**9**

**Driving Efficiency with CloudOps**

Organizations migrating to the cloud need management and governance to ensure best practices in their cloud IT operations. Whether you manage modern applications built using microservices-based architectures, containers, serverless stacks, or legacy applications that have been re-hosted or re-architected for the cloud, you will realize that traditional application development and operations processes could be more effective.

Automation has always been vital for managing cloud operations, increasing efficiency and avoiding human error disruption. However, you will still observe many manual tasks in most organizations, especially IT workload management. With the rise of the cloud, automation has become more critical due to its pay-as-you-go model. Automation helps you improve productivity, resulting in substantial cost savings in human effort and IT resource expenditure. Automation has become key to reducing daily operational costs and cloud organizations spending more on operations than upfront capital investment.

Automation is crucial for cost and other aspects, such as enduring application security and reliability. Automation goes hand in hand with monitoring and alerts. Automation will only work if you have proper monitoring, alerting your automation script when to take a certain action, for example, if you want to run your production app server without compromising the user experience due to a capacity crunch. It’s always recommended to monitor server capacity, such as if the server exhausted memory capacity to 80% or CPU capacity to 70%, and send an alert to autoscaling for server scaling as needed.

AWS provides several services and tools to automate your cloud infrastructure and application with CloudOps fully or if you want to automate security using DevSecOps. In this chapter, you will learn the following topics in detail to understand the need for cloud automation, monitoring, and alerts:

* What is CloudOps?
* AWS CloudOps pillars
* DevOps and DevSecOps in AWS
* AWS cloud management tools for automation
* Cloud automation best practice

By the end of this chapter, you will understand various automation strategies to manage cloud operations. You will learn about various AWS services available at your disposal for cloud automation, monitoring, and alerts and how to use them in your workload.

**What is the cloud operation (CloudOps) model, and what role does automation play?**

CloudOps, or the cloud operational model, encompasses a collection of guidelines and safeguards that are established, tracked, and adjusted as needed to manage expenses, boost productivity, and mitigate potential security risks. It can help guide your people, processes, and the technology associated with your cloud infrastructure, security, and operations. An operational model also helps you develop and implement controls to manage security, budget, and compliance across your workloads in the cloud.

Implementing cloud automation empowers organizations to construct streamlined cloud operational models through the automated creation, modification, and removal of cloud resources. Although the concept of cloud computing initially promised the ability to utilize services as required, many organizations still rely on manual processes to provision resources, conduct testing, recognize resource redundancy, and decommission resources. This approach can result in substantial labor, error-proneness, and expense.

Businesses migrating to the cloud need management and governance to ensure best practices in their cloud IT operations. AWS management and governance services provide faster innovation and firm control over cost, compliance, and security.

The following are the key benefits of the cloud operation model:

* Organizations can unlock the speed and agility that comes with the cloud and accelerate their cloud adoption and application modernization efforts as part of their digital transformation journey.
* Use the power of automation for routine tasks to reduce manual errors and interventions.
* Continue to scale your businesses with the certainty that cloud governance spans all different environments uniformly and at scale.
* Use your skilled personnel effectively to deliver business outcomes.
* Avoid unexpected cost overruns.

While implementing cloud automation can initially demand significant effort, the potential rewards are considerable. Once the initial hurdles are overcome, the ability to perform intricate tasks with just a single click can transform an organization’s operations. In addition to minimizing manual labor, cloud automation offers several other advantages, including:

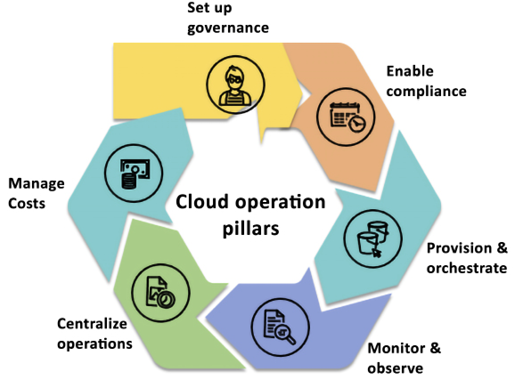
* **Improved security and resilience**: Automation helps you improve security as there are always chances of security lapses due to human error or changes that can be left out, which is outside of individual knowledge for setting up security credentials for newly added dev environment. Also, automation helps to improve resiliency by automated recovery of the environment; for example, if your server reaches over capacity, you can take proactive action by adding more CPU or memory to avoid downtime.
* **Improved backup processes**: Automated backup is one of the most important things for your business process continuation. Protecting your data helps minimize business loss by winning customer trust and rebuilding your environment quickly in case of a disaster recovery event. Automated backup helps you ensure all backups are secure and do not rely on one individual who can forget to take a backup.
* **Improved governance**: Automation helps you to improve governance by making sure all activity is captured across the environment. For example, you need to know what servers and database inventories are running across your company’s IT workload or who is accessing those environments. All these things are possible through an automated governance model.

AWS provides a set of services and third-party tools for modern enterprises as they adopt the cloud operation model. It can help you drive more innovation, faster cloud adoption, improved application performance, and quicker response times to customer feedback while maintaining governance and compliance.

Let’s look at the CloudOps pillars and the services provided to fulfill the requirements of each pillar.

**CloudOps pillars**

While planning your CloudOps model, you need to take a 360-degree look. You want to provision and operate your environment for business agility and governance control. Establishing a CloudOps model, regardless of your cloud migration journey, helps you attain consistent governance and efficient operations across different infrastructure environments. This helps free up critical resources to deliver business outcomes and time-to-market faster while improving safety, ease, efficiency, and cost control. The following diagram shows the key pillars of cloud operation for complete coverage of your IT workload automation.



*Figure 9.1: The pillars of CloudOps*

As shown in the preceding diagram, the following are the key pillars of CloudOps:

1. **Set up Governance**: Set up a well-architected, multi-account AWS environment with guardrails to build the foundation for governance. AWS environments with a Well-Architected Framework checklist ensure you have covered all best practices to monitor and set alerts on your environment for security, operational excellence, cost, reliability, and performance.
2. **Enable Compliance**: Continuously monitor compliance and configurations for your resources, remediate failures in an automated fashion, and gather evidence for audits.
3. **Provision & Orchestrate**: Speed up application and resource provisioning with **infrastructure-as-code** (**IaC**) while maintaining consistency and compliance.
4. **Monitor & Observe**: Measure and manage your applications and resources to identify and resolve issues quickly.
5. **Centralize Operations**: Take seamless and automated operational actions across your entire application portfolio while maintaining safety, security, and compliance.
6. **Manage Costs**: Helps manage costs with transparency, control, regular forecasting, and optimization, thus enabling businesses to achieve greater financial efficiency and transform their operations.

Let’s look at what each of these pillars helps you to achieve in your efforts to enable governance in cloud environments. You will also learn the AWS services that predominantly fulfill the requirement of each pillar.

**First pillar – Set up governance**

The first and the best place to start is by laying a very strong foundation for your governance. In the AWS environment, it begins by setting up a well-architected, multi-account AWS environment and setting up guardrails in each account. You learned about the Well-Architected Framework in *Chapter 2*, *Understanding the AWS Well-Architected Framework and Getting Certified*, where you saw that AWS has a comprehensive checklist to make sure your environment is set up properly to monitor cost, security, performance, reliability, and high availability. You can refer to AWS’s well-architected labs here at <https://www.wellarchitectedlabs.com/>, which provide a very comprehensive, practical, hands-on guide to enable those guardrails against each well-architected pillar. The environment you build must be secure and extensible so that you don’t halt experimentation and innovation as you grow your footprint on AWS. You need it to scale with your usage.

Your business needs to evolve continuously, so you should keep yourself from a single mode of architecting and operating in AWS. Most customers’ environments don’t remain static. They tend to grow with their business. You want to ensure your landing zone grows with your business without encumbrance while adhering to organizational policies. AWS Landing Zone is an offering that assists clients in swiftly configuring a secure and multi-account AWS environment, built around AWS’s industry-leading practices. The solution delivers a preconfigured and secure infrastructure that encompasses key services, standardized AWS account architecture, and robust security controls. The goal of Landing Zone is to provide a secure, well-architected multi-account environment that serves as a starting point for new AWS accounts. It helps customers get started with AWS faster by providing a set of reusable blueprints for common patterns and practices, such as account VPCs, security controls, and identity and access management.

You need a well-defined AWS environment to accommodate the following needs:

* **Many Teams**: Multiple teams could be in the same account, overstepping one another.
* **Isolation**: Each team could have different security needs and want to isolate themselves from one another with a different security profile.
* **Security Controls**: Different applications might have different controls around them to address security and compliance. For example, talking to an auditor is far easier than pointing to a single account hosting the PCI solution. But even within an organization, security controls provide the ability to isolate certain things based on security isolation needs.
* **Business Process**: There are completely different **business units** (**BUs**) or products. For example, the Sales BU is different from the HR BU with an entirely different business process.
* **Billing**: An account is the primary way to divide items at a billing level. Each AWS account is billed separately and has its own set of resources and associated charges. This means that if you have multiple accounts within an organization, each account will have its own billing and cost allocation data.

To enable account control, AWS provides **AWS Organizations** that help you to establish a multi-account structure for centralized governance. You can use it to establish granular control over your AWS accounts, manage across accounts easily, and apply policies as broadly or as narrowly as you need. In the previous chapter, *Chapter 8*, *Best Practices for Application Security, Identity, and Compliance*, you learned about AWS Organizations.

For automated setup, AWS provides **AWS** **Control Tower**, a self-service solution to set up and govern a secure, compliant multi-account AWS environment. It abstracts multiple AWS services under the covers, so you can use it to set up your environment, based on best practices, without needing a lot of AWS knowledge. AWS Control Tower provides the following benefits:

* Automate the setup of your landing zone based on best-practice blueprints
* Apply guardrails for ongoing governance over your AWS workloads
* Automate your account provisioning workflow with an account factory
* Get dashboard visibility into your organizational units, accounts, and guardrails

You learned about AWS Control Tower in the previous chapter, *Chapter 8*, *Best Practices for Application Security, Identity, and Compliance*.

