Zero Trust Remote Access Platform for DevOps

Accelerating time to value, and realizing efficient, safe remote operations for DevOps.

By Marc Hornbeek
Engineering DevOps Consulting
www.engineeringdevops.com

Introduction

DevOps continuous delivery is a central and essential part of enterprise digital transformation strategies. Working remotely has become the normal working practice for many enterprise DevOps team members. Traditional remote access and remote access security controls are especially inefficient for both DevOps workers and administrators that support them because of the dynamic nature of services, team members, and environments. A ***Zero Trust Remote Access platform*** offers a solution.

This white paper explains:

- "Why is a Zero Trust Remote Access platform needed for DevOps?"
- "How does a Zero Trust Remote Access platform work with DevOps?"
- "What does an organization need to do to transform to a Zero Trust Remote Access platform for DevOps?"

Why is a Zero Trust Remote Access platform needed for DevOps?

There are multiple reasons why a Zero Trust Remote Access platform for DevOps is needed. The article "A New Remote Zero Trust Platform is Needed" https://securityboulevard.com/2021/07/a-new-remote-zero-trust-platform-is-needed/ explained reasons in high level terms.

Every DevOps team member requires access to numerous systems and services that make up workflows and toolchains for one or more DevOps value streams, as illustrated in Figure 1. The following is a typical list of tools, systems, and environments for which access is required:

- Communication systems to enable collaboration and sharing information with other team members in development, QA, security, operations, and management roles.
- Access to multiple DevOps environments.
- Feature and Agile backlog planning and management tools.
- Software source code version management tools.
- Artifact Repositories.
- Integrated Development Environments for designing and testing software.
- CI/CD pipeline automation platform tools such as Jenkins and Azure DevOps.

- Development test environments which may require access to complex topologies of interconnected systems and tools.
- Staging test environments which may also require access to complex topologies of interconnected systems and tools.
- Quality and Security test tools such as scanners, manual and automated test tools.
- Continuous Delivery and Deployment tools.
- Monitoring tools to check functional and performance characteristics of application changes.
- Ticket systems for reporting and tracking requests and problems.

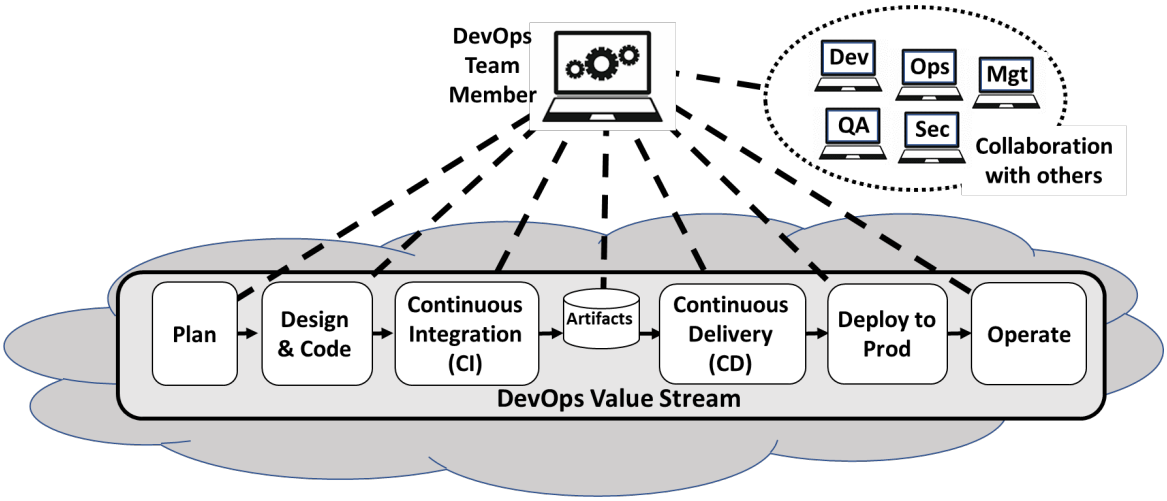

Figure 1: DevOps Team Member Access Requirements

Traditional remote access configurations are not well-suited for DevOps given a shifting user population comprised of employees, contractors, and other third parties often using their own devices; the large number of tools and systems to which access must be provided; and the hybrid nature of the infrastructure itself (on-premises, hybrid- and multi-cloud). As indicated in Figure 2, there are three types of pain points associated with traditional approaches when used with DevOps.

1. DevOps user experience is poor.
2. Management of DevOps user environments is complex.
3. Security for the large number of dynamic DevOps services and users is complex.

