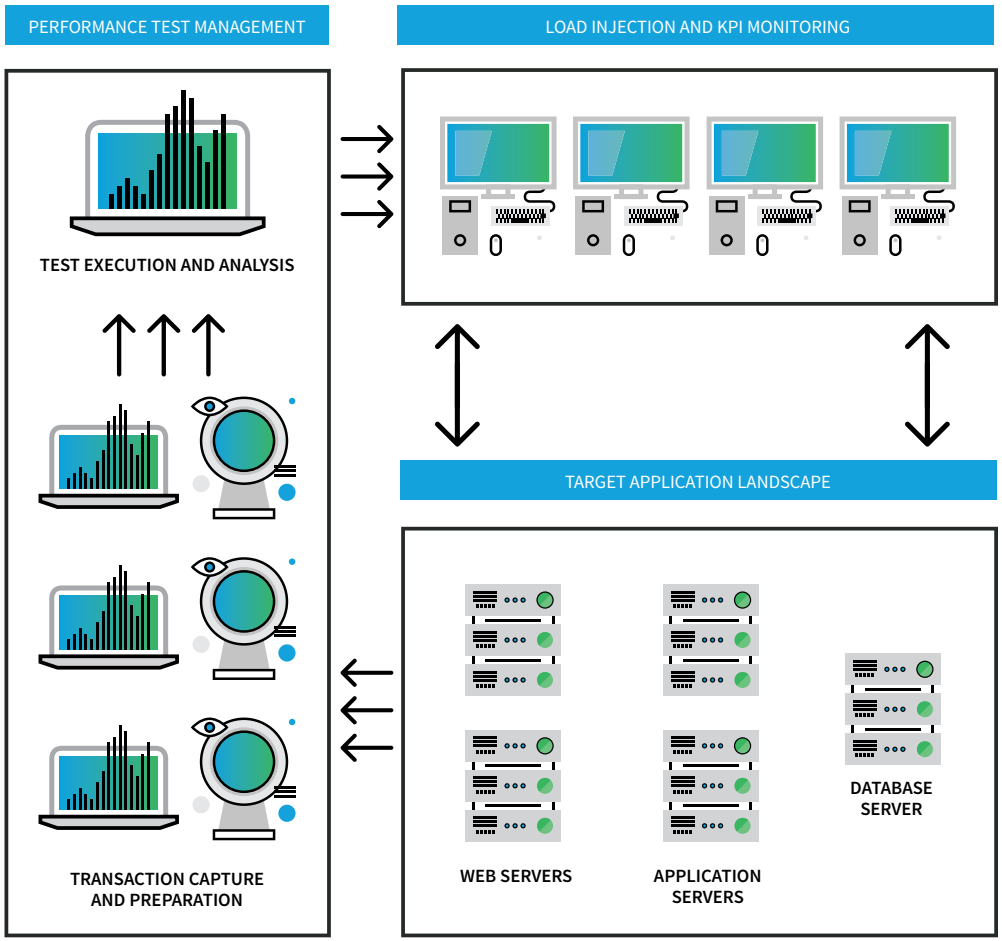
During this same time, the first APM solutions were emerging. These first-generation tool sets grew out of code profiling tools looking to promote the run-time analysis of software at the class and method level beyond development. They were very much stand-alone solutions with little or no ability to integrate with early test automation offerings.

Another downside of these early APM offerings was they tended to generate considerable CPU and memory overhead on the servers that were instrumented. They had to be used with caution particularly in production environments. By comparison, the current generation of APM has a very light footprint and can be used in production with confidence.

ANALYSIS BREAKTHROUGH

Despite these initial shortcomings, effective analysis is a key part of any performance testing capability and APM offered for the first time the ability to include code-level performance in the analysis process. Even though there was no direct sharing of data, you could still deploy APM into your performance test environment to observe application performance under load.

TODAY, THIS DEEP-DIVE CLASS / METHOD VIEW OF APPLICATION BEHAVIOR REMAINS ONE APM'S KEY BENEFITS.



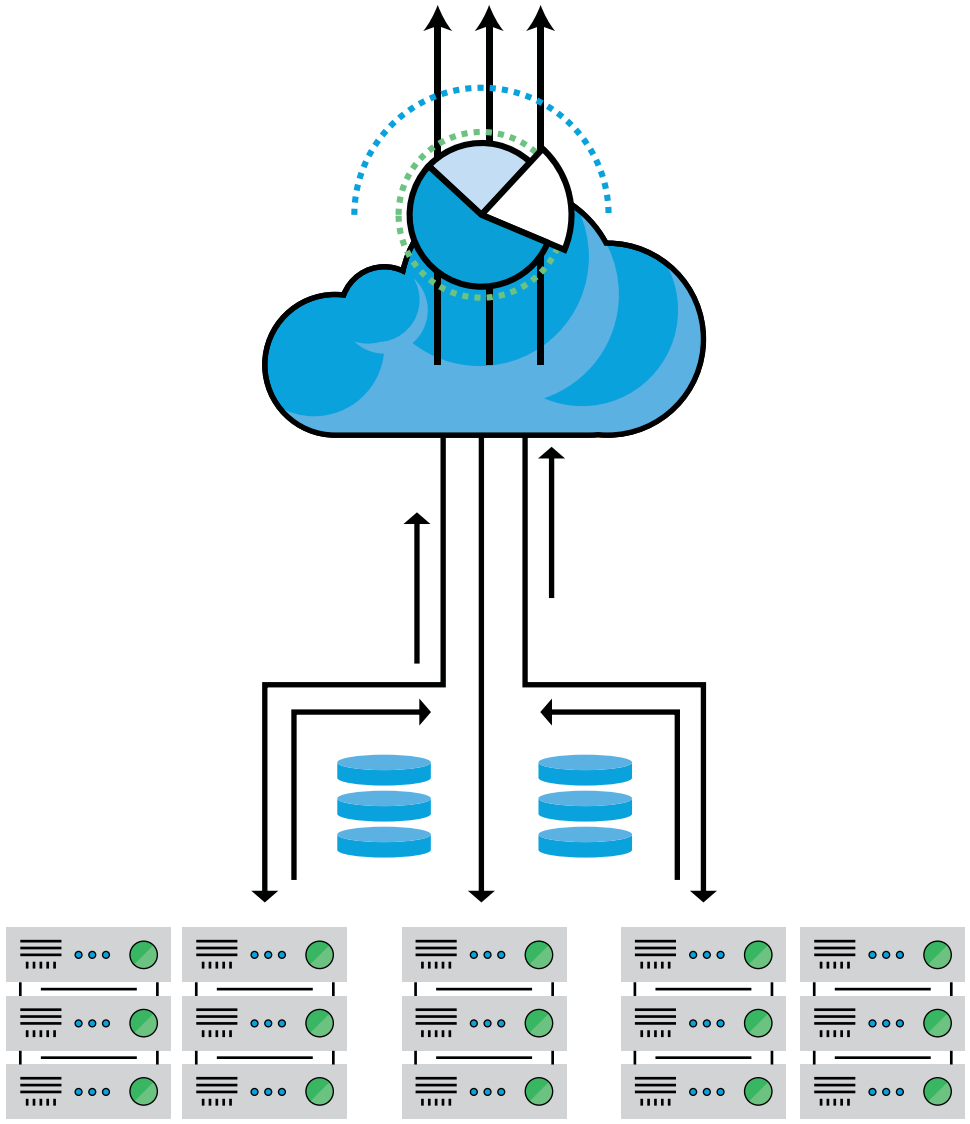
CHAPTER 2: THE CLOUD'S IMPACT ON APM AND PERFORMANCE TESTING

