Published **April 22, 2024**

Manage on-prem Kubernetes clusters with a Private Cloud Gateway



Carolina Delwing Rosa
Education Engineer

Clusters here, there and everywhere

Like most enterprises, you're probably managing multiple Kubernetes clusters — and those clusters are everywhere.

According to our 2023 State of Production Kubernetes report, 69% of organizations already run Kubernetes in multiple clouds or other environments.

And for all the growth in public cloud computing, there's a very good chance that some of those clusters are running in private clouds or enterprise data center infrastructures.

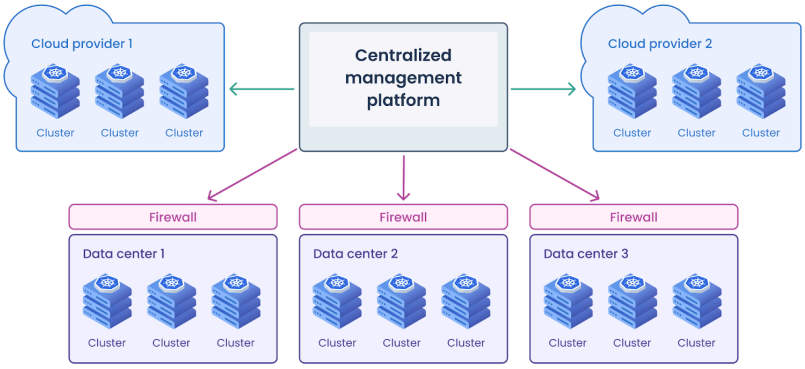
In fact, nearly two-thirds of those using Kubernetes in production are running clusters in virtualized data centers, and one in four are running on bare metal.

Using multiple environments is a good thing — whether motivated by separation of concerns, multi-region availability, or strategic multicloud approaches. It's a hybrid cloud world. But for all the benefits

managing multiple clusters across different environments is a technical challenge.

How can you manage when you're behind a firewall?

Like many, you're probably considering a centralized management platform to give you unified contr of clusters across cloud services, edge, and the data center.



When you have clusters in multiple environments, a centralized management platform makes sense – but it will always be outside of the DC firewall.

But to do its job, somehow this external platform needs to cross your firewall to talk to the on-premises Kubernetes clusters inside your data center or private cloud.

Inbound connections like this commonly raise all kinds of security policy red flags for your network an security teams. They don't like to add exceptions to firewall rules or open ports. Organizations in heavily regulated industries like financial services and healthcare may not allow them at all, or the internal approvals process may be arduous, with a lot of scrutiny from architects and infosec folks.

What if there was a way to create a bridge between your private data center environment and an outside management platform, in minutes and without the need for complex firewall rules? There is.

Introducing the Private Cloud Gateway (PCG)

A private cloud gateway (PCG) is an application that you run inside your private environments. It acts a a bridge between your private cloud or data center and an external management platform, such as Spectro Cloud Palette, by creating a secure outbound connection from within the on-prem environment.

This allows external management tools like Palette to deploy Kubernetes clusters in private environments without accessing the environment directly.

Although there are other ways to establish a connection between a cluster in your private environmen and an external management tool, they are often much more manual and fragile — a gateway-based architecture like Palette's is becoming an industry best practice.

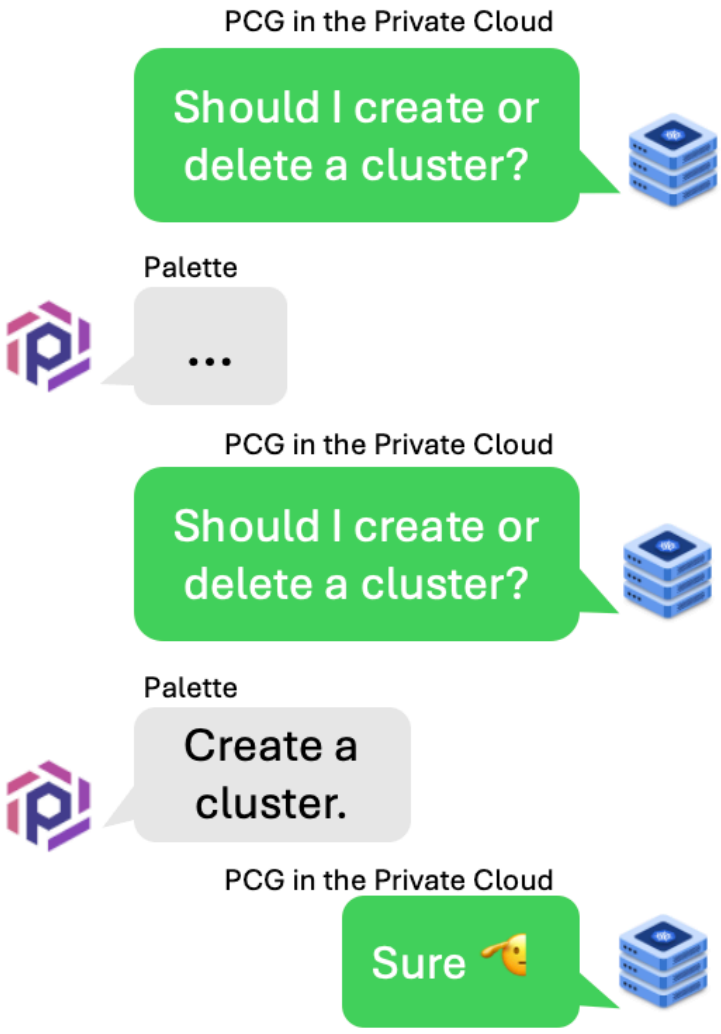
How does a PCG work?

