Hype Cycle for Infrastructure Platforms, 2023

Published 12 July 2023 - ID G00791914 - 76 min read

By Analyst(s): Dennis Smith

Infrastructure platforms support the combining of agility and speed with the safety and soundness needed for enabling digital business strategies. I&O leaders should leverage this Hype Cycle to prepare for delivery of impactful infrastructure platform capabilities.

More on This Topic

This is part of an in-depth collection of research. See the collection:

- 2023 Hype Cycles: Deglobalization, AI at the Cusp and Operational Sustainability

Strategic Planning Assumptions

By 2027, platform engineering principles will influence more than 50% of infrastructure and operations technology decisions, which is a substantial increase from less than 20% today.

By 2027, more than 75% of global enterprises will have formal infrastructure platform organizations, up from less than 20% in 2023.

Analysis

What You Need to Know

Infrastructure platforms are foundational to the ability of infrastructure and operations (I&O) organizations to support the increasing demands of digital business. Infrastructure platforms affect roles, skills and expectations of I&O staff and leaders, and improve end-user engagement, knowledge and experience.

I&O leaders responsible for infrastructure platforms must:

- Continually engage with the infrastructure platform consumers as you evolve and enhance the offered platform.
- Assign a platform owner that ensures that consistent alignment and objectives culminates in successful infrastructure platform delivery.
- Enable self-service capabilities that satisfy the needs of the different personas using infrastructure services within your environment.
- Provide API-enabled platforms that enable extensibility and access to external services.
- Position the platform organization to curate infrastructure while providing added value to users.
- Prepare staffing requirements to meet the needs to build and operate the infrastructure platforms.

The Hype Cycle

Infrastructure platforms are key to I&O's ability to deliver value while managing cost and risk. I&O leaders must view infrastructure platforms as an enabler for the overarching trends of DevOps, container, hybrid cloud computing, edge computing, communications infrastructure applications and platform engineering-driven approaches to delivering business solutions.

The four forces that are primarily responsible for driving the growth of I&O infrastructure platforms are:

- The need for agile and resilient technological solutions.
- Adoption of cloud-native technology.

- Platform engineering as a key enabler for modernizing IT infrastructure.
- Siloed legacy infrastructure tools providing marginal value.
- The need to abstract infrastructure solutions and reduce cognitive load on consumers.

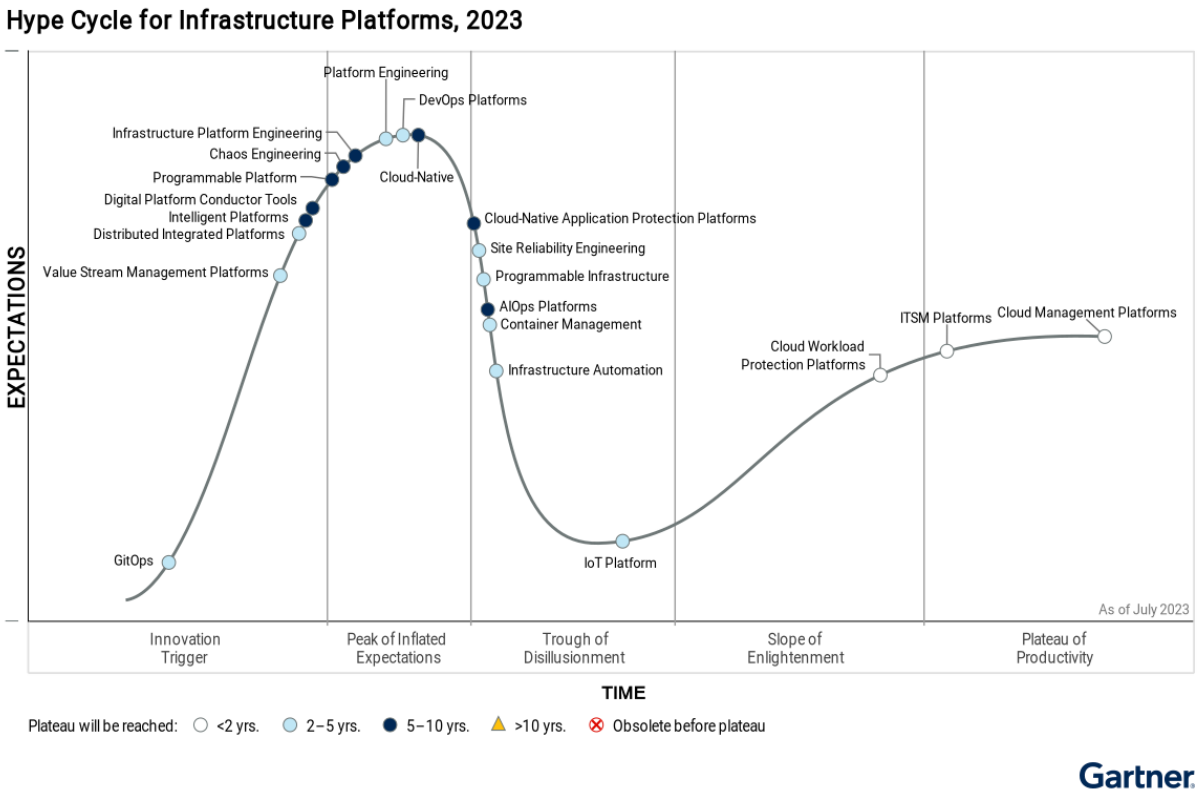
Business pressures (including speed to market and cost efficiencies) continue to place additional challenges on I&O teams. Those teams must provide solutions that combine the safety and soundness that I&O traditionally has stressed, and the agility and speed desired to meet these business challenges.

The adoption of composable and programmable infrastructures has embedded an API-driven approach to managing infrastructure in support of modern applications and services. Additional business pressure has increased the need for I&O solutions to support these initiatives.

Platform teams continue to gain traction among Gartner clients, embedding a product focus (centered around infrastructure platforms) to the delivery of services. I&O leaders must embrace agile methods of working and practices that improve capability, efficiency, resiliency, customer centricity and innovation in delivery to make transformation efforts successful, reliable and sustainable.

Through the years, I&O has deployed siloed tools that have delivered marginal value. As enterprises rationalize these investments, there is a continual push to seek tooling that might be able to derive some value from these disparate legacy tools.

Figure 1: Hype Cycle for Infrastructure Platforms, 2023



The Priority Matrix

The Priority Matrix maps the time to maturity of a technology/framework on a grid in an easy-to-read format. It answers two high-priority questions:

1. How much value will an organization receive from an innovation?

2. When will the innovation be mature enough to provide this value?

During the next two to five years, anticipate continual evolution of second-generation infrastructure platforms that are extensible, API-driven and centered around automation for tasks like remediation. Many of the platforms currently emerging will have evolved and matured to provide tangible enterprise value. This will include an increased focus on gaining value from legacy infrastructure investments.

Many later-stage infrastructure platforms will, in many cases, change their focus, often realigning to a subset of services. These offerings will also have reengineered capabilities, incorporating attributes that are in the second-generation infrastructure platforms.

The key focus will be on achieving tangible value from infrastructure resources. This includes continuing to derive value from legacy tools and also enabling end users to track and measure infrastructure value.

Table 1: Priority Matrix for Infrastructure Platforms, 2023

(Enlarged table in Appendix)

Benefit	*Years to Mainstream Adoption*			
↓	*Less Than 2 Years* ↓	*2 - 5 Years* ↓	*5 - 10 Years* ↓	*More Than 10 Years* ↓
Transformational		Platform Engineering Site Reliability Engineering	Digital Platform Conductor Tools	
High	ITSM Platforms	Container Management DevOps Platforms Distributed Integrated Platforms GitOps Infrastructure Automation IoT Platform Programmable Infrastructure Value Stream Management Platforms	AIOps Platforms Cloud-Native Application Protection Platforms Infrastructure Platform Engineering Intelligent Platforms Programmable Platform	
Moderate	Cloud Workload Protection Platforms		Chaos Engineering	
Low	Cloud Management Platforms			

Source: Gartner (July 2023)

On the Rise

GitOps

Analysis By: Paul Delory, Arun Chandrasekaran

Benefit Rating: High

Market Penetration: 5% to 20% of target audience

Maturity: Early mainstream

Definition:

GitOps is a type of closed-loop control system for cloud-native applications. The term is often used more expansively, usually as a shorthand for automated operations or CI/CD, but this is incorrect. According to the canonical OpenGitOps standard, the state of any system managed by GitOps must be: (1) expressed declaratively, (2) versioned and immutable, (3) pulled automatically, and (4) continuously reconciled. These ideas are not new, but new tools and practices now bring GitOps within reach.

Why This Is Important

GitOps can be transformative. GitOps workflows deploy a verified and traceable configuration (such as a container definition) into a runtime environment, bringing code to production with only a Git pull request. All changes flow through Git, where they are version-controlled, immutable and auditable. Developers interact only with Git, using abstract, declarative logic. GitOps extends a common control plane across Kubernetes (K8s) clusters, which is increasingly important as clusters proliferate.

Business Impact

By operationalizing infrastructure as code, GitOps enhances availability and resilience of services:

- GitOps can be used to improve version control, automation, consistency, collaboration and compliance.
- Artifacts are reusable and can be modularized.
- Configuration of clusters or systems can be updated dynamically. All of this translates to business agility and a faster time to market.

- GitOps artifacts are version-controlled and stored in a central repository, making them easy to verify and audit.

Drivers

- **Kubernetes adoption and maturity:** GitOps must be underpinned by an ecosystem of technologies, including tools for automation, infrastructure as code, continuous integration/continuous deployment (CI/CD), observability and compliance. Kubernetes has emerged as a common substrate for cloud-native applications. This provides a ready-made foundation for GitOps. As Kubernetes adoption grows within the enterprise, so can GitOps, too.
- **Need for increased speed and agility:** Speed and agility of software delivery are critical metrics that CIOs care about. As a result, IT organizations are pursuing better collaboration between infrastructure and operations (I&O) and development teams to drive shorter development cycles, faster delivery and increased deployment frequency. This will enable organizations to respond immediately to market changes, handle workload failures better, and tap into new market opportunities. GitOps is the latest way to drive this type of cross-team collaboration.
- **Need for increased reliability:** Speed without reliability is useless. The key to increased software quality is effective governance, accountability, collaboration and automation. GitOps can enable this through transparent processes and common workflows across development and I&O teams. Automated change management helps to avoid costly human errors that can result in poor software quality and downtime.
- **Talent retention:** Organizations adopting GitOps have an opportunity to upskill existing staff for more automation- and code-oriented I&O roles. This opens up opportunities for staff to learn new skills and technologies, resulting in higher employee satisfaction and retention.
- **Cultural change:** By breaking down organizational silos, development and operations leaders can build cross-functional knowledge and collaboration skills across their teams to enable them to work effectively across boundaries.
- **Cost reduction:** Automation of infrastructure eliminates manual tasks and rework, improving productivity and reducing downtimes, both of which can contribute to cost reduction.

Obstacles

- **Prerequisites:** GitOps is only for cloud-native applications. Many GitOps tools and techniques assume the system is built on Kubernetes (frequently, they also assume that a host of other technologies are built on top of K8s). By definition, GitOps requires software agents to act as listeners for changes and help to implement them. GitOps is possible outside Kubernetes, but in practice K8s will almost certainly be used. Thus, GitOps is necessarily limited in scope.
- **Cultural change:** GitOps requires a cultural change that organizations need to invest in. IT leaders need to embrace process change. This requires discipline and commitment from all participants to doing things in a new way.
- **Skills gaps:** GitOps requires automation and software development skills, which many I&O teams lack.
- **Organizational inertia:** GitOps requires collaboration among different teams. This requires trust among these teams for GitOps to be successful.

