# CONSIDERATIONS FOR MONITORING CLOUD-HOSTED APPS AND INFRASTRUCTURE

Moving infrastructure and apps to the cloud promises agility that better aligns resources to demand. There's also a promise for cost savings depending on what you're planning to do.

However, if you do not manage the consumption of resources—*especially in public clouds like Amazon Web Services (*AWS*) and Microsoft* Azure—you could end up paying a lot more than expected. Being able to monitor and plan for the cost of running infrastructure in the cloud vs on-premises helps you to make good business decisions.

The return on investment (ROI) for moving to the cloud vs on-premises is also tied to outcomes. If you move applications to the cloud and end up losing visibility and control, an outage could be very costly.

Two of the most important questions for IT operations to consider for monitoring include:

- What can and cannot be monitored in the cloud versus on-premises?
- Will you lose any visibility or control if you move your app to the cloud?

Here are several factors to consider when moving apps and infrastructure to the cloud.

## Monitoring cloud-hosted apps and infrastructure

When you consider monitoring hosted apps and infrastructure in the cloud, agents or probes can be installed in the same cloud environment. The agents or probes provide key performance metrics to the monitoring software. You'll get the typical infrastructure metrics for performance and availability including CPU, memory, and processor utilization as well as application performance metrics for transactions. You're instrumenting the same infrastructure as you would on-premises, just in a different environment.

Moving to the cloud should not be a problem in this case.

You should consider whether to run your monitoring software that takes in the agent or probed data hosted on-premises, or alongside the infrastructure and apps in the cloud. In some cases, you might want to consider a hybrid approach where you could have monitoring instances running in the cloud environment collecting and doing analytics and then storing the data back at your private data center. Organizations who do this prefer not to store data in public cloud environments for security reasons.

You'll want to make sure that the monitoring solution you choose can be architected with components that could span from your data center to private and public clouds. We have a large retail customer who has done exactly that. They were able to architect a solution that took advantage of scaling their infrastructure and monitoring in the public cloud while maintaining the security of their sensitive data on-premises.

## Legacy monitoring tools

If you have legacy monitoring tools on hardware appliances installed in your data center, then moving to the cloud is not going to work out so well for monitoring. If there's a software version of the appliance that you can install on a virtual machine (VM) in the cloud, then you can get the same or comparable monitoring coverage as you had on-premises.

## Newer architectures

It gets more complicated when the technology used in the cloud runs the apps differently than what happened on-premises. For example, running apps in Docker containers requires a different approach to instrumentation. You need to monitor the hosting environment of the Docker containers as well as what's going on inside the containers themselves

Here's a great example from one of our TrueSight Solution Engineers who shows us how to achieve the instrumentation for Java apps on Docker:

TrueSight App Visibility in Docker Containers

In a Docker environment, the ability to scale and have a dynamic monitoring solution that can keep up with the changing conditions is crucial.

To maintain the same visibility and control, you will want to make sure that you can monitor the newer technologies as well as instrument the app environment.

## Monitoring end-user experience

Getting real-time, end-user experience metrics from apps that run in the cloud is a relatively standard activity when you have control of the web servers that are serving the apps. You just need to add a snippet of code to the pages being served by the web server and then end-user experience monitoring is enabled. For example, a JavaScript snippet can be automatically injected to the headers of web pages at run time, or you can manually copy them to the pages. You can then monitor the real user experience of the web pages as well as the Ajax calls.

Most monitoring tools include this type of monitoring so that your cloud monitoring decision will not be impeded by this concern.

## What differentiates beyond front-end, end-user experience monitoring?

Going beyond the basics of end-user monitoring requires tying end-user experience monitoring to infrastructure monitoring and downstream analytics. This helps to troubleshoot an issue much faster than going to one console for user monitoring and another one for the infrastructure.

Cloud providers typically provide a level of monitoring natively. For example, for AWS, there are CloudWatch metrics available for the running infrastructure. These can also be integrated into your environment so that you have the infrastructure health from the cloud provider as well as deeper server diagnostics as necessary that you can instrument with agents.

Having a single console that shows this data in an application performance context will help reduce mean time to repair (MTTR) instead of having to flip through multiple sources of monitoring data. End-user experience monitoring for SaaS apps cannot be done as easily if you don't have the ability to instrument the web pages of the service provider of the app.

Downstream analytics using AIOps approaches for your complete IT environment can also have a huge impact on the value of your end-user monitoring in the context of a larger strategy of IT operations.

## Monitoring SaaS apps

You cannot monitor SaaS apps that you subscribe to very easily unless the provider gives you access to their environment.

You would be tied to whatever the SaaS provider offers you in terms of service levels. If you have degraded performance, you'd typically enter a service request to the provider. There might also be an SLA in place to mitigate any dissatisfaction with performance levels.

However, this does not stop you from doing some monitoring. You could do synthetic monitoring of the web pages that you're accessing, particularly if there are disputes between your service provider and your end-user experience. They might be using a less granular approach to monitoring. If they only report on hourly averages, the fact that several users have slow response times might not be captured if their average response time did not reach a critical threshold. You could pinpoint what times and pages were slow based on your synthetic monitoring.

*Note—if you're hosting the SaaS apps as a provider, you'd consider the same things you would consider*

*in the previously discussed examples for monitoring cloud-hosted apps and infrastructure. You'd have access to instrument all or parts of the environment yourself and provide the SaaS offering to your customers.*

## TrueSight for cloud-hosted apps and infrastructure

Whether you're monitoring from the back-end legacy components in your data center environments to the nimble Docker-hosted apps in the cloud, TrueSight can help with:

- End-user experience monitoring on-premise or in the cloud
- Software-based packet capture that can be deployed in the cloud
- Integrated metrics from AWS, Azure, or OpenStack cloud providers
- Synthetic monitoring for simulated end-user experience for SaaS apps

For more information, see
http://www.bmc.com/it-solutions/truesight.html