

VMware Tanzu™ Build Service™

A new way to build containers for Kubernetes

A KUBERNETES-NATIVE CONTAINER BUILD AND MANAGEMENT SYSTEM

- Deploy Build Service to any Kubernetes cluster
- Build containers that run in any Kubernetes cluster
- Rebuild containers automatically upon code, dependency, or OS updates
- Improve security and compliance posture with centralized control of container updates and patches
- Increase developer efficiency and reduce CI sprawl
- Build and maintain containers in air-gapped environments
- Build and manage Windows containers for .NET workloads
- Push containers to any OCI-compliant container registry

The rise of cloud-native computing and software development, accelerated by the explosive growth and ubiquitous adoption of Kubernetes, has driven a complete re-imagining of the way that applications are built, deployed, and maintained. Applications went from components working together in monolithic virtual machines to applications consisting of individual micro-services orchestrated by Kubernetes and working together over a network. This new world brought developers a level of agility they could only dream of a decade ago, but it also introduced new challenges for larger organizations as the number of containers in use quickly scaled into the thousands or tens of thousands.

As Kubernetes scales, containers number in the thousands

Today, most containers are constructed using Docker. Docker uses an image definition called a Dockerfile to specify a base operating image upon which it will sequentially install an application and its dependencies in layers. Using Dockerfiles often adds a steep learning curve in front of Kubernetes adoption for developers, who must learn to navigate the complexity of specifying multiple base images for multi-stage Docker builds before they can ever deploy a single application.

Furthermore, when Dockerfiles are used to maintain containers over time, each layer is pulled from external sources when the image is updated. Even if the only needed change is to apply a patch to one operating system library or one runtime dependency, the entire container will be rebuilt with a complete set of newly downloaded libraries and dependencies. Therein lies the challenge: as Kubernetes usage scales in an organization, thousands of containers are built and rebuilt on different schedules, resulting in inconsistent patch levels for dependencies in each one and no way of verifying what's in each container. For IT Operations teams, ensuring that container images are kept in compliance and free of vulnerabilities becomes an impossible task.

The most common solutions for scaling container builds on Kubernetes today involve deploying dedicated CI tooling to maintain "blessed" base container images, combined with maintaining central repositories of runtime and framework dependencies and libraries. However, both these activities create onerous requirements on developers and operators to manage and coordinate these resources. Security is only had at the expense of efficiency.

Clearly, a new way of building and managing containers that scales alongside Kubernetes adoption is needed.

Take a Kubernetes-native approach to building containers

VMware Tanzu™ Build Service™ programs the Kubernetes API to automatically build and continuously maintain containerized applications at enterprise scale.

It gives developers a simple workflow for constructing and updating Open Container Initiative (OCI)-compliant containers for Kubernetes, while delivering the security benefits of automated dependency and operating system updates to operators.

BUILD AND MAINTAIN CONTAINERS FOR THE MOST POPULAR RUNTIMES

Proprietary Buildpacks for VMware

Tanzu™ included:

- Java/Spring
- Node.js
- Golang
- .NET Core
- PHP
- NGINX

Compatible with open source Paketo Buildpacks:

- Ruby
- Python

TANZU BUILD SERVICE COMPONENTS

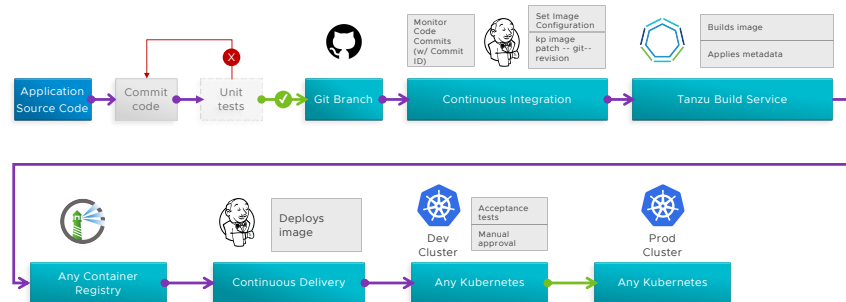
- kpack Kubernetes-native container configuration system
- kp command line interface for controlling kpack builds and updates
- Buildpacks for VMware Tanzu – runtimes and dependencies
- Clusterstack resource: a Kubernetes resource to manage operating system components
- Clusterstore resource: A Kubernetes resource to manage buildpacks

How it works

Tanzu Build Service works with existing Continuous Integration/Continuous Delivery (CI/CD) pipelines. It replaces the existing container build process for most teams by automating the creation and updating of container images. All a developer has to do is execute a simple command in the included kp command line interface (CLI) to tell Tanzu Build Service where the application code is stored. Tanzu Build Service then creates an image configuration that references centrally managed build configurations specifying which buildpacks (for language runtimes and dependencies) and stack (for a base operating system) will be used to construct the container. The system builds the container, including a manifest of its contents written into a digest for visibility and tracking, and stores it in any OCI-compliant registry. Finally, Build Service continuously monitors the image configuration so that it will detect any change to a container's source code or build configurations.

