Edge node architecture with npm Enterprise

Case Study

AT A GLANCE

CATEGORY: Developer Tools

CUSTOMER SINCE: 2015

SOLUTION: Replicated Core

USE CASE: Edge nodes

BEST FEATURES: Updates, Installation

RESULTS: Time to market, CI/CD integration, Enterprise features

OCTOBER 2017
The Story

npm, Inc. is the company that hosts and manages npm, the most widely used package manager for JavaScript, used daily by 3.4 million developers worldwide with 2.5 billion packages downloaded every month. JavaScript developers rely on npm to speed their application development process by providing a robust library of over 500,000 reusable and independently maintained code modules (npm packages). npm is used by more than 30,000 companies, including DocuSign, SiriusXM, Uber, and Visa, to manage and deploy packages.

Ben Coe, Senior Systems Analyst, npm

The Opportunity

Many enterprises have strict requirements that prevent them from using cloud-hosted products for critical parts of their infrastructure. A condition that makes sense from a regulatory compliance perspective, but it makes life inconvenient for developers within those companies who wish to take advantage of open-source code from the npm Registry, or who want to use npm to share and reuse their code with their colleagues.

npm Enterprise (npmE) allows developers at big companies to run a version of the npm Registry behind their firewall. Of course, it wouldn’t be enough for enterprise customers to just deploy a fresh install of npm Enterprise with an empty registry. Much of npm’s value comes from the 500,000 packages available on the public registry and being able to combine these packages with private code. Without access to these packages, developers would waste time reinventing a lot of wheels.

npm Enterprise lets companies mix public packages from the public registry with the private code stored within their private registry without risks or complexity.

Over 500,000 npm Packages available
Enter edge node architecture

The team at npm designed npmE so that each npmE server is a private edge node to the public registry. Every npmE instance can replicate select parts or all of the npm Registry to offer this functionality to end users. It also provides additional local services that are only accessible to these users, based on a company's unique requirements.

An npmE customer can configure a private registry as a full mirror of the public Registry to decrease latency, cut bandwidth costs, and offer npm Registry to end users who are restricted from accessing the public internet. Alternatively, they may selectively mirror npm's Registry using specific whitelists managed by the admin console or the npm command line.

When combined with npmE add-ons which enforce code quality, evaluate how packages and their dependencies are licensed, and scan for security vulnerabilities, this architecture gives companies total control of the public packages their developers use. At the same time, these developers may find, share, and re-use proprietary code by publishing to their private local registry. Private code stays private by never leaving the company's infrastructure.

End users don't have to think about where each package is located; they can just pull from the npm Enterprise server. Behind the scenes, the server automatically determines the scope and proxy of the package pull.
But how?

There are two primary challenges of an edge node architecture:

1. Enabling each enterprise IT admin to independently install, configure, deploy, manage, update, and integrate their instance into their infrastructure.
2. Maintaining the agility of the cloud hosted edition without getting bogged down with enterprise requirements.

For many years, deploying and maintaining a private instance of this kind of architecture would have been prohibitively difficult for everyone but the most advanced IT organizations. These sorts of enterprise software installations took several months to implement and involved manual processes of configuring servers, runtimes, and components. Every enterprise IT org would have taken responsibility for the ops role of their enterprise instance.

Fortunately, it’s now much easier to deploy and maintain private edge nodes thanks to technologies like containerization, orchestration, and scheduling platforms.

Deployment and management are now baked into the design and development effort of modern applications. This creates reproducible and consistent cloud-native deployments, but it also is becoming the foundation of modern enterprise software implementations. This automation and inherent portability allow their customers to deploy into their environments without deep knowledge of their architecture.

Of course, it isn’t quite as easy and magical as it all sounds. npm initially built out their own containerized installation methods by packing all of their services into a single container. This approach still required npm Enterprise customers to be quite technical to complete the deployment. Their system also lacked the tooling for managing versions, customers, backups, and updates.

Prior to leveraging Replicated, npm Enterprise lacked the tooling for managing versions, customers, backups, and updates. These features have sped up customer adoption and streamlined internal processes for maintaining both a cloud and on-prem version.
Enter Replicated

After a few months of banging their heads against the wall, they decided they were dedicating too many resources to deploying and managing Enterprise instances. They switched over to the Replicated platform to be their enterprise management platform. Replicated’s platform provides workflows for integrating existing CI/CD release pipeline with their enterprise release channels.

By leveraging the Replicated release channel management process, they use discrete versioned images for each of the services that make up npm Enterprise. They organize these into the default channels provided — “stable,” “beta,” and “unstable” — and when they promote a set of images to “stable,” Replicated automatically notifies customers that an update to npm Enterprise is available and makes the update as simple as a single click. Their enterprise customers don’t have to update services manually, and they don’t have to manually push containers around to keep the edge nodes of the npm Registry on recent versions.

Beyond deployment and management, the npm team also faced the problem of developing enterprise-specific features such as change management processes, LDAP integration, and an admin dashboard, which enterprise customers require but which fall outside their core product expertise. Many of these features are included (or at least made easier) by the Replicated platform and provide a consistent and familiar experience to enterprise IT admins.

These sorts of enterprise ready features are important to their enterprise customers, but since they aren’t a core part of their value proposition, it has made a ton of sense for them to leverage a partner to power these as much as possible.

What’s next

Edge node architecture is still evolving, but it is gaining more traction in a variety of use cases. An increasing quantity of JS developers rely on npm, and as a result, an increasing number of enterprises will need npm Enterprise. For developers to be effective, it’s imperative that they benefit from the global Registry of npm packages.

By partnering with Replicated to pioneer an architecture that delivers that promise while reducing management overhead and satisfying security requirements, there is bright future emerging that embraces the distributed nature of the internet.