Audio file

Your Recording 23.wav

Transcript

00:00:09 Speaker 2

In this lesson, we're going to look at AWS managed services, we're going to look at what the difference is between a managed and an unmanaged service, we're going to look at why we give you managed services in the AWS platform, and then we'll look at both the challenges and the benefits of using managed services.

00:00:26 Speaker 2

We'll then see how that applies to the Sagemaker AI product and what results we can expect to achieve by using Sagemaker managed services. So let's start thinking about unmanaged services. First of all, I'm going to use the example of the Amazon service called EC2 or Elastic Compute Cloud. The EC2 service is for creating virtual machines. We call them instances. Now with that service, what do I need? I would need an AWS account. I would need something called a VPC, a virtual private cloud.

00:00:55 Speaker 2

Really just a collection of networks. Think of it maybe like a bundle of Vlans. I would need at least one network. We call that a subnet. We would need a security group to control what traffic was allowed in and out. And then we could create our instance, our virtual machine. Now that would all align into one of our, what we call an availability zone. An availability zone is, well, we like to think of it a little bit like a data center. In reality, it's a cluster of data centers, but for conceptual purposes, let's think of it like a data center. If I wanted my virtual machine to be highly available, then I would need.

00:01:25 Speaker 2

Another network, another set of ingress rules. I would need another instance deployed in a second availability zone. Again, think of that like in a second data center at least 100 kilometers away. So before I could even run a Windows or Linux virtual machine, I would need to have provisioned security groups, subnets, and VPC before I could even start creating a virtual machine. So unmanaged service is all about you configuring it yourself. Now we say manual setup, but what we mean here is you could mean in the management console.

00:01:55 Speaker 2

Or by command line, or programmatically, or with infrastructure as code like Cloudformation or Terraform. What we mean is that you need to define everything that needs to be provisioned. It's on you. Now, the advantage here is this places you in complete control over the infrastructure. You own it end to end. So there will be nothing in the virtual infrastructure except what you explicitly asked for. But that means that responsibility for, well, everything, failover, high availability, redundancy, patching, all of that sits with you, the person who deployed that.

00:02:25 Speaker 2

Unmanaged service.

00:02:28 Speaker 2

Hmm, OK, so we do get the control, but we take on all responsibility. Let's compare that to managed services. In a managed service from AWS. The point of the managed service is that we abstract the infrastructure complexity away from you. So we will take care of it. The service will do the infrastructure provisioning. If you need a VPC, it will create one. If it needs a subnet and an instance, it will create it on your behalf. You don't get to see it, but it's there and it as a platform will manage it for you. Now a big advantage here is this means that.

00:02:57 Speaker 2

Things like scaling of your platform and high availability management and failover, well, that will be the responsibility of the managed service.

00:03:04 Speaker 2

So pick the RDS service for example, that is relational database service. If I provision a database with that product, I'm able to tick a box to say yes I want high availability and simply by ticking a box I would get 2 databases with replication and failover. That's a great example of how a managed service differs from an unmanaged 1. You simply said yes I want availability with that and all the complexity of making that happen resides with AWS. So you might be asking, well why would I do this? The reason I would take advantage of managed services.

00:03:34 Speaker 2

Is it gives me a faster time to value? We are not spending time provisioning infrastructure in order to use a service. Imagine it was RDS again and I wanted to use an Oracle database or a Postgres database. I would go to RDS and say, can I have a database please with availability? Thanks very much. And it's provisioned. I can immediately start putting data in there or querying data. Your time to value is far shorter than if you'd started with a virtual machine. You'd have to create the subnets, the VPC provisioned Windows or Linux, install the database product, and then you'd be at the same point.

00:04:04 Speaker 2

So how quickly do you want to get to the point of actually using the service so we get a faster tank to value with a managed service. Now, there's a support aspect here too in that because the service is being managed by Amazon, that means that they have a responsibility to keep it up and running. So you can imagine that there will be entire teams monitoring the platform and making sure that it operates smoothly. And they are doing that at a scale that is very hard to replicate outside of AWS. So again, you are gaining the advantage here by using that managed service because it's highly available and being.

00:04:34 Speaker 2

Tuned constantly. And there's another aspect here and that is maintenance. When I think about when I looked after physical servers many years ago that were in data centers, we would get notification of vulnerabilities in our operating systems and our applications. And it was our responsibility to find where those applications and operating systems were installed and schedule in some time to take them offline and patch them. And I can safely say no one ever enjoyed that process. But we know that CVE vulnerabilities are being found all the time and it is a continuous task to keep them updated.

00:05:02 Speaker 2

Well, again, when it comes to AWS managed services, why should that be you? Why not let the platform itself take care of that? If servers need updated, then let AWS teams provision new ones, make sure they're patched, and then switch us over onto the new instances. If there is any fault tolerance, any failure within a data center within Amazon, well, let them handle it and make sure that your service continues uninterrupted. I would much rather they did that rather than me. So let's make this Sagemaker specific. Now we know to get started in Sagemaker, let's say as a data scientist, we're going to need.

00:05:32 Speaker 2

To do some exploratory data analysis.

00:05:34 Speaker 2

We will do that exploratory data analysis using code, specifically Python code that we will run in the development environment that we call a Jupyter Notebook. Now, when I worked with data scientists in the past, they worked with Jupyter notebooks running locally on their laptops. So that made them specifically limited in terms of compute size, how much CPU and RAM that they had, and also limited us in terms of datasets gaining access to sensitive data sets. So we could just copy them down to laptops and thinking about keeping control of that data.

00:06:02 Speaker 2

So if we can provide an AWS managed service that provides A Jupyter Notebook environment for data scientists that's hosted in the cloud, and we can simply say yes, I want that, please, and the service simply takes care of it, that's a huge win for us. We can get our Jupyter Notebook environment and start working on our data securely in the cloud, likely where our data already is, without having to pull data down locally to devices or rely on the compute requirements locally in our laptops.

00:06:27 Speaker 2

No infrastructure setup was required. We didn't require to create virtual machines in EC2. We simply said we want a Jupyter environment and magically we got 1. So Sagemaker and managed services absolutely go together. So when we are in Sagemaker, we've got a hosted Jupyter Lab environment that is fully managed that Amazon are responsible for delivering to you. We didn't need to provision anything in terms of virtual machines. But what about beyond that? Well, once you've got your Jupyter notebook, what you're likely wanting to do is process some data. For example, maybe you need to scale some of.