User Recommendations

- **Target cloud-native workloads initially:** Your first use case for GitOps should be operating a containerized, cloud-native application that is already using both Kubernetes and a continuous delivery platform such as Flux or ArgoCD.
- **Build an internal operating platform:** This is the foundation of your GitOps efforts. Your platform should manage the underlying infrastructure and deployment pipelines, while enforcing security and policy compliance.
- **Embed security into GitOps workflows:** Security teams need to shift left, so the organization can build holistic CI/CD pipelines that deliver software and configure infrastructure, with security embedded in every layer.
- **Be wary of vendors trying to sell you GitOps:** GitOps isn't a product you can buy, but a workflow and a mindset shift that becomes part of your overall DevOps culture. Tools that expressly enable GitOps can be helpful; but GitOps can be done with nothing more than standard continuous delivery tools that support Git-based automation.

Sample Vendors

GitLab; Harness; Red Hat; Upbound; Weaveworks

Gartner Recommended Reading

Innovation Insight: Top 4 Use Cases for GitOps

Is Using GitOps-Style Automation With Kubernetes Right for Me?

How to Scale DevOps Workflows in Multicluster Kubernetes Environments

Designing and Operating DevOps Workflows to Deploy Containerized Applications With Kubernetes

Automate the Application Delivery Value Stream

Value Stream Management Platforms

Analysis By: Hassan Ennaciri, Akis Sklavounakis

Benefit Rating: High

Market Penetration: 5% to 20% of target audience

Maturity: Early mainstream

Definition:

A value stream management platform (VSMP) is a platform that seeks to optimize end-to-end product delivery and improve business outcomes. VSMPs are typically tool-agnostic. They connect to existing tools and ingest data from all phases of software product delivery — from customers' needs to value delivery. VSMPs help software engineering leaders identify and quantify opportunities to improve software product performance by optimizing cost, operating models, technology and processes.

Why This Is Important

As organizations scale their agile and DevOps practices, higher-level metrics that assess performance and efficiency of their product delivery are essential. VSMPs integrate with multiple data sources to provide DevOps-related telemetry. These insights enable stakeholders to make data-driven decisions in an agile manner and correct course as needed. The visualization capabilities of VSMPs help product teams analyze customer value metrics against the cost required to deliver that value.

Business Impact

VSMPs help organizations bridge the gap between business and IT by enabling stakeholders to align their priorities to focus on delivering customer value. VSMPs can provide CxOs with strategic views of product delivery health and pipelines, allowing them to make data-driven decisions about future product investments. These platforms also provide product teams with end-to-end visibility and insight into the flow of work to help them address constraints and improve delivery.

Drivers

- Improved software delivery with business priorities and objectives.
- Timely decision making driven by insights from data.
- Optimization of delivery flow through reduction of waste and elimination of bottlenecks.
- Visibility and mapping of end-to-end software delivery processes and identification of cross-team dependencies.
- Quality and velocity improvements of product deployments.
- More stringent governance, security and compliance requirements.

Obstacles

- VSMPs are not focused on continuous integration/continuous delivery (CI/CD) capabilities. Execution of the delivery pipeline requires use of a custom toolchain or DevOps platform.
- VSMPs require customization and data from tools used by multiple stakeholders in the organization, sometimes outside of software delivery. Collaboration with these key stakeholders to deliver the desired insights is paramount.
- VSMPs are still evolving and not all vendors have all the core capabilities.

User Recommendations

- Accelerate business outcomes by leveraging real-time, data-driven metrics and value stream insights provided by VSMPs.
- Leverage VSMPs' AI-powered analytics and insights to surface constraints, detect bottlenecks and improve flow.
- Build customized dashboards and views of product delivery for multiple stakeholders and leadership.
- Utilize VSMPs to assess the performance, quality and value of products, including development costs and ROI.
- Use VSMPs to gain a consolidated view of governance, security and compliance across all product lines.

Sample Vendors

Broadcom; ConnectALL; Digital.ai; HCLSoftware; IBM; OpenText; Opsera; Planview; Plutora; ServiceNow

Gartner Recommended Reading

Market Guide for Value Stream Management Platforms

Tools for Delivering Business Metrics to Software Engineering Teams

Market Guide for Value Stream Delivery Platforms

Use the Right Metrics in the Right Way for Enterprise Agile Delivery

Digital Platform Conductor Tools

Analysis By: Roger Williams, Dennis Smith

Benefit Rating: Transformational

Market Penetration: Less than 1% of target audience

Maturity: Embryonic

Definition:

Digital platform conductor (DPC) tools coordinate the various infrastructure tools used to plan, implement, operate and monitor underpinning technology and services for applications and digital products. They enable digital business, regardless of the environments used or who owns them. DPC tools provide a unified view of underpinning technologies and their connection to applications. This augments strategic decision making and improves the value obtained from technology investments.

Why This Is Important

Traditional, cloud and hybrid infrastructure management tools do not inherently provide an integrated view of infrastructure across all environments. Moreover, as infrastructure and operations (I&O) leaders struggle to manage their portfolio of investments to enable composable business, optimize costs and reduce risks, they need help with filling the gaps in visibility, assurance and coordination. DPC tools promise to help close these capability gaps and are improving in their ability to do so.

Business Impact

DPC tools deliver the following benefits not inherent in more focused infrastructure management toolsets:

- Visualizing digital platform performance across all life cycle stages — planning, implementing, operating and monitoring.
- Enabling continual optimal performance and placement of workloads in all environments — on-premises, in the cloud or at the edge.
- Ensuring tangible business value from improvement efforts across all technology architectures — compute, storage, middleware and network layers.

Drivers

- Difficulty in maintaining a coherent view of all technology infrastructure resources and their dependencies that are aligned with changes to services, applications and components, as well as the configuration of their promised performance levels.
- Lack of transparency into spending on hybrid digital infrastructure and how resource capacity aligns with actual application workload demand.
- Need to guide where workloads are processed (data center, public cloud, colocation facility, etc.) based on requirements, including capacity, cost and dependency dynamics.
- Challenges with estimating the value, efficiency, quality and compliance delivered by hybrid digital infrastructure based on aggregated data from performance analysis tools and other hybrid digital infrastructure management (HDIM) toolset data feeds.
- Desire for a single point of entry and reporting for digital platform resource requests, and routing them to appropriate HDIM tooling for fulfillment.
- Desire to reduce the level of skills and effort required within initiatives to improve operations and digital employee experiences.
- Gaps, duplication and conflicts in data to support application workload migration and business continuity goals, as well as protection of data from accidental deletion or malicious activities.
- Inability to confirm compliance of application workloads and digital platforms to identity requirements and security baselines as part of the organization's cybersecurity mesh approach.
- Poor credibility of business cases for digital platform improvements, including: assessing business impact; measuring gaps between current and desired performance; providing oversight of improvement efforts; and validating benefits delivered.

Obstacles

- Lack of interoperability: Tool sprawl and difficulties in integration inhibit DPC tool adoption. The technology landscape is littered with failed approaches that were intended to support data sharing between vendors.
- Lack of data credibility: The desire for a complete, accurate view of all technology as a precondition for decision making has been around for decades, yet is no closer to being realized. Customers that demand perfect data before they act, and vendors that require complete and accurate data for their tools to function properly, will continue to co-create expectations that will not be met.
- Lack of budget: DPC tools may be viewed as “overhead” that does not have a compelling business case. No one likes paying for something that does not appear to address specific pain points felt today.
- Lack of vendor commitment: Many vendors will be tempted to “DPC wash” their existing offerings and claim that these capabilities are already addressed or can be added for very little cost.

User Recommendations

- Build a DPC tooling strategy that supports digital business ambitions by defining the management elements, environments and technology layers required to meet the organization’s infrastructure needs now and in the future.
- Address measurement and coordination gaps by working with key stakeholders to identify infrastructure value and risk and cost objectives, and by making targeted investments in integration, dependency mapping and continuous improvement capabilities.
- Plan for DPC tooling investments by determining which DPC capability aspects are needed in the short, medium and long term. Compare these capabilities to current and future vendor offerings for infrastructure management tooling that can provide initial DPC tool functionality.
- Ensure that DPC tooling investments can deliver sustained value by requiring that DPC tool marketers show how the tool will address current organizational pain points and how it will adapt to future needs as organizational requirements evolve.

Sample Vendors

Cloudsoft; Flexera; HCLTech; IBM (Turbonomic); Oomnitza; OpsRamp; ReadyWorks; Snow Software; Virtana

Gartner Recommended Reading

Market Guide for Digital Platform Conductor Tools

3 Steps to Improve the Reliability of Large, Complex and Distributed IT Systems by Leveraging SRE Principles

Distributed Integrated Platforms

Analysis By: Philip Dawson

Benefit Rating: High

Market Penetration: 5% to 20% of target audience

Maturity: Adolescent

Definition:

Distributed integrated platforms are infrastructure and application platforms deployed in data centers that have infrastructure platforms and solutions or applications integrated, marketed and supported by a primary vendor or provider.

Why This Is Important

Cloud platform delivery is now commonplace in a majority of clients and this is driving a competitive response with server vendors delivering software platform integration as a packaged platform as a service (PaaS). Distributed integrated platforms in data centers are architected, funded, packaged and controlled or managed by the application or cloud platform provider and their partners. This allows new application and cloud infrastructure experience for data center delivery.

Business Impact

Distributed integrated platforms combine application and data layers of integrated infrastructure. This packaging and consumption is similar to integrated systems with more cloudlike elasticity — for both the infrastructure and application across everything as a service (XaaS). These offerings use a control plane or management console aligned with the application platform provider. This may include infrastructure and software platform or application packaging, measurement and chargeback.

Drivers

- On-premises distributed integrated platforms are different from distributed integrated infrastructure as they are dedicated solutions in a similar way to that of traditional integrated systems and integrated platforms whereas integrated infrastructure is for general-purpose applications integrated separately.
- The on-premises element can alleviate the data proximity, data residency and location-based network issues for providers, the control plane audit and compliance capabilities, trade risk and responsibility for remote governance by the application platform vendor. This control shift to the application provider is inhibitive to the overall adoption of a remotely managed solution in an on-premises environment to see both sides of the trade-off.
- Adoption is also more likely where data center staff lack deep integration support skills or where I&O teams wish to focus solely on application delivery rather than infrastructure services.
- Distributed integrated platforms take the packaging of data platforms and/or application platforms on top of the distributed infrastructure. This allows for improved management and operational efficiency as part of the overall integration with the application vendor on their own infrastructure.
- Distributed integrated platforms allow life cycles to be aligned between infrastructure projects and applications services and technical debt to be reduced across a single platform. This is delivered across a common on-premises cloud platform.

Obstacles

- While the primary PaaS stack and application are more integrated with distributed integrated platforms, third-party solutions and software PaaS offerings are more difficult to integrate on top of the infrastructure and platform. Third-party support can be exposed for application integration and transformation is at the provider's pace not the user's.
- Common infrastructure for cloud delivery models makes sense, but alignment to a vendor's offering can create cloud silos tied to vendors that are difficult to integrate into hybrid or multicloud environments, but technical debt overall is suboptimized and reduced.
- While repackaging on-premises platforms to cloudlike delivery, avoid the same vendor increased lock-in and potential issues around forecast consumption, licensing, and platform delivery quotas and compliance.
- This type of integrated platform does reduce technical debt overall but investment and costs increase to maintain the current nature of cloud platforms.

