Section 1

Hello and welcome to the Apache Kafka. Deep Dive, of course. Apache Kafka is a published and subscribe messaging system, which in general serves as a hub for producers and consumers. It accepts many different applications and input streams, which makes it very versatile when talking about other methods of dealing with distributed messaging systems in this course will go through how to install your own Kafka cluster. We'll talk a lot about producers and consumers, and we'll talk about the main method of labeling or groups of messages, which is called a topic. All that more in this Apache Kafka deep die. Of course, I'm excited to get started with you, so let's dive right in.

Section 2

Hello again. I just wanted to make this short video to help you get prepared for this section. I'll be putting a video like this at the beginning of each section in this course, just to get you prepared for what's coming and help you understand the section and the lessons better. So, this this section is called at first glance, and we're really taking a look at what casket is and the components of it and what it's trying to solve.

0:00:30

outlined\_flagFlag

Kafka is a published subscribe messaging

0:00:33

outlined\_flagFlag

system, and the real benefit is

0:00:36

outlined\_flagFlag

is that you can take different applications and plug

0:00:39

outlined\_flagFlag

into it almost like a hub, and you're able

0:00:42

outlined\_flagFlag

to process millions of messages

0:00:45

outlined\_flagFlag

all in real time, which is really,

0:00:47

outlined\_flagFlag

really great for, ah, the efficiency

0:00:50

outlined\_flagFlag

of that. But also it's great for, you know,

0:00:54

outlined\_flagFlag

being able to plug basically anything in

0:00:57

outlined\_flagFlag

and have Kafka not really care about

0:01:00

outlined\_flagFlag

what components are. There is just handling it

0:01:03

outlined\_flagFlag

as any other message. So that's what I want

0:01:06

outlined\_flagFlag

you to think about before we get into. This section of the

0:01:09

outlined\_flagFlag

first section is kind of application metrics and

0:01:12

outlined\_flagFlag

how the conflict cluster plugs into all

0:01:14

outlined\_flagFlag

these different things like producers and consumers.

0:01:17

outlined\_flagFlag

So just have that mine. Kafka cluster is

0:01:20

outlined\_flagFlag

a hub that you can plug everything into and

0:01:24

outlined\_flagFlag

is, uh, you know, processing millions

0:01:26

outlined\_flagFlag

of messages, messages at a time. So with that,

0:01:30

outlined\_flagFlag

I'm looking forward to seeing you in the next lesson. So we'll see you then.

Welcome back to the Apache Casca Deep Dive,

0:00:04

outlined\_flagFlag

Go ahead and get started with the basics of Apache

0:00:06

outlined\_flagFlag

Casca and look at the application

0:00:10

outlined\_flagFlag

metrics. So Captain is a published subscribed

0:00:13

outlined\_flagFlag

messaging system, which means that it's used to categorize

0:00:17

outlined\_flagFlag

published messages into classes without

0:00:20

outlined\_flagFlag

knowledge of the subscriber. So unlike the client

0:00:23

outlined\_flagFlag

server model, CAF Co. Allows for the client to post messages

0:00:27

outlined\_flagFlag

even if the server is not running, allowing for better scalability

0:00:30

outlined\_flagFlag

up to millions of messages per second. This allows us

0:00:33

outlined\_flagFlag

to treat CAFTA as a hub for all types

0:00:36

outlined\_flagFlag

of data streams. No longer do you have to create thousands

0:00:39

outlined\_flagFlag

of separate integrations, each with their own protocol and data

0:00:42

outlined\_flagFlag

format, having to manage those over a period

0:00:45

outlined\_flagFlag

of time. You can now do couple of those components

0:00:48

outlined\_flagFlag

and essentially plug them into the Kafka cluster and it's

0:00:52

outlined\_flagFlag

a cluster, so the nodes of the cluster are called brokers,

0:00:55

outlined\_flagFlag

and we'll talk about brokers again in the future lesson.

0:01:00

outlined\_flagFlag

Some of the use cases for using Cap CA is

0:01:03

outlined\_flagFlag

for traditional messaging systems, activity tracking

0:01:07

outlined\_flagFlag

or location tracking. Using I ot devices together

0:01:11

outlined\_flagFlag

logs from applications, stream processing,

0:01:14

outlined\_flagFlag

using the capita streams, a P I the coupling

0:01:17

outlined\_flagFlag

of the components of your application, like databases

0:01:21

outlined\_flagFlag

or even integrated with big data tools like

0:01:23

outlined\_flagFlag

Apache Spark. Or How do A Real World example

0:01:26

outlined\_flagFlag

of how it's being used today is linked in

0:01:29

outlined\_flagFlag

who is tthe e, creator of Capta, uses CAFTA

0:01:32

outlined\_flagFlag

to improve the accuracy of recommendations

0:01:35

outlined\_flagFlag

it gives you for reaching out to your professional

0:01:37

outlined\_flagFlag

connections on Lincoln. In our interactive diagram

0:01:41

outlined\_flagFlag

over on the right hand side of your screen, you'll see

0:01:43

outlined\_flagFlag

producers, which are the A's and those

0:01:46

outlined\_flagFlag

are publishing messages to the Catholic cluster.

0:01:49

outlined\_flagFlag

You'll also see consumers, which is the B.

0:01:52

outlined\_flagFlag

They allow applications to subscribe to a

0:01:55

outlined\_flagFlag

stream of records so they're receiving the messages,

0:01:59

outlined\_flagFlag

along with producers and consumers. There are

0:02:02

outlined\_flagFlag

stream processors and connectors. Stream

0:02:06

outlined\_flagFlag

processors, the ones that have a C on them,

0:02:09

outlined\_flagFlag

allow the application to consume an input stream

0:02:12

outlined\_flagFlag

and also an output stream simultaneously so

0:02:14

outlined\_flagFlag

they serve as essentially a producer and consumer.

0:02:18

outlined\_flagFlag

And then connectors, which are the ones labeled D,

0:02:21

outlined\_flagFlag

allow you to plug in existing applications to provide functionality

0:02:25

outlined\_flagFlag

like a database, for example, Capta is so popular

0:02:28

outlined\_flagFlag

in high demand because of the ability to process

0:02:31

outlined\_flagFlag

so many streams and offer the extent ability it

0:02:33

outlined\_flagFlag

does. Let's continue learning about the basics by clicking

0:02:37

outlined\_flagFlag

on the next button in the lower right of the interactive diagram,

0:02:43

outlined\_flagFlag

and we'll continue talking about messages and ski mus.

0:02:46

outlined\_flagFlag

So go ahead and mark this lesson complete,

0:02:50

outlined\_flagFlag

and we'll see in the next lesson.