

# Cisco CloudCenter Solution: Optimize DevOps and Continuous Delivery

Three design patterns for automated application  
stack deployment

## Executive summary

Successful DevOps and continuous-delivery strategies change the economics of software development. An integrated and automated tool chain helps accelerate deployment and improve quality. But enterprise IT also needs enterprise-class features that provide governance oversight, with centralized visibility and control across applications, clouds, and users.

The Cisco CloudCenter™ solution is a hybrid-cloud management platform that provides the technology to support DevOps and continuous-delivery strategies. It automates deployment and management of builds and underlying environments consistently and predictably across lifecycle phases that may use different data center or cloud infrastructure. And it supports a variety of starting points with three design patterns: self-service application stack deployment, integrated API-based tool chain, and hybrid-cloud pipeline automation.

## Contents

### Environment matters

### Automated application stack deployment

### Cisco CloudCenter platform in your tool chain

### Design Pattern 1: Push-button application stack deployment

### Design pattern 2: Integrated tool-chain deployment using the API

### Design pattern 3: Hybrid-cloud pipeline

### Benefits

### Conclusion

### For more information

## Environment matters

The first principle of the Manifesto for agile software development is “Our highest priority is to satisfy the customer through early and continuous delivery of valuable software.”

Creating a repeatable and reliable delivery process for agile software development typically involves building a culture of collaboration between development and operations groups. It also involves implementing technologies and processes that enable a continuous-delivery pipeline. Regardless of your target delivery schedule (daily, weekly, or monthly), successful DevOps and continuous-delivery strategies change the economics of software development by accelerating release cycles. This acceleration enables more rapid validation of business ideas, reduces defect rates, and improves service restoration times when failures occur.

A continuous-delivery pipeline is powered by a partially or mostly automated software tool chain that takes delivered code and builds it, tests it, and deploys it. Ideally it facilitates collaboration between the various groups involved while providing visibility into the flow of code changes in the system.

A typical integrated tool chain combines tools into two major groups:

- **Continuous integration:** Front-end tools include feature tracking, development, source-code management, build automation, and artifact management tools that work together to **create builds that are ready for deployment.**
- **Continuous deployment:** Back-end tools include environment management, release management, and Quality-Assurance (QA) testing tools that work together to **verify that the latest code is deployed and that it performs as intended.**

The way you deploy environments to support the latest build determines which of two possible outcomes occurs:

- Is your environment deployment a strong predictor of top performance?

“One of the best predictors of DevOps performance is that IT operations can make available environments on demand to development and test, so that they can build and test the application in an environment that is synchronized with production.” –Gene Kim, co-author, the Phoenix Project

