**Module 7 introduction**

**Learning objectives**

In this module, you will learn how to:

* Summarize approaches to monitoring your AWS environment.
* Describe the benefits of Amazon CloudWatch.
* Describe the benefits of AWS CloudTrail.
* Describe the benefits of AWS Trusted Advisor.

**Video transcript**

–

As the owner of the coffee shop, I want to watch what's going on throughout the day, making sure things are running smoothly. In the coffee shop, I can see people are getting their coffees, and things look generally fine. But I can't just sit there and watch things all day long. I would like to eventually leave and maybe come back at the end of the day, and be able to check in and see how the shop did when I was gone.

For example, I might want to know how many coffees were sold today? How long was the average wait time for someone when ordering a coffee? Did we run out of inventory for anything today? Even better yet, I'd love to be able to be automatically alerted if the wait times become too long. That way I can call in another employee or go pitch in myself. Every business, including this coffee shop, can use metrics to measure how well systems and processes are running.

This idea of observing systems, collecting metrics, evaluating those metrics over time, and then using them to make decisions or take actions, is what we call monitoring.

It's important to set up monitoring in the cloud. With the elastic nature of AWS services that dynamically scale up and down, you'll want to keep a close pulse on your AWS resources to ensure that your systems are running as expected.

For example, if an EC2 instance is being over-utilized, you can trigger a scaling event that automatically would launch another EC2 instance. Or if an application starts sending error responses at an unusually high rate, you can alert an employee to go take a look at what's going on.

In the next few videos, we will cover a variety of tools that help you monitor your AWS environment. This monitoring will help you measure how your systems are performing, alert you when things aren't right, and even help you debug and troubleshoot issues as they come along.

# Amazon CloudWatch

**Video transcript**

–

Now we're making lots of coffee, serving customers and everything seems to be going spiffy in our coffee shop. But as we run espresso machines more and more, use mugs, constantly open and close the fridge, we want to be able to make sure we're alerted to something which might have gone awry. Maybe an espresso machine needs to be cleaned or repaired. The point is, as a business owner, you need visibility into the state of your systems. Are things running well? Do you see a degraded customer experience? Are you frequently delivering the wrong drink to customers or is everyone getting their orders as expected. You have all sorts of questions about the success of your operations.

And the same idea applies to systems built on AWS. You need a way to monitor the health and operations of your solutions. Luckily, you don't need to build out your own monitoring platform. We have done this for you.

Introducing Amazon CloudWatch. CloudWatch allows you to monitor your AWS infrastructure and the applications you run on AWS in real time. It works by tracking and monitoring metrics. Think of metrics as variables that are tied to your resources. For example, the amount of espressos made by an espresso machine or the CPU utilization of an EC2 instance.

Let's take a customer approach for our coffee shop. Say we have an espresso machine and it needs to be cleaned after it makes 100 espressos. CloudWatch allows us to create a custom metric called Espresso Count. And once it hits 100, we want to alert staff to clean the machine. Simple, right? Well with CloudWatch, you can accomplish this by creating what is called a CloudWatch alarm. You set a threshold for a metric, and when that threshold is reached CloudWatch can generate an alert and trigger an action. This means we can alert on the custom metric, in this case, hitting 100, and then perform an action.

Even better, CloudWatch alarms are integrated with SNS. So we can then send an SMS to the manager to say, clean the machine. You can create all sorts of custom alarms for metrics from all different types of AWS resources.

Now, what if we want to aggregate all those metrics in a single pane of glass? Well, we can use CloudWatch's dashboard feature. Think of a dashboard as a screen, which lists out metrics in near real time. In our case, we can create a CloudWatch dashboard which will show us all our espresso machines and their espresso counts so we can proactively monitor them. These dashboards would auto refresh when they are open so we can see an up-to-date view of our resources.