00:06:57 Speaker 2

Numeric values or maybe you need to change categorical data to numerical data. Do some data processing or when you're ready, maybe you want to train your model. So the wonderful thing about Sagemaker AI is that we can create delegated jobs to do that. So if I want to do data processing, I don't have to do it in the context of the compute that's providing me my Jupyter platform. Instead, I can create a dedicated job that will be size appropriate. So if I create a training job, I could specify compute size, how much CPU and memory is needed. I can specify the container image I want to use which will.

00:07:27 Speaker 2

Include the required algorithm I want to leverage and I can specify my training job settings, how long I want the training job to run, if I want the training job to be scaled out over multiple instances so that I can do some training in parallel. So that's all that I need to specify. And Sagemaker AI will take that and it will provision what's needed. It would provision the required compute. Now, again, what's happening under the hood is that Sigmaker is provisioning subnets or EC2 instances, whatever is required on your behalf. You don't get to see or manage that, but that's what it is doing.

00:07:57 Speaker 2

Sagemaker is obtaining the required container image from the Container registry in the same AWS region where your job is running. If you specified in your training job settings that you wanted distributed training, then many instances will be provisioned concurrently and we can be doing concurrent operations in each instance, thereby processing the training job faster. When it comes to hosting a model and using what we call a Sagemaker endpoint to host your model on a compute platform, we call that a Sagemaker endpoint. Well, if we want to, we can specify that a Sagemaker.

00:08:27 Speaker 2

Point is auto scaled, meaning that if there is more traffic coming in for requesting predictions then we can add more compute resource to that endpoint to meet that need so Sagemaker AI is acting as a managed service to help you get to the point of training your model and hosting your model quicker without getting tied up in infrastructure-related activities let's now consider the built-in features of sagemaker AI and the integrations that it has with other AWS services from a permissions and security perspective, we integrate with the AWS IAM service that's.

00:08:57 Speaker 2

Identity and Access Management service. This allows us to have a common way of describing users, groups and what permissions they get in each service. So for example, I might give a data scientist the ability to read and write to a particular S3 bucket, and I might get that same data scientist the ability to invoke a training job and run Jupyter Lab in Sagemaker Studio. From a network security perspective, we have integration with a virtual private cloud Now. We saw in the initial example in this lesson that in EC2 we needed a VPC.

00:09:27 Speaker 2

An environment made-up of networks that we could provision VMS into. And we say the benefit here of a managed service is you don't need to do that. And that is true. So I could create a Sagemaker training job, for example, and I don't need to create a VPC. It would simply run in a default network environment called the default VPC. But there will be times when you need your training job to interact with other endpoints that are in the same network security environment. For example, maybe you need to interact with a relational database or need to interact with some other object store.

00:09:57 Speaker 2

Rather than S3, then it might be a requirement that your training job runs where it can network reach a particular instance. In that case we can specify what VPC the training job is able to talk to. We have support for KMS, our key management service, allowing us to encrypt endpoints. That could be encrypting the contents of an S3 bucket, or that could be encrypting the virtual hard disk of the managed instances that run your training and data processing jobs. From a compute and storage perspective, we integrate with the ECR service.

00:10:24 Speaker 2

ECR is the Elastic Container Registry. This is simply a repository for storing container images. So we have container images that contain all of the data science tools that are needed as well as the built-in algorithms from Xgboos to SVM to LGBM and Linear Learner. They all reside in different container images, but they are stored in the Elastic Container registry. We have managed notebooks, so when I say that I want to have a Jupyter notebook, we create a managed notebook and it's underpinned by an EC2.

00:10:54 Speaker 2

But built into Sagemaker AI is this concept of prebuilt algorithms. If I know I'm going to be doing a linear regression or logistic regression, then using the off the shelf linear learner algorithm might make sense. But maybe if I'm doing a classification exercise, then maybe I want to use something like KNN the K nearest neighbors. But all these algorithms are available to me and I can just consume them. And ultimately all that is happening is a different container image will be used to run your training job. We can optionally auto scale. We can auto scale our endpoints for example if there is more inference traffic and we can deploy.

00:11:24 Speaker 2

New endpoints, new versions of our models where we send only a percentage of the inference traffic to the new version until we are sure that it's behaving correctly before we remove all the inference traffic over onto the new version of the model. So regardless of the user who is using the Sagemaker AI platform, be it the data scientist, a developer, a business user, when you leverage Sagemaker AI, the platform provisions the compute required under the covers, it's creating managed EC2 instances that are running managed containers. Those managed containers contain all.

00:11:54 Speaker 2

The data science tooling that is needed along with the built in algorithms that you need to perform your model training. If I need distributed training that is simply a property of the training job and how additional instances are spun up and created and the training job is distributed is all handled by Sagemaker. If I'm provisioning an endpoint so that I can host a model, then I can choose if I want that endpoint auto scaled. Meaning that if there is going to be more inference requests coming in, how should we handle that? Should we saturate and slow down our inference response or should we provision more?

00:12:24 Speaker 2

And if we choose to use auto scaling, then really we are delegating to Sagemaker to say you do it. These are the rules in which you operate, but you do the actual provisioning and the split of the job. Let's consider now how managed services yield effects and advantages for us.

00:12:39 Speaker 2

Firstly, we get faster results. If we're not having to spend time provisioning cloud infrastructure such as virtual private clouds, EC2 instances, security groups, containers, then we can jump straight to what we want to do, which is the ML tasks of data preparation, feature engineering, training and evaluation. I want to get to that point quicker and reduce the time to value. I want to take advantage of optimized expertise. In other words, if I may be a data engineer or data scientist or ML OPS engineer, then I should be focused on delivering the benefits.

00:13:09 Speaker 2

Of those personas, rather than have a data scientist try to figure out the rules of setting up a VPC or setting up security groups, that's not their focus. Let's keep people in their subject matter area of expertise. Getting a quick start means that we get an answer quicker. Now this isn't simply about faster results. This is failing fast. Now we want to get to the point of knowing, is this viable? And if I have to spend lots of time provisioning infrastructure just to answer the question, is it viable? Maybe hiring more data scientists to help with this project.

