**Microservices in NodeJS and React**

**Stephen Grider**

**RabbitMQ Youtube**

**Course content**

Section 1: Fundamental Ideas Around Microservices

1. How to Get Help

2. Course Resources

3. What Is a Microservice?

4. Data in Microservices

Quiz 1: Quiz - Data in Microservices

5. Big Problems with Data

6. Sync Communication Between Services

7. Event-Based Communication

8. A Crazy Way of Storing Data

9. Pros and Cons of Async Communication

10. Important - Optional Boilerplate

11. App Overview

12. Project Setup

13. Posts Service Creation

14. Testing the Posts Service

15. Implementing a Comments Service

16. Quick Comments Test

17. Note on the React App

18. Addressing Default Export and ReactDom.render Warnings

19. React Project Setup

20. Building Post Submission

21. Handling CORS Errors

22. Fetching and Rendering Posts

23. Creating Comments

24. Dising Comments

25. Completed React App

26. Request Minimization Strategies

27. An Async Solution

28. Common Questions Around Async Events

29. Event Bus Overview

30. Important Note about Node and Unhandled Promise Rejections

31. A Basic Event Bus Implementation

32. Emitting Events

33. Emitting Comment Creation Events

34. Receiving Events

35. Creating the Data Query Service

36. Parsing Incoming Events

37. Using the Query Service

38. Adding a Simple Feature

39. Issues with Comment Filtering

40. A Second Approach

41. How to Handle Resource Updates

42. Creating the Moderation Service

43. Adding Comment Moderation

44. Reminder about Error Catching

45. Handling Moderation

46. Updating Comment Content

47. A Quick Test

48. Rendering Comments by Status

49. Dealing with Missing Events

50. Required Error Handling Update for Query Service

51. Implementing Event Sync

52. Event Syncing in Action

Section 3: Running Services with Docker

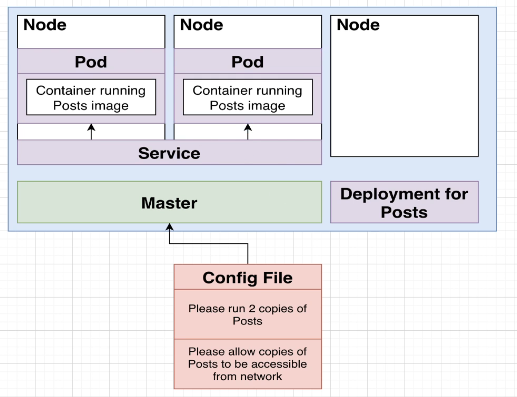
Section 4: Orchestrating Collections of Services with Kubernetes

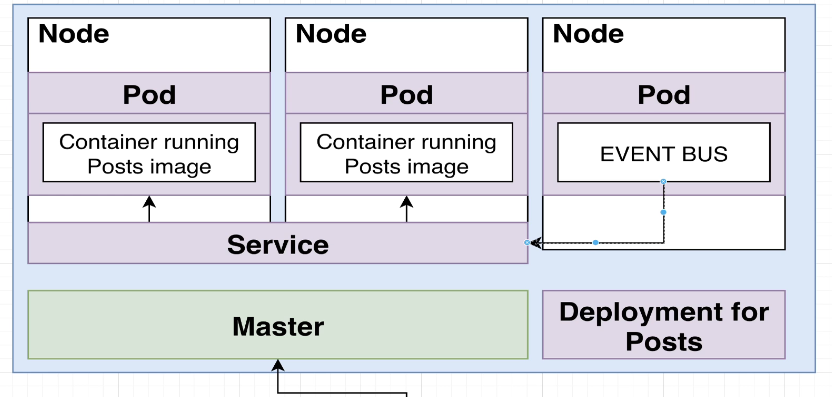
61. Warning on Docker Desktop for Linux

62. Installing Kubernetes

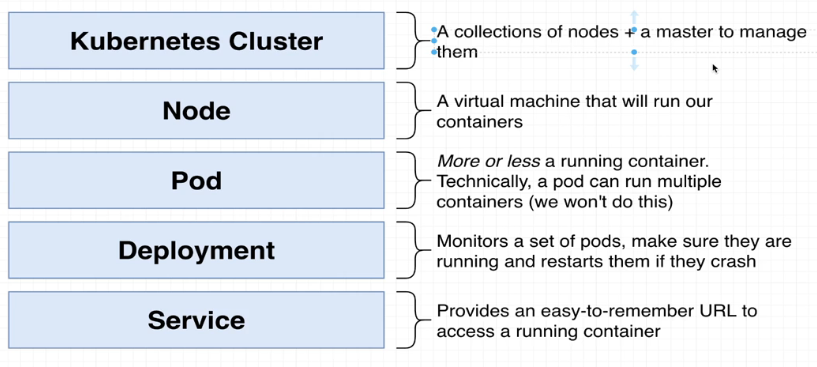
63. IMPORTANT Note for Minikube and MicroK8s Users

64. A Kubernetes Tour

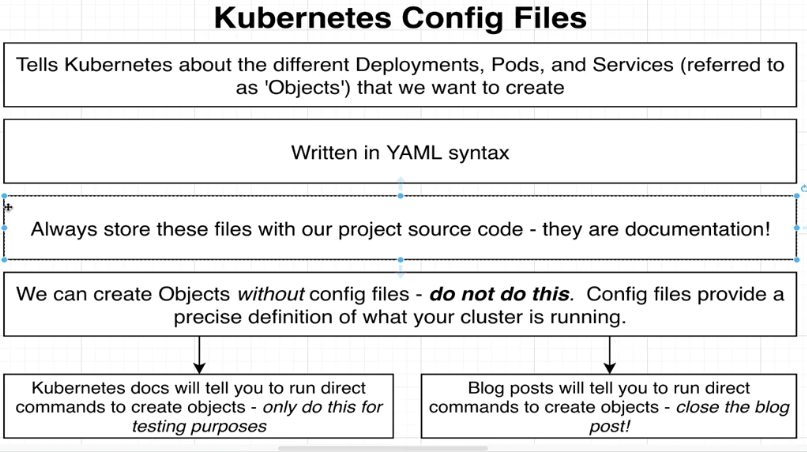




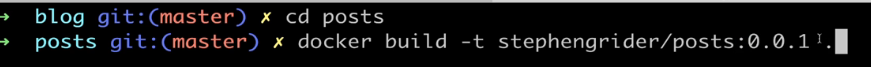
65. Important Kubernetes Terminology



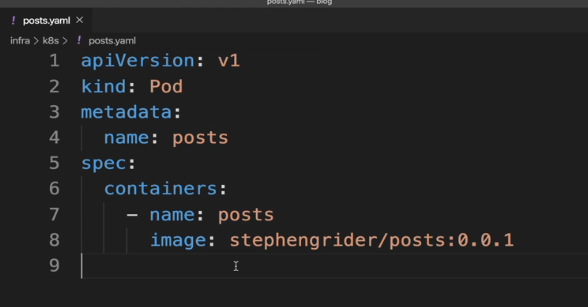
66. Notes on Config Files

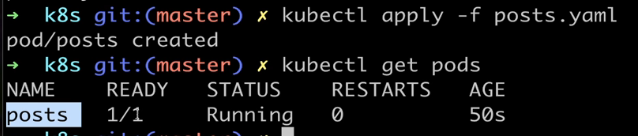


67. Creating a Pod



Infra 🡪 k8s 🡪 posts.yaml





68. ErrImagePull, ErrImageNeverPull and ImagePullBackoff Errors

If your pods are showing *ErrImagePull*, *ErrImageNeverPull*, or *ImagePullBackOff* errors after running kubectl apply, the simplest solution is to provide an imagePullPolicy to the pod.

First, run kubectl delete -f infra/k8s/

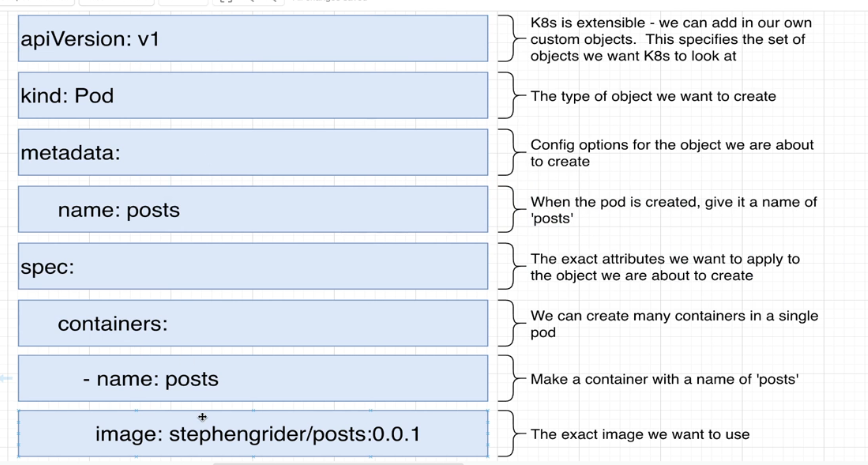
Then, update your pod manifest:

1. spec:
2. containers:
3. - name: posts
4. image: cygnet/posts:0.0.1
5. imagePullPolicy: Never

