This video will cover Elastic Load balancing.

Modern high traffic websites must serve hundreds of thousands, if not millions of concurrent requests from users advice and then return the correct text, images, video or application data in a fast and reliable manner to meet.

High volume command additional servers are generally required.

Elastic Load balancing is an item to a service that distributes incoming applications or network traffic across multiple targets such as EC2 instances and containers, IP addresses and Lambda books in these single availability zone or across.

Elastic Load Balancing scales your load balancer as tracking to your application changes over time.

You can automatically scale 2 most quickly.

There are three types of elastic.

At the application level or layer 7 OSI model, it routes traffic to targets based on content of the request.

It is ideal for advanced load balancing of 8th and H.

An application load balancer provides advanced request routing that is targeted at delivery of modern application architectures including microservices and container based applications and Application load balancers simplifies and improves the security of your application by ensuring that the latest secure socket layer and transport.

Cyphers are and protocols are used at all times.

Additionally, a network load balancer operates the the network transfer level or layer four of the OSI model routing connections to target space.

That it protocol and it works well for load balancing both TCP and UDP tracking.

A network load balancer is capable of handling millions of requests per second while maintaining ultra low latency.

A network load balancer is optimized to handle sudden and volatile network traffic patterns.

Finally, a classic load balancer provides basic load balancing across multiple EC2 entries, and it operates at both the application level and network transfer.

A classic load balancer supports the load balancing of applications that use Https://andssl.

The classic load balancer is an older implementation and when possible it doesn't just recommends that you use a dedicated application or network system.

A load balancer accept incoming traffic from clients and routes requests to its registered targets in one or more availability zones.

You configure your load balancer to accept incoming traffic by specifying one or more listening.

A listener is there a process that checks and action request.

It is configured with a protocol.

Like.

Similarly, it is configured with a protocol and a phone number for connections from the load balancers to the targets.

You can also configure your load balancer to perform health checks, which are used to monitor the health of the registered targets, so that the load balancer only sends requests through the healthiest when the load balancer detects it.

Target it stopped growing traffic 2 desktop.

It then resumes routing traffic to that target when it detects that the target is helping again.

There is a key difference in how the load balancer.

Open application.

Ohh, you registered targets.

And route traffic to.

With class.

You register instances with the loader.

There are many elastic load balance.

Let's consider the main one first.

You can use Elastic Load Balancing to achieve high availability and better fault tolerance for your application.

Elastic Load Balancing across healthy targets with multiple.

One of the other part is in a single e-mail.

And on top.

Well, Ralph targeted traffic to healthy targets.

After the targets returned to a healthy state, load balancing will automatically resume.

That there's an enhanced container support for Elastic Load Melting.

Container.

Across multiple ports.

You can also.

Which provides a fully managed container offering.

You only need to register a service with a load balancer, and Amazon ECS transparently manages registration and deregistration of the Docker container to load balancer, automatically detects the port and dynamically reconfigures itself.

So we're more about Elastic Load balancing use cases.

No, no, no, no, no.

Analyze traffic.

I got you.

Amazon.

Elastic Load Balancer.

Die.

As an ordered.

No one has.

You can use metrics.

Performed.

For example, you can create a cloud watch alarm to monitor specified metric and anticipate an action such as sending a notification to an e-mail address.

If the metric goes outside of what you consider is acceptable, you can then send that message.

You can also use Access to capture detailed.

About the request.

And store them as a log.

Accounts on and three.

You can also.

To analyze traffic.

And then troubleshoot.

Target store back end application.

Finally, you can use a WS Cloud Trail to capture the detailed information about the call.

How are you doing?

You can use this cloud.

Baseline.

Thanks for watching We'll.

Welcome, this video will cover Amazon Cloud.

Watch these WS efficiently.

Any insights from your data resources?

For example, you might want to know what you should launch more Amazon.

Maybe use your applications performance or availability is being affected by the lack of sufficient capacity.

You might also want to know how much infrastructure is actually being used.

But how do you capture this information?

To answer these questions, you capture information with Amazon Cloud Watch.

Amazon Cloud Watch is a monitoring and observability service that is built for DevOps engineers, developers, psycho liability engineers, and high management.

WS resources and the applications that you're running a WS real time.

You can use Code Watch to collect and track records, which are variables that you can measure for your resources and applications.

You will create your Warm to Monitor 8 Amazon Cloud Watch metric in your account and use the alarm to automatically send a notification to Amazon Simple Notification Service or an Amazon SNS or perform an Amazon EC2 Autoscaling action or EC2 Active Directory.

For example, you can create alarms on EC2 instance.

Elastic Load Balancing request with Amazon Dynamo DB table throughput, Amazon Simple Queue service or even charges under AWS.