AWS professional services provide AWS **Landing Zone Accelerator** (**LZA**), which is a set of tools and resources provided by AWS to help customers accelerate the deployment of a secure, multi-account AWS environment.

LZA builds on top of the AWS Landing Zone service, which provides a pre-built, opinionated framework for setting up a secure, multi-account environment.

LZA provides a modular set of landing zone components that can be customized to meet specific requirements and leverages automation to speed up the deployment process. It also provides access to AWS experts who can provide guidance and best practices to help ensure a successful deployment. You can learn more about LZA by referring AWS user guide here: <https://aws.amazon.com/solutions/implementations/landing-zone-accelerator-on-aws/>.

LZA is designed to accelerate the process of setting up a landing zone for workloads that are migrating to AWS. LZA provides a modular set of landing zone components that can be customized to meet specific requirements and leverages automation to speed up the deployment process. This makes LZA a good fit for customers who need to set up a landing zone quickly and want more control over the individual components of their environment.

On the other hand, Control Tower is designed to help customers set up and govern a multi-account AWS environment. Control Tower provides a pre-built set of rules and policies to enforce governance and security best practices across multiple AWS accounts. CT also provides a central dashboard for managing and monitoring multiple accounts, making it easier for customers to maintain governance and compliance across their environment. This makes CT a good fit for customers who need to manage multiple AWS accounts and want a pre-built set of governance policies to enforce best practices.

It is extremely important to have the correct foundation when you are starting with your cloud journey. AWS services like Landing Zone and Control Tower, in combination with the Well-Architected Framework, help you with an automated way to establish the right environment. The teams building this out are not the bottleneck and enable you to be flexible in your approach and know that you might need new accounts, processes, or isolation solutions. That flexibility is what allows us to succeed in the long term. After setting up governance, you must ensure your applications meet compliance requirements. Let’s learn more about automating compliance.

**Second pillar – Managing Configuration, Compliance, and Audit**

As you migrate workloads to the cloud, you need to know that you can maintain cloud compliance and get assurance for your workloads. Once compliance mechanisms and processes are in place, you can empower your development teams to build and innovate while having peace of mind that they are staying compliant.

A resource inventory helps you maintain environment configurations, track change history, depend on one another, and ensure your resources are correctly configured. Once you establish proper configurations, you want to be able to audit, manage, and remediate them quickly.

How many hours do you spend today collecting evidence in response to an audit? Whether internal or external? Wouldn’t it be easier if you were able to keep a running log of all auditable events and remediation actions? And that’s where AWS configuration, compliance, and auditing tools come in. You must continuously monitor configuration and compliance changes within your AWS environment and keep a running audit log to get visibility into your organization’s resource configurations.

Many customers follow the **Institute of Internal Auditors (IIA**) guidance for the three lines of defense:

* **1st Line**: *How to automate compliance management and manage risk* – The implementation of AWS CloudTrail, AWS Config, AWS Control Tower, and AWS License Manager can aid in the automation of compliance management and risk mitigation within an AWS environment.
* **2nd Line**: *How to implement continuous oversight and oversee risk* – By utilizing Amazon CloudWatch and AWS Security Hub, it is possible to understand the operational health and security status of AWS accounts.
* **3rd Line**: *How to assess and independently gather assurance of risk management* – AWS Audit Manager helps assess their security, change management, and software licensing controls.

You learned about AWS Security Hub and Control Tower in the previous chapter, *Chapter 8*, *Best Practices for Application Security, Identity, and Compliance*. Let’s learn about the other services, depending on mentioned above for managing cloud audits and compliance. You can use a combination of these services your IT workload needs, to automate configuration, compliance, and audit.

**AWS CloudTrail**

In AWS, all interactions with AWS services and resources are handled through AWS API calls, and these API calls are monitored and logged by AWS CloudTrail. AWS CloudTrail records all API calls made in your AWS account and provides a complete history of all user activity and API usage. This information can be used for security analysis, compliance auditing, and troubleshooting.

CloudTrail stores all generated log files in an Amazon S3 bucket that you define. These log files are encrypted using **Amazon S3 server-side encryption** (**SSE**), which provides an additional layer of security for your logs. It’s also worth noting that CloudTrail logs all API calls, regardless of whether they come directly from a user or on behalf of a user by an AWS service.

This lets you understand all API activity in your AWS environment, which can be crucial for security and compliance purposes.

AWS CloudTrail is a service that enables governance, compliance, operations, and risk auditing for AWS accounts by logging and monitoring account activity related to actions across the AWS infrastructure. By using CloudTrail, you can continuously monitor and retain logs of all account activity, providing you with a complete history of your AWS account’s event history. This event history includes actions taken through the AWS Management Console, AWS SDKs, command-line tools, and other AWS services.

These logs can be used to aid in governance, compliance, and risk management, providing a clear record of activity across your AWS infrastructure. With CloudTrail, you can also create custom alerts and notifications to help you identify and respond to potential security issues and compliance risks in real time.

Once enabled, CloudTrail will automatically track all Management Events at no charge. Then, you also have several different data event sources you can opt into depending on your application and compliance needs. This event history is another source of observability data, simplifying security analysis, resource change tracking, and troubleshooting. You can learn more about AWS CloudTrail by visiting the AWS page here: <https://aws.amazon.com/cloudtrail/>.

**AWS Config**

You learned about AWS Config in *Chapter 5*, *Storage in AWS – Choosing the Right Tool for the Job*, in the context of S3. AWS Config records and evaluates your AWS resources configuration. AWS Config performs the following activities for AWS resources: record, evaluate, and visualize. Let’s learn about these in more detail in the context of CloudOps:

**Record**

* **Configuration history of AWS resources**: AWS Config records the details of changes made to your AWS resources, providing you with a configuration history timeline. This enables you to track any changes made to a resource’s configuration at any time in the past.
* **Resource relationship tracking**: AWS Config can discover, map, and track relationships between AWS resources in your account. For example, if a new **Amazon Elastic Compute Cloud** (**Amazon EC2**) security group is associated with an Amazon EC2 instance, AWS Config will record the updated configurations of both the Amazon EC2 security group and the Amazon EC2 instance.
* **Configuration history of software**: AWS Config can also record software configuration changes within your Amazon EC2 instances and servers running on-premises or with other cloud providers. It provides a history of both OS and system-level configuration changes and infrastructure configuration changes recorded for Amazon EC2 instances.

**Evaluate**

* **Configurable and customizable rules**: Assess your resource configurations and resource changes for compliance against built-in or custom rules and automate the remediation of non-compliant resources. You can customize pre-built rules provided by AWS Config or create your own custom rules with AWS Lambda to define your internal guidelines and best practices for resource configurations.
* **Conformance packs**: Simplifies organization-wide deployment and reporting of compliance. It deploys a pack of config rules and remediation actions to your AWS Organization.
* **Automatic remediation** enables you to remediate non-compliant resources using Systems Manager Automation documents.

**Visualize**

* **Cloud governance dashboard**: This feature provides a visual dashboard that lets you easily identify non-compliant resources and take the necessary corrective action. You can customize the dashboard to monitor resources based on cost and security.
* **Multi-account, multi-region data aggregation**: AWS Config allows you to aggregate data from multiple AWS accounts and regions, providing you with a centralized view of your resources and their compliance status with AWS Config rules. This feature is particularly useful for enterprise-scale organizations.
* **Configuration snapshots**: AWS Config can take snapshots of your resource configurations at specific points in time. This allows you to quickly identify changes to your resources and compare their configurations across different points in time.

Here is an example AWS Config rule that checks whether Amazon EC2 instances have an associated security group with inbound rules that allow traffic on port 22 (SSH):

{

"Name": "ec2-security-group-has-inbound-rules-on-port-22",

"Description": "Checks whether the security group associated with an EC2 instance has inbound rules that allow traffic on port 22",

"Scope": {

"ComplianceResourceTypes": [

"AWS::EC2::Instance"

]

},

"Source": {

"Owner": "AWS",

"SourceIdentifier": "EC2\_INSTANCE\_HAS\_SECURITY\_GROUP\_WITH\_INBOUND\_RULES\_ON\_PORT\_22"

},

"InputParameters": "{\"allowedProtocols\":\"tcp\",\"portNumber\":22}"

}

This rule checks whether the security group associated with each Amazon EC2 instance has inbound rules that allow traffic on port 22. If any instances do not have such a security group, they will be flagged as non-compliant.

AWS Config help to keep AWS resources compliant. You can learn more about AWS Config by visiting the AWS page here: <https://aws.amazon.com/config/>. Let’s look at the next service for tracking and auditing licenses.

**AWS License Manager**

AWS License Manager is a one-stop solution for managing licenses from various software vendors across hybrid environments. This helps you to stay compliant within your organizational structure and processes. There are no additional charges for using AWS License Manager.

AWS License Manager is targeted at IT administrators who manage licenses and software assets. This includes license administrators, procurement administrators, or asset managers who are responsible for managing license true-ups and vendor audits. In contrast, users spin up instances and use the licensed software on those instances. With AWS License Manager, the administrators can now easily manage licenses. Users in the organization are not required to do additional work to manage licenses and can focus on business as usual.

You can complete your licensing true-ups and audits using AWS License Manager. Administrators start by creating rules based on their enterprise agreements. They can do this using the AWS Management Console, CLI, or API. Furthermore, administrators can enforce licensing rules by attaching them to instance launches. Once rules are enforced, the service automatically keeps track of instances as users spin them up and down.

The organization stays compliant based on its license terms, and administrators can discover users’ software after spinning up instances. Finally, administrators can keep track of usage through the AWS License Manager’s built-in dashboard.

