Networking connectivity options on AWS for SaaS offerings

AWS Prescriptive Guidance

AWS Prescriptive Guidance: Networking connectivity options on AWS for SaaS offerings

Table of Contents

Networking connectivity options on AWS for SaaS offerings

Tomas Sykora and Luca Schumann, Amazon Web Services

September 2025 (document history)

This guide explores common scenarios for connecting consumer applications to software as a service (SaaS) providers. It discusses how to connect to resources that are on-premises, in the AWS Cloud, in other cloud service provider (CSP) clouds, or in hybrid architectures. These scenarios include the following:

- Exposing web services over HTTPS
- Exposing TCP-based services
- Using AWS AppSync to implement publish-subscribe (Pub/Sub) and GraphQL APIs
- Using AWS resources to expose WebSockets for real-time applications
- Enabling bi-directional access for interactive service communication

By aligning with the best practices covered in this guide, SaaS providers can drive customer trust and support scalable, secure, and resilient access to SaaS offerings.

This guide also includes self-assessment criteria to help you evaluate how successfully you are meeting consumer networking requirements for your SaaS offering. Beyond connectivity patterns, you'll find comprehensive comparisons of AWS networking services, high-level architectural diagrams for various deployment scenarios, and practical guidance for how to select the right approach based on your specific business context. The guide explores security considerations for each networking option, discusses common pitfalls to avoid, and provides implementation recommendations that balance technical requirements with operational efficiency. Additionally, you'll find strategic frameworks for aligning your networking decisions with your business model, growth objectives, and regulatory compliance needs.

Intended audience

This guide is intended for SaaS providers. It helps cloud architects, product managers, and network engineers who are designing, implementing and optimizing network connectivity for

SaaS offerings in the AWS Cloud. To understand the concepts and recommendations in this guide, you should be familiar with AWS fundamentals, core SaaS concepts, and high-level networking principles.

Objectives

This guide discusses network architecture options and field-tested best practices that help consumers optimize access to SaaS offerings. Implementing the recommendations in this guide supports the following:

- **Ease of integration –** Provide an uncomplicated customer journey from onboarding to production so that you can accelerate your customers' time to value and shorten their revenue recognition cycle.
- **Adaptability –** Seamlessly integrate with your customers' existing network infrastructures by adapting to their evolving needs. This enhances your product's value proposition.
- **Total cost of ownership –** Standardize network access to reduce change costs and costs per tenant. By improving deployment consistency, you can also reduce the time to perform root cause analysis or repair.
- **Dependency management –** Understand dependencies, long-term implications, and trade-offs of the different network access options. This helps product leaders make well-informed product decisions.
- **Composability and extendibility –** Decouple the development of core functionality from operational infrastructure. This helps development teams move faster and focus on creating value for your customers.
- **Drive trust –** By providing resilient, fault-tolerant, secure, and scalable access to SaaS offerings, you can reduce regulatory risks and earn trust in your ability to support your customers' growth.

Assessing network access decisions for SaaS offerings

Understanding your market

The decisions you make now about networking determine whether the value proposition of your SaaS product can be delivered to your customers. Despite the strategic importance of these decisions, providing access to your SaaS offering is often perceived as a purely technological topic. The risk this perception carries includes prolonged revenue recognition cycles, operational inefficiencies, and misalignment with business strategy. For instance, if rapid expansion is a strategic business objective, then a guiding light of your decision-making process should be whether the solutions you are considering are scalable and flexible enough to support the expansion. Even if you are successful in growing your business, operational overhead must not become a roadblock for future growth, and a misaligned cost structure could consume all of your profits.

For example, consider how the following market considerations affect technical aspects of the product, such as networking:

- If your business model is subscription-based, your customers are likely to prefer solutions with predictable, recurring costs rather than large, upfront investments.
- If your business strategy targets high-value, enterprise-level customers, then security, governance and regulatory compliance criteria determine whether your SaaS offering will even be considered.
- If your target market is mostly startups, ease of integration, time to value, and adaptability are likely important factors. Startups typically prioritize speed and agility. Because they need to build a brand and need to generate profits quickly, they are likely to prefer solutions that are fast and easy to integrate, can cost-effectively scale, reduce dependency on experts, and do not tie up precious cycles.
- Some businesses require stable, high-throughput, and low-latency access. This includes the entertainment and media industry, manufacturing, and financial transaction processing. If these are your target customers, reliability is their primary concern.

In all of these cases, customers might perceive an otherwise healthy SaaS offering if networking access isn't seamless. If networking becomes an obstacle, this doesn't support your business case.

If your customers can't reliably access the services you offer, the value proposition of your SaaS offerings is nil.

Understanding your role

Your role in supporting business objectives depends on who you are, what your specific individual and team objectives are, and who your customers are, and what's important to them. Even if you are not part of a team that typically interacts with customers, you need to be concerned with who they are and what they need. Engineering and development teams must also be concerned with their internal customers, especially those who they interact with on a regular basis. Typically, these are the operations and customer success teams.

If you are part of a sales organization, it's essential that you communicate with product and engineering teams about networking, even though it's a seemingly pure technology topic. Share insights about the structure of the target market. Communicate pain points and the needs of your existing and potential customers and partners. Share data and anecdotes about missed opportunities, predicted growth per segment, and events. Ask questions that challenge your organization's capability to support business growth. This increases the number of opportunities and improves the long-term profitability of your business. Ultimately, this helps your organization fund future expansion and development.

If you are part of the engineering organization, understand the business strategy of your organization before attempting to draft a solution. Alignment with the business strategy helps you choose the right metrics to evaluate different network access options. It can also prevent an expensive, large-scale network redesign as your organization grows. Business alignment helps your team secure and retain the resources required for future challenges. Your team's headcount, budget for professional development, or access to cutting-edge technology is going to depend on your ability to demonstrate business alignment. Ideally, you can show how your decisions contributed to the business success of the organization. Therefore, we suggest that you capture the decision-making process, including metric selection criteria. Periodically review your metrics to confirm that they align with business objectives. This can help your team get the credit that they deserve. Periodic reviews also help validate that your team isn't making decisions based on assumptions or obsolete, historical reasons.

The list of metrics in the following sections is relevant to networking access:

- Product and commercial metrics
- Engineering metrics that influence networking decisions

This guide uses a subset of these metrics throughout to help you identify the optimal network access approaches for your SaaS offerings. Choose the metrics that are most important and relevant to your business, and then evaluate the approaches based on those metrics.

Product and commercial metrics that influence networking decisions

Product and commercial teams use success criteria to evaluate whether they are meeting business objectives. This section describes the product or commercial metrics that can be positively or negatively influenced by the networking access decisions that your organization makes.

Use these metrics and self-assessment questions to evaluate how your network access approach aligns with your business positioning and market strategy. This assessment helps you determine if your current networking decisions support your company's market differentiation, competitive advantages, and target audience needs.

This section contains metrics and self-evaluation questions for the following topics:

- Business model and market positioning
- Total addressable market, new client acquisition rate, growth, and scalability
- Customer experience and retention
- Efficiency and financial performance
- Regulatory compliance and risk management
- Partner strategy

Business model and market positioning

These metrics relate to your company's position in the market, including competitive differentiation, market reach, and brand perception. It is critical that you assess the alignment between the network access approach and the business model. Perform an assessment regardless of whether it's subscription-based, usage-based, freemium, tiered, marketplace, API-first, or white-labeled. Make sure that the model supports the organization's goals and the goals of customers.

High-score criteria

The network access approach seamlessly aligns with the business model. It eases adoption and delivery of the service. It supports the long-term financial viability of the business model, and the cost structure is compatible with expected growth. It minimizes any friction for customers or partners when adopting the offering. This enhances the user experience and encourages broader uptake of the service.

Low-score indicators

The selected network access approach is misaligned with the business model it should support. The cost structure and lead time to deployment represent a blocker for adoption in the target market. The ongoing infrastructure and operational costs inhibit any potential profits. This prevents business growth and makes it difficult to operate at the intended scale. Alternatively, the properties of the network access approach might prevent customers from considering the service due to regulatory reasons.

Self-assessment questions

- What are the cost implications of the selected network access approach for initial deployment and ongoing delivery? What are the fixed and variable costs of the approach?
- Can the network access approach scale effectively and efficiently to meet the growth demands of the business model? Consider the individual tenant size and the number of onboarded tenants.
- Does the network access approach impose any technical or operational limitations that could limit the flexibility or adaptability of the business model?
- For the network access approach, how does the deployment lead time align with the speed to market that the business model requires?

Total addressable market, new client acquisition rate, growth, and scalability

It is crucial that you assess the impact of networking decisions on the organization's capacity to expand into new markets, acquire customers effectively, and maintain operational scalability. These factors affect conversion rates. They also influence whether the network access approach supports expansion into significant market segments or limits you to only serving specific customer types.

High-score criteria

The network access approach helps the organization to reach a significant portion of the target market, or it can be effectively combined with other network approaches to extend market reach. This approach should require minimal additional integration effort. The approach supports short lead times for deployment, rapid market entry, and expansion. It allows for a high number of parallel deployments. Integration is straightforward for customers, which lowers the barriers to adoption and enhances the customer experience. The approach minimizes operational overhead, preserves operational capacity, and supports growth projections.

Low-score indicators

The network access approach supports only a small part of the target market or is suited primarily to niche segments that are not prioritized in the business strategy. It does not effectively complement other, already supported network access approaches. Lead times for deployment lag market demands, which limits market expansion and new client acquisition. The deployment model is sequential, which increases the risks of service bottlenecks as demand grows. Complex integration processes deter potential clients, which negatively affects the acquisition rate and conversion rates. Significant operational overhead diminishes the organization's operational capacity. This becomes a blocker for projected growth.

For these indicators, evaluate whether introducing a new network access approach can help the organization reach its strategic business objectives. Consider whether the new network access approach might create new product dependencies or consume operational resources without delivering the desired outcomes.

Self-assessment questions

- Are there any gaps in the current approach that prevent you from reaching larger segments of the target market?
- What is the minimum set of non-overlapping, standardized, list of network access approaches that you should support to cover 70–90% of the target market?
- What reach does each network access approach enable, and what are the related increases in important metrics, such as infrastructure costs, operational cycles, and dependency on experts?
- How do the deployment capabilities and service limits of the network infrastructure align with the growth expectations in your target markets?
- Does network integration create any barriers to entry for new customers? How can these be addressed to improve conversion rates?
- How does the operational overhead of managing the network affect your capacity for growth and scalability?
- What strategies can you implement to reduce lead times for network deployment and improve market expansion and customer acquisition?

- Are there any dependencies on expert resources that would delay deployment or integration with customer ecosystems?

Customer experience and retention

Metrics in this section help you understand your organization's ability to acquire and, most importantly, retain customers. Understanding the relationship between networking access approaches and customer satisfaction can help product and engineering teams make decisions that are informed by data.

High-score criteria

The network access approach is reliable and easy to manage. It contributes to high customer satisfaction (CSAT) and net promoter score (NPS) results. These scores are indicative of a strong brand reputation and customer loyalty. Thanks to seamless integration with your customers' existing ecosystems, adoption friction is low, and there is a low dependency on experts. Your organization consistently meets service-level agreements (SLAs), which reinforces customer trust and contractual obligations. Because customers enjoy stable and dependable services, you have high customer retention.

Low-score indicators

Difficult integration and inconsistent access to services commonly leads to customer frustration and negative feedback. This damages the brand reputation. New customers fail to convert from free or trial plans to paid services due to a dependency on experts or because of prolonged onboarding and integration times. Frequent failures to meet SLAs results in financial penalties and a loss of credibility, potentially reducing customer retention rates.

Self-assessment questions

- How does network performance (such as speed, uptime, and latency) directly affect CSAT and NPS results? What specific network improvements could drive these scores higher?
- How do current network latency and uptime metrics affect the initial user experience and adoption rates? What specific network performance improvements are required to optimize these metrics?
- Are there any recurring issues in network configurations or security settings that complicate integration for new customers? How can you streamline these processes?

- How does the ease of configuring network access affect the onboarding experience for new users? Are there specific network access points or lead times that can be optimized to enhance initial user impressions?
- What are the challenges to automating network service provisioning for new clients. How can you adjust this process to improve scalability and reliability?
- Analyze the root causes of recent SLA breaches. Were they related to network configuration, capacity planning, or external vendor issues?
- How often do network issues cause you to miss SLA commitments? What are the most frequent network-related failures?
- Which network performance improvements have shown the most significant positive impact on customer satisfaction in the past?

Efficiency and financial performance

This category assesses the financial health and profitability aspects of your business, such as cost efficiency, long-term viability, profitability, return on investment (ROI), and total cost of ownership (TCO). By streamlining network operations through standardization, you can reduce operational overhead and maintenance costs. This supports your organization's growth objectives.

High-score criteria

The cost structure of the network access approach is aligned well with the business model. It supports sustainable growth, and the significant cost savings that you achieve increase profitability. Efficient network access enables rapid customer onboarding, which shortens the time to deliver value and accelerates market penetration. This directly shortens the revenue recognition cycle.

Low-score indicators

Customers are turning to your competition in order to accelerate delivery of their applications and services. Your organization has increased operational costs associated with complex and varied network configurations and extended lead times. The cost structure and business model are misaligned, which might cause high upfront costs for subscription-based services. Cumbersome onboarding processes reduce market penetration and postpone revenue recognition.

Self-assessment questions

- What are the current lead times for new service deployment, and how do they affect the time to market and revenue recognition?

- How effectively do standardized network operations reduce overhead and maintenance costs?
- Are expert resources required to successfully complete the initial integration, operate on a daily basis, troubleshoot issues, or implement changes?
- How sustainable are current network investments in terms of technological advancements? Are you investing in future-proof technologies that align with projected market developments?
- How effectively do you allocate and track costs related to network traffic and usage by individual tenants?

Regulatory compliance and risk management

It's fundamentally important to validate compliance with network-related regulations. This confirms that you are operating legally and can maintain customer trust. Standardization across network operations simplifies the compliance process and promotes consistency across various jurisdictions and geographies. These measures help you expand your services.

High-score criteria

Network operations consistently adhere to legal standards without complications, which contributes to market expansion, decreases adoption friction, and enhances customer trust. Demonstrated compliance with critical regulatory frameworks, such as Digital Operational Resilience Act (DORA) and National Institute of Standards and Technology (NIST), helps you win customers that are sensitive to regulatory compliance. Continuous visibility into your compliance status reduces the time required to complete an audit.

Low-score indicators

Gaps in in network compliance cause high adoption friction, service launch delays, legal challenges, and potential fines. These challenges lead to delayed or canceled plans for expansion into new markets. It is difficult to maintain standard compliance practices across different jurisdictions, and this affects operational efficiency and market reputation.

Self-assessment questions

- How well are your network operations aligned with applicable regulatory or industry guidelines? What did your must recent compliance audit reveal?
- How are you maintaining compliance with emerging regulations in the digital and network security realms?

- How effective is your documentation and reporting process in meeting the requirements of different regulatory bodies?
- What risk management strategies do you have in place to identify and address potential compliance risks before they lead to legal challenges?
- What level of compliance training and awareness do your network management teams require to support your network access approaches?

Partner strategy

Assess how well the network access approach aligns with an ecosystem of recognized partners, platforms, and marketplaces. This is essential, especially if your growth strategy depends on scaling through partners.

High-score criteria

The network access approach is integrated across your partner ecosystem. Its cost structure aligns well with the business models of your key partners. Partners possess the necessary networking skills for seamless integration of your SaaS offerings, and they can deliver sustained access and functionality.

Low-score indicators

The selected network access approach demands specialized skills, resources, or equipment that are scarce or hard to procure. It differs from the standard network access protocols that are commonly used by platforms and marketplaces. This results in an unpredictable cost structure that is challenging to reconcile. The network access approach is misaligned with the business models of your key partners.

Self-assessment questions

- What are the cost implications of the network access approach for partners. How do these costs align with their business models? Which side of the integration bears the bulk of the costs, and how many operational cycles must be invested?
- For the network access approach, are there any barriers to integration or maintenance that could affect partner relationships or ecosystem scalability?
- How can the network access approach be optimized to enhance compatibility and ease of integration across the ecosystem?

Engineering metrics that influence networking decisions

Like product and commercial teams, engineering teams also use success criteria to evaluate whether they are meeting business objectives. However, these metrics differ and they focus on the team's ability to develop, operate, and meet security and compliance requirements. This section describes engineering metrics that can be positively or negatively influenced by the networking access decisions that your organization makes.

Use these metrics and self-assessment questions to evaluate your current network access approach against your business requirements and technical capabilities. This assessment helps you identify gaps in your architecture and prioritize improvements that align with your strategic objectives. By regularly reviewing these criteria, you can make sure that your network access strategy continues to support both your customers' needs and your organization's growth plans.

This section contains metrics and self-evaluation questions for the following categories and topics:

- Development metrics
  - Deployment frequency, time to deploy, and sprint velocity
  - Flexibility and feature delivery
  - Change failure rate
  - Code quality and engineering team performance
  - Technical debt reduction
  - Scalability, capacity, and performance
- Operational excellence metrics
  - Operational resilience and disaster recovery
  - Service and application performance monitoring
- Security and governance metrics
  - Security, compliance, and vulnerability management

Development metrics related to network access for SaaS offerings

This section contains the following metrics:

- Deployment frequency, time to deploy, and sprint velocity
- Flexibility and feature delivery

- Change failure rate
- Code quality and engineering team performance
- Technical debt reduction
- Scalability, capacity, and performance

Deployment frequency, time to deploy, and sprint velocity

To optimize the efficiency of the development cycle, it's essential that you understand the influence of network stack provisioning on sprint velocity.

High-score criteria

Network stack provisioning is streamlined and automated, and it requires minimal manual intervention. It does not significantly impact sprint velocity. Network stack provisioning and redeployment can be performed by any team member. This reduces bottlenecks and dependencies on specialized resources.

Low-score indicators

A high number of story points are necessary for provisioning the network stack. This suggests a complex and time-consuming process that detracts from the development of new features. Frequent redeployment of the network stack incurs substantial time and cost overheads. Network provisioning tasks require specialized engineering expertise, which creates bottlenecks and slows the development cycle.

Self-assessment questions

- What manual steps, if any, are involved in the deployment process. How do they impact the deployment frequency and time?
- How are rollbacks handled in case of deployment failures. What is their impact on deployment frequency and recovery time?
- How many story points are required for provisioning the network stack when you set up new environments?
- How much additional costs and time overhead are associated with frequent redeployment of the network stack during the development process?
- Does provisioning the network stack depend on specialized engineering expertise, or is it a task that can be managed by any team member?

Flexibility and feature delivery

The network access approach can influence the engineering team's ability to innovate and deploy new features efficiently.

High-score criteria

The network access approach offers the flexibility needed for rapid and seamless feature deployment. It supports a wide range of communication protocols, unidirectional and bidirectional communication, and message sizes. It does not impose significant constraints on development processes or innovation.

Low-score indicators

The network access approach restricts the team's ability to roll out new features due a lack of supported communication protocols, inflexibility in message sizes, or dependency on specific technologies and related expert resources. This can lead to slower development cycles and hinder the service's evolution.

Self-assessment questions

- How does the network access approach impact the team's agility in developing and deploying new features?
- Are there limitations in the network access approach that restrict the support of certain communication protocols or technologies?
- How does the approach facilitate or limit the integration of new technologies and innovations into the service?
- How does the network access approach affect development timelines and the product roadmap?

Change failure rate

The network access approach you choose can affect the change failure rate when deploying new services or features. Greater control often means greater flexibility, but it also increases the potential for misconfigurations, such as when managing a complex routing setup.

High-score criteria

You can implement changes to the network stack with minimal risk of failure. Sufficient testing mechanisms are present, efficient rollback mechanisms exist, and effective monitoring helps you to quickly identify and resolve issues.

Low-score indicators

The network access approach is prone to failures during changes. There are limited testing options, complicated deployment strategies, or insufficient monitoring and troubleshooting capabilities. Multiple parties are required to participate in troubleshooting sessions. This can lead to increased downtime and decrease the availability of the SaaS offering.

Self-assessment questions

- What measures are in place to mitigate the risk of change failure when updating the network stack?
- Are there thorough testing and validation processes?
- How quickly can the system recover from a failed change? Is there an efficient rollback process in place?
- Are there proactive monitoring and alerting systems to detect and address issues swiftly during and after network stack changes?
- What is the historical change failure rate for network stack deployments. What lessons have been learned from past incidents?
- How does the network access approach facilitate or limit change implementation. Does the approach minimize service disruption?
- What is the risk of impacting the availability of the SaaS offering in the production environment when you deploy changes that involve the network access approach?

Code quality and engineering team performance

Network access approaches can indirectly affect code quality for SaaS offerings. A lack of standardization in network access can compel the engineering team to support multiple integration approaches, which can lead to a bloated codebase. This, in turn, can hinder the team's ability to develop the depth and control over code quality that is necessary to maintain high-performing engineering teams.

High-score criteria

The engineering team stays focused thanks to code modularity and reusability across supported network access approaches. The network access approaches are compatible with existing deployment pipelines and automated testing strategies.

Low-score indicators

The engineering team performance is reduced due to overhead that is associated with the integration and maintenance of too many network access approaches. Some approaches significantly increase complexity, generate tech debt, or require development of workarounds to address missing or insufficient capabilities.

Self-assessment questions

- How does the network access approach manage network variability?
- Do you need to develop additional code for handling disruptions in connectivity?
- Is a new network access approach seamlessly integrate with existing approaches, or does it require significant custom development?
- What is the extent of the change needed to adopt a new network access approach? Can the existing codebase and automated tests be used effectively?
- How easy or difficult is it to deploy or redeploy the service with the selected network access approach? Can this be done frequently? Are there any dependencies on expert resources?
- Does the network access approach facilitate or complicate adherence to coding standards and best practices?
- How does the approach affect the time-to-market for new features or fixes?

Technical debt reduction

An evaluation of a network access approach's impact on technical debt should consider its scalability, observability, and security capabilities.

High-score criteria

The approach effectively streamlines infrastructure management as the customer base expands. It offers robust observability capabilities out-of-the-box. This promotes efficient monitoring and maintenance.

Low-score indicators

The network access approach inadequately secures communication channels and lacks sufficient tools for qualitative metric observation. It might also require additional development for infrastructure management as the customer base increases, or it might necessitate workarounds for reliability issues.

Self-assessment questions

- How does the network access approach influence the long-term scalability of the infrastructure? Does it facilitate seamless growth with minimal additional investment?
- How comprehensive are the included observability tools? Do they allow for proactive monitoring and issue resolution?
- What is the anticipated impact of the network access approach on the maintenance and evolution of the codebase over time?
- Does the approach integrate well with existing and planned infrastructure. Does it require significant changes or additions?

Scalability, capacity, and performance

To determine the suitability of a network access approach for a SaaS offering, it's essential to analyze how it maintains optimal performance as demand increases.

High-score criteria

The network access approach seamlessly facilitates expansion. It maintains low latency during request processing, and it efficiently handles traffic spikes. It provides consistent performance regardless of increased traffic levels, and it doesn't impose operational limits on growth.