You can also create an alarm on custom metrics that you are specifying for your own applications or extra you can also use.

Cloud Watch events to define rules that match incoming events or changes in your a WS environment and route them to targets for processing.

This can include Amazon into two instances of this Lambda functions.

These streams, Amazon ECS tasks, step function state machines, Amazon SNS topics, Amazon US tools, and building targets within Cloud Watch events.

Cloud Watch events becomes aware of operational changes as they occur and respond to these.

In order to take corrective actions as necessary.

With Amazon Cloud Watch, you're being system wide visibility into resource utilization, application performance, and operational health.

There is no upfront commitment or minimum fee.

You simply pay for what you do.

You can create a cloud watch alarm based on a static threshold and anomaly detection, or a metric mass expression.

When you create an alarm based on a static threshold, you choose a cloud watch metric for the alarm and the threshold for that metric.

The alarm state is triggered when the metric breaches the threshold for the.

Specified number of evaluations.

Let's look at some of the things you must specify when you create an alarm based on static cooking.

The namespace contains cloud block message that you want, for example AWSC.

The metric is the variable that you want to measure, for example the utilization.

Statistics can be made average or some minimum or maximum sample channel, page, predefined percentile or even a customer.

Is the evaluation.

For the alarm.

When the alarm is evaluated, each.

Is aggregated into one data point.

When you specify the conditions for a static threshold, you specify whenever the metric is greater, greater or equal to, lower or equal to, or lower than the threshold value that you specify.

You can also specify additional configuration approach, which includes the number of data points within evaluation periods that we must be preached to trigger the alarm and how close should treat missing data when it evaluates the army.

Finally, you must specify the action that you want.

I'll watch the tape.

You can choose to send a notification to.

Amazon SNS topics or to perform an Amazon EC2 Autoscaling action or even Amazon EC2 actions.

Thank you for watching, we'll see you in the next video.

Welcome This video will cover Amazon's EC2 Auto Scaling.

Scaling is the ability to increase or decrease the computer task of your application.

You understand why automatic scaling is important.

Consider this example of a workload that has varied capacity requirements throughout the week.

In providing capacity to meet the highest demand, which occurs on Wednesday.

If you provide your password this way, then you're running resources that will be underutilized most days of the week.

With this option, the costs are not optimized.

Another option is to allocate less capacity to reduce costs.

However, in this situation you are under capacity on certain days.

Automatic capacity feeling is necessary to support the fluctuating demands from both.

Without stealing, your application could underperform or potentially even become unavailable for your users.

In the cloud, computing power is a programmatic resources.

This means you can take a flexible approach to feeling.

Amazon EC2 Autoscaling is an A doesn't service that helps you maintain application availability and enables you to automatically add or remove each.

You can use the fleet management features with PC through Auto Scaling to maintain the health and availability to sleep.

Amazon EC2 Autoscaling provides several ways to adjust scaling since that's really the needs of your application.

You can add or remove EC2 instances manually on a schedule, in response to changing demands, or in a combination with a WS Auto Scaling or Predictive scaling.

Dynamic feeling and predictive feeling to be used together.

Automatic scaling is useful for predictable workloads, for example, the weekly traffic of the retail company Amazon.

Automatic scaling is also useful for dynamic on demand scaling.

Amazon.com experiences A seasonal peak and.

End of November on Black Friday and Cyber Monday, which are days when US retailers over major fail if Amazon provision so they're getting fixed capacity to accommodate the highest use, 76% of the resources would be idle for most of the year.

An Auto Scaling Group is a collection of Amazon EC2 instances that are treated as a logical group.

The size of the Auto Scaling group depends on the number of interesting users taking the desired.

You can adjust this time to meet the demand either manually or by using automatic scaling.

You can specify the minimum number of instances in each autoscaling group, and Amazon EC2 Autoscaling will prevent the group from going below that size.

You can specify the maximum number of instances in each autoscaling group, and Amazon EC2 Autoscaling will prevent the group from going above this level.

If you specify the desired to ask either when you create the group or any time afterwards, Amazon EC2 Autoscaling is just the size of his group, so it has the specified number of instances.

If you specify Auto Scaling policies, then Amazon EC2 Autoscaling can launch or terminate instances when you have the command or the application increases or decreases.

For example, the diagram shows that this auto scaling group has a minimum size of 1.

Fire capacity of two inches.

And a maximum size of four inches.

The scaling policy is that you define adjust the number of instances you're gaining minimum and maximum number of.

Please.

With autoscaling, launching instances furniture and scaling out and terminating Interesting is referred to as scaling in.

The large basically influencing and other scaling group uses as launch configuration, which is an instance configuration.