The PCG is a containerized application that you install in a small one- or three-node Kubernetes cluste inside your private environment. If your environment is VMware vSphere, MAAS, or OpenStack, we offer first-class support through the Palette CLI. However, if you use a different environment, you can deploy a PCG to an existing Kubernetes cluster.

If you have multiple private data center locations with Kubernetes clusters to manage, you'll deploy a PCG in each of those locations. Palette centrally manages all these PCGs, and you can access them ir the Palette UI.

Once installed, the PCG registers itself with a Palette instance, initiating an outbound connection toward Palette using an encrypted communication channel.

The PCG regularly polls the central Palette management plane, looking for instructions to either deploy or delete Kubernetes clusters within the environment.

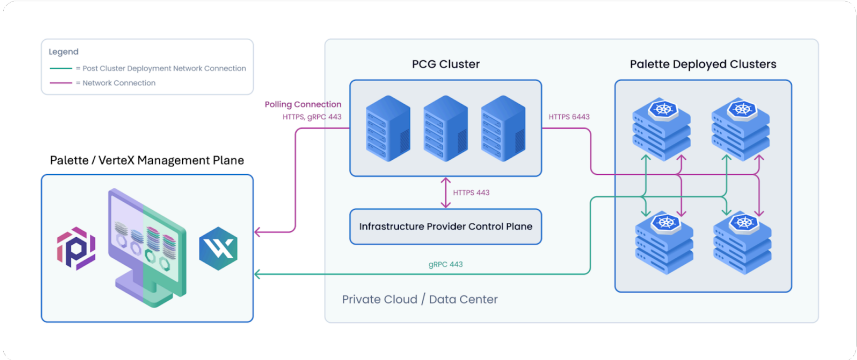


When you click on the "Deploy Cluster" button in the Palette UI, the PCG will see the instruction from Palette on its next poll, and start the deployment process within your environment by communicating with the local infrastructure provider to request resources.

Once the new cluster is up and running, the PCG's work is done. Each cluster communicates directly with the central Palette management plane through the Palette agent available in each cluster. This agent is responsible for originating all outbound network requests toward Palette, managing cluster workloads, and handling day-2 operations like patching and scaling.

Because each Kubernetes cluster communicates with Palette directly for day to day management, th PCG never becomes a bottleneck, even as the number of clusters scales up. In fact, even if the PCG goes down, cluster operations will continue completely unaffected. The PCG plays a very different role than a traditional 'management server' such as Rancher uses.

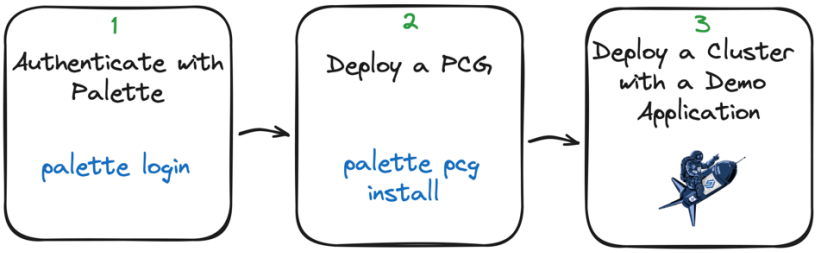
The diagram below illustrates the network communication between Palette and the PCG.