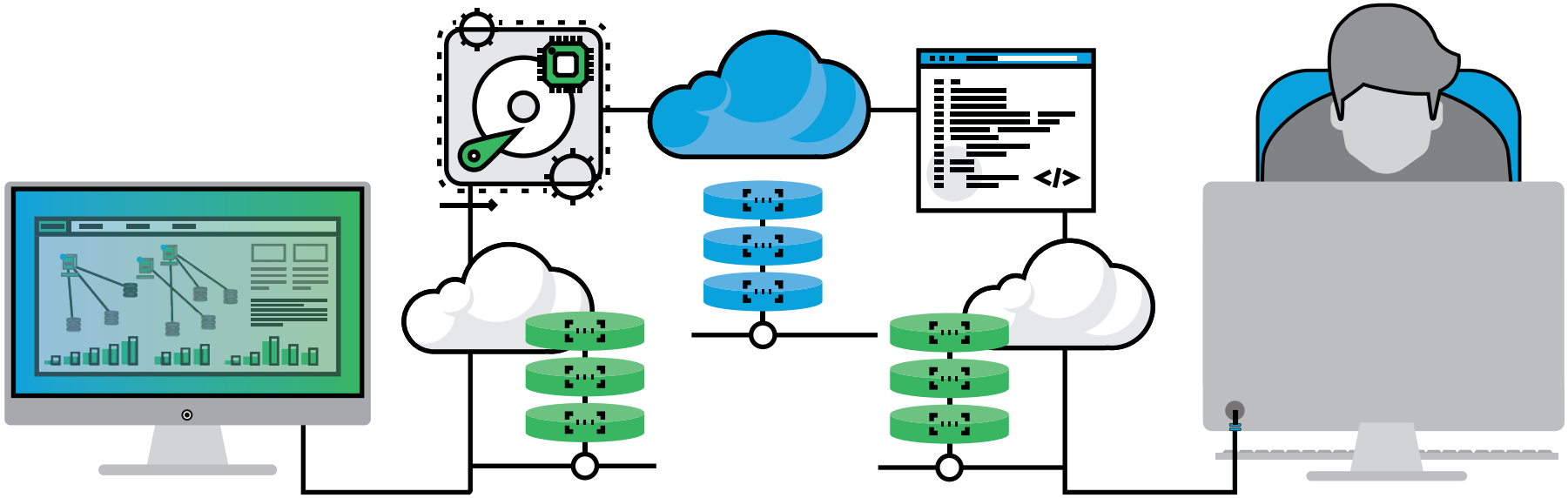
ENTER AMAZON AWS

When Amazon introduced Amazon Web Services (AWS) in early 2006 the cloud revolution began in earnest. The cloud has arguably had the most disruptive influence on IT since the appearance of the World Wide Web. At a stroke, building and deploying an application to the web was available to everyone.

IMPACT ON ENVIRONMENT PROVISIONING

The emergence of the cloud has enabled provisioning and scaling of environments pretty much on demand at whatever level of persistence required (hours, days, ongoing). Importantly this includes not just production deployments but also dedicated environments for functional and non-functional software testing. Such environments can easily be templated and spun up / spun down as required.





THE CLOUD'S IMPACT ON APM

The impact of the cloud on APM has been profound. The value of deep dive analysis into the web and application server tiers (and now database) remains significant but with the increasing dominance of web and mobile technology the ability to combine data center analysis with rich end-user data has become increasingly important. As a result, the focus for leading APM vendors has become the end user – giving a single view of application performance and how it impacts the end user experience. Dynamically baselining and alerting when end-user activity that deviates from expected levels of availability and performance.

From a business perspective, delivering an exceptional end user experience measurably affects conversion rates and overall business performance. This means that the application insight gained from APM analysis is now equally valuable from a technical and business perspective.

In fact, it is quite difficult to argue a case for not implementing APM based on the business value it delivers.

THE CLOUD'S IMPACT ON PERFORMANCE TESTING

The impact on performance testing has been two-fold. First, the problem of sourcing large numbers of load injectors at short notice has been greatly reduced. In AWS an AMI (server image) can be created, configured as a load injector for automation tools with cloud support, and then used to spin up as many load injectors as required for a given test scenario.

APART FROM A SMALL CHARGE TO MAINTAIN THE AMI IMAGE, TESTING COSTS ONLY APPLY WHILE THE LOAD INJECTOR INSTANCES ARE ACTIVE, SO A VERY LARGE TEST WILL NOT BE PROHIBITIVELY EXPENSIVE AS ALONG AS THE INJECTORS ARE SHUT DOWN AT THE END OF EACH TEST CYCLE.

Second, with web deployment for eCommerce becoming the dominant application technology, it is now possible to provision performance test environments to production scale (at least at the web server tier) with the same cost advantages.

In addition, you can easily provision multiple performance test environments for use in Dev and QA that can be scaled to production levels and beyond for early testing of performance SLA compliance at a software component level.

CHANGES IN PERFORMANCE TEST TOOLING

CLOUD SUPPORT

With the rapid rise of the cloud, performance tooling vendors have had to adapt to offer support for cloud-based load injectors. Most leading licensed and open source solutions now provide some level of support, from buying up-front cloud time to integrating self-managed injector instances in AWS.

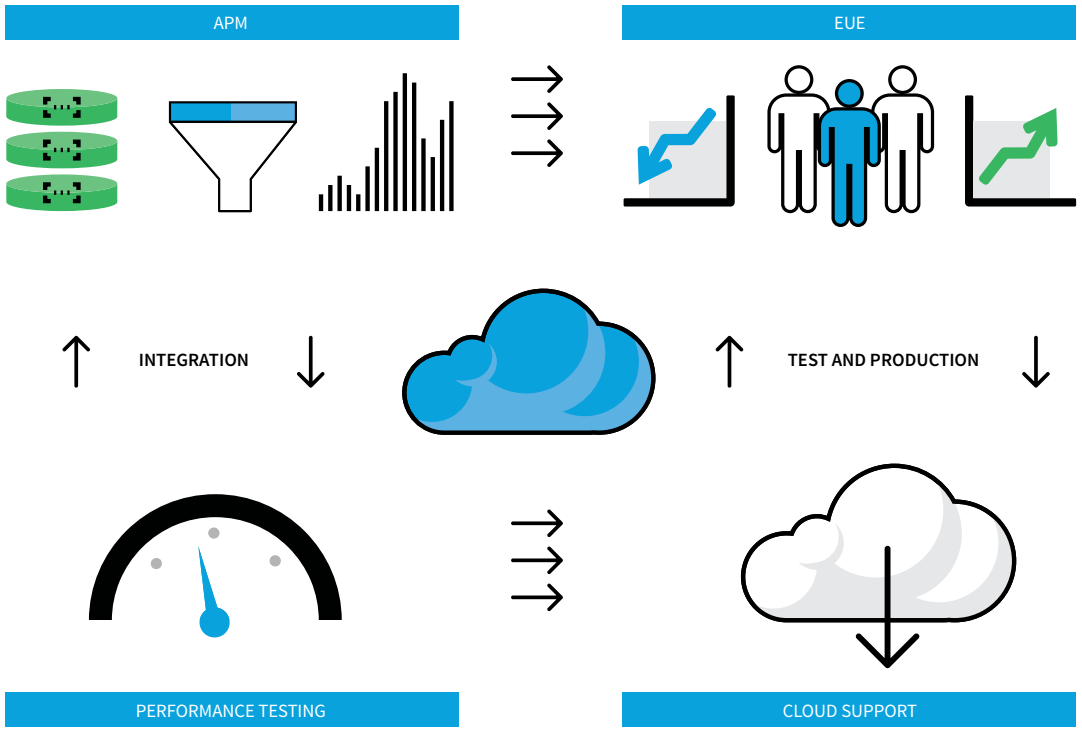
IN MOST CASES, IT IS NOW ALSO POSSIBLE TO MIX CLOUD-BASED AND ON-PREMISE LOAD INJECTORS IN THE SAME TEST CONFIGURATION.