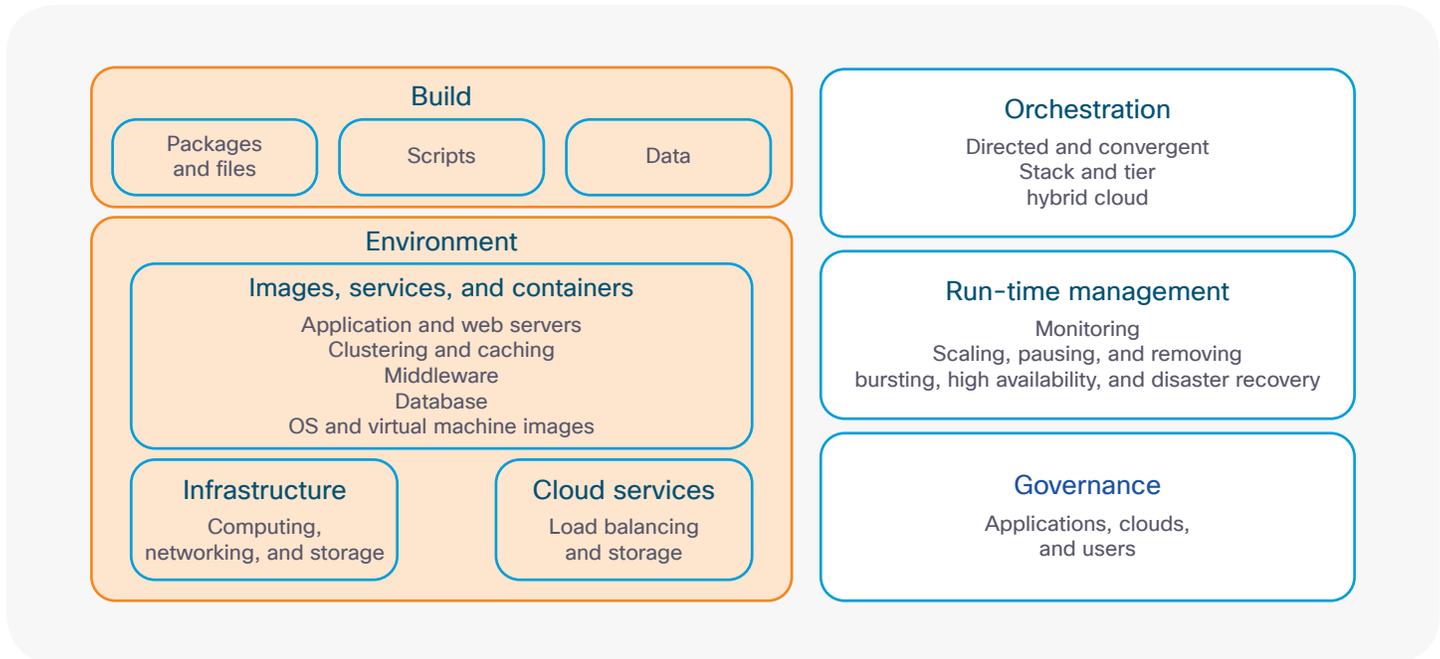
- Or does your environment deployment reflect the number-one killer of agility?

“As code moves from one environment to the next, software often breaks because of its interaction with different configurations within each environment. I have seen teams waste days and even weeks fixing bugs that are due to environmental issues and are not due to errors within the code. I call this ‘environment hell!’” –Mike Kavis, vice president and principal architect for Cloud Technology Partners

## Automated application stack deployment

Consistent and repeatable self-service or API-based deployment of the full application stack—including the latest build and its supporting environment—is critical to the success of any DevOps or continuous-delivery initiative.

Figure 1. Application stack includes build and environment and other requirements



As shown in Figure 1, application-stack deployment includes both the build and the environment:

- **Build:** The build includes the packages and files, scripts, and, optionally, data needed to deploy working code.
- **Environment:** The environment starts with infrastructure resources (computing, network, and storage) or, network services such as load balancing or firewall rules. It includes deployment of OS and virtual machine images, databases, middleware, clustering and caching services, and application and web servers. Code changes or builds can and may be deployed to multiple tiers in the stack.

Application-stack deployment also requires orchestration, management, and governance:

- **Orchestration:** Two types of orchestration can be used. Directed orchestration guides steps in a particular order and may include deployment of

multiple tiers, connection of tiers, and verification of service between tiers. Convergent orchestration converges the state of a tier with a target configuration and includes configuration management tools (such as Chef, Puppet, Salt). Typical build and environment orchestration includes both types of orchestration.

- **Ongoing management:** Ongoing management includes basic management functions that reduce the maintenance needs of preproduction deployments and that provide automated termination and release of infrastructure resources. It also includes functions that use automated deployment capabilities to augment ongoing production management with scaling, bursting, and high-availability and disaster-recovery features.
- **Governance:** Governance rules guide deployment and ongoing management of applications, clouds, and users to help control costs and ensure that basic boundaries are in place to support automated or self-service on-demand stack deployment.

Automated stack deployment reduces the time developers spend building and configuring environments. It also simplifies troubleshooting and reduces sources of errors due to manual process. In addition, automation helps reduce variation from instance to instance across the Software Development LifeCycle (SDLC) and across data center, private cloud, and public cloud environments.

