Use Privileged Identity to Manage Cloud Security

Privileged Identity has the ability to integrate and manage a public cloud environment in many different ways. Out of the box, there is support for Microsoft Azure, Amazon AWS, Rackspace Public Cloud, IBM SoftLayer, and more. If you are using one of the major providers, then you should check to see if we already have an out-of-the-box configuration for you. If you are using a Managed Service Provider (MSP) or other cloud service provider for which we have not built an out-of-the-box solution, there are still plenty of ways you can manage your resources there.

Just as you would with any remote or segmented network you wish to manage, you can drop a Zone Processor into the cloud network to run alongside the other resources you have there. This is identical to how you would manage a WAN connected second site or DMZ in your own infrastructure. The Zone Processor would connect over the WAN or Internet to the main Console system's database to coordinate policies and configurations. And it would run jobs in the cloud environment using authentication and identity that has the right access in that context. With this set up you would be able to manage credentials, discover resources, and generally do anything you would do in any other setting. We call this running *inside the cloud* - in the same context as the servers, databases, applications, and other infrastructure that run on your PaaS and IaaS systems.

You will also want to manage the new administrative credentials and rights on what we call the *outside-of-the-cloud*. These are credentials you would use to log onto, for example, the Amazon AWS portal to manage the servers and services you have running in their cloud. For other providers that will also be similar portal systems that offer the ability to create, manage, and ultimately retire the resources you run in that cloud, there are a few different scenarios here. If the cloud provider offers some form of API (e.g. a web service) that gives access to manage the credentials and other aspects of the system, then this is the best scenario. That is how we do it for all the out-of-the-box, major cloud providers we support. For other MSPs and cloud providers, you can formulate an approach using accounts stored locally in LDAP, AD or some other easily managed target that act as a reflection of the accounts you wish to manage in the cloud provider system, and then use our propagation to call the APIs (commonly via a script) to do the changes to the credentials in the cloud. You could also get fancy and use our SDK to build a custom account store for this cloud provider, but that would mean doing some custom programming. Most opt instead for the reflection option. If they have no API to call, then you can still manage the credentials, but now it would be through a much more complex scripting effort that would manipulate web pages as if you were clicking through them to do the credential management. While possible, this has many pitfalls, not the least of which is that any change to the web site of your provider will likely break the scripting. So using the API and reflection accounts that are stored locally is the best option if it is available.

Finally, many will want to also do some control of access to the portal of the cloud provider. You may wish to let some folks get to it without revealing the password for the administrative account, for example. This would employ our Application Launching capability. There are many examples of the Application Launcher managing web sites and portals (e.g. Twitter or Azure Billing). You can take one of these and modify it to your needs, changing the target site, the places it will need to click and fill in form data, and then Privileged Identity can feed the actual data into this when a user needs to launch it to gain access. Using this, you would have the partner do the management of the outside credentials you managed in the second step and have a way to grant access without exposing the credentials as a result.