User Recommendations

- Look at the realistic benefits of a common, shared, modernized infrastructure and application platforms that distributed integrated platforms deliver instead of getting bogged down with taxonomy.
- Prioritize the integration with third-party applications and software platforms. Don't overinvest in a single vendor's infrastructure and application management and tooling.
- Evaluate the strength of distributed integrated platforms and look for true partnerships and integration in geographic and vertical routes to market.
- Manage a reduction of technical debt by investing in cloud and consumption models absorbing costs as part of XaaS and platform delivery.

Sample Vendors

Amazon Web Services (AWS); Google; IBM; Microsoft; Oracle

Gartner Recommended Reading

Quick Answer: How Can I Optimize the Use of Programmable Platforms for Effective Software Delivery?

Rationalizing Applications and Infrastructure for Cloud Delivery

How Do I Plan for Migrating My Data Center Infrastructure Into an XaaS Model?

Intelligent Platforms

Analysis By: Philip Dawson, Nathan Hill

Benefit Rating: High

Market Penetration: 5% to 20% of target audience

Maturity: Emerging

Definition:

Intelligent platforms provide the administration composability of infrastructure and programmable API functions with automated infrastructure intelligence. They integrate compute, storage and networking assets with some or the entire application software stack, creating dedicated workload architectures. Intelligent platform vendors also include components such as application intimacy, management tools, OSs, and virtualization bought and/or consumed as a service.

Why This Is Important

Intelligent platform solutions are differentiated against integrated system or hyperconverged infrastructure (HCI) solutions with a separate software stack purchase tied to the hardware. Pricing strategies vary greatly throughout the integrated software stack solutions as part of the shift to consumption-based infrastructure delivery. Intelligent platforms also integrate applications and business logic as bundles and partnerships.

Business Impact

Intelligent platforms optimize:

- Delivery of workload performance or application manageability that crosses over from hardware that promises lower operational costs and increased IT agility via automated, pooled resources.
- Automation and machine learning of complete stacks, hardware administration and software programmability on top of integrated systems.

- They are stand-alone running proprietary workloads that rarely compete with each other as the software stacks set the hardware options.

Drivers

- The intelligent platform market is influenced by multiple aspects of resilience and availability across on-premises, hosting or colocation and cloud locations driving composable, programmable and intelligent functions.
- Intelligent platforms are integrated as everything as a service (XaaS), with automation and management, and differ from integrated stack systems, which are hardware-integrated dedicated appliances.
- Multiple vendors are driving the market for intelligent platforms around integrated systems, HCI, cloud and virtualization. Intelligent platforms are built from a software perspective on top of HCI rather than a traditional integrated stack system that is built as a hardware appliance around hyperconverged integrated systems (HCIS).
- Vendors such as Microsoft, Nutanix and VMware are promoting valid intelligent platform software, and the market momentum around HCI software in the cloud now creates a market for multiple hardware vendors to build software management and integration services.

Obstacles

- Hybrid and multicloud strategies may not integrate well with integrated platforms, continuing the silo mentality of cloudlike delivery.
- Other platform as a service (PaaS) momentum is being integrated from packaged vendors such as SAP and Oracle, which are bundling integrated stack systems and distributed cloud infrastructure with application platform and database management system (DBMS) software. Here, the intelligence is with the PaaS software, not the intelligent infrastructure.
- An intelligent platform provides balanced XaaS workload performance, application optimization and integration, but this comes at the expense of greater vendor dependency, and inflexibility for future application customization and workload requirements.

User Recommendations

- Select infrastructure software management frameworks for overlay management as well as links to cloud infrastructure. Do not implement hardware-dependent or locked-in intelligent platform frameworks and adapters.
- Define successful intelligent platform implementations by assessing data center stakeholders and other vested interests (for example, procurement) with other lines of business responsible for agreeing with SLAs.
- Automate the infrastructure requirements for cloud management platforms (CMPs) through the use of intelligent platforms as you deliver XaaS through infrastructure platforms.

Sample Vendors

CU Coding; DataDirect Networks (DDN); Dell Technologies; Hewlett Packard Enterprise; Microsoft; Nutanix; Oracle; VMware

Gartner Recommended Reading

How to Evolve Your Physical Data Center to a Modern Operating Model

Quick Answer: How Can I Optimize the Use of Programmable Platforms for Effective Software Delivery?

How Do I Plan for Migrating My Data Center Infrastructure Into an XaaS Model?

At the Peak

Programmable Platform

Analysis By: Philip Dawson, Bill Blosen

Benefit Rating: High

Market Penetration: 1% to 5% of target audience

Maturity: Emerging

Definition:

Programmable platforms are API-driven for delivery of applications in a cloud model by using and applying methods and tooling from the software development area to management of IT infrastructure and platform concepts. It includes resilient platform architectures and agile techniques.

Why This Is Important

Software engineering and infrastructure and operations (I&O) have coexisted but been separated by their differing toolsets and APIs. Modern digital businesses need their software engineering and I&O teams to deliver a cohesive platform that encompasses both application and infrastructure delivery. Programmable infrastructure delivers the underlying technical capability that enables this integration.

Business Impact

Greater value and agility (rather than cost optimization) is achieved via programmable platforms' ability to drive adaptive application delivery. Programmable infrastructure, API provisioning and automated processes allow faster responses to new business demands, driving service quality and freeing application delivery staff and administration staff from infrastructure functions. Programmable platforms enable a sustainable and highly responsive IT infrastructure service to the business.

Drivers

- Software architects and engineers are moving to modular distributed applications built on containerization, control, data and service architecture. In essence, they use the pattern of separating the application front end from the business logic back end.
- API layers are being adopted through self-service capabilities and organizing the programmable platform APIs into paved roads.

- Software engineers are reducing the cognitive load of using APIs and programmable platforms, improving developer experience, driving productivity and retention of key talent, and also improving adherence to architectural and security guardrails.
- I&O teams are moving workloads and application delivery to cloud infrastructure and platforms as in anything-as-a-service (XaaS) models. In essence, they have embraced programmable infrastructure, that is, applying software development methods, APIs and tooling to manage the control and data planes around I&O services.
- The incumbent architecture of programmable platforms is deploying modular building blocks. Updates to modules are automatically rolled out to any platform that is currently using that module rather than updating each platform discreetly, leveraging efficiency, scale and management dependencies.
- Platform engineering principles guide the programmable platform buildout using templates, APIs and automation to simplify the usage and adoption of programmable infrastructure. The platforms being built must respond to the developers pain points and ease adoption of the most used functions. Agile product ownership is the best model to build feedback loops between the developer programmable platform teams and the platform's users and communities.

Obstacles

- The boundaries between programmable platforms and infrastructure are at best emerging. I&O teams must be conscious of how their architectures and deployments interface with the programmable infrastructure at the control plane and data plane.
- The architecture and order of APIs across and up the stack and across layers need planning and integration, increasing lock-in. I&O and software teams conceptualize these layers as part of one programmable platform with four distinct layers: the application presentation front end, the business logic back-end functions, the control or management, and data tier repository. Moreover, programmable platforms are restricted in both topology and maturity, which drives clients toward platform as a service (PaaS) and cloud delivery.
- Programmable platform governance is limited by the lack of standardization of the APIs. In balance, established mature APIs drive engagement or interface between the layers managed through well-defined and structured APIs.

User Recommendations

- Use platform engineering principles to improve developer experience and ease the cognitive load of programmable platforms. Before considering programmable platforms, balance the aversion to vendor specific services lock-in which is prevalent in PaaS and XaaS.
- Deliver platform engineering principles by providing developer platforms to abstract and address the complexity of the APIs.
- Design programmable platform control plane APIs to not only monitor and manage consumption and provision, but also provide governance and compliance guardrails.
- Use APIs for SLAs, chargeback and consumption models led by the drive of standardization and automation of cloud delivery with programmable platforms.

Sample Vendors

Avesha; CU Coding; Microsoft; Oracle; SAP; Silk

Gartner Recommended Reading

Quick Answer: How Can I Optimize the Use of Programmable Platforms for Effective Software Delivery?

Adopt Platform Engineering to Improve the Developer Experience

A Software Engineering Leader's Guide to Improving Developer Experience

Chaos Engineering

Analysis By: Jim Scheibmeir, Hassan Ennaciri

Benefit Rating: Moderate

Market Penetration: 20% to 50% of target audience

Maturity: Early mainstream

Definition:

Chaos engineering is the use of experimental and potentially destructive failure testing or fault injection to uncover vulnerabilities and weaknesses within a distributed system. Chaos engineering tools provide the ability to systematically plan, document, execute and analyze an attack on components and whole systems throughout the life cycle of the system. This planning may include the injection of random timing or attack executions.

Why This Is Important

Many organizations rely on test plans that overemphasize functionality and underemphasize validating the system's reliability and resilience. The distribution and complexity of systems makes understanding them more difficult. Chaos engineering (CE) shifts the focus of testing a system from the "happy path" toward testing how it can degrade gracefully or continue to be useful and secure while under various levels of impact. Applying CE enables improvements to system knowledge and documentation.

Business Impact

CE is aimed to minimize time to recovery and the change failure rate, while maximizing uptime and availability. Addressing these elements helps improve customer experience, satisfaction, retention and acquisition. Gartner inquiries regarding CE increased by over 11% between 2021 and 2022.

Drivers

- Increased complexity of systems and increasing customer expectations are the two largest drivers of CE and the associated tools.
- As systems become more rich in features, they also become more complex in their composition and more critical to digital business success.
- Overall, CE helps organizations become more resilient across their processes, knowledge and technology.
- Teams often lack the confidence to handle failures and the psychological safety to take action to resolve incidents. CE can help build that confidence.

Obstacles

- Within many organizations, the predominant view of CE is that the practice is random, first implemented during production, and increases, rather than reducing, risk.
- Organizational culture and attitudes toward quality and testing can present barriers to the adoption of CE. When quality and testing are only viewed as overheads, there will be a focus on feature development over application reliability.
- It can be challenging just to secure the time and budget to invest in learning CE and associated technologies. Organizations must reach minimum levels of expertise so that value is returned.

User Recommendations

- Utilize a test-environment-first approach by practicing CE in preproduction environments.
- Incorporate CE into your system development, CI/CD or testing processes.
- Build out incident response protocols and procedures, as well as monitoring, alerting and observability capabilities, in tandem with the advancement of the CE practice.
- Utilize scenario-based tests — known as “game days” — to evaluate and learn about how individual IT systems would respond to certain types of outages or events.
- Investigate opportunities to use CE in production to facilitate learning and improvement at scale as the practice matures. However, Gartner believes that very few organizations purposely use CE in their production environments.
- Formalize the practice by adopting a platform or tool to track the activities and create metrics to build feedback for continuous improvements.

Sample Vendors

Amazon Web Services; ChaosIQ; Gremlin; Harness; Microsoft; Steadybit; Verica

Gartner Recommended Reading

Quick Answer: What Metrics Should We Use to Assess and Improve Software Quality?

Predicts 2023: Observing and Optimizing the Adaptive Organization

Top Strategic Technology Trends for 2023: Digital Immune System

Predicts 2023: How Innovation Will Transform the Software Engineering Life Cycle

Infrastructure Platform Engineering

Analysis By: Hassan Ennaciri, Paul Delory

Benefit Rating: High

Market Penetration: 20% to 50% of target audience

Maturity: Adolescent

Definition:

Infrastructure platform engineering is the discipline of building internal software products that present IT infrastructure to users or other platforms in an easily consumable way. Infrastructure platforms are self-service tools that allow nonexpert users to deploy and manage infrastructure themselves while I&O retains governance, security and compliance. Infrastructure platforms are often used as the foundation of higher-order, self-service layers such as internal developer platforms.