Low-score indicators

The network access approach doesn't scale effectively, possibly due to inherent bandwidth limitations or insufficient infrastructure capacity. Resource provisioning and management increase the complexity or create dependencies. Service performance is degraded due to increased latency, jitter, and throughput variability, particularly in congested network conditions.

Self-assessment questions

- How does the network access approach accommodate an increasing number of tenants and their data volumes?
- Is it inherently scalable to meet future demands?
- What measures are in place to make sure performance is consistent, even during peak traffic periods or rapid scaling events?
- How does the approach handle network latency and jitter? Are there mechanisms to optimize data throughput and minimize delays?

- Can the network access approach adapt to varying network conditions? Can it provide a single-tenant experience for every customer?
- What is the impact of the network access approach on the underlying infrastructure? Does it require significant upgrades or changes to existing systems?

Operational excellence metrics related to network access for SaaS offerings

This section contains the following metrics:

- Operational resilience and disaster recovery
- Service and application performance monitoring

Operational resilience and disaster recovery

The network access approach should help the SaaS offering withstand various types of disruptions and quickly recover from any disasters.

High-score criteria

Established and tested disaster recovery plans consistently show that the network access approach meets the disaster recovery requirements. The network access approach supports high-availability configurations, and it supports automatic, quick, and reliable failover mechanisms.

Low-score indicators

The network access approach makes it difficult to build a coherent disaster recovery strategy. You observe prolonged recovery times after disruptions. Frequent operational failures of the network infrastructure are impacting service delivery.

Self-assessment questions

- When was the last disaster recovery drill, and what were the outcomes?
- How long does it take to recover critical services after a disruption? What portion of the network infrastructure needs to be redeployed?
- What improvements can be made to the network infrastructure to streamline your disaster recovery plans?

- Are redundancies in place for the most critical network components?
- Have you automated the potential redeployment of network infrastructure after a critical outage?
- How does the network access approach support fault tolerance and reliability? Are there built-in mechanisms to handle network interruptions and maintain data integrity?

Service and application performance monitoring

The networking access approach can affect the performance monitoring tools that are used to validate optimal operation and service uptime. Depending on the service, you might have access to low-level metrics (such as packet drop rates) or higher-level metrics (such as session duration). Low-level metrics provide detailed technical insight into network behavior but can be complex to interpret. In contrast, higher-level metrics often offer a more direct and easier way to gauge overall user experience. This is because they aggregate the impact of underlying network conditions into clear indicators of service quality.

High-score criteria

Comprehensive monitoring tools that provide near real-time insights are readily available. You have automated alerts and response systems that address performance issues. You can predict potential service bottlenecks or failures before they affect users.

Low-score indicators

Frequent service interruptions or performance issues happen without being observed or acted upon. The lack of visibility into service performance results in slow response to performance bottlenecks. Multi-party teams are required to troubleshoot network infrastructure issues.

Self-assessment questions

- Which monitoring tools and network infrastructure metrics are currently available? How effective are they at detecting service anomalies?
- How quickly can you identify and resolve performance issues?
- Do you have mechanisms in place that predict potential performance problems?
- What improvements can you make to enhance observability capabilities?

Security and governance metrics related to network access for SaaS offerings

This section contains the following metrics:

- Security, compliance, and vulnerability management

Security, compliance, and vulnerability management

It is critical that you evaluate the security aspects of the network access approach, including compliance with security standards and the management of vulnerabilities.

High-score criteria

The network access approach helps your team adhere to security frameworks, such as International Organization for Standardization (ISO) 27001, System and Organization Controls 2 (SOC 2), or NIST. It makes it easy to conduct regular security audits. Strong encryption and authentication mechanisms are in place. Networks are isolated, and only the necessary resources are exposed to the customer's infrastructure. You can spot networking anomalies in near real-time, without excessive overhead.

Low-score indicators

The network access approach is prone to recurrent security breaches or vulnerabilities, and it is not compliant with key security standards. You frequently observe delayed detection and responses to security incidents.

Self-assessment questions

- Are there any recent security breaches linked to selected network access approach, and what have we learned from them?
- How does your network access approach comply with global security standards?
- How long does it take to detect and respond to security threats? How does the network access help or limit this ability?
- How frequently are security assessments conducted on the network access approaches? Can you use commonly tooling to assess the security of the network access approach, or is specialized software required?
- What level of security is inherent in the network access approach, and how does it align with industry best practices and regulatory requirements?

Overview of AWS networking services for SaaS offerings

This section discusses the AWS networking services that are referenced in this guide. It also compares their capabilities and describes security considerations for each service.

This section contains the following topics:

AWS networking services

The following are the AWS services that are discussed consistently in this guide.

AWS PrivateLink

AWS PrivateLink is a cloud-native service that can provide access to your SaaS offering if your customers are already operating in the AWS Cloud. Your customer connects to the SaaS offering through a interface VPC endpoint. This is an endpoint network interface that is provisioned in one or more subnets in the customer's AWS account. In the scenarios in this guide, the traffic travels through the interface VPC endpoint and arrives at an Network Load Balancer in your account. The Network Load Balancer forwards the traffic to the SaaS application, which you have registered as an *endpoint service*. Through resource VPC endpoints, AWS PrivateLink can also help you access other resources, such as databases.

Amazon VPC Lattice

Amazon VPC Lattice is an application networking service that helps SaaS providers to securely and efficiently offer their services to customers who are operating across multiple VPCs and AWS accounts. Customers access your SaaS offering through VPC Lattice, which delivers consistent network connectivity, robust access controls, and advanced traffic management. In these scenarios, traffic flows through VPC Lattice to your registered application services. It provides scalable and secure communication, regardless of which compute service you use.

VPC peering

VPC peering is a networking connection between two virtual private clouds (VPCs) that routes traffic between them by using private IPv4 addresses or IPv6 addresses. VPC peering is typically used between trusted entities, like those within the same organization. Your customer creates a peering request to one of your VPCs. When you accept it, traffic can flow between both VPCs in either direction. This connection approach stands out for its uniqueness because it involves direct communication between two VPCs without any intermediary service or infrastructure to manage.

AWS Transit Gateway

AWS Transit Gateway is a centralized network transit hub that can connect VPCs, virtual private network (VPN) connections, AWS Direct Connect gateways, third-party virtual appliances in a VPC, and other transit gateways. A transit gateway can have a different route table for each attachment. This provides maximum flexibility for routing, and it helps you isolate the networks. It's often used to connect many VPCs together or for centralized inspection.

AWS Site-to-Site VPN

AWS Site-to-Site VPN can use internet protocol security (IPsec) technology to establish connections between on-premises networks, remote offices, factories, other cloud providers, and the AWS global network. The connection is established from a virtual private gateway or transit gateway in a VPC in the AWS Cloud to a physical or software-based customer gateway, which can be in the AWS Cloud, on-premises, or in another CSP's cloud. The connection can be through the Internet or through a physical AWS Direct Connect connection. It is also possible to have an accelerated Site-to-Site VPN connection by using AWS Global Accelerator. An accelerated connection routes traffic to an AWS edge location, and it offers reduced latency and improved performance.

AWS Direct Connect

AWS Direct Connect establishes a high-speed, private connection between an on-premises data center and the AWS Cloud. By bypassing the public internet, AWS Direct Connect provides a more reliable, secure, and consistent low latency connection to the AWS Cloud. Customers connect to one of the AWS Direct Connect locations and then choose either a hosted or a dedicated connection to AWS. Although this is an uncommon architecture choice for SaaS offerings, it can be well suited for SaaS providers that have few but large enterprise consumers.

Comparing service capabilities

The following table outlines the supported capabilities of the AWS services that are discussed in this guide. The following are descriptions of the capabilities included in this table:

- **Overlapping CIDR ranges** – Can connect two or more networks with the same or overlapping CIDR ranges
- **Bidirectional communication –** Can support a two-way communication channel so that the SaaS consumer can expose internal resources, such as a database, to the SaaS provider
- **IPv6 –** Can support IPv6, either single or dual-stack
- **Jumbo frame –** Can support jumbo frames with a frame size up to 8,500 bytes
- **Hybrid-cloud** – Can support a connection with an on-premises network
- **Multi-cloud** – Can support a connection between networks on different cloud service providers

Service or approach	Overlapping CIDR ranges	Bidirectional communication	IPv6	Jumbo frame	Hybrid cloud	Multi-cloud
VPC peering	No	Yes	Yes	Yes[5]	No	No
AWS PrivateLink	Yes	Yes[1]	Yes	Yes	No[6]	No[6]
Amazon VPC Lattice	Yes	Yes[1]	Yes	Yes	No[6]	No[6]
AWS Transit Gateway	No	Yes	Yes	Yes	Yes[3]	Yes[3]

AWS Site-to-Site VPN	No	Yes	Yes	No	Yes	Yes
AWS Direct Connect	No	Yes	Yes	Yes[2]	Yes	Yes
Public internet access[4]	Not applicable	No	Yes	Yes	Yes	Yes

1. With VPC resources in Amazon VPC Lattice
2. Only for private and transit virtual interfaces
3. With Site-to-Site VPN or AWS Direct Connect attachments
4. As a general term for AWS resources that make an application publicly accessible, such as an Application Load Balancer
5. Only for peering connections within one AWS Region
6. Possible through a preexisting Layer 3 connection between the environments

Security features and considerations

The following table outlines the security features of the AWS services that are discussed in this guide.

- **Means of authentication –** How you can make sure that only your customers can connect to your service. Another level of authentication for incoming requests is usually still required, especially in shared tenant environments.
- **Encryption in transit –** Describes whether encryption in transit is provided by default. *Native encryption* describes encryption that AWS provides for all traffic within VPCs, across VPCs, or across data centers. *Supplementary encryption* describes encryption that you control and that can be stopped by the respective service.

Service or approach	Means of authentication	Encryption in transit

VPC peering	You initiate a peering request to the AWS account and VPC of your customer or accept a request that they initiate. See Accept or reject a VPC peering connection.	Native encryption only
AWS PrivateLink	You choose which AWS accounts are allowed to create endpoints to your service. These accounts are known as *allowed principals*. See Accept or reject connection requests.	Native encryption only
Amazon VPC Lattice	You share a VPC Lattice service or service network with your customers' AWS accounts. See Share your VPC Lattice entities.	Native encryption and supplementary TLS encryption
AWS Transit Gateway	Your customer creates a peering attachment request from their AWS account, or you initiate the request. See Transit gateway peering attachments in Amazon VPC Transit Gateways.	Native encryption and supplementary IPsec encryption with a VPN attachment
AWS Site-to-Site VPN	You use IPsec pre-shared keys or a private certificate on the customer's device. See AWS Site-to-Site VPN tunnel authentication options.	Supplementary IPsec encryption

AWS Direct Connect	Your customer creates a virtual interface request from their AWS account. See AWS Direct Connect virtual interfaces and hosted virtual interfaces.	Supplementary Layer 2 encryption possible at selected sites. See AWS Direct Connect Locations.
Public internet access[1]	Custom authentication is required.	Supplementary TLS encryption possible

[1] As a general term for AWS resources that make an application publicly accessible, such as an Application Load Balancer

Evaluating network access options for SaaS offerings

The metrics that are important to your organization will depend on who your customers are, your business strategy, and your organizational objectives. This guide presents metrics that you can use to choose a networking access approach, but you should prioritize those that meet the unique requirements of your use case.

This section contains the following topics:

- Evaluation metrics
- Total cost of ownership
- Networking value map

Evaluation metrics

Some metrics are consistent across organizations and use cases, and these are the metrics that we can help you rate. The following are these metrics:

- **Ease of integration –** How quickly and easily can you onboard new customers?
- **Total cost of ownership (TCO) –** What is the cost structure? Beyond fixed and variable infrastructure costs, there are major additional cost considerations associated with operational overhead, dependency on experts, cost of implementing changes, and compliance. For more information, see the Total cost of ownership section.
- **Scalability –** Is your network access approach able to scale in order to support your company's growth? Scaling your customer base has important architectural and organizational considerations. Consider how you might scale to accommodate 5—100 times as many customers as you support today.
- **Adaptability –** Can you implement changes easily? Changes might include a new application, a new capability, a different platform, or a different network.
- **Network isolation –** How much of the network infrastructure are you exposing to your customers? Are you providing just the right degree of access, or are you exposing whole networks? If you isolate network resources early, it will be easier to provide security, privacy, and compliance assurances later.
- **Observability –** What's your ability to detect service failure or degradation? How easy and fast is it to identify the problem? How quickly (and with what overhead) can you help your customers understand their points of failure and help them resolve it?

- **Time to repair –** What's the lead time between the detection of a service failure or degradation and resuming operations? What are the factors that affect this ability?

Other metrics are unique to your organization or offering because they relate to your business operations, strategy, or goals. Only you can rate these metrics. The following are these metrics:

- **Business model alignment –** What is your business model, and how well do individual access approaches align with it?
- **Total addressable market (TAM) –** What is your current and future market, and how well is it covered by the network access approach?
- **Return on investment (ROI) –** What improvements do you expect in profitability and margins? Are the expected financial benefits sufficient to meet your needs for adaptable and flexible service access?
- **Regulatory compliance –** What kind of regulatory requirements apply, and in which market?
- **Service-level agreements (SLAs) –** Do customers need your SaaS offering to be highly available? What sort of commitments are you contractually obliged to uphold?

Total cost of ownership

This section explores total cost of ownership (TCO), which is one of the evaluation metrics used to compare the network access approaches. TCO is a composite metric consisting of fixed and variable infrastructure costs, operational overhead, specialist dependency, cost of change, and compliance costs.

The TCO rating for each network access approach might vary for your use case. For example, the *cost of change* for a SaaS provider with a simple web-service and five tenants differs from a SaaS provider with a complex, interconnected product portfolio and hundreds or thousands of tenants. Additionally, not all components have the same weight. For example, hiring a networking specialist is often more expensive than the infrastructure costs that support an individual deployment of your service. Use the values in the following table for initial orientation and as a reference point for further discussion.

Access approach	Fixed infrastructure costs	Variable infrastructure costs	Operational overhead	Specialist dependency	Cost of change	Compliance costs

VPC peering	None	None	High	Low	High	Medium
AWS PrivateLink	Low	Low	Low	None	Low	Low
Amazon VPC Lattice	Medium	Medium	Low	Low	Low	Low
AWS Transit Gateway	Medium	Medium	Low	Low	Low	Medium
AWS Site-to-Site VPN	Medium	High	High	Medium	Medium	Low
AWS Direct Connect	High	Medium	Medium	High	High	Low
Public internet access	Low	High	Medium	Low	Low	High

VPC peering costs

There is no direct infrastructure cost associated with a VPC peering connection. When traffic stays within the same Availability Zone, there is no data transfer charge. However, operational overhead can be significant because management and complexity grow exponentially with each additional peering connection. Some basic understanding of networking is enough to set up a peering connection, but changes on the network are difficult to implement with more than a handful of peering connections. Compliance costs are slightly higher because both parties are exposing an entire VPC to each other, rather than individual services.

AWS PrivateLink costs

AWS PrivateLink is often a cost-effective solution with small operational overhead. This is because the SaaS provider must manage only a Network Load Balancer, and the consumer must manage only VPC endpoints. You can make changes on both sides transparently, which reduces expensive and resource-intensive cross-organizational collaboration. Compliance costs tend to be low because the SaaS provider is exposing only the services that they want and not the entire network.

Amazon VPC Lattice costs

Amazon VPC Lattice offers a balanced cost structure with moderate fixed and variable infrastructure costs. As a fully managed service network, it significantly reduces operational overhead by automating service discovery, traffic management, and access controls across multiple VPCs. This simplifies both initial deployment and ongoing management compared to manual networking configurations. You can implement changes through policy-based controls without complex routing updates, which reduces the dependency on networking specialists. Compliance costs tend to be lower than traditional networking approaches because VPC Lattice provides fine-grained access controls and comprehensive visibility through built-in monitoring and logging capabilities. This can make it easier to demonstrate regulatory compliance.

AWS Transit Gateway costs

AWS Transit Gateway has larger hourly and data processing charges than AWS PrivateLink, but it has similar operational overhead. You must have deeper knowledge of the AWS Transit Gateway service and routing on AWS in order to correctly set up all the route tables. Infrastructure changes might require routing or DNS updates. Compliance costs are similar to VPC peering because both parties are potentially exposing subnetworks or entire VPCs to each other. AWS Transit Gateway route tables also need to be handled with care because they're shared by multiple consumers, and you must not allow any traffic between them.

AWS Site-to-Site VPN costs

Because Site-to-Site VPN essentially sends traffic to the internet, the variable cost is highest in comparison due to data transfer charges. Although it's a managed virtual private network (VPN) service, it comes with significant operational overhead, especially on the customer gateway. Provisioning and operations require advanced knowledge of networking, and changes often require action from both parties. Compliance costs are usually low because security teams often preapprove IPsec tunnels without additional review.

AWS Direct Connect costs

AWS Direct Connect comes with the largest fixed infrastructure cost because it is a private physical connection directly into the AWS Cloud. Specialist knowledge is required to set up and operate a Border Gateway Protocol (BGP) session (if required), to operate a VPN connection, and to perform traffic engineering. This service reduces the effort for security teams because it blends private connectivity with the option of additionally having Media Access Control Security (MACsec) and IPsec encryption.

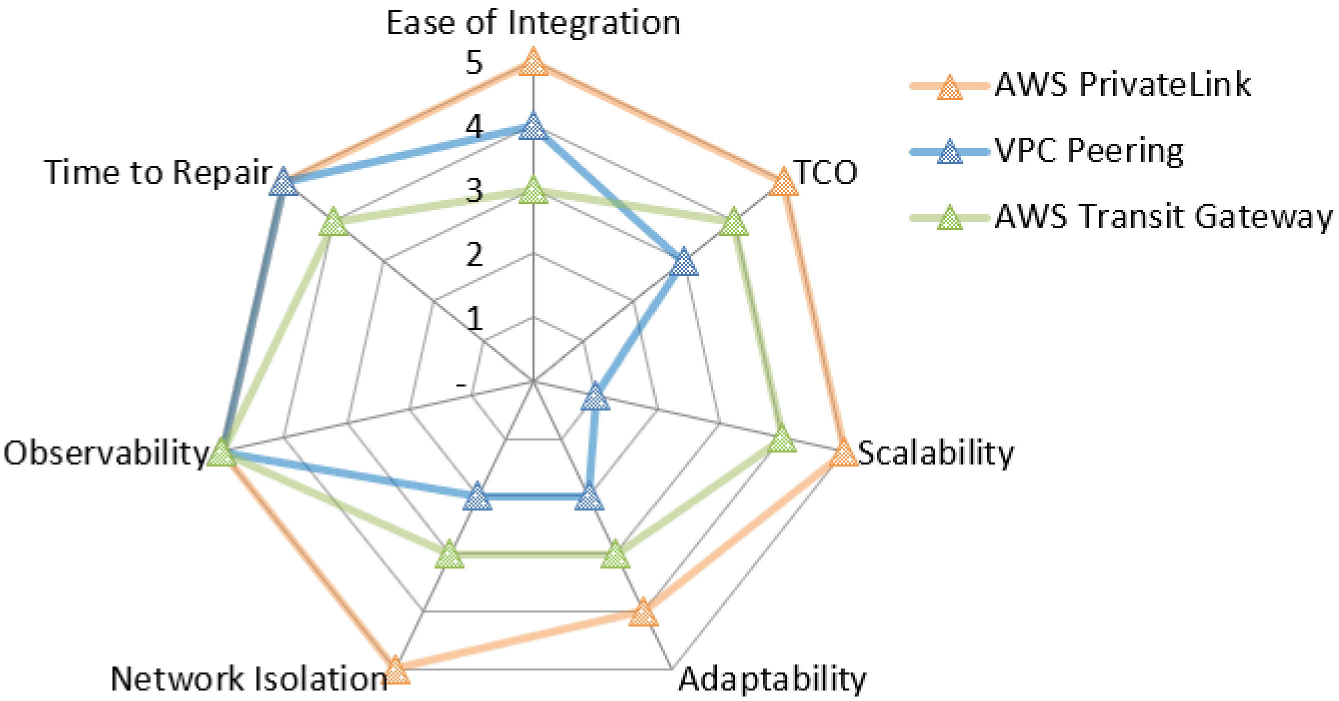
Public internet access costs

Public internet access refers to the AWS resources that you can use to make an application publicly accessible, such as an Application Load Balancer. For this approach, there are variable costs linked to providing access to your services, including charges for data transfer out to the internet. Operational overhead and compliance costs can be significant because you're exposing the service to the Internet and will require additional security and authentication mechanisms. However, there is no complex routing involved, and neither party has to know details about each other's infrastructure.

Networking value map

To help you see the big picture and make informed decisions, this guide includes a networking value map for each scenario. Because the ratings differ from scenario to scenario, the same service might score differently for two scenarios. The value maps are radar charts, where a hypothetical perfect score would be a five in all categories.

For example, the following image shows a sample radar chart. It includes only the metrics that we can help evaluate. We recommend that you create your own value map that includes the additional metrics that only you can evaluate.


Ease of Integration
Time to Repair
TCO
Observability
Scalability
Network Isolation
Adaptability
5
4
3
2
1
-
AWS PrivateLink
VPC Peering
AWS Transit Gateway

Networking access scenarios for SaaS offerings in the AWS Cloud

This section covers different network access options for your SaaS offerings in the AWS Cloud. It discusses the approaches from the perspective of your consumer, who might have connectivity needs within the AWS Cloud, from on-premises data centers, or from other cloud service providers (CSPs). Additionally, you might need to support access from multiple types of consumer environments.

Understanding the network connectivity requirements across these diverse environments is essential for creating a comprehensive access strategy. Your architectural decisions must account for varying security models, performance expectations, and technical constraints while maintaining operational efficiency. The right approach provides secure, reliable connectivity that scales with your business growth and minimizes both implementation complexity and ongoing management overhead.

When evaluating network access options, consider how each approach affects your total cost of ownership, including not just infrastructure costs but also operational overhead and compliance requirements. Some approaches excel at scalability but may introduce complexity, while others prioritize ease of integration at the expense of network isolation. Your consumers' technical capabilities and resources also play a significant role in determining the most appropriate solution.

For consumers on the AWS Cloud, services such as AWS PrivateLink offer significant advantages in security and scalability. On-premises consumers might benefit from AWS Direct Connect for consistent performance or benefit from Site-to-Site VPN for cost-effective connectivity. Multi-cloud scenarios often require careful consideration of interoperability challenges, and you might use transit VPC architectures to standardize access patterns. In all cases, your design should anticipate future consumer and traffic growth so that your network architecture remains resilient and adaptable as your SaaS offering evolves.