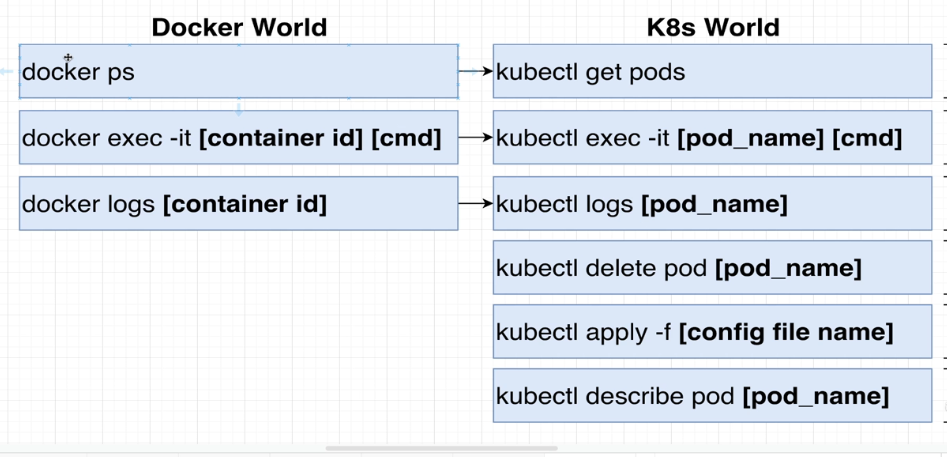
Then, run kubectl apply -f infra/k8s/

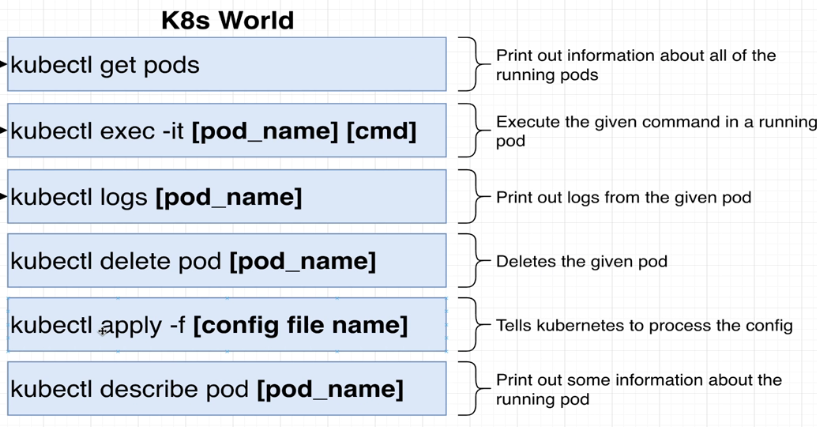
This will ensure that Kubernetes will use the image built locally from your image cache instead of attempting to pull from a registry.

69. Understanding a Pod Spec



70. Common Kubectl Commands





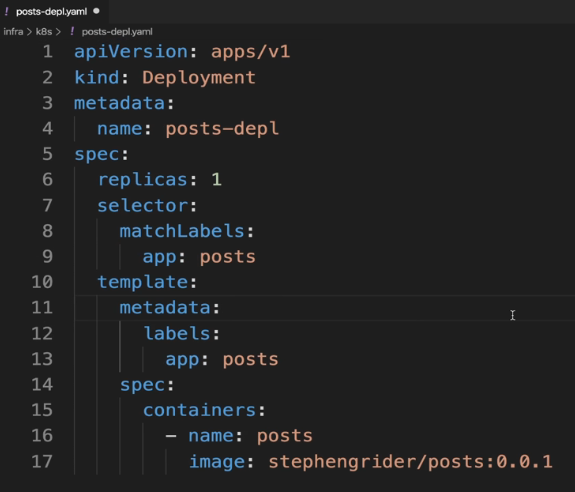
71. A Time-Saving Alias

72. Introducing Deployments

Deployment manages set of pods

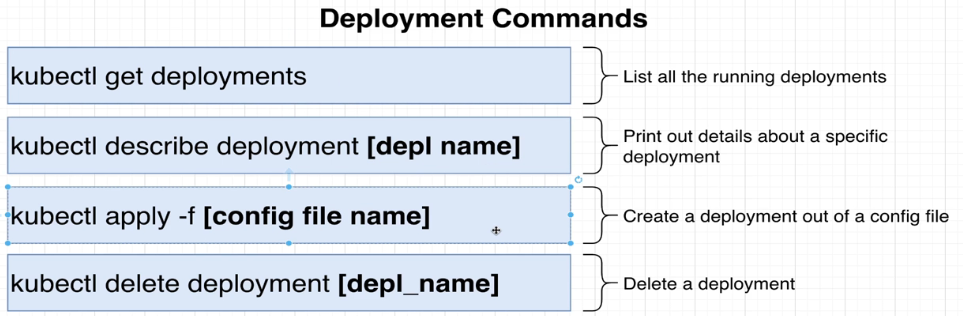
73. Creating a Deployment

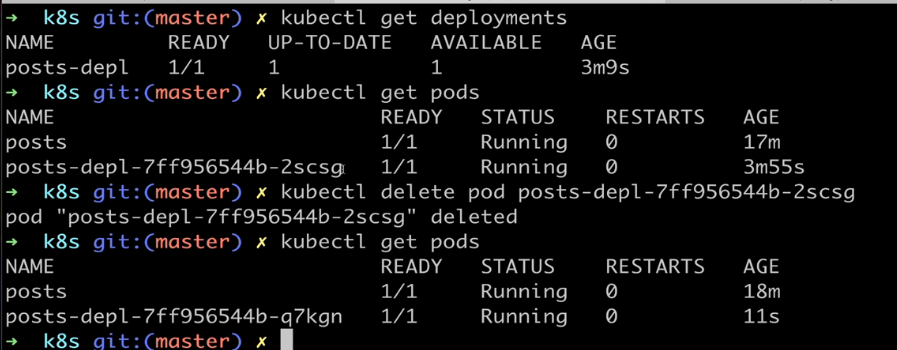
Delete individual pod file

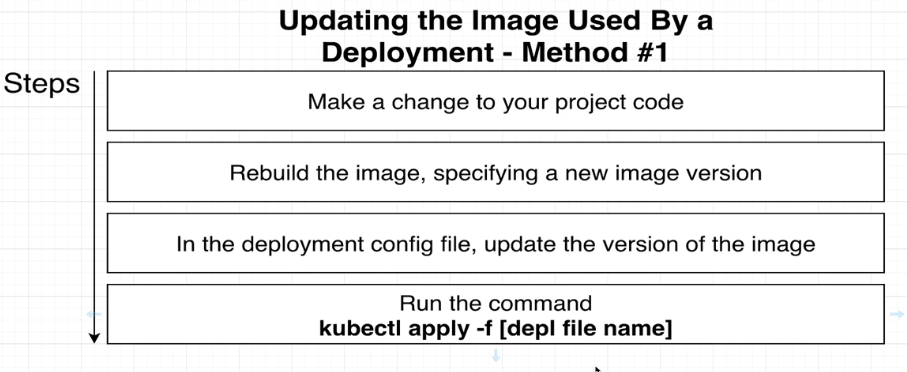




74. Common Commands Around Deployments

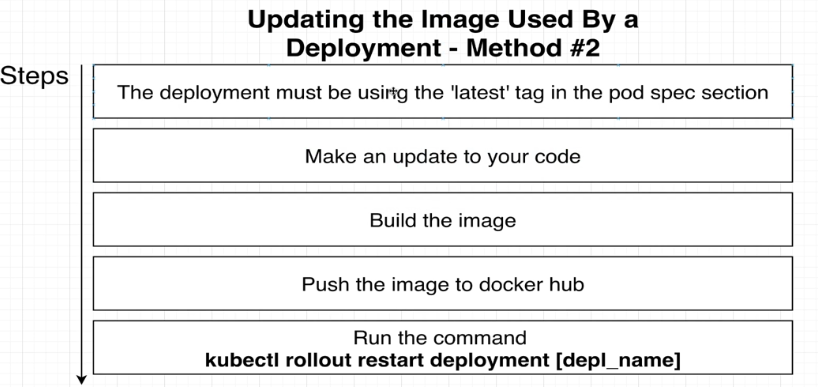




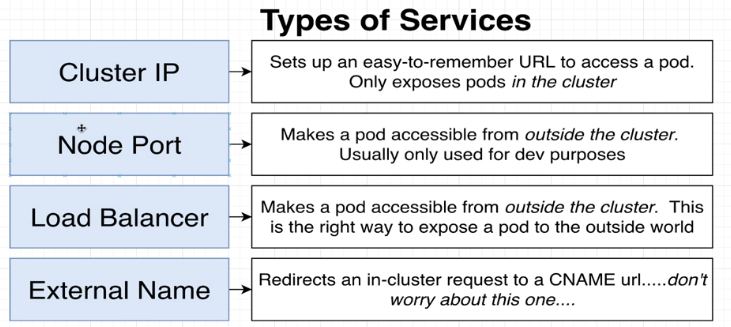
75. 