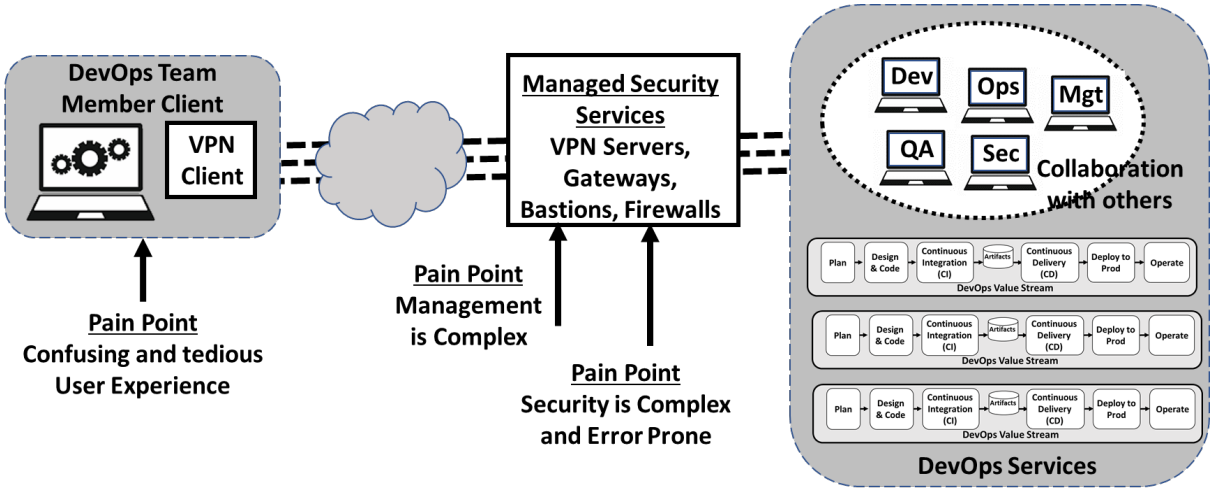

Figure 2: Traditional DevOps Remote Access Pain Points

User Experience is Poor in traditional remote access configurations. Each DevOps service must be accessed as separate secured sessions. Given the large number of systems and services used by DevOps team members the user experience is confusing and tedious.

Management is Complex, in traditional remote access configurations. System management requires IP whitelisting rules in VPN, static SSH keys in Bastion, and firewall rules controlled by different teams, and application-specific authentication. The coordination and execution of so many touch points for common actions including the onboarding of new team members, changing roles, or adding a new service invites mistakes that cause productivity issues and introduces unforeseen security holes. Each employee or service to be added generates a multitude of tasks across different teams with each task having multiple actions (E.g., new ticket, provision/de-provision access, end-user setup and typically some troubleshooting). While the actions are simple to perform, significant time is spent waiting on cross-functional teams which can take days.

Security is Complex and Error Prone, in traditional remote access configurations. DevOps users must be granted restricted access to the individual services they need to be productive, rather than overly-broad access to entire network segments. Administering restricted access for individual and groups takes time and care. Security policies typically require monitoring detailed audit logs of user accesses. Revoking access or adjusting access rights according to policy changes is time consuming and error prone.

How does a Zero Trust Remote Access platform work with DevOps?

Figure 3 illustrates how a Zero Trust Remote Access platform can resolve the requirements for improved DevOps user experience, consistent and manageable administration, and simplified, less error-prone security.

