ScienceLogic

5 Things to Know About Network Monitoring in a Multi-Cloud World

Introduction

We must evolve network monitoring to best serve new cloud computing environments. Following are five best practices to plan for current and future IT demands.

1. Monitoring services, not devices
2. Monitor within the cloud and external to the cloud
3. Work with DevOps to enable Day 2 operations
4. Embrace automation
5. Avoid cloud vendor lock

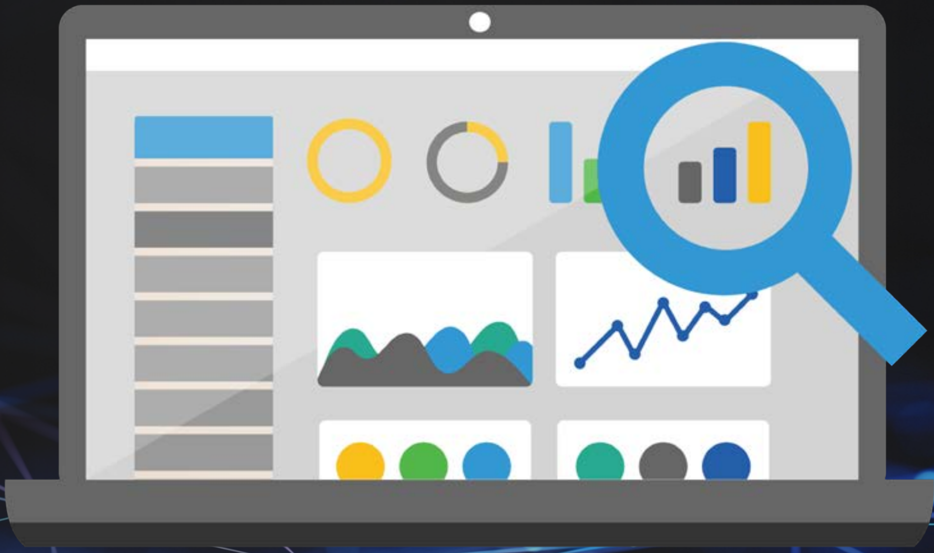
1

Monitor Services, Not Devices

You don’t own the network infrastructure. It belongs to the cloud provider, and they may not provide network management access to it. One of the advantages of the cloud is that you’re using infrastructure maintained by the provider. You should not have to monitor network devices. In some cases, network devices developed by the cloud provider (think Google and Amazon) may not support industry-standard interfaces like SNMP. The reduction in monitoring reduces tooling and staffing costs. The provider amortizes its internal monitoring costs over many customers. Cloud providers also design resilient infrastructure and have well-defined procedures for handling problems.

Service monitoring of your own application becomes your focus instead of network monitoring.

You need to make sure application services are available and functioning within the desired service level agreements (SLAs). For example, a web-based application might have a screen display SLA that sets the maximum display time of two seconds for a customer-requested screen. This would incorporate latency, server processing, database access (for applications that use a database tier), network bandwidth, potentially SAN storage, and the list goes on. The SLA might also specify the service availability, such as 99.99% available from the continental U.S. These two SLAs are the basis for service monitoring of this application.



It is useful to have several types of application-level tests, each measuring a separate tier of the application. For example, opening an application connection and downloading a static file indicates that the web tier is functioning. Another test could measure the web tier and its communication with an application tier by prompting the application to perform a simple function like returning a dynamically built page. A third test could exercise the web, application, and database tiers by retrieving some data from the database. You can also expand application tests to measure responsiveness when accessing data from larger database tables—where scaling problems can cause the whole system to slow to a crawl.

Leverage application-level tests to measure a separate tier of the application. Examples include:

- Opening an application connection and downloading a static file
- Prompting the application to perform a single function like returning a dynamically built page
- Retrieving some data from the database or accessing data from larger database tables

2

Monitor Within the Cloud & External to the Cloud

Monitoring within the cloud confirms any service outages in the cloud provider's infrastructure. This discerns the difference between the application not responding (an application problem) and the service not being reachable (an infrastructure problem). Monitor load balancers and firewalls separately from the servers to improve detection of problems within the front-end systems.