Several factors affect the capability to automate stack deployment. These include:

- **Composite topologies:** Many traditional enterprise application stacks now include five or more tiers to meet requirements for features such as caching, load balancing, tier-specific scaling, and inter-tier queuing. Some tiers, such as load-balancing and relational database services, may be delivered as services through a cloud provider or in a software defined data center. In addition, application architectures are moving away from traditional multitier stacks to more loosely coupled, distributed service architectures that can be deployed using a container technology such as Docker, increasing complexity.
- **Optional tear-down function:** Some continuous-delivery pipeline steps may not require the environment to be kept in place after a build has been deployed and tested. Other steps may require a new build to be deployed in a long-lived environment. Consequently, an automated stack tool chain must support both deployment of a new build in a new stack with automated tear down and deployment of the latest build in an existing environment without tear down.

## Cisco CloudCenter platform in your tool chain

What you need are integrated tools and processes that allow developers to move quickly, but within well-defined boundaries. That makes them efficient code developers. But that keeps them from being distracted by infrastructure. Picture a sports car accelerating between guardrails on a banked interstate onramp as opposed to a dune buggy accelerating in an open desert.

You need a solution that gives developers on-demand access to development and test environments and that automates stack deployment across SDLC phases and across different data center and cloud environments. But you also need enterprise-class features that put guardrails on the onramps: for example, deployment policies, use and cost controls, and centralized visibility and control across applications, clouds, and users. You need precise Role-Based Access Control (RBAC) to help ensure measured progression through the SDLC phases. And if your development team is part of a multi business unit organization, you may need multitenant management of cloud accounts and financial plans to control costs and enable effective usage reporting.

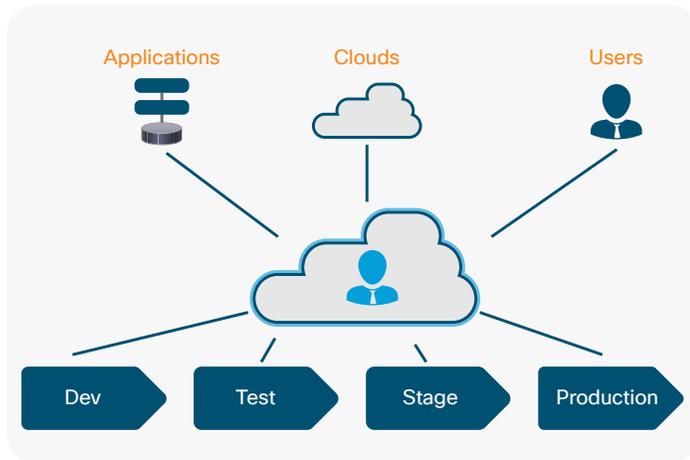
The Cisco CloudCenter platform provides an enterprise ready DevOps and Continuous Delivery solution. The Cisco CloudCenter solution is a hybrid-cloud management platform that securely provisions infrastructure resources and deploys application components and data to more than 19 data center and private and public cloud environments.

At the solution's core is a cloud-independent, multitenant orchestration and management engine. The engine provides both directed and convergent orchestration of simple or complex and new or existing applications. The Cisco CloudCenter solution coordinates the implementation of infrastructure resources as well as the deployment and configuration of the bottom-to-top application stack and security settings.

The solution uniquely combines a cloud-independent application profile that defines deployment and management requirements for the application stack, with a cloud-specific orchestrator that deploys the full application stack using the best practices for each environment.

The Cisco CloudCenter solution delivers features that make developers' jobs easier, and it offers visibility and control that makes the jobs of IT operations staff easier too. A scalable, multitenant orchestrator with mature APIs, the solution provides a powerful foundation for the back end of a contiguous delivery tool chain.

Figure 2. Deploying and managing the build and environment across the software development lifecycle



As Figure 2 shows, the Cisco CloudCenter solution provides a single platform for deploying and managing multiple applications with multiple users across multiple clouds through all phases of the software development lifecycle.

