Architecting an Amazon Web Services Account Governance and VPC Design Strategy

Published 15 May 2018 - ID G00347966 - 45 min read

By Analysts Jim Burton, Elias Khnaser

Supporting Key Initiative is Cloud Computing

Designing an effective AWS account governance and VPC design strategy provides technical professionals the ability to effectively scale, avoid sprawl, reduce networking and management complexities, and automate. This research offers several AWS account and VPC design options and best practices.

More on This Topic

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

- What You Need to Know Before Buying and Deploying Cloud Offerings: A Gartner Trend Insight Report

Overview

Key Findings

- Organizations are often faced with AWS account and Virtual Private Cloud (VPC) sprawl due to a lack of clear criteria that dictates when and why an AWS account or VPC should be created.
- Creating multiple AWS accounts can improve isolation, billing and security, but it greatly complicates the overall management of the AWS infrastructure.
- Organizations typically spend a considerable amount of time getting AWS accounts, VPCs, role-based access controls (RBACs) and management policies in place prior to widespread usage. Having defined criteria and an automated system greatly reduces complexity.
- Organizations with a large developer community often struggle to deliver an automated self-service AWS account and VPC life cycle management process.

Recommendations

Technical professionals focused on cloud computing within Amazon Web Services should:

- Select your AWS account and VPC governance model based on the organizational structure of your company to establish solid justification criteria for account and VPC creation.
- Automate and enable self-service of end-to-end AWS account and VPC life cycle management using AWS Organizations.
- Use scripts and orchestration templates, such as Amazon CloudFormation, to further extend the AWS Organizations automation capacities.
- Use existing in-house structures for financial accounting practices (for example, cost centers) as a guide to how similar hierarchies within AWS could be implemented.
- Work to ensure that adequate identity management and RBACs are implemented within the platform from the start. Refactoring permissions underneath established applications can be highly disruptive.

Problem Statement

In today's fast-paced world, digital business heavily prioritizes time to market in order to achieve a competitive advantage and capture market share. Although time to market definitely gives an organization that initial advantage, it also hinders its ability to effectively scale its business and maintain its competitive advantage.

Cloud services exacerbate this problem because speed and immediate access to infrastructure and services are but a credit card swipe away. Hence, it is important for organizations to balance the speed of adoption with careful governance planning. This balance allows organizations to scale effectively at their desired speed.

This research provides technical professionals with guidance on the core governance items to understand and plan, thus avoiding disruptive retrofitting of the infrastructure months or years after it has transformed into a critical production platform.

The Gartner Approach

Properly designing an environment with governance is a slow process. However, it is time well-spent because it allows the organization to adopt cloud faster and at a sustained rate. Effective governance allows for:

- Role-based access control, which gives the right users the right permissions.
- Billing transparency and the foundation for chargeback or showback.

- A repeatable process that is the foundation of effective scalability, especially when coupled with automation.
- Rapid deployment of environments, according to validated designs.

If we examine the early days of server virtualization, it is apparent that virtual environments were built on a whim — without proper planning or designing. This lack of planning led to virtual machine sprawl, misconfigured and underperforming storage arrays, and increased costs of managing the environment. In every instance, technical professionals had to go back, properly design the environment and optimize it in order to gain maximum performance and capacity of the different assets.

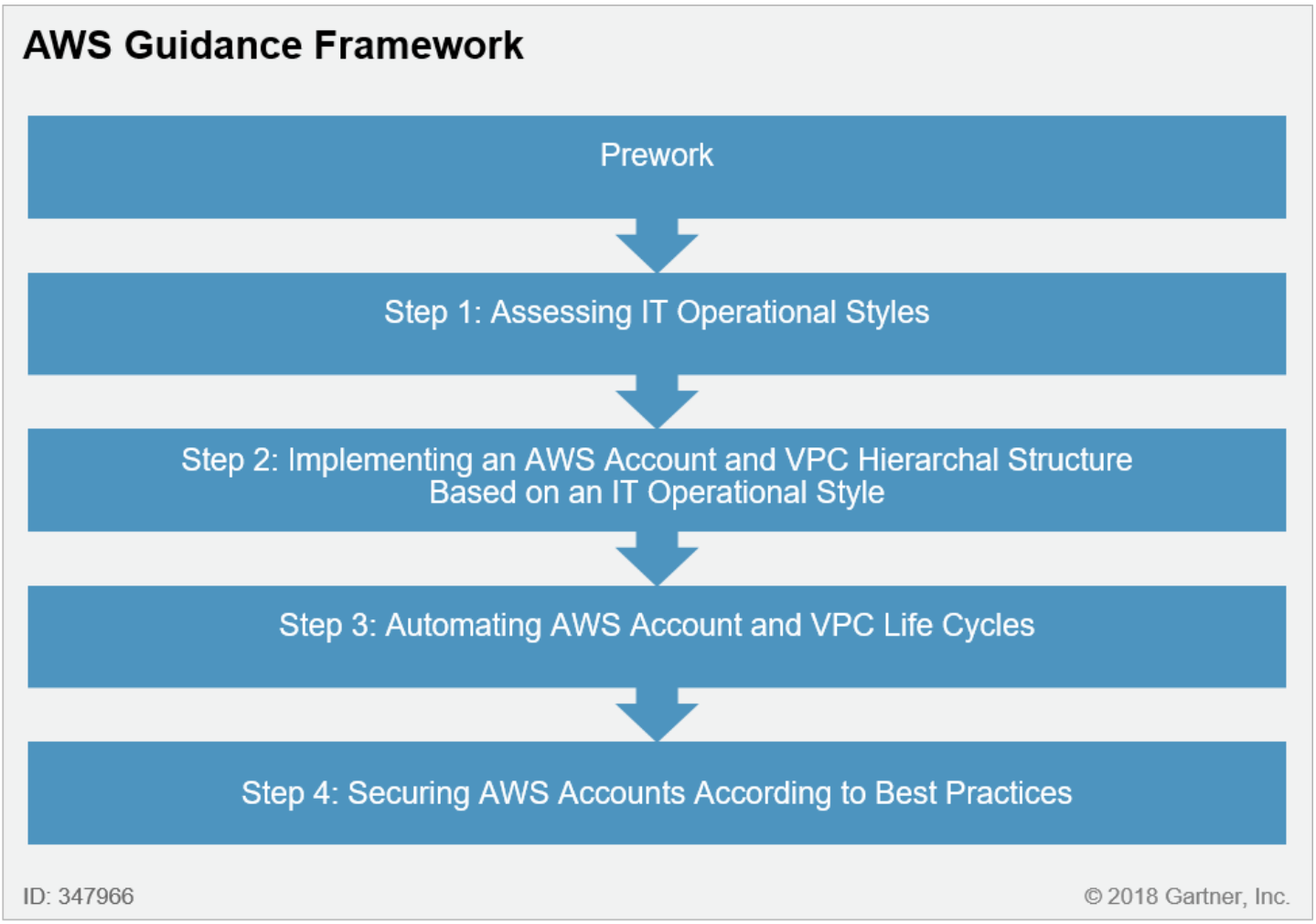
The public cloud era amplifies the need for proper planning and designing because most cloud environments began as side projects that were not IT-led, and therefore, they had little or no governance. This led to security concerns, inefficient use of resources, increased costs, difficulty operating the environment and, as a result, difficulty scaling the environment in a manageable way.

Gartner's approach to planning and designing a production-grade environment on Amazon Web Services (AWS) focuses on key governance elements that organizations will need to consider in advance to minimize potential refactoring disruptions at a later date.

The Guidance Framework

Gartner's governance framework for Amazon Web Services focuses on several initial setup tasks that organizations should spend sufficient time planning during the adoption phase. Organizations that focus on these core items at the start of AWS adoption will avoid having to refactor foundational architectural elements, which is typically difficult to do once usage starts to increase. The framework is divided into prework and four key phases, as shown in Figure 1.

Figure 1. AWS Governance Framework



Source: Gartner (May 2018)

Phases of the AWS governance framework:

- **Prework:** Establishes a foundation for the strategy. During this stage, technical professionals identify the current AWS governance maturity level and establish a solid understanding of AWS terminology. They will also assess the trade-offs between having one or multiple AWS accounts and Amazon Virtual Private Clouds (VPCs).
- **Step 1 — Assessing IT Operational Styles:** Presents four IT operational style and aids technical professionals in assessing which style their organization most closely adheres to.
- **Step 2 — Implementing an AWS Account and VPC Hierarchal Structure Based on an IT Operational Style:** Details the various AWS accounts and VPC hierarchal structures, including pros and cons of each.
- **Step 3 — Automating AWS Account and VPC Life Cycles:** Defines how to automate AWS account and VPC life cycle management using AWS Organizations.

- **Step 4 — Securing AWS Accounts According to Best Practices:** Outlines the AWS account security best practices.

Prework

Before attempting to plot out a production-grade governance framework within Amazon Web Services, technical professionals need to understand:

- Fundamentals of AWS governance
- Current AWS governance maturity level
- Challenges of having a single AWS account

Fundamentals of AWS Governance

AWS services are full of somewhat unique acronyms and terminology. It is important to know what these are in order to understand the rest of this research.

AWS Accounts

An AWS account offers the highest level of isolation between your resources, users and financial obligation. AWS accounts are tenants within the AWS multitenant environment that isolate resources from other AWS accounts. These accounts have absolute separation for every aspect of service, including security, billing, networking and business processes. Billing is handled through the AWS account structure. Usage for the account can be rolled up to a central account, or AWS tagging can deliver a more fine-grained billing based on project or employee if multiple employees are sharing a single account. AWS Organizations was recently launched and is an important tool for billing aggregation.

Identity and Access Management

AWS Identity and Access Management (IAM) is configured at the AWS account level. AWS uses its Identity and Access Management solution to automate and centrally control user access to AWS resources. User accounts are created in the AWS Management Console and they can either be assigned privileges at the individual account level or consolidated into IAM groups according to job function. Once in an IAM group, privileges can be assigned to the group, and every user within the group will have the same privileges. Multifactor authentication and password rotation can be added to further enhance security. For more information, see "Implementing an Identity Strategy for Amazon Web Services." (https://www.gartner.com/document/code/292666?ref=grbody&refval=3875212)

Cross-Account Roles

As the organization's multiaccount strategy grows, there will be a need to standardize and centrally delegate access to resources in different accounts. Using cross-account roles, technical

professionals can create roles and permissions in one account and grant users access to resources in other accounts. For example, a developer may need access to the development and testing accounts. They may also need access to the production account to fix problems. This requires that the developer have different roles in different accounts with varying degrees of permissions. Creating a cross-account role allows this developer to gain granular access to specific resources across all accounts without having to create and assign permissions in each account. This standardization allows technical professionals to consistently grant or deny access to users at scale.

Amazon Virtual Private Cloud

A VPC allows technical professionals to create a logically isolated virtual network on the AWS infrastructure. Multiple VPCs can exist in the same AWS account, and they are isolated from one another and from any other virtual network on the AWS infrastructure. Technical professionals can organize, secure and isolate resources using subnets, security groups and network access control lists (NACLs). There is a lot that can be done with networking at the VPC level. However, in some cases, the business process requirements or the level of isolation that the organization needs is not met by networking alone — hence the need for another AWS account.