00:13:39 Speaker 2

When I actually just want to do a quick check, I want to do maybe a quick prototype. I could answer that question far quicker. If I can get to the point of feature engineering and quick model build in the space of half a day, it allows me to experiment and determine if I'm going to spend more time and resource on this project. Fail fast is a great capability. And if we're having to build infrastructure before we even get a viability question, we've kind of not got the benefit that cloud brings us. Cloud allows us to do things quickly and get up and running and answer those questions, but we also have that less infrastructure.

00:14:09 Speaker 2

Burden in that we, as we provision more things like more models hosted in Sagemaker, we develop more models. We're not building up a technical debt. We're not building up a legacy estate that we need to manage and maintain because it's already being managed and maintained by the managed service. We're not patching, we're not having to take things offline and liaise with the business for those infrastructure updates. Instead, Amazon are taking care of that for us. All we needed to do was simply build a model, deploy a model and forget about it. So what have we seen in this lesson we've seen that an AWS service could be managed or.

00:14:39 Speaker 2

Managed examples of managed services would be Relational Database Service or S3 service, or Sagemaker AI service. A good example of an unmanaged service would be EC2. But if I use managed services, they are abstracting a lot of the infrastructure tasks, allowing me to focus on the main purpose of what that service does. Or managed services give us control, but not quite as much control as an unmanaged service. Think of the example of EC2. We had to create the account we created to VPC. We created the security group then and only then.

00:15:09 Speaker 2

Create the instance that was all on us to create, but it gave us total control. Sagemaker, on the other hand, is going to utilize a background VPC by default. And if we want to, we can say, well, actually I want you to use this other one over here because it's got an endpoint I want you to talk to, but it doesn't allow us to specify every aspect of creating a new one. So there is a trade off here. OK, definitely is a trade off, but generally that trade off is worth it for the QuickTime to value experimentation and the ability to leverage a platform that is being managed and scaled by AWS. Remember that Sage?

00:15:39 Speaker 2

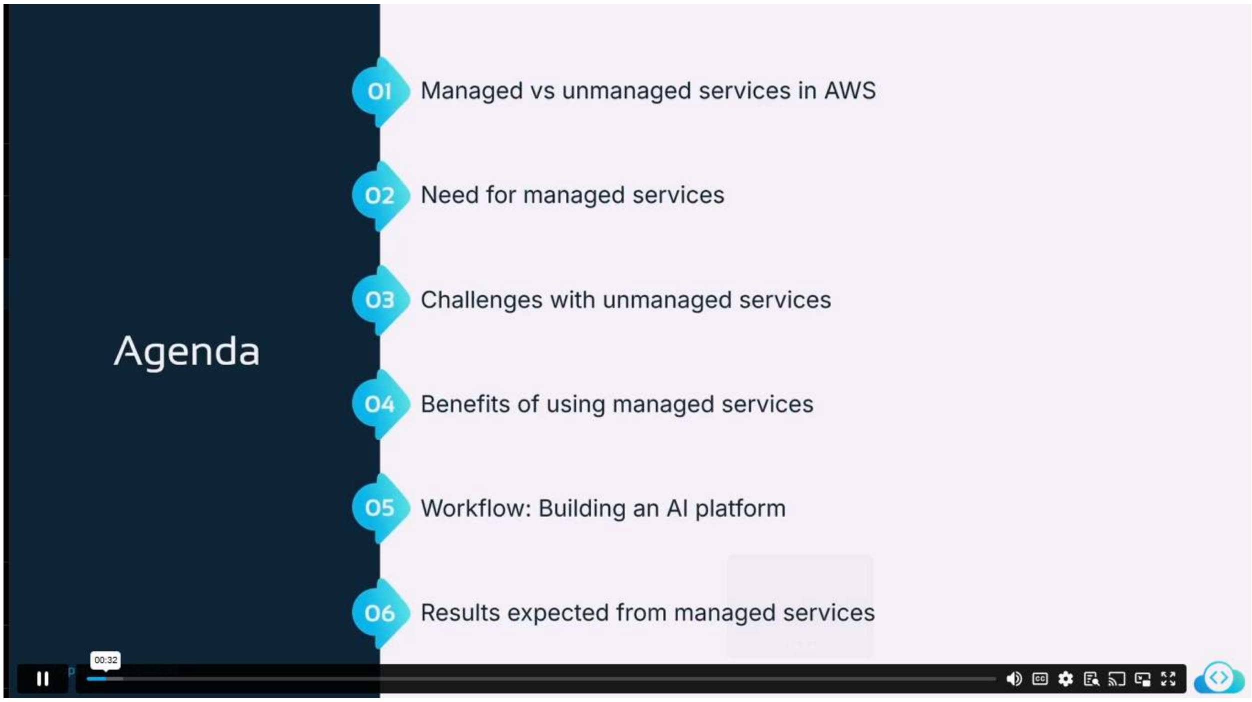
Is going to take all those infrastructure tasks, but you don't sacrifice compute sizing or the rules around scaling. For example, if I create a training job and I know it's a relatively small model and data set, maybe I only need 4 CPUs and 16 gigabytes of RAM. But I know for another deep learning job maybe I need 24 CPUs and half a TB of RAM and maybe 2 NVIDIA GPUs. Then you can specify that so it doesn't completely remove your ability to customize compute for the job required. But crucially, Sagemaker is going to accelerate your time to value.

00:16:09 Speaker 2

If you've identified and meet four machine learning to solve a business problem, then let's get to the point of inference as quickly as we can and prove that there's value in this approach. If I have to spend time and money and effort on human resource, on provisioning cloud infrastructure before I've even started, it becomes too long and too expensive and impactful on other projects. I want to fail fast. I want to get to the point where we can determine how much value this is going to deliver, if any. If we can get there quickly without tying up lots of people or needing lots of external experts, that's a huge win. So that wraps up this lesson in our.

00:16:39 Speaker 2

Lesson We're going to get an introduction to Jupiter notebooks and we can see how we can start our exploratory data analysis. See you there.