This section contains the following scenarios:

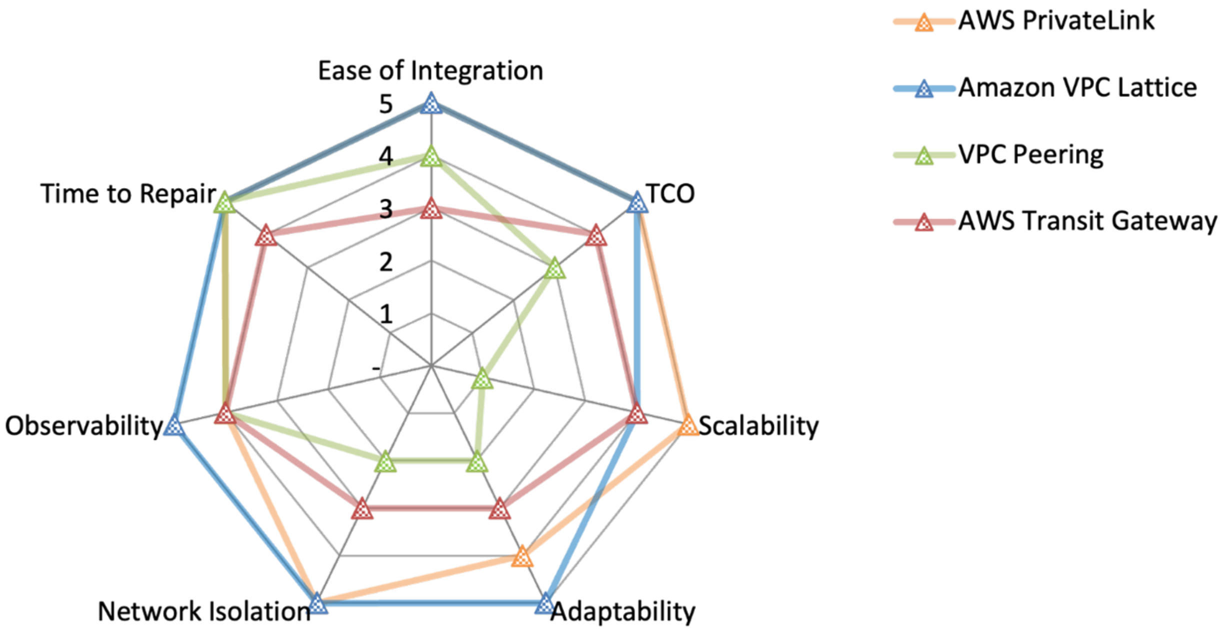
SaaS consumers operating on AWS

This section discusses connectivity options if both you and your consumers are operating in the AWS Cloud. This scenario offers the largest flexibility because many AWS services natively integrate and because both parties have access to the entire AWS service portfolio.

This section discusses the following network access approaches:

- Integrating with AWS PrivateLink
- Sharing an Amazon VPC Lattice service
- Creating VPC peering connections
- Connecting VPCs with AWS Transit Gateway

The following networking value map summarizes how each of these options scores for each evaluation metric. For more information about the evaluation metrics, see Evaluation metrics in this guide. In the map, a five represents the best score, such as the lowest TCO, best network isolation, or lowest time to repair. For more information about how to read this radar chart, see Networking value map in this guide.



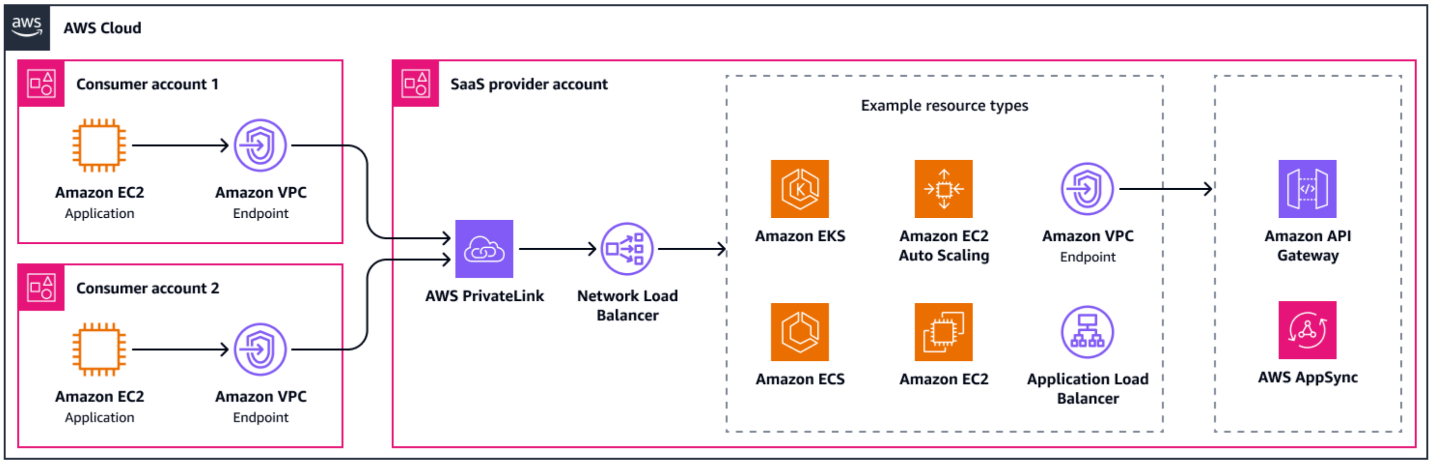
The radar chart shows the following values.

Evaluation metric	AWS PrivateLink	Amazon VPC Lattice	VPC peering	AWS Transit Gateway
Ease of integration	5	5	4	3
TCO	5	5	3	4
Scalability	5	4	1	4
Adaptability	4	5	2	3
Network isolation	5	5	2	3
Observability	4	5	4	4
Time to repair	5	5	5	4

Integrating with AWS PrivateLink

AWS PrivateLink is the most cloud-native way to integrate a SaaS offering. SaaS providers can host their application either behind a Network Load Balancer. The Network Load Balancer directly integrates with an Application Load Balancer, Amazon Elastic Container Service (Amazon ECS), Amazon Elastic Kubernetes Service (Amazon EKS), and Auto Scaling groups. It is also possible to route traffic from the Network Load Balancer to interface VPC endpoints in the SaaS provider account. This helps you use an API to reach applications, such as through Amazon API Gateway or AWS AppSync. If your application requires access to resources in the customer environment that aren't load balanced, such as a database, you can use resource VPC endpoints.

AWS PrivateLink supports a bandwidth of up to 100 Gbps per Availability Zone. The following diagram shows a basic configuration with some possible integrations. It connects two consumer accounts to the SaaS provider account through AWS PrivateLink. There are service endpoints in the consumer accounts and a Network Load Balancer in the SaaS provider account.



The following are the benefits of this approach:

- Ease of integration: No route table changes required
- Ease of integration: You can offer endpoint services through AWS Marketplace
- Ease of integration: VPC endpoints support friendly DNS names
- Scalability: It can scale to thousands of SaaS consumers
- Adaptability: Support for overlapping CIDR ranges
- Adaptability: Support for IPv6
- Adaptability: Cross-Region support
- TCO: AWS PrivateLink is a fully managed service, so it requires less operational effort
- Network isolation: Security benefit for the SaaS consumer because traffic can't be initiated from the SaaS provider
- Network isolation: Security benefit for the SaaS provider because they are not exposing an entire subnet or VPC

The following are the drawbacks of this approach:

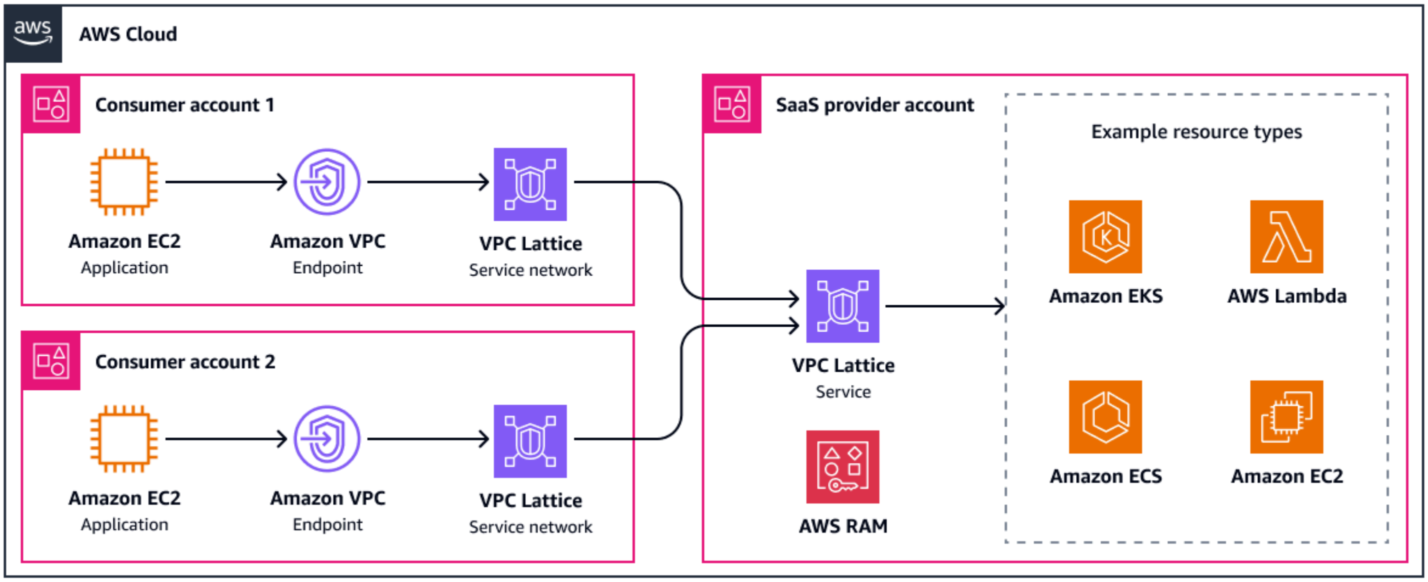
- Adaptability: SaaS provider must use the same Availability Zones as the consumer
- Adaptability: Support only for client-initiated connections, and resource VPC endpoints are required for service-initiated communication
- Adaptability: Network Load Balancer is the only direct integration for AWS PrivateLink

Sharing an Amazon VPC Lattice service

To use Amazon VPC Lattice as a connectivity option for your SaaS application, you first create one or more VPC Lattice services that represent your SaaS application components. You configure listeners and routing rules to direct traffic to your backend targets, such as Amazon EC2 instances, containers, or AWS Lambda functions. For more information, see Connecting Saas services within a VPC Lattice service network (AWS blog post). Concept-wise, this is almost the same as configuring an Application Load Balancer. Then, you share your SaaS service securely with customer AWS accounts or organizations by using AWS Resource Access Manager (AWS RAM), specifying what permissions they have. After customers accept the resource share, they can associate your SaaS service with their existing or newly created VPC Lattice service networks to enable service-to-service communication.

Each VPC Lattice service can support up to 10 Gbps and 10,000 requests per second per Availability Zone. By implementing auth policies, your customers can have fine-grained control over which services and resources can access the SaaS application. You can use resource gateways to access resources that require a TCP connection. For example, this might be an Amazon EKS cluster that you manage, or it might be a customer-managed resource that your application needs to access. For more information about using resource gateways for SaaS offerings, see Extend SaaS capabilities across AWS accounts using AWS PrivateLink support for VPC resources (AWS blog post).

The following diagram shows a high-level VPC Lattice configuration with some example integrations. It uses customer-managed service networks to access the SaaS application.



The following are the benefits of this approach:

- Ease of integration: No route table changes required
- Ease of integration: Service discovery out of the box
- Scalability: It can scale to thousands of SaaS consumers
- Adaptability: Support for overlapping CIDR ranges
- Adaptability: Support for IPv6
- Adaptability: Integrates with any AWS compute service as a VPC Lattice service
- TCO: VPC Lattice is a fully managed service, so it requires less operational effort
- TCO: Built-in load balancing with advanced traffic routing
- Network isolation: Fine-grained authorization with auth policies
- Network isolation: Security benefit for the SaaS consumer because traffic cannot be initiated from the SaaS provider
- Network isolation: Security benefit for the SaaS provider because you are not exposing an entire subnet or VPC

The following are the drawbacks of this approach:

- Adaptability: Support only for client-initiated connections, and resource gateways are required for service-initiated communication
- Adaptability: No cross-Region support

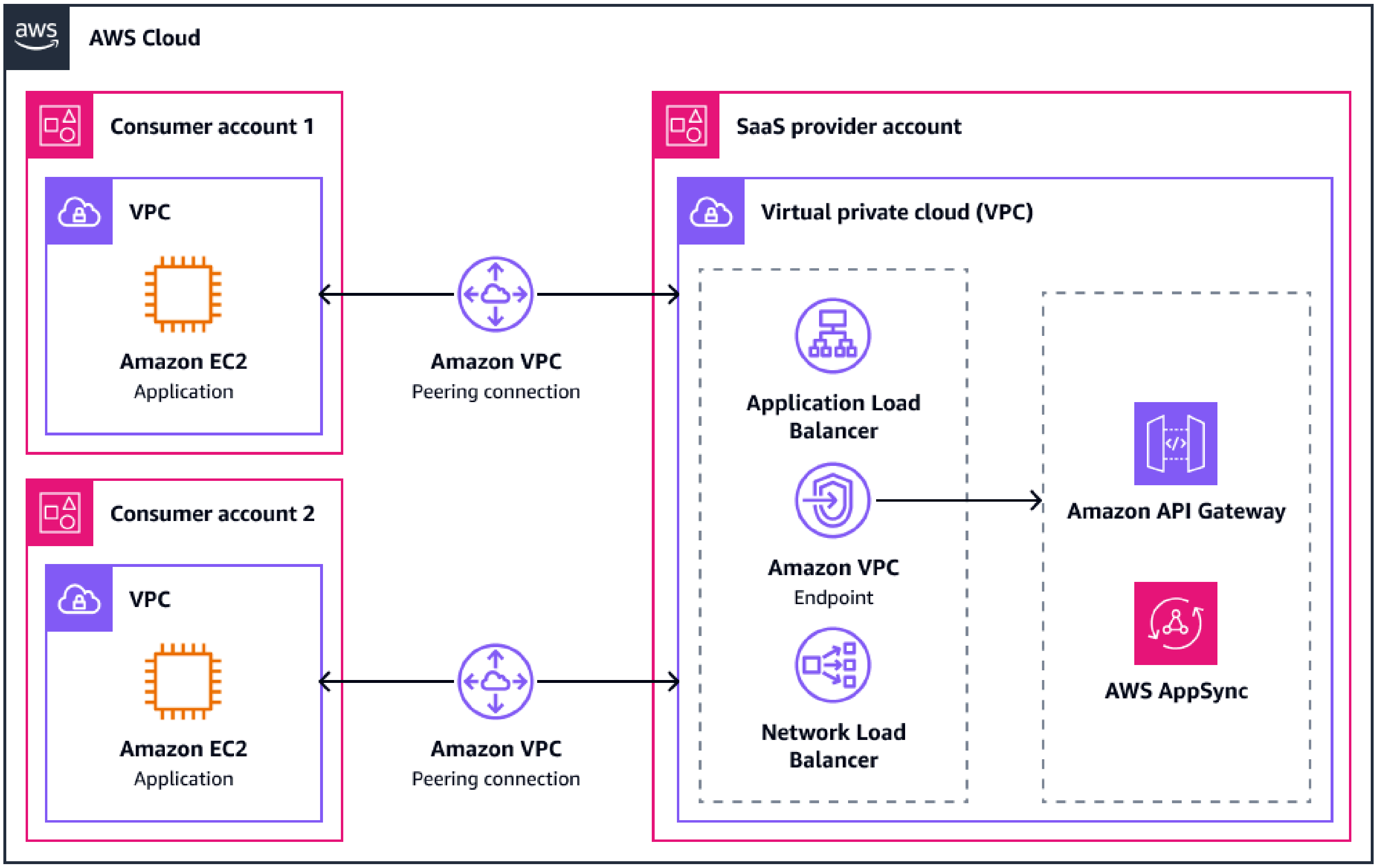
Creating VPC peering connections

When you use VPC peering to connect the SaaS provider's VPC with the consumer's VPC, both parties are able to initiate connections. This requires proper configuration of security groups, firewalls, and network-access control lists (NACLs) in both accounts. Otherwise, unwanted traffic might enter the network through the peering connection. You can use security groups to reference security groups from peered VPCs. This can help you control access to your application because allow-listing security groups provides more explicit and granular access control compared to allow-listing IP addresses.

With VPC peering, the SaaS offering can be reached through a service or resource deployed in the VPC. Most SaaS applications sit behind an Application Load Balancer or Network Load Balancer. AWS AppSync private APIs or Amazon API Gateway private APIs are other common entry points to SaaS applications because they can be a target over a peering connection through interface VPC endpoints.

After you establish a peering connection, you must update the route tables for the VPCs in both accounts to define the peering connection as the next hop for the respective CIDR range. This solution is recommended only for SaaS providers who have a few consumers because managing multiple peering connections quickly becomes too complex.

The following diagram shows a basic configuration with some possible integrations. VPCs in two consumer accounts have a peering connection with a VPC in the SaaS provider account.



The following are the benefits of this approach:

- Time to repair: No single point of failure for communication
- Scalability: No bandwidth limitations over VPC peering
- TCO: No cost for peering connection or traffic over the peering connection within the same Availability Zone
- TCO: No infrastructure to manage
- Adaptability: Support for IPv6
- Adaptability: Inter-Region peering supported

The following are the drawbacks of this approach:

- Adaptability: No support for transitive routing
- Adaptability: No support for overlapping CIDR ranges
- Scalability: Limited scalability (maximum 125 peering connections per VPC)
- TCO: Complexity grows exponentially with each additional peering connection
- TCO: Overhead from managing route tables, peering connections themselves, security group rules, and traffic inspection
- Network isolation: Tight security controls required because entire VPCs of both parties are exposed

Connecting VPCs with AWS Transit Gateway

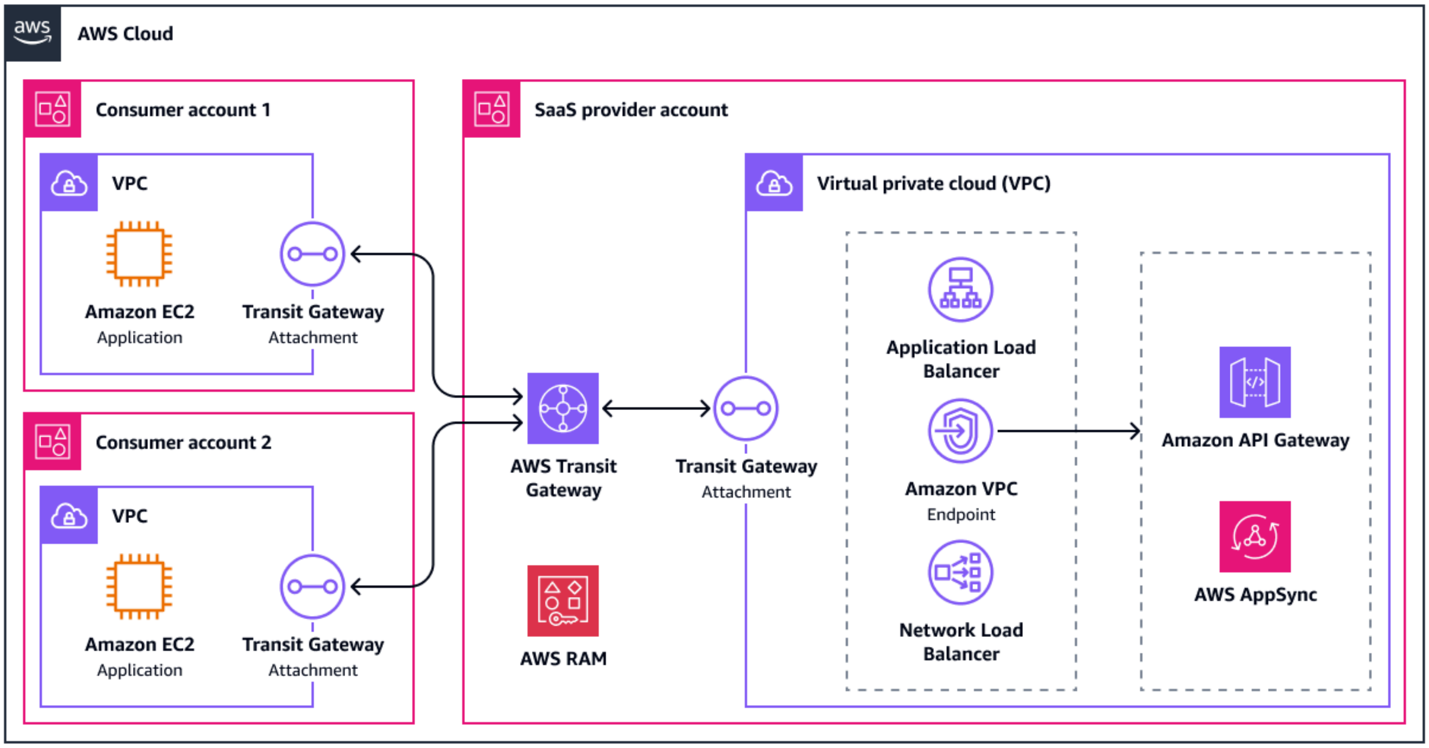
When you connect VPCs through AWS Transit Gateway, it creates VPC attachments and deploys network interfaces in the subnets of each Availability Zone that should route traffic to and from the VPC. It is recommended to have a dedicated /28 subnet in every Availability Zone for the VPC attachment. For more information, see Amazon VPC Transit Gateways design best practices. The VPCs need an updated route table to send traffic through the deployed network interface, and the Transit Gateway route tables need to be updated accordingly. In a multi-tenant configuration, you want the SaaS provider's VPC to have a route to all of the consumers' VPCs. The consumer's VPCs should have a route only to the SaaS provider's VPC.

Transit Gateway is highly available by design. It supports monitoring with VPC Flow Logs, and the maximum bandwidth for a Transit Gateway attachment is 100 Gbps per Availability Zone. Like VPC peering, this approach enables cross-VPC security group referencing, which simplifies access control between the environments.

There are two main options to connect consumers to your SaaS offering with Transit Gateway.