Additionally, a VPC allows technical professionals to launch some resources and services within it. These services and resources benefit from the logical isolation capabilities of a VPC. However, it is important to note that not all services and resources can be launched within a VPC, which creates a need for isolation at the account level. The next section highlights the services and resources that can be launched within a VPC.

VPC Integration

Not all AWS services integrate with VPC. These services are internet-facing, with public IP addresses. Therefore, technical professionals may need to create separate AWS accounts in order to isolate AWS services that are at the account level (another potential reason for a multiaccount strategy). AWS services that do integrate with VPCs benefit from the network fence that VPCs offer. They also benefit from the fact that all their IP addresses are private. Technical professionals must be able to identify services that integrate with VPC in order to determine whether or not it is a factor that justifies the creation of another account.

Examples of services that integrate with VPCs include:

- Amazon EC2
- Elastic Load Balancing (ELB)
- Amazon Redshift
- AWS Direct Connect (DX)
- Amazon Elastic File System (EFS)

- AWS Elastic Beanstalk
- Amazon CloudHSM
- Amazon SageMaker
- AWS Lambda
- Amazon Relational Database Service (RDS)
- Amazon MQ
- Amazon Elasticsearch
- Amazon Elastic MapReduce (EMR)
- Amazon API Gateway
- Amazon AppStream

About 76 out of 100-plus AWS services have meaningful integration with Virtual Private Cloud. The remaining services — such as Amazon Polly, Amazon Lex and Amazon Athena — do not currently have VPC integration and, in some cases, may not need that integration. For these types of services, isolation is accomplished by having a separate AWS account because isolation is not possible with VPC.

AWS Organizations Overview

AWS Organizations is a powerful, centrally managed tool that is used to create and apply policy-based management to multiple AWS accounts without requiring custom scripts or manual processes.

AWS Organizations has the following benefits:

- Automate the creation of multiple AWS accounts through APIs.
- Logically group AWS accounts for simplified management.
- Centrally apply policies to control access to AWS services across multiple AWS accounts through service control policies (SCPs).
- Centrally enable consolidated billing for multiple AWS accounts.
- Centralize identity management.
- AWS Organizations is free.

Technical professionals who are just getting started with AWS should use AWS Organizations because it simplifies and standardizes the creation, ongoing management and security of multiple AWS accounts. Technical professionals who already have multiple accounts or who don't have a desire to use AWS Organizations can still apply all the principles, approaches and best practices discussed in this research, except they would be implementing them manually. For example, consolidated billing can be configured between multiple AWS accounts manually, or it can be automatically enabled for multiple accounts by using AWS Organizations.

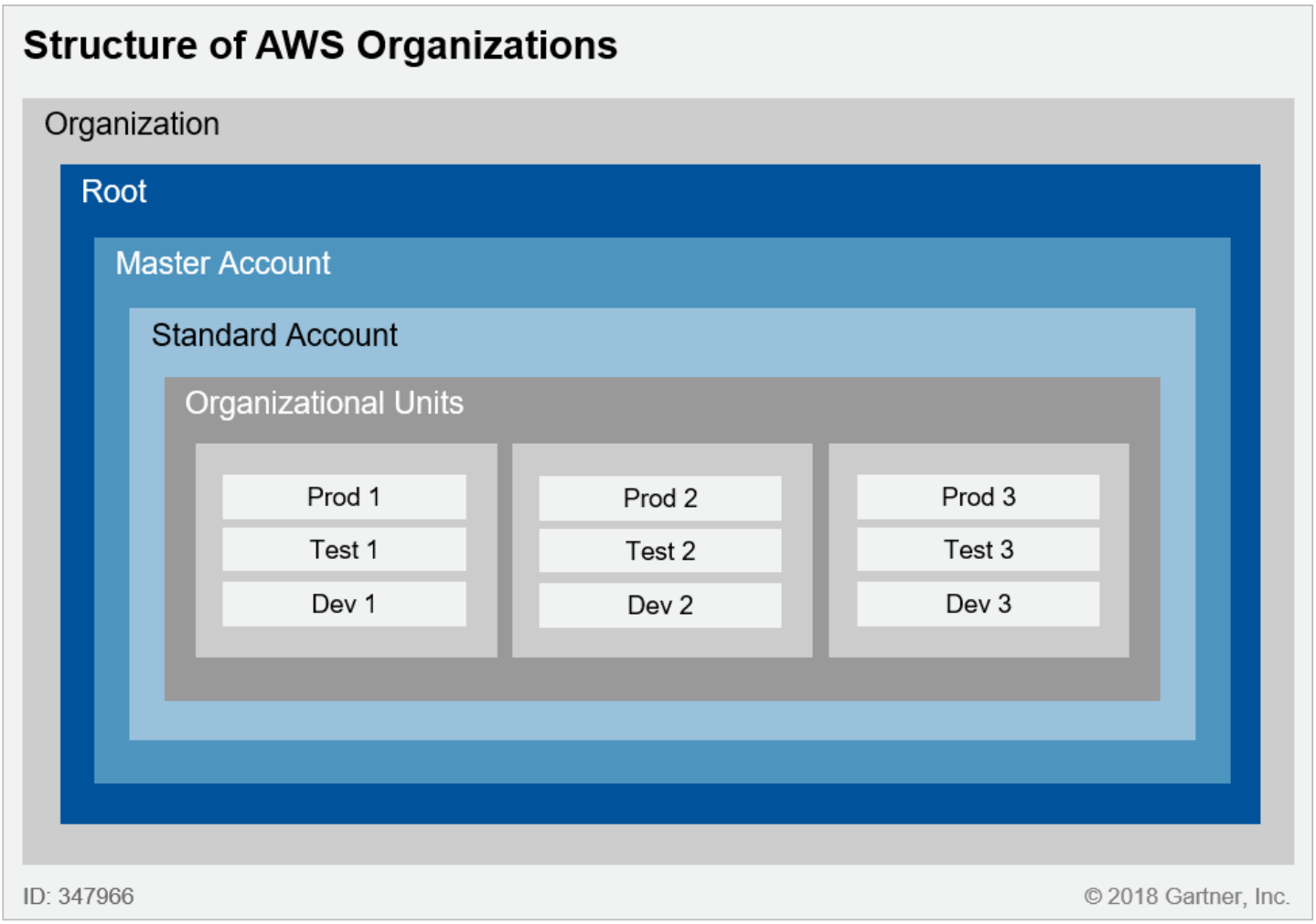
Figure 2 shows a typical structure for AWS Organizations. If you previously had a consolidated billing family, Organizations converted it to an organization, and the payer account was migrated to the master account of the organization.

The master account automatically assumes full administrative control and can be used to:

- View details of all accounts
- Create new accounts
- Move accounts
- Delete accounts
- Change privileges

The master account is also used to create organizational units (OUs). In this case, three organizational units have been created according to corporate department, and eight accounts have been assigned to its respective OU.

Figure 2. Structure of AWS Organizations



Source: Gartner (May 2018)

Organization

An organization is an entity within AWS Organizations that is created to consolidate AWS accounts. An organization will have one master account and, based on your requirements, can have multiple standard accounts. An organization can also be organized in a hierarchal tree with a root and organizational units. Accounts can be created directly under the root or nested within OUs.

Root

The root is the parent container of all accounts in AWS Organizations. Any policies applied at the root account will be applied to all OUs and accounts in the organization by default. The root is automatically created as soon as your organization is created.

Master Account

Every organization has one account designated as the master account. This account consolidates billing for all other participating member accounts. The master account shoulders the responsibility for all billing activities and is responsible for paying all bills.

Standard Account

A standard account is one that contains all AWS services and assets that you are consuming within that account.

Organizational Units

The organizational unit is a container of AWS accounts within a root. OUs are a way of grouping accounts together to form a single unit. Instead of managing individual accounts, you can attach policy-based controls to OUs, and the accounts within that OU will automatically inherit those controls. You can nest OUs within other OUs, and you can move accounts around among OUs. With the introduction of AWS Organizations, OUs take on a powerful role in managing accounts.

Be aware of some OU characteristics:

- OU depth cannot exceed five levels
- OU name must be unique within an organization
- Service control policies are the only supported policy type
- You cannot automate the creation of the root password
- You cannot automate the activation of enterprise support plans
- You cannot automate the setup of an alternate email contacts

AWS Organizations Permissions and Control Management

When using AWS Organizations, technical professionals can use the same IAM permission policies that they have used in the past to control access to resources. Additionally, service control policies can be used to restrict access to services and actions. IAM and SCPs can be used in collaboration to automate permissions and control management.

IAM Permissions

Identity and Access Management permissions control what users, groups or roles within your organizations can and cannot do within AWS. IAM grants or denies permission to access AWS APIs and specific resources. IAM can also be used to grant permissions during specific times of the day or from a specific IP address.

An identity account can be created using AWS Organizations that allows identity and access to be managed from a central location. This allows users and groups to access resources in other accounts using IAM cross-account roles. This can be accomplished by creating an IAM role and using AWS Security Token Service (STS). These roles grant temporary access based on an established trust relationship between AWS accounts. In a similar manner, IAM can be used to federate users from a different directory.

Service Control Policies

Technical professionals can use SCPs to control which actions within an AWS service can be accessed by the account root, IAM users and IAM roles. When an SCP is attached to an account, the effective permissions are those that are explicitly permitted by IAM and explicitly permitted by the SCP. Effectively, the cumulative permissions between IAM and SCP are least privileged.

The important thing to remember is that IAM permissions must be granted before they can be filtered using SCPs. Without IAM permissions, users and groups will have no permission. SCPs cannot grant permission. They can only restrict permissions for access to services and actions.

AWS Organizations allows SCPs to be attached to all levels, including root, OUs and individual accounts. If applied to the root, that filtering applies to the root and all of the OUs and individual accounts attached to the root. No entity under the root has permissions if it has been blocked at a higher level, such as the root. Gartner strongly recommends attaching SCPs to OUs rather than to the root due to the far-reaching effects of SCPs at the root level.

