

Evaluating Continuous Delivery for Puppet Enterprise

Continuous Delivery for Puppet Enterprise offers a complete end-to-end continuous delivery solution that's purpose-built for building, testing, and deploying Puppet automation. With Continuous Delivery for Puppet Enterprise, organizations can confidently scale Puppet automation across multiple teams, increasing IT velocity without sacrificing reliability.

IT organizations find Continuous Delivery for Puppet Enterprise best suited to:

- Confidently scale the IT organization's ability to contribute to infrastructure configuration changes
- Protect the infrastructure from unplanned or unwanted changes by analyzing the potential impact of a change before rolling the change out
- Codify and measure the delivery pipeline to continuously improve code deployment
- Test infrastructure changes prior to deployment, eliminating common bottlenecks within the software delivery lifecycle process








IT organizations find Puppet Enterprise without Continuous Delivery is best suited when:

- There's one small team managing all of the infrastructure change
- The infrastructure is relatively small and unintentional changes are rare

Solution Comparisons

Below are a set of important capabilities that organizations look at when deciding on a solution.

Capability	Continuous Delivery for Puppet Enterprise	Puppet Enterprise without Continuous Delivery	Other solutions
<p>Impact analysis</p> <ul style="list-style-type: none"> • What it is: Understanding the impact of a proposed change before accepting the new changes into the mainline Puppet code base • Why it matters: Understanding the potential impact of a change prevents changes from having an unintended scope on the infrastructure. 	✓	✗	✗
<p>Rolling infrastructure deployments</p> <ul style="list-style-type: none"> • What it is: Ability to apply an infrastructure change in batches to catch problems early. • Why it matters: Pushing a change out to all systems at once doesn't scale well and makes it impossible to catch problems early to minimize the impact. 	✓	✗	<p>—</p> <p>No other solution can perform rolling updates of Puppet changes, but some solutions can roll out other infrastructure changes.</p>
<p>Deployment approvals</p> <ul style="list-style-type: none"> • What it is: Protect mission critical Puppet environments from unauthorized deployments. • Why it matters: Ensure all changes to mission critical Puppet environments are safe and have followed the defined CI process. 	✓	✗	✗

<p>Holistic view of all in-flight changes and status</p> <ul style="list-style-type: none"> • What it is: View all in-flight control repo and module changes, their status, and where they're currently deployed. • Why it matters: Understanding the infrastructure's current state enables teams to quickly remove blockers, identify potential issues, and debug problems quickly. 			<p>—</p> <p>Custom solutions can be built on other CD products, but nothing exists out-of-the-box</p>
<p>Purpose built delivery pipelines</p> <ul style="list-style-type: none"> • What it is: Pipeline tests, deployments, and visualizations that are purpose built for Puppet Enterprise and works out-of-the-box. • Why it matters: Simplifying the adoption of CD practices for infrastructure automation helps build competency, trust, and promotes automation across an IT organization. 			
<p>Decentralized automation with centralized governance</p> <ul style="list-style-type: none"> • What it is: Give IT teams ownership of the automation for the services and applications they're responsible for. Give Core Infrastructure IT ability to keep things running. • Why it matters: IT teams need to deliver change at the pace of development while core IT needs to keep the business running. 		<p>—</p> <p>Puppet Enterprise can help decentralize automation across multiple teams, but requires manual promotion processes or custom solutions.</p>	

Scaling Automation Throughout the Organization

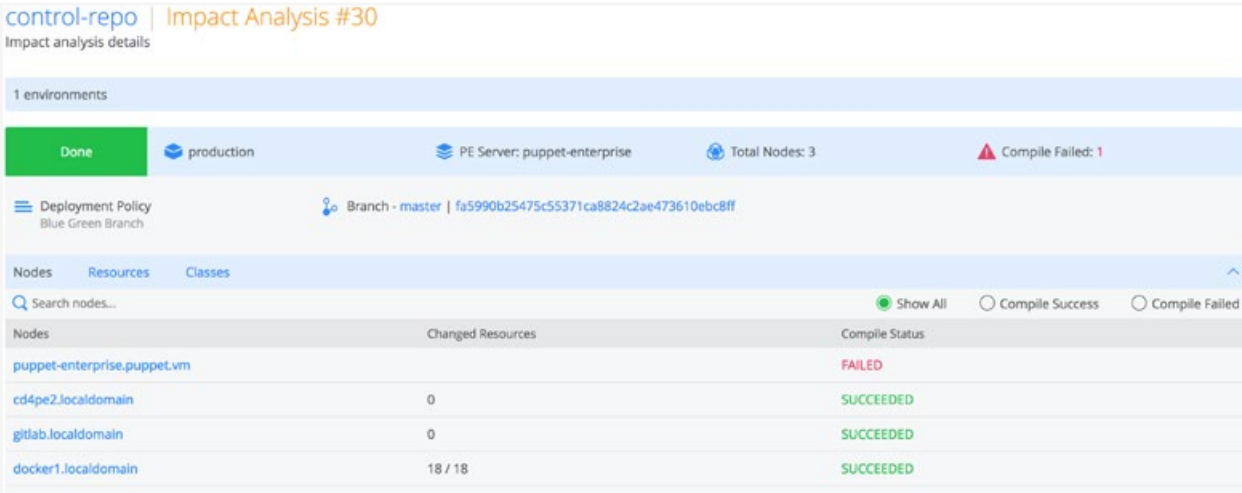
Larger IT organizations often have many individual contributors and specialized teams all working together to manage the infrastructure. Operating at scale is a challenge that will never go away, and the tools you choose can make the difference between working effectively as a team and perpetuating — or even reinforcing — siloed thinking, turf wars, and disjointed processes.