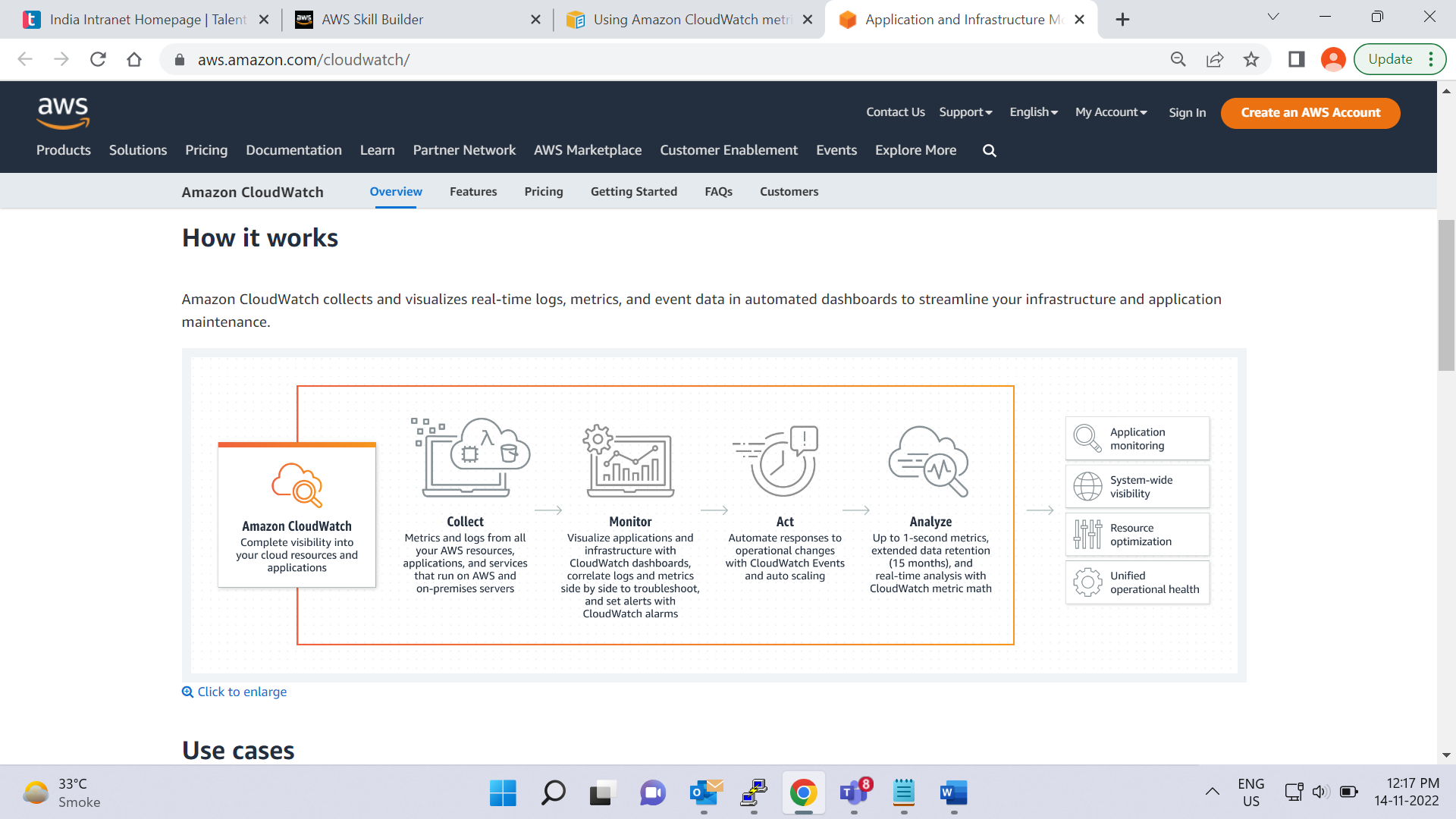
Finally, what are the benefits of using a service such as CloudWatch? Well, the first one is that you can have access to all your metrics from a central location. This enables you to collect metrics and logs from all your AWS resources applications, and services that run on AWS and on-premises servers, helping you break down silos so that you can easily gain system-wide visibility.

You can also get visibility across your applications, infrastructure, and services, which means you gain insights across your distributed stack so you can correlate and visualize metrics and logs to quickly pinpoint and resolve issues. This in turn means you can reduce mean time to resolution, or MTTR, and improve total cost of ownership, or TCO. So in our coffee shop, if the MTTR of cleaning hours restaurant machines is shorter then we can save on TCO with them. This means freeing up important resources like developers to focus on adding business value.

Lastly, you can drive insights to optimize applications and operational resources. By, for example, aggregating usage across an entire fleet of EC2 instances to derive operational and utilization insights. And now our staff can focus on serving coffee versus constantly cleaning machines before they are due to be cleaned.

And there you have it, folks - you're up to speed on what CloudWatch entails. Now if you excuse me, I have a date with an espresso machine to refill my coffee.