APM INTEGRATION

As discussed in Chapter 1, early APM offerings had little or no integration with performance test automation. Most leading performance automation tool vendors now offer at least some level of integration.

Out of the box this is typically web-focused and takes the form of insertion and detection of custom HTTP headers. This tagging approach allows application activity generated by performance test automation to be identified and reported on separately from other types of application traffic. It also permits discrete tracing of specific application activity distinct from background noise.

For more complex requirements there is often an API available that can be used to implement and integrate custom tagging.



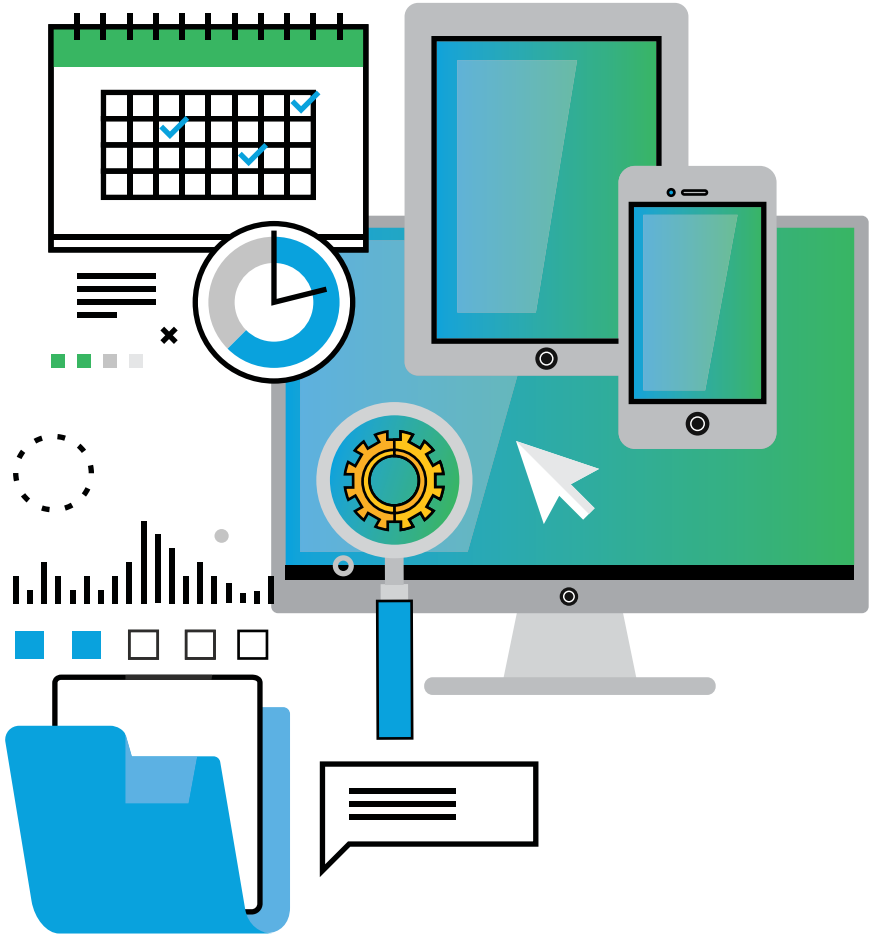
CHAPTER 3: PERFORMANCE TESTING AND APM IN THE CLOUD (AWS)

PERFORMANCE TESTING APPROACH

To be effective, Performance Testing of whatever variety must be based on sound principles and testing in the AWS cloud is no exception. Having worked as a performance consultant for many years I use the following guidelines to ensure I capture the essential pre-requisites:

Business

- Scope (strategic, tactical)
- Confirming expected deliverables (ensuring no surprises)
- Confirming client platforms to test (desktop, mobile, service consumers, batch)
- Identifying key use cases, test data requirements and test scenarios for each client platform.
- Identifying the key performance indicators (KPI) that must be captured to accurately measure application performance.
- Building an accurate workload model (Essential)
- Resourcing (in-house, outsourced)



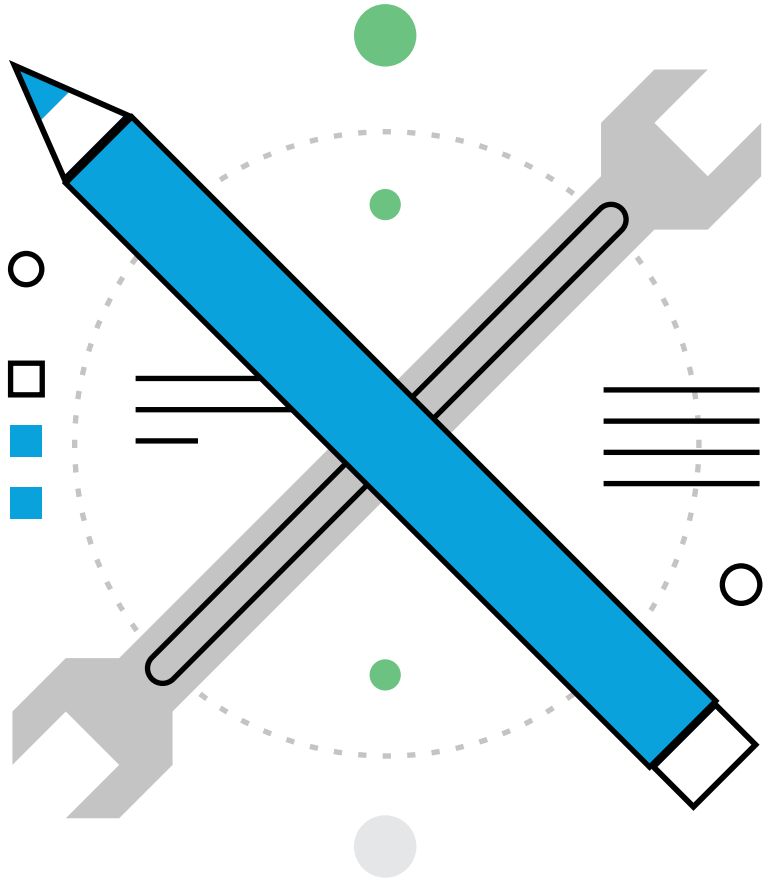
TECHNICAL

- Test environment management and provisioning
- Performance automation tooling selection appropriate for technical fit, skill set and budget (ideally via POC)
- Monitoring approach (ideally based on APM)
- Data collection, analysis, correlation and reporting

THE KEY IS TO BUILD YOUR TESTING ASSETS SO THEY ARE A TRUE REPRESENTATION OF APPLICATION BEHAVIOR AND THEN TO TEST AGAINST AN APPLICATION DEPLOYMENT IN AN ENVIRONMENT THAT IS APPROPRIATE FOR THE LEVEL OF TESTING REQUIRED.