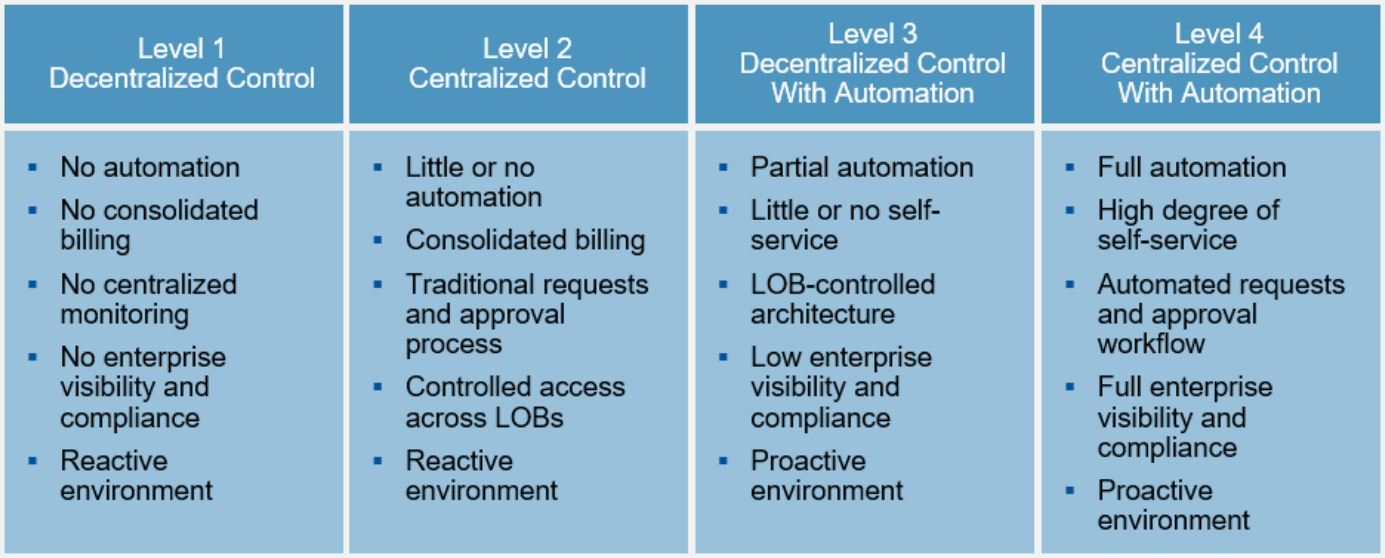
Current AWS Governance Maturity Level

Technical professionals must assess their current state in order to identify gaps that they need to focus on. Figure 3 offers four levels of maturity that technical professionals can use to assess their current state:

- Level 1: Decentralized Control
- Level 2: Centralized Control
- Level 3: Decentralized Control With Automation
- Level 4: Centralized Control With Automation

Figure 3. AWS Accounts and VPC Governance Maturity Levels

AWS Accounts and VPC Governance Maturity Levels



Level 1 Decentralized Control	Level 2 Centralized Control	Level 3 Decentralized Control With Automation	Level 4 Centralized Control With Automation
▪ No automation ▪ No consolidated billing ▪ No centralized monitoring ▪ No enterprise visibility and compliance ▪ Reactive environment	▪ Little or no automation ▪ Consolidated billing ▪ Traditional requests and approval process ▪ Controlled access across LOBs ▪ Reactive environment	▪ Partial automation ▪ Little or no self-service ▪ LOB-controlled architecture ▪ Low enterprise visibility and compliance ▪ Proactive environment	▪ Full automation ▪ High degree of self-service ▪ Automated requests and approval workflow ▪ Full enterprise visibility and compliance ▪ Proactive environment

ID: 347966

Source: Gartner (May 2018)

Each maturity level is composed of general assumptions to help you further identify your current state:

- Level of automation and self-service
- Use of consolidated billing
- Centralized or decentralized monitoring
- Approval workflow approach
- Degree of compliance and visibility
- Proactive or reactive environment

For example, Levels 3 and 4 are characterized primarily by automation. Consequently, as you are assessing your maturity level, if you feel that you are in Level 3 or Level 4 but lack automation, it inherently identifies that your next focus area should be automation in order to optimize your deployment.

Challenges of Having a Single AWS Account

One account means everything is in one place, making it easy to view and track the overall AWS spending for the company, standardize security, and isolate resources at the network level by using VPCs. However, that means having to use a process such as financial tagging to break out the bills

for various organizations and entities within the company. That also means carving out the VPC into multiple subnets for isolation and management purposes. Depending on the organizational structures and business processes of organizations, a single account and VPC approach may not meet their needs.

Multiple accounts are useful in many scenarios. One such scenario is when groups within the company operate autonomously on a business and financial level. Another example could be organizations that are geographically separated with different budgets, security and regulatory compliance requirements.

Technical professionals often struggle to understand the benefits of a multiaccount strategy and the complexities involved with implementing one. Therefore, they must first determine if a single AWS account can meet their business requirements. In order to do that, they must examine the challenges associated with operating with a single AWS account by considering:

- Financial governance
- Multiple tenants and resource isolation
- Security governance
- Business processes
- Resource governance

Financial Governance

Billing is often the culprit that pushes technical professionals to adopt multiple AWS accounts. The complexity of establishing and enforcing a robust tagging strategy often leads to a billing nightmare. Hence, using a multi-AWS-account strategy becomes a lot more attractive, even if only to simplify billing and chargeback.

Multiple Tenants and Resource Isolation

Many organizations have several development teams that are working on separate projects. Isolating these teams within a single AWS account is definitely possible using IAM and VPCs. Technical professionals would also need a robust tagging structure to ensure billing transparency and chargeback are possible.

In addition to isolating tenants, technical professionals may want to also isolate resources and allow or prevent applications or services from communicating with each other. For example, if you had a database with sensitive information, you may want a higher level of resource isolation for this database in order to prevent other applications from being able to access its information. In single account and VPC, we now have to consider separate subnets, security groups, NACLs and locked-

down VPC peering. All of these are certainly possible, they are but time-consuming and complex. They are also prone to potentially damaging mistakes if misconfigured.

Security Governance

The security controls are defined by the AWS account where AWS IAM, security groups and network access control lists can be applied to restrict where users can go and what they can do. Most companies choose to establish a dedicated information security account in an Amazon Simple Storage Service (S3) bucket where security-related issues can be collected and analyzed and cross-account role access can be granted.

Business Processes

Technical professionals must also consider that the organization may have different business units with different processes from one another. These business units could be using different development groups and different pipelines and have different requirements and budgets. These teams may be using different third-party tools for their applications — for example, different identity providers or different encryption requirements. Implementing these in a single account is possible, but very complex and time-consuming. Hence, it is critical for technical architects to understand the business objectives and the processes that an organization needs in order to design an AWS governance framework that is easy to manage and is scalable.

Resource Governance

Resource governance is another important factor to consider as you evaluate whether one AWS account is enough for your organization. Having separate AWS accounts allows technical professionals to overcome hard account limits. See the Amazon VPC Limits (https://docs.aws.amazon.com/general/latest/gr/aws_service_limits.html#limits_vpc) page for more information on AWS account and VPC soft and hard limits.

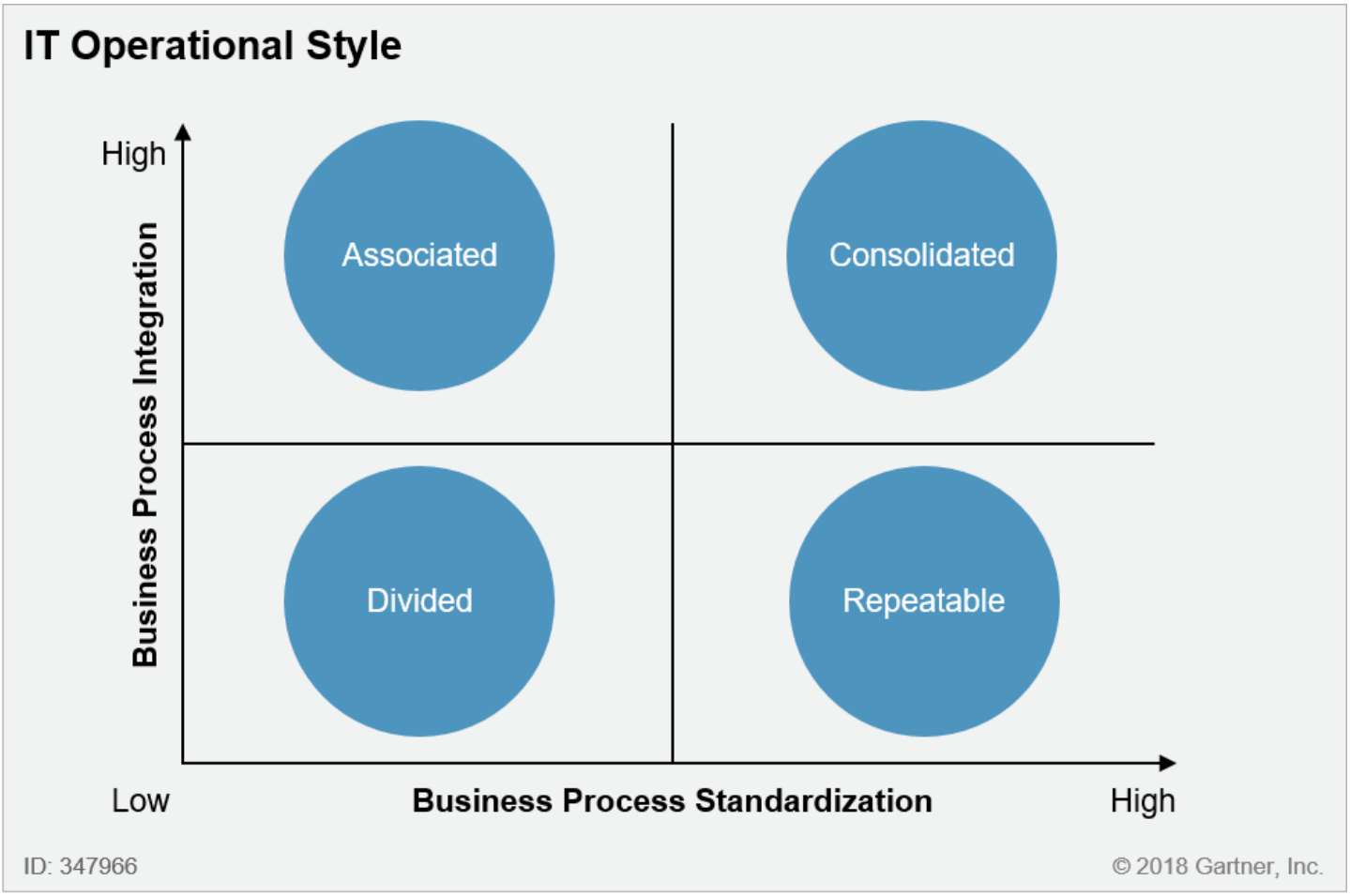
While considering multiple accounts, it is also important to keep in mind that:

- Resources cannot span multiple AWS accounts. However, resources in one account can be accessed from another account if the right permissions are granted.
- Resources cannot dynamically move between AWS accounts.

Framework Step No. 1: Assessing IT Operational Styles

Before implementing an AWS account and VPC governance strategy, assess and identify which IT operational style your organization most closely adheres to. Figure 4 displays four examples of operational styles. These examples are classified into these four categories based on the amount of standardization and integration that they have across their business.

Figure 4. IT Operational Styles



Source: Gartner (May 2018)

Consolidated

The company operates as a single business unit with standardized processes and unified global data access. This style is high on business integration and standardization.

The key business goals that the consolidated operating style requires are:

- Centralized management
- Standardized processes
- Shared infrastructure and data

Divided

The company operates as independent business units with different customers, unique processes and separate global data access. This style is low on business integration and standardization.

The key business goals that the divided operating style requires are:

- Each line of business (LOB) has separate application requirements.
- Each LOB has autonomous decision-making capabilities.
- Each LOB has different financial structure.
- LOBs have no standardized IT processes.
- LOBs do no share infrastructure.
- LOBs share little or no customer and product data.

Associated

The company operates as independent, unique business units serving a common customer. This style is high on business integration and low on standardization.

The key business goals that the associated operating style requires are:

- Each LOB has different standardized IT processes.
- Each LOB has separate application requirements.
- Each LOB has autonomous decision-making capabilities.
- All LOBs share infrastructure.
- All LOBs share customer and product data.