**Amazon CloudWatch**



[**Amazon CloudWatch**](https://aws.amazon.com/cloudwatch/) is a web service that enables you to monitor and manage various metrics and configure alarm actions based on data from those metrics.

(Metrics are data about the performance of your systems. By default, many services provide free metrics for resources (such as Amazon EC2 instances, Amazon EBS volumes, and Amazon RDS DB instances). You can also enable detailed monitoring for some resources, such as your Amazon EC2 instances, or publish your own application metrics. Amazon CloudWatch can load all the metrics in your account (both AWS resource metrics and application metrics that you provide) for search, graphing, and alarms.

Metric data is kept for 15 months, enabling you to view both up-to-the-minute data and historical data.

To graph metrics in the console, you can use CloudWatch Metrics Insights, a high-performance SQL query engine that you can use to identify trends and patterns within all your metrics in real time.

)

CloudWatch uses [**metrics**](https://docs.aws.amazon.com/AmazonCloudWatch/latest/monitoring/working_with_metrics.html) to represent the data points for your resources. AWS services send metrics to CloudWatch. CloudWatch then uses these metrics to create graphs automatically that show how performance has changed over time.

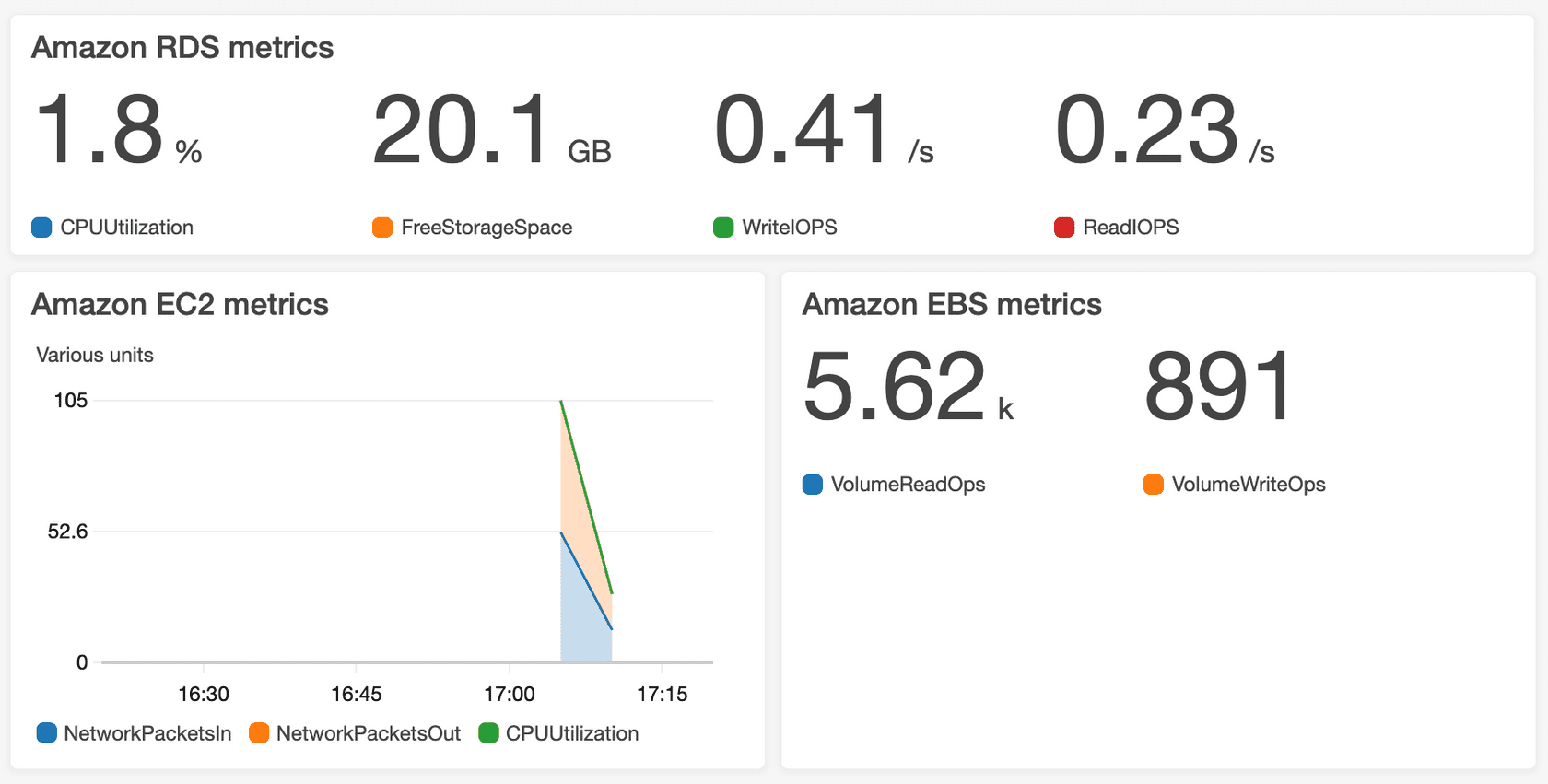
**CloudWatch alarms**

With CloudWatch, you can create [**alarms**](https://docs.aws.amazon.com/AmazonCloudWatch/latest/monitoring/AlarmThatSendsEmail.html) that automatically perform actions if the value of your metric has gone above or below a predefined threshold.

For example, suppose that your company’s developers use Amazon EC2 instances for application development or testing purposes. If the developers occasionally forget to stop the instances, the instances will continue to run and incur charges.