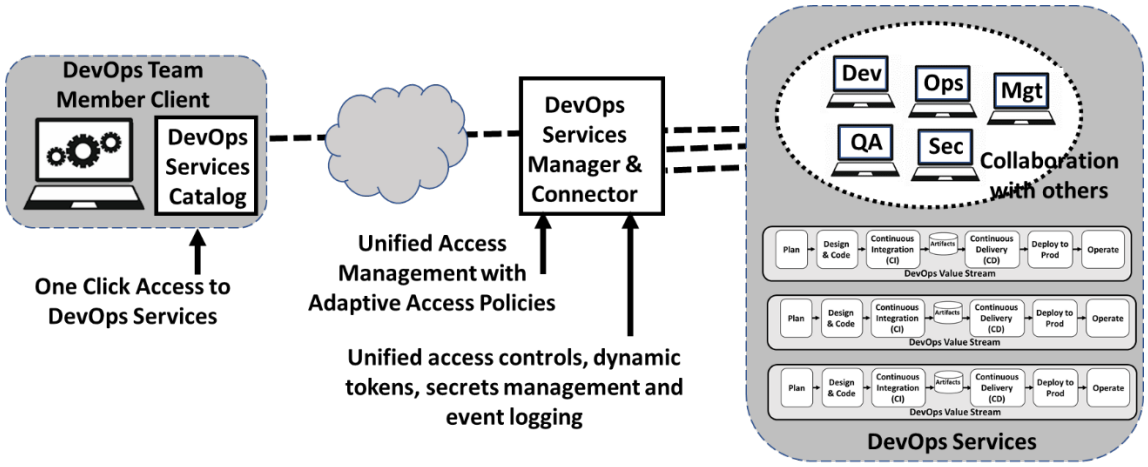

Figure 3: Zero Trust Remote Access Platform For DevOps

Improved DevOps User Experience

By replacing the need for the user to tediously set up separate sessions for each DevOps service they need to use, a client-side DevOps service catalog enables all DevOps services to be accessed with a single click. The example by Banyan Security is shown in Figure 4. This service catalog approach is not only more convenient for DevOps users, but it also provides a form of DevOps governance because it encourages DevOps teams to select choices that are available in the catalog instead of creating new choices which results in expensive "DevOps tool sprawl".

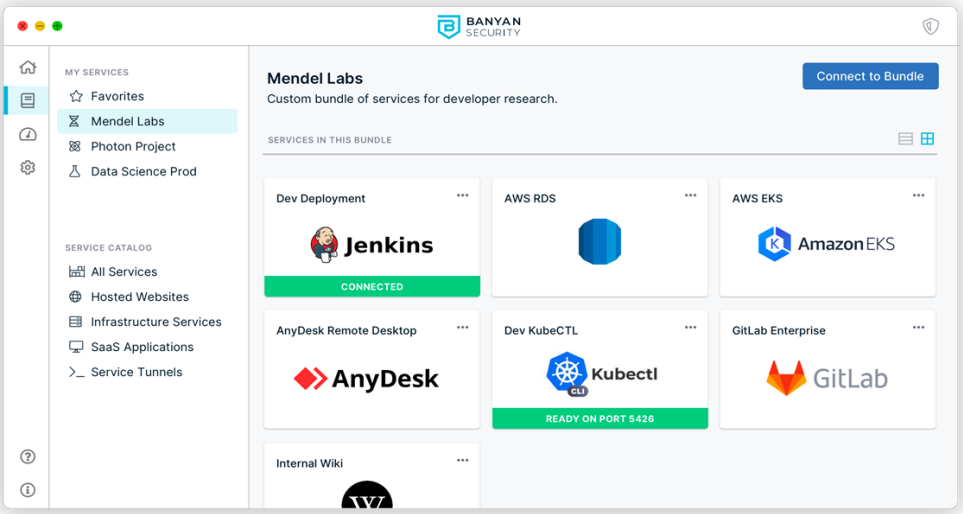

Figure 4: DevOps Client Service Catalog
(Credit: Banyan Security)

Simplified DevOps Management

Management of remote DevOps users is simplified by consolidating access for all DevOps users and services under a single unified service manager and connector. Table 1 illustrates that the need to manage each user and service with different teams, each with their own processes, is unified and streamlined when the management requirements for DevOps users and services are consolidated under a common Zero Trust Remote Access platform. In addition, this approach makes it much each easier to implement more adaptive policies across the breadth of users and services.

Onboarding requirements for new DevOps users typically involves setting up IDP groups and roles, VPN access, Bastion access and application authentication, while new DevOps services often require modification/creation of Firewall rules, VPN access, application authentication, and server authentication. However, with a Zero Trust Remote Access platform this is accomplished within a single session – not separate sessions for each onboarding requirement.

Management Tasks	Traditional Approach	Zero Trust Remote Access Platform
IDP Group & Role	Identity Team setup per DevOps user	Simplified DevOps user setup process
VPN access	Network Sec Team setup per user and DevOps service	Simplified DevOps user setup process
Bastion access	DevOps Team per DevOps user and DevOps service	Simplified DevOps user setup process
Application authentication	Application Team setup per DevOps user and Application	Simplified DevOps user setup process
Firewall rule	Firewall Team per DevOps service	Simplified DevOps service setup
Application authentication	Application Team setup per Application	Simplified DevOps service setup
Server authentication	Server Team setup per service	Simplified DevOps service setup

Table 1: Simplified management for Zero Trust Remote Access

Simplified and Less Error-Prone Security

Consolidating access to DevOps services under a common Zero Trust Remote Access platform with unified services and services connectors, enables simplified and less error prone security controls, improved security monitoring and policy management. Audit logs of accesses are reported consistently for all services. Monitoring and changing access rules can be supported for individuals or group, when

adjusting a policy. Trust-Based Access Controls can be implemented consistently. Granular policies ensure access is predicated on appropriate device Trust Score thresholds being met. Rather than a single check prior to granting access, these production services policies can continuously assess the security of the users and devices accessing the services.