AWS License Manager automatically keeps track of instance launches, and the built-in dashboard is populated. Administrators can view usage limit alerts and take actions such as procuring more licenses as needed. When there is an upcoming license true-up or audit, administrators no longer have to determine which instances use which licenses. With AWS License Manager’s built-in dashboard, figuring out how many licenses they are using and which resources are using them is no longer a challenge. You can learn more about AWS License Manager by visiting the AWS page here: <https://aws.amazon.com/license-manager/>.

**Amazon CloudWatch**

CloudWatch is one of the essential services for running your cloud operation. It allows you to monitor your AWS workload and take action based on alerts. In addition, CloudWatch provides observability for your AWS resources on a single platform across applications and infrastructure.

Amazon CloudWatch is a powerful monitoring service that is designed to help you optimize your AWS resources and applications. It offers a wide range of capabilities, including:

* **Data and operational insights**: CloudWatch provides valuable insights into the performance and health of your AWS resources and applications. With CloudWatch, you can collect and track metrics, monitor log files, and set alarms.
* **Resource monitoring**: CloudWatch can monitor a variety of AWS resources, including Amazon EC2 instances, Amazon S3 buckets, and Amazon RDS instances. This allows you to quickly identify and troubleshoot any issues that arise.
* **Custom metrics**: CloudWatch allows you to create custom metrics based on the data generated by your applications. This provides you with greater flexibility and control over the monitoring process.
* **Log monitoring**: CloudWatch can also monitor the log files generated by your applications. This enables you to quickly identify and troubleshoot any issues that are related to your application code.

Amazon CloudWatch is an essential tool for anyone running applications on the AWS cloud platform. Its powerful monitoring capabilities can help you optimize your resources, improve application performance, and maintain the operational health of your systems.

**CloudWatch alarms** are a powerful feature that enable you to receive notifications or automate actions based on the rules you define. With CloudWatch alarms, you can monitor a wide range of metrics and set up alerts that notify you when certain conditions are met. For example, you can send an email alert to the admin whenever the average network latency of an Amazon RDS database exceeds 10 seconds or when the CPU usage of an Amazon EC2 instance falls below 10%. You can also create more complex alarms that automatically trigger actions, such as launching additional instances to handle increased traffic or scaling down resources during periods of low demand.

CloudWatch alarms provide a flexible and customizable way to monitor your AWS resources and take automated actions based on your specific needs. Whether you need to monitor resource utilization, application performance, or other key metrics, CloudWatch alarms can help you stay on top of your cloud infrastructure and ensure that it is always running at peak performance.

In addition, CloudWatch provides data for the past two weeks so that you can access historical data for analysis of past events. It also integrates with other AWS services, such as Amazon EC2 Auto Scaling, Amazon SNS, and AWS Lambda, enabling you to use CloudWatch to react to changes in your resources and applications.

Some key features of CloudWatch include the following:

* **Metrics**: CloudWatch allows you to collect metrics for your resources and applications, such as CPU usage, network traffic, and the disk reads/writes. You can view these metrics in the CloudWatch console or use the CloudWatch API to retrieve them programmatically.
* **Alarms**: You can set alarms in CloudWatch to be notified when certain thresholds are breached. For example, you can schedule an alarm to send an email or SMS message to you if the CPU usage on one of your Amazon EC2 instances exceeds a certain threshold.
* **Logs**: CloudWatch allows storing and accessing your log files in a centralized location. You can use CloudWatch Logs Insights to search and analyze your log data or use CloudWatch Logs to export your log data to third-party tools for further analysis.
* **Dashboards**: You can use CloudWatch dashboards to create custom views of your metrics and log data to quickly get an overview of your system’s health and performance.

Amazon CloudWatch is a powerful and flexible monitoring service that can help you ensure the availability, performance, and efficiency of your AWS resources and applications.

**Amazon CloudWatch Events** is a service that allows you to respond to changes in state in your AWS resources in real time. With CloudWatch Events, you can monitor for operational changes in your resources and automatically trigger actions based on those changes. For example, you can create a CloudWatch Events rule that triggers an AWS Lambda function whenever a new Amazon EC2 instance is launched in your account. This Lambda function can perform actions like tagging the instance, configuring its security groups, or starting a set of preconfigured applications on the instance.

CloudWatch Events allows you to react to operational changes in your AWS resources quickly and efficiently. Automating your response to these events can save time and reduce the risk of manual errors. With CloudWatch Events, you can easily create rules that define the conditions that trigger your actions and specify the targets for executing those actions.

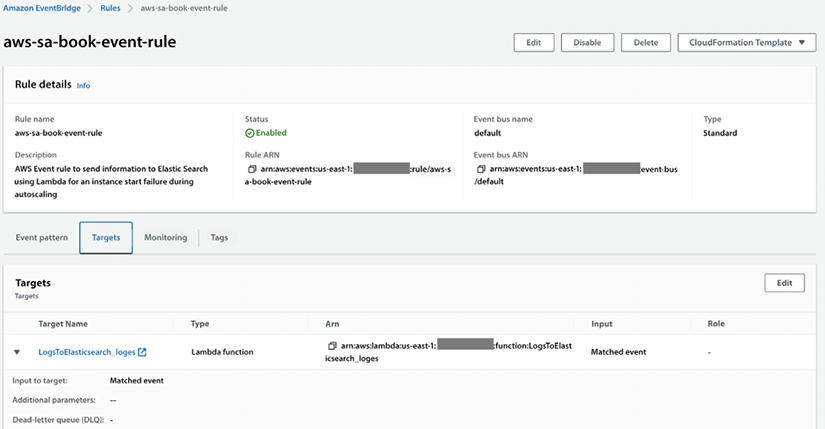
**CloudWatch Events rules** enable you to match event patterns and take actions in response to those patterns. A rule can have one or more event patterns, and you can specify the type of action that CloudWatch Events takes when it detects a pattern. For example, you can set up a rule to send an email message when a new Amazon EC2 instance is launched or to stop an Amazon EC2 instance when the CPU utilization is too high.

CloudWatch Events can be used to schedule automated actions that self-trigger at a specific time or when a specified event occurs. For example, you can use CloudWatch Events to schedule the automatic stopping of Amazon EC2 instances so that you don’t incur charges for no longer needed instances. You can learn more about AWS CloudWatch by visiting the AWS page here: <https://aws.amazon.com/cloudwatch/>.

**Amazon EventBridge**

Amazon EventBridge is a fully-managed, serverless event bus service that simplifies connecting your applications, integrated SaaS applications, and AWS services. By creating event-driven architectures, EventBridge allows various applications and services to communicate with each other in a flexible and scalable manner.

The following screenshot shows an EventBridge rule set up to send information to Elasticsearch using Lambda for an instance start failure during autoscaling. This Lambda function then sends information to Elasticsearch, allowing you to analyze and troubleshoot the failure.



*Figure 9.2: An AWS EventBridge rule for an autoscaling failure*

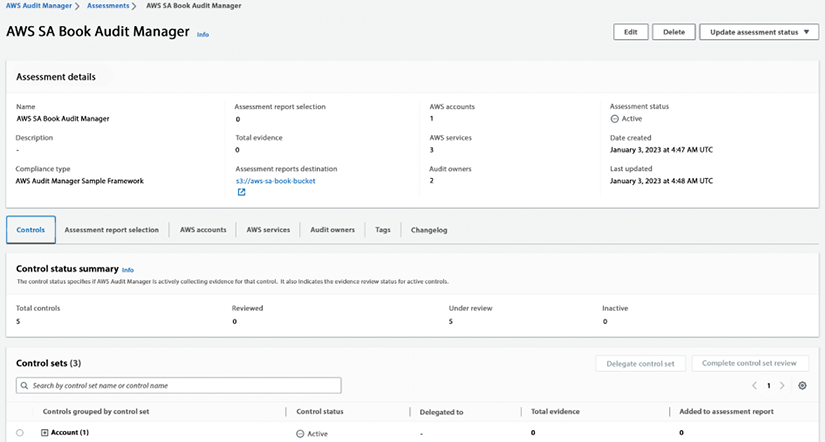
EventBridge simplifies the setup and management of rules that dictate event routing and processing, enabling you to create complex and resilient architectures for your applications. With EventBridge, you can easily build event-driven applications and receive notifications about events from AWS DevOps services and automation application deployment pipelines. You can explore more about EventBridge by visiting the AWS page here: <https://aws.amazon.com/eventbridge/>.

**AWS Audit Manager**

The audit Manager continuously accesses risk and compliance controls and provides the following benefits.

* Easily map your AWS usage to controls: Use pre-built compliance frameworks or build custom frameworks to collect evidence for compliance controls.
* Save time with an automated collection of evidence across accounts: Focus on reviewing the relevant evidence to ensure your controls are working as intended.
* Be continually prepared to produce audit-ready reports: Evidence is continuously collected and organized by control so you can effortlessly search, filter, and review evidence to ensure controls are working as intended.
* Ensure assessment report and evidence integrity: When it is time for an audit, build assessment reports with evidence that has been continuously collected, securely stored, and remains unaltered.

Below you can find a screenshot of AWS Audit Manager:



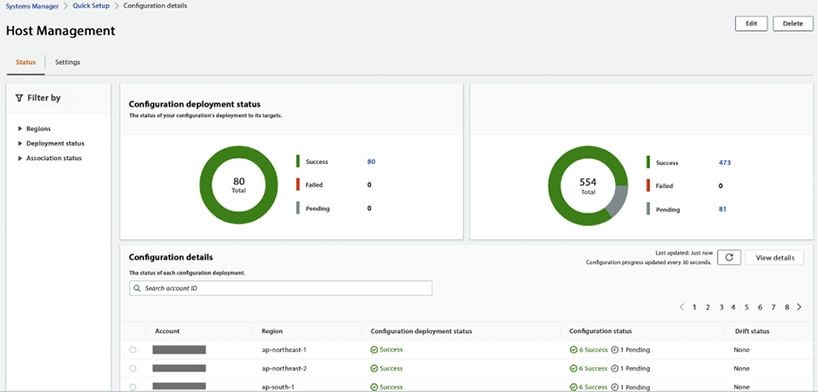
*Figure 9.3: AWS Audit Manager*

You can learn more about AWS Audit Manager by visiting the AWS page here: <https://aws.amazon.com/audit-manager/>.

