HOW CONTAINERS FIT IN A DEVOPS DELIVERY PIPELINE



DevOps practices are being embraced through the tech world as a better system for reacting to the constantly shifting demands of the marketplaces of today. Answering these demands means developing and delivering software solutions that are both timely and reliable. In search of providing speedy software solutions to businesses around the world, DevOps practitioners leveraged the idea of continuous delivery pipelines for containerized applications.

What is a Containerized Application?

The whole point of DevOps is to speed up production while simultaneously increasing the quality of the product. Achieving both of these goals simultaneously may seem like a pipe dream, but techniques like containerization help increase the stability and reliability of applications by expanding upon virtualization approaches to make application management easier. Containerization virtualizes an operating system to allow applications to be distributed across a single host without needing separate virtual machines. This is accomplished by providing apps with access to an operating system kernel (the core module of an operating system).

Utilizing a kernel in this way allows for the creation of isolated containers that don't require their own operating system. Thus, applications are containerized into smaller and more resource-efficient chunks that can run on any machine. The portability of containerized applications eliminates dependencies and allows you to host more applications on a single server thanks to the shared kernel. These apps will boot and run faster for the end user while also requiring less resources from the server. Containerization also directly supports the entire DevOps culture by providing the capacity for scalability devs want and allowing operations to achieve maximal efficiency.

Containerization also directly supports continuous delivery pipeline infrastructure by fitting perfectly

into the processes.

What is a Continuous Delivery Pipeline?

Continuous delivery has become a mandatory process in the technology sectors of today's business world. The main selling point of continuous delivery is the ability to deploy software more frequently while maintaining stability and reliability of the systems. The continuous delivery pipeline is a stack of tools used to ensure software deployments are as pain free and low risk as possible. Automation is employed whenever possible to make the deployment process faster, less work intensive, and as error free as possible.

Automation provides faster time to market speeds by taking tasks that were once performed regularly by humans and turning them into processes that are triggered at regular intervals or after specific conditions are met. This creates a development environment where devs can focus on improving functionality, usability, and performance with the security of knowing changes they make will take nearly immediate effect on the service upon which they're working.

In addition to automation, DevOps pros use various tricks and techniques that help bolster production speed and reliability. One especially useful technique

What is Blue/Green Deployment?

Blue/green deployment utilizes two separate application environments that take turns functioning as the active development environment and the live service environment. For illustration purposes think of the blue version as the instance of the application that is currently being used by customers while the green version is in active production. Once updates have been made and tested to satisfactory levels on the green instance, traffic can be switched over to green while blue is taken offline. Development can then take place on the now offline blue instance of the application and the process endlessly swaps between the two versions.

This process allows you to deploy updates without interrupting service for users by transitioning seamlessly to updated versions that have already been tested in the same environment. Additionally, blue/green deployment provides you with a simple way of rolling back changes to the most recent version if there are unexpected issues once the newly updated instance is completely rolled out. This technique reduces risk of deployment and drastically improves the user experience by reducing service downtime during update transitions. It can also improve team morale by removing the need for launching updates at bizarre hours of the night (thereby requiring IT to be on call throughout the night) in an attempt to avoid inconveniencing users during prime traffic periods.

Combining containerization technology with continuous delivery pipeline tools and strategies provides you with the best of both worlds for rapid, efficient, and stable application development and deployment.

How Does it all Fit Together?

The DevOps chain has a lot of moving parts when it comes to the various tools and processes involved at each link. Containerization cuts down on some of this bulk and provides lighter and faster applications that require less resources from servers while also reducing the variance of dependencies. Neatly packing everything an app needs to execute into a single container achieves

the goals of operations while bolstering the capacity for developers to meet their own goals. Containers are cheaper and faster to run while still providing a powerful capacity for versioning.

Continuous delivery emphasizes the automation of the processes that function best when optimized while still acknowledging the value of manual testing and deployment. Integrating automation into the deployment process provides the best of both worlds where delivery is handled quickly without compromising the quality of deliverables. Testers provide their unique insights into the system while automation enhances their focus onto the areas that require a deft touch. Automation isn't the answer to every problem, but continuous delivery pipelines are built to identify where it works best while providing tools and strategies for implementation.

Strategies like blue/green deployment that speed up development time while simultaneously increasing build and deployment stability are another piece of the DevOps package designed to optimize IT performance. Developing containerized applications and funneling them through your continuous delivery pipeline will lead to faster and more successful deployments. Creating smaller and more efficient applications while utilizing continuous delivery tools embraces the essence of DevOps for maximizing the potential of your teams.

DevOps: Solutions for You

If DevOps sounds like a good fit for your organization's needs but you want to make sure you get it right the first time, BMC is the IT solution partner you need. Read more about how automation and DevOps systems can help increase the rate at which you deploy products with BMC's free eBook: *Automate Cloud and DevOps Initiatives*.

To find out more on how to get started using containerization methods for your enterprise, check out BMC's 3 Steps to Introduce Docker Containers blog post. BMC expert consultants are available to work with you to bring their knowledge and expertise to your organization. BMC also provides custom tailored Implementation Services for your organization to tackle the unique challenges you face. When partnering with BMC, you get:

- **Faster service delivery**: Agile releases that keep up with rapid demand
- **Visibility across data**: Ensure compliance and data accuracy
- **Cost-effective service**: Increased productivity and performance
- **Experienced DevOps professionals**: Equip you with the tools you need for success
- **Conversion or upgrade**: Seamless modernization or total replacement
- ***All tailored for the specific needs of your organization***.

Download or view the Solution Implementation Overview online to learn more about how BMC Consulting Services can help you. Learn more about how DevOps and Application Deployment best practices can enable your teams to create better software faster than ever before. Then contact the experts at BMC to learn more about how to incorporate containerized continuous delivery pipeline systems into your DevOps practices for enhanced building, testing, and deployment success.