May introduce errors as deployment files go large

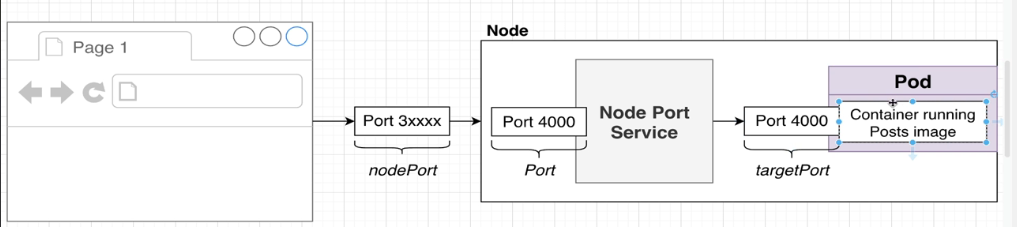
76. Preferred Method for Updating Deployments

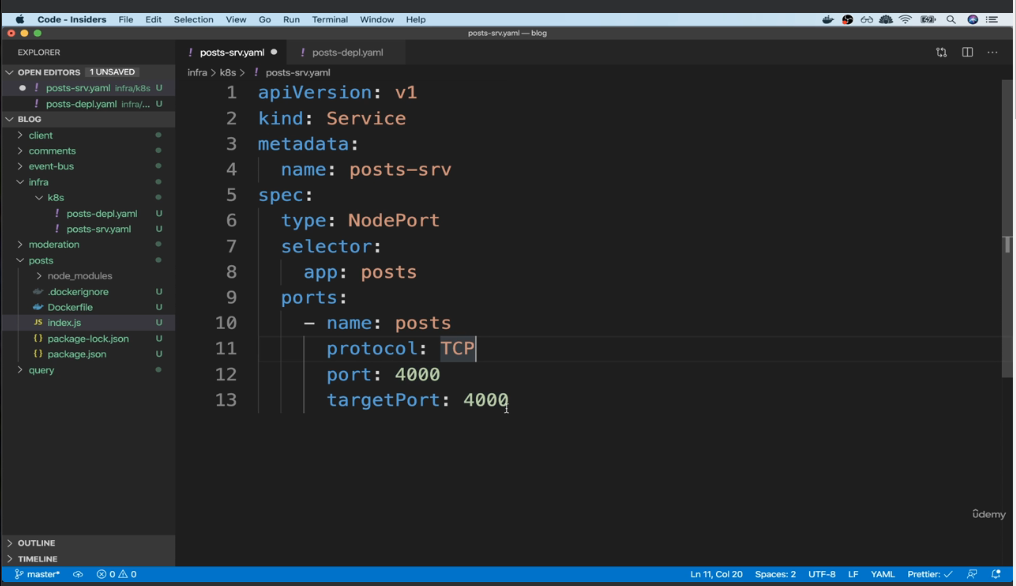


77. Networking With Services



78. Creating a NodePort Service

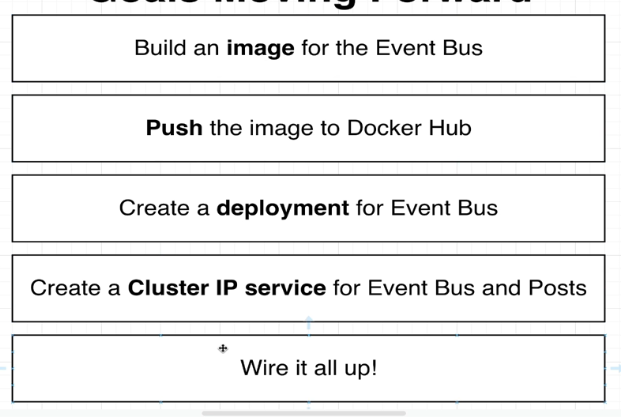




79. Accessing NodePort Services

80. Setting Up Cluster IP Services

81. Building a Deployment for the Event Bus



82. Adding ClusterIP Services

83. How to Communicate Between Services

84. Updating Service Addresses

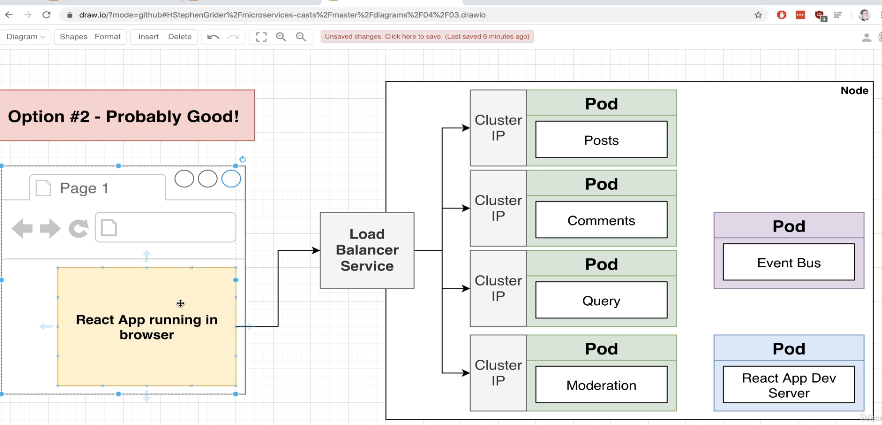
85. Verifying Communication

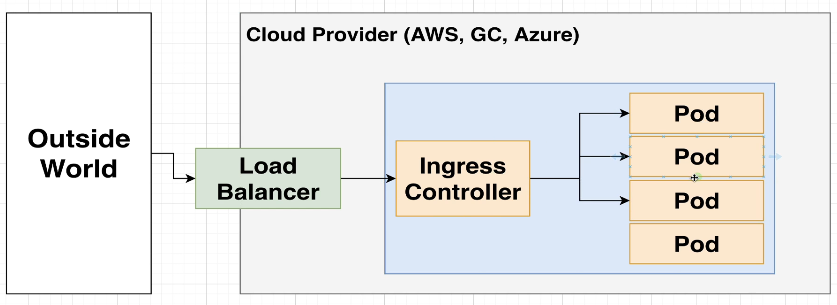
86. Adding Query, Moderation and Comments

87. Testing Communication

88. Load Balancer Services

89. Load Balancers and Ingress





An Ingress controller abstracts away the complexity of Kubernetes application traffic routing and provides a bridge between Kubernetes services and external ones. Kubernetes Ingress controllers: Accept traffic from outside the Kubernetes platform, and load balance it to pods (containers) running inside the platform.

<https://www.baeldung.com/ops/kubernetes-ingress-vs-load-balancer>

**1. Introduction**

For many software applications, communicating with external services is necessary to complete their tasks. Whether sending messages or consuming APIs, **most applications rely on other systems to function properly**.

However, as more and more users move their applications into Kubernetes, providing secure and reliable access to them becomes more challenging. Navigating through various deployments and services can make it difficult for network traffic to reach the right place.

Luckily, there are a number of different mechanisms to help manage network traffic and ensure requests get to their desired destination inside a cluster. In this tutorial, we will look at two of these mechanisms: ingresses and load balancers.

**2. Workloads and Services**

Before we discuss either of these, we must first take a step back and look at how applications are deployed and managed in Kubernetes.

**2.1. Workloads and Pods**

We start by packaging our applications into docker images. Those docker images are then used to create one of the pre-defined [workload](https://kubernetes.io/docs/concepts/workloads/) types, for example:

* *ReplicaSet*: Ensures a minimum number of *pods* are available at all times
* *StatefulSet*: Provides a predictable and unique ordering when increasing or decreasing the number of *pods*
* *DaemonSet*: Ensures a specific number of *pods* are running on some or all *nodes* at all times

**All of these workloads create one or more *pods* being deployed in the cluster**. A *pod* is the smallest deployable unit that we can use in Kubernetes. It essentially represents an application running somewhere in the cluster.

By default, every *pod* has a unique IP that is accessible by other members of the cluster. However, **accessing a *pod* using its IP address is not a good practice** for multiple reasons.

First of all, it isn’t easy to know what IP will be assigned to a *pod* ahead of time. This makes it nearly impossible to store IP information in a configuration where other applications can access it. Second, many workloads create multiple pods — in some cases, dynamically. This means, at any point in time, we may not know how many pods are running for a given application.

And finally, *pods* are nonpermanent resources. They are meant to start and stop over time, and each time this happens, it’s more than likely they will get a new IP address.

For all of these reasons, communicating with pods using their IP address is a bad idea. Instead, we should use *services*.

**2.2. Services**

**A Kubernetes *service* is an abstraction that exposes a group of pods as a network service**. The *service* handles all the complexity of identifying running pods and their IP addresses. Every *service* gets a unique URL that is accessible across the cluster. So instead of using IPs to communicate, pods simply need to use the provided service URLs.

Let’s look at a sample service definition:

apiVersion: v1

kind: Service

metadata:

name: api-service

spec:

selector:

app: api-app

ports:

- protocol: TCP

port: 80

targetPort: 8080Copy

This will create a *service* named *api-service* in the cluster. This *service* is bound to any pod running the application *api-app*, regardless of how many of those *pods* exist or where they are running. And, as new *pods* of this type start-up, the *service* will automatically discover them.

**Using a *service* is a great start to decoupling pods from the applications that use them**. But, on their own, they don’t always achieve our desired goal. That’s where ingresses and load balancers come in.

**3. Ingress**

By default, a Kubernetes *service* is private to the cluster. This means only applications inside the cluster can access them. There are a number of ways around this, and one of the best is an *ingress*.

In Kubernetes, **an *ingress* lets us route traffic from outside the cluster to one or more *services* inside the cluster**. Typically, the ingress works as a single point of entry for all incoming traffic.

An ingress receives a public IP, meaning it is accessible outside the cluster. Then, using a set of rules, it forwards all of its traffic to an appropriate *service*. In turn, that *service* will send the request to a *pod* that can actually handle the request.

There are a few things to keep in mind when creating an *ingress.* First, **they are designed to handle web traffic (HTTP or HTTPS)**. While it is possible to use an *ingress* with other types of protocols, it typically requires extra configuration.

Second, an *ingress* can do more than just routing. Some other use cases include load balancing and SSL termination.

Most importantly, **the *ingress* object by itself does not actually do anything**. So, for an *ingress* to actually do anything, we need to have an *ingress controller* available.

**3.1. Ingress Controllers**

As with most Kubernetes objects, an *ingress* requires an associated controller to manage it. However, while Kubernetes provides controllers for most objects like *deployments* and *services*, **it does not include an *ingress controller* by default**. Therefore, it is up to the cluster administrator to ensure an appropriate controller is available.

Most cloud platforms provide their own *ingress controllers*, but there are also plenty of open-source options to choose from. Perhaps the most popular is the [nginx ingress controller](https://www.nginx.com/products/nginx-ingress-controller/), which is built on top of the popular web server of the same name.