**AWS Systems Manager**

AWS Systems Manager is a management service that allows you to take actions on your AWS resources as necessary. It provides you with a quick view of operational data for groups of resources, making it easy to detect any issues that could impact applications that rely on those resources. You can group resources by various criteria, such as applications, application layers, or production vs. development environments. Systems Manager displays operational data for your resource groups on a single dashboard, eliminating the need to switch between different AWS consoles. For example, you can create a resource group for an application that uses Amazon EC2, Amazon S3, and Amazon RDS. Systems Manager can check for software changes installed on your Amazon EC2 instances, changes in your S3 objects, or stopped database instances.

The following screenshot shows a configuration deployment status for servers in a given AWS account:



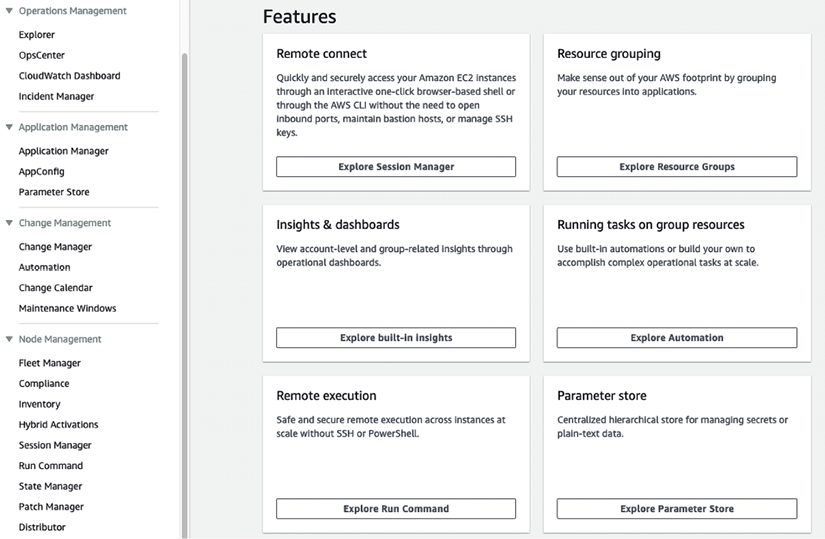
*Figure 9.4: AWS Systems Manager host management*

AWS Systems Manager provides detailed insights into the current state of your resource groups, allowing you to understand and control them quickly. The Systems Manager Explorer and Inventory dashboards offer various tools to view system configurations, such as operating system patch levels, software installations, and application configurations. Moreover, it is integrated with AWS Config, allowing you to track changes across your resources over time.

AWS Systems Manager offers several features to help maintain security and compliance in your environment. It can scan your instances against your patch, configuration, and custom policies, helping you identify and address potential security issues. With Systems Manager, you can define patch baselines, ensure that your anti-virus definitions are up-to-date, and enforce firewall policies, among other things. Systems Manager also enables you to manage your servers at scale remotely without manually logging in to each server. This feature can be especially helpful in large-scale environments, where managing resources individually can be time-consuming and error-prone.

In addition, Systems Manager provides a centralized store for managing your configuration data, including plain text items such as database strings and secrets like passwords. By separating your secrets and configuration data from your code, you can help reduce the risk of security breaches and simplify your development and deployment processes.

You can use System Manager to achieve all components of the second pillar as it provides a one-stop shop, as shown in the left-hand navigation bar in the below screenshot:



*Figure 9.5: AWS Systems Manager for CloudOps monitoring and audit*

AWS Systems Manager is the hub of your operation for managing all your AWS applications and resources along with your on-premise environments, keeping everything in one place for easy monitoring and auditing. AWS has done a great job explaining how Systems Manager works, which you can learn about by referring to this link: <https://docs.aws.amazon.com/systems-manager/latest/userguide/what-is-systems-manager.html>.

In this section, you learned about managing cloud configuration and compliance along with various AWS services that can help to achieve that. Now, let’s learn about the following step: provision and orchestration.

**Third pillar – Provisioning & orchestration**

Once you have set up the environment, you need to speed up the provisioning and orchestration of your applications and any associated resources in a repeatable and immutable fashion.

Let’s look at history; in the old days, infrastructure deployment started with manual deployments, where you used wikis and playbooks, which were sometimes outdated. Most of us can relate to a project where we had to use them. They were sometimes outdated, or some details needed to be mentioned, and we encountered situations during the provisioning which needed to be stated in the manual. The best way to solve the problem was to ask the person who did it the last few times in the past and hopes that this person was not on vacation and still working at the company!

The next step was scripting everything in Bash. It worked well until the complexity was too great because Bash was not designed to build complex deployment frameworks, so it was hard to maintain. The best advice was: it worked the last time, don’t touch it!

As digital transformation increasingly occurs within an organization, more applications either move to or are built on the cloud. These applications themselves solve complex problems and require complex infrastructure. Teams need more tools to manage this complexity, be productive, and innovate. You need highly specialized tools to manage the applications and to be able to choose from a varied set of tools based on their use cases. Managing infrastructure is a big part of managing cloud complexity, and one of the ways to manage infrastructure is by treating infrastructure as code.

**Infrastructure-as-code** (**IaC**) templates help you to model and provision resources, whether AWS-native, third-party or open source, and applications on AWS and on-premises. So, you can speed up application and resource provisioning while improving consistency and compliance. For example, a developer wants to provision an S3 bucket that meets their company’s security requirements. They will no longer have to dig through documentation to determine what bucket resource properties to set or how to set them. Instead, they can reuse a pre-built Secure S3 bucket module to provision a bucket quickly while automatically aligning with their company’s requirements.

A cloud application typically consists of many components: networking (i.e., traffic gateways), compute (Amazon EC2, containers), databases, streams, security groups, users, roles, etc. All these servers and resources are the infrastructure components of your cloud application. Managing cloud applications involves managing the life cycle of their resources: create, update, or delete. By codifying your infrastructure, you can manage your infrastructure code in a way that is similar to your application code. This means that you can use a code editor to create it, store it in a version control system, and collaborate with your team members before deploying it to production. The advantages of using IaC are numerous, including:

* A single source of truth for deploying the entire stack.
* The ability to replicate, redeploy, and repurpose your infrastructure.
* The ability to version control both your infrastructure and your application code.
* Automatic rollback to the previous working state in case of failures.
* The ability to build and test your infrastructure as part of your CI/CD pipeline.

AWS offers multiple services that help customers manage their infrastructure. AWS CloudFormation is the provisioning engine. It helps speed up cloud provisioning with IaC.

**AWS Service Catalog** allows organizations to create and maintain a list of IT services authorized to be used on the AWS platform. These lists can include everything from VM images, servers, software, and databases to complete multi-tier application architectures. Users can then browse and launch these pre-approved IT services through a self-service portal, which helps ensure that they are using approved resources that meet compliance and security requirements. Additionally, administrators can set up workflows to automatically provision resources and control access and permissions to specific resources within the catalog.

You can choose to combine these services as per your workload needs. Let’s learn about these services in more detail.

**AWS CloudFormation**

CloudFormation is a tool that supports the implementation of IaC. With CloudFormation, you can write your IaC using the CloudFormation template language, available in YAML and JSON formats. You can start from scratch or leverage any of the pre-existing sample templates to create your infrastructure. You can use the CloudFormation service through a web-based console, command-line tools, or APIs to create a stack based on your template code. Once you have defined your stack and resources in the template, CloudFormation provisions and configures them accordingly.

CloudFormation helps you model, provision, and manage AWS resources. It allows you to use a template to create and delete multiple related AWS resources in a predictable and consistent way. Here is a simple example of a CloudFormation template written in YAML syntax:

---

AWSTemplateFormatVersion: '2010-09-09'

Resources:

MyEC2Instance:

Type: AWS::EC2::Instance

Properties:

ImageId: ami-101013

InstanceType: m4.xlarge

KeyName: gen-key-pair

SecurityGroups:

- !Ref ServerSecurityGroup

ServerSecurityGroup:

Type: AWS::EC2::SecurityGroup

Properties:

GroupDescription: Allow ssh access

SecurityGroupIngress:

- IpProtocol: tcp

FromPort: '22'

ToPort: '22'

CidrIp: 10.1.1.16/0

This template creates an Amazon EC2 instance with a security group that allows incoming SSH connections. The security group is created first and then referenced when creating the Amazon EC2 instance.

To use this template, save it to a file (e.g., servertemplate.yml) and then use the AWS CLI to create a stack:

aws cloudformation create-stack --stack-name sa-book-stack --template-body file://servertemplate.yml

You can then use the AWS Management Console, the AWS CLI, or the CloudFormation API to monitor the progress of the stack creation. Once the stack is created, you will have an Amazon EC2 instance running in your AWS account. You can make changes to the stack by updating the template and using the update-stack command. You can learn more about AWS CloudFormation by visiting the AWS page here: <https://aws.amazon.com/cloudformation/>.