In this scenario, you could create a CloudWatch alarm that automatically stops an Amazon EC2 instance when the CPU utilization percentage has remained below a certain threshold for a specified period. When configuring the alarm, you can specify to receive a notification whenever this alarm is triggered.

**CloudWatch dashboard**



The CloudWatch [**dashboard**](https://docs.aws.amazon.com/AmazonCloudWatch/latest/monitoring/CloudWatch_Dashboards.html) feature enables you to access all the metrics for your resources from a single location. For example, you can use a CloudWatch dashboard to monitor the CPU utilization of an Amazon EC2 instance, the total number of requests made to an Amazon S3 bucket, and more. You can even customize separate dashboards for different business purposes, applications, or resources.

(Amazon CloudWatch dashboards are customizable home pages in the CloudWatch console that you can use to monitor your resources in a single view, even those resources that are spread across different Regions. You can use CloudWatch dashboards to create customized views of the metrics and alarms for your AWS resources)

# AWS CloudTrail

**Video transcript**

–

The Cash Register, one of the world's first self-auditing devices. The principle is simple: trust but verify. You have your entire store being run by a clerk that you trust, but you want to make sure that the cash in the drawer matches the actual sales. So every transaction then is recorded and tabulated. So at the end of the day, you know exactly what should be there.

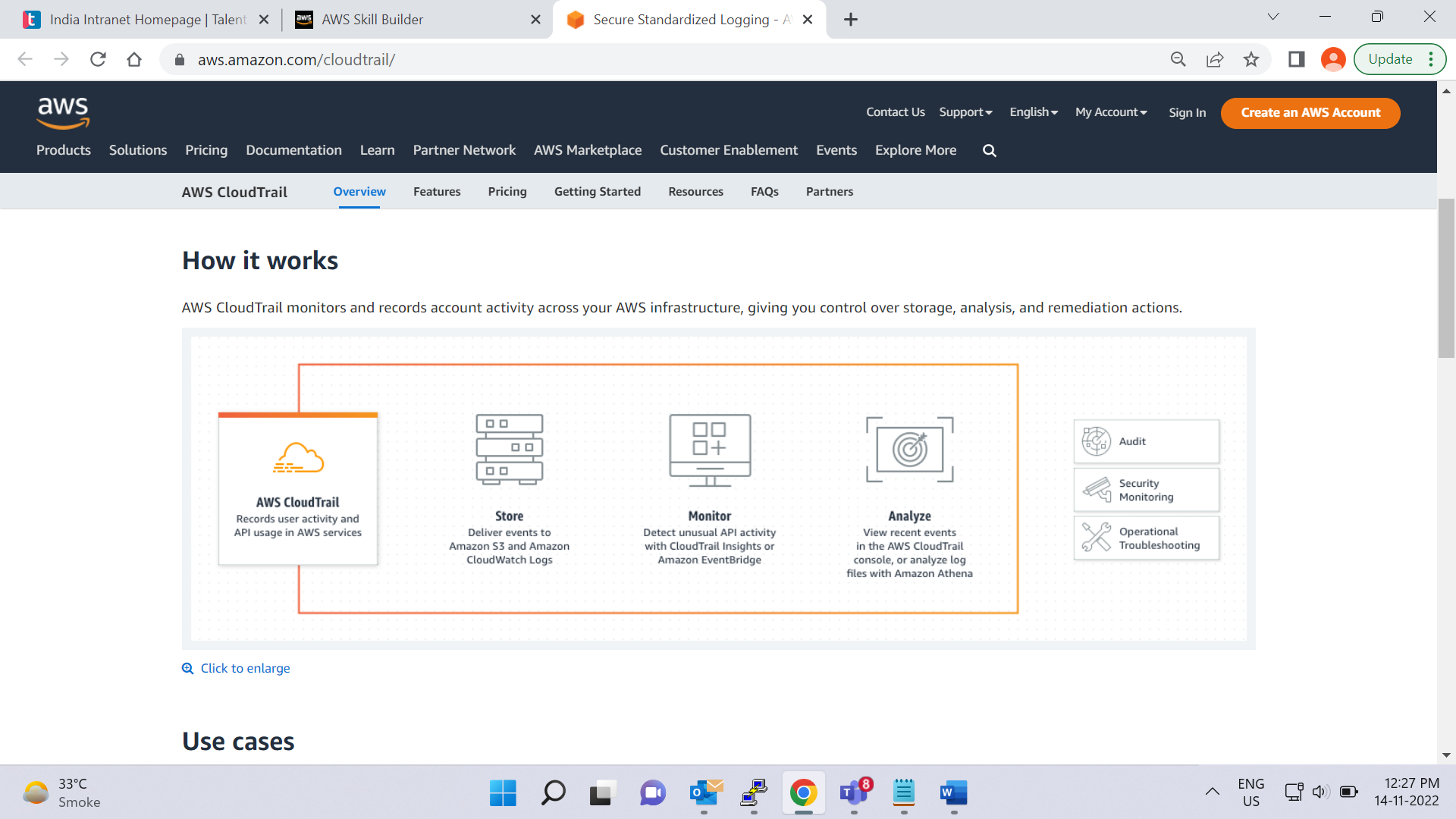
Being able to audit transactions in IT is a critical element in most compliance structures. But in a physical data center, there are so many places where a human can, even by accident, make changes without any record of that change getting recorded. At AWS, that problem goes away because everything is programmatic.

