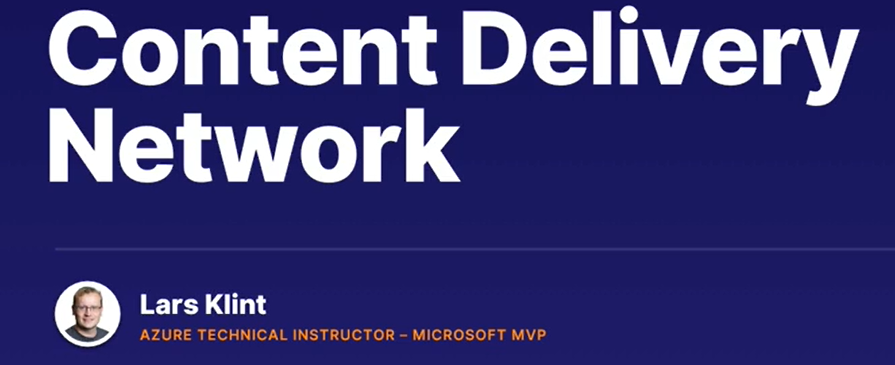
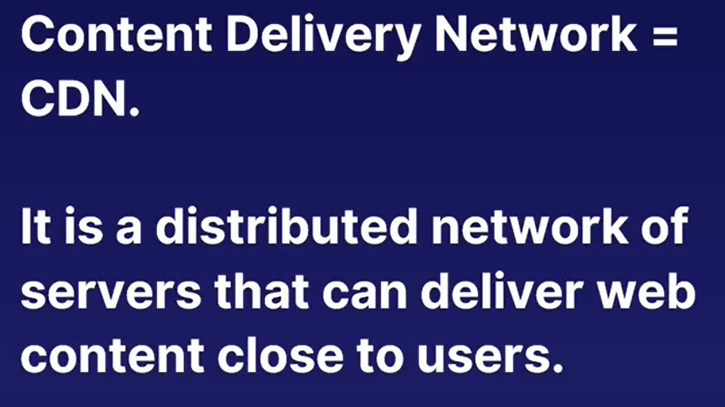
1. When it comes to world domination of your leisure lava rental business, you want your users to load your site and resources as fast as possible, and if your application is hosted in Australia, and you have users from Paris, then they will wait a while to get that information.
2. The fix is using a content delivery network.



1. A content delivery network is most often called a CDN. It's a distributed network of servers that can deliver web content close to users.
2. CDNs store cached content on what is called edge servers in locations that are close to end users to minimize latency.



1. Let me explain.
2. Here's the internet!
3. Well, it looks like a table, but it's definitely the internet. and here is the data center and service that is hosting your online store selling kits for making frozen yogurt.
4. You have customers from here, all over the world, but you have just the one location for your online resources this data center here.
5. So what happens when your customers from far away, like up here, wants to load your website?
6. Well, it takes longer. They got to send their dog, or their data, all the way down here and back again.
7. And that's just to get the pricing on the Japanese apple flavor of frozen yogurt.
8. And sometimes it takes a lot longer - so long they might eventually give up.
9. Instead, what if they only had to request the data from the closest Azure data center to them?
10. That is what the CDN does. It places copies of the data of your application on what is known as edge nodes. So here are three edge nodes - the edge of the internet closest to your users - each of the three users.
11. So now your users only have to go and get the data from close by, and everyone's happy.
12. "But wait," I hear you say.
13. "Lars, what if my website changes?
14. I might update my prices, and I want everyone to see that."
15. Well, that is where caching and data invalidation comes in.
16. So each piece of data on the edge node - which are these here - has an expiry date. Usually it is some hours, depending on how often the data might change.
17. So if your user goes to the edge node - sends the dog - to get the data, and it has expired, meaning they have to get a fresh set of data, then they can request it from the master copy.
18. And then they get a new cached version, like that. They take that back, that goes on the edge node, and then the user sees that.
19. That's how a CDN works and makes your web presence global. A CDN gives you global superpowers at a whim using leverage of the Azure infrastructure as if it was your own.

A picture containing person, indoor

Description automatically generated

1. The benefits of using Azure CDN to deliver website assets include better performance and improved user experience for end users, especially when using applications in which multiple round trips are required to load content.
2. Large scaling to better handle instantaneous high loads, such as the start of a product launch event.
3. A CDN will protect your backend from traffic hitting it as well.
4. Distribution of user requests and serving of content directly from edge servers, so that less traffic is sent to the origin server.

Graphical user interface, text

Description automatically generated

1. Now, I've mentioned two terms in this lecture that could use some clarification.
2. The first is a cache or caching. A cache is a collection of temporary copies of original files, such as a copy of a image.
3. The cache's primary purpose in any computing scenario is to speed things up.
4. For Azure, the cache is placed close to the end user, so the latency, which is the travel time of the data, is less.
5. This makes, for example, websites a lot faster. A cached item can also expire, at which point a new copy is needed.
6. And that brings me to the origin server. This is the original place that your web application is stored. The origin server is the master copy of your files for your application.

A screenshot of a computer

Description automatically generated with medium confidence