Overview

In the digital economy, fast development and deployment of applications is critical to success. To thrive in this new application-oriented business environment, IT organizations are acting now to change their tools and processes to better support agile development methodologies. With more frequent releases of more complex application architectures across a growing mix of execution environments, organizations require changes in the ways that development and operations interact.

Continuous delivery—including both continuous integration and continuous deployment—is a methodology that automates and accelerates individual tasks at each phase as well as the overall software development lifecycle (Figure 1).

During the software development lifecycle, Developers (Dev) and Operations (Ops) staff typically interact in two ways:

- Dev requests infrastructure resources from Ops. Developers need temporary deployment of computing, networking, and storage resources to develop and test new applications and to update features of existing applications.
- Dev passes each new software release to Ops. Code changes move through a defined process from the development and test environments to the staging environment and finally to the production environment, where the operations staff typically implements strict process and environment controls.

Relationships at both of these touchpoints are often strained due to misaligned incentives, different skill sets, and different standard operating procedures.

DevOps was created to bring these two groups closer together and bridge the cultural divide separating them (speed versus stability).
Automation challenge

The development and continuous integration processes used to aggregate code changes and produce a build package with related artifacts are well defined and highly automated. However, the continuous deployment processes used to configure the infrastructure and environment and install the latest build (or deployable package) for the testing, staging, and production environments are often less mature. Lack of continuous deployment process maturity plus environmental complexity increase the difficulty of automating this critical processes.

Continuous deployment processes require two levels of automation:

- **Infrastructure automation**: Provision an environment, which might include virtual machines, storage resources, and network IP addresses as well as application and web servers, load balancers, and other software components.

- **Application automation**: Deploy and configure the latest build package, which includes code changes.

Artifacts that guide infrastructure and application automation (such as scripts, manifests, recipes, blueprints, and templates) are often environment specific. So automation routines used for deployment at one phase in one cloud or data center environment don’t usually work in another phase in another environment. Consequently, hard-wired automation presents a problem when different phases of the lifecycle occur in different environments.

IT organizations can’t increase speed and implement continuous integration and continuous deployment without automation. But the automation offered by traditional solutions doesn’t integrate application and infrastructure deployment, and it doesn’t work across environments. Because hard-wired automation doesn’t scale across cloud environments, multiple automation artifacts must be created, version controlled, and maintained. In addition, with the use of multiple single-purpose tools, organizations don’t get an overall project view that federates management of applications and projects, user groups, and execution environments.

To optimize the value of DevOps and continuous integration and continuous deployment automation, enterprise IT organizations need to combine application and infrastructure automation and enable consistent deployment across different data center and cloud environments. They also need to be able to manage deployed applications across a federation of users, applications, and clouds.

Unique benefits of the Cisco CloudCenter solution

The Cisco CloudCenter solution provides the technology foundation to support automated DevOps and continuous delivery strategies. The solution deploys both build (a deployable package) and environment (including infrastructure, database, middleware, web, application server, and other resources) resources on demand or through an integrated tool chain in any supported data center or cloud environment.
Application profile and orchestrator

Users create an application profile, which is a cloud-independent deployable blueprint of the entire application stack. The profile defines the deployment and ongoing management requirements for the build and environment. The Cisco CloudCenter solution also includes a cloud-specific orchestrator that is installed in each target deployment environment. The orchestrator abstracts the APIs and cloud services for each environment.

The combination of application profile and orchestrator enables developers, DevOps engineers, and operations personnel to quickly and repeatedly deploy applications to different development, test, and production environments using a fully automated process.

For example, an application can be developed in the Amazon cloud, tested in OpenStack, and implemented for production in a virtualized data center. Developers and project managers can launch a development project, get end-to-end visibility, and perform tasks such as scale and promote deployments, all without detailed knowledge of automation tools, APIs, or underlying infrastructure resources.

Single profile

Whereas other solutions separate infrastructure automation and application automation, the Cisco CloudCenter solution requires only a single application profile for each application stack. After a user creates a profile, that profile can be deployed on demand or through an integrated tool chain in any supported environment.

One platform

Unlike deployment solutions that are hard-wired to specific target environments such as Amazon Web Services (AWS) or VMware vCenter, the Cisco CloudCenter solution automates deployment in any environment. So users get fully automated, consistent deployment across all supported deployment environments, without the need to create and maintain target-specific deployment artifacts such as scripts and templates.

In addition to its model-once, deploy-anywhere capability, the Cisco CloudCenter solution offers enterprise-class administration and governance features that provide end-to-end visibility and control. Unlike tool chains, which consist of a patchwork of single-purpose tools, the Cisco® solution provides governance control including cost and usage controls, that spans the boundaries of applications, users, and clouds, all from a single platform.

Three design patterns

The Cisco CloudCenter solution supports three basic design patterns for DevOps strategies and automated continuous delivery processes:

- **Self-service environment:** Developers can use self-service to deploy on demand simple or complex application profiles to any supported data center or cloud. An example of a simple application is a Microsoft Windows Server 2008 or an Ubuntu image. An example of a complex application is a multitier application stack that includes a database, application server, load balancer, and multiple application packages, which may all be needed to develop and test code changes. The Cisco CloudCenter self-service capability empowers developers, and also reduces the amount of manual provisioning that IT staff has to perform. Cisco CloudCenter Role-Based Access Control (RBAC) determines which application profiles are available to each developer. Specified plans and budgets can be used to control the quantity and cost of resources deployed. And Cisco CloudCenter tag-based governance can be used to automate placement, deployment, and run-time decisions that guide developer actions—without the need for the user to know the details of the underlying policies.