Why This Is Important

Digital enterprises are pressured to innovate and deliver products faster to meet customer needs. This requires adopting new operating models and modern practices to deliver scalable, reliable platforms that enable faster product delivery. Infrastructure platform engineering provides automated delivery of curated secure, reliable and scalable infrastructure services that can be available via self services or APIs and reduce the effort and cycle time for users to request and access the products.

Business Impact

Infrastructure platform engineering abstracts the complexity of the digital infrastructure to deliver platforms that continuously evolve to meet customer needs. It is an agile approach necessary to enable software products' value streams to meet customer needs and expectations. It also provides on-demand, fast access to environments, services and tools that improve customer experience and productivity.

Drivers

- **Business agility and innovation:** Digital businesses are required to be responsive to customers' needs and changing market conditions. They must have the ability to quickly deliver products that meet these changing demands and requirements.
- **Cost optimization:** Infrastructure platform engineering teams leverage automation to deliver scalable, reliable and secure platforms. This helps to improve efficiency, reduce resource cost due to manual work and reduce downtime due to change failures. Standardizing tools and platforms also optimizes resource utilizations and reduces cost incurred in tool proliferation.
- **Digital infrastructure and platform complexity:** Public cloud IaaS and PaaS deliver extensive capabilities and are designed to be consumable by developers, but most enterprises need additional governance and management that is best delivered by a platform engineering team.
- **Improve developer experience and productivity:** Infrastructure platform engineering abstracts complexity from developers and provides them with quick access or self-service in the environments they need to develop and test their software. Services can be made via an internal developer portal (IDP) such as Backstage, Calibo or Humanitec.
- **Compliance and security:** Infrastructure platform engineering automates and integrates compliance and security controls into software delivery pipelines, improving the organization's security posture and reducing the burden from developers.

Obstacles

- **Confusion:** There is a lot of hype and confusion about platform engineering and what it means. Many vendors are defining it to help sell their products, causing uncertainty with teams trying to adopt it.
- **Cultural:** This operating model is a new, modern approach that requires a shift in how teams work and collaborate, which is the hardest obstacle to overcome for many organizations.
- **Lack of skills:** Infrastructure platform engineering requires software engineering and specialized skills that may not exist in the organization.
- **Structure of traditional I&O operating models:** The organizational structure of many I&O teams is set up by domain specializations, making it hard to develop and deliver end-to-end services.
- **IT service management approaches:** The current approaches are process-heavy and rely on tickets and handoffs.
- **Complexity:** Successful implementation of infrastructure platform engineering is challenging because it requires new roles and involvement from many stakeholders.

User Recommendations

- **Start small and evolve:** Define initial goals and objectives of the platform by understanding common user needs and delivering viable products that continuously evolve to meet those needs.
- **Build a dedicated team with the right skills:** Successful infrastructure engineering practice requires dedicated teams with diverse skills in infrastructure platforms and software engineering.
- **Identify and fill critical roles such as platform owner and platform architect.** Acquire new talent with the required technical skills, the right mindset and strong interpersonal skills. Develop existing resources by provisioning continuous learning opportunities.
- **Adopt a product mindset:** Thread platform users as customers and ensure that you talk to them and continuously get their feedback to meet their existing needs as well as anticipate their future needs. Enable users and reduce the level of effort required to use the platform products.

Gartner Recommended Reading

Adopt Platform Engineering to Improve the Developer Experience

Top Strategic Technology Trends for 2023: Platform Engineering

Innovation Insight for Internal Developer Portals

Quick Answer: How Can I Optimize the Use of Programmable Platforms for Effective Software Delivery?

Guidance Framework for Implementing Cloud Platform Operations

Platform Engineering

Analysis By: Bill Blosen, Paul Delory

Benefit Rating: Transformational

Market Penetration: 20% to 50% of target audience

Maturity: Adolescent

Definition:

Platform engineering is the discipline of building and operating self-service developer platforms for software development and delivery. A platform is a layer of tools, automations and information maintained as products by a dedicated platform team, designed to support software developers or other engineers by abstracting underlying complexity. The goal of platform engineering is to optimize the developer experience and accelerate delivery of customer value.

Why This Is Important

Digital enterprises need to respond quickly to customer and internal demands; therefore, flexible, complex distributed software architectures have become popular. Software product teams struggle to focus on features due to this complexity, which results in poor developer experience. Platform engineering provides a self-service, curated set of tools, automations and information driven by developer priorities to accelerate value delivery in line with internal stakeholders, such as security and architecture.

Business Impact

Platform engineering empowers application teams to deliver software value faster. It removes the burden of underlying infrastructure construction and maintenance and increases teams' capacity to dedicate time to customer value and learning. It makes compliance and controls more consistent and simplifies the chaotic explosion of tools used to deliver software. Platform engineering also improves the developer experience, thus reducing employee frustration and attrition.

Drivers

- Scale: As more teams embrace modern software development practices and patterns, economies of scale are created, whereby there is enough value to justify creating a platform capability shared by multiple teams.
- Cognitive load: Adoption of modern, distributed architectural patterns and software delivery practices means that the process of getting software into production involves more tools, subsystems and moving parts than ever before. This places a burden on product teams to build a delivery system in addition to the actual software they are trying to produce.
- Need for increased speed and agility: The speed and agility of software delivery is critical to CIOs. As a result, software organizations are pursuing DevOps which is a tighter collaboration of infrastructure and operations (I&O) and development teams to drive shorter development cycles, faster delivery and increased deployment frequency. This will enable organizations to respond immediately to market changes, handle workload failures better and tap into new market opportunities. Platform engineering can drive this type of cross-team collaboration.
- Emerging platform construction tools: Many organizations have built their own platforms, but to date, these platforms have been homegrown, individual efforts tailored to the unique circumstances of the organizations that build them. Platforms generally have not been transferable to other companies or sometimes even to other teams within the same company. However, a new generation of platform-building tools is emerging to change that.
- Infrastructure modernization: During digital modernization, some forward-looking I&O teams embrace a new platform engineering role as a way to deliver more value, increasing their relevance to the business.

Obstacles

- Lack of skills: Platform engineering requires solid skills in software engineering, product management and modern infrastructure, all of which are in short supply.
- Platform engineering is easily misunderstood: Traditional models of mandated platforms with limited regard for developer experience can easily be relabeled and thus not achieve the true benefits of platform engineering.
- Outdated management/governance models: Many organizations still use request-based provisioning models. Those need to give way to a self-service, declarative model, with the primary focus being the effectiveness of the end users developing and operating solutions using the platform.
- Internal politics: There are many intraorganizational fights that could derail platform engineering. Product teams may resist giving up control of their customized toolchains. There might also be no appetite to improve the developer experience. Enterprises may also refuse to fund platform engineering without a clear ROI.

User Recommendations

- Start small with cloud-native workloads: Begin platform-building efforts with thinnest viable platforms for the infrastructure underneath cloud-native applications such as containers and Kubernetes.
- Embed security into platforms: Enable shift-left security within DevOps pipeline platforms, which will provide a compelling paved road to engineers.
- Don’t expect to buy a complete platform: Any commercially available tool is unlikely to provide the entirety of the platform you need. Thus, the job of the platform team is to integrate the components necessary for the platform to meet your needs.
- Implement a developer portal as part of your platform: An internal developer portal (IDP) serves as the user interface that enables self-service discovery and access to internal developer platform capabilities. Consider Backstage open-source or other commercial tools. Note: “IDP” has multiple meanings in this context, as well as in the industry.

Gartner Recommended Reading

How to Start and Scale Your Platform Engineering Team

Guidance Framework for Implementing Cloud Platform Operations

Adopt Platform Engineering to Improve the Developer Experience

Innovation Insight for Internal Developer Portals

DevOps Platforms

Analysis By: Manjunath Bhat

Benefit Rating: High

Market Penetration: 5% to 20% of target audience

Maturity: Early mainstream

Definition:

DevOps platforms provide fully integrated capabilities to enable continuous delivery of software using agile and DevOps practices. These span the software development life cycle (SDLC) and include product planning, version control, continuous integration, test automation, continuous deployment, release orchestration, automating security and compliance policies, monitoring, and observability. DevOps platforms support team collaboration, secure software development and software delivery metrics.

Why This Is Important

Organizations use DevOps platforms to minimize tool friction and operational complexity resulting from disparate toolchains, manual handoffs, and lack of consistent visibility throughout the SDLC. This enables product teams to deliver faster customer value without compromising quality. The DevOps platforms market reflects the consolidation of technologies across development, security, infrastructure and operations to streamline software delivery.

Business Impact

DevOps platforms are the software delivery pipelines that enable continuous delivery of business value. The seamless integration, automation, extensibility and shared visibility between development, security and operations workflows help bridge the silos that exist between these teams. Using a common platform for development, security and operations accelerates agile transformation, and helps organizations move toward a product and platform team operating model.

Drivers

- **Modernizing application architectures:** Modernizing applications to take advantage of emerging cloud-native architectures requires fundamental changes to underlying DevOps practices and tools.
- **Increased emphasis/focus on enhancing developer experience:** Improved developer experience, agility and the need to improve delivery cadence by reducing cognitive load due to constant context switches and repetitive low-value work.
- **An integrated approach to security and compliance:** Integrating and automating security, compliance and governance as part of the development and delivery process is becoming a priority. A few DevOps platform providers include SCA capabilities as features in their offerings. Example vendors include GitHub, GitLab and JFrog.
- **Improved visibility into the flow of work:** Organizations are under pressure to reduce friction and manual handoffs, and this requires complete visibility into software delivery pipelines from ideation to production.

Obstacles

- Organizations that want to unlock the full benefits of DevOps platforms must be willing to replace an existing toolchain — either completely or in part. Teams can view the change as a disruption to their established ways of working and resist any change to the tools they have been using.
- Organizations accrue technical and skill debt over time due to outmoded automation workflows and legacy applications. This hinders teams from adopting new tools.
- Dependency on a single provider for a majority of their software development needs increases concentration risk and lowers bargaining leverage.
- Most DevOps platforms currently fall short in providing the full set of software delivery capabilities that organizations require to build, deliver, measure and improve the flow of value in the software delivery life cycle.

User Recommendations

- To fully reap the business benefits of DevOps platforms, organizations must adopt agile methods and practices.
- Scale and deliver capability by providing DevOps platforms as self-service platforms to reduce overhead, lower complexity, and ensure consistent and templatized workflows across multiple teams.
- Improve the flow of value by streamlining the software delivery life cycle with DevOps platforms that provide enhanced visibility, traceability, auditability and observability across the DevOps pipeline.
- Support InnerSource efforts by building InnerSource portals using source control repositories available in DevOps platforms.
- Reduce inconsistency in CI/CD pipeline definitions between teams by leveraging declarative and shareable pipeline capabilities in DevOps platforms.

Sample Vendors

Atlassian; CircleCI; CloudBees; GitHub; GitLab; Harness; JetBrains; JFrog; Red Hat; VMware

Gartner Recommended Reading

Keys to DevOps Success

Research Roundup for DevOps, 2022

Cloud-Native

Analysis By: David Smith, Michael Warrilow

Benefit Rating: High

Market Penetration: 5% to 20% of target audience

Maturity: Adolescent

Definition:

Cloud-native refers to something created to optimally leverage or implement cloud characteristics. Those cloud characteristics are part of the original definition of cloud computing, and include capabilities delivered as a service. Cloud computing characteristics also include being scalable and elastic, shared, metered by use, service based, and ubiquitous by means of internet technologies.