Repeatable

The company operates as independent, but similar, business units sharing best practices and repeatable processes.

The key business goals that the repeatable operating style requires are:

- Each LOB has separate application requirements.
- LOBs have standardized IT processes and infrastructure via a shared services model.
- LOBs have standardized data structures, but data is maintained by each line of business.
- LOBs share little or no customer and product data.

Framework Step No. 2: Implementing an AWS Account and VPC Hierarchal Structure Based on an IT Operational Style

There are a number of variations of hierarchal structures that can be implemented for AWS accounts and VPCs. Technical professionals should thoroughly assess their IT operational style. They should implement the option that best fits with their organization and that meets the current business requirements without hindering organizational growth and expandability.

This step offers five commonly used options:

- Option 1: Consolidated
- Option 2: Divided
- Option 3: Associated
- Option 4: Repeatable
- Option 5: Gartner Commonly Observed

Option 1: Consolidated

The consolidated style is the most straightforward and easiest to implement of all the models. It closely resembles the on-premises deployment model and follows similar principles. In Figure 5, the suggested design is similar to a centralized on-premises data center with a single network partitioned and isolated using subnets. Cloud architects may choose to partition a subnet for production with a separate subnet for development and test. Alternatively, subnets can be used for each application. Note that, when using this consolidated approach, isolation is limited to network access control lists and security groups. Cloud architects will also extend existing role-based access control processes by integrating with AWS Identity and Access Management. However, because everything is centralized, RBAC is more complex, which makes it more difficult to implement and manage.

This design has several advantages, such as:

- Simplified networking connectivity and management
- Identity integration and federation
- Infrastructure management
- Monitoring

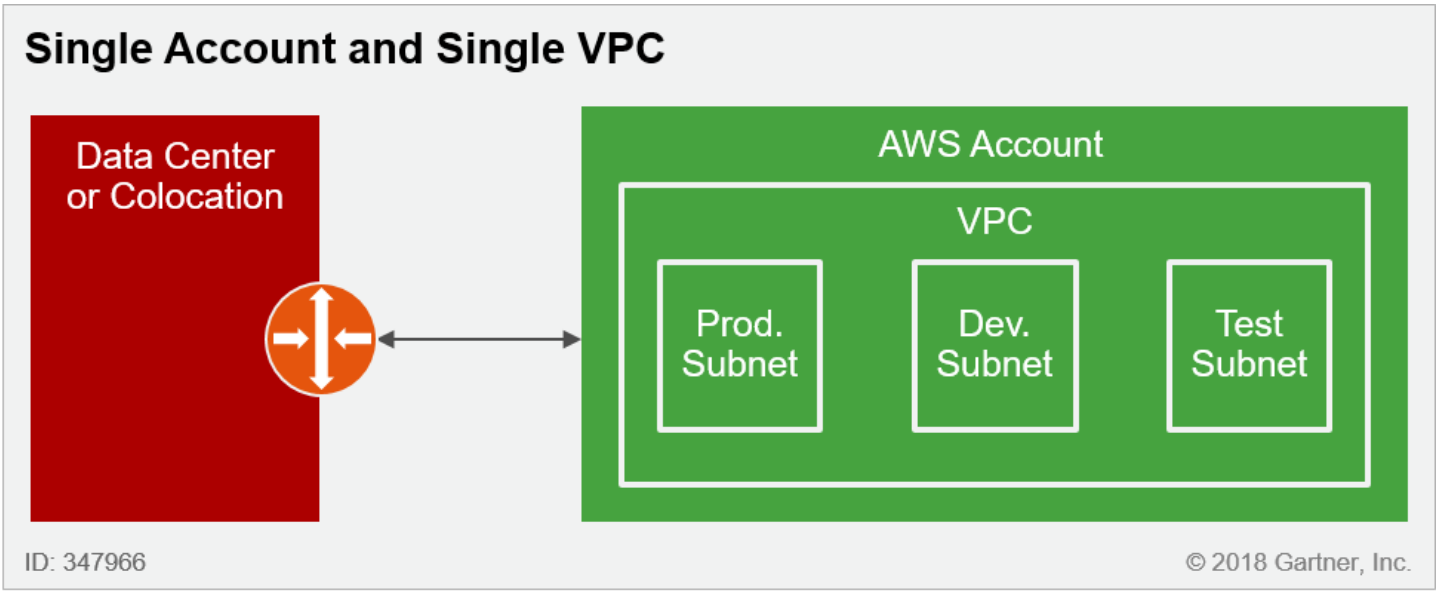
This design also has several disadvantages, such as the potential to reach account and VPC limits faster. Remember, AWS accounts and VPCs have soft and hard limits that, if reached, would require the cloud architect to expand the AWS infrastructure by adding another VPC or potentially creating another AWS account. For the consolidated option, cloud architects must be familiar with AWS account and VPC limits to make a sound technical decision on the governance model. An effective

governance model will serve the organization today and allow for scalability and flexibility in the future. See the Amazon VPC Limits (https://docs.aws.amazon.com/general/latest/gr/aws_service_limits.html#limits_vpc) page for more information on AWS account and VPC soft and hard limits.

The consolidated option is ideal for:

- Small organizations that are just getting started on AWS
- Organizations that have assessed the pros and cons, have especially analyzed the AWS service limits, and don't think they will ever need to grow beyond those limits
- Organizations that have a centralized IT team with standardized processes

Figure 5. Single Account and Single VPC



Source: Gartner (May 2018)

Pros

The consolidated IT operating style has several favorable design characteristics:

- Built on existing on-premises data center concepts and best practices to facilitate transition to the cloud
- Centralized and simplified infrastructure management
- Centralized and simplified network connectivity options
- Use of existing on-premises security processes and controls to manage AWS infrastructure
- Blast radius control based on AWS IAM, security groups and NACLs

- Cost allocation tagging enforced at the workload level
- Simplified chargeback or showback using AWS Cost Explorer

Cons

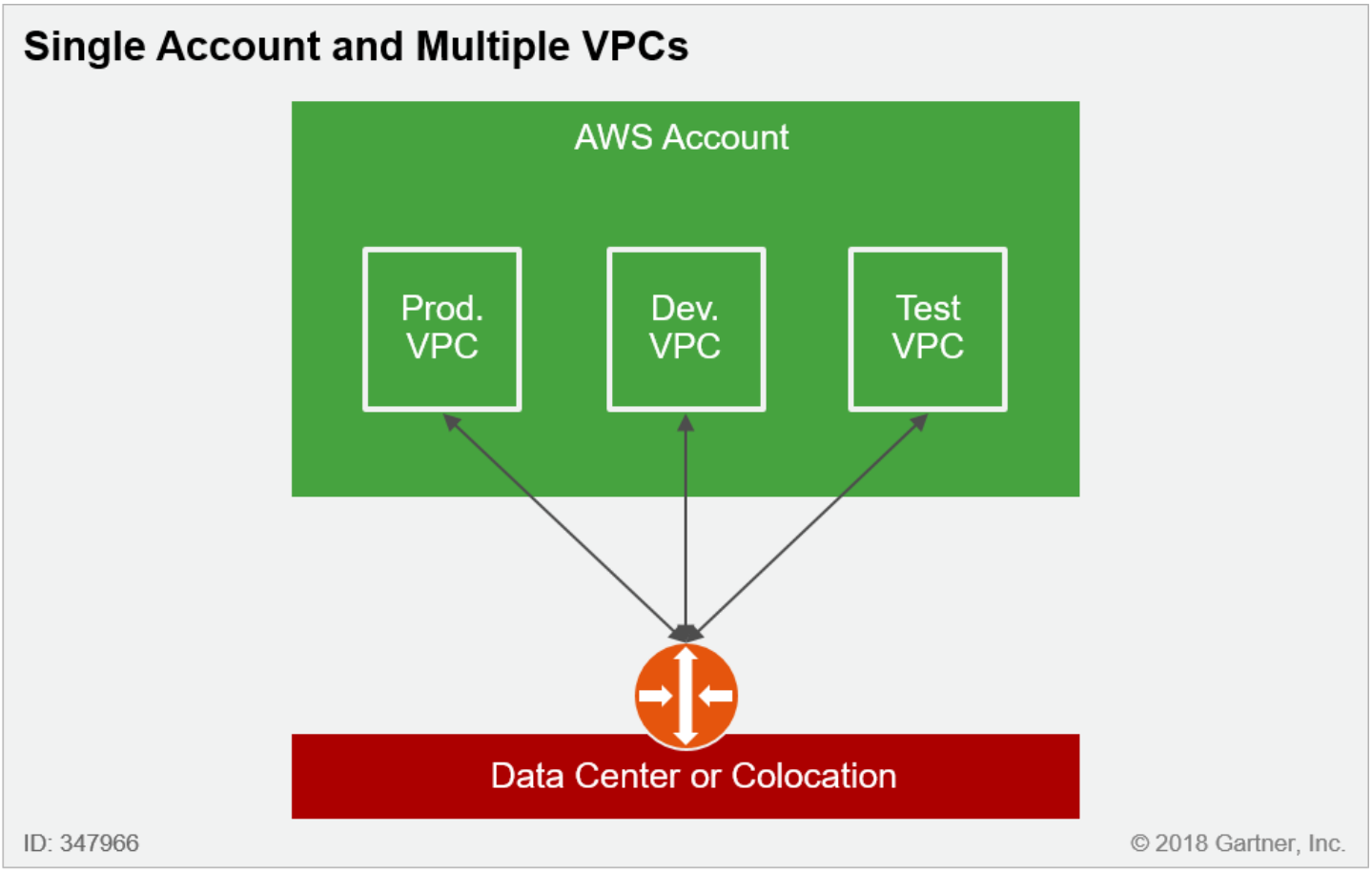
The consolidated IT operating style has the following disadvantages:

- Hard limits on VPCs and accounts, creating a higher chance of reaching those limits
- Complex IAM required to support role-based access control
- Test/dev and production that run off the same account, resulting in huge blast radius when things go wrong
- Cannot purchase different level of support for production and test/dev
- Budgeting and forecasting that requires coordination across multiple teams
- Isolation that is limited to VPCs and subnets

Consolidated Plus

Even within the same IT operational style, several designs can be implemented to meet different objectives. For example, consider the single account and multiple VPCs design proposed in Figure 6. It is similar to the consolidated option, except it offers a higher degree of network isolation and security by partitioning environments that use VPCs instead of subnets. Doing so also allows cloud architects to bypass potential VPC limits.

Figure 6. Single Account and Multiple VPCs



Source: Gartner (May 2018)

Every design has pros and cons. In this case, the consolidated-plus option improves security and isolation at the expense of increased network complexity. As shown in Figure 6, when connecting an on-premises data center, cloud architects must connect each VPC independently, delivering the highest level of isolation and security. AWS does offer VPC peering among VPCs, but given the design calls for production, development and test environments, these environments should not communicate directly. Alternatively, this design may be modified by adding a VPC for shared services that can then be used to peer with other VPCs and offer access back to the right resources on-premises. Introducing a shared services VPC allows the cloud architect to implement a design that leverages centralization and a high degree of security and isolation while simplifying networking.

