CONTAINER SPRAWL: WHAT IT IS & HOW TO AVOID IT

"Container sprawl is the new virtual machine sprawl" is quickly starting to sound like a cliche. But it's difficult to disagree. Inappropriate management of the number of containers used leads to:

- High costs
- Security concerns
- Potentially, a lack of governance

As more organizations are expanding their cloud presence, there is a growing need for containers. Effective container management allows your employees to quickly access applications on a variety of operating systems. And it's big business: container providers are expecting growth to nearly $500 billion by 2023. Organizations like Kubernetes and Docker are leading the cloud-based push—but do we truly understand how to effectively manage containers yet?

The main issue for these companies spending a significant part of their budget on containers is two-fold:

- Avoiding sprawl
- Ensuring that proper management can stop excessive spending and employee wastefully spinning up containers

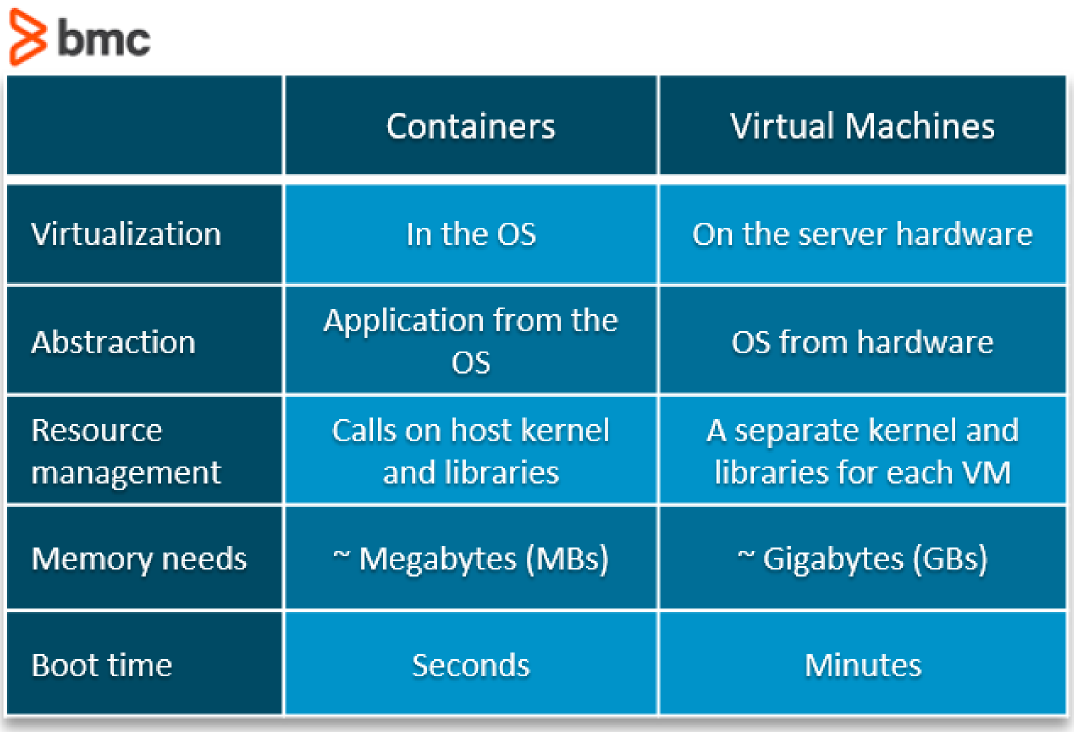
So, in this article, let's take a look at container sprawl. We'll start with a brief recap of containers and their benefits.

How containers work

A container is a piece of software that contains an application and all associated code in one place. This allows the container and its associated application to be "lifted out" of its native OS and placed somewhere new.

Containers offer a smaller-scale type of virtualization, but they're different from virtual machines (VMs) While using a virtual machine creates a new "machine" on the hardware, containers only create applications in the OS.

(Read our full explainer on VMs vs containers.)



	Containers	Virtual Machines
Virtualization	In the OS	On the server hardware
Abstraction	Application from the OS	OS from hardware
Resource management	Calls on host kernel and libraries	A separate kernel and libraries for each VM
Memory needs	~ Megabytes (MBs)	~ Gigabytes (GBs)
Boot time	Seconds	Minutes

Why use containers?

