DOCKER MANAGEMENT TIPS

Part of our series on Docker. Check out How to Introduce Docker Containers in Enterprise, Docker Production Deployment Security Considerations, and Docker 101

Digital technology is helping consumers get what they want, when they want it, at a competitive price. It's also transforming businesses and industries. Digital music flattened CDs. Online travel is sending agents on a permanent vacation. Other digital business models from Uber and Lyft are driving the taxi industry crazy. Companies of all shapes and sizes need to innovate quickly or get left behind.

In an environment where speed is critical and most technological innovations must go through developers, IT must help developers to become more productive. How? By removing obstacles and giving them fast access to resources, minimizing process overhead, and letting them use the tools they prefer. That's where Docker comes in – an open platform for developing, shipping, and running applications quickly and easily.

With Docker you can separate applications from your infrastructure and treat the infrastructure like a managed application. A Docker container, which includes instances of images, can be deployed in seconds instead of minutes. The container holds everything needed for an application to run. However, before you move those "Dockerized" applications into production you need tools that provide security, governance, automation, and orchestration of containers to manage them successfully across the complete lifecycle.

Here are four considerations to help you be prepared when it's time to implement Docker in your enterprise.

1. **Want to lead? Then focus on speed – but be careful**
   Docker emerged about two years ago and it's one of the fastest-growing open source technologies. Because the technology is new, it's mostly being used in test and development. However, it's only a matter of time before Docker becomes more common in production. When that happens, be prepared to manage it effectively; to monitor Docker applications, incorporate Docker containers into your release management process, identify and patch security vulnerabilities, and control the sprawl of containers.

2. **Remediate security vulnerabilities**
   Docker containers often have short lives that are enabled because they are easy to create and quick to deploy. In testing environments where the applications are stood up, tested, and removed, Docker works really well. However, in production environments, these containers could have longer lives and may need to be patched or reconfigured to remediate security vulnerabilities.

With Docker, the operating system resides on the Docker host and not inside the containers. So, depending on the type of vulnerability, you might need to patch the host or an individual container. You may even decide that it is easier to fix the configuration in the Docker image, create a new container and redeploy the container rather than patch it live. Each of these approaches to remediating vulnerabilities has its own set of pros and cons. Regardless of the method chosen, you need the right tools to help you identify these vulnerabilities, remediate the changes, and keep track of the changes that are made.

3. **Monitor Containers and Manage Container Sprawl:**
   With Docker, it's very easy to create an application and so developers may be more likely to create a new one instead of looking for an existing one. This can lead to container sprawl. Management systems technology, fortunately, can be used to identify what's in the catalog and enable developers to select the images they want to use rather than having to create new ones. That's where having a self-service catalog for requesting Docker hosts and Docker clusters can be most helpful.

Docker applications can run on a fraction of a server. There is the potential to have far more containers on a single server than you'd have with a virtual infrastructure. So, how do you manage all of the Docker containers and applications that are running them? The key is to recognize that as organizations transition to Docker they will have software that runs on physical servers and virtual machines in physical environments. They need a tool that will help them keep Docker as part of an overall cloud and contain sprawl to reduce infrastructure costs. In addition, as new builds for Docker images are produced they can be made available for testing complex, multi-container applications using a cloud management platform.

4. **Focus on release management for Docker containers:**
   Incorporating Docker containers as part of a complete release lifecycle management process can ensure successful releases. The Docker container may include an application server, library, application code and other components to run an application. With release lifecycle management technology, you can handle, deploy, and include these elements in the release process and move quickly through development and testing and into production. This technology and best-practice processes result in more successful release deployments that can minimize downtime during application releases.

Looking ahead

If your organization hasn't moved to Docker yet, it's likely to move soon. Docker drives density, which increases complexity in the infrastructure. With the right solutions that integrate with and manage Docker, you can reap the benefits of speed and agility in developing and releasing applications – the benefits of innovation. Plus, you can keep Docker in control, manage vulnerabilities, and monitor applications that Docker containerizes.

For more information, visit:

- BMC SecOps
- BMC TrueSight