For end-to-end testing, this should be a copy or close approximation of production. For other testing requirements, such as component level performance testing in Dev, this could be an environment of sufficient scale to test SLA compliance for a discrete subset of application functionality.

Finally, you need to ensure that you can capture and correlate the KPI metrics you need from the application and the environment, otherwise your performance analysis will be incomplete and potentially misleading.



CLOUD SPECIFIC CONSIDERATIONS FOR PERFORMANCE TESTING

In Chapter 2 we discussed how the cloud has greatly simplified server provisioning for load injectors and test environments but there are some other important considerations we need to address including:

- Environment management
- Geo-location
- The impact of Content Delivery Networks (CDN)
- Security

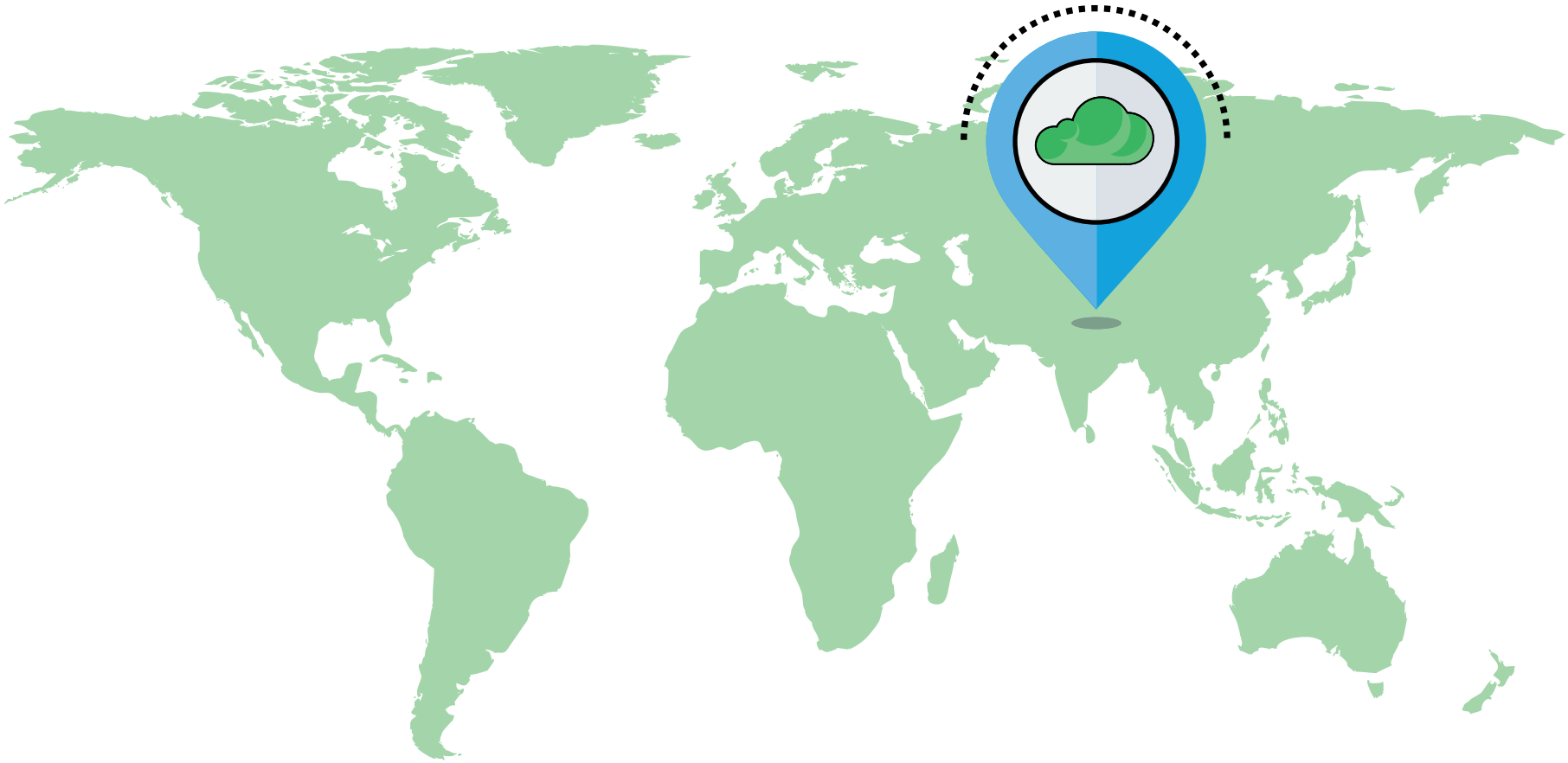
ENVIRONMENT MANAGEMENT

The cloud is flexible and convenient, but like any service it comes at a cost so it is very important to regularly curate your environments and running instances.

It is all too easy to leave those 100 load injector instances running at the end of a test cycle only to find out a day or so later that your cloud budget of $100 has "scaled" to become $10,000.

In the same vein only keep test environments active as long they serve a useful purpose. Using cloud to enable functional and non-functional testing across Dev teams is great for productivity but unless carefully managed can be a recipe for unexpectedly inflated cloud costs.

It really is curious how so many businesses persist in having little or no central management of test environments when this is straight-forward to implement and highly beneficial.



GEO-LOCATION

For a multi-national online business deploying to the cloud greatly simplifies the provisioning of services to a global audience. However when performance testing it is important to configure load injection that is representative of the expected user demographic.

Most cloud providers offer multiple availability zones or equivalent that allow you to position injector instances in different international geographies. You need to ensure that you have a realistic distribution of injectors across these availability zones if this is a requirement of your test scenario.

If you are unsure about your user distribution then APM can help by providing a detailed breakdown of end user activity across local and international geographies.

THE IMPACT OF CONTENT DELIVERY NETWORKS (CDN)

Content Delivery Networks have effectively become a must-have for any large-scale cloud web deployment greatly reducing the burden of hosting and serving static and selected dynamic content from the data center. The challenge this presents to performance testing is whether to include or ignore content that from an end user perspective would be sourced in the first instance from the CDN provider.