When you create a launch configuration, you specify what information instances will use when.

The information is specified includes the ID of the AMI instance type, IMO one or more security groups and NDS Home.

Next, you specify where you want to scale.

You define the minimum and maximum number of instances and desired capacity of your Auto Scaling group.

Then you're launching into a subnet within a VPC.

Amazon EC2 Autoscaling integrates with Elastic Load Balancer to enable you to attack.

One or more load balancers to an existing Auto Scaling group.

After you attempt to load balancer, it automatically registered instances of the group and distributes incoming traffic across.

Finally, you specify when you want to scale the event.

You may have other.

You can configure your Auto Scaling groups to maintain a specified number of running instances at all times because they want to maintain the current problems and Amazon EC2 autoscaling performance with periodic health checks on running it.

Auditory.

When Amazon EC2 autoscaling finds an unhealthy.

Terminate.

And that.

Wait for the new.

If you choose the manual feeling useless by only the change in the maximum and minimum for desired capacity of the Auto Scaling group with scheduled scaling, scaling actions are performed automatically as a function of state.

This is useful for disable workload when you know exactly where to increase or decrease the number of instances in the roof over a day week.

You can configure dynamic on demand scaling and the more advanced weight scale your resources where you've defined parameters that control the scaling.

Using scaling.

Feeling event for on demand Feeling can use Amazon Cloud Watch to check with PQ utilization and only.

Threshold.

This option is useful for scaling and response.

When you don't know when these.

Finally, you can use Amazon EC2 autoscaling with a WS implements between two scaling where you're.

Say something.

Predictive feeling using native and is collected through.

And here is the further informed by billions of data points that have gone from our own optimism.

WSN uses well trained machine learning models to predict your expected traffic and you can see including daily and weekly tasks.

This model needs a beautiful day, historical data to start making predictions and is reevaluated every 24 hours to create a forecast for the next 40.

The prediction process.

Plants that can drives water more groups automatically to scaling these distributions.

One common configuration for implementing dynamic scaling is to create a cloud watch alarm that is based on performance.

Message from your PC.

Lower back.

When a performance threshold is reached, its Cloud Watch alarm triggers an automatic scaling event that either scales out or scales in EQ.

One of the things how it works, Consider the example.

First, you create an Amazon Cloud watch along and monitor CPU utilization.

And run automatic scaling policies with the average keeps utilization across the entire fleet goes above 60.

For more than 5 minutes.

Next, Amazon EC2 Autoscaling adds a new EC2 instance.

You know in groups based on the log configuration that you create, after the new instance is padded, Amazon EC2 Autoscaling makes a call the Elastic Load Balancing to register the new execute.

Auto scale.

Finally, Elastic Load Balancing performs the required health checks and starts contributing traffic to that input.

Elastic Load Balancing then routes traffic between EC2.

So far you've learned about scaling easy 2 instances with Amazon EC2 Autoscaling.

You also learned that you can use Amazon EC2 Autoscaling, AWS Autoscaling to perform predictive.

A WS Auto Scaling is a separate service that monitors your application.

Automatically adjust the message.

Predictable performance at the lowest possible.

This service provides a simple, powerful user interface that enables you to build scaling plans for resources such as Amazon Instances and Spot.

Amazon Elastic Container for Amazon Dynamo DB tables and indexes and Amazon Aurora Replica.

If you're only using Amazon EC Auto Scaling dynamically failure EC2 instances, you can now use it with a WS Auto Scaling to scale additional resources for other native UI services.

Thanks for watching, I'll see you in the next video.

It's now time to review the module and wrap up with the discussion of a practice certification exam.

In summary, you learn how to distribute traffic across Amazon Institute institutes using Elastic Load Balancer.

Identify how Amazon Cloud Watch enables you to monitor AWS resources and applications in real time.

You can explain how Amazon EC2 Autoscaling launches and releases servers and response to workload changes.

You can perform scaling and load balancing tests to improve an architecture.

Great work, Let's get you a look at that samples for the future plan.

Which service would you use to send alerts based on Amazon Cloud Watch?

In this example, the keywords to recognize are Send Alerts and Amazon Cloud Watch alarms.

Based on the keywords that we identified, we can look at the correct answer as a Amazon Simple Notification Service.

When you create a problem, like a lot, you configure it to send a notification to the Amazon SNS topic.

We're able to eliminate a WS Cloud Trail because while you can configure Cloud Trails to send notifications, test has coverage.

The keywords are based on cloud watch alarms.

We can eliminate eight others.

Trusted Advisor, which is a service that looks at your.

WS Environment and provides real time guidance to help you provision your resources according to best friends.

Finally, we can eliminate Amazon Route 53 which is a DNS website.

Thanks for watching.