Container use is incredibly convenient, especially when you need resources in a distributed workplace. Here are three of the top reasons for using containers in your operations:

- **Compatibility issues**. If a company is using an older version of a programming language but knows that a newer version will run it, the development team can run it in a virtual environment without having to install/uninstall the necessary content.
- **Cost-cutting**. Instead of splashing out on any physical server or end-point technologies, applications can be packaged to be used anywhere.
- **Security**. Securing your applications into one place isolates any vulnerable aspects of necessary services. Instead of these applications exposing

Growth of cloud-native apps & containers

Cloud-native applications (that is, apps entirely hosted and operated in the cloud) are becoming more popular, especially during the global pandemic. Among the greatest benefactors are your IT organization's DevOps teams: with containers, your developers can quickly build full stacks to test new code in a faux infrastructure that mimics the organization.

The best part? All this can happen from anywhere.

Cloud-native apps are logical for deploying applications to a distributed working environment. Containers are also allowing businesses to host their own full-stack development platforms from anywhere in the world. Physical access to servers becomes less and less a problem.

Now, let's turn to the problem with too many containers: sprawl.

What is container sprawl?

Container sprawl is the tendency to run up an unnecessary number of containers. Although there are differences between running a cloud-native container system and a physical data center, the biggest problem is effectively the same:

Cost.

Spinning up a lot of containers causes business problems, typically in cloud computing fees and management issues that result in inefficiencies. To be clear: even though creating containers is significantly more convenient than setting up new physical servers, the cost implications can quickly grow out of control.

Container sprawl vs VM sprawl

The convenience of opening new images and running containers means that we are reaching a critical level. Kubernetes, in particular, is guilty of creating a new image and a new container with each change made. Unless you've got the team skills and effective cluster management tools, however, operations quickly spiral out of control.

Because a variety of environments can be quickly spun up without affecting the overall performance of a network, a lack of central governance is increasingly problematic. Managing clusters in a virtual workplace is difficult because resources are often mismanaged.

But the main takeaway is that container sprawl is much easier to manage. Sensible VM policies can stop your team from indulging in container sprawl too.

(Understand how containers & Kubernetes work together.)

How to manage containers

But security operations and management teams don't have to think about abandoning what's useful about containers. Instead, the use of clusters will cut down on management complexity and sprawl that is potentially harmful to the organization's security. And the right management approach will also help cut costs.

Here are three suggestions for improving your team's chances in the fight against container sprawl.

Combine applications

Sometimes, applications should be combined. Creating clusters of what you might turn into three or four containers results in:

- Fewer individual images being loaded

- Overall decreased costs

Overzealous container fans may want to turn every single process into a container—but beware: combining containers can actually improve functionality and save costs at the same time. Central governance and change management of containers will lead to lower costs thanks to combined container functionality.

You don't need to place every single application in its own image if you only ever use them in conjunction with other ones!

Combine VMs & containers

At the risk of inviting VM sprawl, using virtual machines to create containers can cut down on the overall costs of out-of-control container creation.

Host a VM in the cloud and then add a container image to that machine. For example, hosting Kubernetes or Docker in an infrastructure as a service (IaaS), such as GCP or AWS, can lead to lower costs than simply subscribing to a container service. You may sacrifice pure performance, but the cost-benefit analysis may pay off.

Abandon servers

Abandon servers altogether?! Slow down! We're not abandoning them entirely. In fact, we're not abandoning anything—this is just moving more applications and components to the cloud.

The main benefit of this approach is slowing the costs of using containers. If your team only uses an app once a month, you will only be charged for that time. Moving infrequently used applications and processes into the cloud on a cost-per-use basis will slow the financial bite of container sprawl.

Using containers without inviting container sprawl

Containers are becoming more necessary in day-to-day operations, so managing container sprawl is now, unfortunately, a necessary evil for IT management. Implementing effective container and virtual machine policies is the best way to make the most of the functionality that containers bring without letting costs spin out of control.

Think about the suggestions above and how they could aid your company. Is it time to start creating containers or is it time to move all applications into the cloud? Either way, avoiding container sprawl is key for effective governance, slowing expenditure, and effectively managing this incredibly useful functionality.

Related reading

- BMC DevOps Blog
- Container Management Platforms: Which Are Most Popular?
- Docker Security: 14 Best Practices for Securing Docker Containers
- Containers Aren't Always the Solution
- Containers as a Service (CaaS) Explained
- The State of Containers Today: A Report Summary