

DevOps without Ops? Is it possible?

The All-Mighty Virtual Machine

2016 is the year of Zero-Client! Maybe not, but Virtual Machines or VMs are definitely here to stay... or are they? Technologies such as Docker and Amazon Web Services (AWS) Lambda are gaining more popularity and may change the way VMs are used in their current sense. We may be paying too much for our Virtual Machines.

Containers: Docker

By now, most of the IT world has been introduced to Docker and the benefits of containerization over straight virtualization using VMware, Citrix, Azure, or one of the cloud providers such as AWS or Google Cloud. The problem here is that containers still rely in part on virtual machines, and those VMs require Ops to manage them. Yes, you can run a container on bare metal (as Joyent has shown with their Triton offering) to reduce the virtual footprint, but to some capacity, Ops is still involved to manage the VM lifecycle.

Amazon Web Services: Lambda

Although AWS Lambda has been available for some time, it's interesting that containers and VMs are still being used in their current capacity. AWS Lambda spins up a system for use and runs code via APIs in response to a client's API methods (GET, PUT, POST, DELETE). Code is run in chunks of 100 ms, and the AWS EC2 instance is spun up and back down to reflect the code being run. That being said, can an entire app or enterprise backend run on AWS Lambda and avoid dealing with VMs? Lambda essentially takes the Ops out of DevOps, decreases provisioning time, increases efficiency, and allows costs to be almost negligible.

The case for Automation

I advocate for automation at every opportunity I have. If I have to perform a task more than once, there must be repeatable tasks, and those tasks can be codified and automated. What then happens when code is run in micro-instances via APIs in the AWS cloud? If secured properly, entire enterprise workloads can be entered by the client while the backend receives instructions to perform automated tasks. The server can then run those tasks while output is displayed on the client device.

There's still a need for virtual machines here. We just don't need to utilize them as often as we do to cause our VMs to be over-provisioned. The increasing use of AWS Lambda may change that, and Docker may very well be the vehicle to create portable and in-house API-based backends for cross-enterprise collaboration.