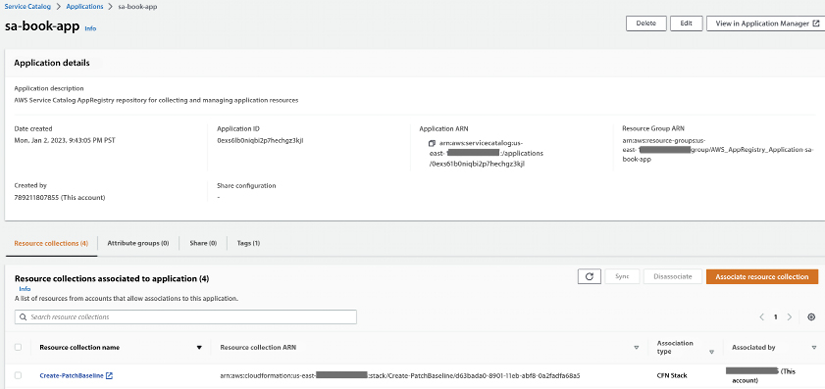
**AWS Service Catalog**

AWS Service Catalog provides a centralized platform to manage catalogs of IT services, ensuring adherence to corporate standards and compliance. Organizations can easily control IT service availability, configurations, and access permissions by individual, group, department, or cost center. The platform also simplifies the process of finding and deploying approved IT services for employees. Organizations can define their catalog of AWS services and AWS Marketplace software and make them available for their employees through a self-service portal.

**AWS Service Management Connectors** allow **IT service management** (**ITSM**) administrators to enhance the governance of provisioned AWS and third-party products.

For instance, by integrating with AWS Service Catalog, ServiceNow and Jira Service Desk can request, provision, and manage AWS and third-party services and resources for their users, streamlining the ITSM process.

**AWS Service Catalog AppRegistry** is a centralized repository that enables organizations to manage and govern their application resources on AWS. It provides a single place for collecting and managing application metadata, including their name, owner, purpose, and associated resources. This information can be used to improve application visibility and governance and to enable better collaboration between teams that work on different parts of the application stack. With AppRegistry, you can track and manage all your applications’ resources, including AWS resources, third-party software, and external resources such as domain names or IP addresses. The following screenshot shows an app registry in the system catalog:



*Figure 9.6: AWS System Catalog app registry*

With AppRegistry, you can define your application metadata, such as ownership, data sensitivity, and cost centers, and include a reference to your application within the infrastructure code. This helps business stakeholders have up-to-date information about the application’s contents and metadata. In addition, AppRegistry provides a central place to track and manage changes to your application resources, helping you maintain a comprehensive view of your applications and their dependencies. You can learn more about AWS Service Catalog by visiting the AWS page here: <https://aws.amazon.com/servicecatalog/>.

**AWS Proton**

AWS Proton is a managed application deployment service that allows developers to quickly and easily deploy and manage container and serverless applications on AWS. It provides a fully managed, opinionated environment for defining, deploying, and managing applications. With Proton, developers can focus on writing code and building applications while the service handles the underlying infrastructure and deployment workflows. This helps to accelerate the development process, reduce the risk of errors, and improve overall application quality.

AWS provides sample Proton templates that help you start building your application’s infrastructure. You can fork those samples using AWS samples code link: (<https://github.com/aws-samples/aws-proton-cloudformation-sample-templates>) and refer to them while building the Proton environment.

AWS Proton can help you update out-of-date applications with a single click when you adopt a new feature or best practice. This helps ensure that your applications remain up-to-date and compliant with industry standards. Additionally, by providing a consistent architecture across your organization, Proton helps improve collaboration and reduces the risk of errors or misconfigurations. You can learn more about AWS Proton by visiting the AWS page here: <https://aws.amazon.com/proton/>.

**AWS Cloud Development Kit (CDK)**

The AWS CDK is an open-source software development framework that enables developers to define cloud infrastructure and resources using familiar programming languages such as TypeScript, JavaScript, Python, Java, and C#. It provisions and deploys the infrastructure using AWS CloudFormation, providing the benefits of IaC.

With CDK, you will work much faster because you are using your familiar language, concepts, classes, and methods without a context switch. You also have all the tool support from the programming language, such as autocomplete, inline documentation, tests, and a debugger. The most important part is that you can build your abstractions and components of the infrastructure and application. AWS provides many default values, so there is no need to read a lot of documentation; you can start quickly.

The AWS CDK consists of three main components: the core framework, the AWS-construct library, and the CLI. The core framework enables you to define and organize your AWS infrastructure using high-level programming languages. You can create and structure apps that consist of one or multiple stacks. Stacks are the fundamental deployment unit in AWS CDK. They are a logical grouping of AWS resources that are provisioned and managed as a single unit. Each stack is mapped one-to-one to a CloudFormation stack and can be independently deployed, updated, or deleted.

It is good practice to divide resources into stacks with different life cycles: i.e., you would create one stack for network infrastructure such as a VPC, another stack would have an Elastic Container Service cluster, and yet another stack would be the application that is running in this cluster.

The AWS-constructed library in CDK is a collection of pre-built components designed by AWS for creating resources for specific services. This allows for decoupling libraries and using only the necessary dependencies in your project. The library is developed with best practices and security considerations in mind to provide an excellent developer experience, ease of use, and fast iteration cycles. The CDK CLI interacts with the core framework, helping to initialize project structure, inspect deployment differences, and deploy your project quickly to AWS. Here is an example of creating an Amazon S3 bucket using CDK in TypeScript:

import \* as cdk from 'aws-cdk-lib';

import \* as s3 from 'aws-cdk-lib/aws-s3';

class MyBookStack extends cdk.Stack {

constructor(scope: CDK.App, id: string, props?: CDK.StackProps) {

super(scope, id, props);

new s3.Bucket(this, 'BookBucket', {

bucketName: 'aws-sa-book-bucket',

publicReadAccess: true

});

}

}

const app = new cdk.App();

new MyBookStack(app, 'MyBookStack');

app.synth();

This code defines a CDK stack with a single Amazon S3 bucket and synthesizes a CloudFormation template for the stack. You can then use the cdk deploy command to deploy the stack to your AWS account.

AWS CDK provides a paradigm shift in how you provision multiple environments. With CloudFormation, you can use one template with parameters for multiple environments, i.e., dev and test. But with CDK, you have a shift where multiple templates are generated for each environment, ending in different stacks. This decoupling helps us to contain and maintain differences between environments by having less expensive resources in the dev environment. You can learn more about AWS CDK by visiting the AWS page here: <https://aws.amazon.com/cdk/>.

**AWS Amplify**

AWS Amplify is a suite of specialized tools designed to help developers quickly build feature-rich, full-stack web and mobile applications on AWS. It allows developers to utilize a wide range of AWS services as their use cases evolve. With Amplify, developers can configure a backend for their web or mobile app, visually create a web frontend UI, connect the two, and manage app content without needing to access the AWS console. At a high level, AWS Amplify provides the following features, tools, and services:

* **Amplify Libraries**: Frontend developers can use purpose-built Amplify libraries for interacting with AWS services. You can use the case-centric Amplify Libraries for connecting frontend iOS, Android, web, and React Native apps to an AWS backend and UI components for auth, data, and storage. Customers can use Amplify Libraries to build a new app backend or connect an existing backend.
* **Amplify Hosting**: Amplify Hosting is a fully managed CI/CD service for modern web apps. It offers hundreds of global points of presence for fast and reliable hosting of static and server-side rendered apps that scale with your business needs. With Amplify Hosting, you can deploy updates to your web app on every code commit to the Git repository. The app is then deployed and hosted globally using CloudFront. Amplify Hosting supports modern web frameworks such as React, Angular, Vue, Next.js, Gatsby, Hugo, Jekyll, and more.
* **Amplify Studio**: Amplify Studio is a visual development environment that provides an abstraction layer on top of the Amplify CLI. It allows you to create full-stack apps on AWS by building an app backend, creating custom UI components, and connecting a UI to the app backend with minimal coding. With Amplify Studio, you can select from dozens of popular React components, such as buttons, forms, and marketing templates, and customize them to fit your style guide. You can also import UX designs from the popular design prototyping tool, Figma, as clean React code for seamless collaboration. Amplify Studio exports all UI and infrastructure artifacts as code so you can maintain complete control over your app design and behavior.
* **The Amplify CLI**: Provides flexibility and integration with existing CI/CD tools through the new Amplify extensibility features. The Amplify CLI allows frontend developers to set up backend resources in the cloud easily. It’s designed to work with the Amplify JavaScript library and the AWS Mobile SDKs for iOS and Android. The Amplify CLI provisions and manages the mobile or web backend with guided workflows for common app use cases such as authentication, data, and storage on AWS. You can reconfigure Amplify-generated backend resources to optimize for specific use cases, leveraging the entire feature set of AWS, or modify Amplify deployment operations to comply with your enterprise DevOps guidelines.

Here’s a code example of how to use AWS Amplify in a web application to store and retrieve data from a cloud database:

import { API, graphqlOperation } from 'aws-amplify'

*// Add a new item to the cloud database*

async function addItem(item) {

const AddItemMutation = `mutation AddItem($item: ItemInput!) {

addItem(item: $item) {

id

name

description

}

}`

const result = await API.graphql(graphqlOperation(AddItemMutation, { item }))

console.log(result)

}

*// Retrieve a list of items from the cloud database*

async function listItems() {

const ListItemsQuery = `query ListItems {

listItems {

items {

id

name

description

}

}

}`

const result = await API.graphql(graphqlOperation(ListItemsQuery))

console.log(result)

}

The API object provided by Amplify enables you to call GraphQL operations to interact with the cloud database. The addItem function uses the addItem mutation to create a new item in the database, while the listItems function uses the listItems query to retrieve a list of items.

AWS Amplify ties into a broader array of tools and services provided by AWS. You can customize your Amplify toolkit and leverage AWS services’ breadth and depth to service your modern application development needs. You can choose the services that suit your application and business requirements and scale confidently on AWS. You can learn more about AWS Amplify by visiting the AWS page here: <https://aws.amazon.com/amplify/>.