Introducing AWS CloudTrail, the comprehensive API auditing tool. The engine is simple, every request made to AWS, it doesn't matter if it's to launch an EC2 instance, or add a row to a DynamoDB table, or change a user's permissions. Every request gets logged in the CloudTrail engine. The engine records exactly who made the request, which operator, when did they send the API call? Where were they? What was their IP address? And what was the response? Did something change? And what is the new state? Was the request denied?

From an auditing perspective, well, this is fabulous. So imagine that you're dealing with an auditor who is checking to make sure that nobody from the outside can access your database. That's a good thing. Well, okay, you build a security group that locks out external traffic. But remember that a root-level administrator still has permissions to change those settings, right?

Well, so how do you prove to the auditor that the security group settings never changed? The answer is CloudTrail. And then CloudTrail can save those logs indefinitely in secure S3 buckets. In addition, with tamper-proof methods like Vault Lock, you then can show absolute provenance of all of these critical security audit logs.

**AWS CloudTrail**



[**AWS CloudTrail**](https://aws.amazon.com/cloudtrail/) records API calls for your account. The recorded information includes the identity of the API caller, the time of the API call, the source IP address of the API caller, and more. You can think of CloudTrail as a “trail” of breadcrumbs (or a log of actions) that someone has left behind them.

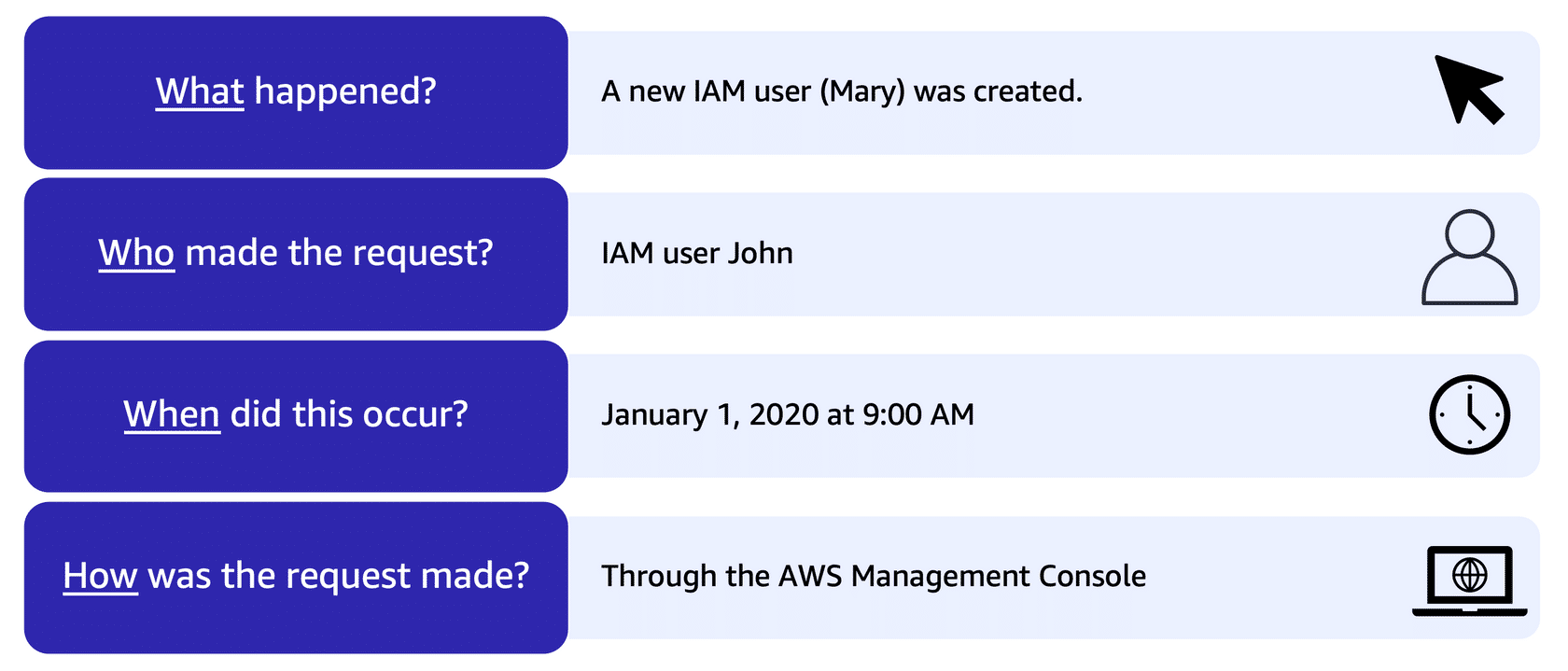
Recall that you can use API calls to provision, manage, and configure your AWS resources. With CloudTrail, you can view a complete history of user activity and API calls for your applications and resources.

Events are typically updated in CloudTrail within 15 minutes after an API call. You can filter events by specifying the time and date that an API call occurred, the user who requested the action, the type of resource that was involved in the API call, and more.

**Example: AWS CloudTrail event**

Suppose that the coffee shop owner is browsing through the AWS Identity and Access Management (IAM) section of the AWS Management Console. They discover that a new IAM user named Mary was created, but they do not who, when, or which method created the user.