Agenda

01 Managed vs unmanaged services in AWS

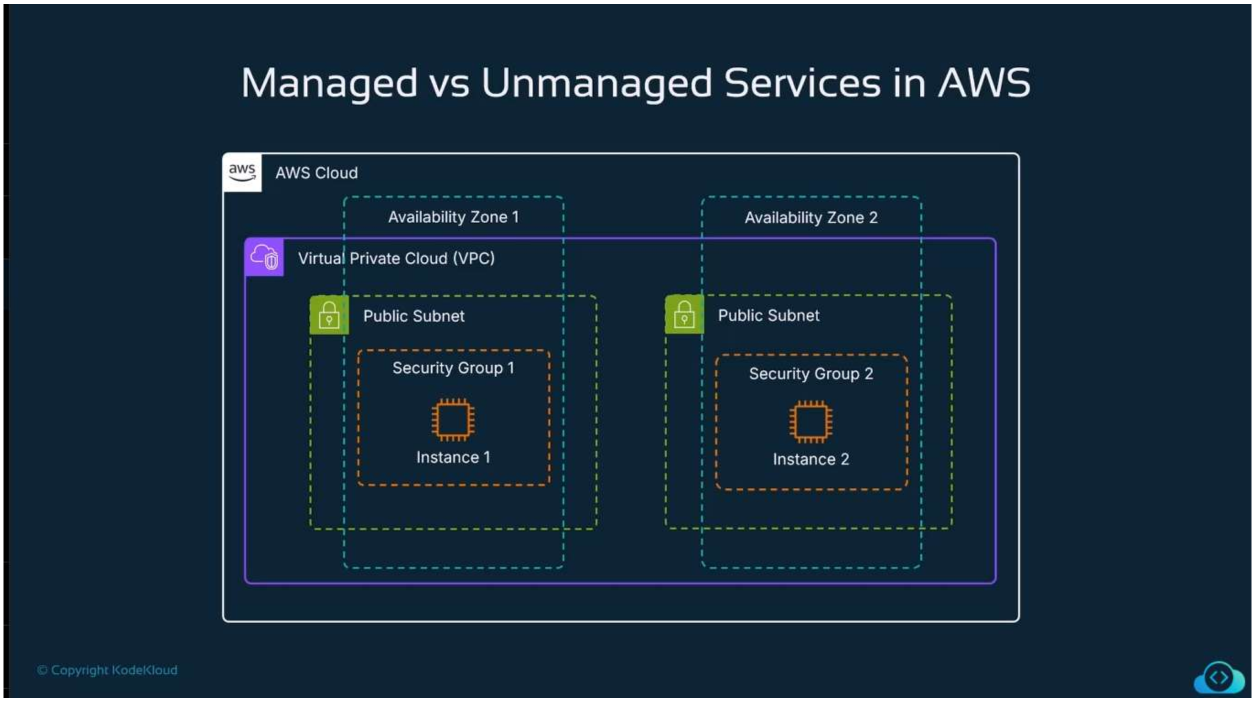
02 Need for managed services

03 Challenges with unmanaged services

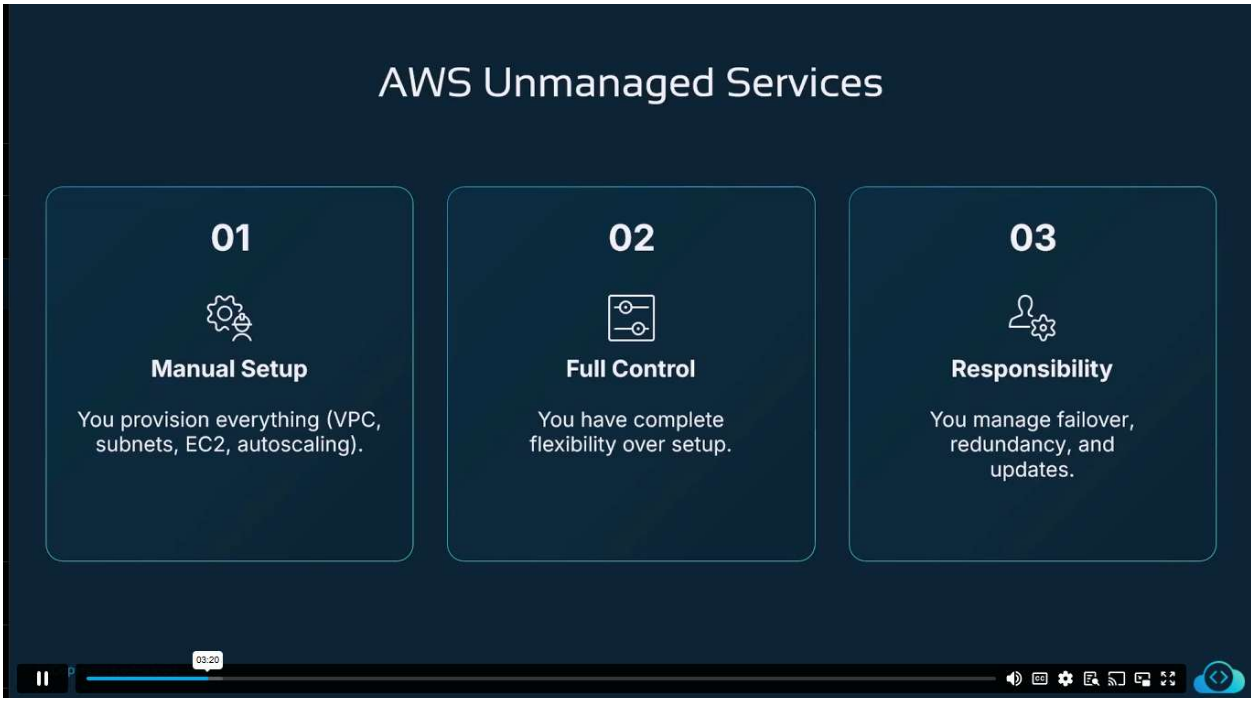
04 Benefits of using managed services

05 Workflow: Building an AI platform

06 Results expected from managed services



Managed vs Unmanaged Services in AWS



AWS Unmanaged Services

01

Manual Setup

You provision everything (VPC, subnets, EC2, autoscaling).