In this section, you were introduced to AWS CDK, which is a provisioning and orchestration solution that facilitates the consistent and repeatable provisioning of resources. By utilizing AWS CDK, you can scale your organization’s infrastructure and applications on AWS in a sustainable manner. Additionally, AWS CDK allows you to create your infrastructure as code using programming languages. You can simplify and accelerate the governance and distribution of IaC templates using the AWS Service Catalog to create repeatable infrastructure and application patterns with best practices. Now let’s learn about the next step to set up monitoring and observations from your applications.

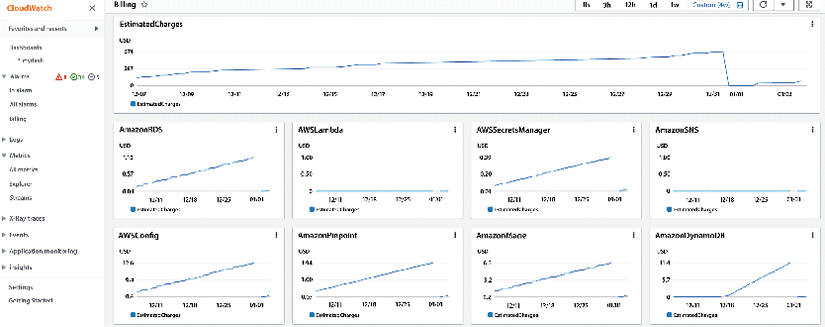
**Fourth pillar – Monitor & observe your applications**

As the saying goes, you manage what you measure, so after you’ve provisioned your application, you have to be able to start measuring its health and performance. Monitoring and observability tools help to collect metrics, logs, traces, and event data. You can quickly identify and resolve application issues for serverless, containerized, or other applications built using microservices-based architectures.

AWS provides native monitoring, logging, alarming, and dashboards with CloudWatch and tracing through X-Ray. When deployed together, they provide the three pillars of an observability solution: metrics, logs, and traces. X-Ray (tracing) is fundamental to observability and is therefore included in the motions alongside CloudWatch. Furthermore, AWS provides open-source observability for Prometheus and Grafana and support for Open Telemetry. A well-defined monitoring and observability strategy implemented with CloudWatch and X-Ray provides insights and data to monitor and respond to your application’s performance issues by providing a consolidated view of the operation.

The following are the key AWS services for observability and monitoring, which you can choose to use as per your workload needs or combine them together.

**AWS CloudWatch** helps you collect, view, and analyze metrics and set alarms to get notified when certain thresholds are breached. Here you can see a screenshot of the billing metrics dashboard in CloudWatch:

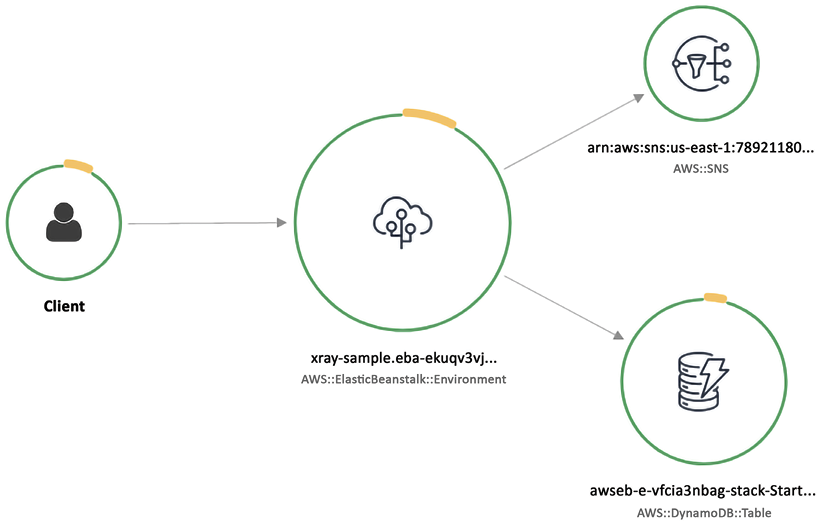


*Figure 9.7: AWS CloudWatch billing metrics dashboard*

You can also use CloudWatch to track log files from Amazon EC2 instances, Amazon RDS DB instances, and other resources and troubleshoot issues with your applications. You learned about AWS CloudWatch earlier in this chapter.

**AWS X-Ray** is a distributed tracing service that allows developers to analyze and debug their applications, especially those built using a microservices architecture. It helps identify performance bottlenecks and errors, allowing developers to optimize application performance and enhance the end-user experience.

It allows you to trace requests as they flow through your application and see the performance of each component of your application:



*Figure 9.8: AWS X-Ray service map graph*

In the AWS X-Ray console, you can view a trace of a request as it flows through your application and see the performance of each trace segment. You can also use the console to search for traces and analyze performance data for your application. To use AWS X-Ray, you first need to instrument your application code to send data to the X-Ray service. This can be done using one of the AWS SDKs or the X-Ray daemon. You can then view and analyze the data using the X-Ray console or the X-Ray API. You can learn more about AWS X-Ray by visiting the AWS page here: <https://aws.amazon.com/xray/>.

**Amazon Managed Service for Prometheus** (**AMSP**) is a fully managed service that makes it easy to run and scale Prometheus, an open-source monitoring and alerting system, in the cloud. AMSP automatically handles tasks such as scaling and maintenance, allowing you to focus on monitoring your applications. It also includes integration with other AWS services, such as Amazon CloudWatch and Amazon SNS, which allows you to view and analyze your monitoring data and set up alerts and notifications. The followings are the benefits of using AMSP:

* **Fully managed**: AMSP takes care of the underlying infrastructure and maintenance tasks, so you can focus on monitoring your applications.
* **Scalability**: AMSP automatically scales to handle changes in workload, so you don’t have to worry about capacity planning.
* **Integration with other AWS services**: AMSP integrates with other AWS services, such as CloudWatch and SNS, which allows you to view and analyze your monitoring data and set up alerts and notifications.
* **Security**: AMSP includes built-in security measures, such as encryption at rest and network isolation, to help protect your monitoring data.
* **Cost-effective**: AMSP is a pay-as-you-go service, which means you only pay for the resources you use. This can be more cost-effective than running and maintaining your own Prometheus infrastructure.

You can learn more about AMSP by visiting the AWS page here: <https://aws.amazon.com/prometheus/>.

**Amazon Managed Service for Grafana** (**AMG**) is a fully managed service that simplifies the visualization and analysis of operational data at scale. It leverages the popular open-source analytics platform Grafana to query, visualize, and alert on metrics stored across AWS, third-party ISVs, databases, and other IT resources. AMG removes the need for server provisioning, software configuration, and security and scaling concerns, enabling you to analyze your metrics, logs, and traces without the heavy lifting in production. You can learn more about AMG by visiting the AWS page here: <https://aws.amazon.com/grafana/>.

In this section, you learned that an observable environment reduces risk, increases agility, and improves customer experience. Observability provides insights and context about the environment you monitor. AWS enables you to transform from monitoring to observability so that you can have full-service visibility from metrics, logs, and traces by combining AWS CloudWatch and X-Ray. Let’s learn about the next step in building centralized operations.

**Fifth pillar – Centralized operations management**

IT teams need to take operational actions across hundreds, sometimes thousands, of applications while maintaining safety, security, and compliance simultaneously. To help make ops management as easy and efficient as possible, you must safely manage and operate your IT infrastructure at scale. To achieve that, you should have a central location and interface to view operational data from multiple AWS services. You can then automate operational tasks on applications and resources, especially common operational changes, such as rotating certificates, increasing service limits, taking backups, and resizing instances. You laid down all without compromising any safety, security, or compliance guardrails in the foundation stage.

To help enable cloud operations, you can use **AWS Systems Manager**. Systems Manager is a fully managed service that helps customers safely manage and operate their IT infrastructure at scale. It provides a central location and interfaces to view operational data from multiple AWS services. Customers can then use it to automate operational tasks on their applications and resources, especially common operational changes, such as rotating certificates, increasing service limits, taking backups, and resizing instances.

You learned about AWS Systems Manager earlier in this chapter. Here you will learn about Systems Manager’s ability to view, manage, operate, and report on cloud operations. There are **four** stages of implementing CloudOps using AWS Systems Manager:

1. **Build a foundation for cloud operations**: To build a foundation for cloud operations, the Systems Manager helps you set up the management of service configurations, IT assets, infrastructure, and platforms. Systems Manager integrates with AWS Config to collect and maintain an inventory of infrastructure resources and application and OS data starting with your environment and account structure. Simple, automated setup processes allow you to quickly enable operational best practices, such as continuous vulnerability scanning and collecting insights into improving an application’s performance and availability. It also helps you to automate operational best practices by codifying operations runbooks and defining execution parameters, such as freeze periods and permissions. AWS Config rules enable continuous compliance by enforcing security, risk, and compliance policies.
2. **Enable visibility into applications and infrastructure**: The second stage is to enable visibility into applications and infrastructure by continuously tracking key technical and business measures, providing visibility, and triggering automation and actions if needed.

Systems Manager provides operational visibility and actionable insights to customers through a widget-based, customizable dashboard that can be tailored for users such as IT operators, DevOps engineers, IT leaders, and executives. Operational teams can set up operational event management and reporting for their infrastructure and resources at scale using pre-defined configuration bundles that reflect operational best practices to filter out and capture specific operational events that require an operator’s attention for diagnosis and action/remediation. The dashboard provides a holistic view of relevant data across multiple AWS accounts and AWS Regions, such as inventory and CMDB, patch, resource configuration and compliance, support tickets, insights from EC2 Compute Optimizer, Trusted Advisor, Personal Health Dashboard, Amazon CloudWatch, and trends on outstanding operational issues.