What does an organization need to do to transform to a Zero Trust Remote Access platform for DevOps?

Transformations of all types, and especially those for DevOps, are made more certain by following a Seven-Step Transformation process, described in "Engineering DevOps" by Marc Hornbeek. This process serves as a framework for implementing the *Zero Trust Remote Access platform.*

The first two steps "***Vision and Team Alignment***" involves getting the leaders, key influencers, and implementers of an organization to recognize and establish actionable goals for the implementation of the Zero Trust Remote Access platform. This is supported by emphasizing the benefits of the approach which were described earlier in this paper.

The third step, "**Assessment**", requires taking inventory of the current state of DevOps users, DevOps services and policies that need to be supported by the new platform.

The fourth step, "***RoadMap***" requires determining the plan for implementation. Typically, it makes sense to start with one application. Start with a set of DevOps users and DevOps services needed for a single application or pipeline which can serve as a model to drive adoption to other applications as the solution is proved and accepted.

The fifth step, "***Realize***" is the step in which the *Zero Trust Remote Access platform* is installed and setup for the initial model application or pipeline. An article "Experience Zero Trust Network Access with Banyan Security Test Drive" https://securityboulevard.com/2021/04/experience-zero-trust-network-access-ztna-with-banyan-security-test-drive/ explains how tool vendors help to simplify this step. Capabilities to look in for in tools include:

- Deploys in minutes, with no integration requirements on any existing DevOps services or infrastructure.
- No need to setup VPNs, open inbound firewall ports, or manage DNS.
- Use existing DevOps user devices without diminishing security.
- DevOps Service Catalog provides users visibility and one-click access to DevOps services.
- Support for bundled services, favorites, and autorun capabilities.

In step six, "***Operationalize***" the procedures for commissioning and using the *Zero Trust Remote Access platform* are proven, benefits are measured, and playbooks are hardened for the organization.

In step seven, "**Expansion**" the *Zero Trust Remote Access platform* deployment is expanded to other DevOps applications pipelines, DevOps users, and DevOps services across the enterprise.

What This Means

Traditional remote access configurations are not well suited for DevOps administrators or users given the dynamic nature of services, team members, and environments. A ***Zero Trust Remote Access platform*** addresses the need for an improved DevOps admin and user experience, simplified access management to DevOps services, and simpler, less error-prone Security practices and monitoring. Benefits include:

- Setup and management of zero trust access to DevOps services and infrastructures does not require integration or cross-functional work.
- Onboarding new DevOps users and DevOps services and infrastructure for remote access is simplified.
- DevOps users can view and access all of the DevOps services and infrastructure with a single click.
- No need to setup and support complex VPNs.
- SSH key management using short-lived certificates is easy to deploy, renew, and revoke access.
- New infrastructure services and hosted applications can be made available in minutes rather than days.
- Security is improved with trust-based policy control, least privilege access, and continuous authorization ensuring end user productivity while providing admin visibility, control, and security.

I want to express my appreciation for time spent with the Banyan Security team, who has pioneered many of the concepts outlined in this paper, and contributed valuable insight. This paper is reproduced by Banyan Security with my permission.

References

- Book “Engineering DevOps”, 2019, Marc Hornbeek
- https://securityboulevard.com/2021/07/a-new-remote-zero-trust-platform-is-needed/
- https://securityintelligence.com/articles/vpn-zero-trust-best-for-remote-working-security/
- https://securityboulevard.com/2021/04/experience-zero-trust-network-access-ztna-with-banyan-security-test-drive/

About the Author

Marc Hornbeek, a.k.a., DevOps-the-Gray esq. is a globally recognized expert for **Software Testing, DevOps**, **DevSecOp**s, and **SRE**. He is **CEO and Principal Consultant** at **Engineering DevOps Consulting** , **author** of the book ***Engineering DevOps*** , **Analyst** for the **Accelerated Strategy Group** and **Ambassador and Author for** The **DevOps Institute** .