Option 2: Divided

The divided IT style is the opposite of the consolidated style described in Option 1. It is designed for business units that share the same set of customers but offer different products and services to those customers. Figure 7 shows the overall design, which is essentially a collection of autonomous businesses serving the same group of customers. Each line of business is responsible for its own applications from development through production. There is a strong separation between the business units so that, for instance, development within one business unit does not affect production in another business unit.

Figure 7 illustrates a design that calls for three enterprise accounts for each LOB and three general accounts for each LOB. The enterprise accounts are:

- **Billing:** This account is used for consolidated billing and does not have any other resources or capabilities. Billing for each LOB rolls up to that business unit and is then consolidated to a corporate billing account solely for volume discount purposes. This structure allows for detailed billing within the business unit, where the overall responsibility for those charges resides. It also offers some value-added capabilities:
  - Centralized AWS account creation function
  - Combined usage that streamlines budget tracking
  - The ability to negotiate an enterprise contract and take advantage of volume discounts
  - Analytics that are leveraged over all accounts
- **Shared services:** This account is used to consolidate and simplify shared services and operations among other AWS accounts and VPCs. It offers the following value-added capabilities:
  - It provides simplified networking management between AWS and the on-premises data center. It is also used to simplify networking with other VPCs via VPC peering.
  - Technical professionals can house commonly used platform operation services such as directory services, Domain Name System (DNS), Network Time Protocol (NTP) and proxy.
  - This account will also house all cloud operation functions.

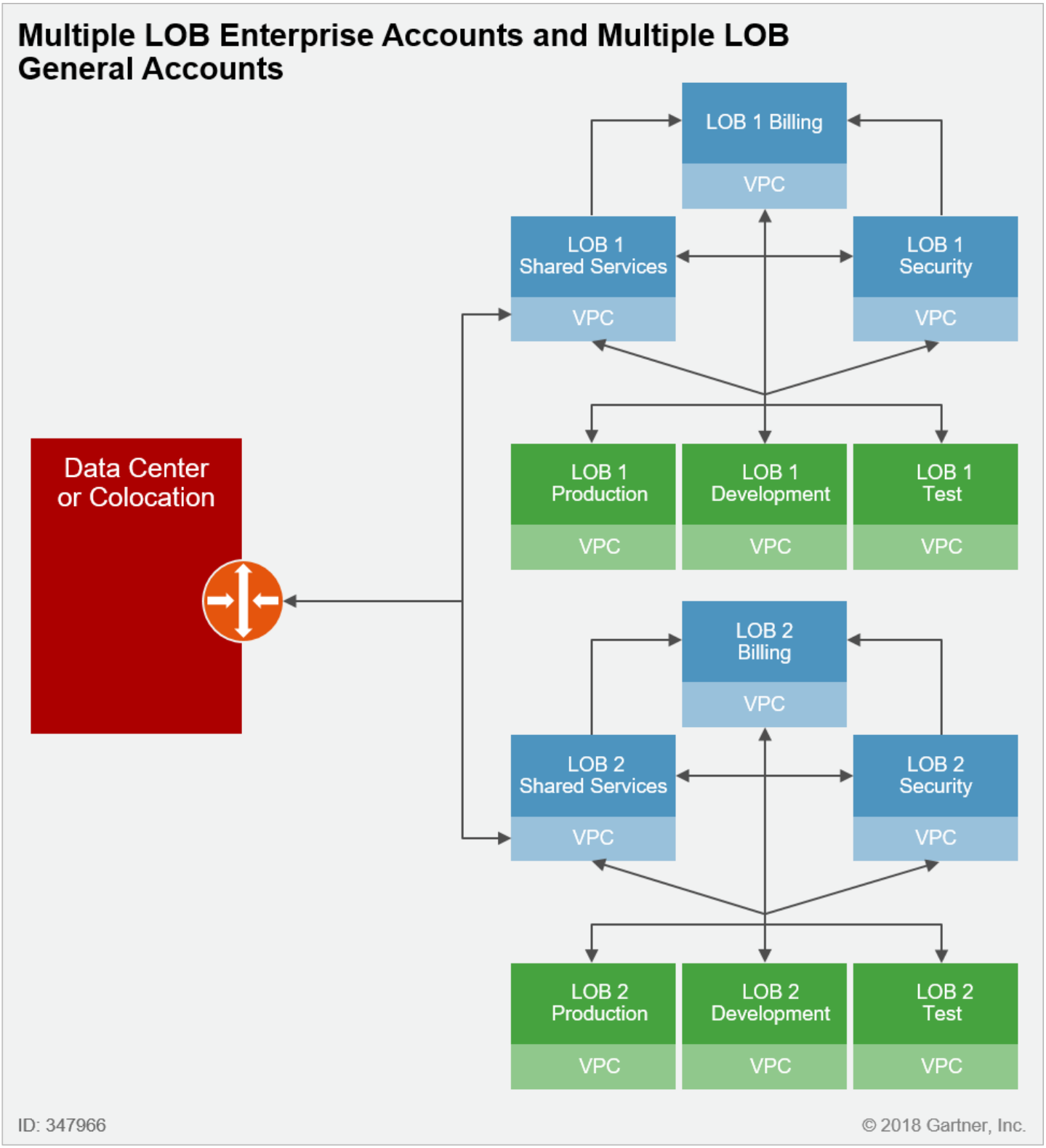
  This design style calls for a single VPC within the shared services account.
- **Security:** This account is managed by the security team and is responsible for developing security best practices while ensuring that security governance is standardized across all the AWS accounts. It offers the following value-added capabilities:
  - Consolidated CloudTrail log
  - Consolidated application logs
  - Security incident management
  - Security audits
  - Use of cross-account access to centralize security access between AWS accounts

Production, development and test, as well as other functions, are separated within the business unit by VPC boundaries, allowing each department within the business to spin up resources as needed. Lines for security federation, whether they are IAM or Lightweight Directory Access Protocol/Active Directory (LDAP/AD), also roll up within the business unit. Shared services are available only within the respective business unit through VPC peering.

The divided option is ideal for:

- Large, multinational companies where business units operate totally autonomously and may be widely geographically dispersed.
- Companies that have acquired other companies but not yet integrated them into the parent company.
- Large holding companies where the lines of business are substantially different from each other.

Figure 7. Multiple LOB Enterprise Accounts and Multiple LOB General Accounts



Source: Gartner (May 2018)

Pros

This divided IT operating style has several favorable design characteristics:

- Strong security separation

- Delegated access by LOB
- Separation of environments and applications, reducing blast radius
- Scalability, achieved by adding AWS accounts and/or VPCs
- Ability to use detailed billing reports to gain a granular view for each LOB
- Reduced risk of hitting AWS service limits

Cons

The divided IT operating style has the following disadvantages:

- Network isolation that is only based on VPC boundaries
- Network complexity (routing, VPC peering and connectivity back to on-premises)
- No standardization across LOBs
- Every LOB is responsible for managing its own budget and forecast
- More difficulty in developing an aggregate financial view

Option 3: Associated

The associated style offers technical professionals the ability to standardize IT processes by having a single production account, separated into subnets for easier management and security, for several LOB applications. The associated style (see Figure 8) uses a consolidated billing model by connecting all AWS accounts to a centralized billing account that takes advantage of volume discounts and better visibility of AWS spend.

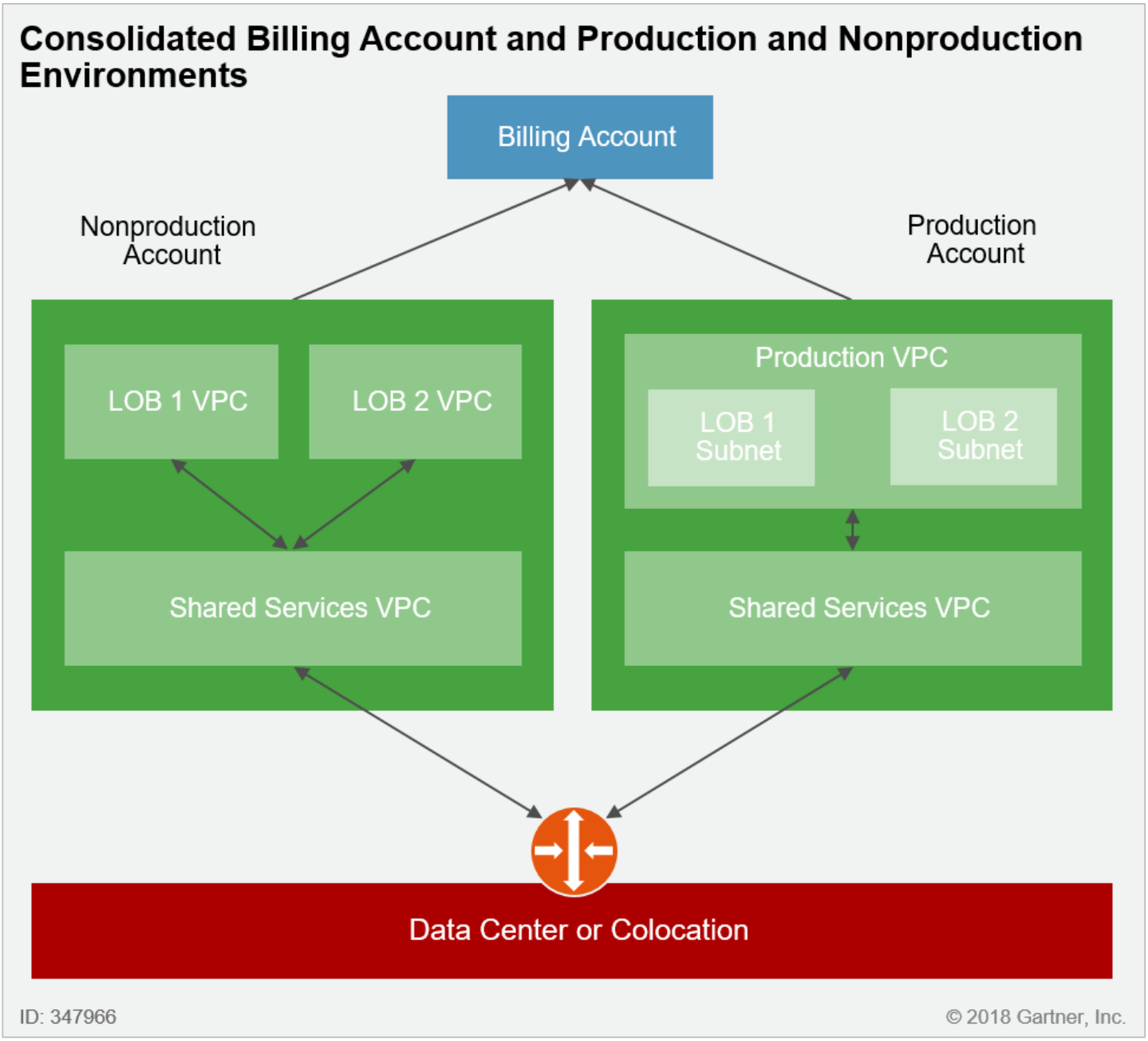
The associated style also benefits from a shared services model, which allows common services to be deployed, configured and managed centrally while allowing other resources in different VPCs to peer and take advantage of these services. Instead of connecting an on-premises environment into every VPC in the AWS account, which would lead to a networking management nightmare, this approach allows on-premises connectivity into the shared services VPC for distribution to other VPCs via VPC peering. For more information, see "Best Practices for Amazon VPC and Azure VNet." (https://www.gartner.com/document/code/337223?ref=grbody&refval=3875212)

Additionally, the associated style allows technical professionals to give LOB application owners freedom to develop and test their applications in the nonproduction environment within their own VPCs. This makes the development and test processes easier and more isolated. However, as they transition into production, they adopt IT processes, fall under the production account and get partitioned into subnets.