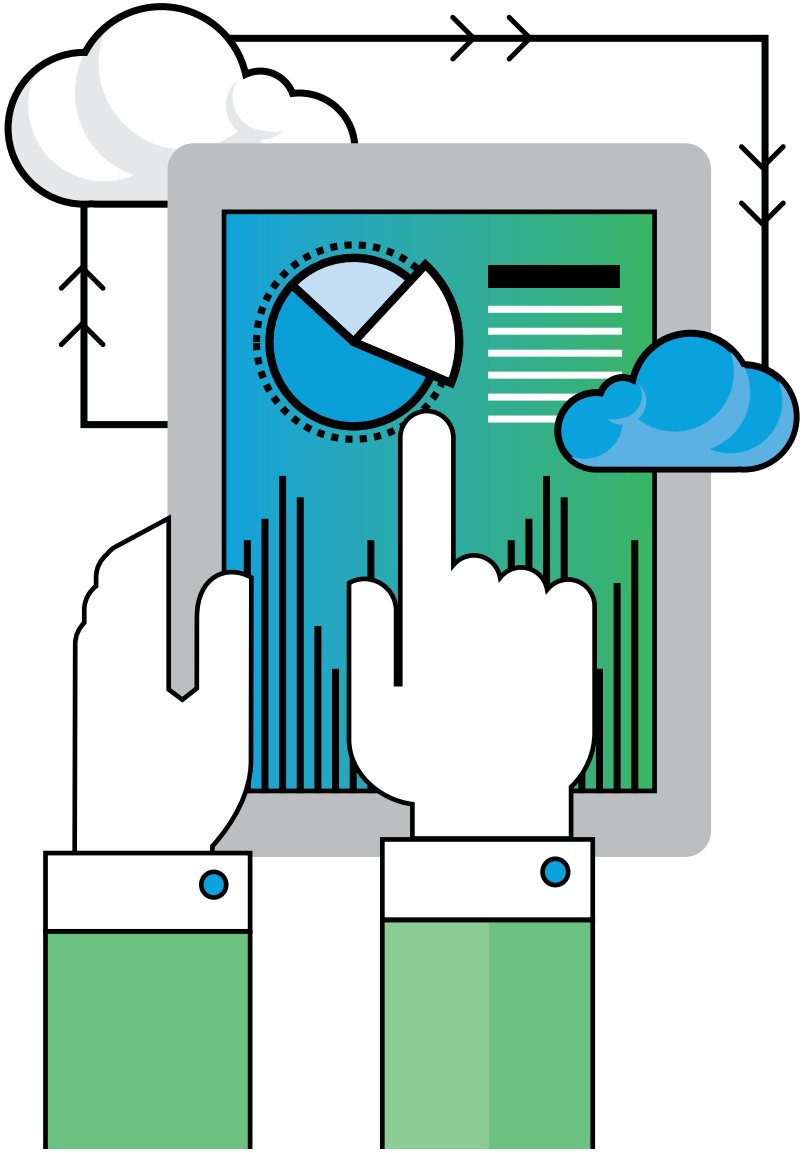
I have been involved in more than one lively discussion on the relative merits of inclusion or exclusion so would offer the following observations:

- If you choose to ignore CDN content your test results could be considered more reflective of the native capacity of your hosting environment but are based on the assumption that your CDN provider will deliver on any contractual SLA's around performance and availability.

- If you include CDN based content it can be argued this is a more realistic approach but bear in mind your test results will be impacted by the performance of your CDN provider which could mask load related problems deeper in your application stack.

Perhaps the best solution is to carry out exclusive and inclusive tests if time and budget permit paying close attention to performance delta.

By testing first with CDN excluded any issues introduced by CDN sourced content should become clear. You also get to validate that your CDN configuration is correct and your end users in Germany are not viewing content in French.



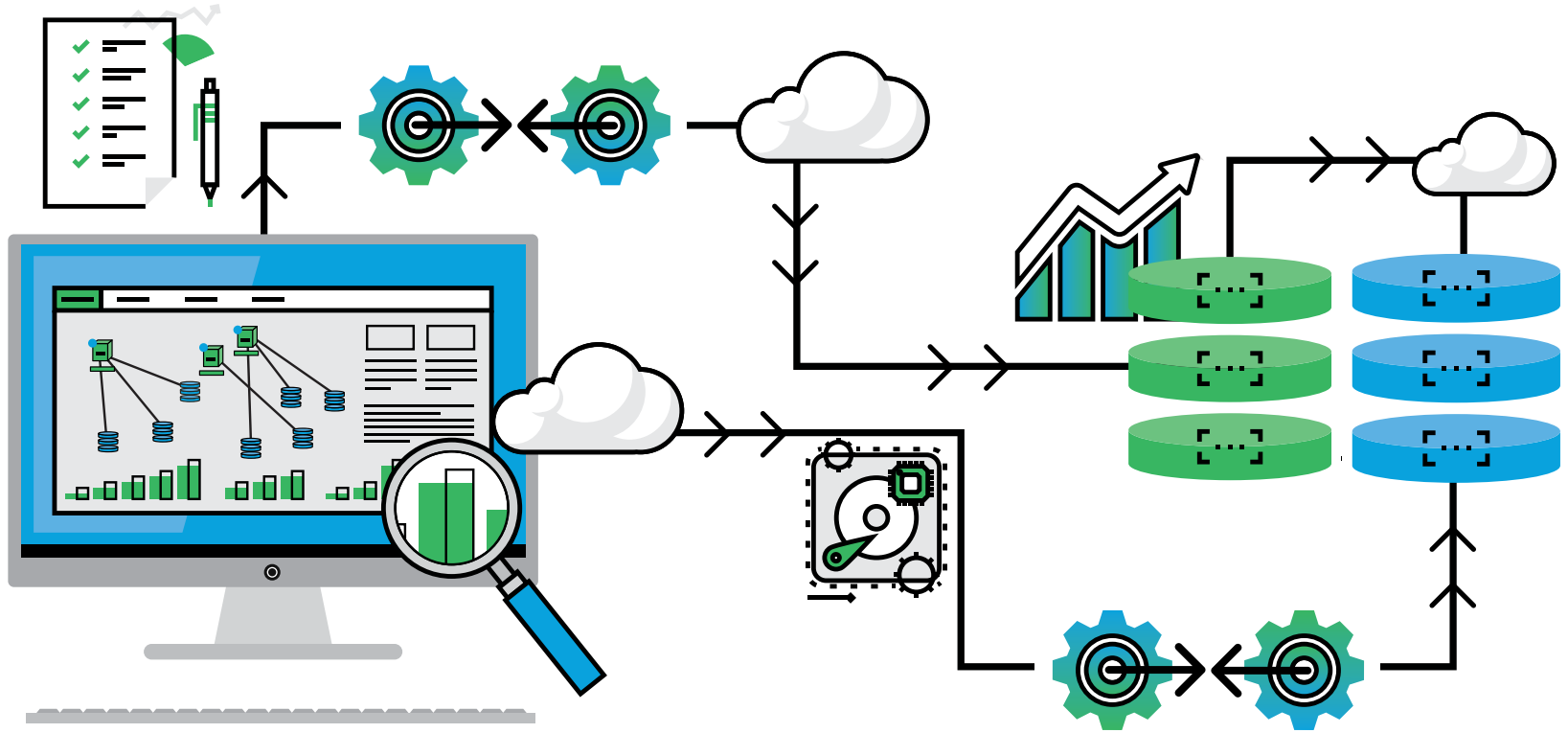
IN BOTH CASES APM CAN ASSIST IN ANALYZING PROBLEMS AND IDENTIFYING IF THE CDN PROVIDER OR THE APPLICATION DEPLOYMENT ARE AT FAULT.