Deploy a Cluster with Palette and a PCG

If you're keen to see the PCG concept in action, we have a new tutorial on our documentation site, Deploy App Workloads with a PCG, that will walk you through step by step.

It gives you detailed instructions on how to deploy a PCG in a VMware vSphere environment and then using this PCG to deploy a cluster and launch the Hello Universe Palette demo application into this environment.



Let's take a look at the sequence of steps to learn how to securely deploy and manage clusters in private environments with Palette and a PCG.

Authenticate with Palette

The first step to deploying a PCG is to use the Palette CLI to authenticate with your Palette account. You will provide your Palette URL and API key. Once you're authenticated, every command you issue using the Palette CLI will be executed against the configured Palette environment.

Deploy the PCG

Once authenticated, the next step is to use the Palette CLI and the 'palette pcg install' command to deploy your PCG cluster.

The CLI will prompt you for the information required to connect to your environment, such as the endpoint, username, and password. You'll also be asked where to place your PCG cluster within your environment, such as the Resource Pool and Network. These parameters are specific to the environment you use.

Lastly, you'll set up the size of your PCG cluster to suit your needs: the larger the PCG, the more concurrent cluster deployments it can support. You can specify the number of nodes, IP placement strategy, and the configuration of your nodes (CPU, memory, and storage). This is a brief review of the parameters; please take a look at the tutorial for the complete list.

After answering the prompts, you can let Palette CLI do its magic. It will create a local kind cluster on the machine from which you're executing the Palette CLI commands, which will then be used to bootstrap the PCG cluster in your environment. You can monitor the progress of the PCG creation through your terminal. Within a few minutes, your PCG will be installed and ready to use.

Deploy a cluster

Now the PCG is functioning, you basically don't need to worry about it again. In the background, as we described above, it'll regularly poll outbound to the Palette management plane, asking if there are any pending instructions to create or destroy a cluster.

From your perspective, you can now head into Palette and choose your private cloud environment as the destination for a new cluster, as many times as you like. You won't need to provide credentials or connect directly to the data center to initiate the deployment; the PCG will take care of it.

The final part of our tutorial leads you through creating a Cluster Profile, which allows you to model what goes into your cluster — in this case, it will include a layer for the Hello Universe demo application. Then, you will deploy a cluster using the created Cluster Profile and access the application via your browser to prove it all works!

What's next?

In this blog, we explored why private environments are challenging to manage, and how a PCG approach solves the problem by avoiding the need for inbound connections. You've learned how Palette's PCG works step by step, and how to deploy a PCG yourself in a short walkthrough.

Check out the Deploy App Workloads with a PCG tutorial for a detailed step-by-step guide on deploying Kubernetes workloads using a PCG in a VMware environment. To complete this tutorial, you will need a Palette account. Book a demo with us to get started.

You can also explore the Private Cloud Gateway section in our documentation. There, you'll find detailed instructions on how to deploy a PCG to VMware vSphere, MAAS, and OpenStack. If your

environment is not listed, don't worry; we've got you covered with the Deploy a PCG to an Existing Kubernetes Cluster guide.

If you found this content useful or have any ideas or feedback, please don't hesitate to reach out. You can head to our Slack community to connect with the Spectro Cloud team.

Apr 22, 2024

Spectro Cloud

The fast lane to production AI

Spectro Cloud helps platform teams and cloud providers modernize and manage infrastructure for the AI era without adding more tools or operational complexity.

With PaletteAI, enterprises, public sector organizations, neoclouds and sovereign clouds can build, govern and operate full-stack environments across VMs,

Products

PaletteAI

Editions

Deployment options

Services and support

Integrations and environments

Security

Solutions

AI factories

Edge AI

Kubernetes fleets

Token costs

AI as a Service

Sovereign AI

Edge infrastructure

VM estates

Government

Resources

Getting started

Palette docs

PaletteAI docs

Support portal

API docs

Resource center

Design hub

Status page

Why Spectro Cloud

For AI

For edge

For fleet management

For government

Awards

Company

Contact us

About us

Trust center

News

Community

Careers

Partners

NVIDIA

AMD

AWS

HPE

Our ecosystem

Partner portal

Kubernetes, edge, regulated and air-gapped locations, and AI infrastructure.

PaletteAI Launchpads help teams start quickly with urgent outcomes such as VMware migration, token cost control or edge modernization, then scale into enterprise-wide lifecycle management, governance and fleet operations on the same platform.

Get all the latest from Spectro Cloud

Sign up for our newsletter

Follow us

©2026 Spectro Cloud. All rights reserved.