02

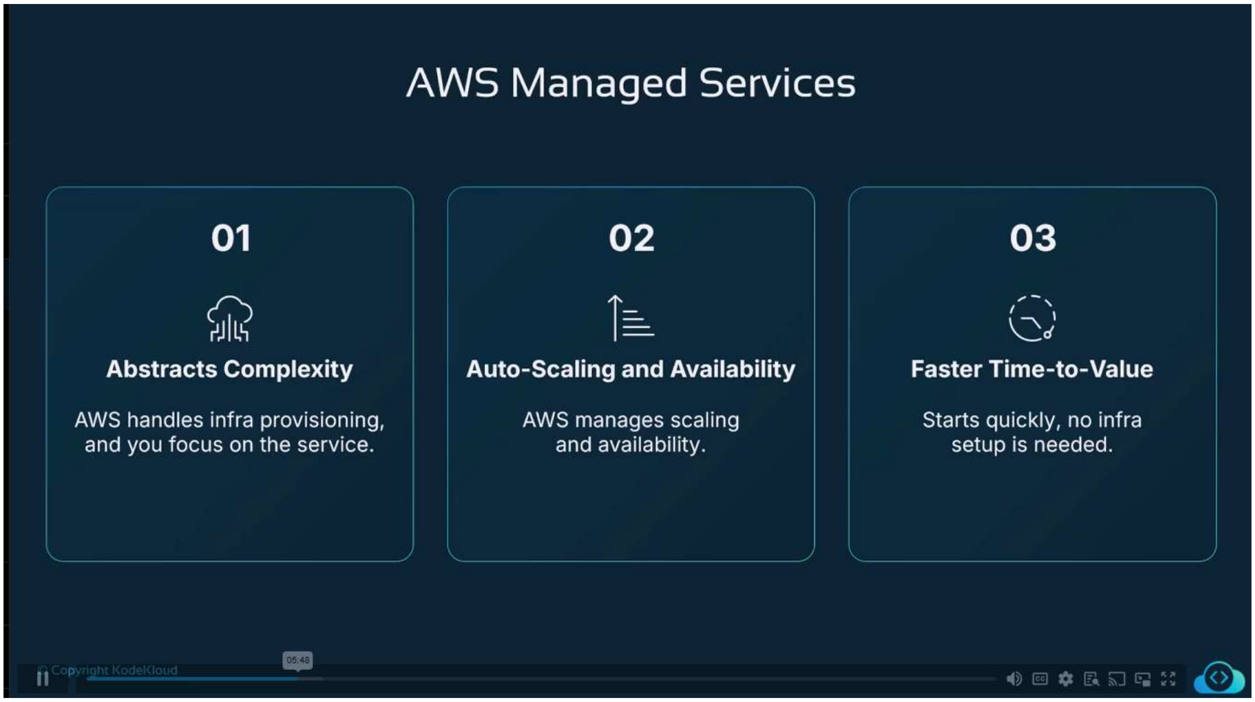
Full Control

You have complete flexibility over setup.

03

Responsibility

You manage failover, redundancy, and updates.



AWS Managed Services

01

Abstracts Complexity

AWS handles infra provisioning, and you focus on the service.

02

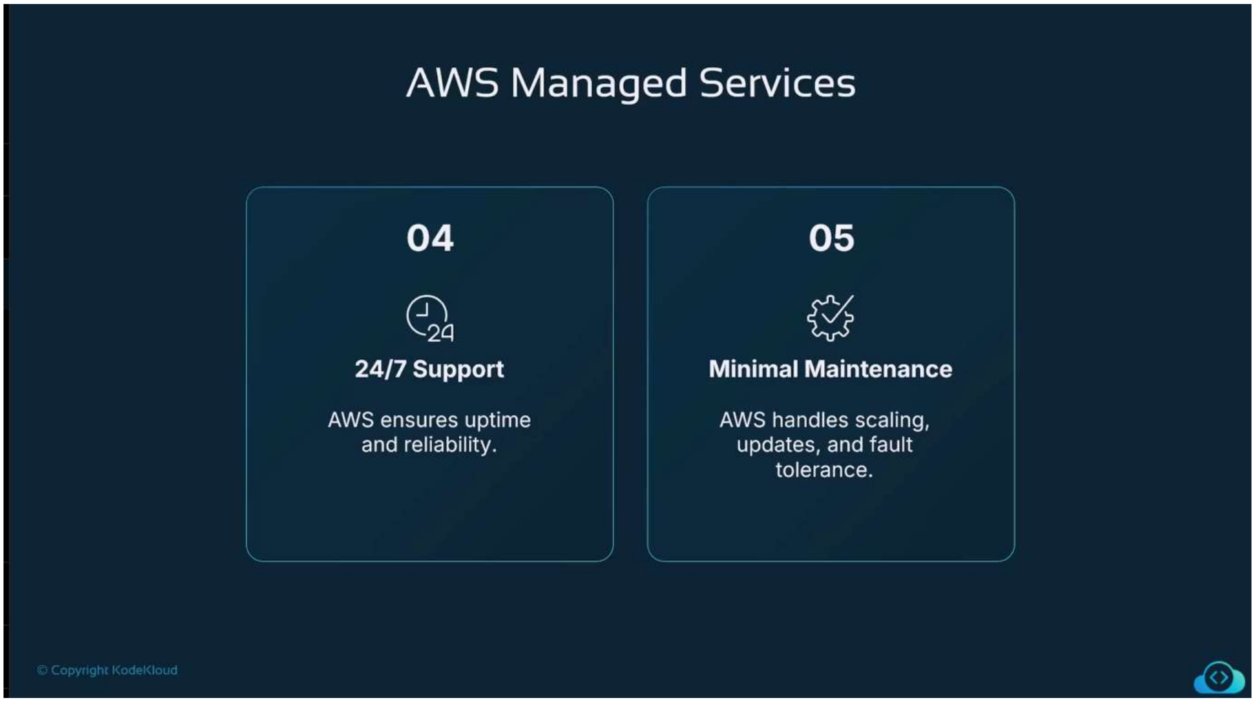
Auto-Scaling and Availability

AWS manages scaling and availability.

03

Faster Time-to-Value

Starts quickly, no infra setup is needed.