- **Multiple applications:** Cisco CloudCenter supports automated deployment of new and existing applications for a range of application types and technologies, including simple batch and traditional n-tier enterprise applications and more complex distributed microservice architectures. It includes a library of ready-to-use and custom services, images, and containers that work across cloud environments and can easily be tailored to meet specific needs.
- **Multiple users:** Cisco CloudCenter supports a wide range of DevOps roles, including developers, QA personnel, DevOps engineers, and production administrators. It supports role-based authentication that tailors work streams to each user role. Developers and QA personnel have access to self-service features for on-demand provisioning of environments and the capability to promote applications to the next SDLC stage. Administrators can set up shared access to deployment environments for each lifecycle phase.
- **Multiple clouds:** Cisco CloudCenter provides centralized management of accounts and financial plans across all data center and private cloud and public cloud service providers. IT maintains control with

detailed metering, notifications, usage plans, aggregate accounts, and billing resources. Administrators can set policies that control sprawl and suspend or eliminate systems that are not in use. Cisco CloudCenter provides a management and orchestration tier for all deployments, automating a hybrid-cloud SDLC pipeline that improves efficiency and reduces risk.

The solution accommodates multiple starting points and supports three design patterns for DevOps and continuous delivery, with each pattern enhancing the capabilities of the previous one. These patterns are described in the following sections.

## Design pattern 1: Push-button application stack deployment

In design pattern 1, developers are given a “button” that delivers instant on-demand access to the application stack they need when they need it. They don’t have to open a help-desk ticket, and they don’t have to manually install and configure tools and middleware on top of an automatically provisioned data center or cloud virtual machine resources.

With the Cisco CloudCenter solution, developers can think in terms of working with a fully configured environment as a service, or deploy as a service, rather than of consuming a virtual machine as a service. That’s because they have access to self-service, on-demand deployment of application profiles that model the full application stack, including the environment (underlying infrastructure resources and cloud services, web servers and other middleware, load-balancing and caching services, and other tool-chain components) as well as the latest build.