Why This Is Important

Cloud-native is a popular term. Depending on its meaning, it can be described as taking full advantage of the cloud capabilities of a cloud provider, or using approaches pioneered in the cloud to deliver benefits wherever needed, via specific technologies such as containers. Cloud-native is not one thing, and there is a battle of ideas.

Business Impact

Cloud-native is a popular, hyped concept that aspires to attain and maximize the benefits of cloud computing; however, the realization of those benefits varies. For example, if a traditional, noncloud application is migrated to the cloud through a lift-and-shift approach, the application is unlikely to fully leverage cloud characteristics and deliver the maximum benefits. An application rewritten to take advantage of cloud capabilities is more likely to deliver the expected cloud outcomes.

Drivers

- The primary driver for cloud-native is the desire to "get the most out of the cloud." The cloud itself means different things to different constituencies, so it's not surprising that cloud-native means different things. What drives people to one or another of these approaches varies.
- Cloud-native can optimally leverage cloud technologies and benefits. The two most common meanings in use are contradictory. CSP-native is all about using native features and, therefore, locking yourself into a provider. Container-native focuses on containers, and may evolve into other technologies. This doesn't guarantee portability, but is directionally consistent with the goal.
- There are multiple aspects to cloud-native, ranging from design to architectural to operational practices. Examples include LIFESPAR and the Twelve-Factor App (i.e., cloud-native application design) and DevOps (cloud-native operations).
- Cloud-native can be viewed on a continuum. It's not a question of whether something is cloud-native or not; it's the degree to which it is. The more it aligns with cloud characteristics, the more cloud-native it is.

Obstacles

- Cloud-native is confusing due to its many interpretations. It's especially challenging with respect to hype, because confusion amplifies hype. The biggest obstacle is getting beyond the confusion to focus on desired outcomes.
- It is essential to be realistic about the portability that can be achieved and the cost. Otherwise, these features may not be used "with your eyes open," and you may not be aware you are doing so.
- In cloud strategy efforts, principles are the most important component. Cloud-native and multicloud are often stated as principles in a cloud strategy. These principles can contradict each other, and require further explanation.
- Use of the term "cloud-native" requires clarification of which meaning is being used. This is a function of the hype surrounding cloud-native. Being clear about goals is key to optimally leveraging cloud-native. Assuming that containerizing an application will inherently make it cloud-native is an obstacle. We call this "container-native."

User Recommendations

- Focus on the outcomes you want from using the cloud, rather than focusing purely on the definition of cloud-native. The more your use cases align with core cloud characteristics, the more likely you are to realize the benefits of using the cloud.
- Assess vendor claims about their cloud-native capabilities with skepticism. Vendors use the term "cloud-native" to promote their offerings, regardless of how cloud-native their offerings are.
- Ensure that the supporting tools, processes and operations support cloud characteristics when building or acquiring cloud-native applications or services. The value of cloud-native applications can be subverted when the approaches of the supporting elements are not cloud-native.
- Embrace services designed to bring you closer to cloud-native outcomes. These can include containers, microservices architecture, serverless design, functions and many platform-as-a-service (PaaS) services. However, using these technologies should be a means, not a goal.

Gartner Recommended Reading

The Cloud Strategy Cookbook, 2023

Infographic: Cloud-Native and Multicloud — Buzzwords or Key Principles in Your Cloud Strategy

A CTO's Guide to Cloud-Native: Answering the Top 10 FAQs

Define and Understand New Cloud Terms to Succeed in the New Cloud Era

Sliding into the Trough

Cloud-Native Application Protection Platforms

Analysis By: Neil MacDonald, Charlie Winckless

Benefit Rating: High

Market Penetration: 1% to 5% of target audience

Maturity: Emerging

Definition:

Cloud-native application protection platforms (CNAPPs) are an integrated set of security and compliance capabilities designed to help secure and protect cloud-native applications across development and production. CNAPPs consolidate a large number of previously siloed capabilities, including container scanning, cloud security posture management, infrastructure as code scanning, cloud infrastructure entitlements management and runtime workload protection.

Why This Is Important

Comprehensively securing cloud-native applications requires the use of multiple tools from multiple vendors that are rarely well-integrated. This lack of integration and automation slows developers down and creates fragmented visibility of risk and friction. CNAPP offerings allow an organization to use a single integrated offering to protect the entire life cycle of a cloud-native application.

Business Impact

Cloud-native application protection platforms consolidate disparate fragmented security testing and protection tools that increase cost and complexity for IT. Using a CNAPP offering will improve developer and security professional efficacy. It will also reduce complexity and costs while maintaining development agility and improving the developer's experience.

Drivers

CNAPPs:

- Reduce the chance of misconfiguration, mistake or mismanagement as cloud-native applications are rapidly developed, released into production and iterated.

- Converge and reduce the number of tools and vendors involved in the continuous integration/continuous delivery (CI/CD) pipeline.
- Reduce the complexity and costs associated with creating secure and compliant cloud-native applications.
- Facilitate the reporting and auditing of cloud security posture/status.
- Improve developer acceptance with security-scanning capabilities that seamlessly integrate into their development pipelines and tooling.
- Place an emphasis on scanning proactively in development and rely less on runtime protection, which is well-suited for container as a service and serverless function environments.

Obstacles

- Cloud workload protection platform (CWPP) vendors that are good at runtime protection aren't necessarily good at integrating into development and vice versa.
- Cloud-native workloads in the form of containers and serverless functions don't require heavyweight runtime protection capabilities.
- There is no single CNAPP offering that does everything. Convergence of capabilities will occur, but will take place over several years.
- Organizations may have siloed purchases of application security testing tooling that is chosen by a different team that manages the runtime protection of workloads. Even at runtime, a separate team may be responsible for web application protection.
- Organizational immaturity in terms of cloud-native application development may inhibit adoption and fragment buying motions.
- Buying centers and influencers are shifting to newer roles such as DevOps architects and cloud security engineering, requiring information security teams to coordinate with these users.

User Recommendations

- Sign contracts of only one to two years because the market for CNAPP is changing rapidly.

- Solicit CWPP vendors to scan containers in development and add cloud security posture management (CSPM) capabilities, including infrastructure-as-code scanning.
- Select integrated offerings with flexible licensing models that allow you to only pay for the capabilities your organization is prepared to use.
- Evaluate the CSPM vendor's ability to add scan of Kubernetes security posture management (KSPM) as well as provide runtime Kubernetes protection capabilities.
- Consolidate open-source software (OSS) vulnerability scanning and software composition analysis through integrations or replacement within a CNAPP offering.
- Scan containers proactively in development for all types of vulnerabilities, not just vulnerable components, including hard-coded secrets, malware and Kubernetes misconfiguration.

Sample Vendors

Aqua Security; Cisco; Lacework; Microsoft; Orca Security; Palo Alto Networks; Rapid7; Sysdig; Trend Micro; Wiz

Gartner Recommended Reading

Market Guide for Cloud-Native Application Protection Platforms

How to Select DevSecOps Tools for Secure Software Delivery

How to Make Cloud More Secure Than Your Own Data Center

Site Reliability Engineering

Analysis By: George Spafford, Daniel Betts

Benefit Rating: Transformational

Market Penetration: 5% to 20% of target audience

Maturity: Adolescent

Definition:

Site reliability engineering (SRE) is a collection of systems and software engineering principles used to design and operate scalable resilient systems. Site reliability engineers work with the customer or product owner to understand operational requirements and define service-level objectives (SLOs). Site reliability engineers work with product or platform teams to design and continuously improve systems that meet defined SLOs.

Why This Is Important

SRE emphasizes the engineering disciplines that lead to resilience; but individual organizations implement SRE in widely varying ways such as a defined role or a set of practices. SRE teams can serve as an operations function, and nearly all such teams have a strong emphasis on blameless root cause analysis. This is to decrease the probability and/or impact of future events and to enable organizational learning, continual improvement and reductions in unplanned work.

Business Impact

The SRE approach to improving reliability and resilience is intended for products and platforms that need to deliver customer value at speed at scale while managing risk. The two primary use cases are to improve the reliability of existing products/platforms or to create new products or platforms that need reliability from the start.

Drivers

- Clients are under pressure to meet customer requirements for reliability while scaling their digital services and are looking for guidance to help them.
- While Google originated what became known as SRE and continued to evolve it, practitioners are developing and sharing new practices as well. Potential practitioners looking for pragmatic guidance to improve the reliability of their systems have a rich body of knowledge they can leverage that works well with agile and DevOps.
- Organizations are adopting highly skilled automation practices (usually DevOps), and usage of infrastructure-as-code capabilities (which usually requires a cloud platform) to deliver digital business products reliably.
- The most common use case based on inquiry calls with clients is to leverage SRE concepts to improve the reliability of existing systems that are not meeting customer requirements for availability, performance or are proving difficult to scale.

Obstacles

- Insufficient internal marketing to understand what agile, DevOps or product teams need or would value and then explaining how the value SRE can deliver will justify the costs and risks incurred. Without marketing its benefits, SRE adoption tends to be less certain or slower. The SRE concept by itself is insufficient — people must continuously believe it is worthwhile.
- Finding SRE candidates who have the right mix of development, operations and people skills is a big challenge for clients. Impacts on initial adoption and scaling efforts as well.
- Rebranding of a traditional operations team without changing to adopt SRE practices, only SRE in name.
- Clients have voiced problems with product owners who overly focus on functional requirements and not nonfunctional requirements thus slowing improvements and support of SRE within the organization.

User Recommendations

- Leverage practices pragmatically based on need. Don't feel that you must implement SRE exactly the way Google does it, learn what works for you.
- Detect an opportunity to begin that is politically friendly, will demonstrate sufficient value and has an acceptable risk profile.
- Start small, focus, learn, improve, and demonstrate value — do not try to change everything at once.
- Work with the customer or product owner to define clear, obtainable SLOs based on their needs.
- Implement monitoring and improve observability to objectively report on actual performance relative to the SLOs.
- Product owners must be accountable for functional and non-functional requirements of their products.
- Instill collaborative working between site reliability engineers, developers and other stakeholders to help them learn how to design, build and evolve their products to meet SLOs.
- Create a community, implement effective organizational learning practices and evolve SRE practices.

Sample Vendors

Atlassian; Blameless; Datadog; Dynatrace; New Relic; OpsRamp; PagerDuty; Splunk

Programmable Infrastructure

Analysis By: Philip Dawson, Nathan Hill

Benefit Rating: High

Market Penetration: 5% to 20% of target audience

Maturity: Adolescent

Definition:

Programmable infrastructure is the concept of using and applying methods and tooling from the software development area to management of IT infrastructure. This includes, but is not limited to, APIs, immutability, resilient architectures and agile techniques.

Why This Is Important

Programmable infrastructure ensures optimal resource utilization, while driving cost efficiencies. A continuous delivery approach requires continuous insight and the ability to automate application responses. Moving to an API-driven infrastructure is the key first necessary step to enabling anti-fragile and sustainable automation through programmatic techniques.

Business Impact

Greater value (rather than cost reduction) is achieved via programmable infrastructure's ability to drive adaptive automation — responding faster to new business infrastructure demands, driving service quality and freeing staff from manual operations. Programmable infrastructure reduces technical debt with investment and enables a sustainable and highly responsive IT infrastructure service to the business.