AWS Managed Services

04

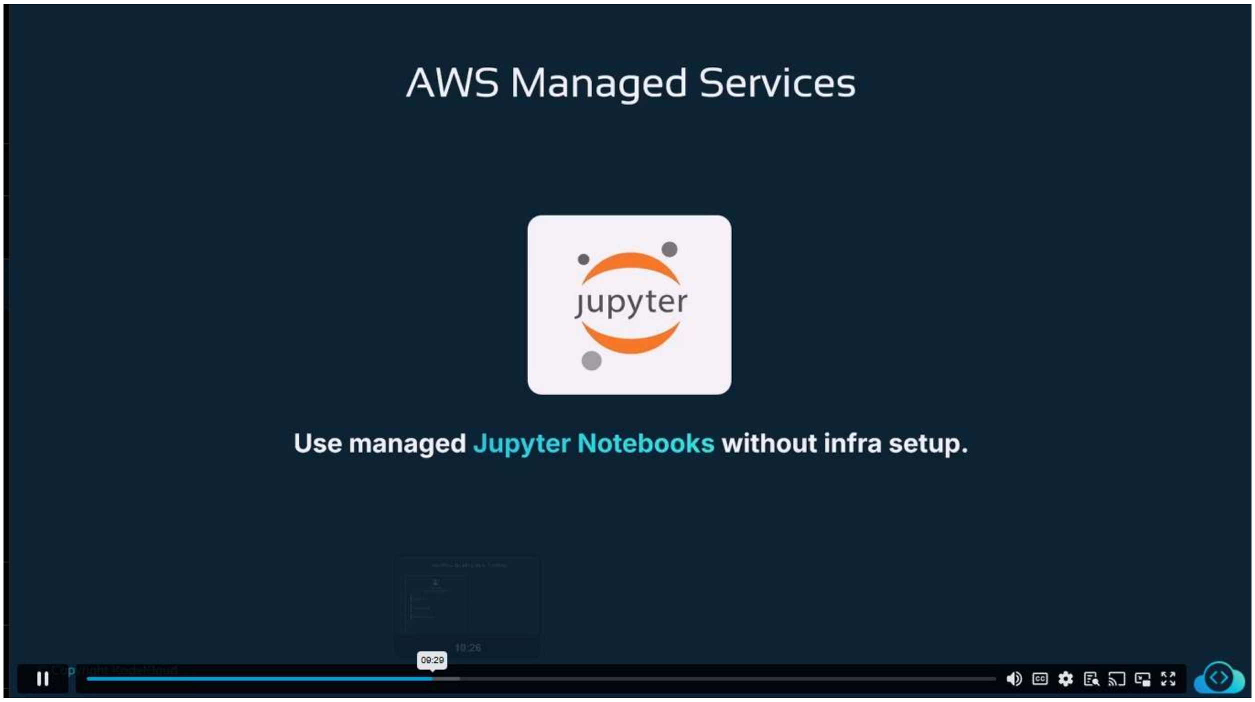
24/7 Support

AWS ensures uptime and reliability.

05

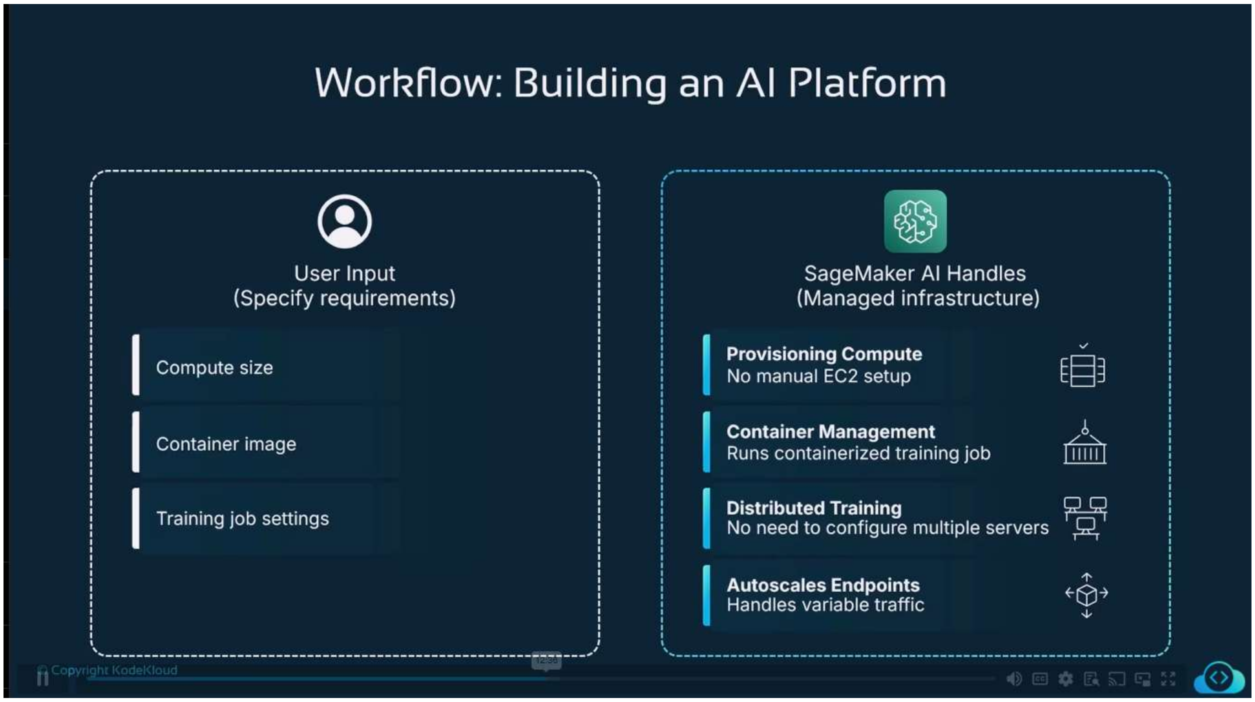
Minimal Maintenance

AWS handles scaling, updates, and fault tolerance.



AWS Managed Services

Use managed Jupyter Notebooks without infra setup.