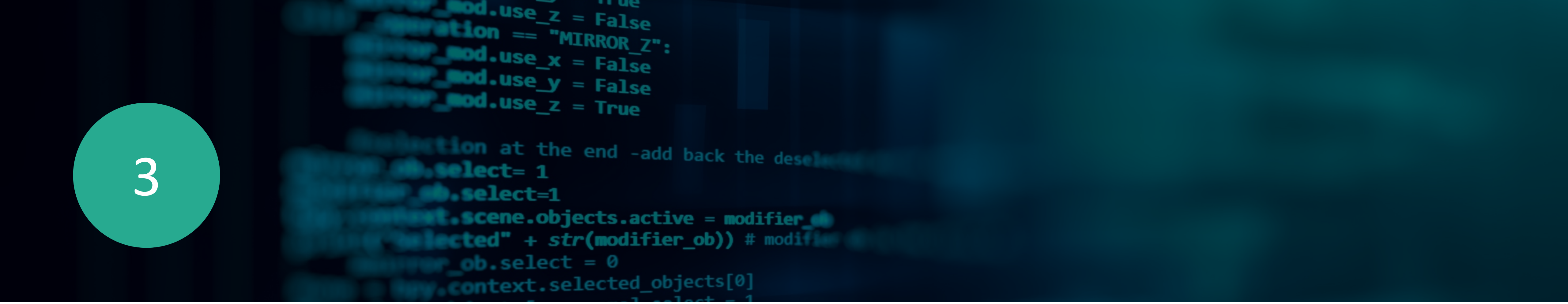
External monitoring detects WAN connectivity problems that impact application reachability and performance.

Work with the cloud provider to monitor several systems and determine if connectivity problems are due to the WAN interfaces or the internal infrastructure. Remember, your external network monitoring will have higher latency than internal monitoring. Correlating monitoring alerts and logs between internal and external can account for this difference.

Syslog allows your applications to asynchronously notify you of problems because no polling is required. Send syslog to both an internal log server and to your external monitoring system's location for optimum visibility. Then configure external logging servers to generate an alert when logging information from the cloud-based application server stops.

Add client response-time checks and use SLAs to trigger alerts if they get too large.

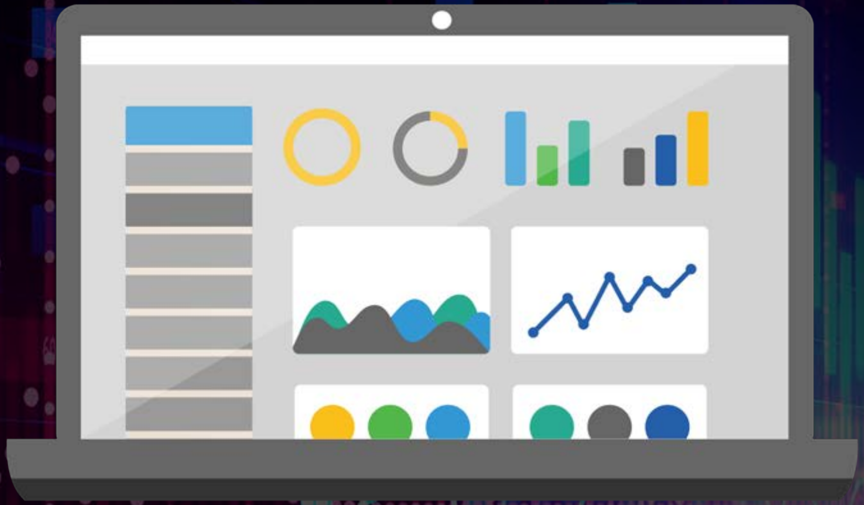
This approach requires access to the application servers (something that's not possible if you're using SaaS applications like SalesForce). In SaaS and PaaS examples you would also want to query their API and compare response times versus your expectations.



3

Work With DevOps to Enable Day 2 Operations

DevOps is a fast-growing approach to software development. This approach allows the developer to quickly build, test, prototype, and release new packages for your customers. The problem arises when the DevOps team needs to move onto a new challenge, and the resulting operation falls into operation's hands. The DevOps teams used very specific tools to understand how their application was performing in various environments. Rather than replacing the APM, NPM, and open source tools the team used, combine their data with your existing monitoring and management tools.



Create visualizations of how an application is working within the public cloud, your internal network, SaaS environments, and the underlying infrastructure to diagnose where problems are taking place. Discovery of all components, including the DevOps environment, is critical to maintain new products and an ever-changing environment.