Drivers

- Programmable infrastructure strategies are applied to private cloud, hybrid cloud and infrastructure platforms as well as public cloud. Demand for programmable infrastructure grows as heterogeneous infrastructure strategies are embraced.
- Programmable infrastructure is needed to manage the life cycle of infrastructure delivery from provisioning, resizing and reallocation to reclamation, and in the case of external resources, manage elasticity and the termination of consumption.
- Programmable infrastructure is needed to optimize and reduce the dependency on the infrastructure life cycle. More importantly, it enables the desired (performance, cost, speed) infrastructure provisioning and orchestration in line with business demands.

Obstacles

- The ongoing cost of refreshing API-enabled infrastructure components on-premises after initial implementation adds financial pressure to organizations.
- Applying automation to existing monolithic infrastructure components fails due to the lack of platform agility and vendor lock-in.
- While APIs enable integration across different infrastructure platforms, the lack of open APIs/API compatibility across vendor platforms creates a siloed mentality.
- The implementation of programmable infrastructure is hampered by the early adoption of it within infrastructure and operations (I&O), and the shortage of skilled software engineering resources to comprehensively exploit it (especially in web technologies such as HTTP and JSON to develop these APIs).

User Recommendations

- Deploy a programmable infrastructure to further abstract application from infrastructure delivery and pursue an agile digital business outcome.
- Implement a programmable infrastructure by investing in infrastructure automation tools and continuous delivery (example vendors for these markets are listed below, but no single vendor or platform can enable an organizationwide programmable infrastructure strategy) leading to API-led programmable platforms.
- Invest in infrastructure and DevOps, and modernize legacy IT architectures to implement an API-driven infrastructure.
- Examine reusable programmable infrastructure building blocks leveraging programmable infrastructure strategy built on repeatable and available skills from providers.

Sample Vendors

Amazon Web Services; CU Coding; Google; IBM; Microsoft; Oracle; Quality Technology Services; RackN; Tencent; VMware

Gartner Recommended Reading

Market Guide for Servers

Predicts 2023: XaaS Is Transforming Data Center Infrastructure

Quick Answer: How Can I Optimize the Use of Programmable Platforms for Effective Software Delivery?

AIOps Platforms

Analysis By: Matt Crossley, Matthew Brisse

Benefit Rating: High

Market Penetration: 20% to 50% of target audience

Maturity: Early mainstream

Definition:

Gartner defines AIOps platform as the application of AI/ML and data analytics at the event management level in order to augment, accelerate and automate manual efforts in the event management process and associated procedures. AIOps platforms are defined by the key characteristics of cross-domain event ingestion, topology assembly, event correlation and reduction, pattern recognition, and remediation augmentation.

Why This Is Important

The combination of increasing application complexity, monitoring tool proliferation, and increasing volumes and varieties of telemetry has shifted complexity from gathering data to interpreting data. AIOps platforms apply machine learning (ML) and data analytics to classify and cluster cross-domain events in near real time, at scale, and in ways that can exceed human capacity. These inferences can augment human analysis, accelerate human response, or automate a process to resolve an issue.

Business Impact

AIOps platforms deliver value through:

- Agility and productivity: By reducing alert fatigue through identification and correlation of related events, operators can focus on fewer, more critical events.
- Service availability and triage cost: By reducing the time and effort required to identify root causes and augmenting, accelerating, or automating remediation.
- Increased value from monitoring tools: By unifying events from siloed tools and learning actionable event patterns across domains.

Drivers

Demand for AIOps platform capabilities is accelerating and is fueled by:

- **Increasing complexity:** Organizations use an increasingly complex mix of IT assets that rely on a highly integrated combination of on-premises assets, cloud IaaS/PaaS providers and SaaS platforms to deliver solutions.
- **Increasing monitoring expectations:** Investments and improvements in monitoring and the pursuit of observability are generating more data from more sources. Increasing demand and advances in monitoring trends, like application performance management (APM) and digital experience monitoring (DEM), present operators with extremely detailed views into their business applications and the end-user experience. Effective use of this additional data requires near-real-time analysis and rationalization of events from related assets and services.
- **Demands for reliability:** Shifts in roles and responsibilities driven by modern operating models, like DevOps and SRE, in the pursuit of greater availability and faster incident resolution. AIOps platforms enable agility by offloading some of the mechanical tasks of event triage, root cause analysis and solution identification. This both accelerates response for common issues and frees up human creative capacity for novel events and business priorities.

Obstacles

- **Unrealistic expectations:** Hype is a major obstacle to AIOps platform adoption. Clients struggle to separate claims of AI and magical automation from achievable use cases. This impacts demonstrating value of AIOps platforms, specifically quantifiable return on investment.
- **Maturity of dependencies:** Benefits of AIOps platforms beyond event correlation requires maturity in dependencies such as automation.
- **Time to value:** AIOps platforms learn through observation, modeling normal data patterns, and associate a solution with these patterns. This can take time depending on the frequency of occurrence. Developing accurate detection models for rare events can take months.
- **Market shifts and maturity:** Monitoring vendors are moving up the stack, AIOps platform vendors are reaching into monitoring domains, and ITSM vendors use AIOps capabilities to extend their reach. Expect further convergence and market shifts to change the definition of “state of the art.”

User Recommendations

- Establish clear, realistic use cases for an AIOps platform pilot and validate them individually, rather than all at once. This approach helps reveal pockets of potential value that might be missed when evaluating only the aggregate impact. Ultimately, this fundamental step underpins an eventual strategy, while scoping the vendor landscape, clarifying technical and process dependencies, and separating hype from reality.
- Layer the AIOps features within monitoring tools with the cross-domain analysis of an AIOps platform. This approach enables efficient data ingestion and analysis, and the surfacing of insights across domains.
- Do not require automation outcomes for all AIOps applications. There is tremendous value in accelerating and augmenting human activity. These approaches often avoid the challenge of the probabilistic uncertainty combined with automated change in production environments.

Sample Vendors

BigPanda; BMC Software; Digitate; IBM; Interlink; Moogsoft; OpsRamp; PagerDuty; ServiceNow; Splunk

Gartner Recommended Reading

Market Guide for AIOps Platforms

Deliver Value to Succeed in Implementing AIOps Platforms

Infographic: Artificial Intelligence Use-Case Prism for AIOps

Infographic: AIOps Architecture for Analyzing Operational Telemetry

How Do I Plan for Migrating My Data Center Infrastructure Into an XaaS Model?

Container Management

Analysis By: Dennis Smith, Michael Warrilow

Benefit Rating: High

Market Penetration: 20% to 50% of target audience

Maturity: Early mainstream

Definition:

Gartner defines container management as offerings that enable the development and operation of containerized workloads. Delivery methods include cloud, managed service and software for containers running on-premises, in the public cloud and/or at the edge. Associated technologies include orchestration and scheduling, service discovery and registration, image registry, routing and networking, service catalog, management user interface, and APIs.

Why This Is Important

Container management automates the provisioning, operation and life cycle management of container images at scale. Centralized governance and security are used to manage container instances and associated resources. Container management supports the requirements of modern applications, including platform engineering, cloud management and continuous integration/continuous delivery (CI/CD) pipelines. Benefits include improved agility, elasticity and access to innovation.

Business Impact

Industry surveys and client interactions show that demand for containers continues to rise. This trend is due to application developers' and DevOps teams' preference for container runtimes, which use container packaging formats. Developers have progressed from leveraging containers on their desktops to needing environments that can run and operate containers at scale, introducing the need for container management.

Drivers

- The adoption of DevOps-based application development processes.
- The rise of cloud-native application architecture based on microservices.
- New system management approaches based on immutable infrastructure, which gives the ability to update systems frequently and reliably maintained in a “last known good state” rather than repeatedly patched.
- Cloud-based services built with replaceable and horizontally scalable components.
- A vibrant open-source ecosystem and competitive vendor market have culminated in a wide range of container management offerings. Many vendors enable management capabilities across hybrid cloud or multicloud environments. Container management software can run on-premises, in public infrastructure as a service (IaaS), or simultaneously in both.
- Container-related edge computing use cases have increased in industries that need to get compute and data closer to the activity (for example, telcos, manufacturing plants, etc.).
- AI/ML use cases have emerged over the past few years, leveraging the scalability capabilities of container orchestration.
- Cluster management tooling that enables the management of container nodes and clusters across different environments is increasingly in demand.
- All major public cloud service providers now offer on-premises container solutions.
- Independent software vendors (ISVs) are increasingly packaging their software for container management systems through container marketplaces.
- Some enterprises have scaled sophisticated deployments, and many more are planning container deployments. This trend is expected to increase as enterprises continue application modernization projects.

Obstacles

- More abstracted, serverless offerings may enable enterprises to forgo container management. These services embed container management in a manner that is transparent to the user.
- Third-party container management software faces huge competition in the container offerings from the public cloud providers, both with public cloud deployments and the extension of software to on-premises environments. These offerings are also challenged by ISVs that choose to craft open-source components with their software during the distribution process.
- Organizations that perform relatively little app development or make limited use of DevOps principles are served by SaaS, ISV and/or traditional application development packaging methods.

User Recommendations

- Determine if your organization is a good candidate for container management software adoption by weighing organizational goals of increased software velocity and immutable infrastructure, and its hybrid cloud requirements, against the effort required to operate third-party container management software.
- Leverage container management capabilities integrated into cloud IaaS and platform as a service (PaaS) providers' service offerings by experimenting with process and workflow changes that accommodate the incorporation of containers.
- Avoid using upstream open source (e.g., Kubernetes) directly unless the organization has adequate in-house expertise to support.

Sample Vendors

Alibaba Cloud; Amazon Web Services; Google; IBM; Microsoft; Mirantis; Red Hat; SUSE; VMware

Gartner Recommended Reading

Market Guide for Container Management

Infrastructure Automation

Analysis By: Chris Saunderson

Benefit Rating: High

Market Penetration: 20% to 50% of target audience

Maturity: Mature mainstream

Definition:

Infrastructure automation (IA) enables DevOps and infrastructure and operations (I&O) teams to deliver automated infrastructure services across on-premises and cloud environments. This includes the life cycle of services through creation, configuration, operation and retirement. These infrastructure services are then made available through platform delivery, self-service catalogs, direct invocation and API integrations.

Why This Is Important

IA delivers velocity, quality, efficiency and reliability, with scalable, declarative approaches for deploying and managing infrastructure. These tools integrate into delivery pipelines targeting deployment topologies that range from on-premises to the cloud, and enable infrastructure consumers to build what is needed when they need it. Once deployed, IA provides day-2 and beyond operational automation, and extends to provide policy compliance and enforcement capabilities.

Business Impact

Implementing and maturing IA services will enable:

- **Agility** — continuous infrastructure delivery and operations
- **Productivity** — version-controlled, declarative, repeatable, efficient deployments
- **Cost improvement** — reductions in manual effort expended via increased automation
- **Risk mitigation** — compliance driven by standardized configurations
- **Collaboration** — delivering environments that product teams need with security, cost and compliance requirements baked in.

Drivers

I&O leaders must automate delivery through tool and skills investments to mature beyond simple deployments. The target should be standardized platforms that deliver the systemic, transparent management of platform deployments. This same discipline must be applied to the operation of these deployed platforms, ensuring that efficient operations (including automated incident response) can be achieved. IA tools deliver the following key capabilities to support this maturation:

- Multicloud/hybrid cloud infrastructure delivery
- Support for immutable and programmable infrastructures
- Predictable delivery enabling automated operations
- Self-service and on-demand environment creation
- Integration into DevOps initiatives (continuous integration/delivery/deployment)
- Resource provisioning, including cost optimization capabilities
- Operational configuration management efficiencies
- Policy-based delivery and assessment/enforcement of deployments against internal and external policy requirements
- Enterprise-level framework to enable maturing of automation strategies
- Skills and practice development inside infrastructure teams, enabling agile and iterative development and sustaining of services

Obstacles

- The combination of tools needed to deliver IA capability can increase tool count and complexity.
- Software engineering skills and practices are required to get maximum value from tool investments.
- IA vendor capability expansion overlaps and confuses the tool landscape, resulting in over-investment.
- Steep learning curves can cause developers and administrators to revert to familiar scripting methods to deliver required capabilities.

