Why DevOps needs ServiceNow® Cloud Provisioning and Governance

Agile governance for continuous deployment and operations

June 2020
Introduction

DevOps teams face a unique challenge: delivering a never-ending torrent of releases while still maintaining appropriate controls. This need to balance speed and governance—without compromising on either—doesn’t just apply to software. It also applies to the cloud infrastructure on which these applications run.

DevOps teams typically provision cloud infrastructure using infrastructure-as-code (IaC) tools such as Terraform, Amazon CloudFormation, Azure ARM, or Google GDM. However, while this template-based approach is agile, it doesn’t provide the governance needed for business-critical applications—for example, ensuring enterprise policy compliance or appropriate patching levels. It also doesn’t create the crucial link with operational processes, such as updating the CMDB to support topology-based event correlation and other functions.

ServiceNow® Cloud Provisioning and Governance (CPG, formerly ServiceNow® Cloud Management Platform) fills this critical gap. It integrates seamlessly with leading IaC tools to deliver comprehensive governance and effective operational integration without compromising speed and agility.
CPG and Terraform: from code to deployment

How does CPG manage the cloud infrastructure journey from code to deployment? Let’s look at an example of how it does this with Terraform, providing a complete multi-cloud provisioning and governance solution:

- It starts by importing a Terraform template. This template defines the deployment architecture of a particular cloud stack. The template can either be a specific development artifact (e.g., in Git) or a generic best-practice template.
- The template is attached to a ServiceNow catalog item. This catalog item is a service offering and uses the Terraform template to provision the service on demand.
- The catalog item also defines and enforces role-based policies for the service offering—for instance, user permissions, approvals, quotas, naming conventions, or workload placement.
- In addition, the item can automate post-provisioning activities to bring the service instance into compliance—for example, applying patches to harden operating systems. It can also install additional applications (such as monitoring tools or Tomcat) as required.
- Because the catalog item drives the deployment workflow, it can also prepare the operational environment for the service. This includes automatically updating the CMDB—as discussed previously—issuing change requests for auditability, and sending notifications to appropriate stakeholders.

Developers request instances of these cloud services using the catalog item. This can be done through the ServiceNow UI. It can also be done using an API, allowing CPG to be directly integrated into the CI/CD pipeline.

Note that while the preceding example describes how CPG works with Terraform, similar integrations are available for vendor-specific IaC tools including CloudFormation, ARM, and GDM.
Guardrails, not straightjackets

In the cloud, change is measured in hours—or even minutes. This makes traditional governance approaches far too slow and resource intensive, placing DevOps in a governance straitjacket.

That’s why CPG provides non-intrusive policy guardrails. These guardrails only come into play when there is a policy exception. CPG only escalates requests for approval when users stray beyond these configurable guardrails—for instance, exceeding a quota or trying to modify a production environment without permissions. Otherwise, CPG provisions the service right away and automatically closes out the corresponding change request.

This means that introducing governance as a deployment step in a Jenkins or Azure DevOps CI/CD pipeline doesn’t slow the pipeline down.

Infrastructure deployment and management with guardrails

Frictionless consumption for DevOps teams while still ensuring adherence to corporate IT governance policies

Outcomes

**Secure**—restricted images, ACL-driven catalog item access

**Compliant**—tagging enforced, region associations

**Governed**—not just deployment but ongoing management including deprovisioning and configuration checking

**Change mgmt**—actions/ops integrated with platform change management

**Policy driven**—naming, lease, schedule, tags, approvals

**Audited**—trails for each and every action
**CPG works the way your developers work**

Application developers are used to writing deployment scripts. With CPG, they continue to do this—it uses exactly the same Terraform templates that your developers already create. They don’t even need to create a catalog item. By having a separate cloud catalog designer who works with developers to create catalog items, you can centralize this competency and ensure that governance is applied consistently across all of your cloud infrastructure deployments.

And, once the catalog item is created, there’s no need for developers to go to a cloud portal to deploy the environment. They simply need to create a script that provides the deployment parameters in JSON format.

**Why CPG and Terraform?**

Increasingly, ServiceNow customers are adopting template-based IaC tools to provision cloud infrastructure. These tools are well documented and constantly updated, making it easy to specify cloud infrastructure and leverage the latest cloud vendor capabilities. However, IaC tools such as Cloud Formation, ARM, and GDM are limited to a single cloud vendor. Terraform, on the other hand, provides a multi-vendor cloud provisioning solution, allowing customers to provision multi-cloud infrastructure using a single, consistent tool.

ServiceNow CPG works seamlessly with Terraform to deliver a comprehensive multi-cloud lifecycle management solution that includes provisioning, governance, discovery, and ongoing resource management and operations. By combining Terraform—the most popular IaC platform—with world-leading ServiceNow IT Service Management and IT Operations Management, DevOps teams drive agility while ensuring compliance with enterprise standards and policies. ServiceNow’s foundational CMDB enables unauthorized change detection, drift tracking, and broad governance to complement Terraform’s easy deployment of cloud resources, making this solution a critical pillar in enterprise digital transformation.

To find out more about ServiceNow Cloud Provisioning and Governance, visit [https://www.servicenow.com/products/cloud-management.html](https://www.servicenow.com/products/cloud-management.html)

---

**Multi-cloud management with ServiceNow and Terraform**

<table>
<thead>
<tr>
<th>ServiceNow Visibility (CMDB)</th>
<th>Automation</th>
<th>Governance</th>
</tr>
</thead>
<tbody>
<tr>
<td>Provisioning (Terraform)</td>
<td>Discovery</td>
<td>Event Management</td>
</tr>
<tr>
<td></td>
<td>Event</td>
<td>Day 2 Operations</td>
</tr>
<tr>
<td></td>
<td>Management</td>
<td>Lifecycle Management</td>
</tr>
</tbody>
</table>

**Multi-vendor cloud infrastructure**

- Google
- AWS
- Azure
- VMware
- Other
Additional information

CPG Terraform connector in the ServiceNow store (free add-on)
https://store.servicenow.com/sn_appstore_store.do#!/store/application/9b04ff62b1323002530a387b6673a1d/1.0.8

Setting up Terraform integration (blog)
https://community.servicenow.com/community?id=community_article&sys_id=ec83826f1b430490d01143f6fe4bcb2

Scripted REST APIs offered by ServiceNow CPG for catalog requests (blog)
https://community.servicenow.com/community?id=community_article&sys_id=8d5521d8db633b805129a851ca9619a0