SECURITY

As using public cloud effectively involves hosting by a 3rd party security considerations are important when preparing to performance test.

These include providing your application owner with a whitelist of the load injector IP addresses that you will be using so that your performance test is not mistaken for a Denial of Service attack.

Deploying to the cloud without considering security makes you vulnerable. I'm sure that most businesses will follow recommended guidelines from AWS and other cloud vendors over the correct way to implement secure production deployments but don't cut corners with cloud-based test environments otherwise they can become easy targets for hackers.



LEVERAGING APM IN THE CLOUD

Cloud does not present any barriers to the value that APM provides.

Whether monitoring production websites or leveraging enhanced deep-dive analysis during performance testing, the latest generation of APM solutions builds on the original class-method view of application behavior by adding rich transactional and end-user analysis across highly distributed and diverse architectures.

Importantly it is possible to leverage APM across all types of testing environment from component level testing in Dev through QA and end-to-end testing.

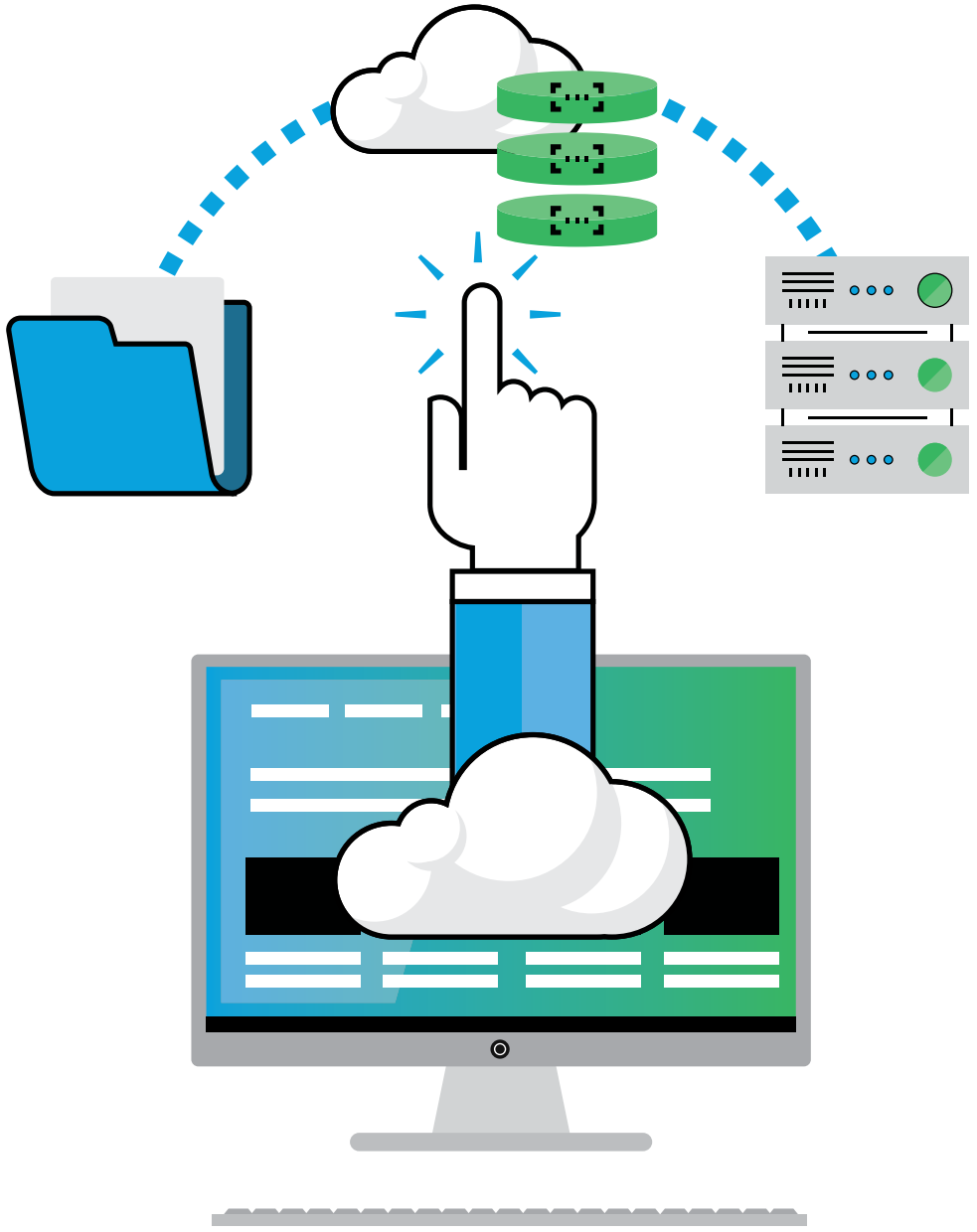
The metrics captured at each stage of testing can be rolled forward providing a consistent performance baseline to assess the impact on performance as the release is progressively assembled from its component parts.

THE FINAL PERFORMANCE METRICS DERIVED FROM END-TO-END TESTING CAN BE USED AS THE BASIS OF PRODUCTION MONITORING.

APM AND AWS

Amazon remains the largest and most mature cloud vendor despite fierce competition. They arguably provide the most diverse choice of server platforms and supporting services and host some of the most complex cloud deployments.

ONE OF THE GREAT STRENGTHS OF APM IS THE ABILITY TO **CORRELATE AND ANALYZE MULTIPLE DIVERSE DATA STREAMS.** AMAZON'S PUBLISHED API APPROACH TO THEIR APPLICATION AND SERVICE OFFERINGS MAKES IT VERY EASY TO DO THIS IN THE AMAZON CLOUD.



The following are a few examples:

SERVICE CONSUMPTION

Amazon offer CloudWatch as a generic application and host monitoring service. Most modern APM solutions already offer a default host monitoring capability as part of agent deployment but this is typically restricted to CPU, Memory, Disk, and Network I/O.