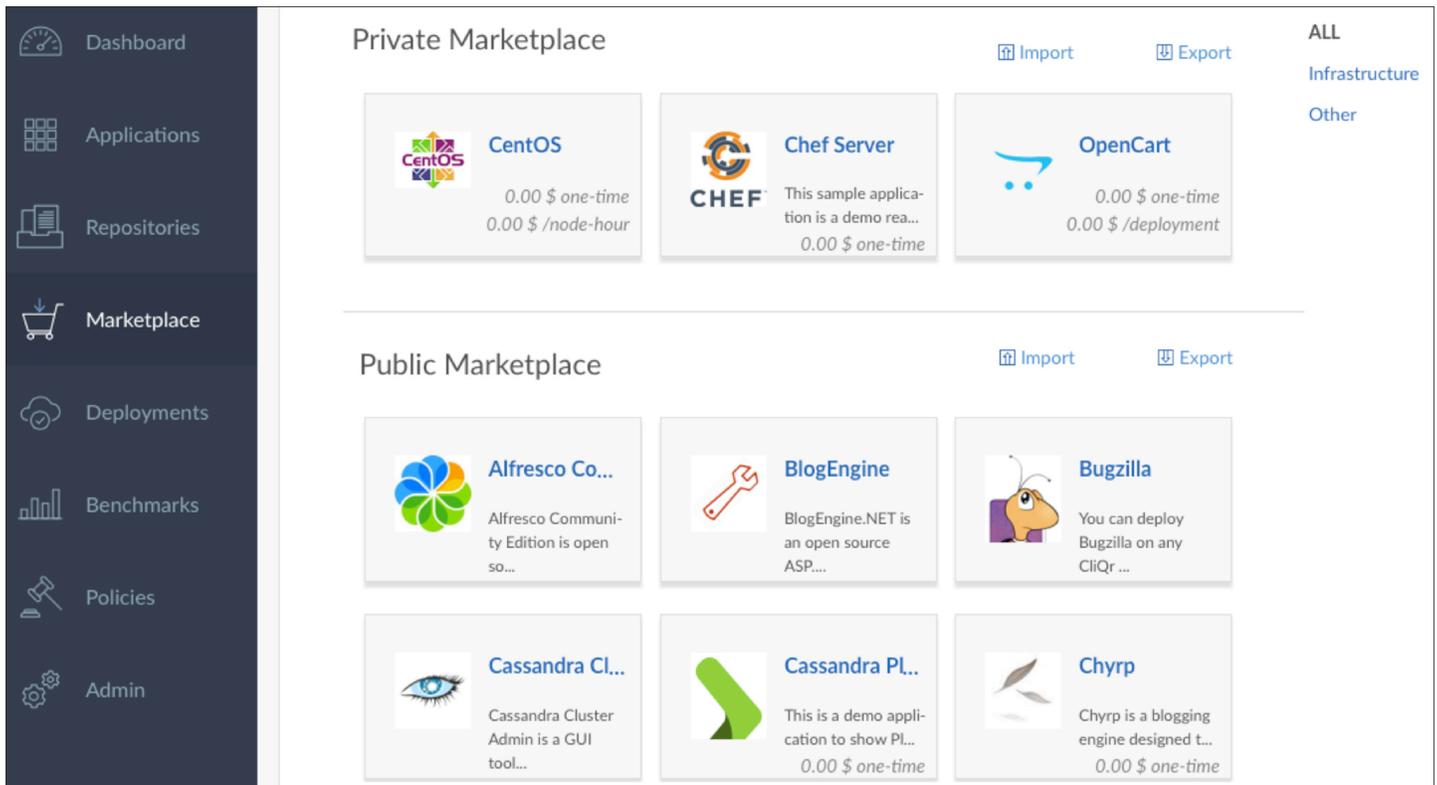
Cisco CloudCenter RBAC allows users to see only the application profiles they have rights to deploy, as shown in Figure 3. They can share with other users by publishing to the private or public marketplace, publishing to a third-party service catalog such as ServiceNow, or sharing application profiles directly with other Cisco CloudCenter users. At deployment time, users can select the appropriate deployment environment and view or modify default settings with a wide range of attributes and variables. Deployment of each application profile is also guided by a simple tag-based governance framework.

Cisco CloudCenter increases developer productivity and reduces risk by eliminating the variability inherent in manual provisioning and configuration. This variability often results in deployment problems that distract developers and negatively affect productivity. Variability also increases risk because it can negatively affect application quality and performance after the application leaves the development environment and moves into the production environment and begins operations.

Design pattern 1 best practices include the following:

- **Give users control.** Allow self-service, on-demand deployment to various cloud environments.
- **Ask users what they need.** Create application profiles for the most common deployment scenarios, and also give users the capability to model their own application profiles for their specific needs.
- **Don't forget DevOps engineers.** Give DevOps engineers their own application profiles to automate deploying common DevOps tools such as Chef Server.

Figure 3. Examples of application profiles ready for deployment



Cisco CloudCenter includes two notable features that simplify deployment while helping ensure control:

- **Deployment environment:** The deployment environment permits shared user access to an environment that may span clouds, but it is intended for a specific use and can be tied to a specific account. Users don't need detailed knowledge about specific cloud accounts, financial plans, or billing, and administrators and project funders retain visibility and control.
- **Tag-based governance:** This feature provides a framework to guide placement, deployment, and management decisions. Users select the appropriate tags when deploying a stack: for example, "Dev" or "Test" or "Prod." Tags can be tied to deployment environments, security profiles, and run-time management policies. With tags, users don't have to understand the details of the underlying policies.