Let’s check out a sample configuration using the nginx *ingress controller*:

apiVersion: networking.k8s.io/v1

kind: Ingress

metadata:

name: ingress-example

annotations:

nginx.ingress.kubernetes.io/rewrite-target: /

spec:

ingressClassName: nginx-example

rules:

- http:

paths:

- path: /api

pathType: Prefix

backend:

service:

name: api-service

port:

number: 80Copy

In this example, we create an *ingress* that routes any request that starts with */api* to a Kubernetes *service* named *api-service*.

Notice the *annotations* field contains values that are specific to nginx. Because all *ingress controllers* use the same API object, we typically use the annotations field to pass specific configurations into the *ingress controller*.

There are dozens of [available *ingress controllers*](https://kubernetes.io/docs/concepts/services-networking/ingress-controllers/) in the Kubernetes ecosystem, and covering them all is well beyond the scope of this article. However, **because they use the same API object, they share some common features**:

* Ingress Rules: the set of rules that define how to route traffic to a specific service (typically based on URL or hostname)
* Default Backend: A default resource that handles traffic that does not match any rule
* TLS: A secret that defines a private key and certificate to allow TLS termination

Different ingress controllers build on these concepts and add their own functionality and features.

**4. Load Balancers**

[Load Balancers](https://kubernetes.io/docs/concepts/services-networking/service/#loadbalancer) in Kubernetes have quite a bit of overlap with *ingresses*. This is because they are primarily used to expose *services* to the internet, which, as we saw above, is also a feature of *ingresses*.

**However, load balancers have different characteristics from *ingresses***. Rather than a standalone object like an *ingress*, a load balancer is just an extension of a *service*.

Let’s see a simple example of a service with a load balancer:

apiVersion: v1

kind: Service

metadata:

name: api-service

spec:

selector:

app: api-app

ports:

- protocol: TCP

port: 80

targetPort: 8080

type: LoadBalancerCopy

Just like our earlier service example, here, we’re creating a *service* that routes traffic to any *pod* running our API application. In this case, we’ve included a *LoadBalancer* configuration.

For this to work, **the cluster must be running on a provider that supports external load balancers**. All of the major cloud providers support external load balancers using their own resource types:

* AWS uses a [**Network Load Balancer**](https://docs.aws.amazon.com/eks/latest/userguide/aws-load-balancer-controller.html)
* GKE also uses a [**Network Load Balancer**](https://cloud.google.com/load-balancing/docs/network)
* Azure uses a [**Public Load Balancer**](https://docs.microsoft.com/en-us/azure/aks/load-balancer-standard)

Just like we saw with different *ingress controllers*, different load balancer providers have their own settings. Typically, we manage these settings directly with either a CLI or tool specific to the infrastructure rather than with YAML. Different load balancer implementations will also provide additional features such as SSL termination.

Because load balancers are defined per *service*, **they can only route to a single *service***. This is different from an *ingress*, which has the ability to route to multiple *services* inside the cluster.

Also, keep in mind that regardless of the provider, **using an external load balancer will typically come with additional costs**. This is because, unlike *ingresses* and their controllers, external load balancers **exist outside of the Kubernetes cluster**. Because of this, most cloud providers will charge for the additional resource usage beyond the cluster itself.

**5. Conclusion**

In this article, we’ve looked at some core concepts from Kubernetes, including *deployments*, *services*, and *ingresses*. Each of these objects plays a key role in routing network traffic between various *pods.*

While *ingresses* and load balancers have a lot of overlap in functionality, they behave differently. **The main difference is *ingresses* are native objects inside the cluster that can route to multiple *services*, while load balancers are external to the cluster and only route to a single *service.***

90. Important - DO NOT SKIP - Ingress Nginx Installation Info

91. Installing Ingress-Nginx

92. Ingress v1 API Required Update + pathType Warning

93. Writing Ingress Config Files

94. Important Note About Port 80

95. Hosts File Tweak

96. Important Note to Add Environment Variable

97. Deploying the React App

98. Unique Route Paths

99. Final Route Config

100. Introducing Skaffold

101. Skaffold API version Update

102. Skaffold Setup

103. First Time Skaffold up

104. A Few Notes on Skaffold

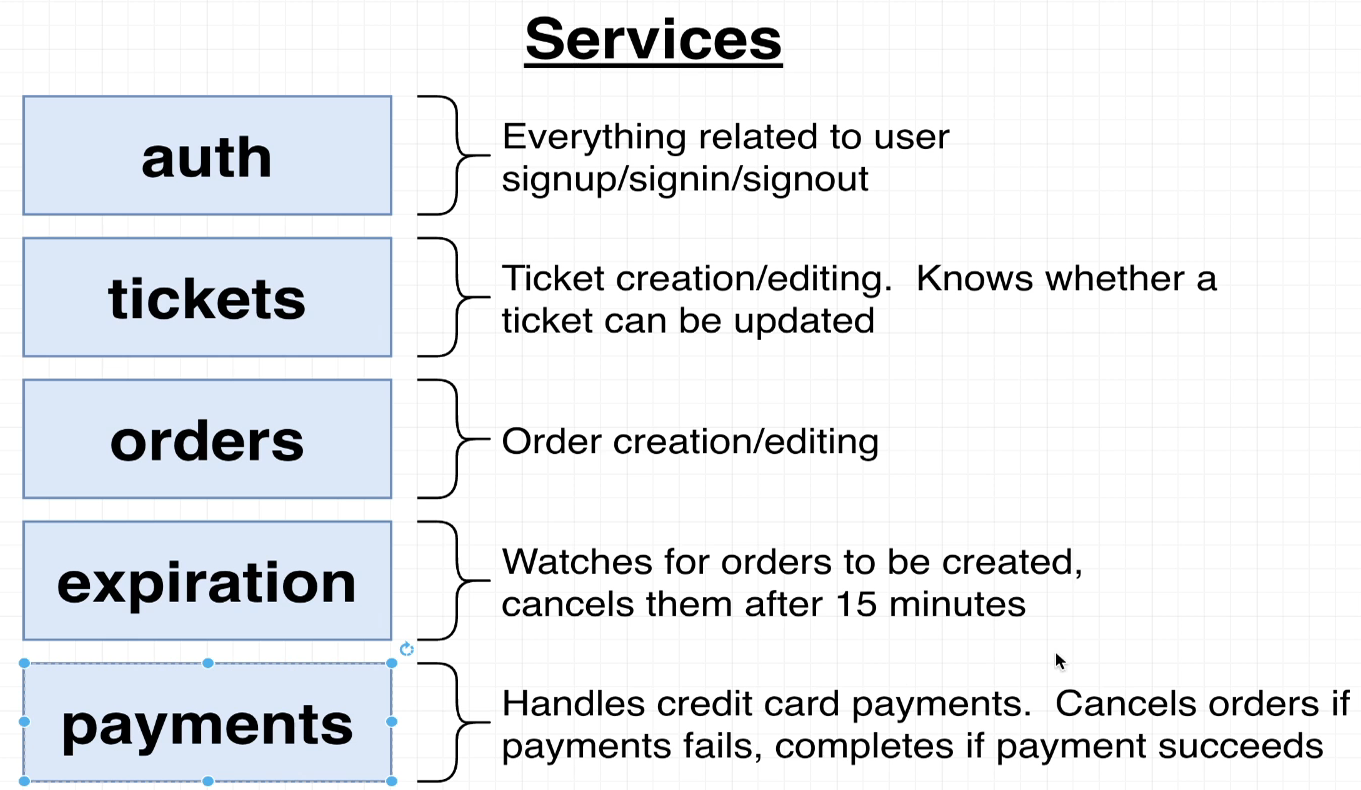
# Section 5: Architecture of Multi-Service Apps