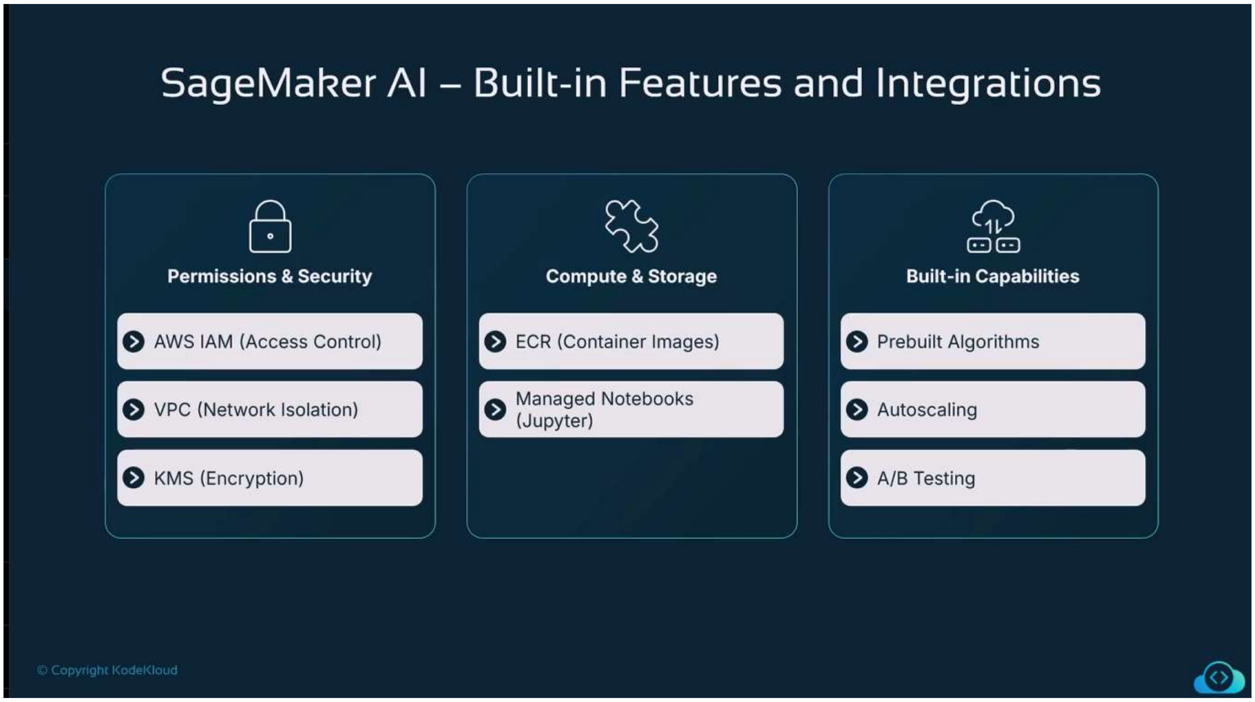
Workflow: Building an AI Platform

User Input
(Specify requirements)

- Compute size
- Container image
- Training job settings

SageMaker AI Handles
(Managed infrastructure)

- **Provisioning Compute**
  No manual EC2 setup
- **Container Management**
  Runs containerized training job
- **Distributed Training**
  No need to configure multiple servers
- **Autoscales Endpoints**
  Handles variable traffic



SageMaker AI – Built-in Features and Integrations

Permissions & Security

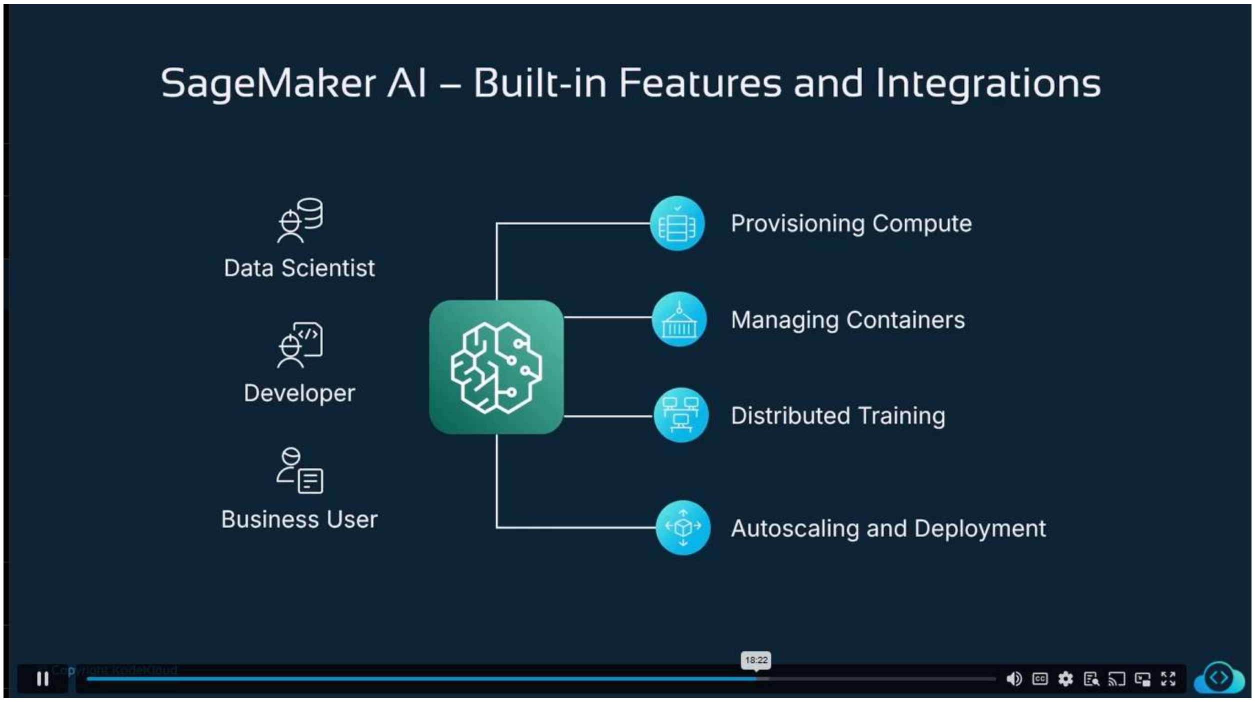
- AWS IAM (Access Control)
- VPC (Network Isolation)
- KMS (Encryption)

Compute & Storage

- ECR (Container Images)
- Managed Notebooks (Jupyter)