1. **Automate operations at scale**: To proactively automate operations at scale, Systems Manager gives teams the ability to automate patch management to keep resources secure. Systems Manager provides change management capabilities with built-in approval workflows and secures automated or manual change execution when making application and environment changes. Only approved changes can be deployed to the environment by authorized resources, and these come with detailed reporting. You can automate server administration, providing central IT teams with a consistent, integrated console to perform common administrative tasks. Additionally, you can manage and troubleshoot resources on AWS and on-premises. Operators can manage their VM fleet when manual actions are required by connecting directly from the console. You can also operationalize risk management by creating rules. Here is an example AWS Systems Manager rule that operationalizes risk management by checking for security vulnerabilities in installed software packages on Amazon EC2 instances:
2. {
3. "Name": "ec2-check-for-security-vulnerabilities",
4. "Description": "Scans installed software packages on EC2 instances for security vulnerabilities",
5. "ResourceId": "\*",
6. "ResourceType": "AWS::EC2::Instance",
7. "ComplianceType": "NON\_COMPLIANT",
8. "RulePriority": 1,
9. "Operator": "EQUALS",
10. "Parameters": {
11. "ExecutionFrequency": "OneTime",
12. "OutputS3BucketName": "sa-book-s3-bucket",
13. "OutputS3KeyPrefix": "ec2-security-scans/"
14. },
15. "Actions": [
16. {
17. "Type": "RunCommand",
18. "Properties": {
19. "Comment": "Scan installed software packages for security vulnerabilities",
20. "OutputS3BucketName": "sa-book-s3-bucket",
21. "OutputS3KeyPrefix": "ec2-security-scans/",
22. "DocumentName": "AWS-RunShellScript",
23. "Parameters": {
24. "commands": [
25. "apt update",
26. "apt-get install -y unattended-upgrades",
27. "apt-get install -y --only-upgrade bash",
28. "apt-get install -y --only-upgrade glibc",
29. "apt-get install -y --only-upgrade libstdc++6",
30. "apt-get install -y --only-upgrade libgcc1",
31. "apt-get install -y --only-upgrade libc6",
32. "apt-get install -y --only-upgrade libc-bin",
33. "apt-get install -y --only-upgrade libpam-modules",
34. "apt-get install -y --only-upgrade libpam-runtime",
35. "apt-get install -y --only-upgrade libpam0g",
36. "apt-get install -y --only-upgrade login",
37. "apt-get install -y --only-upgrade passwd",
38. "apt-get install -y --only-upgrade libssl1.0.0",
39. "apt-get install -y --only-upgrade openssl",
40. "apt-get install -y --only-upgrade dpkg",
41. "apt-get install -y --only-upgrade apt",
42. "apt-get install -y --only-upgrade libapt-pkg4.12",
43. "apt-get install -y --only-upgrade apt-utils",
44. "apt-get install -y --only-upgrade libdb5.3",
45. "apt-get install -y --only-upgrade bzip2",
46. "apt-get install -y --only-upgrade libbz2-1.0",
47. "apt-get install -y --only-upgrade liblzma5",
48. "apt-get install -y --only-upgrade libtinfo5",
49. "apt-get install -y --only-upgrade libreadline7",
50. "apt
51. ]
52. }

By running analyses to proactively detect and remediate risks across applications and infrastructure, such as expiring certificates, a lack of database backup, and the use of blocked software, identified risks are assigned to owners and can be remediated using automation runbooks with automated reporting.

1. **Remediate issues and incidents**: Finally, when unexpected issues arise, you must be able to remediate issues and incidents quickly. With the incident, event, and risk management capabilities within Systems Manager, you can employ various AWS services to trigger relevant issues or incidents. It integrates with Amazon GuardDuty for threat detection and AWS Inspector for security assessments, keeps a running check on vulnerabilities in the environment, and allows automated remediation. It provides a consolidated view of incidents, changes, operational risks and failures, operational alarms, compliance, and vulnerability management reports. It allows operations teams to take manual or automated action to resolve issues. Systems Manager speeds up issue resolution by automating common, repeatable remediation actions, such as failover to a backup system and capturing failed state for root cause analysis.

Systems Manager’s incident management capability automates a response plan for application issues by notifying the appropriate people to respond, providing them with relevant troubleshooting data, and enabling chat-based collaboration. You can easily access and analyze operational data from multiple AWS services and track updates related to incidents, such as changes in alarm status and response plans. Operators can resolve incidents manually by logging into the instance or executing automation runbooks. Systems Manager integrates with AWS Chatbot to invoke commands in the appropriate channel for an incident so central IT teams can resolve issues quickly. Let’s learn about the final and most important step in the cloud operation model: managing cost.

**Sixth pillar – Manage your cloud’s finance**

Cloud adoption has enabled technology teams to innovate faster by reducing approval, procurement, and infrastructure deployment cycles. It also helps finance organizations eliminate the failure cost, as cloud resources can be terminated with just a few clicks or API calls. As a result, technology teams are no longer just builders, but they also operate and own their products.

They are responsible for most activities that were traditionally associated with finance and operations teams, such as procurement and deployment.

**Cloud Financial Management** (**CFM**) enables finance, product, technology, and business organizations to manage, optimize and plan costs as they grow their usage and scale on AWS. The primary goal of CFM is to enable customers to achieve their business outcomes cost-efficiently and accelerate economic and business value creation while balancing agility and control. CFM has the following four dimensions to manage and save costs:

**Plan and evaluate**

When planning for future cloud spending, you should first define a goal for the monthly cost of the individual project. The project team needs to make sure cloud resources related to a project are correctly tagged with cost allocation tags and/or cost categories. This way, they can calculate and track the monthly cost of the project with the cost and usage data available in their AWS Cost Explorer and AWS Cost and Usage Reports. These reports provide the data you need to understand how your AWS costs are incurred and optimize your AWS usage and cost management. These reports can be customized to include only the needed data and can be delivered to an Amazon S3 bucket or an Amazon SNS topic. You can also use the data in these reports to create custom cost and usage reports, set up cost and usage alarms, and create budgets.

You can decide on a project’s budget based on the growth trend of the project as well as the available funds set aside for the project. Then, you can set the budget thresholds using AWS Budgets for cost or resource usage. You can also use AWS Budgets to set coverage and utilization targets for the project team’s Reserved Instances and Savings Plans. These are two options that allow you to save money on your AWS usage costs. They both enable you to purchase a discounted rate for your AWS usage, in exchange for committing to a certain usage level over a specific period. **Reserved Instances** are a type of pricing model that allows you to save up to 75% on your Amazon EC2 and RDS usage costs by committing to a one- or three-year term. With Reserved Instances, you pay a discounted hourly rate for the usage of a specific instance type in a specific region, and you can choose between Standard and Convertible Reserved Instances.

**AWS Savings Plans** is a new pricing model that allows you to save up to 72% on your AWS usage costs by committing to a one-year or three-year term. With Savings Plans, you pay a discounted hourly rate for your AWS usage, and you can choose between Compute Savings Plans and Amazon EC2 Instance Savings Plans. Compute Savings Plans offer a discount on a wide range of AWS services, including Amazon EC2, Fargate, and Lambda, while Amazon EC2 Instance Savings Plans only offer a discount on Amazon EC2 usage.

The AWS Budgets Reports dashboard allows you to monitor the progress of your budget portfolio by comparing actual costs with the budgeted costs and forecasted costs with the budgeted costs. You can set up notification alerts to receive updates when the cost and usage are expected to exceed the threshold limit. These alerts can be sent via email or Amazon **Simple Notification Service** (**SNS**). You can learn more about AWS Budgets by visiting the AWS page here: <https://aws.amazon.com/aws-cost-management/aws-budgets/>. After planning your budget, let’s learn about managing it.

**Manage and Control**

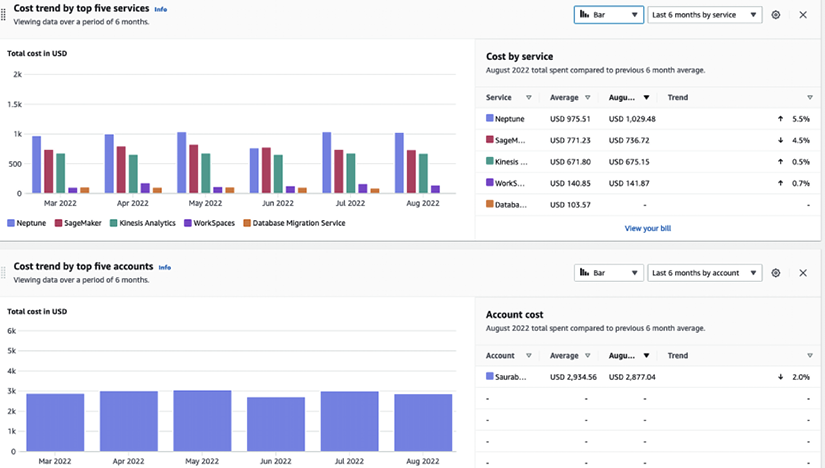
As businesses grow and scale on AWS, you need to give your team the freedom to experiment and innovate in the cloud while maintaining control over cost, governance, and security. And while fostering innovation and speed is essential, you also want to avoid getting surprised by the bill.

You can achieve this by establishing centralized ownership through a center of excellence. Cost management elements are shared responsibilities across the entire organization. A centralized team is essential to design policies and governance mechanisms, implement and monitor the effort, and help drive company-wide best practices.

You can utilize services like **Identity and Access Management** (**IAM**) to ensure secure access control to AWS resources. IAM allows you to create and manage user identities, groups, and roles and grants permissions for IAM users to access specific AWS resources. This way, you can control and restrict individual and group access to AWS resources, ensuring the security of your infrastructure and data.

Use **AWS Organizations** to enable automatic policy-based account creation, management, and billing at scale. Finally, using the **AWS Billing Console**, you can easily track overall spending and view cost breakdown by service and account by accessing Billing Dashboard. You can view the overall monthly spending from last month, the current month, and the current forecasted month.