105. Big Ticket Items

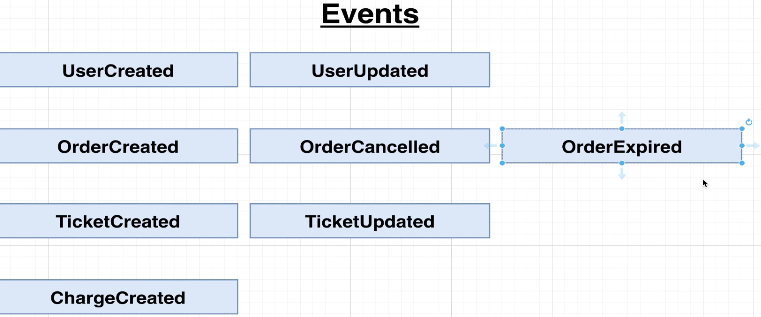
106. App Overview

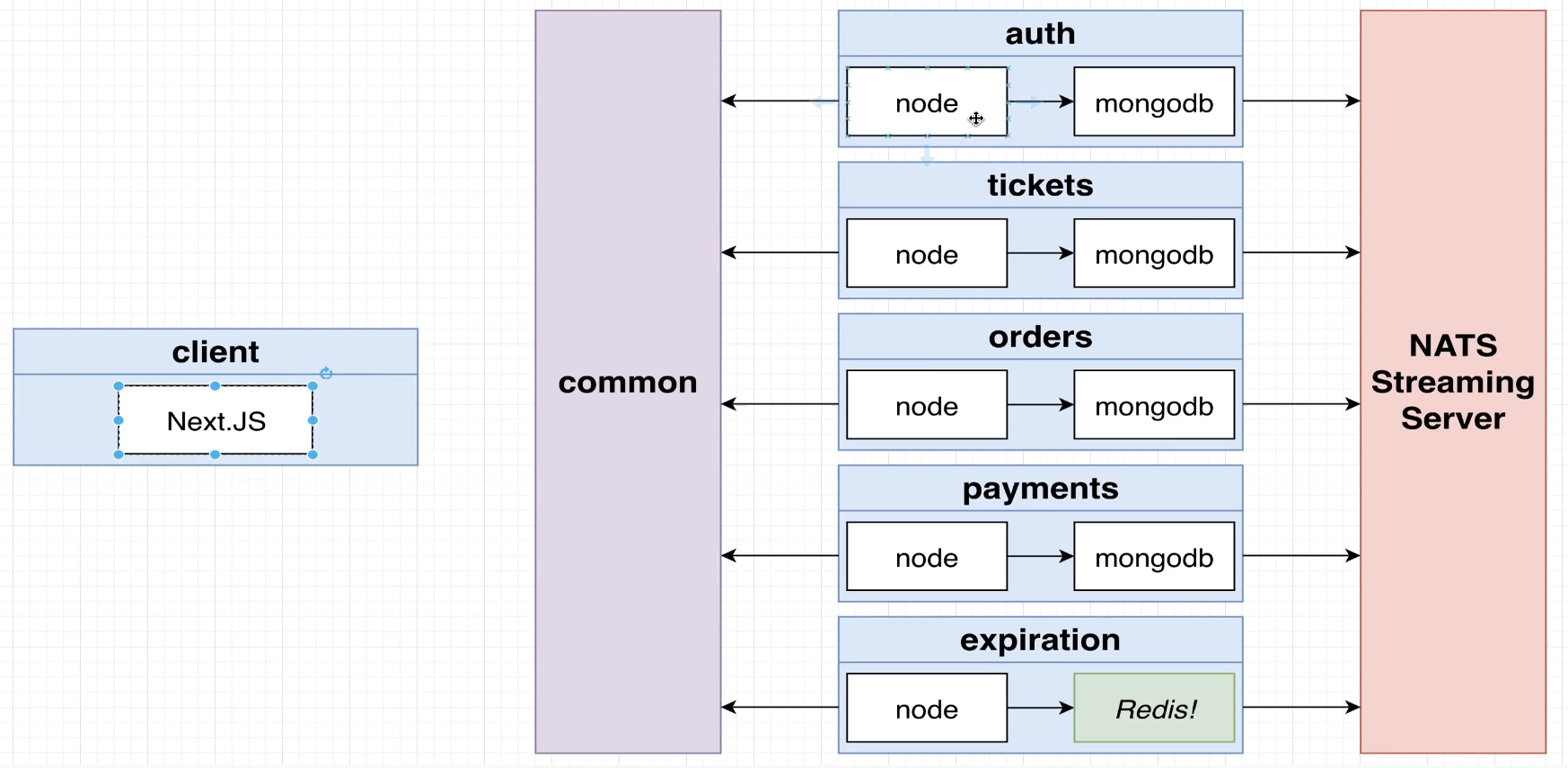
107. Resource Types

108. Service Types



109. Events and Architecture Design





110. Note on Typescript

111. Auth Service Setup

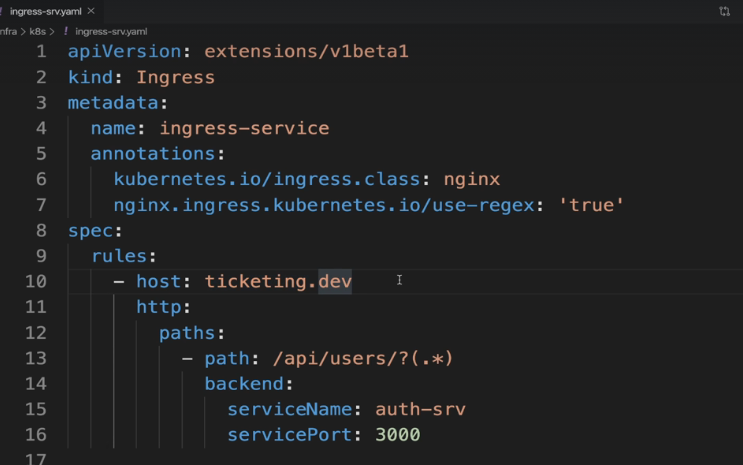
112. Auth K8s Setup

113. Adding Skaffold

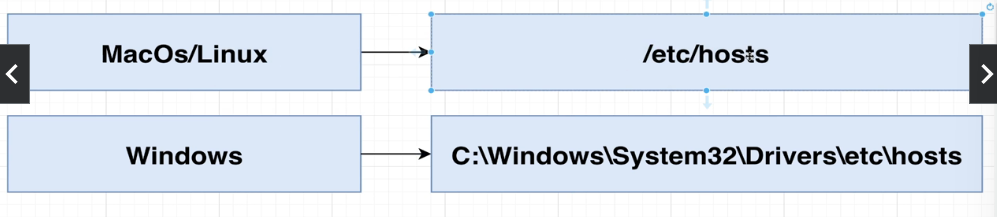
114. Note on Code Reloading

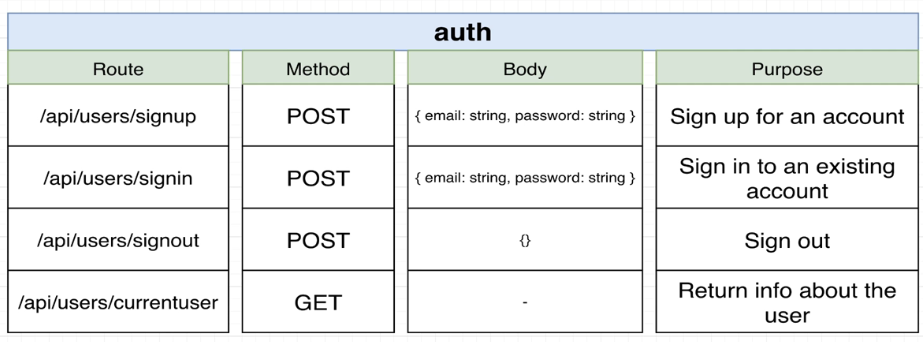
115. Ingress v1 API Required Update

116. Ingress-Nginx Setup



117. Hosts File and Security Warning





Section 6: Leveraging a Cloud Environment for Development

118. Note on Remote Development

119. Remote Dev with Skaffold

120. Free Google Cloud Credits

121. Google Cloud Initial Setup

122. Kubernetes Cluster Creation

123. Kubectl Contexts

124. Initializing the GCloud SDK

125. Installing the GCloud Context

126. Updating the Skaffold Config

127. More Skaffold Updates

128. Creating a Load Balancer

129. Final Config and Test

Section 7: Response Normalization Strategies

130. Creating Route Handlers

131. Scaffolding Routes

132. Adding Validation

133. Handling Validation Errors

134. Postman HTTPS Issues

135. Surprising Complexity Around Errors

136. Other Sources of Errors

137. Solution for Error Handling

138. Building an Error Handling Middleware

139. Communicating More Info to the Error Handler

140. Encoding More Information In an Error

141. Subclassing for Custom Errors

142. Determining Error Type

143. Property 'param' does not exist on type 'AlternativeValidationError'

144. Converting Errors to Responses

145. Moving Logic Into Errors

146. serializeErrors' not assignable to the same property in base type 'CustomError'

147. Verifying Our Custom Errors

148. Final Error Related Code

149. How to Define New Custom Errors

150. Uh Oh... Async Error Handling

Section 8: Database Management and Modeling

151. Creating Databases in Kubernetes

152. Connecting to MongoDB

153. Understanding the Signup Flow

154. Getting TypeScript and Mongoose to Cooperate

155. Creating the User Model

156. Type Checking User Properties

157. Adding Static Properties to a Model

158. Defining Extra Document Properties

159. What's That Angle Bracket For?

160. User Creation

161. Proper Error Handling

162. Note on Password Hashing

163. Reminder on Password Hashing

164. Adding Password Hashing

165. Comparing Hashed Password

166. Mongoose Pre-Save Hooks

Section 9: Authentication Strategies and Options

167. Fundamental Authentication Strategies

168. Huge Issues with Authentication Strategies

169. So Which Option?

* Lecture incomplete. Progress cannot be changed for this item.

170. Solving Issues with Option #2

171. Reminder on Cookies vs JWT's

172. Microservices Auth Requirements

173. Issues with JWT's and Server Side Rendering

174. Cookies and Encryption

175. Adding Session Support

176. Generating a JWT

177. JWT Signing Keys

178. Securely Storing Secrets with Kubernetes

179. Creating and Accessing Secrets

180. Accessing Env Variables in a Pod

181. Common Response Properties

182. Formatting JSON Properties

183. The Signin Flow

184. Common Request Validation Middleware

185. Sign In Logic

186. Quick Sign In Test

187. Current User Handler

188. Returning the Current User

189. Signing Out

190. Creating a Current User Middleware

191. Augmenting Type Definitions

192. Requiring Auth for Route Access

Section 10: Testing Isolated Microservices

193. Scope of Testing

194. Testing Goals

195. Testing Architecture

196. Index to App Refactor

197. Replacing --only=prod Install Flag

198. A Few Dependencies

199. Required MongoMemoryServer Updates

1min

* Lecture incomplete. Progress cannot be changed for this item.

200. Test Environment Setup

8min

Resources

* Lecture incomplete. Progress cannot be changed for this item.

201. Our First Test

6min

Resources

* Lecture incomplete. Progress cannot be changed for this item.

202. An Important Note

2min

* Lecture incomplete. Progress cannot be changed for this item.

203. Testing Invalid Input

5min

Resources

* Lecture incomplete. Progress cannot be changed for this item.

204. Requiring Unique Emails

2min

Resources

* Lecture incomplete. Progress cannot be changed for this item.