## Design pattern 2: Integrated tool-chain deployment using the API

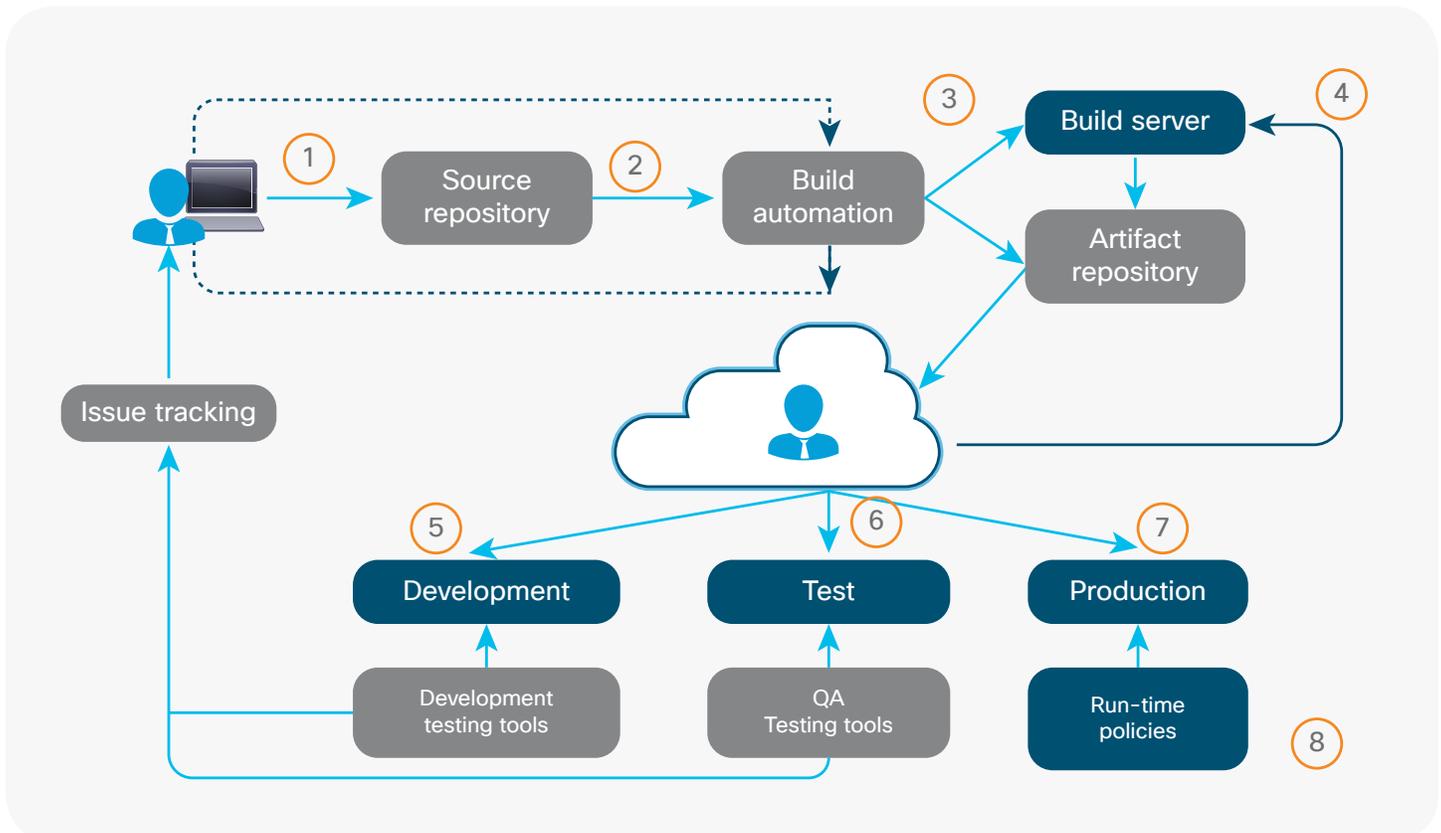
In design pattern 2, automation is increased. Instead of giving users a button for self-service provisioning of application stacks, this pattern exposes that service as a Cisco CloudCenter API. Development tools such as the Jenkins plug-in, and pipeline orchestration tools such as VMware CodeStream can call the API to automatically deploy the latest build of the application stack and run automated test routines.