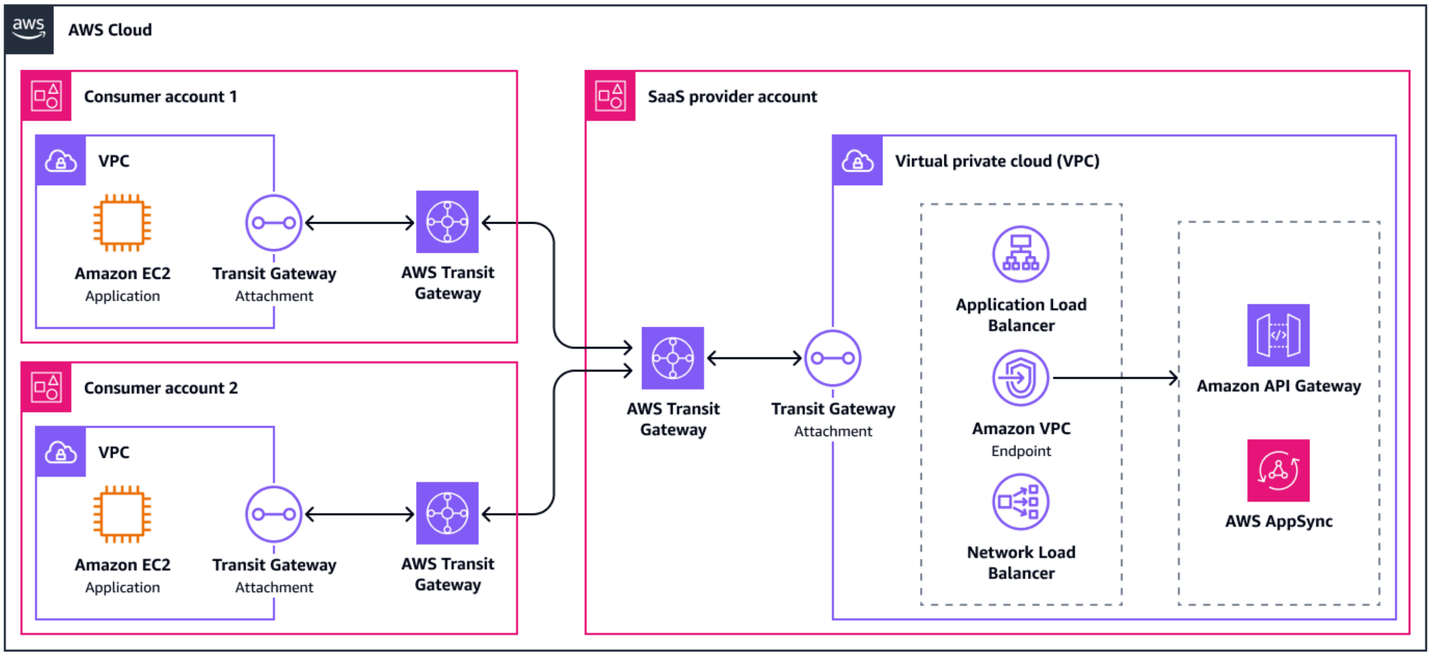
Option 1: Using RAM

In the first option, the service provider shares the Transit Gateway with the consumers by using AWS Resource Access Manager (AWS RAM). This allows the consumers to deploy the VPC attachments in their own accounts. The following diagram shows this option at a high level.



Option 2: Peered transit gateways

The second option is to peer your transit gateway with a transit gateway in the consumers' accounts. This provides consumers with more flexibility because they can now fully control the route tables within their transit gateway. For example, they could set up centralized inspection between the service and their workloads. A drawback of this option is only static routing between transit gateways is supported. The following diagram shows this option at a high level.



The following are the benefits of this approach:

- Scalability: Support for up to 5,000 attachments
- Scalability: One place to manage and monitor all connected VPCs
- Adaptability: Transit Gateway can also attach to VPNs, AWS Direct Connect gateways, and third-party SD-WAN appliances
- Adaptability: Flexible architecture, such as adding an inspection VPC
- Adaptability: Support for transitive routing
- Adaptability: Can peer intra-Region and inter-Region transit gateways
- Adaptability: Support for IPv6
- TCO: AWS Transit Gateway is a fully managed service, so it requires less operational effort
- TCO: TCO grows linearly with each additional transit gateway attachment

The following are the drawbacks of this approach:

- Ease of integration: Routing configuration requires advanced networking knowledge
- Adaptability: No support for overlapping CIDR ranges
- TCO: Overhead from managing route tables entries, security group rules, and traffic inspection
- Security: Tight security controls required because entire VPCs of both parties are exposed

Service consumers operating on premises

This section discusses connectivity options between SaaS workloads in the AWS Cloud and the on-premises data centers. Many consumers with on-premises requirements, especially at the enterprise level, see the cloud as an extension of their physical network, and they want to reflect that in their architecture. That means private connectivity to the SaaS offering in the cloud, either through logical tunnels or even through a private physical connection. Other consumers will accept connectivity through the public internet, which is also discussed in this section.

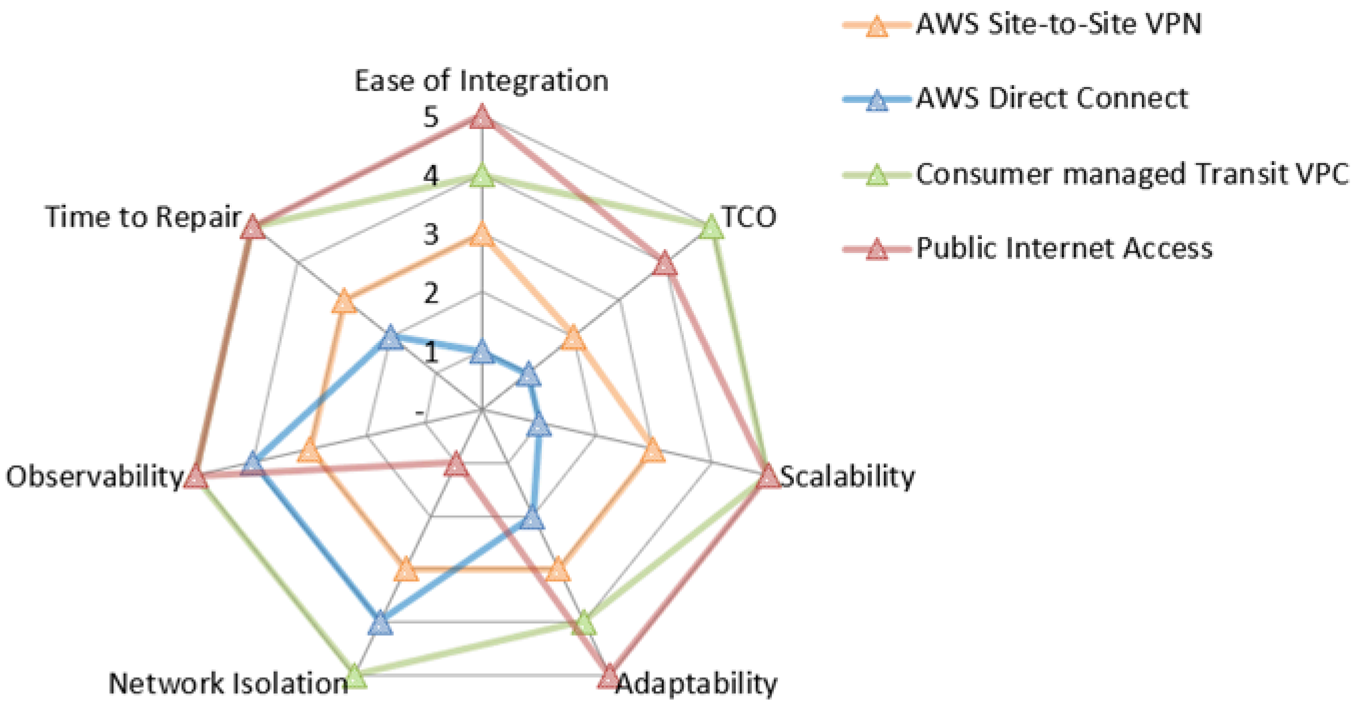
This section discusses the following network access approaches:

- Connecting with AWS Site-to-Site VPN
- Connecting with AWS Direct Connect
- Connecting with a transit VPC architecture
- Connecting through the public internet

The following networking value map summarizes how each of these options scores for each evaluation metric. For more information about the evaluation metrics, see Evaluation metrics in this guide. In the map, a five represents the best score, such as the lowest TCO, best network isolation, or lowest time to repair. For more information about how to read this radar chart, see Networking value map in this guide.

Note

The provider-managed transit VPC option is excluded because the scores heavily depend on which services are being operated.



The radar chart shows the following values.

Evaluation metric	AWS Site-to-Site VPN	AWS Direct Connect	Consumer-managed transit VPC	Public internet access
Ease of integration	3	1	4	5
TCO	2	1	5	4
Scalability	3	1	5	5
Adaptability	3	2	4	5
Network isolation	3	4	5	1
Observability	3	4	5	5
Time to repair	3	2	5	5

Connecting with AWS Site-to-Site VPN

AWS Site-to-Site VPN connections can terminate on either a virtual private gateway or a transit gateway. A *virtual private gateway* is the VPN endpoint on the AWS side of your Site-to-Site VPN connection that can be attached to a single VPC. A *transit gateway* is a transit hub that can be used to interconnect multiple VPCs and on-premises networks. It can also be used as a VPN endpoint for the AWS side of the Site-to-Site VPN connection. This section discusses both options.

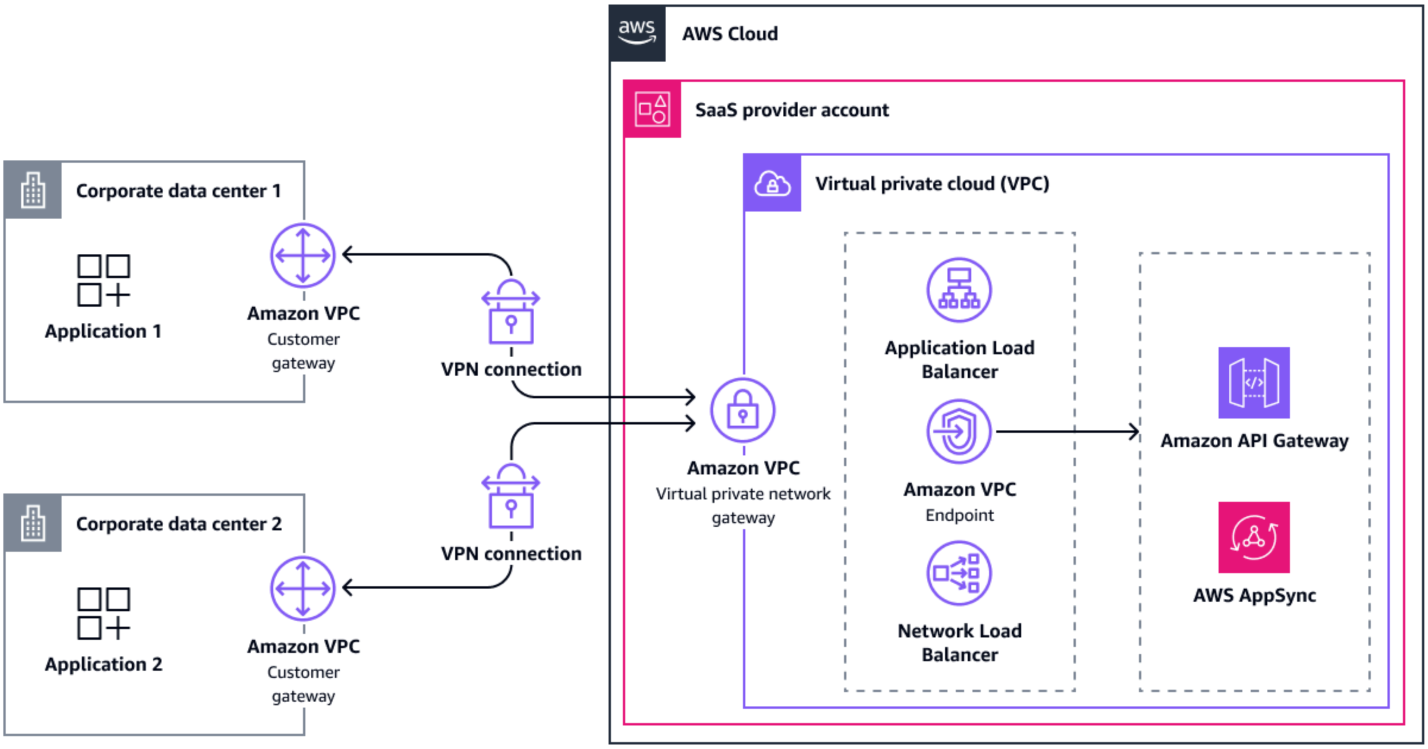
Connection through a virtual private gateway

After you create a virtual private gateway, you attach it to the VPC that contains your SaaS offering. Then, you enable route propagation to propagate the VPN routes to the VPC route table. Those routes can be either static or BGP-advertised dynamic routes.

For high availability, an Site-to-Site VPN connection has two VPN tunnels that terminate in two Availability Zones on the AWS side. If one becomes unavailable, the second tunnel can take over. A single tunnel allows a maximum bandwidth of 1.25 Gbps. Because virtual private gateways do not support equal-cost multi-path routing (ECMP), you can use only one tunnel at a time.

To increase fault tolerance, you can set up a second VPN connection to a second physical customer gateway. After the connection is established, the consumer can reach resources in the SaaS provider's VPC.

The following diagram shows this architecture.



The following are the benefits of this approach:

- Time to repair: Managed failover to secondary VPN tunnel
- Observability: Integration for managed active monitoring by using Network Synthetic Monitor
- Ease of integration: Dynamic routing support through BGP
- Adaptability: Compatibility with most on-premises networking equipment
- Adaptability: IPv6 support
- TCO: AWS Site-to-Site VPN is a fully managed service, so it requires less operational effort
- TCO: No cost for virtual gateways, although there are charges for the two public IPv4 addresses on each
- Network isolation: Enables secure private communication through the internet

The following are the drawbacks of this approach:

- Ease of integration: The consumer must configure their customer gateway
- Scalability: Lack of ECMP support limits bandwidth to 1.25 Gbps per virtual gateway
- Scalability: Limited scaling due to increased network complexity and operational overhead
- Adaptability: IPv6 support only for the inside IP addresses of the VPN tunnels

- Adaptability: No transitive routing
- TCO: Operational overhead to maintain, manage, and configure numerous VPN connections for the SaaS provider

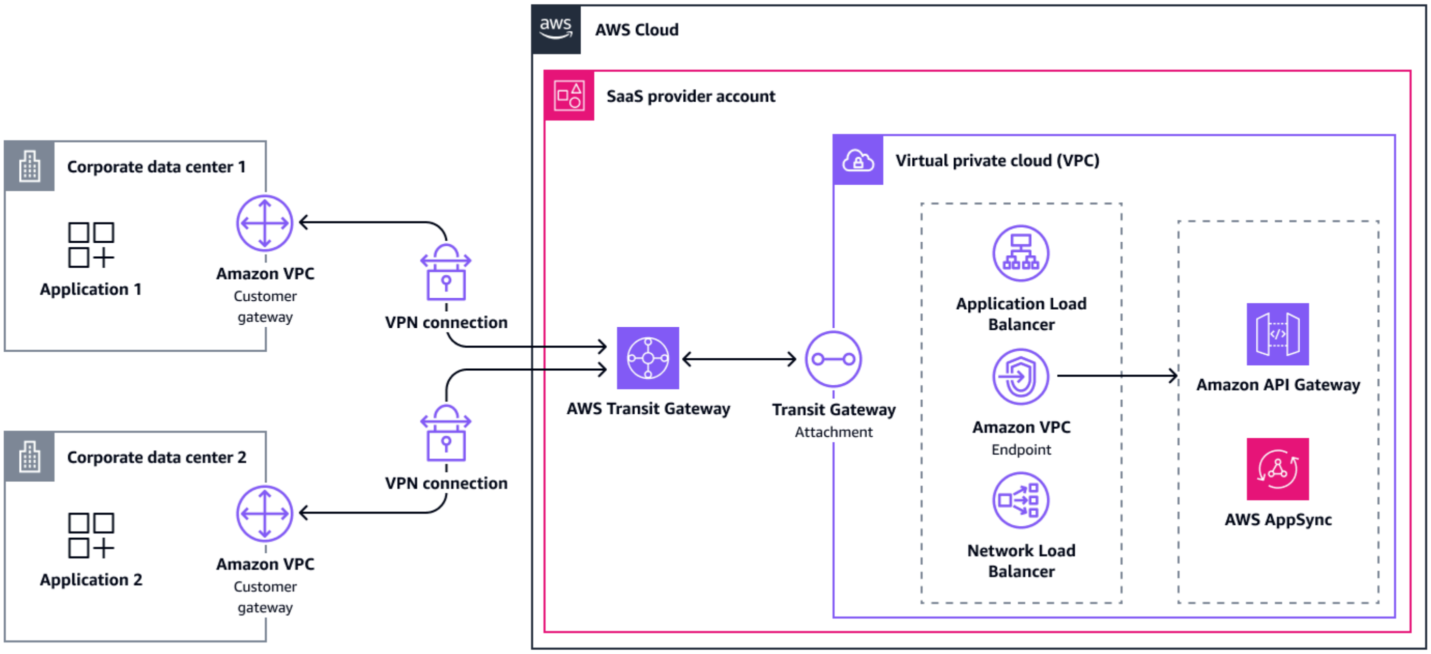
Connection through a transit gateway

Connections through transit gateways are similar to virtual gateways. However, there are a few differences to keep in mind.

First, routes for the VPN attachment can be automatically propagated within the transit gateway route table, but you must manually add the routes to the attached VPCs.

Compared to a virtual gateway, Transit Gateway supports ECMP. If the customer gateway supports ECMP, it can use both tunnels to achieve a total maximum throughput of 2.5 Gbps. You can establish multiple connections between the same on-premises network to the transit gateway. Using this approach, you can increase the maximum bandwidth by up to 2.5 Gbps per connection.

The following diagram shows this architecture.



The following are the benefits of this approach:

- Time to repair: Managed failover to secondary VPN tunnel
- Observability: Integration for managed active monitoring by using Network Synthetic Monitor
- Ease of integration: Dynamic routing support through BGP

- Scalability: ECMP support allows scaling VPN throughput to satisfy large bandwidth requirements
- Scalability: Large number of VPN connections supported by a single transit gateway (up to almost 5,000)
- Scalability: One place to manage and monitor all the VPN connections
- Adaptability: Compatibility with most on-premises networking equipment
- Adaptability: IPv6 support
- Adaptability: Inherit flexibility of AWS Transit Gateway
- TCO: AWS Transit Gateway is a fully managed service, so it requires less operational effort
- TCO: No cost for virtual gateways, although there are charges for the two public IPv4 addresses on each
- Network isolation: Enables secure private communication through the internet

The following are the drawbacks of this approach:

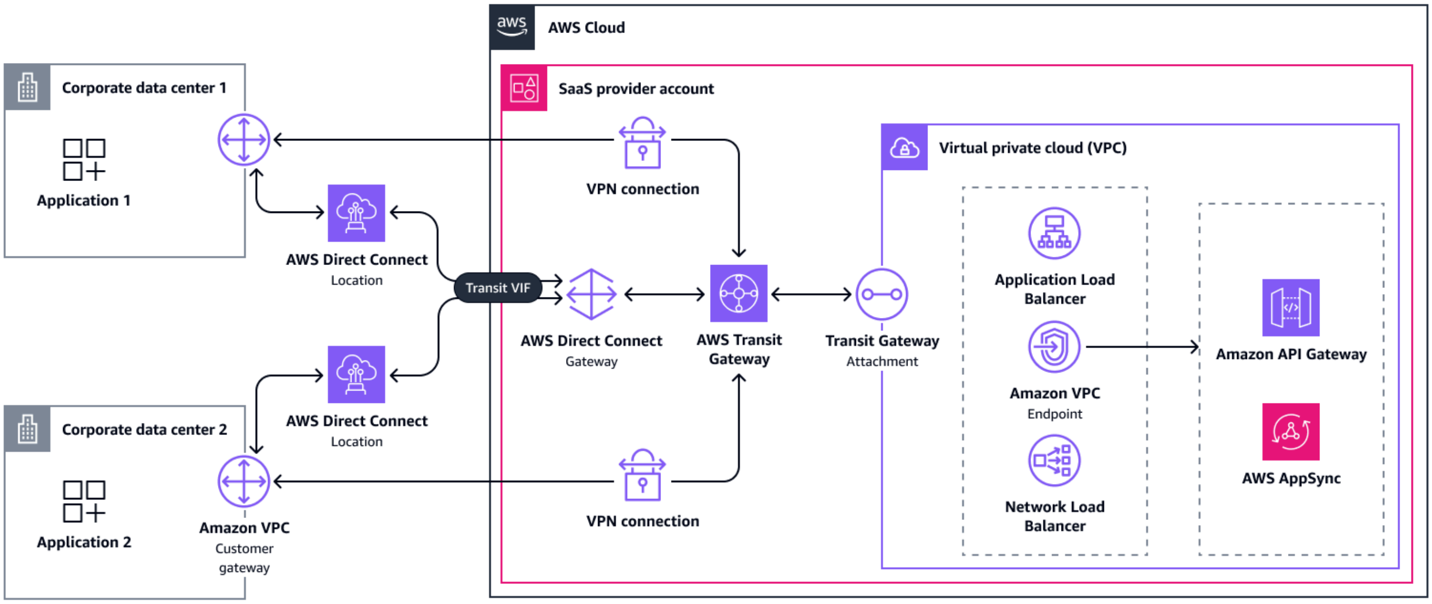
- Ease of integration: The consumer must configure their customer gateway
- Scalability: Limited scaling due to increased network complexity and operational overhead
- Adaptability: IPv6 support only for the inside IP addresses of the VPN tunnels
- TCO: Operational overhead to maintain, manage, and configure numerous VPN connections for the SaaS provider
- TCO: Extra charges for use of AWS Transit Gateway
- TCO: Additional complexity managing the transit gateway route tables

Connecting with AWS Direct Connect

AWS Direct Connect links your internal network to a AWS Direct Connect location over a standard Ethernet fiber-optic cable. Unlike the other architecture options, a dedicated connection cannot be established in a few minutes. Instead, this process can take up to several days if all requirements are met. If not, it might take longer. Therefore, we suggest that you reach out to your AWS account team or AWS Support for help with this approach. Optionally, you can choose a hosted connection that is provided by an AWS Partner and shared with other customers. The architecture is the same regardless. You might choose AWS Direct Connect because it reduces latency, improves bandwidth, or complies with regulatory requirements.

To use the AWS Direct Connect connection, consumers must create either a public, private, or transit virtual interface. There are different architecture options available. The most flexible one to connect multiple on-premises locations to the AWS Cloud is a transit virtual interface connected to an AWS Direct Connect gateway. An AWS Direct Connect gateway is a global, logical component that allows the service provider to connect up to six transit gateways to it. Furthermore, you can connect up to 30 virtual interfaces to the gateway. For scale, you can create additional AWS Direct Connect gateways. In the SaaS provider account, the transit gateways then attach to the VPCs, as described previously.

Consumers can connect using one to four AWS Direct Connect connections from a total of one or two AWS Direct Connect locations, depending on the desired level of resiliency. For more information, see Configure AWS Direct Connect for maximum resiliency. An AWS Site-to-Site VPN connection over the internet might also serve as a lower-cost backup path for an AWS Direct Connect connection. Supported AWS Direct Connect dedicated connections can use MACsec to encrypt the link on Layer 2 between the AWS Direct Connect location and the data center. It is common to have a Site-to-Site VPN connection for additional confidentiality of the data. The Site-to-Site VPN connection can terminate on the transit gateway by using a normal VPN attachment. The following diagram shows this architecture.