To answer these questions, the owner navigates to AWS CloudTrail.



In the CloudTrail Event History section, the owner applies a filter to display only the events for the “CreateUser” API action in IAM. The owner locates the event for the API call that created an IAM user for Mary. This event record provides complete details about what occurred:

On January 1, 2020 at 9:00 AM, IAM user John created a new IAM user (Mary) through the AWS Management Console.

**CloudTrail Insights**

Within CloudTrail, you can also enable [**CloudTrail Insights**](https://docs.aws.amazon.com/awscloudtrail/latest/userguide/logging-insights-events-with-cloudtrail.html). This optional feature allows CloudTrail to automatically detect unusual API activities in your AWS account.

For example, CloudTrail Insights might detect that a higher number of Amazon EC2 instances than usual have recently launched in your account. You can then review the full event details to determine which actions you need to take next.

# AWS Trusted Advisor

**Video transcript**

–

When running a business, you might need some advisers who can come in from the outside and say, hey, this process should be streamlined. Or, hey, I have some good tips on how to save money on your overhead. Or even, hey, I noticed I was able to waltz right in, go behind your cash register, and open the drawer without anyone noticing. Not good.

The point I'm trying to make is, sometimes it's nice to have someone who knows the industry best practices, and knows what to look for, come in, and tell you what you need to change in order to run more efficiently, be more secure, or to save some money.

AWS has an automated advisor called AWS Trusted Advisor. This is a service that you can use in your AWS account that will evaluate your resources against five pillars. The pillars are cost optimization, performance, security, fault tolerance, and service limits. Trusted Advisor in real time runs through a series of checks for each pillar in your account, based on AWS best practices, and it compiles categorized items for you to look into, and you can view them directly in the AWS console.