User Recommendations

- Identify existing IA tools in use to catalog capabilities, identify use cases and document overlaps to aid decision making.
- Assess existing internal IT skills to incorporate training needs that more fully enable IA, especially for an automation architect role to coordinate standards development and implementation.
- Baseline how managed systems and tooling will be consumed (e.g., engineer, self-service catalog, API or on-demand).
- Integrate security and compliance requirements into scope for automation and delivery activities.
- Develop an IA tooling strategy that incorporates current needs and near-term roadmap evolution.

Sample Vendors

Amazon Web Services; HashiCorp; Microsoft; Perforce; Pliant; Progress; Pulumi; RackN; Upbound; VMware

Gartner Recommended Reading

Market Guide for Infrastructure Automation Tools

Innovation Insight for Continuous Infrastructure Automation

To Automate Your Automation, Apply Agile and DevOps Practices to Infrastructure and Operations

How to Start and Scale Your Platform Engineering Team

IoT Platform

Analysis By: Alfonso Velosa, Eric Goodness, Scot Kim

Benefit Rating: High

Market Penetration: 5% to 20% of target audience

Maturity: Adolescent

Definition:

An Internet of Things (IoT) platform enables the connection and capture of data from IoT-enabled assets to develop, deploy, and manage business solutions that improve operations such as monitoring remote assets or optimizing maintenance. Capabilities include device management, integrated data management, analytics, application enablement and management, and security. It may be delivered as edge or on-premises software, or cloud IoT platform as a service, or a hybrid combination.

Why This Is Important

Enterprises continue adding IoT capabilities to assets and products, to achieve benefits such as cost optimization, process optimization, improved customer experience, sustainability and new opportunities such as product as a service. The complexity, scale and business value of these IoT solution requirements call for specialized technology resources, most often implemented as an IoT platform.

Business Impact

IoT platforms are required to implement IoT-enabled solutions to make better business decisions from the data generated by connected assets. Goals include:

- Differentiated smart products
- Cost optimization strategies centered on improved maintenance
- Optimizing output by coordinating asset health with process health
- Opportunities to sell new services and data products or adopt new business models such as product servitization
- Sustainability improvements and reporting

Drivers

- Asset-intensive (oil and gas, manufacturing) and asset-light industries (healthcare, insurance) are implementing IoT-enabled projects to meet business objectives.
- IoT platforms help enterprises accelerate time to market for smart products while consolidating and structuring the data.
- Enterprises are finding IoT platforms already incorporated in their equipment by their OEMs to help them lower operating costs, reduce waste, minimize carbon footprint, avoid unplanned downtime and enhance worker safety.
- Technology providers' are increasing their focus on business outcomes, encouraging enterprise customers to implement IoT projects. Tech provider investments in improved ecosystems and channel partners make it easier for clients to achieve business value.
- In parallel, technology providers continue to improve their technology, user experience and vertical market templates, to ensure they can deliver business solutions, such as reduced waste, for their customers.

Obstacles

- IoT platforms still require extensive customization to achieve business outcomes for large-scale deployments, driving up cost and schedule.
- Many enterprises approach IoT projects as technology projects, instead of business projects that use IoT platforms to achieve business outcomes.
- Many enterprises operate in siloed fashions, adopting different IoT platforms for each use case, limiting their ability to scale and adding complexity.
- Enterprise leaders often underinvest in culture change processes or in training key employees. This leads employees to underuse or reject the data produced by the IoT platform, leading the project to underperform against its objectives.
- Gaps in enterprise IT and operational skills to address IoT technical needs and complexity often create project delays.
- Technology providers have yet to clearly demonstrate they can deploy and support their platforms at a large scale on a global basis.

User Recommendations

- Start small. Treat initial IoT platform projects as IT and business capability programs to acquire implementation lessons, identify challenges and opportunities, and verify alignment with business KPIs and needs.
- Develop a scenario analysis for the probability IT will have to assume IoT platform budget and long-term management.
- Identify the differing enterprise needs for IoT platforms and establish an IT team to establish a multiplatform architecture. These include simple projects with new assets using new protocols versus complex projects in legacy plants that connect to heterogeneous assets and protocols.
- Use a skills gap for IoT platforms to build an improvement plan for IT team capabilities such as integration or digital twin development or security.
- Evaluate vendors across criteria such as vertical market expertise, proof of value projects, the ability to drive large-scale deployments, technology portfolio and partner ecosystem.

Sample Vendors

Alleantia; Alibaba; Arduino; AVEVA Group; Covacsis Technologies; Haier Group; Intelligent Plant; NEC; Panasonic; Vodafone Group

Gartner Recommended Reading

Magic Quadrant for Global Industrial IoT Platforms

Technology Opportunity Prism: Internet of Things

Competitive Landscape: IoT Service Providers

Infographic: IoT Use-Case Prism for Sustainability and ESG

Climbing the Slope

Cloud Workload Protection Platforms

Analysis By: Charlie Winckless, Neil MacDonald

Benefit Rating: Moderate

Market Penetration: 20% to 50% of target audience

Maturity: Mature mainstream

Definition:

Cloud workload protection platforms (CWPPs) protect workloads in hybrid and cloud deployments. CWPPs provide consistent visibility and control over physical machines, virtual machines, containers and serverless workloads, regardless of location. CWPP offerings protect the workload using a combination of system integrity protection, application control, behavioral monitoring, intrusion prevention and optional anti-malware protection.

Why This Is Important

As enterprises spread diverse workloads across data centers and public clouds, they need to maintain their visibility and control over these workloads during runtime. The most effective way to address the speed, scale and complexity of cloud-based workload security is to use an offering that is fit for purpose. Simply using a solution designed for on-premises data centers or end-user endpoints is a poor approach to these diverse workloads.

Business Impact

Enterprises are implementing hybrid data center architectures, with workloads spanning on-premises and public cloud IaaS providers, container-based implementations and serverless functions. These workloads are diverse, and their security requirements and threat models differ significantly from end-user systems and even traditional servers. To secure these workloads and realize the benefits of cloud-native applications, it is necessary to use security tooling designed for this environment.

Drivers

- The most effective way to address the speed, scale, complexity, and the ephemeral and elastic nature of cloud workload protection is to use a tool designed for how these workloads are deployed.

- Simply using a solution designed for on-premises data centers or end-user endpoint protection platforms (EPPs) is suboptimal. Thus, many vendors, including both startups and established EPP vendors, are now explicitly targeting the CWPP market.
- Cloud server workload protection strategies must be based on a foundation of solid operational hygiene, including proper administrative control, patching discipline and workload configuration management.
- Workloads are no longer remotely homogeneous. Tools must protect containers, virtual machines and serverless workloads, and grant the appropriate levels of visibility and security to each.
- Unlike end-user endpoints, server workloads do not commonly encounter and execute unknown arbitrary code, thus lending themselves to a default deny, zero-trust-based protection strategy that well-engineered CWPPs are built to support.
- As vendor convergence continues to be important to Gartner clients, the convergence of CWPP and cloud security posture management (CSPM) into a cloud-native application platform (CNAPP) consolidates previously siloed offerings and provides the same or greater value.

Obstacles

- Some organizations are maturing their approach to cloud protection and have not identified a need for cloud-native security toolsets, or prefer to continue with existing endpoint tools despite their lack of suitability for cloud deployments. Such organizations often still wish to extend on-premises controls and control patterns to the cloud, regardless of suitability.
- Single cloud-using organizations may wish to use CSP-native tools. This can be suboptimal due to potential future multicloud deployments, increasing options for cost and feature improvements.
- Not all vendors offer all capabilities. Some specialize in only one or two forms of workload protection.
- Not all vendors offer support for physical servers or out-of-support and older operating systems that still require protection.
- Some vendors utilize eBPF, which supports only newer versions of Linux.

- Serverless functions require new approaches that don't require agents or privileged containers.

User Recommendations

- Don't use an end-user EPP solution to protect cloud server workloads.
- Architect for consistent visibility and control of all workloads, regardless of location, size or type, as well as in cases where runtime agents may not be used or may not make sense.
- Evaluate converged CNAPP offerings to see if their CWPP capabilities are sufficient, since information sharing is highly valuable for detection, false positive reduction and prioritization.
- Require vendors to support well-integrated deployments in leading cloud platforms, especially for managed container and serverless approaches.
- Prioritize a default deny or application control approach.
- Extend workload scanning and compliance efforts into development (DevSecOps), especially for containers and serverless functions. Prefer platforms that support container and serverless environments.
- Require CWPP offerings to expose all functionality via API.

Sample Vendors

Aqua Security; CrowdStrike; Lacework; Microsoft; Palo Alto Networks; SentinelOne; Sophos; Sysdig; Trellix; Trend Micro

Gartner Recommended Reading

Market Guide for Cloud-Native Application Protection Platforms

5 Things You Must Absolutely Get Right for Secure IaaS and PaaS

Magic Quadrant for Endpoint Protection Platforms

Entering the Plateau

ITSM Platforms

Analysis By: Rich Doheny

Benefit Rating: High

Market Penetration: More than 50% of target audience

Maturity: Mature mainstream

Definition:

IT service management (ITSM) platforms offer workflow management that enables organizations to design, automate, manage, and deliver integrated IT services and digital experiences. Supported processes include request, incident, problem, change, knowledge and configuration management, and case management for non-IT business needs. IT leaders select these solutions to be consumed by service desks and service operations, and for business workflow administration in other IT-adjacent departments.

Why This Is Important

IT leaders require robust ITSM platforms to drive business value in the services they provide and enable digital business transformation outside of IT. These platforms help infrastructure and operations (I&O) teams to automate processes, design workflows, and support continual service improvement initiatives. They provide actionable insights that enable service operations, orchestration and process automation, and multichannel support.

Business Impact

ITSM platforms are most heavily used by IT service support and IT service delivery functions to enable the tasks and workflows for ITSM processes. They drive agility and help scale service delivery efforts through integration into adjacent IT operations management, digital experience, collaboration, and development solutions. In addition, out-of-the-box case management and low-code features extend request management into other areas of the business.

Drivers

- The ITSM platforms market is functionally mature. Features aligned with common ITSM practices and standards are commoditized.

- IT leaders are increasingly inquiring about applying service management into other areas of the business through a unified platform offering. ITSM platforms continue to expand service management workflows with out-of-the-box content to support line-of-business needs, such as HR and facilities case management.
- DevOps and DevSecOps are driving the need for more ITSM platform functions to be integrated into adjacent tooling and workflows for greater efficiencies, agility, and visibility.
- ITSM vendors are investing in new capabilities supporting more federated and agile ITSM practices, AI and artificial intelligence for IT operations (AIOps) integration, workforce and process optimization, integration with development and collaboration tools, and multichannel support.

Obstacles

- More than 400 vendors offer ITSM products, but most are basic or intermediate tools that focus on IT service desk and ticketing functions targeted at basic service desk requirements. With core process workflows built around common frameworks, many vendors struggle to create meaningful differentiated messaging and help their customers justify the ROI.
- Despite the large number of participants, the enterprise market is dominated by a small number of vendors.
- Advanced features typically require pricey add-ons or higher tiers of licensing. This will be a challenge for customers who do not align their product roadmap with their budget planning.
- Customers who try to implement too many platform features at once, across IT and multiple lines of business, will struggle to mature their practices and often lose momentum on their investments.