The associated option is ideal for:

- Small or midsize organizations that prioritize IT process and business integration
- Organizations that are content with subnet isolation and that want to adopt on-premises best practices
- Simplified and restricted network connectivity and management that connects back to the on-premises environment

Figure 8. Consolidated Billing Account and Production and Nonproduction Environments



Source: Gartner (May 2018)

Pros

This associated IT operating style has several favorable design characteristics:

- Easy separation of environment by production, development and test
- Ability to control connectivity to on-premises resources using existing security tools
- User and network access separation by account
- Standardized production environment
- Need to tag resources for cost allocation

Cons

The associated IT operating style has the following disadvantages:

- Cost increase as a result of VPC peering
- Budgeting and forecasting that requires coordination between multiple teams
- Network complexity (routing, VPC peering and connectivity back to on-premises resources)
- Federated access into multiple AWS accounts required

Option 4: Repeatable

The repeatable model is ideal for business units that have similar IT needs, but prefer to operate autonomously. The goal here is to standardize as much of the IT infrastructure and applications across the company and to leverage IT development and knowledge equally among all business units. Individual business units have their own AWS accounts and VPC, with no data or infrastructure shared among business units. The central IT planning group drives standardization down to all of the business units and communicates helpful ideas, developed within individual business units, to all business units.

The AWS setup for this type of model features three enterprise accounts (see Figure 9). The enterprise accounts are:

- **Billing:** This account is used for consolidated billing and does not have any other resources or capabilities. Billing for each LOB rolls up to that business unit and is then consolidated to a corporate billing account solely for volume discount purposes. This structure allows for detailed billing within the business unit, where the overall responsibility for those charges resides. It also offers some value-added capabilities:
  - Centralized AWS account creation function
  - Combined usage that streamlines budget tracking

- The ability to negotiate an enterprise contract and take advantage of volume discounts
- Analytics that are leveraged over all accounts

- **Shared services:** This account is used to consolidate and simplify shared services and operations among other AWS accounts and VPCs. It offers the following value-added capabilities:
  - It provides simplified networking management between AWS and the on-premises data center. It is also used to simplify networking with other VPCs via VPC peering.
  - Technical professionals can house commonly used platform operation services such as directory services, DNS, NTP and proxy.
  - This account will also house all cloud operation functions.

  This design style calls for a single VPC within the shared services account.

- **Security:** This account is managed by the security team and is responsible for developing security best practices while ensuring that security governance is standardized across all the AWS accounts. It offers the following value-added capabilities:
  - Consolidated CloudTrail log
  - Consolidated application logs
  - Security incident management
  - Security audits
  - Use of cross-account access to centralize security access between AWS accounts

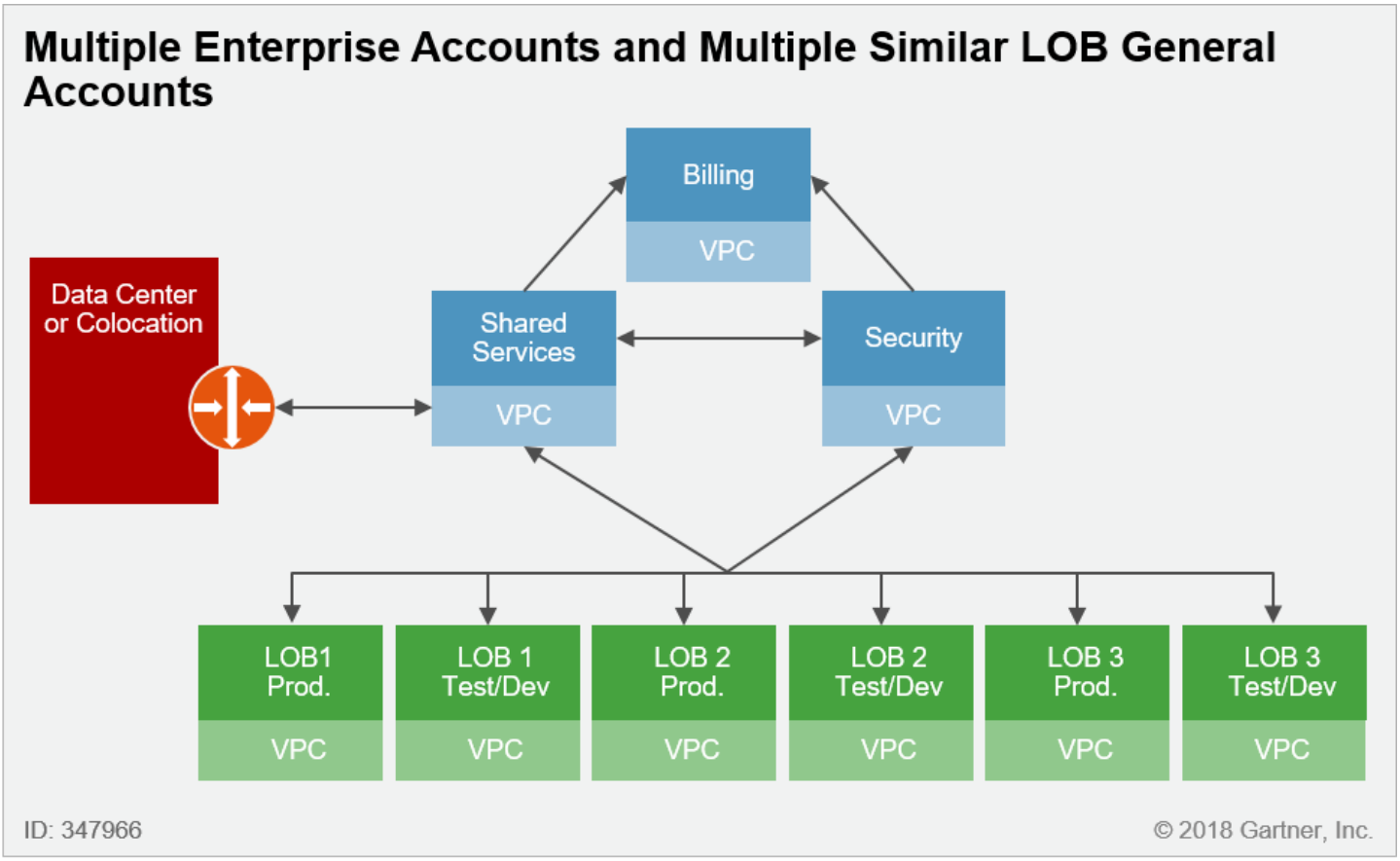
Enterprise accounts are connected to the general accounts within the business units through VPCs. Security federation is through either LDAP/AD or native IAM and is completely independent from the business units. The various application teams all work through role-based permissions and can share services through VPC peering. Business units work autonomously and only share knowledge through the central IT organization.

The repeatable option is ideal for:

- Very large companies seeking economic and operational efficiency across all of the business units
- National, state and provincial governments

- Large banks, financial companies and medical/pharmaceutical companies

Figure 9. Multiple Enterprise Accounts and Multiple Similar LOB General Accounts



Source: Gartner (May 2018)

Pros

This repeatable IT operating style has several favorable design characteristics:

- Separation of environments and applications, thus limiting the blast radius
- Delegated access and VPC configuration to different teams across LOBs
- Scalability by adding AWS accounts and VPCs
- Standardized templates and configuration management that can be leveraged across LOBs
- Separation of production and nonproduction spend by cost center
- Financial accountability by LOB via discrete AWS accounts
- Centralized financial view and centralized volume discounts for cost optimization through consolidated billing

Cons

The repeatable IT operating style has the following disadvantages:

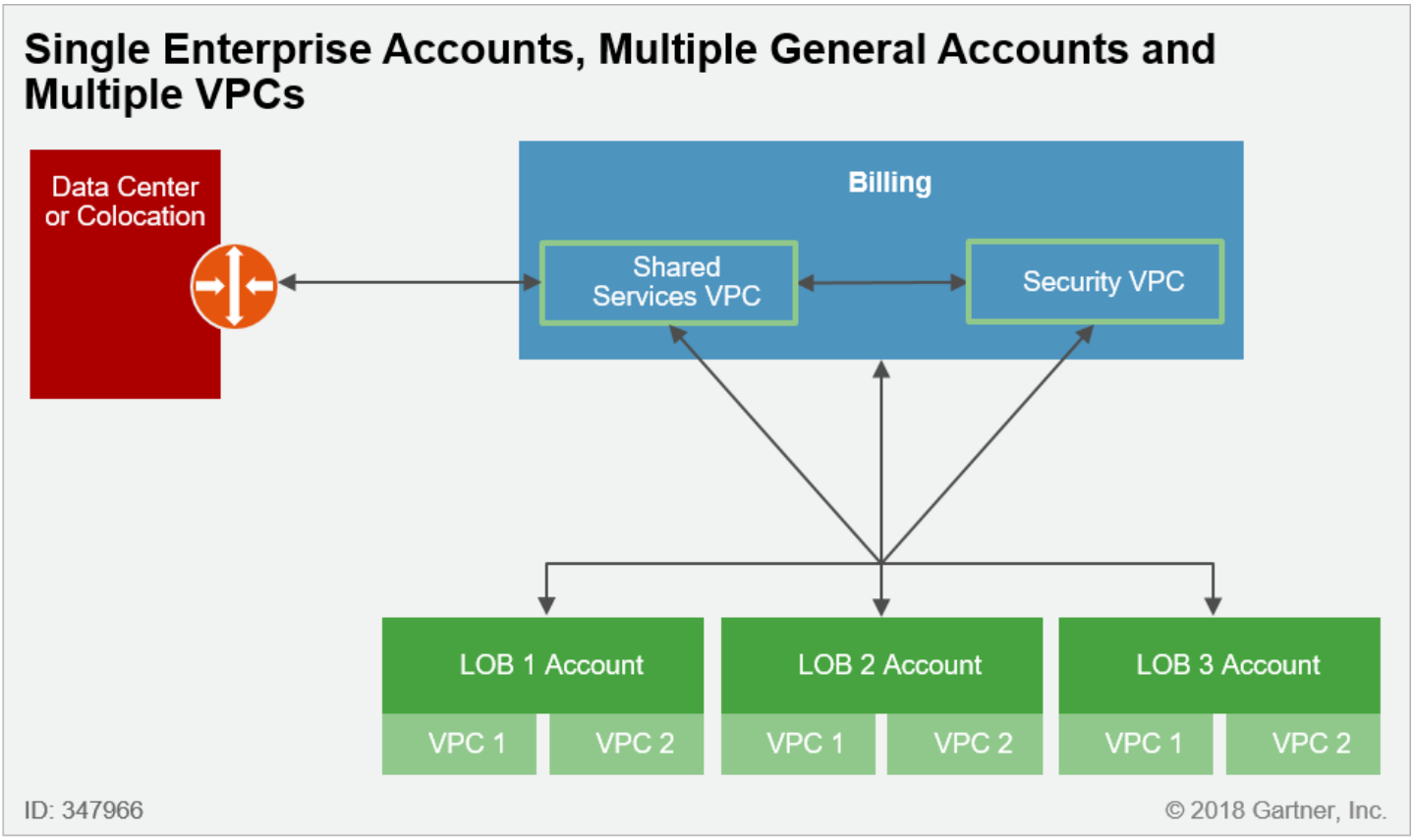
- Separate network routing for each LOB (a complex management task)
- Increased complexity with network configuration

Repeatable Plus

Technical professionals who are looking to simplify the repeatable style approach might be interested in this design. Repeatable plus has the same pros and cons as repeatable, but it presents a design that is based on an enterprise account that combines billing, shared services and security, as opposed to three distinct accounts (see Figure 10).