The following are the benefits of this approach:

- Observability: Integration for managed active monitoring by using Network Synthetic Monitor
- Scalability: Support for increased bandwidth throughput
- Adaptability: IPv6 support

- TCO: Potential to reduce data transfer
- TCO: Consistent network experience
- Network isolation: Private connectivity that can fulfill regulatory requirements

The following are the drawbacks of this approach:

- Ease of integration: Time and manual effort to set up
- Scalability: Limited scalability beyond tens of AWS Direct Connect connections because there are multiple quotas to track
- Adaptability: Configuration options depend on the available AWS Direct Connect locations
- TCO: Scheduled AWS Direct Connect maintenance can cause downtime that requires action

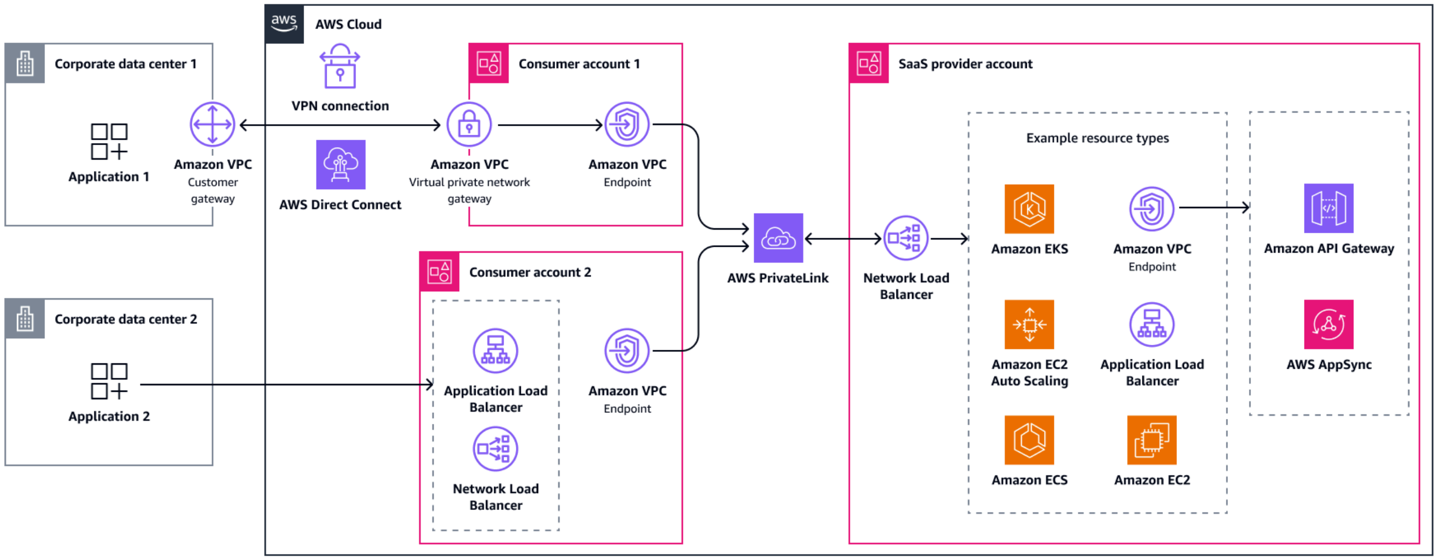
Connecting with a transit VPC architecture

Transit VPC is an architecture option that gives flexibility to the consumers for how to connect to AWS, and it allows SaaS providers to benefit from having unified access to their service through AWS PrivateLink. The consumer connects from on premises to a transit VPC that contains only an entry point (such as a virtual private gateway) and an interface VPC endpoint, which is an AWS PrivateLink resource. The transit VPCs should either be owned by the SaaS provider or by the consumers. This section discusses both options.

You can create the transit VPC and subnets with CIDR ranges that are compatible with the on-premises data center. If they require private connectivity, consumers can connect to that VPC through AWS Direct Connect or AWS Site-to-Site VPN. You can also configure access to the transit account from the public internet by using an Application Load Balancer or Network Load Balancer that points to the VPC endpoint.

Consumer-managed transit VPC

In this approach, the SaaS provider leaves management of the transit VPCs up to the consumers. From a technical point of view, the SaaS provider's architecture is the same as when connecting to consumers in the AWS Cloud through AWS PrivateLink. From sales and product perspective, it is additional effort because some consumers don't have AWS accounts yet. They might be hesitant to open and operate an account. The SaaS provider should give guidance to their consumers about how to create AWS accounts and connect their on-premises data center. The following diagram shows a mix of public and private access, where the consumers own the transit VPCs.



The following are the benefits of this approach:

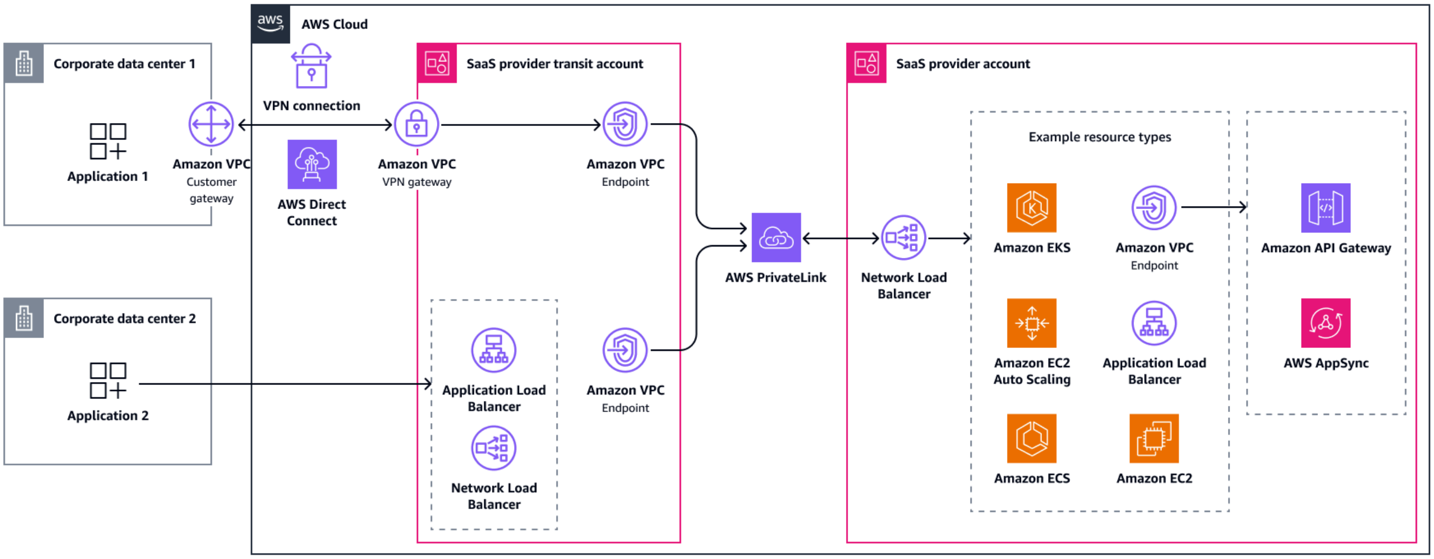
- Time to repair: Operational overhead is largely offloaded to SaaS consumers
- Adaptability: SaaS consumers can choose from different access options
- Adaptability: No CIDR range conflicts, even when using Site-to-Site VPN or AWS Direct Connect
- All metrics: Service provider inherits AWS PrivateLink benefits

The following are the drawbacks of this approach:

- Ease of integration: SaaS consumers require at least one AWS account
- TCO: A transit VPC is an architecture, not a fully managed service, so it requires more operational effort

Provider-managed transit VPC

This approach uses the same technologies, but the account boundaries and responsibilities change. Here, the SaaS provider owns the transit VPCs, preferably in a separate account from the SaaS offering. This decoupling reduces costs, reduces risks, and allows the transit account to scale independently. For environments that require a high degree of isolation, you can create additional separation between tenants by using a subnet or by creating a separate transit VPC for each consumer. The consumers can then choose how to connect to the transit VPC. This approach provides more options to expand the total addressable market, but it has a higher TCO for the SaaS provider due to the need to operate and monitor additional architectural components.



The following are the benefits of this approach:

- Adaptability: SaaS consumers can choose from different access options
- Adaptability: SaaS consumers don't need to have an AWS account
- Adaptability: No CIDR range conflicts, even when using Site-to-Site VPN or AWS Direct Connect

The following are the drawbacks of this approach:

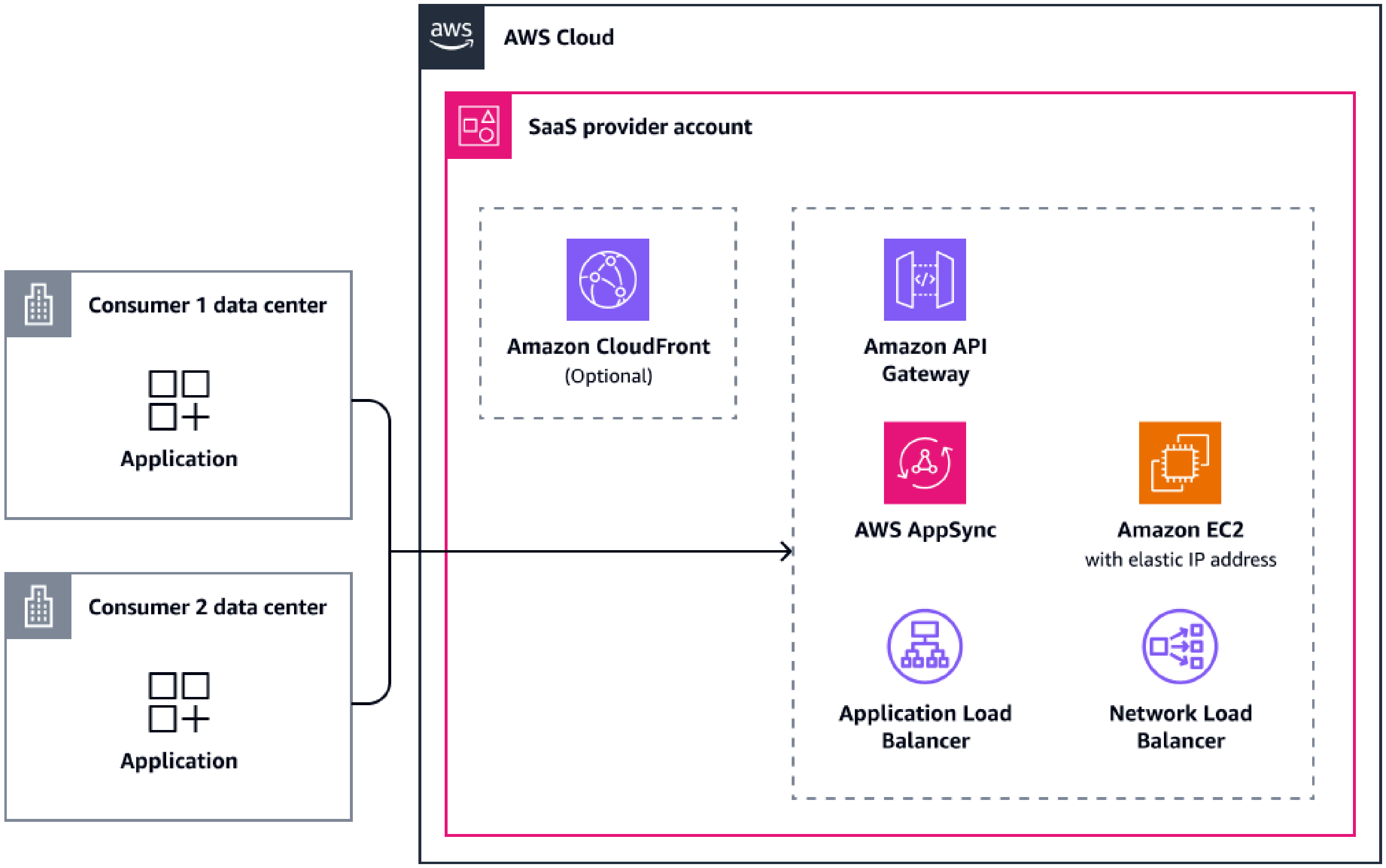
- TCO: A transit VPC is an architecture, not a fully managed service, so it requires more operational effort
- TCO: SaaS provider needs to operate and monitor additional architectural components

Connecting through the public internet

Public internet access is also a valid option for providing access to a SaaS offering, although it does not offer private connectivity in the traditional sense. Some consumers might still prefer a public access approach because it requires no additional networking infrastructure between them and the SaaS provider. It reduces complexity, cost, and integration time in exchange for an increased attack surface. Strong authentication and authorization mechanisms can help mitigate the increased threat level, and you should always encrypt traffic. It is still recommended that you have an additional layer of security in this scenario, such as by using AWS WAF.

The architecture in this scenario is straightforward. The consumer connects to a public host (the SaaS provider) through the internet. The application can be hosted directly on a public Amazon

Elastic Compute Cloud (Amazon EC2) instance with an Elastic IP address. The preferred option is to host it behind an Application Load Balancer or similar service. For better performance and caching static assets, you can use a content delivery network, such as Amazon CloudFront. To serve an application with minimum latency over two global static Anycast IP addresses, you can place AWS Global Accelerator in front of an Amazon EC2 instance, Network Load Balancer, or Application Load Balancer. In addition, CloudFront, Application Load Balancers, AWS AppSync, and Amazon API Gateway all integrate with AWS WAF. The following diagram provides an overview of the public internet access connectivity options.



The following table describes supported protocols and integrations for this scenario.

Service or resource	IPv6	AWS WAF integration	Can be a Global Accelerator endpoint
Amazon CloudFront	Supported	Supported	Not supported
Amazon API Gateway	Supported	Supported	Not supported

AWS AppSync	Partially supported	Supported	Not supported
Amazon EC2 with an Elastic IP address	Supported	Not supported	Supported
Application Load Balancer	Supported	Supported	Supported
Network Load Balancer	Supported	Not supported	Supported

The following are the benefits of this approach:

- Ease of integration: Simplicity and accessibility
- Scalability: Unlimited scale
- Adaptability: No CIDR range conflicts possible
- Adaptability: CloudFront support

The following are the drawbacks of this approach:

- Network isolation: No private connectivity
- Network isolation: Strong security measures required

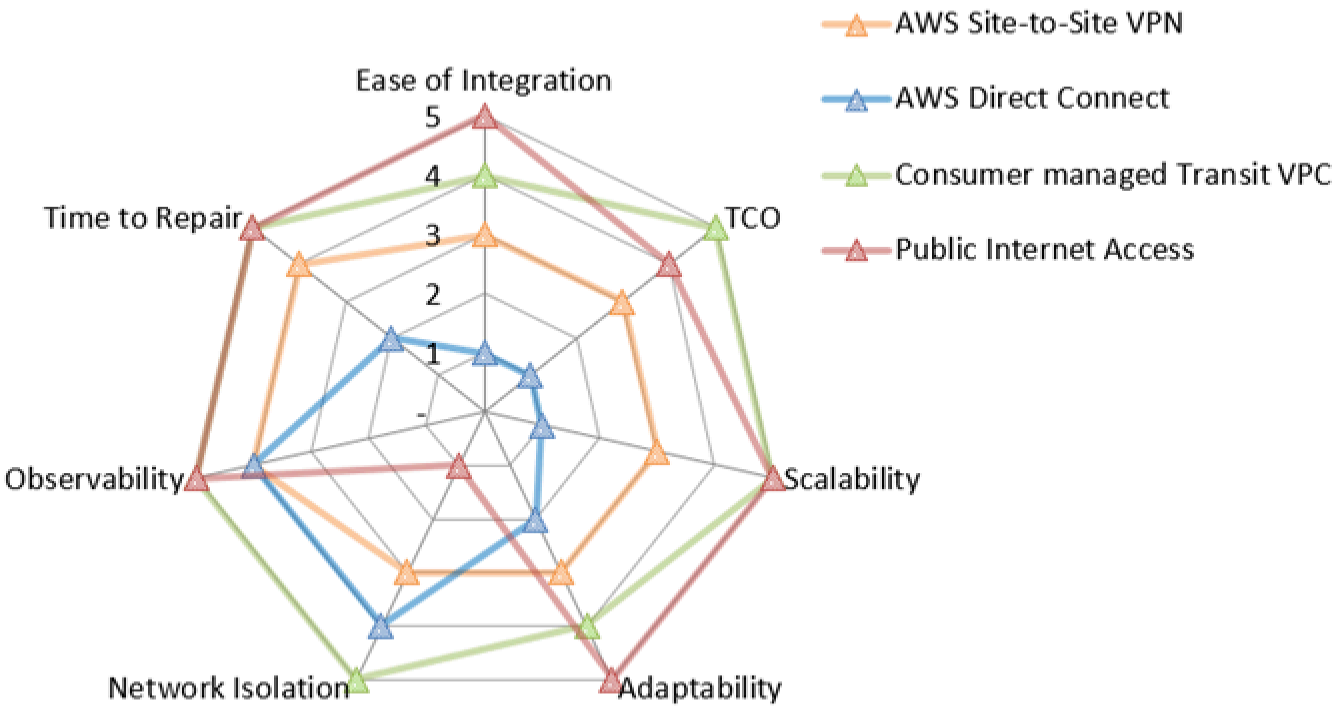
Other benefits and drawbacks apply, depending on the services that you choose.

SaaS consumers operating on other cloud service providers

This scenario describes solutions for consumers on other cloud service providers (CSPs). This scenario shares some commonalities with connections to on-premises data centers. In fact, all connectivity options for on-premises environments are equally valid for consumers on other CSPs, even a private connection with AWS Direct Connect is possible with some CSPs. Most CSPs offer documentation and support about how to connect to the AWS Cloud through AWS Site-to-Site VPN or AWS Direct Connect.

When choosing Site-to-Site VPN, consumers can benefit from managed gateways or similar resources from their respective CSP. Consumers don't necessarily have to set them up themselves, as in the on-premises scenario. This influences some of the metrics for Site-to-Site VPN, such as improvements to time to repair and observability. This is because both ends of the connection are now managed.

The following networking value map summarizes how each of these options scores for each evaluation metric. It is very similar to the networking value map for on-premises connections, although the values for Site-to-Site VPN are different. For more information about the evaluation metrics, see Evaluation metrics in this guide. In the map, a five represents the best score, such as the lowest TCO, best network isolation, or lowest time to repair. For more information about how to read this radar chart, see Networking value map in this guide.



The radar chart shows the following values.

Evaluation metric	AWS Site-to-Site VPN	AWS Direct Connect	Consumer-managed transit VPC	Public internet access
Ease of integration	3	1	4	5
TCO	3	1	5	4

Scalability	3	1	5	5
Adaptability	3	2	4	5
Network isolation	3	4	5	1
Observability	4	4	5	5
Time to repair	4	2	5	5

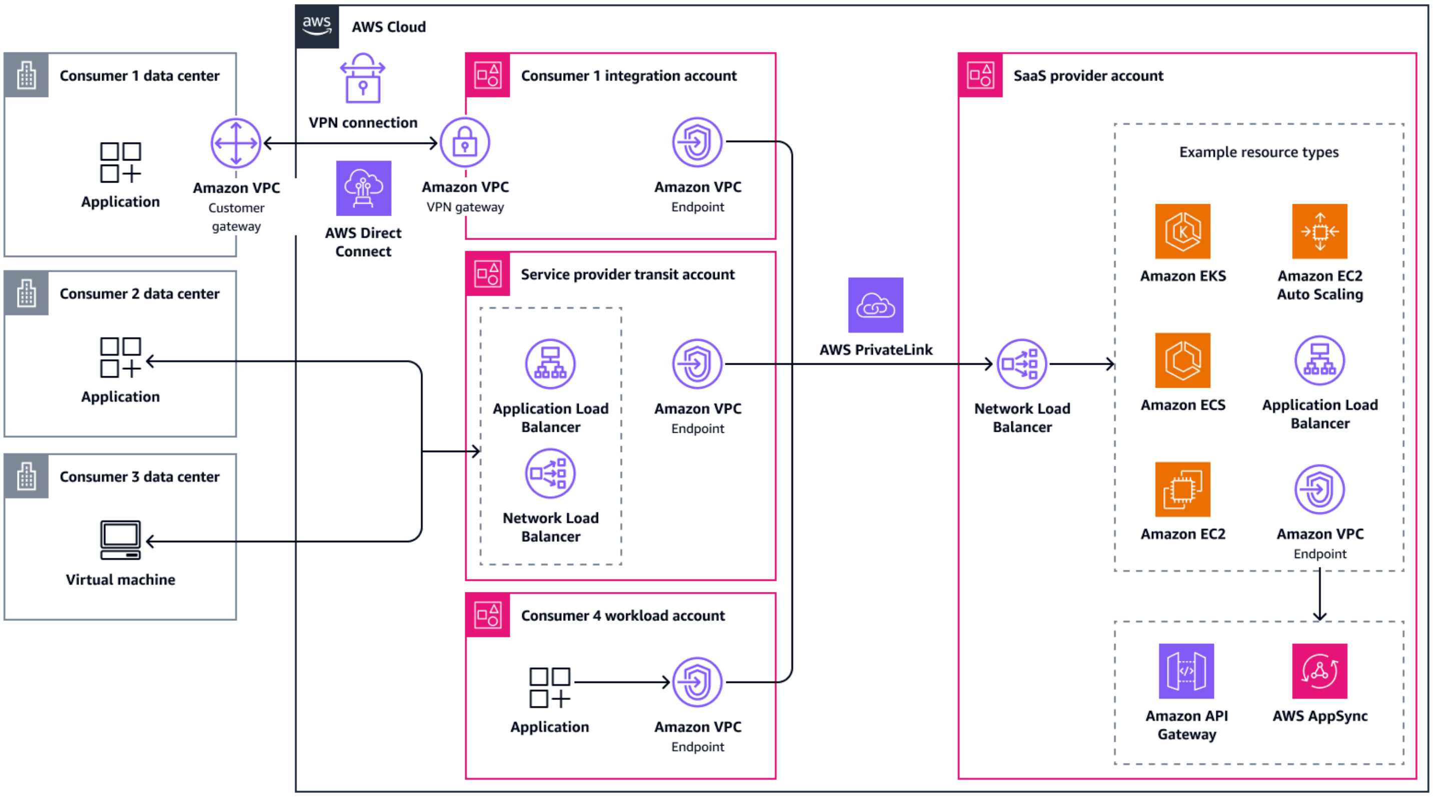
Supporting hybrid environments

It's common for consumers to come from different environments, each with its own technical and security constraints. Some customers may operate entirely from on-premises data centers that require secure connectivity over the Internet or through dedicated network links. Others might already be running workloads within AWS and expect low-latency, private network paths. A third group might rely on other CSPs, where connectivity must bridge different cloud networks.