Some checks are free and are included in your AWS account, and others are available depending on the level of your support plan. Some examples of checks are, if you don't have multi-factor authentication turned on for your root user, it's going to let you know. If you have underutilized EC2 instances that might be able to be turned off in order to save money, or if you have EBS volumes that haven't been backed up in a reasonable amount of time, it will let you know that, too. To get a better idea, let's look at AWS Trusted Advisor in my own account.

I'm already logged into the console, and I'm going to type in Trusted Advisor into the search. This brings me to the Trusted Advisor dashboard and I'm going to click on the Cost Optimization pillar first to view what it has found. You can see right away there are three levels of items it is showing me. There was the red circle, which means action recommended. The orange triangle, which means investigation recommended. And the green square, which means that there were no problems detected.

Lucky for me, I don't have any red items for Cost Optimization, but I do have some orange items. Under Cost Optimization checks, I can read more about each check that Trusted Advisor ran. In this account, I do have some RDS instances that are idle, as well as some underutilized EC2 instances and EBS volumes. I could decide to scale these instances down vertically to save on cost, or if they aren't being used at all I could decide to delete those instances and those EBS volumes altogether. It would require some investigation for me to determine what to do here.

Now, let's click on the Performance pillar. On this one, I don't have any warnings, but we can still read what checks Trusted Advisor performed. Like this one, for example, which checks for EBS volumes whose performance might've been affected by the throughput capability of the EC2 instance that it's attached to.

Next, let's select the security pillar. In this demo account, you can see there are four alerts that are at the action recommended level. Trusted Advisor is trying to alert me, telling me I have weak password policies for IAM users, multi-factor authentication is not turned on for the root user, and there are security groups allowing public access to EC2 instances. All of these items are putting the resources in this account at risk, and should be dealt with as soon as possible.

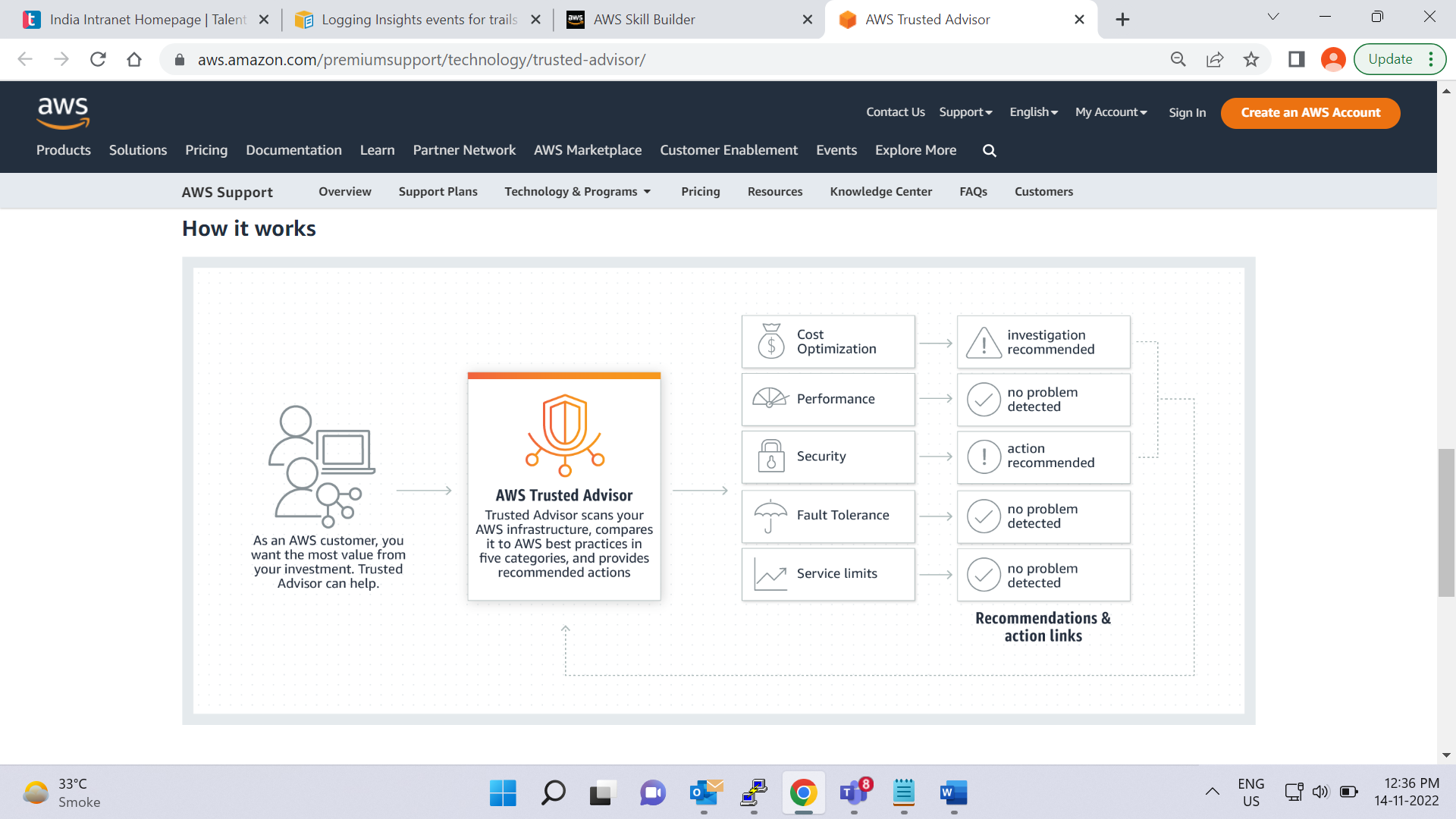
Now let's move on to fault tolerance. For this, there were a few things that are found to be insufficient. First, Trusted Advisor is telling me that there are EBS volumes without snapshots in this account. Remember, a snapshot is a backup. So without backups, if I had an EBS volume fail, I would lose that data.