The tool you use needs to address several challenges:

- **Detect unintended impact.** Changes can often have a much larger scope than intended. Having confidence where a change is going to apply, not to mention what exactly is going to change, is nearly impossible.
- **Detect conflict.** Any infrastructure tool needs to understand when multiple teams or contributors intend to manage the same configuration differently. This needs to be known prior to any work being done on the system.
- **Decentralize infrastructure change.** IT organizations need to move at the pace of development. Application delivery teams need to push updates to supporting infrastructure, while Core Infrastructure IT teams need to ensure all the systems stay running. An infrastructure CD solution needs to keep IT teams unblocked, but prove that all changes meet core governance policies and safety checks.

Impact Analysis

Impact Analysis is a feature in Continuous Delivery for Puppet Enterprise that analyzes new proposed Puppet code manifest changes and analyzes what nodes and configurations will be affected by the new change. Armed with this information, you can quickly review new pull requests and have the confidence that the changes will only affect the infrastructure and configurations they are intended for.



The screenshot displays the 'Impact Analysis #30' interface in Puppet Enterprise. It shows details for a deployment policy named 'Blue: Green Branch' on the 'production' environment. The interface indicates that the compile failed for 1 node out of a total of 3 nodes. A table below lists the nodes and their compile status:

Nodes	Changed Resources	Compile Status
puppet-enterprise.puppet.vm		FAILED
cd4pe2.localdomain	0	SUCCEEDED
gitlab.localdomain	0	SUCCEEDED
docker1.localdomain	18 / 18	SUCCEEDED

Automation at organizational scale

IT organizations often have to balance between moving faster to keep up with business needs and providing safety to keep the business running. IT operations is responsible for defining the requirements to ensure infrastructure changes are safe. At the same time, other infrastructure and application deployment teams need to define the delivery process that's right for them and deliver changes at the pace of development.

Often, core IT automation teams will gate all changes to any infrastructure environment, including pre-production environments, in order to maintain safety. This greatly reduces IT throughput as core Puppet teams become a bottleneck to delivery infrastructure change.

What's needed is a way to build CD pipelines for each Puppet module that can be deployed and tested independently of other modules. When it's time to release the changes to a production environment, core IT needs to ensure that all changes have gone through the required SDLC process and are safe.

Questions to answer

As you consider the right tools for your organization, it helps to ask some questions about how your organization is running now and your organization's future plans. The questions below should help you in your decision making.

How many contributors to the infrastructure code base will we have?

If the answer is more than a few, you need to plan how you're going to scale everyone's contributions to the infrastructure code base. You will want to make sure your tool of choice can:

- Prevent a single configuration from being managed multiple ways by different contributors.
- Be easily approachable (i.e., easy to read and learn) from a diverse set of background disciplines.
- Enable teams to independently author, deploy, and validate changes and go through an integration process to ensure safety.

How complex are our deployments?

Many times, the complexity of a deployment is in managing the relationships between systems, rather than in managing the same system through multiple states. You'll need the ability to roll-out changes, understand the progress of the deployment, and automatically halt if things aren't going as planned.

What Continuous Delivery experience do we have?

Adopting continuous delivery practices can be a long process. Each bottleneck in the process must be assessed individually, and solutions implemented quickly. If your team is new to continuous delivery, will they have the expertise to build something completely custom for your Puppet Enterprise automation?

We want to hear from you

As you investigate your continuous delivery needs, let us know how we can help you answer questions and where CD practices and Puppet Enterprise automation can help your organization move faster with less downtime.