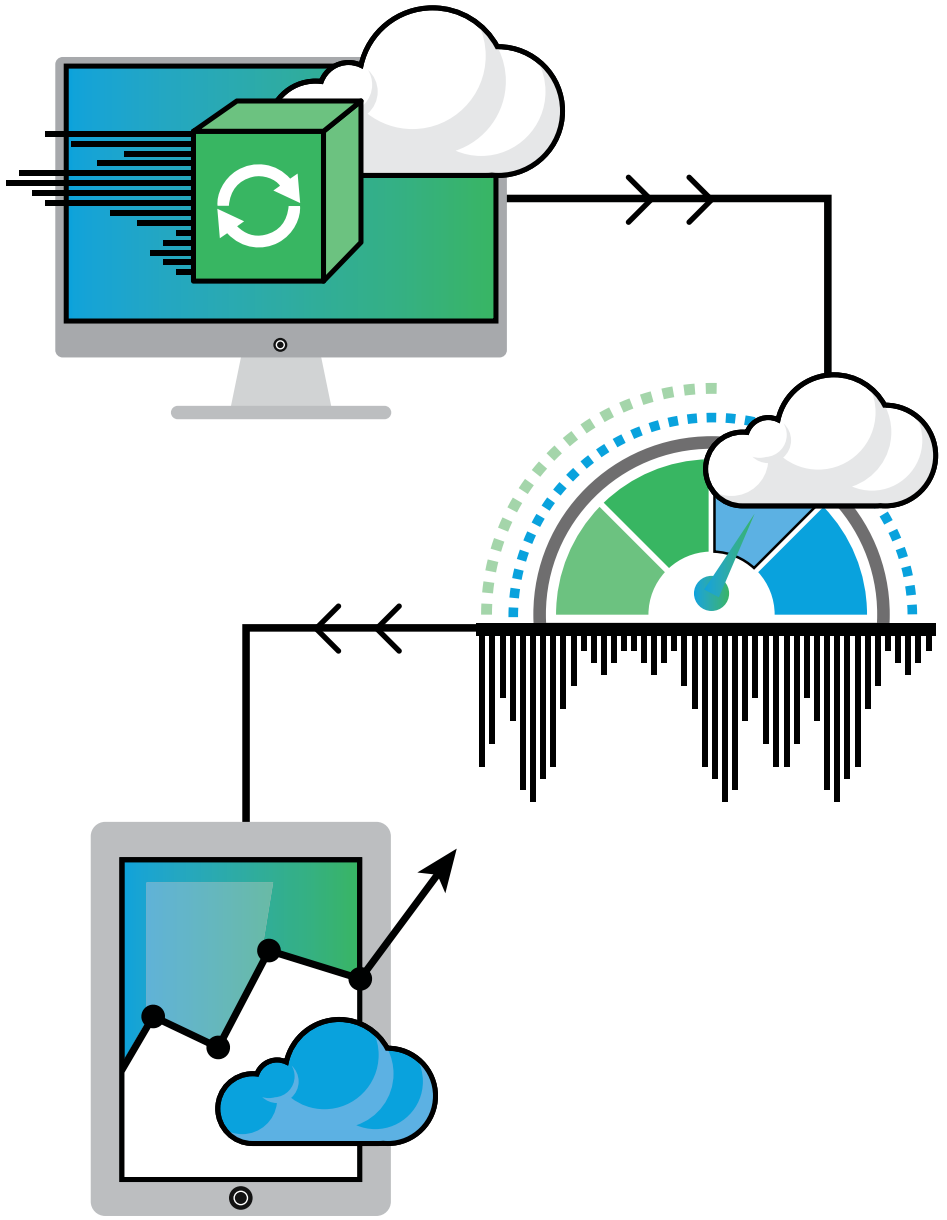
By using APM to leverage the CloudWatch API additional application and host metrics can be captured to enrich the view of application and host performance during production monitoring or performance testing.

NATIVE CDN MONITORING

Amazon has their own CDN offering in CloudFront. We can use APM to monitor the behavior and performance of CloudFront and include this data in our APM analysis.

NOSQL DATABASE MONITORING

There are many NoSQL database offerings available today and Amazon provides their own solution in DynamoDB. With APM we can natively monitor the performance of DynamoDB.



APM DEPLOYMENT CONSIDERATIONS IN THE CLOUD

Finally, implementing APM in the cloud is straight-forward enough providing you keep the following in mind:

AGENT LICENSING

The ability to spin up /spin down AWS instances on demand can have an impact on licensing. If your AMI includes an APM agent install then you could accidentally exceed your licensing entitlement if you are licensed for 10 concurrent agents and you happen to spin up 20 instances.

This was often an issue in the early days of cloud (not just for APM) so make sure you are covered for the number of APM agent instances you need.

Very large implementations can also hit hard-stops on the number of agent seats per account from some cloud vendors so correctly sizing your APM implementation for current and future requirements is very important.

SCALE

Early APM offerings just didn't scale well and somewhat ironically often ran into performance problems trying to deal with increasing numbers of agents, data storage and processing requirements.

Modern APM solutions seek to address this challenge by making agents light-weight and easy to install, compressing and rationalizing data capture, grouping agents into logical tiers and consolidating data transfer between agent and controller. Greatly increased flexibility in data storage and archiving options completes the picture.

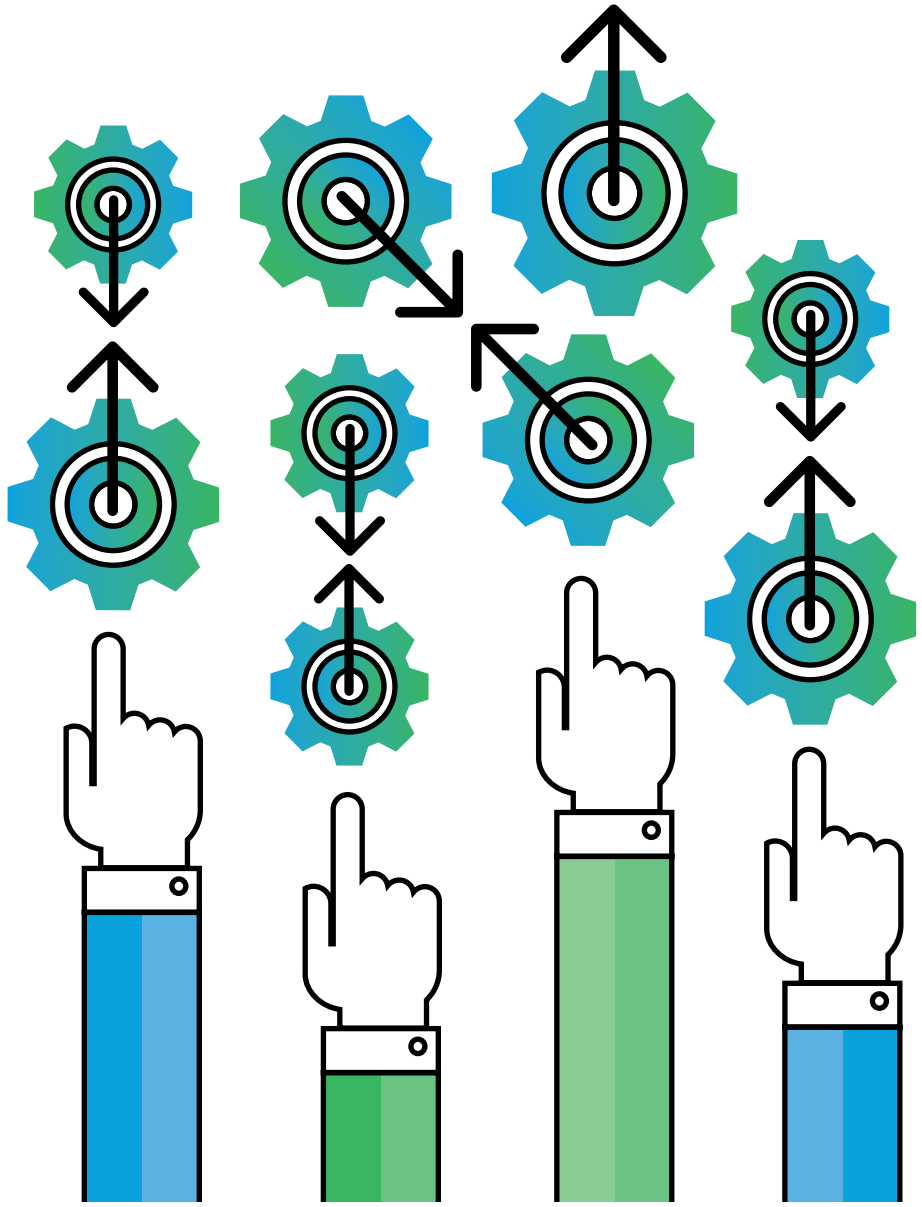
Make sure that your APM deployment will scale in line with anticipated business growth.