205. Changing Node Env During Tests

6min

Resources

* Lecture incomplete. Progress cannot be changed for this item.

206. Tests Around Sign In Functionality

6min

Resources

* Lecture incomplete. Progress cannot be changed for this item.

207. Testing Sign Out

5min

Resources

* Lecture incomplete. Progress cannot be changed for this item.

208. Issues with Cookies During Testing

5min

Resources

* Lecture incomplete. Progress cannot be changed for this item.

209. Easy Auth Solution

3min

Resources

* Lecture incomplete. Progress cannot be changed for this item.

210. globalThis has no index signature TS Error

1min

* Lecture incomplete. Progress cannot be changed for this item.

211. Auth Helper Function

7min

* Lecture incomplete. Progress cannot be changed for this item.

212. Testing Non-Authed Requests

2min

Resources

Section 11: Integrating a Server-Side-Rendered React App

0 / 40 | 3hr 1min0 of 40 lectures completed3hr 1min

* Lecture incomplete. Progress cannot be changed for this item.

213. ing the React App

2min

* Lecture incomplete. Progress cannot be changed for this item.

214. Reminder on Server Side Rendering

4min

* Lecture incomplete. Progress cannot be changed for this item.

215. Suggestion Regarding a Default Export Warning

1min

* Lecture incomplete. Progress cannot be changed for this item.

216. Basics of Next JS

5min

* Lecture incomplete. Progress cannot be changed for this item.

217. Building a Next Image

4min

Resources

* Lecture incomplete. Progress cannot be changed for this item.

218. Running Next in Kubernetes

10min

Resources

* Lecture incomplete. Progress cannot be changed for this item.

219. Small Update for Custom Webpack Config

1min

* Lecture incomplete. Progress cannot be changed for this item.

220. Note on File Change Detection

4min

Resources

* Lecture incomplete. Progress cannot be changed for this item.

221. Adding Global CSS

5min

Resources

* Lecture incomplete. Progress cannot be changed for this item.

222. Adding a Sign Up Form

3min

Resources

* Lecture incomplete. Progress cannot be changed for this item.

223. Handling Email and Password Inputs

4min

Resources

* Lecture incomplete. Progress cannot be changed for this item.

224. Successful Account Signup

5min

Resources

* Lecture incomplete. Progress cannot be changed for this item.

225. Handling Validation Errors

7min

Resources

* Lecture incomplete. Progress cannot be changed for this item.

226. The useRequest Hook

7min

* Lecture incomplete. Progress cannot be changed for this item.

227. Using the useRequest Hook

4min

Resources

* Lecture incomplete. Progress cannot be changed for this item.

228. An onSuccess Callback

6min

Resources

* Lecture incomplete. Progress cannot be changed for this item.

229. Overview on Server Side Rendering

7min

Resources

* Lecture incomplete. Progress cannot be changed for this item.

230. A note about ECONNREFUSED errors

1min

* Lecture incomplete. Progress cannot be changed for this item.

231. Fetching Data During SSR

5min

Resources

* Lecture incomplete. Progress cannot be changed for this item.

232. Why the Error?

9min

* Lecture incomplete. Progress cannot be changed for this item.

233. Two Possible Solutions

7min

* Lecture incomplete. Progress cannot be changed for this item.

234. Cross Namespace Service Communication

8min

* Lecture incomplete. Progress cannot be changed for this item.

235. When is GetInitialProps Called?

7min

Resources

* Lecture incomplete. Progress cannot be changed for this item.

236. On the Server or the Browser

2min

Resources

* Lecture incomplete. Progress cannot be changed for this item.

237. Ingress-Nginx Namespace and Service - Important Update

1min

* Lecture incomplete. Progress cannot be changed for this item.

238. Specifying the Host

8min

Resources

* Lecture incomplete. Progress cannot be changed for this item.

239. Passing Through the Cookies

4min

Resources

* Lecture incomplete. Progress cannot be changed for this item.

240. A Reusable API Client

7min

Resources

* Lecture incomplete. Progress cannot be changed for this item.

241. Content on the Landing Page

2min

Resources

* Lecture incomplete. Progress cannot be changed for this item.

242. The Sign In Form

3min

Resources

* Lecture incomplete. Progress cannot be changed for this item.

243. A Reusable Header

5min

* Lecture incomplete. Progress cannot be changed for this item.

244. Moving GetInitialProps

2min

Resources

* Lecture incomplete. Progress cannot be changed for this item.

245. Issues with Custom App GetInitialProps

7min

Resources

* Lecture incomplete. Progress cannot be changed for this item.

246. Handling Multiple GetInitialProps

6min

Resources

* Lecture incomplete. Progress cannot be changed for this item.

247. Passing Props Through

3min

Resources

* Lecture incomplete. Progress cannot be changed for this item.

248. Error: Invalid <Link> with <a> child

1min

* Lecture incomplete. Progress cannot be changed for this item.

249. Building the Header

5min

Resources

* Lecture incomplete. Progress cannot be changed for this item.

250. Conditionally Showing Links

6min

Resources

* Lecture incomplete. Progress cannot be changed for this item.

251. Signing Out

5min

Resources

* Lecture incomplete. Progress cannot be changed for this item.

252. React App Catchup & Checkpoint

1min

Resources

Section 12: Code Sharing and Reuse Between Services

0 / 11 | 52min0 of 11 lectures completed52min

* Lecture incomplete. Progress cannot be changed for this item.

253. Shared Logic Between Services

5min

* Lecture incomplete. Progress cannot be changed for this item.

254. Options for Code Sharing

5min

* Lecture incomplete. Progress cannot be changed for this item.

255. NPM Organizations

4min

* Lecture incomplete. Progress cannot be changed for this item.

256. Publishing NPM Modules

4min

* Lecture incomplete. Progress cannot be changed for this item.

257. Project Setup

8min

Resources

* Lecture incomplete. Progress cannot be changed for this item.

258. Typo in package.json "files" Field - Do Not Skip

1min

* Lecture incomplete. Progress cannot be changed for this item.

259. An Easy Publish Command

7min

Resources

* Lecture incomplete. Progress cannot be changed for this item.

260. Relocating Shared Code

7min

Resources

* Lecture incomplete. Progress cannot be changed for this item.

261. Updating Import Statements

5min

Resources

* Lecture incomplete. Progress cannot be changed for this item.

262. NPM Update Command

1min

* Lecture incomplete. Progress cannot be changed for this item.

263. Updating the Common Module

6min

Section 13: Create-Read-Update-Destroy Server Setup

0 / 26 | 2hr 28min0 of 26 lectures completed2hr 28min

* Lecture incomplete. Progress cannot be changed for this item.

264. Ticketing Service Overview

3min

* Lecture incomplete. Progress cannot be changed for this item.

265. Project Setup

5min

Resources

* Lecture incomplete. Progress cannot be changed for this item.

266. Running the Ticket Service

6min

Resources

* Lecture incomplete. Progress cannot be changed for this item.

267. Mongo Connection URI

8min

Resources

* Lecture incomplete. Progress cannot be changed for this item.

268. Quick Auth Update

2min

Resources

* Lecture incomplete. Progress cannot be changed for this item.

269. Test-First Approach

4min

Resources

* Lecture incomplete. Progress cannot be changed for this item.

270. Creating the Router

5min

Resources

* Lecture incomplete. Progress cannot be changed for this item.

271. Adding Auth Protection

8min

Resources

* Lecture incomplete. Progress cannot be changed for this item.

272. Faking Authentication During Tests

7min

Resources

* Lecture incomplete. Progress cannot be changed for this item.

273. A Required Session Fix and a Global Signin Reminder

1min

* Lecture incomplete. Progress cannot be changed for this item.

274. Building a Session

6min

Resources

* Lecture incomplete. Progress cannot be changed for this item.

275. Testing Request Validation

4min

Resources

* Lecture incomplete. Progress cannot be changed for this item.

276. Validating Title and Price

4min

Resources

* Lecture incomplete. Progress cannot be changed for this item.

277. Reminder on Mongoose with TypeScript

7min

Resources

* Lecture incomplete. Progress cannot be changed for this item.

278. Defining the Ticket Model

4min

Resources

* Lecture incomplete. Progress cannot be changed for this item.

279. Creation via Route Handler

8min

Resources

* Lecture incomplete. Progress cannot be changed for this item.

280. Testing Show Routes

7min

Resources

* Lecture incomplete. Progress cannot be changed for this item.

281. Unexpected Failure!

5min

Resources

* Lecture incomplete. Progress cannot be changed for this item.

282. What's that Error?!

9min

* Lecture incomplete. Progress cannot be changed for this item.

283. Better Error Logging

5min

Resources

* Lecture incomplete. Progress cannot be changed for this item.

284. Complete Index Route Implementation

7min

Resources

* Lecture incomplete. Progress cannot be changed for this item.

285. Ticket Updating

6min

Resources

* Lecture incomplete. Progress cannot be changed for this item.

286. Handling Updates

4min

Resources

* Lecture incomplete. Progress cannot be changed for this item.

287. Permission Checking

7min

Resources

* Lecture incomplete. Progress cannot be changed for this item.

288. Final Update Changes

9min

Resources

* Lecture incomplete. Progress cannot be changed for this item.

289. Manual Testing

6min

Resources

Section 14: NATS Streaming Server - An Event Bus Implementation

0 / 23 | 2hr 57min0 of 23 lectures completed2hr 57min

* Lecture incomplete. Progress cannot be changed for this item.

290. What Now?

3min

* Lecture incomplete. Progress cannot be changed for this item.

291. NATS Server Status - IMPORTANT NOTE

1min

* Lecture incomplete. Progress cannot be changed for this item.

292. Three Important Items

5min

Resources

* Lecture incomplete. Progress cannot be changed for this item.