Built-in Capabilities

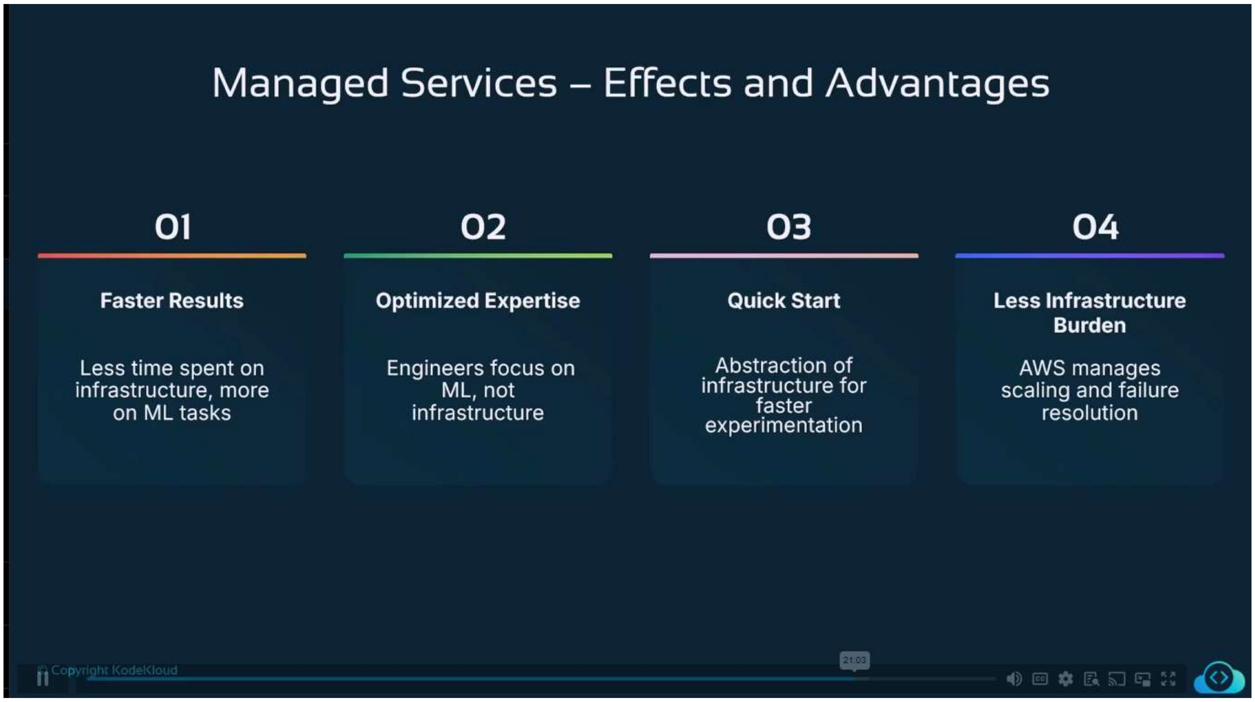
- Prebuilt Algorithms
- Autoscaling
- A/B Testing



SageMaker AI – Built-in Features and Integrations

- Data Scientist
- Developer
- Business User

- Provisioning Compute
- Managing Containers
- Distributed Training
- Autoscaling and Deployment



Managed Services – Effects and Advantages

01

Faster Results

Less time spent on infrastructure, more on ML tasks

02

Optimized Expertise

Engineers focus on ML, not infrastructure

03

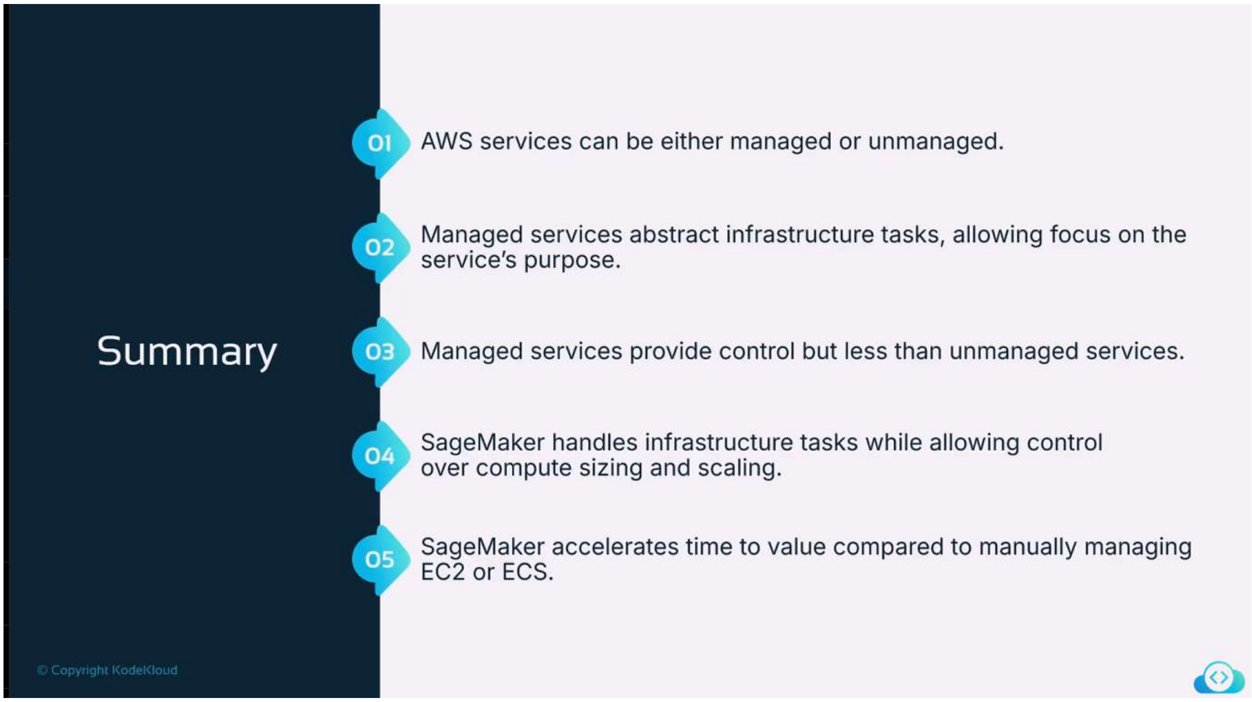
Quick Start

Abstraction of infrastructure for faster experimentation

04

Less Infrastructure Burden

AWS manages scaling and failure resolution



Summary

01 AWS services can be either managed or unmanaged.

02 Managed services abstract infrastructure tasks, allowing focus on the service's purpose.

03 Managed services provide control but less than unmanaged services.

04 SageMaker handles infrastructure tasks while allowing control over compute sizing and scaling.

05 SageMaker accelerates time to value compared to manually managing EC2 or ECS.