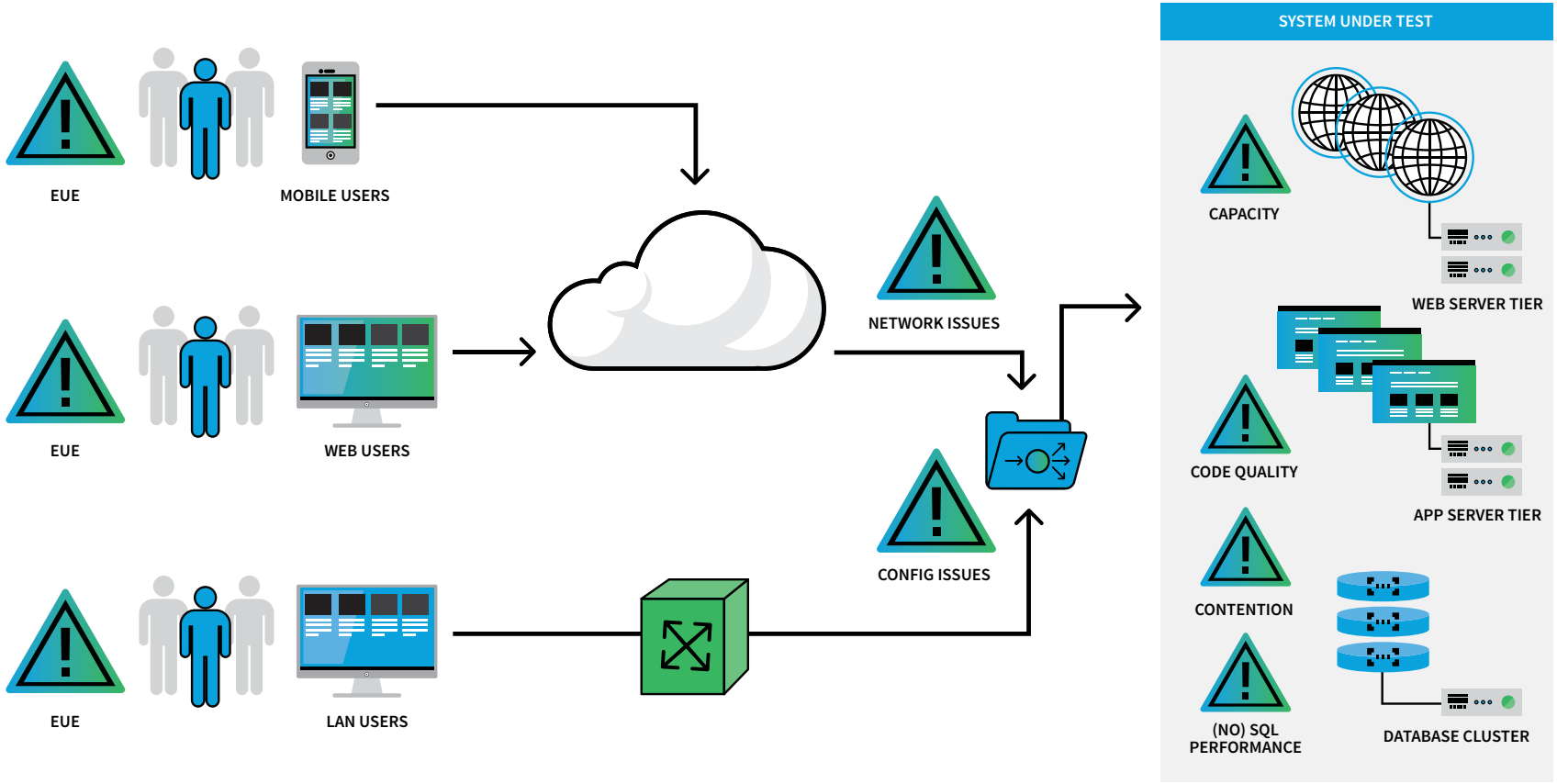
DEPLOYMENT MANAGEMENT

The ability to centrally manage your APM deployment in response to growth and change is critical. And if this can be achieved via API so much the better.

Management essentials include:

- Instrumentation: Automatically applying upgrades, downgrades, and configuration changes to agents while minimizing server downtime.
- Security: The ability to control user access to sensitive data ideally using a role-based approach. This is especially important in highly regulated environments.





NETWORK SECURITY

You can implement APM as a service or host the implementation yourself across a mix of cloud based and on premise servers. A potential pitfall is sec-ops push-back on opening ports on public facing firewalls to allow APM controllers and agents to communicate.

This tends to impact on-premise servers more than cloud but is not something you want to find out during implementation. To avoid this, facilitate a pre-adoption reality-check on expectations with all stakeholders over APM network access requirements.

Data anonymization could also be an issue but most APM vendors now have this well-covered allowing you to conceal IP addresses, host names and other potentially sensitive data.

ABOUT THE AUTHOR:

Ian Molyneaux has been involved in performance testing and APM for the past 15 years encompassing the arrival and subsequent normalization of cloud deployment and the impact this has had on performance testing, APM, and software testing in general.

CONCLUSION

Application performance management technologies have evolved substantially since the advent of mainstream cloud computing technologies.

The growth of Amazon Web Services presents new challenges for the APM sector, but the added flexibility and scalability of cloud ecosystems counterbalances those concerns and gives organizations opportunities to leverage sophisticated APM capabilities that weren't readily available in the past.

AWS and APM have matured side-by-side, leading to complementary capabilities that allow organizations to test applications with a greater degree of precision. As a result, AWS has emerged as a stable, productive environment for APM initiatives.

The AppyDynamics platform is built to drive application performance excellence in such an environment. Our solution provides end-to-end performance testing and monitoring for cloud applications, giving businesses the tools they need to accelerate and simplify performance management. Check out our AWS solution page or contact us to learn more about how app performance management can fuel business gains.

WWW.APPDYNAMICS.COM

1.415.442.8400
INFO@APPDYNAMICS.COM

APPDYNAMICS