293. Creating a NATS Streaming Deployment

6min

Resources

* Lecture incomplete. Progress cannot be changed for this item.

294. Big Notes on NATS Streaming

10min

* Lecture incomplete. Progress cannot be changed for this item.

295. Building a NATS Test Project

8min

Resources

* Lecture incomplete. Progress cannot be changed for this item.

296. Port-Forwarding with Kubectl

5min

* Lecture incomplete. Progress cannot be changed for this item.

297. Publishing Events

6min

Resources

* Lecture incomplete. Progress cannot be changed for this item.

298. Small Required Command Change

1min

* Lecture incomplete. Progress cannot be changed for this item.

299. Listening For Data

7min

Resources

* Lecture incomplete. Progress cannot be changed for this item.

300. Accessing Event Data

7min

Resources

* Lecture incomplete. Progress cannot be changed for this item.

301. Client ID Generation

5min

Resources

* Lecture incomplete. Progress cannot be changed for this item.

302. Queue Groups

7min

Resources

* Lecture incomplete. Progress cannot be changed for this item.

303. Manual Ack Mode

10min

Resources

* Lecture incomplete. Progress cannot be changed for this item.

304. Client Health Checks

10min

* Lecture incomplete. Progress cannot be changed for this item.

305. Graceful Client Shutdown

7min

Resources

* Lecture incomplete. Progress cannot be changed for this item.

306. Core Concurrency Issues

13min

* Lecture incomplete. Progress cannot be changed for this item.

307. Common Questions

8min

* Lecture incomplete. Progress cannot be changed for this item.

308. [Optional] More Possible Concurrency Solutions

17min

* Lecture incomplete. Progress cannot be changed for this item.

309. Solving Concurrency Issues

20min

* Lecture incomplete. Progress cannot be changed for this item.

310. Concurrency Control with the Tickets App

10min

* Lecture incomplete. Progress cannot be changed for this item.

311. Event Redelivery

5min

Resources

* Lecture incomplete. Progress cannot be changed for this item.

312. Durable Subscriptions

9min

Resources

Section 15: Connecting to NATS in a Node JS World

0 / 17 | 1hr 22min0 of 17 lectures completed1hr 22min

* Lecture incomplete. Progress cannot be changed for this item.

313. Reusable NATS Listeners

5min

* Lecture incomplete. Progress cannot be changed for this item.

314. The Listener Abstract Class

9min

Resources

* Lecture incomplete. Progress cannot be changed for this item.

315. Extending the Listener

6min

Resources

* Lecture incomplete. Progress cannot be changed for this item.

316. Quick Refactor

3min

Resources

* Lecture incomplete. Progress cannot be changed for this item.

317. Leveraging TypeScript for Listener Validation

5min

Resources

* Lecture incomplete. Progress cannot be changed for this item.

318. Subjects Enum

3min

* Lecture incomplete. Progress cannot be changed for this item.

319. Custom Event Interface

2min

Resources

* Lecture incomplete. Progress cannot be changed for this item.

320. Enforcing Listener Subjects

7min

Resources

* Lecture incomplete. Progress cannot be changed for this item.

321. Quick Note: 'readonly' in Typescript

1min

* Lecture incomplete. Progress cannot be changed for this item.

322. Enforcing Data Types

4min

Resources

* Lecture incomplete. Progress cannot be changed for this item.

323. Where Does this Get Used?

3min

* Lecture incomplete. Progress cannot be changed for this item.

324. Custom Publisher

8min

Resources

* Lecture incomplete. Progress cannot be changed for this item.

325. Using the Custom Publisher

4min

Resources

* Lecture incomplete. Progress cannot be changed for this item.

326. Awaiting Event Publication

4min

Resources

* Lecture incomplete. Progress cannot be changed for this item.

327. Common Event Definitions Summary

6min

* Lecture incomplete. Progress cannot be changed for this item.

328. Updating the Common Module

8min

Resources

* Lecture incomplete. Progress cannot be changed for this item.

329. Reing NATS

2min

Section 16: Managing a NATS Client

0 / 19 | 1hr 37min0 of 19 lectures completed1hr 37min

* Lecture incomplete. Progress cannot be changed for this item.

330. Publishing Ticket Creation

4min

Resources

* Lecture incomplete. Progress cannot be changed for this item.

331. More on Publishing

3min

Resources

* Lecture incomplete. Progress cannot be changed for this item.

332. NATS Client Singleton

5min

* Lecture incomplete. Progress cannot be changed for this item.

333. Node Nats Streaming Installation

1min

* Lecture incomplete. Progress cannot be changed for this item.

334. Remember Mongoose?

5min

Resources

* Lecture incomplete. Progress cannot be changed for this item.

335. TS Error - Did you forget to include 'void' in your type argument

1min

* Lecture incomplete. Progress cannot be changed for this item.

336. Singleton Implementation

9min

Resources

* Lecture incomplete. Progress cannot be changed for this item.

337. Accessing the NATS Client

4min

Resources

* Lecture incomplete. Progress cannot be changed for this item.

338. Graceful Shutdown

7min

Resources

* Lecture incomplete. Progress cannot be changed for this item.

339. Successful Listen!

3min

* Lecture incomplete. Progress cannot be changed for this item.

340. Ticket Update Publishing

4min

Resources

* Lecture incomplete. Progress cannot be changed for this item.

341. Failed Event Publishing

7min

* Lecture incomplete. Progress cannot be changed for this item.

342. Handling Publish Failures

6min

* Lecture incomplete. Progress cannot be changed for this item.

343. Fixing a Few Tests

4min

* Lecture incomplete. Progress cannot be changed for this item.

344. Redirecting Imports

6min

* Lecture incomplete. Progress cannot be changed for this item.

345. Providing a Mock Implementation

9min

* Lecture incomplete. Progress cannot be changed for this item.

346. Test-Suite Wide Mocks

2min

Resources

* Lecture incomplete. Progress cannot be changed for this item.

347. Ensuring Mock Invocations

9min

Resources

* Lecture incomplete. Progress cannot be changed for this item.

348. NATS Env Variables

8min

Resources

Section 17: Cross-Service Data Replication In Action

0 / 28 | 2hr 44min0 of 28 lectures completed2hr 44min

* Lecture incomplete. Progress cannot be changed for this item.

349. The Orders Service

5min

* Lecture incomplete. Progress cannot be changed for this item.

350. Scaffolding the Orders Service

5min

Resources

* Lecture incomplete. Progress cannot be changed for this item.

351. A Touch More Setup

8min

Resources

* Lecture incomplete. Progress cannot be changed for this item.

352. Ingress Routing Rules

2min

Resources

* Lecture incomplete. Progress cannot be changed for this item.

353. Scaffolding a Few Route Handlers

10min

Resources

* Lecture incomplete. Progress cannot be changed for this item.

354. Subtle Service Coupling

7min

Resources

* Lecture incomplete. Progress cannot be changed for this item.

355. Associating Orders and Tickets

7min

* Lecture incomplete. Progress cannot be changed for this item.

356. Order Model Setup

9min

Resources

* Lecture incomplete. Progress cannot be changed for this item.

357. The Need for an Enum

6min

* Lecture incomplete. Progress cannot be changed for this item.

358. Creating an Order Status Enum

9min

Resources

* Lecture incomplete. Progress cannot be changed for this item.

359. More on Mongoose Refs

3min

* Lecture incomplete. Progress cannot be changed for this item.

360. Defining the Ticket Model

7min

Resources

* Lecture incomplete. Progress cannot be changed for this item.

361. Order Creation Logic

6min

Resources

* Lecture incomplete. Progress cannot be changed for this item.

362. Finding Reserved Tickets

6min

Resources

* Lecture incomplete. Progress cannot be changed for this item.

363. Convenience Document Methods

8min

Resources

* Lecture incomplete. Progress cannot be changed for this item.

364. Order Expiration Times

6min

Resources

* Lecture incomplete. Progress cannot be changed for this item.

365. globalThis has no index signature TS Error

1min

* Lecture incomplete. Progress cannot be changed for this item.

366. Test Suite Setup

3min

Resources

* Lecture incomplete. Progress cannot be changed for this item.

367. Small Update for "Value of type 'typeof ObjectId' is not callable"

1min

* Lecture incomplete. Progress cannot be changed for this item.

368. Asserting Tickets Exist

6min

Resources

* Lecture incomplete. Progress cannot be changed for this item.

369. Asserting Reserved Tickets

5min

Resources

* Lecture incomplete. Progress cannot be changed for this item.

370. Testing the Success Case

4min

Resources

* Lecture incomplete. Progress cannot be changed for this item.

371. Fetching a User's Orders

5min

Resources

* Lecture incomplete. Progress cannot be changed for this item.

372. A Slightly Complicated Test

12min

Resources

* Lecture incomplete. Progress cannot be changed for this item.

373. Fetching Individual Orders

5min

Resources

* Lecture incomplete. Progress cannot be changed for this item.

374. Does Fetching Work?

7min

Resources

* Lecture incomplete. Progress cannot be changed for this item.

375. Cancelling an Order

5min

Resources

* Lecture incomplete. Progress cannot be changed for this item.

376. Can We Cancel?

9min

Resources

Section 18: Understanding Event Flow

0 / 6 | 30min0 of 6 lectures completed30min

* Lecture incomplete. Progress cannot be changed for this item.

377. Orders Service Events

5min

* Lecture incomplete. Progress cannot be changed for this item.

378. Creating the Events

9min

Resources

* Lecture incomplete. Progress cannot be changed for this item.

379. Implementing the Publishers

3min

* Lecture incomplete. Progress cannot be changed for this item.

380. Publishing the Order Creation

5min

* Lecture incomplete. Progress cannot be changed for this item.