Any application, buildpack, or operating system update triggers a rebuild, which is pushed to a registry and can then be deployed by a CD system. Tanzu Build Service builds containers modularly, enabling one layer to be updated without affecting the others. A dependency or operating system change that affects multiple containers can be applied simultaneously to all of them without degrading their functionality.

The result is that developers who were previously maintaining complicated Dockerfiles in addition to their code can now focus only on their code. Operators who were formerly struggling to track which dependencies and OS patch levels existed in thousands of unique containers are now able to push updates to **all** of them simultaneously, without inconveniencing developers.



A look at the open source foundation of Tanzu Build Service

Tanzu Build Service brings multiple open source technologies together into one automated service for enterprises adopting Kubernetes. There are four key elements of Tanzu Build Service that add enterprise value onto its open source foundation:

Buildpacks for VMware Tanzu

Based on open source Paketo Cloud Native Buildpacks, Buildpacks for VMware Tanzu are container-building binaries that build Open Container Initiative (OCI)-compliant container images from source code. They automatically construct container images with the latest patched versions of dependencies necessary for the code to run. Changes in dependency versions are propagated to every container that was constructed using that particular buildpack. Buildpacks for VMware Tanzu include enterprise capabilities such as support for multiple Linux distributions and Windows, as well as building containers in air-gapped environments.

kpack

kpack is a Kubernetes-native container image build service. Developers use kpack to specify an image configuration which is used to build images and schedule rebuilds using Buildpacks for VMware Tanzu. kpack extends the Kubernetes API to monitor the other Tanzu Build Service resources for updates, and triggers container rebuilds whenever changes are detected. The end result: automated updates to containers and their dependencies.

"The first time I saw Build Service, I saw what we were doing in a much better and more elegant solution. Building and maintaining a baseline and using it to provision a whole bunch of different flavors of apps but still being able to certify that I know exactly what version and what libraries are deployed in what image sounds great. Building a docker container is easy, but maintaining a trail and making it accessible to developers is really hard."

-SENIOR SYSTEM ADMINISTRATOR AT A
LARGE US FINANCIAL EXCHANGE

Operator control interface

Operators are able to use Tanzu Build Service to maintain fine-grained control of which software configurations are available to developers or teams in an organization. This control is accomplished through resources that enable operators to create policies specifying which workloads, settings, and runtimes are accessible to a given set of developers. Operators are also able to use Kubernetes-native role based access control (RBAC) settings, simplifying access management. Operators can use Tanzu Build Service through the included kp CLI, or through standard methods of accessing Kubernetes such as Kubectl.

Enterprise support from the experts at VMware

Enterprise support is provided for each of the Tanzu Build Service components, enabling teams to feel confident that VMware has their backs if something goes wrong. Buildpacks for VMware Tanzu™ are backed by a dedicated engineering team that provides timely updates. Application runtime support is delivered through support for Buildpacks for VMware Tanzu.

Build and manage containers with a scalable system

Simplify your current code-to-container workflows: Let developers and operators get out of the business of maintaining "blessed" container images either manually or through expensive dedicated CI/CD tooling, none of which adds any business value. Tanzu Build Service builds OCI-compliant container images that work on any Kubernetes platform, and it keeps those images patched and up-to-date without error-prone human intervention.

Bake compliance into every custom application: Enables Operators to enforce policies specifying that security settings be configured by default for applications running in production environments. With Tanzu Build Service, policy is defined centrally and implemented automatically whenever a container is built or updated from source.

Automate vulnerability mitigation: Keep your applications patched and free of vulnerabilities by automating dependency updates across your entire IT organization. Rather than manually tracking which versions of dependencies exist across hundreds or thousands of individually packaged containers, know that Tanzu Build Service not only tracks when a CVE is patch is available, but also applies the patch to that vulnerability automatically. Comprehensive metadata is built into every container, bringing a new level of transparency and auditability to your custom software.

Centrally manage app dependencies: Create and enforce policies for runtime and dependency usage by application teams. Enable stronger compliance with industry regulations without slowing developers down. Tanzu Build Service allows developers to focus on writing code and generating business value, while operators can verify that dependencies are up-to-date with the latest patches.

Get started with Tanzu Build Service

Download an evaluation copy of Tanzu Build Service and start building containers today:

- Create a Tanzu Network account at <https://network.pivotal.io>
- Navigate to the Tanzu Build Service tile
- Follow the instructions to deploy Tanzu Build Service onto Kubernetes

Visit <https://tanzu.vmware.com/build-service> for more information and a get-started video.