Regardless, you should aim for standardized network access to your SaaS application to simplify your architecture and reduce operational complexity. Two of the previously presented approaches —public internet access and transit VPCs—work well across these scenarios. Public internet access offers the fastest onboarding path with minimal setup for your customers. Transit VPCs offer more controlled and private access, often using AWS PrivateLink.

When designing your SaaS offering, you can adopt a single network access model or combine multiple approaches into a tiered offering. For example, you might offer a public access deployment tier for customers who prioritize ease of connection and rapid onboarding, and you might offer a private access deployment tier for customers who have strict compliance or security control requirements. These tiers come with different cost, performance and risk profiles. It is also possible to combine both approaches into a single architecture. In that case, make sure that you have strong security measures so that public and private paths remain isolated.

The following diagram shows a hybrid access approach, where consumers have the option to connect privately from their data center or CSP, publicly, or directly through AWS PrivateLink (if they have workloads in the AWS Cloud).


AWS Cloud
Consumer 1 data center
Application
Amazon VPC
Customer gateway
VPN connection
AWS Direct Connect
Consumer 1 integration account
Amazon VPC
VPN gateway
Amazon VPC
Endpoint
Consumer 2 data center
Application
Consumer 3 data center
Virtual machine
Service provider transit account
Application Load Balancer
Network Load Balancer
Amazon VPC
Endpoint
Consumer 4 workload account
Application
Amazon VPC
Endpoint
AWS PrivateLink
SaaS provider account
Network Load Balancer
Example resource types
Amazon EKS
Amazon EC2 Auto Scaling
Amazon ECS
Application Load Balancer
Amazon EC2
Amazon VPC
Endpoint
Amazon API Gateway
AWS AppSync

Advanced networking access scenarios for SaaS offerings in the AWS Cloud

The architectures discussed in the Networking access scenarios for SaaS offerings in the AWS Cloud section should help you find a solution for the majority of use cases. However, there are some scenarios that have specific technical requirements. Many are beyond the scope of this guide.

This section discusses the following advanced technical requirements and considerations:

- Bidirectional communication
- TCP, UDP, and proprietary protocols

Bidirectional communication

In some cases, applications require bidirectional traffic in order operate as expected. Common use cases are webhooks or notification services. Generally, you can achieve this by having a WebSocket connection between the server and the client. This connection keeps the TCP session open and allows both participants to send traffic over the connection. Most of the services discussed in this guide natively support WebSocket, including Network Load Balancers, Application Load Balancers, Amazon API Gateway, AWS PrivateLink, and AWS AppSync (through private real-time endpoints).

In other cases, an application on the SaaS provider side might need access to resources on the consumer side, such as a database. When you connect through bidirectional channels, such as an AWS Site-to-Site VPN connection, that is not an issue.

On the other hand, AWS PrivateLink and Elastic Load Balancing support only unidirectional traffic. If you use these services, you must set up another network path for the traffic that initiates from your SaaS offering. For example, this might be an additional AWS PrivateLink connection that goes in the reverse direction.

TCP, UDP, and proprietary protocols

Many applications are served through HTTP or HTTPS, but not all. Some may use other Layer 7 protocols on top of TCP, such as Message Queuing Telemetry Support (MQTT). Others might even use UDP to serve consumers. In rare cases, services use proprietary protocols that must be transmitted inside packets (Layer 3). For these scenarios, it is important to understand which services support your SaaS offering.

For Layer 3 services, you can use AWS PrivateLink and Network Load Balancers, both of which support all TCP and UDP traffic.

For Layer 7 services, Application Load Balancers and Amazon CloudFront support HTTP, HTTPS, WebSocket, and Google Remote Procedure Calls (gRPC). Similarly, Amazon API Gateway and AWS AppSync each support HTTP, HTTPS, and WebSocket. Amazon CloudFront is the only service that currently supports HTTP/3.

You can use Amazon VPC Lattice to connect Layer 7 applications and Layer 3 resources. It supports HTTP, HTTPS, gRPC, TCP, and TLS passthrough.

If the application can serve traffic only over Layer 3, it is crucial that you use core AWS networking services, such as AWS Transit Gateway, AWS Direct Connect, AWS Site-to-Site VPN, and VPC peering. The traffic should then be routed directly from the SaaS consumer to the compute layer of the SaaS offering.

Anti-patterns for network access in the AWS Cloud

An *anti-pattern* is a frequently used solution for a recurring issue where the solution is counter-productive, ineffective, or less effective than an alternative. The design options mentioned in this section usually work, but they come with significant disadvantages. If possible, they should be avoided because better alternatives are available.

This section discusses the following anti-patterns and challenges:

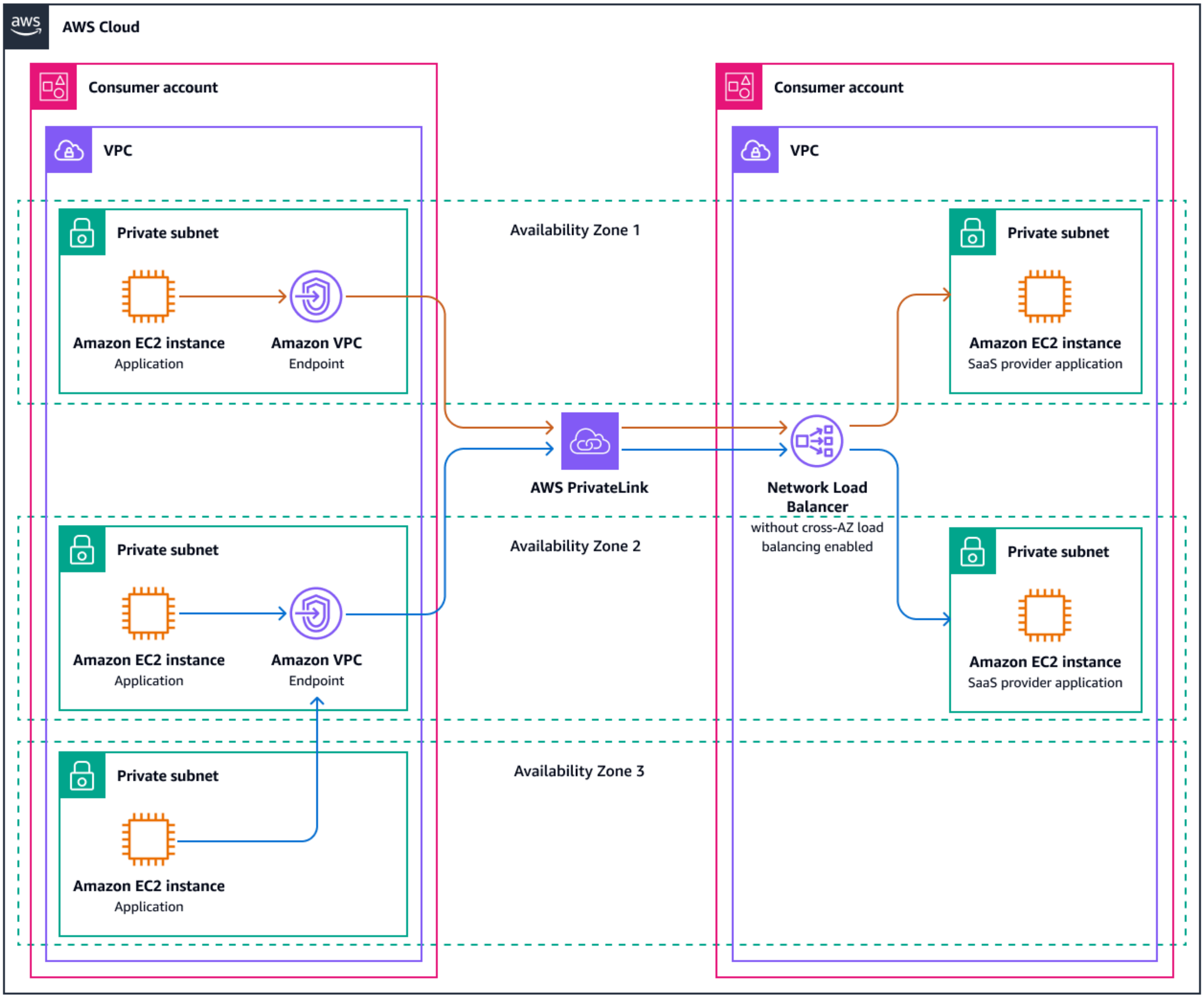
- Availability Zone mismatch with AWS PrivateLink
- AWS Site-to-Site VPN connections between AWS accounts

Availability Zone mismatch with AWS PrivateLink

When providing access to an application through AWS PrivateLink, SaaS consumers can create interface VPC endpoints only in the Availability Zones where the application is deployed. For example, if the application is deployed in `use1-az1` and `use1-az2`, the consumer cannot deploy a VPC endpoint in `use1-az3`. We recommend that you deploy the SaaS offering in every Availability Zone. The majority of AWS Regions have three Availability Zones, although some have more. For a comprehensive list, see Regions and Availability Zones. Consider the number of Availability Zones when choosing an AWS Region.

> **Note**
>
> Availability zone names are different from Availability Zone IDs. For more information, see Availability Zone IDs for your AWS resources.

If a SaaS provider chooses not to deploy in all Availability Zones, there are some consequences. Assume the SaaS offering is deployed in `use1-az1` and `use1-az2`, but the consumer is using all three Availability Zones, including `use1-az3`. The interface VPC endpoints are deployed on the consumer side in `use1-az1` and `use1-az2`, and now the application in `use1-az3` needs to access one of these endpoints. First of all, traffic must be allowed from the subnets in the unmatched Availability Zones into the respective VPC endpoints. The consumer can decide to use the regional AWS PrivateLink DNS name, which can resolve to either VPC endpoint and which evenly distributes the traffic between the two. Or the consumer might choose to send traffic directly to an endpoint, such as `use1-az2`. This results in 67% of the traffic arriving on the provider side in `use1-az2` and 33% in `use1-az1`. The following figure depicts this scenario.



With a significant number of consumers and an uneven distribution of traffic, a workload might run into capacity issues in one Availability Zone and be under capacity in another. To tackle that issue, the SaaS provider can decide to evenly load balance the traffic on their side by enabling cross-zone load balancing on the Network Load Balancer. This incurs additional charges.

If only one Availability Zone is matched by the service provider, then all traffic will enter over a single endpoint. This creates an even larger imbalance. As a result, the SaaS offering is no longer highly available for the consumer. It doesn't matter to the consumer if the application is served over additional Availability Zones that they are not using themselves. In the worst case, a SaaS provider might not be able to serve a consumer who doesn't use any of the same Availability Zones.

In the rare case that there is no feasible option for the SaaS provider to provision their application over all Availability Zones, it is also possible to create a subnet only in the missing Availability Zones and then extend the service to those empty Availability Zones. Cross-zone load balancing can then distribute the incoming traffic over the actual application endpoints in the other Availability Zones.

AWS Site-to-Site VPN connections between AWS accounts

Companies migrating from on-premises environments into the cloud sometimes try to lift and shift the entire network. This can cause problems because there are significant differences between on-premises and cloud networking practices. If this mindset shift does not happen, things like AWS Site-to-Site VPN connections from one VPC to another VPC can happen. This approach fails to take advantage of purpose-built networking services in the AWS Cloud, which simplify management and improve performance. Adapting to cloud-native designs helps reduce operational overhead and results in more reliable, scalable connectivity between VPCs.

If you're thinking of providing this connectivity option as a SaaS provider, ask yourself or the consumer why AWS Site-to-Site VPN should be used. Then, work backwards from those requirements to find a better connectivity option. The Comparing service capabilities section of this guide contains a matrix that you can use to help identify options. Then, you can work through the relevant sections of this guide to find an architectural approach that addresses your use case.

Next steps

This guide described various network access approaches in different scenarios, and it describes the benefits and drawbacks of each architecture. You should understand why choosing a network access approach should not be a purely technology discussion. Alignment between business and technology is essential. The following next steps and recommendations can help you assess and standardize your network architecture strategy by evaluating current capabilities, analyzing market needs, and implementing governance controls.

This section contains the following topics:

Assessing current architecture and capabilities

Review the current network architecture against relevant data sources, such as the self-assessment framework in this guide, current regulatory requirements, and the current state of the market (both in terms of your customer and a competitive analysis). For example, consider using the AWS Well-Architected Framework, which is based on decades of experience running production systems at scale in the AWS Cloud.

Review any potential exceptions, one-offs, and historical product decisions. Be curious, challenge them, and do not automatically assume their validity. Customer requirements from years ago might no longer be valid. Challenging assumptions creates opportunity to simplify and reduce the complexity of your architecture.

In simple terms, document the observations so that they can be accessed and understood by diverse roles in your organization. Capture where the current state differs from target state, what the target state is, the impact, and when observations were made. Recording this information helps your organizations make decisions based on fresh data.

Market and customer analysis

Gather insights into market trends. What's the currently preferred way of consumers to access SaaS offerings like yours? Are you still meeting your customers where they are? Did the customer cohorts or behavior change? Did your executives steer the ship toward a new market, a geography with specific regulatory requirements, or a new customer tier? Did your business, or operating model change? For example, are you considering white-labeling your services? Does your growth plan include working with partners so that your service is available to customers when they connect with those partners?

Strategic alignment

When you understand your current capabilities, current architecture, market, and customers, call a strategic alignment meeting. With the relevant product, business, and technology stakeholders, challenge which requirements are still valid and which new requirements need to be considered. Find opportunities to reduce complexity by dropping requirements that are no longer needed. This is not a design by committee; the engineering team needs to prepare and own the actual architecture and implementation details. However, this meeting should clarify why this is the set of requirements that maximizes the benefits for your customers and organization.

Standardization

To attract customers, it might be tempting to let each one freely choose how to connect to your service. After all, any solution might technically work, and you might also have the know-how and resources to manage and operate all of them. This can work well up to a certain point, but as your business scales, it becomes difficult to manage. Your observability stack needs to support metrics from multiple solutions, and your site reliability engineers also need to be able to understand them. You need up-to-date documentation for each connectivity approach. Major changes in your application need to be evaluated against each access approach you are offering. You need to write and maintain automations and infrastructure as code (IaC) for each access approach. The additional overhead of not standardizing access to your service must be weighed against the flexibility that you want to offer to your customers.

If you need a north star to guide your decision making, we suggest standardization. Standardization of how your customers interact with the services you provide is typically the single most impactful action you can take to improve many success metrics across your organization. Standardization makes it easier for product teams to understand the cost structure of your services and make data-driven product decisions. It is easier for operations teams to troubleshoot problems and automate parts of the troubleshooting process in an environment that is developed, rolled out, and operated according to predefined standards. It can help you detect anomalies, unexpected behaviors, or actions by a malicious actor. Standardization also reduces technical debt. It takes fewer cycles for engineering teams to test and roll out changes to production. It can also increase your speed to market, improve self-service onboarding success, and reduce regulatory risk.

Therefore, we suggest that you also review any one-offs that may be in place today. Quantify the number of operational cycles that you spend supporting existing customers. Compare your results with historical data, and assess whether your current approach scales for the years to come. Whenever there is a need to divert from standards, challenge the requirements behind those requests. Evaluate the impact, and balance the immediate benefits with long-term commitments.

In cases where customization is inevitable but in conflict with your standards, consider a shared responsibility model. In this model, your products are largely shielded from the requested changes, and the customization happens in a minimalist, dedicated environment. For an example, see the Connecting with a transit VPC architecture section.

Governance

For compliance with regulatory requirements and your own, internal standards, governance is essential. With proper governance in place, you can control where and how to enforce standards. You also establish to controls to detect divergence from standards and inform resource owners about necessary corrective actions. AWS Organizations, AWS Config, AWS CloudTrail, and AWS Control Tower are a few of many AWS services that can help you manage and govern your workloads in the AWS Cloud.

Repetition

Using learnings from your initial efforts, set up a lightweight, repeatable process to stay aligned in the future. Define which roles you need inputs from, how often, how accurate the data needs to be, how the data will be shared, and who will act on it.

Resources

AWS documentation

- Integrating third-party services in the AWS Cloud (AWS Prescriptive Guidance)
- Multi-tenant SaaS authorization and API access control (AWS Prescriptive Guidance)
- Manage tenants across multiple SaaS products on a single control plane (AWS Prescriptive Guidance)
- What is AWS Direct Connect? (AWS Direct Connect documentation)
- What is AWS PrivateLink? (Amazon VPC documentation)
- What is AWS Site-to-Site VPN? (AWS Site-to-Site VPN documentation)
- What is AWS Transit Gateway? (Amazon VPC documentation)
- What is VPC peering? (Amazon VPC documentation)

Other AWS resources

- Amazon Virtual Private Cloud Connectivity Options (AWS Whitepaper)
- AWS re:Invent 2021 - How to choose the right load balancer for your AWS workloads (YouTube)
- What is SaaS? (AWS website)
- AWS SaaS Factory Program (AWS Partner program)
- Guidance for Multi-Tenant Architectures on AWS (AWS Solutions Library)

Document history

The following table describes significant changes to this guide. If you want to be notified about future updates, you can subscribe to an RSS feed.

Change	Description	Date
Initial publication	—	September 12, 2025

AWS Prescriptive Guidance glossary

The following are commonly used terms in strategies, guides, and patterns provided by AWS Prescriptive Guidance. To suggest entries, please use the **Provide feedback** link at the end of the glossary.

Numbers

7 Rs

Seven common migration strategies for moving applications to the cloud. These strategies build upon the 5 Rs that Gartner identified in 2011 and consist of the following:

- Refactor/re-architect – Move an application and modify its architecture by taking full advantage of cloud-native features to improve agility, performance, and scalability. This typically involves porting the operating system and database. Example: Migrate your on-premises Oracle database to the Amazon Aurora PostgreSQL-Compatible Edition.
- Replatform (lift and reshape) – Move an application to the cloud, and introduce some level of optimization to take advantage of cloud capabilities. Example: Migrate your on-premises Oracle database to Amazon Relational Database Service (Amazon RDS) for Oracle in the AWS Cloud.
- Repurchase (drop and shop) – Switch to a different product, typically by moving from a traditional license to a SaaS model. Example: Migrate your customer relationship management (CRM) system to Salesforce.com.
- Rehost (lift and shift) – Move an application to the cloud without making any changes to take advantage of cloud capabilities. Example: Migrate your on-premises Oracle database to Oracle on an EC2 instance in the AWS Cloud.
- Relocate (hypervisor-level lift and shift) – Move infrastructure to the cloud without purchasing new hardware, rewriting applications, or modifying your existing operations. You migrate servers from an on-premises platform to a cloud service for the same platform. Example: Migrate a Microsoft Hyper-V application to AWS.
- Retain (revisit) – Keep applications in your source environment. These might include applications that require major refactoring, and you want to postpone that work until a later time, and legacy applications that you want to retain, because there’s no business justification for migrating them.

- Retire – Decommission or remove applications that are no longer needed in your source environment.

A

ABAC

See attribute-based access control.

abstracted services

See managed services.

ACID

See atomicity, consistency, isolation, durability.

active-active migration

A database migration method in which the source and target databases are kept in sync (by using a bidirectional replication tool or dual write operations), and both databases handle transactions from connecting applications during migration. This method supports migration in small, controlled batches instead of requiring a one-time cutover. It's more flexible but requires more work than active-passive migration.

active-passive migration

A database migration method in which the source and target databases are kept in sync, but only the source database handles transactions from connecting applications while data is replicated to the target database. The target database doesn't accept any transactions during migration.

aggregate function

A SQL function that operates on a group of rows and calculates a single return value for the group. Examples of aggregate functions include `SUM` and `MAX`.

AI

See artificial intelligence.

AIOps

See artificial intelligence operations.

anonymization

The process of permanently deleting personal information in a dataset. Anonymization can help protect personal privacy. Anonymized data is no longer considered to be personal data.

anti-pattern

A frequently used solution for a recurring issue where the solution is counter-productive, ineffective, or less effective than an alternative.

application control

A security approach that allows the use of only approved applications in order to help protect a system from malware.

application portfolio

A collection of detailed information about each application used by an organization, including the cost to build and maintain the application, and its business value. This information is key to the portfolio discovery and analysis process and helps identify and prioritize the applications to be migrated, modernized, and optimized.

artificial intelligence (AI)

The field of computer science that is dedicated to using computing technologies to perform cognitive functions that are typically associated with humans, such as learning, solving problems, and recognizing patterns. For more information, see What is Artificial Intelligence?

artificial intelligence operations (AIOps)

The process of using machine learning techniques to solve operational problems, reduce operational incidents and human intervention, and increase service quality. For more information about how AIOps is used in the AWS migration strategy, see the operations integration guide.

asymmetric encryption

An encryption algorithm that uses a pair of keys, a public key for encryption and a private key for decryption. You can share the public key because it isn't used for decryption, but access to the private key should be highly restricted.

atomicity, consistency, isolation, durability (ACID)

A set of software properties that guarantee the data validity and operational reliability of a database, even in the case of errors, power failures, or other problems.

attribute-based access control (ABAC)

The practice of creating fine-grained permissions based on user attributes, such as department, job role, and team name. For more information, see ABAC for AWS in the AWS Identity and Access Management (IAM) documentation.

authoritative data source

A location where you store the primary version of data, which is considered to be the most reliable source of information. You can copy data from the authoritative data source to other locations for the purposes of processing or modifying the data, such as anonymizing, redacting, or pseudonymizing it.

Availability Zone

A distinct location within an AWS Region that is insulated from failures in other Availability Zones and provides inexpensive, low-latency network connectivity to other Availability Zones in the same Region.

AWS Cloud Adoption Framework (AWS CAF)

A framework of guidelines and best practices from AWS to help organizations develop an efficient and effective plan to move successfully to the cloud. AWS CAF organizes guidance into six focus areas called perspectives: business, people, governance, platform, security, and operations. The business, people, and governance perspectives focus on business skills and processes; the platform, security, and operations perspectives focus on technical skills and processes. For example, the people perspective targets stakeholders who handle human resources (HR), staffing functions, and people management. For this perspective, AWS CAF provides guidance for people development, training, and communications to help ready the organization for successful cloud adoption. For more information, see the AWS CAF website and the AWS CAF whitepaper.

AWS Workload Qualification Framework (AWS WQF)

A tool that evaluates database migration workloads, recommends migration strategies, and provides work estimates. AWS WQF is included with AWS Schema Conversion Tool (AWS SCT). It analyzes database schemas and code objects, application code, dependencies, and performance characteristics, and provides assessment reports.

B

bad bot

A bot that is intended to disrupt or cause harm to individuals or organizations.