User Recommendations

- Identify current ITSM needs along with what you can pragmatically deploy over an 18-month roadmap to avoid overspending.
- Avoid costly customization by prioritizing tools that provide advanced process support and machine learning, as well as strong orchestration tools and out-of-the-box integration with other IT operations management and collaboration solutions.

- Select tools that support adaptive process models and integration into your DevOps toolchains, if you are pursuing DevOps and agile methodologies.
- Account for the total resource overhead associated with the product by factoring in licensing, cost and timing of implementation, ongoing maintenance, training required, and third-party products to meet base requirements.
- Involve business leaders for any non-IT case management decisions to ensure minimum functionality is met. Identify multichannel access and broader integration requirements into other line-of-business systems of record.

Sample Vendors

Atlassian; BMC Software; EasyVista; Freshworks; Ivanti; ManageEngine; ServiceNow

Gartner Recommended Reading

Magic Quadrant for IT Service Management Platforms

Critical Capabilities for IT Service Management Platforms

A Buyer's Guide to ITSM Platforms

Quick Answer: How to Successfully Implement Your ITSM Platform

Cloud Management Platforms

Analysis By: Dennis Smith

Benefit Rating: Low

Market Penetration: 5% to 20% of target audience

Maturity: Mature mainstream

Definition:

Cloud management platforms (CMPs) enable organizations to manage private, public and multicloud services and resources. Their functionality combines provisioning and orchestration; service request management; inventory and classification; monitoring and analytics; cost management and resource optimization; cloud migration; backup and disaster recovery; and identity, security and compliance. Functionality can be provided by a single product or a set of offerings with some degree of integration.

Why This Is Important

Enterprises and managed service providers will deploy CMPs to increase agility, reduce the cost of providing services and increase the likelihood of meeting service levels. CMPs promote cost-effective governance and accountability through self-service interfaces, automation, and adherence to standard best practices. They also play an important role in creating a unified layer of consumption and abstraction for organizations adopting a hybrid and multicloud model.

Business Impact

CMPs address issues related to aggregation, integration, customization and governance in the adoption of hybrid multicloud by organizations. The recent CMP market continually focuses on preventing enterprises from overspending or leaving themselves vulnerable to security issues — two key items to avoid when adopting cloud services. CMPs also address the need of managed cloud service providers for multitenant customer operations and support through a single management portal.

Drivers

- Organizations are acknowledging the need to foster end-user self-service provisioning with embedded governance.
- Vendors are addressing the key issues enterprises face. Many vendors are looking to combine cost management and security functionality into governance tooling. A few vendors are also looking to provide infrastructure-as-code assistance by overlaying cloud management functionality to this capability.
- Many vendors have added container management to their CMP offerings. The ability to serve both application developers, and infrastructure and operations (I&O) personas is key. This requires CMPs to be linked into the application development process to enable I&O teams to enforce provisioning standards, without imposing a workflow that inhibits agility.
- Global system integrators (GSIs) and management service providers (MSPs) leverage CMPs to build a services layer in order to create multicloud managed services.

Obstacles

- Cloud management comprises a complex and varied set of activities, and no CMP platform has been successful at addressing all customer needs.
- The CMP market has been gradually fragmenting, moving from integrated all-in-one tools to specialty tools that focus on a subset of capabilities, such as security, operations or financial management.
- Challenges vendors face include interfacing with multiple public clouds, cost transparency with workload optimization to remediate cost overruns, handling newer functions (e.g., containers and serverless deployments), and edge computing.
- Market dynamics have caused confusion among potential buyers. There have been acquisitions by CMP vendors and vendors in adjacent markets, combining CMP functionality with their existing functionality. Cloud service providers (CSPs) and MSPs have entered the market, and many long-standing vendors have introduced new products that target gaps in their previous ones.

User Recommendations

- Identify your full vertical (infrastructure as a service [IaaS], platform as a service [PaaS] and SaaS) and horizontal (on-premises, public cloud and edge) needs.
- Choose a single-focused specialized tooling, if your requirements do not expand beyond a limited scope.
- Select between native cloud services if you value depth with a cloud provider, or CMPs if you want breadth across cloud providers and your on-premises deployment.
- Determine the utility of functionally-focused tools by defining the organization's functionality needs. Integrate cloud management or traditional management tools because no vendor has a total cloud management solution.
- Ensure staffing to operate CMP platforms by planning new roles (e.g., cloud engineers and/or operators).
- Develop skills in financial and capacity management.
- Align with GSIs and MSPs to embed your CMP in their offerings as a key component of their multicloud managed services offering.

Sample Vendors

Arrow Electronics; CloudBolt; Kion; Morpheus Data; Snow Software; Turbot; VMware; Wipro Enterprises

Gartner Recommended Reading

Market Guide for Cloud Management Tooling

Market Guide for Container Management

6 Best Practices to Create a Cloud Management Services Offering in the World of Multicloud and Hybrid Cloud

Appendixes

Hype Cycle Phases, Benefit Ratings and Maturity Levels

Table 2: Hype Cycle Phases

(Enlarged table in Appendix)

Phase	Definition
Innovation Trigger	A breakthrough, public demonstration, product launch or other event generates significant media and industry interest.
Peak of Inflated Expectations	During this phase of overenthusiasm and unrealistic projections, a flurry of well-publicized activity by technology leaders results in some successes, but more failures, as the innovation is pushed to its limits. The only enterprises making money are conference organizers and content publishers.
Trough of Disillusionment	Because the innovation does not live up to its overinflated expectations, it rapidly becomes unfashionable. Media interest wanes, except for a few cautionary tales.
Slope of Enlightenment	Focused experimentation and solid hard work by an increasingly diverse range of organizations lead to a true understanding of the innovation's applicability, risks and benefits. Commercial off-the-shelf methodologies and tools ease the development process.
Plateau of Productivity	The real-world benefits of the innovation are demonstrated and accepted. Tools and methodologies are increasingly stable as they enter their second and third generations. Growing numbers of organizations feel comfortable with the reduced level of risk; the rapid growth phase of adoption begins. Approximately 20% of the technology's target audience has adopted or is adopting the technology as it enters this phase.
Years to Mainstream Adoption	The time required for the innovation to reach the Plateau of Productivity.

Source: Gartner (July 2023)

Table 3: Benefit Ratings

Benefit Rating ↓	*Definition* ↓
Transformational	Enables new ways of doing business across industries that will result in major shifts in industry dynamics
High	Enables new ways of performing horizontal or vertical processes that will result in significantly increased revenue or cost savings for an enterprise
Moderate	Provides incremental improvements to established processes that will result in increased revenue or cost savings for an enterprise
Low	Slightly improves processes (for example, improved user experience) that will be difficult to translate into increased revenue or cost savings

Source: Gartner (July 2023)

Table 4: Maturity Levels

(Enlarged table in Appendix)

Maturity Levels	Status	Products/Vendors
Embryonic	■ In labs	■ None
Emerging	■ Commercialization by vendors ■ Pilots and deployments by industry leaders	■ First generation ■ High price ■ Much customization
Adolescent	■ Maturing technology capabilities and process understanding ■ Uptake beyond early adopters	■ Second generation ■ Less customization
Early mainstream	■ Proven technology ■ Vendors, technology and adoption rapidly evolving	■ Third generation ■ More out-of-box methodologies
Mature mainstream	■ Robust technology ■ Not much evolution in vendors or technology	■ Several dominant vendors
Legacy	■ Not appropriate for new developments ■ Cost of migration constraints replacement	■ Maintenance revenue focus
Obsolete	■ Rarely used	■ Used/resale market only

Source: Gartner (July 2023)

Recommended by the Author

Some documents may not be available as part of your current Gartner subscription.

Understanding Gartner's Hype Cycles

Tool: Create Your Own Hype Cycle With Gartner's Hype Cycle Builder

How to Leverage Platform Engineering Principles to Deliver Infrastructure Platforms

Modernizing Infrastructure Platforms and Operating Models in Support of Digital Foundations

Table 1: Priority Matrix for Infrastructure Platforms, 2023

Benefit ↓	*Years to Mainstream Adoption*			
	Less Than 2 Years ↓	*2 - 5 Years* ↓	*5 - 10 Years* ↓	*More Than 10 Years* ↓
Transformational		Platform Engineering Site Reliability Engineering	Digital Platform Conductor Tools	
High	ITSM Platforms	Container Management DevOps Platforms Distributed Integrated Platforms GitOps Infrastructure Automation IoT Platform Programmable Infrastructure Value Stream Management Platforms	AIOps Platforms Cloud-Native Cloud-Native Application Protection Platforms Infrastructure Platform Engineering Intelligent Platforms Programmable Platform	
Moderate	Cloud Workload Protection Platforms		Chaos Engineering	
Low	Cloud Management Platforms			

Source: Gartner (July 2023)

Table 2: Hype Cycle Phases

Phase	*Definition*
Innovation Trigger	A breakthrough, public demonstration, product launch or other event generates significant media and industry interest.
Peak of Inflated Expectations	During this phase of overenthusiasm and unrealistic projections, a flurry of well-publicized activity by technology leaders results in some successes, but more failures, as the innovation is pushed to its limits. The only enterprises making money are conference organizers and content publishers.
Trough of Disillusionment	Because the innovation does not live up to its overinflated expectations, it rapidly becomes unfashionable. Media interest wanes, except for a few cautionary tales.
Slope of Enlightenment	Focused experimentation and solid hard work by an increasingly diverse range of organizations lead to a true understanding of the innovation's applicability, risks and benefits. Commercial off-the-shelf methodologies and tools ease the development process.
Plateau of Productivity	The real-world benefits of the innovation are demonstrated and accepted. Tools and methodologies are increasingly stable as they enter their second and third generations. Growing numbers of organizations feel comfortable with the reduced level of risk; the rapid growth phase of adoption begins. Approximately 20% of the technology's target audience has adopted or is adopting the technology as it enters this phase.
Years to Mainstream Adoption	The time required for the innovation to reach the Plateau of Productivity.

Phase ↓	Definition ↓

Source: Gartner (July 2023)

Table 3: Benefit Ratings

Benefit Rating ↓	Definition ↓
Transformational	Enables new ways of doing business across industries that will result in major shifts in industry dynamics
High	Enables new ways of performing horizontal or vertical processes that will result in significantly increased revenue or cost savings for an enterprise
Moderate	Provides incremental improvements to established processes that will result in increased revenue or cost savings for an enterprise
Low	Slightly improves processes (for example, improved user experience) that will be difficult to translate into increased revenue or cost savings

Source: Gartner (July 2023)

Table 4: Maturity Levels

Maturity Levels ↓	*Status* ↓	*Products/Vendors* ↓
Embryonic	■ In labs	■ None
Emerging	■ Commercialization by vendors ■ Pilots and deployments by industry leaders	■ First generation ■ High price ■ Much customization
Adolescent	■ Maturing technology capabilities and process understanding ■ Uptake beyond early adopters	■ Second generation ■ Less customization
Early mainstream	■ Proven technology ■ Vendors, technology and adoption rapidly evolving	■ Third generation ■ More out-of-box methodologies

Maturity Levels ↓	*Status* ↓	*Products/Vendors* ↓
Mature mainstream	■ Robust technology ■ Not much evolution in vendors or technology	■ Several dominant vendors
Legacy	■ Not appropriate for new developments ■ Cost of migration constraints replacement	■ Maintenance revenue focus
Obsolete	■ Rarely used	■ Used/resale market only

Source: Gartner (July 2023)