Instead of having a billing account that is doing nothing except billing, the account is expanded to encompass shared services and a VPC. If security is a primary concern, add security as a VPC and give complete control of this account to the security team. The more restrictive the account, the better. And because all general accounts tie into these three enterprise accounts, it makes sense to have one account that is restrictive.

Figure 10. Single Enterprise Accounts, Multiple General Accounts and Multiple VPCs

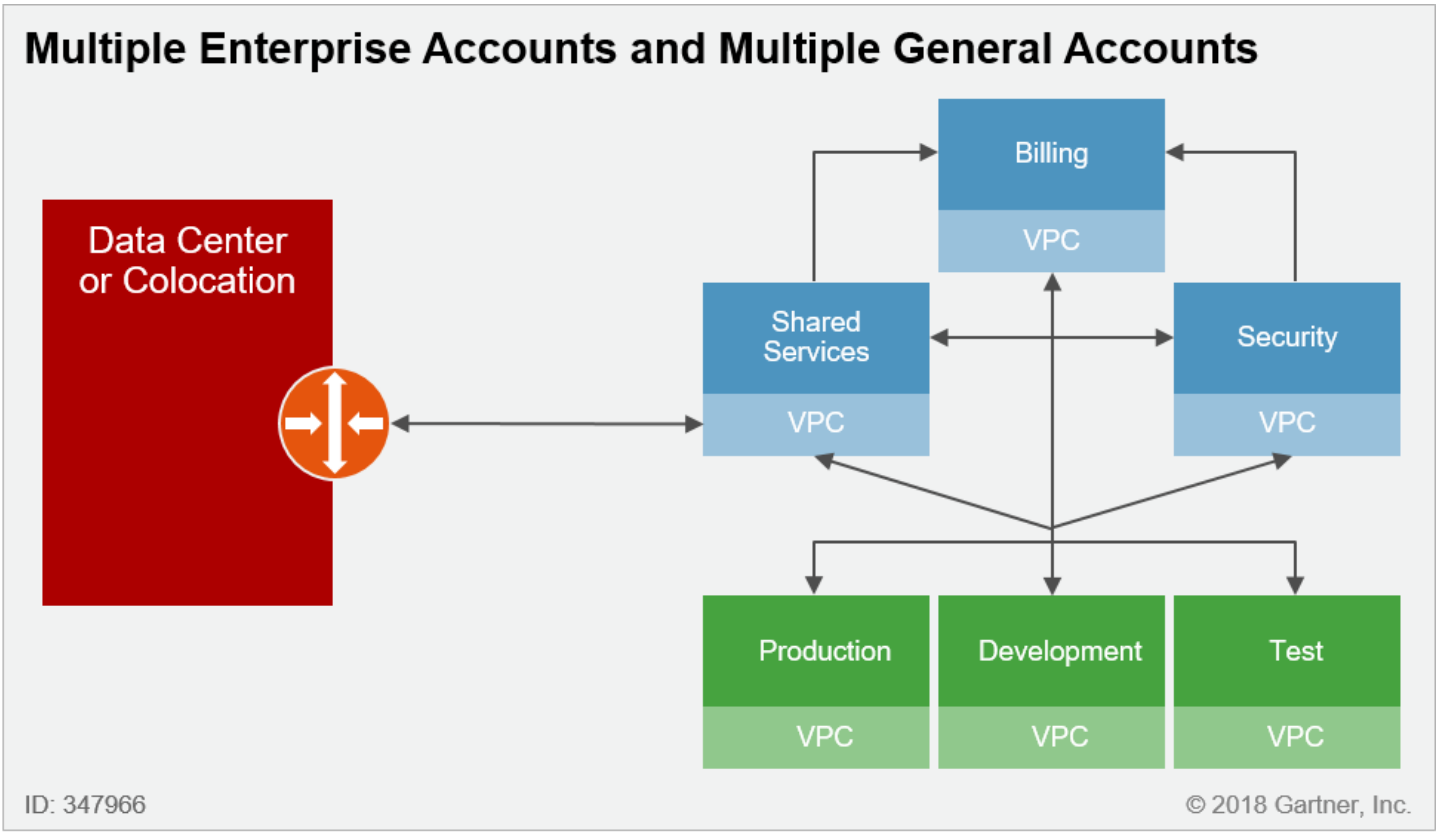


Source: Gartner (May 2018)

Option 5: Gartner Commonly Observed

This fifth and final style is a bonus option that is the culmination of many Gartner client interactions. While advising clients on the best style to adopt for their AWS account governance, we have commonly come across the model depicted in Figure 11.

Figure 11. Multiple Enterprise Accounts and Multiple General Accounts



Source: Gartner (May 2018)

It is a "hybrid" model that consolidates the best of all the options in a scalable, secure and standardized method. The implementation suggests three enterprise accounts:

- **Billing:** This account is used for consolidated billing and does not have any other resources or capabilities. It also offers some value-added capabilities:
  - Centralized AWS account creation function
  - Combined usage that streamlines budget tracking
  - The ability to negotiate an enterprise contract and take advantage of volume discounts
  - Analytics that are leveraged over all accounts
- **Shared services:** This account is used to consolidate and simplify shared services and operations among other AWS accounts and VPCs. It offers the following value-added capabilities:

- It provides simplified networking management between AWS and the on-premises data center. It is also used to simplify networking with other VPCs via VPC peering.
- Technical professionals can house commonly used platform operation services such as directory services, DNS, NTP and proxy.
- This account will also house all cloud operation functions.

This design style calls for a single VPC within the shared services account.

- **Security:** This account is managed by the security team and is responsible for developing security best practices while ensuring that security governance is standardized across all the AWS accounts. It offers the following value-added capabilities:
  - Consolidated CloudTrail log
  - Consolidated application logs
  - Security incident management
  - Security audits
  - Use of cross-account access to centralize security access between AWS accounts

In addition to these systemwide accounts, there is a need for three additional account types:

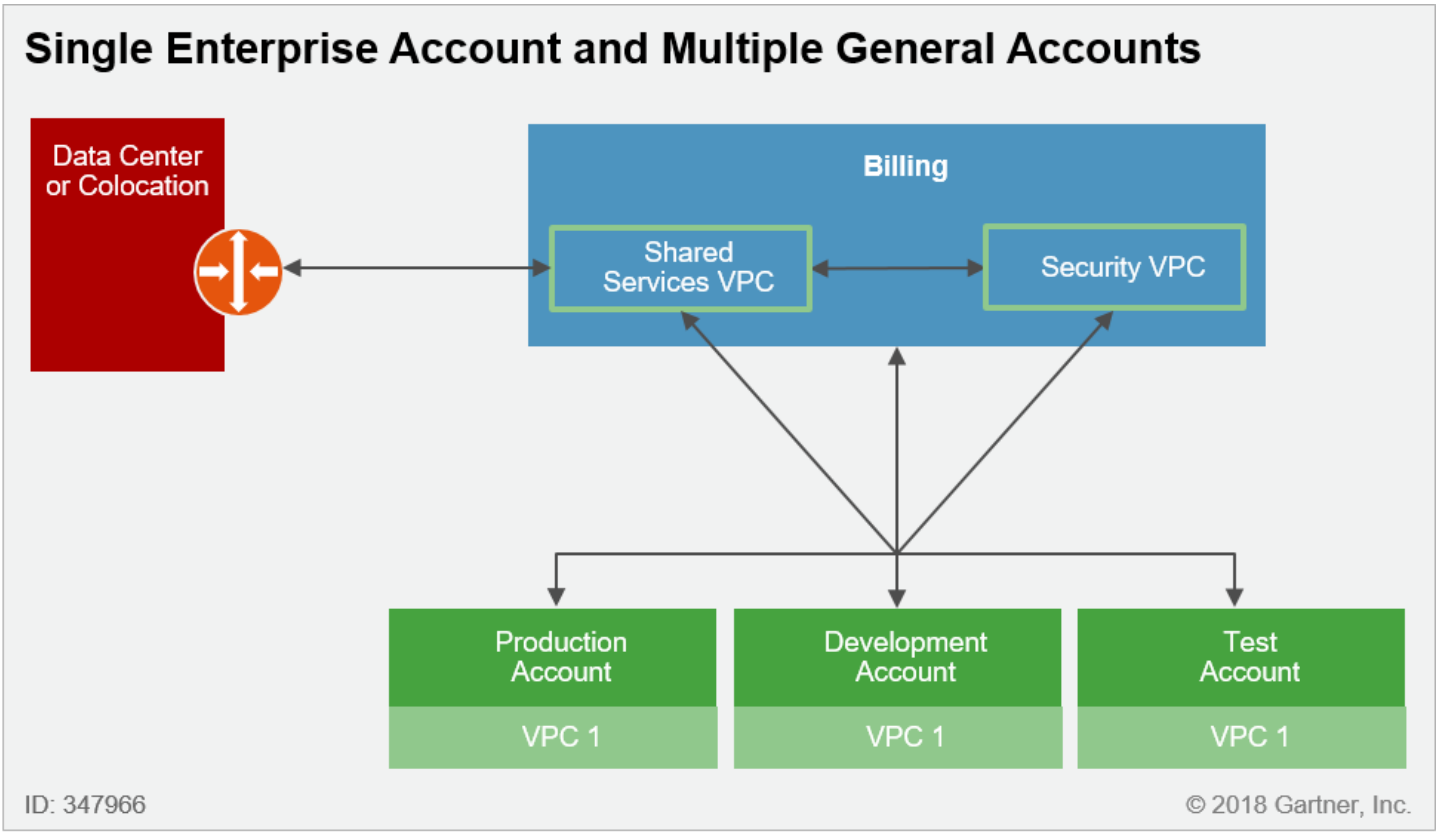
- **Production:** The production account may have one or more VPCs, depending on the need. Technical professionals may opt to have a single VPC and divide applications by subnet. Alternatively, they may choose to have one VPC per LOB or even per app. The common thread here is that this is the production account. It links back to billing for consolidated billing, links to shared services as needed, and receives its security posture from the security account.
- **Development:** As the name implies, this is a development account and can be partitioned in several different ways. It connects to the enterprise shared services, billing and security as needed.
- **Test:** The test account is similar to the development account and is an account that is used for testing, sandboxing and so on.

Gartner Commonly Observed Plus

Technical professionals who are looking to further simplify the AWS account governance process might be interested in the design shown in Figure 12. Gartner clients have often suggested even more simplification of the environment. As a result, this is a simplified version that combines the three enterprise accounts into one.