Here, the Cisco CloudCenter solution is a foundational piece of an integrated continuous delivery tool chain.

It supports interoperability with a wide range of tools with different levels of integration:

- **Tight integration with plug-ins:** Cisco CloudCenter is tightly integrated through plug-ins with various tools such as Docker and the Jenkins plug-in.
- **API calls:** Cisco CloudCenter is loosely coupled through APIs to the source repository, artifact repository, and tier-level configurations: for example, using Chef, Puppet, or Salt.
- **Modeling as part of application profiles:** Various configuration management tools, test and QA tools, and Platform-as-a-Service (PaaS) services can be modeled as part of an application profile that consumes those services.

Figure 4. Cisco CloudCenter foundation for an integrated tool chain



As shown in Figure 4, a simplified integrated flow might include:

1. Developer checks in code.
2. Build automation monitors the source repository and initiates the build.
3. Cisco CloudCenter dynamically provisions build servers as needed.
4. A build process creates output (.war, etc.) and adds it to an artifact repository.
5. Cisco CloudCenter deploys the environment and latest build based on artifacts in the repository.
6. When development testing is successfully completed, Cisco CloudCenter promotes the application to the test phase and deploys the stack as needed.
7. When QA testing is successfully completed and the release engineer has given approval, Cisco CloudCenter promotes the application to the production phase and deploys the stack as needed.
8. After the application is in the production phase, Cisco CloudCenter can scale in place or implement hybrid-cloud high availability and disaster recovery as needed.

The Cisco CloudCenter solution includes two notable features that enable it to function as the foundation of the integrated tool chain:

- **Jenkins plug-in:** The plug-in provides tight integration with this popular build automation tool, so that Jenkins can call Cisco CloudCenter as part of the Jenkins build-management process. When Jenkins detects a code change, and creates a build, it then calls Cisco CloudCenter to deploy the appropriate environment and build as a single API call.
- **Mature API:** The mature API allows loosely coupled integration with a wide range of related tools. Each application profile has a unique ID that allows deployment from other tools.

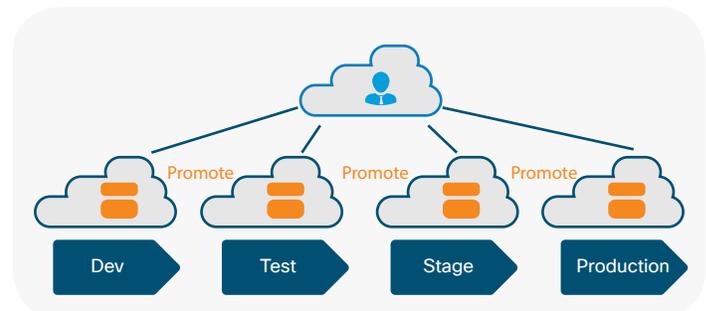
## Design pattern 3: Hybrid-cloud pipeline

Increasingly, enterprise IT organizations are using different environments for development, testing, staging, and production. For example, IT may use a public cloud or an OpenStack private cloud for development and testing activities, and then use a data center or private cloud VMware or Cisco environment for production.

The Cisco CloudCenter solution is well suited for this design pattern. The solution's "model once, deploy anywhere" technology makes it easy to create an application profile that defines the application stack's infrastructure requirements. The profile can be deployed in any supported cloud environment. And you can deploy it without the need to write cloud-specific scripts or workflows and without the need for cloud-specific teams and management tools.

As shown in Figure 5, Cisco CloudCenter supports deployment of both the build and environment and then promotion through the SDLC stages across the hybrid-cloud pipeline.

Figure 5. The Cisco CloudCenter solution supports hybrid-cloud SDLC



A hybrid-cloud pipeline increases business agility and cost efficiency by using cloud resources for temporary deployment during development and testing and then reducing risk by moving to a data center environment for more secure deployment.