Another alert we have here is that my EC2 AZ balance is at the action recommended level. This means that my EC2 instances are not properly launched across AZs. So if one AZ has trouble, my application might incur an outage. Deploying across AZs would be the answer to this one.

Finally, let's check out Service Limits. This pillar will tell you when you are approaching or hitting AWS service limits. A lot of service limits are soft limits, meaning they are restrictions that can be lifted to some degree. It's good to know when you are approaching one of those limits, and when it's time to turn in a Support ticket with AWS to get them changed. In this account, I can see that I have five VPCs, which is the regional limit. So I've hit that specific service limit.

Trusted Advisor can help point you in the right direction when it comes to the five pillars. You can set up email alerts that go out to billing, operations, and security contacts, as checks get run in your account. Make sure you have Trusted Advisor turned on so that you too can start taking action to optimize your AWS account.

**AWS Trusted Advisor**



[**AWS Trusted Advisor**](https://aws.amazon.com/premiumsupport/technology/trusted-advisor/) is a web service that inspects your AWS environment and provides real-time recommendations in accordance with AWS best practices.

Trusted Advisor compares its findings to AWS best practices in five categories: cost optimization, performance, security, fault tolerance, and service limits. For the checks in each category, Trusted Advisor offers a list of recommended actions and additional resources to learn more about AWS best practices.

The guidance provided by AWS Trusted Advisor can benefit your company at all stages of deployment. For example, you can use AWS Trusted Advisor to assist you while you are creating new workflows and developing new applications. Or you can use it while you are making ongoing improvements to existing applications and resources.

**AWS Trusted Advisor dashboard**



When you access the Trusted Advisor dashboard on the AWS Management Console, you can review completed checks for cost optimization, performance, security, fault tolerance, and service limits.

For each category:

* bullet

The green check indicates the number of items for which it detected **no problems**.

* bullet

The orange triangle represents the number of recommended **investigations**.

* bullet

The red circle represents the number of recommended **actions**.

# Module 7 summary

In Module 7, you learned about the following concepts:

* Amazon CloudWatch
* AWS CloudTrail
* AWS Trusted Advisor

**Video transcript**

–

Understanding what is happening in your environment is key to maintaining efficient, secure, and compliant applications.

We discussed how CloudWatch can provide near real-time understanding of how your system is behaving, including being alerted to conditions that require your attention. CloudWatch also gives you the ability to look at those metrics over time as you tune your system for maximum performance.

We discussed how CloudTrail can help you know exactly who did what, when, and from where. It answers all of your AWS auditing questions, except why they did it.

And finally, we looked at Trusted Advisor that compiles a quick dashboard of over 40 common concerns around cost, performance, security, and resilience in an actionable dashboard.

There are, of course, many additional monitoring and analytical tools available for your use, but this will help you get a good feeling for some of the variety AWS offers your business.