BCP

See business continuity planning.

behavior graph

A unified, interactive view of resource behavior and interactions over time. You can use a behavior graph with Amazon Detective to examine failed logon attempts, suspicious API calls, and similar actions. For more information, see Data in a behavior graph in the Detective documentation.

big-endian system

A system that stores the most significant byte first. See also endianness.

binary classification

A process that predicts a binary outcome (one of two possible classes). For example, your ML model might need to predict problems such as “Is this email spam or not spam?" or "Is this product a book or a car?"

bloom filter

A probabilistic, memory-efficient data structure that is used to test whether an element is a member of a set.

blue/green deployment

A deployment strategy where you create two separate but identical environments. You run the current application version in one environment (blue) and the new application version in the other environment (green). This strategy helps you quickly roll back with minimal impact.

bot

A software application that runs automated tasks over the internet and simulates human activity or interaction. Some bots are useful or beneficial, such as web crawlers that index information on the internet. Some other bots, known as *bad bots*, are intended to disrupt or cause harm to individuals or organizations.

botnet

Networks of bots that are infected by malware and are under the control of a single party, known as a *bot herder* or *bot operator*. Botnets are the best-known mechanism to scale bots and their impact.

branch

A contained area of a code repository. The first branch created in a repository is the *main branch*. You can create a new branch from an existing branch, and you can then develop features or fix bugs in the new branch. A branch you create to build a feature is commonly referred to as a *feature branch*. When the feature is ready for release, you merge the feature branch back into the main branch. For more information, see About branches (GitHub documentation).

break-glass access

In exceptional circumstances and through an approved process, a quick means for a user to gain access to an AWS account that they don't typically have permissions to access. For more information, see the Implement break-glass procedures indicator in the AWS Well-Architected guidance.

brownfield strategy

The existing infrastructure in your environment. When adopting a brownfield strategy for a system architecture, you design the architecture around the constraints of the current systems and infrastructure. If you are expanding the existing infrastructure, you might blend brownfield and greenfield strategies.

buffer cache

The memory area where the most frequently accessed data is stored.

business capability

What a business does to generate value (for example, sales, customer service, or marketing). Microservices architectures and development decisions can be driven by business capabilities. For more information, see the Organized around business capabilities section of the Running containerized microservices on AWS whitepaper.

business continuity planning (BCP)

A plan that addresses the potential impact of a disruptive event, such as a large-scale migration, on operations and enables a business to resume operations quickly.

C

CAF

See AWS Cloud Adoption Framework.

canary deployment

The slow and incremental release of a version to end users. When you are confident, you deploy the new version and replace the current version in its entirety.

CCoE

See Cloud Center of Excellence.

CDC

See change data capture.

change data capture (CDC)

The process of tracking changes to a data source, such as a database table, and recording metadata about the change. You can use CDC for various purposes, such as auditing or replicating changes in a target system to maintain synchronization.

chaos engineering

Intentionally introducing failures or disruptive events to test a system's resilience. You can use AWS Fault Injection Service (AWS FIS) to perform experiments that stress your AWS workloads and evaluate their response.

CI/CD

See continuous integration and continuous delivery.

classification

A categorization process that helps generate predictions. ML models for classification problems predict a discrete value. Discrete values are always distinct from one another. For example, a model might need to evaluate whether or not there is a car in an image.

client-side encryption

Encryption of data locally, before the target AWS service receives it.

Cloud Center of Excellence (CCoE)

A multi-disciplinary team that drives cloud adoption efforts across an organization, including developing cloud best practices, mobilizing resources, establishing migration timelines, and leading the organization through large-scale transformations. For more information, see the CCoE posts on the AWS Cloud Enterprise Strategy Blog.

cloud computing

The cloud technology that is typically used for remote data storage and IoT device management. Cloud computing is commonly connected to edge computing technology.

cloud operating model

In an IT organization, the operating model that is used to build, mature, and optimize one or more cloud environments. For more information, see Building your Cloud Operating Model.

cloud stages of adoption

The four phases that organizations typically go through when they migrate to the AWS Cloud:

- Project – Running a few cloud-related projects for proof of concept and learning purposes
- Foundation – Making foundational investments to scale your cloud adoption (e.g., creating a landing zone, defining a CCoE, establishing an operations model)
- Migration – Migrating individual applications
- Re-invention – Optimizing products and services, and innovating in the cloud

These stages were defined by Stephen Orban in the blog post The Journey Toward Cloud-First & the Stages of Adoption on the AWS Cloud Enterprise Strategy blog. For information about how they relate to the AWS migration strategy, see the migration readiness guide.

CMDB

See configuration management database.

code repository

A location where source code and other assets, such as documentation, samples, and scripts, are stored and updated through version control processes. Common cloud repositories include GitHub or Bitbucket Cloud. Each version of the code is called a *branch*. In a microservice structure, each repository is devoted to a single piece of functionality. A single CI/CD pipeline can use multiple repositories.

cold cache

A buffer cache that is empty, not well populated, or contains stale or irrelevant data. This affects performance because the database instance must read from the main memory or disk, which is slower than reading from the buffer cache.

cold data

Data that is rarely accessed and is typically historical. When querying this kind of data, slow queries are typically acceptable. Moving this data to lower-performing and less expensive storage tiers or classes can reduce costs.

computer vision (CV)

A field of AI that uses machine learning to analyze and extract information from visual formats such as digital images and videos. For example, Amazon SageMaker AI provides image processing algorithms for CV.

configuration drift

For a workload, a configuration change from the expected state. It might cause the workload to become noncompliant, and it's typically gradual and unintentional.

configuration management database (CMDB)

A repository that stores and manages information about a database and its IT environment, including both hardware and software components and their configurations. You typically use data from a CMDB in the portfolio discovery and analysis stage of migration.

conformance pack

A collection of AWS Config rules and remediation actions that you can assemble to customize your compliance and security checks. You can deploy a conformance pack as a single entity in an AWS account and Region, or across an organization, by using a YAML template. For more information, see Conformance packs in the AWS Config documentation.

continuous integration and continuous delivery (CI/CD)

The process of automating the source, build, test, staging, and production stages of the software release process. CI/CD is commonly described as a pipeline. CI/CD can help you automate processes, improve productivity, improve code quality, and deliver faster. For more information, see Benefits of continuous delivery. CD can also stand for *continuous deployment*. For more information, see Continuous Delivery vs. Continuous Deployment.

CV

See computer vision.

D

data at rest

Data that is stationary in your network, such as data that is in storage.

data classification

A process for identifying and categorizing the data in your network based on its criticality and sensitivity. It is a critical component of any cybersecurity risk management strategy because it helps you determine the appropriate protection and retention controls for the data. Data classification is a component of the security pillar in the AWS Well-Architected Framework. For more information, see Data classification.

data drift

A meaningful variation between the production data and the data that was used to train an ML model, or a meaningful change in the input data over time. Data drift can reduce the overall quality, accuracy, and fairness in ML model predictions.

data in transit

Data that is actively moving through your network, such as between network resources.

data mesh

An architectural framework that provides distributed, decentralized data ownership with centralized management and governance.

data minimization

The principle of collecting and processing only the data that is strictly necessary. Practicing data minimization in the AWS Cloud can reduce privacy risks, costs, and your analytics carbon footprint.

data perimeter

A set of preventive guardrails in your AWS environment that help make sure that only trusted identities are accessing trusted resources from expected networks. For more information, see Building a data perimeter on AWS.

data preprocessing

To transform raw data into a format that is easily parsed by your ML model. Preprocessing data can mean removing certain columns or rows and addressing missing, inconsistent, or duplicate values.

data provenance

The process of tracking the origin and history of data throughout its lifecycle, such as how the data was generated, transmitted, and stored.

data subject

An individual whose data is being collected and processed.

data warehouse

A data management system that supports business intelligence, such as analytics. Data warehouses commonly contain large amounts of historical data, and they are typically used for queries and analysis.

database definition language (DDL)

Statements or commands for creating or modifying the structure of tables and objects in a database.

database manipulation language (DML)

Statements or commands for modifying (inserting, updating, and deleting) information in a database.

DDL

See database definition language.

deep ensemble

To combine multiple deep learning models for prediction. You can use deep ensembles to obtain a more accurate prediction or for estimating uncertainty in predictions.

deep learning

An ML subfield that uses multiple layers of artificial neural networks to identify mapping between input data and target variables of interest.

defense-in-depth

An information security approach in which a series of security mechanisms and controls are thoughtfully layered throughout a computer network to protect the confidentiality, integrity, and availability of the network and the data within. When you adopt this strategy on AWS, you add multiple controls at different layers of the AWS Organizations structure to help secure resources. For example, a defense-in-depth approach might combine multi-factor authentication, network segmentation, and encryption.

delegated administrator

In AWS Organizations, a compatible service can register an AWS member account to administer the organization's accounts and manage permissions for that service. This account is called the *delegated administrator* for that service. For more information and a list of compatible services, see Services that work with AWS Organizations in the AWS Organizations documentation.

deployment

The process of making an application, new features, or code fixes available in the target environment. Deployment involves implementing changes in a code base and then building and running that code base in the application's environments.

development environment

See environment.

detective control

A security control that is designed to detect, log, and alert after an event has occurred. These controls are a second line of defense, alerting you to security events that bypassed the preventative controls in place. For more information, see Detective controls in *Implementing security controls on AWS*.

development value stream mapping (DVSM)

A process used to identify and prioritize constraints that adversely affect speed and quality in a software development lifecycle. DVSM extends the value stream mapping process originally designed for lean manufacturing practices. It focuses on the steps and teams required to create and move value through the software development process.

digital twin

A virtual representation of a real-world system, such as a building, factory, industrial equipment, or production line. Digital twins support predictive maintenance, remote monitoring, and production optimization.

dimension table

In a star schema, a smaller table that contains data attributes about quantitative data in a fact table. Dimension table attributes are typically text fields or discrete numbers that behave like text. These attributes are commonly used for query constraining, filtering, and result set labeling.

disaster

An event that prevents a workload or system from fulfilling its business objectives in its primary deployed location. These events can be natural disasters, technical failures, or the result of human actions, such as unintentional misconfiguration or a malware attack.

disaster recovery (DR)

The strategy and process you use to minimize downtime and data loss caused by a disaster. For more information, see Disaster Recovery of Workloads on AWS: Recovery in the Cloud in the AWS Well-Architected Framework.

DML

See database manipulation language.

domain-driven design

An approach to developing a complex software system by connecting its components to evolving domains, or core business goals, that each component serves. This concept was introduced by Eric Evans in his book, *Domain-Driven Design: Tackling Complexity in the Heart of Software* (Boston: Addison-Wesley Professional, 2003). For information about how you can use domain-driven design with the strangler fig pattern, see Modernizing legacy Microsoft ASP.NET (ASMX) web services incrementally by using containers and Amazon API Gateway.

DR

See disaster recovery.

drift detection

Tracking deviations from a baselined configuration. For example, you can use AWS CloudFormation to detect drift in system resources, or you can use AWS Control Tower to detect changes in your landing zone that might affect compliance with governance requirements.

DVSM

See development value stream mapping.

E

EDA

See exploratory data analysis.

EDI

See electronic data interchange.

edge computing

The technology that increases the computing power for smart devices at the edges of an IoT network. When compared with cloud computing, edge computing can reduce communication latency and improve response time.

electronic data interchange (EDI)

The automated exchange of business documents between organizations. For more information, see What is Electronic Data Interchange.

encryption

A computing process that transforms plaintext data, which is human-readable, into ciphertext.

encryption key

A cryptographic string of randomized bits that is generated by an encryption algorithm. Keys can vary in length, and each key is designed to be unpredictable and unique.

endianness

The order in which bytes are stored in computer memory. Big-endian systems store the most significant byte first. Little-endian systems store the least significant byte first.

endpoint

See service endpoint.

endpoint service

A service that you can host in a virtual private cloud (VPC) to share with other users. You can create an endpoint service with AWS PrivateLink and grant permissions to other AWS accounts or to AWS Identity and Access Management (IAM) principals. These accounts or principals can connect to your endpoint service privately by creating interface VPC endpoints. For more information, see Create an endpoint service in the Amazon Virtual Private Cloud (Amazon VPC) documentation.

enterprise resource planning (ERP)

A system that automates and manages key business processes (such as accounting, MES, and project management) for an enterprise.

envelope encryption

The process of encrypting an encryption key with another encryption key. For more information, see Envelope encryption in the AWS Key Management Service (AWS KMS) documentation.

environment

An instance of a running application. The following are common types of environments in cloud computing:

- development environment – An instance of a running application that is available only to the core team responsible for maintaining the application. Development environments are used to test changes before promoting them to upper environments. This type of environment is sometimes referred to as a *test environment*.
- lower environments – All development environments for an application, such as those used for initial builds and tests.
- production environment – An instance of a running application that end users can access. In a CI/CD pipeline, the production environment is the last deployment environment.
- upper environments – All environments that can be accessed by users other than the core development team. This can include a production environment, preproduction environments, and environments for user acceptance testing.

epic

In agile methodologies, functional categories that help organize and prioritize your work. Epics provide a high-level description of requirements and implementation tasks. For example, AWS CAF security epics include identity and access management, detective controls, infrastructure security, data protection, and incident response. For more information about epics in the AWS migration strategy, see the program implementation guide.

ERP

See enterprise resource planning.

exploratory data analysis (EDA)

The process of analyzing a dataset to understand its main characteristics. You collect or aggregate data and then perform initial investigations to find patterns, detect anomalies, and check assumptions. EDA is performed by calculating summary statistics and creating data visualizations.

F

fact table

The central table in a star schema. It stores quantitative data about business operations. Typically, a fact table contains two types of columns: those that contain measures and those that contain a foreign key to a dimension table.

fail fast

A philosophy that uses frequent and incremental testing to reduce the development lifecycle. It is a critical part of an agile approach.

fault isolation boundary

In the AWS Cloud, a boundary such as an Availability Zone, AWS Region, control plane, or data plane that limits the effect of a failure and helps improve the resilience of workloads. For more information, see AWS Fault Isolation Boundaries.

feature branch

See branch.

features

The input data that you use to make a prediction. For example, in a manufacturing context, features could be images that are periodically captured from the manufacturing line.

feature importance

How significant a feature is for a model's predictions. This is usually expressed as a numerical score that can be calculated through various techniques, such as Shapley Additive Explanations (SHAP) and integrated gradients. For more information, see Machine learning model interpretability with AWS.

feature transformation

To optimize data for the ML process, including enriching data with additional sources, scaling values, or extracting multiple sets of information from a single data field. This enables the ML model to benefit from the data. For example, if you break down the “2021-05-27 00:15:37” date into “2021”, “May”, “Thu”, and “15”, you can help the learning algorithm learn nuanced patterns associated with different data components.

few-shot prompting

Providing an LLM with a small number of examples that demonstrate the task and desired output before asking it to perform a similar task. This technique is an application of in-context learning, where models learn from examples (*shots*) that are embedded in prompts. Few-shot prompting can be effective for tasks that require specific formatting, reasoning, or domain knowledge. See also zero-shot prompting.

FGAC

See fine-grained access control.

fine-grained access control (FGAC)

The use of multiple conditions to allow or deny an access request.

flash-cut migration

A database migration method that uses continuous data replication through change data capture to migrate data in the shortest time possible, instead of using a phased approach. The objective is to keep downtime to a minimum.

FM

See foundation model.

foundation model (FM)

A large deep-learning neural network that has been training on massive datasets of generalized and unlabeled data. FMs are capable of performing a wide variety of general tasks, such as understanding language, generating text and images, and conversing in natural language. For more information, see What are Foundation Models.

G

generative AI

A subset of AI models that have been trained on large amounts of data and that can use a simple text prompt to create new content and artifacts, such as images, videos, text, and audio. For more information, see What is Generative AI.

geo blocking

See geographic restrictions.

geographic restrictions (geo blocking)

In Amazon CloudFront, an option to prevent users in specific countries from accessing content distributions. You can use an allow list or block list to specify approved and banned countries. For more information, see Restricting the geographic distribution of your content in the CloudFront documentation.

Gitflow workflow

An approach in which lower and upper environments use different branches in a source code repository. The Gitflow workflow is considered legacy, and the trunk-based workflow is the modern, preferred approach.

golden image

A snapshot of a system or software that is used as a template to deploy new instances of that system or software. For example, in manufacturing, a golden image can be used to provision software on multiple devices and helps improve speed, scalability, and productivity in device manufacturing operations.

greenfield strategy

The absence of existing infrastructure in a new environment. When adopting a greenfield strategy for a system architecture, you can select all new technologies without the restriction of compatibility with existing infrastructure, also known as brownfield. If you are expanding the existing infrastructure, you might blend brownfield and greenfield strategies.

guardrail

A high-level rule that helps govern resources, policies, and compliance across organizational units (OUs). *Preventive guardrails* enforce policies to ensure alignment to compliance standards. They are implemented by using service control policies and IAM permissions boundaries.

Detective guardrails detect policy violations and compliance issues, and generate alerts for remediation. They are implemented by using AWS Config, AWS Security Hub, Amazon GuardDuty, AWS Trusted Advisor, Amazon Inspector, and custom AWS Lambda checks.

H

HA

See high availability.

heterogeneous database migration

Migrating your source database to a target database that uses a different database engine (for example, Oracle to Amazon Aurora). Heterogeneous migration is typically part of a re-architecting effort, and converting the schema can be a complex task. AWS provides AWS SCT that helps with schema conversions.

high availability (HA)

The ability of a workload to operate continuously, without intervention, in the event of challenges or disasters. HA systems are designed to automatically fail over, consistently deliver high-quality performance, and handle different loads and failures with minimal performance impact.

historian modernization

An approach used to modernize and upgrade operational technology (OT) systems to better serve the needs of the manufacturing industry. A *historian* is a type of database that is used to collect and store data from various sources in a factory.

holdout data

A portion of historical, labeled data that is withheld from a dataset that is used to train a machine learning model. You can use holdout data to evaluate the model performance by comparing the model predictions against the holdout data.

homogeneous database migration

Migrating your source database to a target database that shares the same database engine (for example, Microsoft SQL Server to Amazon RDS for SQL Server). Homogeneous migration is typically part of a rehosting or replatforming effort. You can use native database utilities to migrate the schema.

hot data

Data that is frequently accessed, such as real-time data or recent translational data. This data typically requires a high-performance storage tier or class to provide fast query responses.

hotfix

An urgent fix for a critical issue in a production environment. Due to its urgency, a hotfix is usually made outside of the typical DevOps release workflow.

hypercare period

Immediately following cutover, the period of time when a migration team manages and monitors the migrated applications in the cloud in order to address any issues. Typically, this period is 1–4 days in length. At the end of the hypercare period, the migration team typically transfers responsibility for the applications to the cloud operations team.

I

IaC

See infrastructure as code.

identity-based policy

A policy attached to one or more IAM principals that defines their permissions within the AWS Cloud environment.

idle application

An application that has an average CPU and memory usage between 5 and 20 percent over a period of 90 days. In a migration project, it is common to retire these applications or retain them on premises.

IIoT

See industrial Internet of Things.

immutable infrastructure

A model that deploys new infrastructure for production workloads instead of updating, patching, or modifying the existing infrastructure. Immutable infrastructures are inherently more consistent, reliable, and predictable than mutable infrastructure. For more information, see the Deploy using immutable infrastructure best practice in the AWS Well-Architected Framework.

inbound (ingress) VPC

In an AWS multi-account architecture, a VPC that accepts, inspects, and routes network connections from outside an application. The AWS Security Reference Architecture recommends setting up your Network account with inbound, outbound, and inspection VPCs to protect the two-way interface between your application and the broader internet.

incremental migration

A cutover strategy in which you migrate your application in small parts instead of performing a single, full cutover. For example, you might move only a few microservices or users to the new system initially. After you verify that everything is working properly, you can incrementally move additional microservices or users until you can decommission your legacy system. This strategy reduces the risks associated with large migrations.

Industry 4.0

A term that was introduced by Klaus Schwab in 2016 to refer to the modernization of manufacturing processes through advances in connectivity, real-time data, automation, analytics, and AI/ML.

infrastructure

All of the resources and assets contained within an application's environment.

infrastructure as code (IaC)

The process of provisioning and managing an application's infrastructure through a set of configuration files. IaC is designed to help you centralize infrastructure management, standardize resources, and scale quickly so that new environments are repeatable, reliable, and consistent.

industrial Internet of Things (IIoT)

The use of internet-connected sensors and devices in the industrial sectors, such as manufacturing, energy, automotive, healthcare, life sciences, and agriculture. For more information, see Building an industrial Internet of Things (IIoT) digital transformation strategy.

inspection VPC

In an AWS multi-account architecture, a centralized VPC that manages inspections of network traffic between VPCs (in the same or different AWS Regions), the internet, and on-premises networks. The AWS Security Reference Architecture recommends setting up your Network account with inbound, outbound, and inspection VPCs to protect the two-way interface between your application and the broader internet.

Internet of Things (IoT)

The network of connected physical objects with embedded sensors or processors that communicate with other devices and systems through the internet or over a local communication network. For more information, see What is IoT?

interpretability

A characteristic of a machine learning model that describes the degree to which a human can understand how the model's predictions depend on its inputs. For more information, see Machine learning model interpretability with AWS.

IoT

See Internet of Things.

IT information library (ITIL)

A set of best practices for delivering IT services and aligning these services with business requirements. ITIL provides the foundation for ITSM.

IT service management (ITSM)

Activities associated with designing, implementing, managing, and supporting IT services for an organization. For information about integrating cloud operations with ITSM tools, see the operations integration guide.

ITIL

See IT information library.

ITSM

See IT service management.

L

label-based access control (LBAC)

An implementation of mandatory access control (MAC) where the users and the data itself are each explicitly assigned a security label value. The intersection between the user security label and data security label determines which rows and columns can be seen by the user.

landing zone

A landing zone is a well-architected, multi-account AWS environment that is scalable and secure. This is a starting point from which your organizations can quickly launch and deploy workloads and applications with confidence in their security and infrastructure environment. For more information about landing zones, see Setting up a secure and scalable multi-account AWS environment.

large language model (LLM)

A deep learning AI model that is pretrained on a vast amount of data. An LLM can perform multiple tasks, such as answering questions, summarizing documents, translating text into other languages, and completing sentences. For more information, see What are LLMs.

large migration

A migration of 300 or more servers.

LBAC

See label-based access control.

least privilege

The security best practice of granting the minimum permissions required to perform a task. For more information, see Apply least-privilege permissions in the IAM documentation.

lift and shift

See 7 Rs.

little-endian system

A system that stores the least significant byte first. See also endianness.

LLM

See large language model.

lower environments

See environment.

M

machine learning (ML)

A type of artificial intelligence that uses algorithms and techniques for pattern recognition and learning. ML analyzes and learns from recorded data, such as Internet of Things (IoT) data, to generate a statistical model based on patterns. For more information, see Machine Learning.

main branch

See branch.

malware

Software that is designed to compromise computer security or privacy. Malware might disrupt computer systems, leak sensitive information, or gain unauthorized access. Examples of malware include viruses, worms, ransomware, Trojan horses, spyware, and keyloggers.

managed services

AWS services for which AWS operates the infrastructure layer, the operating system, and platforms, and you access the endpoints to store and retrieve data. Amazon Simple Storage Service (Amazon S3) and Amazon DynamoDB are examples of managed services. These are also known as *abstracted services*.

manufacturing execution system (MES)

A software system for tracking, monitoring, documenting, and controlling production processes that convert raw materials to finished products on the shop floor.