Instead of having a billing account that is doing nothing except billing, this option adds shared services as a VPC. If security is a primary concern, add security as a VPC and give complete control of this account to the security team. The more restrictive the account, the better. And because all general accounts tie into these three enterprise accounts, it makes sense to have one account that is restrictive.

Figure 12. Single Enterprise Account and Multiple General Accounts



Source: Gartner (May 2018)

Framework Step No. 3: Automating AWS Account and VPC Life Cycles

A wide range of tools, available from AWS and other companies, can be used to speed the process of creating and managing AWS accounts and VPCs. Many common tasks can be automated using AWS Organizations either through the AWS Organizations API, AWS CloudFormation or scripts. AWS also provides the AWS SDKs that consist of libraries and sample code in various programming languages that can be used to shortcut automation tasks. Here are some of the ways to automate with AWS Organizations:

- AWS Organizations API and CloudFormation Templates
- CloudFormation StackSets
- AWS Config
- Scripting

AWS Organizations API and CloudFormation Templates

The Organizations API and CloudFormation templates can be used to easily set up and configure every account just the way you need it without having to use scripts. Together, these can automate the entire process for such tasks as:

- Creating IAM users
- Setting up roles and policies
- Setting up logging
- Setting up networking
- Creating and configuring VPCs

The basis of CloudFormation is a simple text file that acts as a template to help speed the creation of standardized attributes assigned to a single account or group of accounts. These templates are useful not only in standardizing account attributes, but also in troubleshooting and rolling back changes, if necessary. CloudFormation templates can be created from scratch, or you can use prebuilt templates supplied by AWS.

CloudFormation StackSets

Although CloudFormation templates are a great way to automate the AWS account governance life cycle, it is limited to a single AWS account and to a single region. CloudFormation StackSets extend the capabilities of CloudFormation templates to multiple accounts and multiple regions. This is crucial for the automation process of multiple AWS accounts and multiple VPCs because it allows technical professionals to centrally deploy a CloudFormation template consistently across multiple accounts and regions. When using CloudFormation StackSets, you must understand the different components involved in the architecture as follows:

- **Administrator account:** This account is used to deploy the CloudFormation StackSet. It is an account that has all the right permissions to deploy the StackSet to target accounts in different regions. Technical professionals use a CloudFormation template to deploy the StackSet into the administrator account, which then deploys it into target accounts. This same process is used for management purposes of the stacks.
- **Target account:** This account is the target location to which stacks will be deployed. Target accounts delegate trust to the administrator account in order to accept the deployment and maintenance of a stack.

A good example of how CloudFormation StackSets can now be used to maintain consistency and compliance across multiple accounts is to enable CloudTrail and apply AWS Config rules. Another

example of how CloudFormation StackSets can be used is to automate the creation and configuration of a VPC. StackSets can easily configure a virtual private network (VPN) between on-premises resources and AWS, configuring Classless Inter-Domain Routing (CIDR) blocks, subnets, route tables and so on.

AWS Config

AWS Config enables users to monitor AWS resource configurations and automatically evaluate them against the desired configurations. When used with AWS Organizations, AWS Config allows customers to protect against "configuration drift" across all accounts in their organizations.

Scripting

Third-party tools, such as Bash CLI, Python, PowerShell or Terraform, can be used to automate account creation, automate setup or move accounts to a new OU. They can also be used to call CloudFormation templates to simplify the provisioning of services or application stacks. Scripts can be downloaded from GitHub (https://github.com/awslabs) or written from scratch.

Using Terraform to Create a VPC

Terraform is a continuous configuration automation tool that abstracts out the interaction with services such as AWS. It is similar to CloudFormation, and AWS supports both. For more information see "Assessing Terraform for Provisioning Cloud Infrastructure." (https://www.gartner.com/document/code/328206?ref=grbody&refval=3875212)

Instructions for downloading an example of a Terraform script can be found in the Appendix section of this research.

What Cannot Be Automated?

AWS provides a number of valuable tools to automate account management. However, some things cannot be automated because AWS has not yet created a way to automate them, or because it is too difficult or risky to automate. Table 1 shows the tasks that can and cannot be automated using AWS Organizations.

Table 1: What Can and Cannot Be Automated

Can Be Automated ↓	Cannot Be Automated ↓

Can Be Automated ↓	Cannot Be Automated ↓
■ End-to-end account creation using the APIs ■ Identity and Access Management ■ Consolidated billing ■ Usage reporting ■ Corporate security and compliance ■ Policy-based management for groups of users ■ Health Insurance Portability and Accountability Act (HIPAA) compliance	■ Multifactor authentication (MFA) on an AWS account ■ Removing a programmatically created account from your OU ■ Moving AWS accounts between OUs

Source: Gartner (May 2018)

Most of the more common tasks can be automated, such as account creation, assigning attributes to accounts, billing, reporting, security and compliance.

However, there are three tasks that cannot be automated:

- **MFA on an AWS account:** Assigning MFA to a user account is a complex process that must be done manually.
- **Removing a programmatically created account from your OU:** When removing an account from your OU, the account must have all of the information needed to make it a stand-alone account. Information such as support plan, payment plan and contact information must be populated manually.
- **Moving AWS accounts between OUs**: You must first remove the account from the current OU and make it a stand-alone account before it can be invited to join another OU.

Framework Step No. 4: Securing AWS Accounts According to Best Practices

Security is one of the primary concerns for newcomers to AWS. Gartner best practices for securing your AWS accounts are:

- **Enable MFA for the root account:** MFA provides an extra level of protection beyond a simple username and password. When enabled, users will be prompted for their AWS username and password followed by a request to enter an authentication code from another approved device, such as a smartphone. Gartner recommends always enabling MFA for the root account because it controls all other accounts. A hacked root account can be disastrous for the corporation. MFA should also be applied to any other account that has significant privileges.

- **Enable CloudTrail:** CloudTrail provides operational auditing, risk auditing, governance and compliance for your AWS account. Essentially, it logs every change in the account and assigns a time stamp for that change. Workflows can be defined and set to execute automatically when security vulnerabilities are detected, allowing the security violation to be stopped instantly and the appropriate people to be notified.
- **Don't manage resources with the master account:** Only manage organizationwide shared resources in the master account (like a company directory). Use role-based access control to ensure that users who need to manage those resources assume a role with permissions defined specifically so that they can accomplish management of that resource, without having full administrative access to the master account. Once someone else has access to the master, it is impossible to restrict that person's access.
- **Control new account creation:** AWS accounts can be created easily. Therefore, it is important to define when a new account should be created and when one should not be created. You should understand upfront why you are creating the account and how it will be used. Once an account is created, it is difficult — if not impossible — to merge it with another account or split it into two accounts.
- **Use OUs to assign controls:** Rather than assigning policy-based controls to individual accounts, create an OU and move the account into the OU. By having all accounts reside in OUs, you can avoid potential errors associated with manually assigning controls to individual accounts.
- **Take advantage of consolidated billing:** Consolidated billing is a key benefit of AWS Organizations and is automatically enabled upon creating an organization. However, in situations where AWS Organizations will not be used, enable consolidated billing across all AWS accounts to leverage volume discounting and centralized billing. Use AWS billing tools to track overall cost and cost per account, as well as to detect improper use of an account.
- **Don't assign controls to the root account:** Controls assigned to the root account affect every account in the entire organization. Accidentally assigning the wrong controls could cause a serious problem. Therefore, Gartner recommends assigning controls further down the hierarchy, where a mistake may not be as disastrous.
- **Only assign necessary permissions:** Carefully plan out who needs what privileges and only grant those privileges necessary to get the work done. You should avoid granting all privileges to anyone. Once a privilege is assigned, in most cases, it can and will be used.
- **Assess on-premises enterprise roles and map to IAM by using federation:** Establishing a federated relationship between your corporate identity service and AWS is possible, if your organization is using an identity service that is compatible with SAML 2.0 (such as Active Directory). Alternatively, technical professionals can also use a custom proxy server that translates user identities from the corporate identity service to IAM roles.

- **Use AWS Single Sign-On (AWS SSO):** AWS Single Sign-On allows users to sign in using their existing corporate credentials and to access all of their assigned AWS accounts and business applications from one place. AWS SSO integrates with AWS Directory Service and is managed using AWS Organizations.
- **Make use of cross-account roles:** In many situation, AWS users assume different roles, such as development and production. Instead of logging out of one account and logging into another, AWS allows users in one account to seamlessly share resources in another account. This can be accomplished by creating IAM roles with different permissions and then allowing users to switch roles and gain different levels of access in different accounts without having to create and assign permissions in each account.

Appendix

In the Downloads menu associated with this research, you can download a .zip file that contains a set of sample Terraform configuration scripts that can be customized and used.

The .zip file contains:

- Org — Directory with Terraform that creates an organization and elects the current account as the master account for that organization.
- Acct — Directory with Terraform that creates three additional accounts within the organization created above.
- Vpc — Directory with Terraform that creates a VPC, six subnets, and network access translation NAT/VPN gateways in each of the three accounts created using the "acct."

You must manually insert the account IDs from the accounts that have been created, as well as the cross-account role name.

Recommended by the Authors

Implementing a Governance Framework for Microsoft Azure Services (https://www.gartner.com/document/3838868?ref=ddrec&refval=3875212)

Assessing the Strengths and Weaknesses of High-Value IaaS and PaaS Multicloud Use Cases (https://www.gartner.com/document/3846474?ref=ddrec&refval=3875212)

In-Depth Assessment of Amazon Web Services IaaS, March 2018 (https://www.gartner.com/document/3867872?ref=ddrec&refval=3875212)

Implementing an Identity Strategy for Amazon Web Services (https://www.gartner.com/document/3620417?ref=ddrec&refval=3875212)

Best Practices for Amazon VPC and Azure VNet (https://www.gartner.com/document/3830363?ref=ddrec&refval=3875212)

Assessing Terraform for Provisioning Cloud Infrastructure (https://www.gartner.com/document/3823221?ref=ddrec&refval=3875212)

2018 Planning Guide for Cloud Computing (https://www.gartner.com/document/3810365?ref=ddrec&refval=3875212)

Assessing Cloud Security Monitoring and Compliance Capabilities in Amazon Web Services (https://www.gartner.com/document/3606021?ref=ddrec&refval=3875212)

Securing Privileged Access to Your IaaS Environment (https://www.gartner.com/document/3777664?ref=ddrec&refval=3875212)

Using DevOps Tools for Infrastructure Automation (https://www.gartner.com/document/3745722?ref=ddrec&refval=3875212)

Defining and Implementing Effective Cloud Security Architecture in Amazon Web Services (https://www.gartner.com/document/3454732?ref=ddrec&refval=3875212)

Recommended For You

Foundations of a Production-Grade Public Cloud IaaS and PaaS Architecture (https://www.gartner.com/document/3878720?ref=ddrec&refval=3875212)

Key Services Differences Between AWS, Azure and GCP: Availability and Network (https://www.gartner.com/document/3891204?ref=ddrec&refval=3875212)

How to Manage Public Cloud Costs on Amazon Web Services and Microsoft Azure (https://www.gartner.com/document/3831270?ref=ddrec&refval=3875212)