The following screenshot is a sample billing dashboard:



*Figure 9.9: AWS Billing Dashboard*

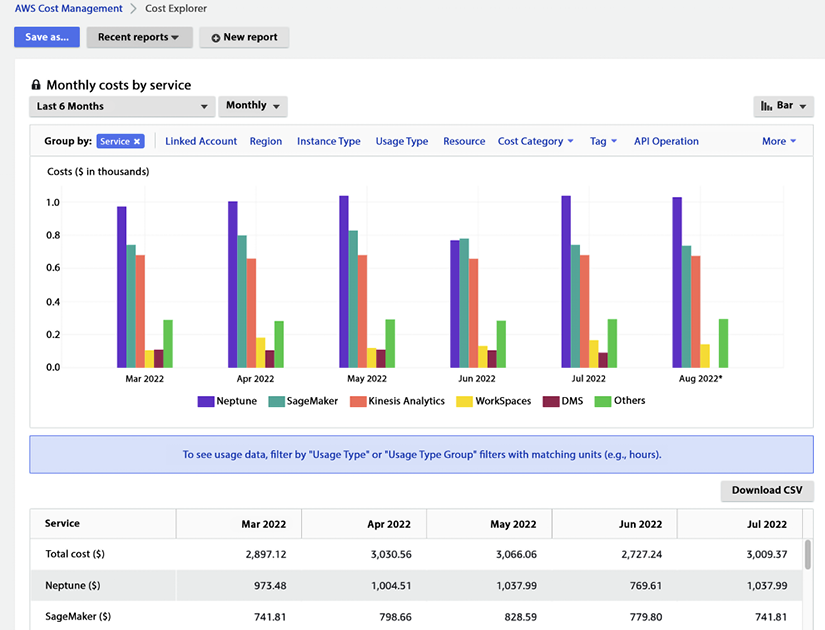
The above billing dashboard shows six months spend by service and cost trend. Using the Billing Console, you can receive a unified view of spend in a single bill and establish rules for organizing costs, sharing discount benefits associated with Reserved Instances and Savings Plans, and many other controls. You can learn more about the AWS Billing Console by visiting the AWS page here: <https://aws.amazon.com/aws-cost-management/aws-billing/>.

**AWS Purchase Order Management** enables you to use **purchase orders** (**POs**) to procure AWS services and approve invoices for payment. With this service, you can configure multiple POs, map them to your invoices, and access the invoices generated against those POs. Additionally, you can manage the status of your POs, track their balance and expiration, and set up email notifications for contacts to receive alerts when POs are running low on balance or close to their expiration date. You can learn more about AWS Purchase Order Management by visiting the AWS page here: <https://aws.amazon.com/aws-cost-management/aws-purchase-order-management/>.

**AWS Cost Anomaly Detection** is a machine learning service that automates cost anomaly alerts and root cause analysis. It can save time investigating spending anomalies by providing automated root cause analysis and identifying potential cost drivers, such as specific AWS services, usage types (e.g., data transfer cost), regions, and member accounts. You can learn more about AWS Cost Anomaly Detection by visiting the AWS page here: <https://aws.amazon.com/aws-cost-management/aws-cost-anomaly-detection/>. As you manage your cost, let’s learn how to track it.

**Track and allocate**

You need to ask three questions to understand your billing uses. The **first** is, *What is causing our bill to increase?* AWS provides **AWS Cost Explorer** to help answer this question. Cost Explorer provides a quick visualization of cost and utilization with default reports and creates specific views with filters and grouping, as shown below. This tool can help show which AWS services are leading to increased spending:



*Figure 9.10: AWS Cost Explorer*

In the above Cost Explorer dashboard, you can see service expense grouping in the costs chart showing AWS Neptune has the highest cost, followed by SageMaker. It also provides the ability to download CSV files for detailed analysis. You can learn more about AWS Cost Explorer by visiting the AWS page here: <https://aws.amazon.com/aws-cost-management/aws-cost-explorer/>.

The **second** question is a common follow-up: *Which of my lines of business, teams, or organizations drove increased spending and usage?* AWS Cost Explorer’s data becomes even more valuable when paired with Cost Allocation Tags. These features allow customers to categorize their resources and spending to fit their organization’s needs. Categorizing spending by specific teams, sometimes referred to as show back, allows for better analysis and easier identification of savings opportunities. You can also use the AWS **Cost and Usage Report** (**CUR**) to bring in cost and usage data, including tags, into your analysis tool of choice. This approach also allows for combining the data with other business-specific data. You can learn more about CUR by visiting the AWS page here: <https://aws.amazon.com/aws-cost-management/aws-cost-and-usage-reporting/>.

Finally, the **third** question is, *How do I understand the cost impact of Reserved Instances?* AWS Cost Explorer provides multiple ways to view and dive deep into this data, such as tag service with meaningful info, e.g., owner, project, application, or you can define cost categories by group accounts, tags, services, and charge types with custom rules. You can use CUR to deliver cost data to the S3 bucket, which can be integrated with Amazon Athena and/or ingested into your ERP system.

Furthermore, **AWS Billing Conductor** is a billing and cost management tool that helps you monitor and optimize your AWS costs and usage and provides insights into how you are using your resources. AWS Billing Conductor provides recommendations for ways to optimize your costs, such as by identifying idle or underutilized resources that can be turned off or scaled down. You can learn more about AWS Billing Conductor by visiting the AWS page here: <https://aws.amazon.com/aws-cost-management/aws-billing-conductor/>.

**AWS Application Cost Profiler** is a service that allows you to collect and correlate usage data from your multi-tenant applications with your AWS billing information. This enables you to generate detailed, tenant-specific cost reports with hourly granularity delivered daily or monthly. This can be useful for understanding the cost breakdown of your multi-tenant application and allocating costs to individual tenants or customers. You can learn more about AWS Application Cost Profiler by visiting the AWS page here: <https://aws.amazon.com/aws-cost-management/aws-application-cost-profiler/>. Now let’s learn about the final step to optimize costs and increase savings.

**Optimize and save**

Cost optimization is about making sure you pay only for what you need. AWS offers tools and services that make it easier for you to build cost-efficient workloads, which help you to continue saving as you scale on AWS. While there are hundreds of ways you can pull to minimize spending, at its core, two of the following most impactful levers you can pull to optimize spending are detailed next.

**Using the right pricing models**: AWS offers services through multiple pricing models – on-demand, pay-as-you-go, commitment-based Reserved Instances, and Spot instances for up to 90% discount compared to on-demand pricing. Amazon EC2 Spot Instances are a type of AWS computing resource that is available at a reduced price and can be used to run your applications. These instances are spare capacities in the AWS cloud that can be interrupted at any time, based on the resource demand. Spot Instances are suitable for applications with flexible start and end times and can handle interruptions without causing significant issues. You can launch a Spot Instance by specifying the maximum price you are willing to pay per hour (known as the “bid price”). If the current Spot price is less than your bid price, your Spot Instance will be launched, and you will be charged the current Spot price. However, if the current Spot price exceeds your bid price, your Spot Instance will be interrupted, and you will not be charged for the usage.

Suppose you have predictable, steady workloads on Amazon EC2, Amazon ECS, and Amazon RDS. If you use Reserved Instances, you can save up to 75% over on-demand capacity. Reserved Instances are a pricing option in AWS that allows you to pay up front for a commitment to use a certain amount of resources over a specific period in exchange for a discounted price. There are three options for purchasing Reserved Instances:

* **All up-front (AURI)**: This option requires you to pay for the entire term of the Reserved Instance up front and provides the most significant discount.
* **Partial up-front (PURI)**: This option requires a partial payment up front and provides a smaller discount than the AURI option.
* **No upfront payments (NURI)**: This option does not require any up front payment and provides the smallest discount.

By choosing the AURI option, you can receive the largest discount on your Reserved Instances. The PURI and NURI options offer lower discounts but allow you to pay less up front or avoid upfront payments altogether.

AWS offers Reserved Instances, and Savings Plans purchase recommendations via Cost Explorer based on your past usage. Cost Explorer identifies and recommends the estimated value resulting in the largest savings. It allows you to generate a recommendation specific to your purchase preference. From there, you can track your investment using cost explorer. Any usage above the commitment level will be charged at on-demand rates. You can revisit commitment levels, make incremental purchases, and track coverage and utilization using pre-built reports in cost explorer.

With AWS purchase option recommendations, you can receive tailored Reserved Instance, or Savings Plans purchase recommendations based on your historical usage. You can select parameters for recommendations, such as the type of plan, term commitment, and payment option that makes sense for your business.

**Identify and eliminate idle or over-provisioned resources**: AWS Cost Explorer resources can be used for top-level **key performance indicators** (**KPIs**), rightsizing, and instance selection. Cost Explorer will estimate monthly savings, which is the sum of the projected monthly savings associated with each recommendation.

Cost Explorer rightsizing recommendations generate recommendations by identifying idle and underutilized instances and searching for smaller instance sizes in the same instance family. Idle instances are defined as CPU utilization of <1%, while underutilized instances are defined as those with CPU utilization between 1% and 40%.

In this section, you learned about various ways to manage your cost and make the cloud more profitable to run your workload and business.

**Summary**

Working in a cloud environment is different from on-premise. The out-of-the-box tools and services are available in AWS can make a huge difference when operating your workload in the cloud. To realize the full value of the cloud, it’s important to understand cloud operation and how to apply automation everywhere. In this chapter, you learned about the cloud operation model and the six pillars of CloudOps, which help you to understand how to plan your cloud operation efficiently.

Under the CloudOps pillars, you learned about building cloud governance, infrastructure provisioning, monitoring, centralized operation management, and cost optimization. You learned about various AWS services that can help you to build end-to-end cloud operation pillars. These services include AWS CloudTrail, AWS Config, AWS CloudWatch, AWS Systems Manager, AWS CloudFormation, AWS Service Catalog, and AWS cost explorer.

Data is one of the important drivers of moving to the cloud. In the next chapter, you will learn about the AWS services available for extracting, transforming and loading large volumes of data in AWS.