MAP

See Migration Acceleration Program.

mechanism

A complete process in which you create a tool, drive adoption of the tool, and then inspect the results in order to make adjustments. A mechanism is a cycle that reinforces and improves itself as it operates. For more information, see Building mechanisms in the AWS Well-Architected Framework.

member account

All AWS accounts other than the management account that are part of an organization in AWS Organizations. An account can be a member of only one organization at a time.

MES

See manufacturing execution system.

Message Queuing Telemetry Transport (MQTT)

A lightweight, machine-to-machine (M2M) communication protocol, based on the publish/subscribe pattern, for resource-constrained IoT devices.

microservice

A small, independent service that communicates over well-defined APIs and is typically owned by small, self-contained teams. For example, an insurance system might include microservices that map to business capabilities, such as sales or marketing, or subdomains, such as purchasing, claims, or analytics. The benefits of microservices include agility, flexible scaling, easy deployment, reusable code, and resilience. For more information, see Integrating microservices by using AWS serverless services.

microservices architecture

An approach to building an application with independent components that run each application process as a microservice. These microservices communicate through a well-defined interface by using lightweight APIs. Each microservice in this architecture can be updated, deployed, and scaled to meet demand for specific functions of an application. For more information, see Implementing microservices on AWS.

Migration Acceleration Program (MAP)

An AWS program that provides consulting support, training, and services to help organizations build a strong operational foundation for moving to the cloud, and to help offset the initial cost of migrations. MAP includes a migration methodology for executing legacy migrations in a methodical way and a set of tools to automate and accelerate common migration scenarios.

migration at scale

The process of moving the majority of the application portfolio to the cloud in waves, with more applications moved at a faster rate in each wave. This phase uses the best practices and lessons learned from the earlier phases to implement a *migration factory* of teams, tools, and processes to streamline the migration of workloads through automation and agile delivery. This is the third phase of the AWS migration strategy.

migration factory

Cross-functional teams that streamline the migration of workloads through automated, agile approaches. Migration factory teams typically include operations, business analysts and owners, migration engineers, developers, and DevOps professionals working in sprints. Between 20 and 50 percent of an enterprise application portfolio consists of repeated patterns that can be optimized by a factory approach. For more information, see the discussion of migration factories and the Cloud Migration Factory guide in this content set.

migration metadata

The information about the application and server that is needed to complete the migration. Each migration pattern requires a different set of migration metadata. Examples of migration metadata include the target subnet, security group, and AWS account.

migration pattern

A repeatable migration task that details the migration strategy, the migration destination, and the migration application or service used. Example: Rehost migration to Amazon EC2 with AWS Application Migration Service.

Migration Portfolio Assessment (MPA)

An online tool that provides information for validating the business case for migrating to the AWS Cloud. MPA provides detailed portfolio assessment (server right-sizing, pricing, TCO comparisons, migration cost analysis) as well as migration planning (application data analysis and data collection, application grouping, migration prioritization, and wave planning). The MPA tool (requires login) is available free of charge to all AWS consultants and APN Partner consultants.

Migration Readiness Assessment (MRA)

The process of gaining insights about an organization's cloud readiness status, identifying strengths and weaknesses, and building an action plan to close identified gaps, using the AWS CAF. For more information, see the migration readiness guide. MRA is the first phase of the AWS migration strategy.

migration strategy

The approach used to migrate a workload to the AWS Cloud. For more information, see the 7 Rs entry in this glossary and see Mobilize your organization to accelerate large-scale migrations.

ML

See machine learning.

modernization

Transforming an outdated (legacy or monolithic) application and its infrastructure into an agile, elastic, and highly available system in the cloud to reduce costs, gain efficiencies, and take advantage of innovations. For more information, see Strategy for modernizing applications in the AWS Cloud.

modernization readiness assessment

An evaluation that helps determine the modernization readiness of an organization’s applications; identifies benefits, risks, and dependencies; and determines how well the organization can support the future state of those applications. The outcome of the assessment is a blueprint of the target architecture, a roadmap that details development phases and milestones for the modernization process, and an action plan for addressing identified gaps. For more information, see Evaluating modernization readiness for applications in the AWS Cloud.

monolithic applications (monoliths)

Applications that run as a single service with tightly coupled processes. Monolithic applications have several drawbacks. If one application feature experiences a spike in demand, the entire architecture must be scaled. Adding or improving a monolithic application’s features also becomes more complex when the code base grows. To address these issues, you can use a microservices architecture. For more information, see Decomposing monoliths into microservices.

MPA

See Migration Portfolio Assessment.

MQTT

See Message Queuing Telemetry Transport.

multiclass classification

A process that helps generate predictions for multiple classes (predicting one of more than two outcomes). For example, an ML model might ask "Is this product a book, car, or phone?" or "Which product category is most interesting to this customer?"

mutable infrastructure

A model that updates and modifies the existing infrastructure for production workloads. For improved consistency, reliability, and predictability, the AWS Well-Architected Framework recommends the use of immutable infrastructure as a best practice.

O

OAC

See origin access control.

OAI

See origin access identity.

OCM

See organizational change management.

offline migration

A migration method in which the source workload is taken down during the migration process. This method involves extended downtime and is typically used for small, non-critical workloads.

OI

See operations integration.

OLA

See operational-level agreement.

online migration

A migration method in which the source workload is copied to the target system without being taken offline. Applications that are connected to the workload can continue to function during the migration. This method involves zero to minimal downtime and is typically used for critical production workloads.

OPC-UA

See Open Process Communications - Unified Architecture.

Open Process Communications - Unified Architecture (OPC-UA)

A machine-to-machine (M2M) communication protocol for industrial automation. OPC-UA provides an interoperability standard with data encryption, authentication, and authorization schemes.

operational-level agreement (OLA)

An agreement that clarifies what functional IT groups promise to deliver to each other, to support a service-level agreement (SLA).

operational readiness review (ORR)

A checklist of questions and associated best practices that help you understand, evaluate, prevent, or reduce the scope of incidents and possible failures. For more information, see Operational Readiness Reviews (ORR) in the AWS Well-Architected Framework.

operational technology (OT)

Hardware and software systems that work with the physical environment to control industrial operations, equipment, and infrastructure. In manufacturing, the integration of OT and information technology (IT) systems is a key focus for Industry 4.0 transformations.

operations integration (OI)

The process of modernizing operations in the cloud, which involves readiness planning, automation, and integration. For more information, see the operations integration guide.

organization trail

A trail that's created by AWS CloudTrail that logs all events for all AWS accounts in an organization in AWS Organizations. This trail is created in each AWS account that's part of the organization and tracks the activity in each account. For more information, see Creating a trail for an organization in the CloudTrail documentation.

organizational change management (OCM)

A framework for managing major, disruptive business transformations from a people, culture, and leadership perspective. OCM helps organizations prepare for, and transition to, new systems and strategies by accelerating change adoption, addressing transitional issues, and driving cultural and organizational changes. In the AWS migration strategy, this framework is called *people acceleration*, because of the speed of change required in cloud adoption projects. For more information, see the OCM guide.

origin access control (OAC)

In CloudFront, an enhanced option for restricting access to secure your Amazon Simple Storage Service (Amazon S3) content. OAC supports all S3 buckets in all AWS Regions, server-side encryption with AWS KMS (SSE-KMS), and dynamic `PUT` and `DELETE` requests to the S3 bucket.

origin access identity (OAI)

In CloudFront, an option for restricting access to secure your Amazon S3 content. When you use OAI, CloudFront creates a principal that Amazon S3 can authenticate with. Authenticated principals can access content in an S3 bucket only through a specific CloudFront distribution. See also OAC, which provides more granular and enhanced access control.

ORR

See operational readiness review.

OT

See operational technology.

outbound (egress) VPC

In an AWS multi-account architecture, a VPC that handles network connections that are initiated from within an application. The AWS Security Reference Architecture recommends setting up your Network account with inbound, outbound, and inspection VPCs to protect the two-way interface between your application and the broader internet.

P

permissions boundary

An IAM management policy that is attached to IAM principals to set the maximum permissions that the user or role can have. For more information, see Permissions boundaries in the IAM documentation.

personally identifiable information (PII)

Information that, when viewed directly or paired with other related data, can be used to reasonably infer the identity of an individual. Examples of PII include names, addresses, and contact information.

PII

See personally identifiable information.

playbook

A set of predefined steps that capture the work associated with migrations, such as delivering core operations functions in the cloud. A playbook can take the form of scripts, automated runbooks, or a summary of processes or steps required to operate your modernized environment.

PLC

See programmable logic controller.

PLM

See product lifecycle management.

policy

An object that can define permissions (see identity-based policy), specify access conditions (see resource-based policy), or define the maximum permissions for all accounts in an organization in AWS Organizations (see service control policy).

polyglot persistence

Independently choosing a microservice's data storage technology based on data access patterns and other requirements. If your microservices have the same data storage technology, they can encounter implementation challenges or experience poor performance. Microservices are more easily implemented and achieve better performance and scalability if they use the data store best adapted to their requirements. For more information, see Enabling data persistence in microservices.

portfolio assessment

A process of discovering, analyzing, and prioritizing the application portfolio in order to plan the migration. For more information, see Evaluating migration readiness.

predicate

A query condition that returns `true` or `false`, commonly located in a `WHERE` clause.

predicate pushdown

A database query optimization technique that filters the data in the query before transfer. This reduces the amount of data that must be retrieved and processed from the relational database, and it improves query performance.

preventative control

A security control that is designed to prevent an event from occurring. These controls are a first line of defense to help prevent unauthorized access or unwanted changes to your network. For more information, see Preventative controls in *Implementing security controls on AWS*.

principal

An entity in AWS that can perform actions and access resources. This entity is typically a root user for an AWS account, an IAM role, or a user. For more information, see *Principal* in Roles terms and concepts in the IAM documentation.

privacy by design

A system engineering approach that takes privacy into account through the whole development process.

private hosted zones

A container that holds information about how you want Amazon Route 53 to respond to DNS queries for a domain and its subdomains within one or more VPCs. For more information, see Working with private hosted zones in the Route 53 documentation.

proactive control

A security control designed to prevent the deployment of noncompliant resources. These controls scan resources before they are provisioned. If the resource is not compliant with the control, then it isn't provisioned. For more information, see the Controls reference guide in the AWS Control Tower documentation and see Proactive controls in *Implementing security controls on AWS*.

product lifecycle management (PLM)

The management of data and processes for a product throughout its entire lifecycle, from design, development, and launch, through growth and maturity, to decline and removal.

production environment

See environment.

programmable logic controller (PLC)

In manufacturing, a highly reliable, adaptable computer that monitors machines and automates manufacturing processes.

prompt chaining

Using the output of one LLM prompt as the input for the next prompt to generate better responses. This technique is used to break down a complex task into subtasks, or to iteratively refine or expand a preliminary response. It helps improve the accuracy and relevance of a model’s responses and allows for more granular, personalized results.

pseudonymization

The process of replacing personal identifiers in a dataset with placeholder values. Pseudonymization can help protect personal privacy. Pseudonymized data is still considered to be personal data.

publish/subscribe (pub/sub)

A pattern that enables asynchronous communications among microservices to improve scalability and responsiveness. For example, in a microservices-based MES, a microservice can publish event messages to a channel that other microservices can subscribe to. The system can add new microservices without changing the publishing service.

Q

query plan

A series of steps, like instructions, that are used to access the data in a SQL relational database system.

query plan regression

When a database service optimizer chooses a less optimal plan than it did before a given change to the database environment. This can be caused by changes to statistics, constraints, environment settings, query parameter bindings, and updates to the database engine.

R

RACI matrix

See responsible, accountable, consulted, informed (RACI).

RAG

See Retrieval Augmented Generation.

ransomware

A malicious software that is designed to block access to a computer system or data until a payment is made.

RASCI matrix

See responsible, accountable, consulted, informed (RACI).

RCAC

See row and column access control.

read replica

A copy of a database that’s used for read-only purposes. You can route queries to the read replica to reduce the load on your primary database.

re-architect

See 7 Rs.

recovery point objective (RPO)

The maximum acceptable amount of time since the last data recovery point. This determines what is considered an acceptable loss of data between the last recovery point and the interruption of service.

recovery time objective (RTO)

The maximum acceptable delay between the interruption of service and restoration of service.

refactor

See 7 Rs.

Region

A collection of AWS resources in a geographic area. Each AWS Region is isolated and independent of the others to provide fault tolerance, stability, and resilience. For more information, see Specify which AWS Regions your account can use.

regression

An ML technique that predicts a numeric value. For example, to solve the problem of "What price will this house sell for?" an ML model could use a linear regression model to predict a house's sale price based on known facts about the house (for example, the square footage).

rehost

See 7 Rs.

release

In a deployment process, the act of promoting changes to a production environment.

relocate

See 7 Rs.

replatform

See 7 Rs.

repurchase

See 7 Rs.

resiliency

An application's ability to resist or recover from disruptions. High availability and disaster recovery are common considerations when planning for resiliency in the AWS Cloud. For more information, see AWS Cloud Resilience.

resource-based policy

A policy attached to a resource, such as an Amazon S3 bucket, an endpoint, or an encryption key. This type of policy specifies which principals are allowed access, supported actions, and any other conditions that must be met.

responsible, accountable, consulted, informed (RACI) matrix

A matrix that defines the roles and responsibilities for all parties involved in migration activities and cloud operations. The matrix name is derived from the responsibility types defined in the matrix: responsible (R), accountable (A), consulted (C), and informed (I). The support (S) type is optional. If you include support, the matrix is called a *RASCI matrix*, and if you exclude it, it's called a *RACI matrix*.

responsive control

A security control that is designed to drive remediation of adverse events or deviations from your security baseline. For more information, see Responsive controls in *Implementing security controls on AWS*.

retain

See 7 Rs.

retire

See 7 Rs.

Retrieval Augmented Generation (RAG)

A generative AI technology in which an LLM references an authoritative data source that is outside of its training data sources before generating a response. For example, a RAG model might perform a semantic search of an organization's knowledge base or custom data. For more information, see What is RAG.

rotation

The process of periodically updating a secret to make it more difficult for an attacker to access the credentials.

row and column access control (RCAC)

The use of basic, flexible SQL expressions that have defined access rules. RCAC consists of row permissions and column masks.

RPO

See recovery point objective.

RTO

See recovery time objective.

runbook

A set of manual or automated procedures required to perform a specific task. These are typically built to streamline repetitive operations or procedures with high error rates.

S

SAML 2.0

An open standard that many identity providers (IdPs) use. This feature enables federated single sign-on (SSO), so users can log into the AWS Management Console or call the AWS API operations without you having to create user in IAM for everyone in your organization. For more information about SAML 2.0-based federation, see About SAML 2.0-based federation in the IAM documentation.

SCADA

See supervisory control and data acquisition.

SCP

See service control policy.

secret

In AWS Secrets Manager, confidential or restricted information, such as a password or user credentials, that you store in encrypted form. It consists of the secret value and its metadata.

The secret value can be binary, a single string, or multiple strings. For more information, see What's in a Secrets Manager secret? in the Secrets Manager documentation.

security by design

A system engineering approach that takes security into account through the whole development process.

security control

A technical or administrative guardrail that prevents, detects, or reduces the ability of a threat actor to exploit a security vulnerability. There are four primary types of security controls: preventative, detective, responsive, and proactive.

security hardening

The process of reducing the attack surface to make it more resistant to attacks. This can include actions such as removing resources that are no longer needed, implementing the security best practice of granting least privilege, or deactivating unnecessary features in configuration files.

security information and event management (SIEM) system

Tools and services that combine security information management (SIM) and security event management (SEM) systems. A SIEM system collects, monitors, and analyzes data from servers, networks, devices, and other sources to detect threats and security breaches, and to generate alerts.

security response automation

A predefined and programmed action that is designed to automatically respond to or remediate a security event. These automations serve as detective or responsive security controls that help you implement AWS security best practices. Examples of automated response actions include modifying a VPC security group, patching an Amazon EC2 instance, or rotating credentials.

server-side encryption

Encryption of data at its destination, by the AWS service that receives it.

service control policy (SCP)

A policy that provides centralized control over permissions for all accounts in an organization in AWS Organizations. SCPs define guardrails or set limits on actions that an administrator can delegate to users or roles. You can use SCPs as allow lists or deny lists, to specify which services or actions are permitted or prohibited. For more information, see Service control policies in the AWS Organizations documentation.

service endpoint

The URL of the entry point for an AWS service. You can use the endpoint to connect programmatically to the target service. For more information, see AWS service endpoints in *AWS General Reference*.

service-level agreement (SLA)

An agreement that clarifies what an IT team promises to deliver to their customers, such as service uptime and performance.

service-level indicator (SLI)

A measurement of a performance aspect of a service, such as its error rate, availability, or throughput.

service-level objective (SLO)

A target metric that represents the health of a service, as measured by a service-level indicator.

shared responsibility model

A model describing the responsibility you share with AWS for cloud security and compliance. AWS is responsible for security *of* the cloud, whereas you are responsible for security *in* the cloud. For more information, see Shared responsibility model.

SIEM

See security information and event management system.

single point of failure (SPOF)

A failure in a single, critical component of an application that can disrupt the system.

SLA

See service-level agreement.

SLI

See service-level indicator.

SLO

See service-level objective.

split-and-seed model

A pattern for scaling and accelerating modernization projects. As new features and product releases are defined, the core team splits up to create new product teams. This helps scale your organization's capabilities and services, improves developer productivity, and supports rapid innovation. For more information, see Phased approach to modernizing applications in the AWS Cloud.

SPOF

See single point of failure.

star schema

A database organizational structure that uses one large fact table to store transactional or measured data and uses one or more smaller dimensional tables to store data attributes. This structure is designed for use in a data warehouse or for business intelligence purposes.

strangler fig pattern

An approach to modernizing monolithic systems by incrementally rewriting and replacing system functionality until the legacy system can be decommissioned. This pattern uses the analogy of a fig vine that grows into an established tree and eventually overcomes and replaces its host. The pattern was introduced by Martin Fowler as a way to manage risk when rewriting monolithic systems. For an example of how to apply this pattern, see Modernizing legacy Microsoft ASP.NET (ASMX) web services incrementally by using containers and Amazon API Gateway.

subnet

A range of IP addresses in your VPC. A subnet must reside in a single Availability Zone.

supervisory control and data acquisition (SCADA)

In manufacturing, a system that uses hardware and software to monitor physical assets and production operations.

symmetric encryption

An encryption algorithm that uses the same key to encrypt and decrypt the data.

synthetic testing

Testing a system in a way that simulates user interactions to detect potential issues or to monitor performance. You can use Amazon CloudWatch Synthetics to create these tests.

system prompt

A technique for providing context, instructions, or guidelines to an LLM to direct its behavior. System prompts help set context and establish rules for interactions with users.

T

tags

Key-value pairs that act as metadata for organizing your AWS resources. Tags can help you manage, identify, organize, search for, and filter resources. For more information, see Tagging your AWS resources.

target variable

The value that you are trying to predict in supervised ML. This is also referred to as an *outcome variable*. For example, in a manufacturing setting the target variable could be a product defect.

task list

A tool that is used to track progress through a runbook. A task list contains an overview of the runbook and a list of general tasks to be completed. For each general task, it includes the estimated amount of time required, the owner, and the progress.

test environment

See environment.

training

To provide data for your ML model to learn from. The training data must contain the correct answer. The learning algorithm finds patterns in the training data that map the input data attributes to the target (the answer that you want to predict). It outputs an ML model that captures these patterns. You can then use the ML model to make predictions on new data for which you don’t know the target.

transit gateway

A network transit hub that you can use to interconnect your VPCs and on-premises networks. For more information, see What is a transit gateway in the AWS Transit Gateway documentation.

trunk-based workflow

An approach in which developers build and test features locally in a feature branch and then merge those changes into the main branch. The main branch is then built to the development, preproduction, and production environments, sequentially.

trusted access

Granting permissions to a service that you specify to perform tasks in your organization in AWS Organizations and in its accounts on your behalf. The trusted service creates a service-linked role in each account, when that role is needed, to perform management tasks for you. For more information, see Using AWS Organizations with other AWS services in the AWS Organizations documentation.

tuning

To change aspects of your training process to improve the ML model's accuracy. For example, you can train the ML model by generating a labeling set, adding labels, and then repeating these steps several times under different settings to optimize the model.

two-pizza team

A small DevOps team that you can feed with two pizzas. A two-pizza team size ensures the best possible opportunity for collaboration in software development.

U

uncertainty

A concept that refers to imprecise, incomplete, or unknown information that can undermine the reliability of predictive ML models. There are two types of uncertainty: *Epistemic uncertainty* is caused by limited, incomplete data, whereas *aleatoric uncertainty* is caused by the noise and randomness inherent in the data. For more information, see the Quantifying uncertainty in deep learning systems guide.

undifferentiated tasks

Also known as *heavy lifting*, work that is necessary to create and operate an application but that doesn’t provide direct value to the end user or provide competitive advantage. Examples of undifferentiated tasks include procurement, maintenance, and capacity planning.

upper environments

See environment.

V

vacuuming

A database maintenance operation that involves cleaning up after incremental updates to reclaim storage and improve performance.

version control

Processes and tools that track changes, such as changes to source code in a repository.

VPC peering

A connection between two VPCs that allows you to route traffic by using private IP addresses. For more information, see What is VPC peering in the Amazon VPC documentation.

vulnerability

A software or hardware flaw that compromises the security of the system.

W

warm cache

A buffer cache that contains current, relevant data that is frequently accessed. The database instance can read from the buffer cache, which is faster than reading from the main memory or disk.

warm data

Data that is infrequently accessed. When querying this kind of data, moderately slow queries are typically acceptable.

window function

A SQL function that performs a calculation on a group of rows that relate in some way to the current record. Window functions are useful for processing tasks, such as calculating a moving average or accessing the value of rows based on the relative position of the current row.

workload

A collection of resources and code that delivers business value, such as a customer-facing application or backend process.

workstream

Functional groups in a migration project that are responsible for a specific set of tasks. Each workstream is independent but supports the other workstreams in the project. For example, the portfolio workstream is responsible for prioritizing applications, wave planning, and collecting migration metadata. The portfolio workstream delivers these assets to the migration workstream, which then migrates the servers and applications.

WORM

See write once, read many.

WQF

See AWS Workload Qualification Framework.

write once, read many (WORM)

A storage model that writes data a single time and prevents the data from being deleted or modified. Authorized users can read the data as many times as needed, but they cannot change it. This data storage infrastructure is considered immutable.

Z

zero-day exploit

An attack, typically malware, that takes advantage of a zero-day vulnerability.

zero-day vulnerability

An unmitigated flaw or vulnerability in a production system. Threat actors can use this type of vulnerability to attack the system. Developers frequently become aware of the vulnerability as a result of the attack.

zero-shot prompting

Providing an LLM with instructions for performing a task but no examples (*shots*) that can help guide it. The LLM must use its pre-trained knowledge to handle the task. The effectiveness of zero-shot prompting depends on the complexity of the task and the quality of the prompt. See also few-shot prompting.

zombie application

An application that has an average CPU and memory usage below 5 percent. In a migration project, it is common to retire these applications.