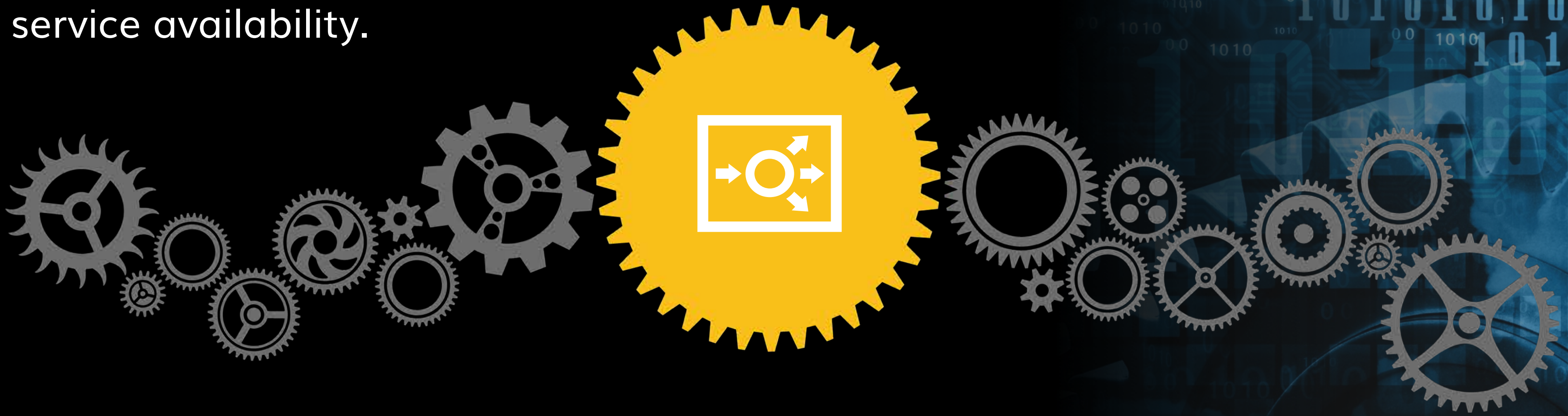
4

Embrace Automation

Automation is changing the ability of organizations to move in many different directions. Once you have a better view into the service and understand the critical components, you should automate the entire service management practice. An advantage to cloud computing is the ability to quickly add additional compute and storage resources to an application and distribute the application geographically. Automation is key to quickly responding to changes in application load. Don't forget to include the monitoring and management of applications as the underlying infrastructure changes. Ideally your tool will take care of these changes automatically.

You should also automate resolution of critical alarms, or at least the incident management process. Your monitoring platform should be able to automate incident creation and asset syncing between your tools of choice—cloud or on-premises. Cloud providers ability to auto-spin resources means that monitoring becomes more critical than ever. As you recognize the most common issues in your environment, you can script the spin-up and spin-down of cloud resources to cut costs and increase service availability.

Automation is key to quickly responding to changes in application load.



Avoid Cloud Vendor Lock

Don’t rely on the basic monitoring products provided by your cloud vendor. These products are designed to only look at the vendor’s environment and lack any ability to baseline usage across multiple locations (including the on-premises datacenter you use today). If you’re using a cloud provider’s network connection from your datacenter to their cloud (such as AWS Direct Connect), the monitoring tools may only show you the trunk status. They might not reflect the status of each vLAN or tell you anything about the traffic that traverses the connection.

Cloud vendor monitoring tools may have other limitations, such as:

- Lack of event management, correlation, or de-duplication of alarms
- Limited ability to configure monitoring dashboards
- No visibility into the OS or application performance

When monitoring a multi-cloud environment, you can conduct complete workload lifecycle management. In other words, you can compare performance of your application or IT service as it migrates to the cloud, and then between cloud providers or even regions/zones to determine where that workload is best suited. Be sure to compare performance and fault statistics of your environments in both production and test environments.

The surest way to avoid being stuck with one cloud vendor over another is to look at resources as what they are: resources. It shouldn't matter if the CPU data is being reported from in cloud, out of cloud, AWS, Azure, or SoftLayer.

Your ability to succeed in a cloud-based world relies on accurate data to drive successful outcomes.

Conclusion

- Traditional network management systems will not be very useful in a cloud computing environment where infrastructure is designed and implemented by the cloud provider. Service monitoring and using application-level tests provide visibility into the application's availability and performance metrics. Monitor the cloud infrastructure from both an internal and external perspective so that you can best identify the location of problems and facilitate troubleshooting.

- Cloud computing environments are dynamic so your tools need to be dynamic as well. Automation is required to shift monitoring to match the changes in compute systems. You must measure differences in each tier of the application as well as components that may encounter scaling problems, down to the infrastructure layer.

- Examine your existing suite of network management tools to find the right match. Many tools contain fundamental components but lack all the required automation functions and ability to natively support both multi- and hybrid cloud environments. Integration of some parts of your existing network management toolset may be needed to create the ideal solution and desired level of visibility.

Many tools contain fundamental components but lack all the required automation functions and ability to natively support both multi- and hybrid cloud environments.

ScienceLogic