From a developer's perspective, a deployment error or incorrect configuration can cause outage just as crippling as a software bug. As a result, environment errors can clog the delivery pipeline and lead to costly delays.

To avoid these problems, you need to maintain consistency across preproduction and production environments. You thus need to manage the preproduction environment with the same rigor as for the production environment. You need to handle automation code and configuration settings with the same processes and discipline as for application source code, and you need to automate the release and promotion processes to quickly, transparently, and reliably move from one environment to another. Using the single, unified Cisco CloudCenter platform to manage build and environment deployment in each different cloud environment solves this problem.

The Cisco CloudCenter solution can support a hybrid-cloud DevOps or continuous delivery strategy and can simplify and optimize development efforts, while at the same time optimizing business flexibility and cloud-scale economics.

The Cisco CloudCenter solution includes two notable features that enable it to support a hybrid-cloud pipeline:

- **Application profile:** Cisco CloudCenter uses patented technology to define application deployment and management requirements in a cloud-independent format. This JavaScript Object Notation (JSON) and XML file includes all the information that Cisco CloudCenter Orchestrator needs to deploy the build and environment in any supported cloud.
- **Continuous-integration and continuous-delivery project board:** This dashboard offers a visual view of project phases, deployment health, and the overall project cost and status. This dashboard is a feature and not a separate tool, so actions are linked to underlying infrastructure and application automation. Because the patented Cisco CloudCenter technology combines a cloud-independent application profile with a cloud-specific orchestrator, the solution works transparently across data center and cloud environments.

## Benefits

The Cisco CloudCenter solution is a powerful foundation for DevOps and continuous-delivery strategies. It helps match agile development with agile operations, increases development and operations efficiency, and delivers enterprise-class platform features that reduce complexity and the risk inherent in multiple-application, multiple-cloud, multiple-user integrated tool chains.

- **For developers:** Cisco CloudCenter boosts productivity and decreases time to market by empowering developers with the capability to use self-service to provision fully configured application stacks in any environment. They can focus on customer needs, and not infrastructure and deployment services. Developers no longer have to spend time manually installing and configuring or debugging deployment environments. And they have confidence in the consistency and repeatability of environments.
- **For DevOps engineers:** Cisco CloudCenter plays a foundational role in an integrated tool chain with automated work streams. Engineers can automate the deployment of builds and environments at multiple steps in a continuous-delivery pipeline across multiple data center and cloud environments.
- **For IT operations:** Cisco CloudCenter increases operation efficiency by delivering a single platform that helps ensure consistency across preproduction and production environments. It improves visibility and control by delivering a single management and orchestration solution that works across multiple applications, clouds, and users. With Cisco CloudCenter, IT can increase developer speed and agility while at the same time controlling costs and helping ensure governance control.
- **For IT executives:** Cisco CloudCenter enables a hybrid IT strategy with a flexible mix of data center and cloud IT services that eliminate cloud lock-in and exit costs as business needs change. Executives can support both business time-to-market and cost demands, while reducing complexity and managing the risk inherent in a hybrid-cloud strategy.

## Conclusion

For most IT organizations, DevOps and continuous-delivery strategies are a process, not a one-time project. Although many organizations work to link a variety of tools to automate the building, deployment, and testing of code, those chained tools need to work in the context of enterprise IT governance requirements.

The Cisco CloudCenter hybrid-cloud management platform automates deployment of application stacks, including the latest build and underlying environment across a range of data center and private and public cloud environments. It also provides enterprise-class features that reduce the complexity entailed in managing multiple applications, clouds, and users.

The Cisco CloudCenter solution adds value for those just starting their journey and seeking to automate deployment of application stacks in a single environment, those looking for automated deployment as part of an integrated tool chain, and those seeking to optimize hybrid-cloud pipeline management. The Cisco CloudCenter solution serves as a foundational part of a DevOps and continuous-delivery strategy for any enterprise IT organization.

## For more information

<https://www.cisco.com/go/cloudcenter>