- **Integrated tool chain:** The Cisco CloudCenter solution provides the foundation of an integrated DevOps tool chain. Code changes in the Integrated Development Environment (IDE) can trigger automated build and deployment processes:
  - The Jenkins plug-in allows a Jenkins build to be deployed to any environment provisioned by the Cisco CloudCenter solution.
  - The Cisco CloudCenter solution leverages existing Chef and Puppet investments. Users can easily model nodes provisioned by these and other configuration management tools in the application profile.
  - Docker containers can be modeled as part of an application profile and deployed to any supported data center or cloud environment.
A ServiceNow service catalog can include Cisco CloudCenter application profiles. The profiles can be automatically imported. Because only one profile is required for each application stack and it can be deployed to any target cloud, the number of catalog items is significantly reduced.

**Cross-cloud software development lifecycle:** A software development lifecycle that crosses multiple environments can be managed by a single platform. A project can be promoted from AWS development environment, an OpenStack private cloud test environment, to a virtualized data center production environment. Use of a single application profile to automate deployment in both preproduction and production environments reduces errors and inconsistencies that impede developer efficiency.

**Advanced features**

The Cisco CloudCenter solution delivers a number of innovative features that support automated deployment as part of DevOps or continuous integration and continuous deployment efforts.

**CICD project board**

The solution includes an integrated CICD project board with a visual view of project phases, deployment health, and overall project costs and status (Figure 2). The project board is a Cisco CloudCenter management platform feature and not a separate tool.

Actions are linked to underlying infrastructure and application automation. Because the patented Cisco CloudCenter technology combines the cloud-independent application profile with the cloud-specific orchestrator, the solution works transparently across data center and cloud environments. Developers and project managers can improve efficiency and simplify tasks without having to understand the underlying infrastructure and application automation, cloud services, or APIs.

**Jenkins Plug-in**

The Jenkins plug-in allows Jenkins to call the Cisco CloudCenter API to deploy the latest build in a Cisco CloudCenter provisioned environment. A developer making a code change in the IDE triggers deployment of an application stack that includes the latest Jenkins build in the target data center or cloud environment. The Cisco solution maintains version control of each application profile and can track multiple deployments at each project phase.
Composite Topologies

The Cisco CloudCenter solution can model and deploy application profiles that include a variety of OS image formats, application and cloud services, containers, and nodes deployed by configuration management solutions.

As illustrated in Figure 3, composite topologies can be easily modeled as part of any application profile and then deployed and managed in any data center or cloud environment. With the Cisco CloudCenter graphical topology modeler, users can drag and drop more than 30 ready-to-use application services into the profile and easily modify services. Users can also create their own custom services and add Docker containers and Chef and Puppet nodes.

Figure 3. Model with application services, containers, and configuration management

Real-world examples

Cisco customers have used the power of the Cisco CloudCenter platform in a range of DevOps scenarios:

- **A global positioning technology provider** offers a proprietary global positioning platform for use by heavy industry. The provider uses the Cisco CloudCenter solution to automate deployment of custom development and production environments on AWS, with a composite topology that includes more than 20 virtual machines, multiple load balancers, and 30 custom parameters that link back to global positioning data.

The company has replaced its previous traditional service request and manual provisioning process, which took four days to complete. With the Cisco CloudCenter solution, developers can immediately obtain the environment they need with one click, and when applications are ready, they can be deployed to the production environment on demand. The approach has reduced the company’s overall new product release time from 18 months to 12 months and allowed the company to shift from quarterly product updates to continuous deployment.
• **An independent software vendor** has simplified its development process to enable rapid deployment of production releases to multiple data center and cloud environments. The vendor can deploy applications on demand and manage multiple release versions for production in any cloud that its customers choose. The vendor now uses the Cisco CloudCenter solution to model most current releases and multiple past releases in cloud-independent application profiles. The company deploys the profiles on demand to target clouds and maintains and supports ongoing production.

• **A media content delivery provider** has optimized the continuous integration and continuous deployment process with automated deployment at every phase of the application lifecycle, including production. The provider has created profiles and automated releases for more than 25 applications. The company replaced its previous manual service request ticket and provisioning with the Cisco CloudCenter self-service solution for developers. The result is reduced deployment effort, increased quality, fewer errors, and accelerated deployment across the application portfolio.

**Cost Controls**
The Cisco CloudCenter solution supports cost- and usage-based plans and bundles to help ensure that developer self-service operations do not result in cost overruns that undermine the strategic value of DevOps and continuous integration and continuous deployment automation.

IT can centrally control and monitor cost and usage and report costs back to specific development projects, even if lifecycle phases are in different data center and private and public cloud environments. Developers can use the self-service process to deploy to the cloud on demand and don’t have to manage cloud accounts and billing.