381. Publishing Order Cancellation

3min

* Lecture incomplete. Progress cannot be changed for this item.

382. Testing Event Publishing

6min

Resources

Section 19: Listening for Events and Handling Concurrency Issues

0 / 48 | 4hr 13min0 of 48 lectures completed4hr 13min

* Lecture incomplete. Progress cannot be changed for this item.

383. Heads Up Regarding Some Mongoose TS Errors

1min

* Lecture incomplete. Progress cannot be changed for this item.

384. Time for Listeners!

2min

* Lecture incomplete. Progress cannot be changed for this item.

385. Reminder on Listeners

2min

* Lecture incomplete. Progress cannot be changed for this item.

386. Blueprint for Listeners

4min

Resources

* Lecture incomplete. Progress cannot be changed for this item.

387. A Few More Reminders

5min

Resources

* Lecture incomplete. Progress cannot be changed for this item.

388. Simple onMessage Implementation

2min

Resources

* Lecture incomplete. Progress cannot be changed for this item.

389. ID Adjustment

6min

Resources

* Lecture incomplete. Progress cannot be changed for this item.

390. Ticket Updated Listener Implementation

5min

Resources

* Lecture incomplete. Progress cannot be changed for this item.

391. Initializing the Listeners

3min

Resources

* Lecture incomplete. Progress cannot be changed for this item.

392. A Quick Manual Test

3min

* Lecture incomplete. Progress cannot be changed for this item.

393. Clear Concurrency Issues

13min

* Lecture incomplete. Progress cannot be changed for this item.

394. Reminder on Versioning Records

7min

* Lecture incomplete. Progress cannot be changed for this item.

395. Optimistic Concurrency Control

6min

* Lecture incomplete. Progress cannot be changed for this item.

396. Mongoose Update-If-Current

4min

* Lecture incomplete. Progress cannot be changed for this item.

397. Implementing OCC with Mongoose

4min

Resources

* Lecture incomplete. Progress cannot be changed for this item.

398. Test functions cannot both take a 'done' callback and return something Error

1min

* Lecture incomplete. Progress cannot be changed for this item.

399. Testing OCC

9min

Resources

* Lecture incomplete. Progress cannot be changed for this item.

400. One More Test

4min

Resources

* Lecture incomplete. Progress cannot be changed for this item.

401. Who Updates Versions?

6min

* Lecture incomplete. Progress cannot be changed for this item.

402. Including Versions in Events

3min

Resources

* Lecture incomplete. Progress cannot be changed for this item.

403. Updating Tickets Event Definitions

4min

Resources

* Lecture incomplete. Progress cannot be changed for this item.

404. Property 'version' is missing TS Errors After Running Skaffold

1min

* Lecture incomplete. Progress cannot be changed for this item.

405. Applying a Version Query

7min

Resources

* Lecture incomplete. Progress cannot be changed for this item.

406. Did it Work?

5min

* Lecture incomplete. Progress cannot be changed for this item.

407. Abstracted Query Method

6min

Resources

* Lecture incomplete. Progress cannot be changed for this item.

408. [Optional] Versioning Without Update-If-Current

19min

* Lecture incomplete. Progress cannot be changed for this item.

409. Testing Listeners

5min

Resources

* Lecture incomplete. Progress cannot be changed for this item.

410. A Complete Listener Test

9min

Resources

* Lecture incomplete. Progress cannot be changed for this item.

411. Testing the Ack Call

2min

Resources

* Lecture incomplete. Progress cannot be changed for this item.

412. Testing the Ticket Updated Listener

8min

Resources

* Lecture incomplete. Progress cannot be changed for this item.

413. Success Case Testing

5min

Resources

* Lecture incomplete. Progress cannot be changed for this item.

414. Out-Of-Order Events

5min

Resources

* Lecture incomplete. Progress cannot be changed for this item.

415. The Next Few Videos

4min

Resources

* Lecture incomplete. Progress cannot be changed for this item.

416. Fixing a Few Tests

7min

Resources

* Lecture incomplete. Progress cannot be changed for this item.

417. Listeners in the Tickets Service

2min

* Lecture incomplete. Progress cannot be changed for this item.

418. Building the Listener

5min

Resources

* Lecture incomplete. Progress cannot be changed for this item.

419. Strategies for Locking a Ticket

5min

* Lecture incomplete. Progress cannot be changed for this item.

420. Reserving a Ticket

4min

Resources

* Lecture incomplete. Progress cannot be changed for this item.

421. Setup for Testing Reservation

7min

Resources

* Lecture incomplete. Progress cannot be changed for this item.

422. Test Implementation

4min

Resources

* Lecture incomplete. Progress cannot be changed for this item.

423. Missing Update Event

6min

* Lecture incomplete. Progress cannot be changed for this item.

424. Private vs Protected Properties

7min

* Lecture incomplete. Progress cannot be changed for this item.

425. Publishing While Listening

7min

Resources

* Lecture incomplete. Progress cannot be changed for this item.

426. Mock Function Arguments

10min

Resources

* Lecture incomplete. Progress cannot be changed for this item.

427. Order Cancelled Listener

7min

Resources

* Lecture incomplete. Progress cannot be changed for this item.

428. A Lightning-Quick Test

7min

Resources

* Lecture incomplete. Progress cannot be changed for this item.

429. Don't Forget to Listen!

2min

Resources

* Lecture incomplete. Progress cannot be changed for this item.

430. Rejecting Edits of Reserved Tickets

6min

Resources

Section 20: Worker Services

0 / 19 | 1hr 36min0 of 19 lectures completed1hr 36min

* Lecture incomplete. Progress cannot be changed for this item.

431. The Expiration Service

3min

* Lecture incomplete. Progress cannot be changed for this item.

432. Expiration Options

8min

* Lecture incomplete. Progress cannot be changed for this item.

433. Initial Setup

6min

Resources

* Lecture incomplete. Progress cannot be changed for this item.

434. Skaffold errors - Expiration Image Can't be Pulled

1min

* Lecture incomplete. Progress cannot be changed for this item.

435. A Touch of Kubernetes Setup

8min

Resources

* Lecture incomplete. Progress cannot be changed for this item.

436. File Sync Setup

3min

* Lecture incomplete. Progress cannot be changed for this item.

437. Listener Creation

4min

Resources

* Lecture incomplete. Progress cannot be changed for this item.

438. What's Bull All About?

4min

* Lecture incomplete. Progress cannot be changed for this item.

439. Creating a Queue

9min

Resources

* Lecture incomplete. Progress cannot be changed for this item.

440. Queueing a Job on Event Arrival

5min

Resources

* Lecture incomplete. Progress cannot be changed for this item.

441. Testing Job Processing

3min

* Lecture incomplete. Progress cannot be changed for this item.

442. Delaying Job Processing

5min

Resources

* Lecture incomplete. Progress cannot be changed for this item.

443. Defining the Expiration Complete Event

4min

Resources

* Lecture incomplete. Progress cannot be changed for this item.

444. Publishing an Event on Job Processing

6min

Resources

* Lecture incomplete. Progress cannot be changed for this item.

445. Handling an Expiration Event

6min

Resources

* Lecture incomplete. Progress cannot be changed for this item.

446. Emitting the Order Cancelled Event

6min

Resources

* Lecture incomplete. Progress cannot be changed for this item.

447. Testing the Expiration Complete Listener

6min

Resources

* Lecture incomplete. Progress cannot be changed for this item.

448. A Touch More Testing

7min

Resources

* Lecture incomplete. Progress cannot be changed for this item.

449. Listening for Expiration

2min

Resources

Section 21: Handling Payments

0 / 31 | 2hr 40min0 of 31 lectures completed2hr 40min

* Lecture incomplete. Progress cannot be changed for this item.

450. The Payments Service

2min

* Lecture incomplete. Progress cannot be changed for this item.

451. globalThis has no index signature TS Error

1min

* Lecture incomplete. Progress cannot be changed for this item.

452. Initial Setup

9min

Resources

* Lecture incomplete. Progress cannot be changed for this item.

453. Replicated Fields

6min

* Lecture incomplete. Progress cannot be changed for this item.

454. Another Order Model!

8min

* Lecture incomplete. Progress cannot be changed for this item.

455. Update-If-Current

1min

* Lecture incomplete. Progress cannot be changed for this item.

456. Replicating Orders

4min

* Lecture incomplete. Progress cannot be changed for this item.

457. Testing Order Creation

6min

Resources

* Lecture incomplete. Progress cannot be changed for this item.

458. Marking an Order as Cancelled

6min

Resources

* Lecture incomplete. Progress cannot be changed for this item.

459. Cancelled Testing

7min

Resources

* Lecture incomplete. Progress cannot be changed for this item.

460. ing the Listeners

4min

Resources

* Lecture incomplete. Progress cannot be changed for this item.

461. Payments Flow with Stripe

5min

* Lecture incomplete. Progress cannot be changed for this item.

462. Implementing the Create Charge Handler

9min

Resources

* Lecture incomplete. Progress cannot be changed for this item.

463. Validating Order Payment

4min

Resources

* Lecture incomplete. Progress cannot be changed for this item.

464. Testing Order Validation Before Payment

7min

Resources

* Lecture incomplete. Progress cannot be changed for this item.

465. Testing Same-User Validation

5min

Resources

* Lecture incomplete. Progress cannot be changed for this item.

466. Stripe Setup

4min

* Lecture incomplete. Progress cannot be changed for this item.

467. Creating a Stripe Secret

3min

* Lecture incomplete. Progress cannot be changed for this item.

468. Creating a Charge with Stripe

6min

Resources

* Lecture incomplete. Progress cannot be changed for this item.

469. Manual Testing of Payments

7min

* Lecture incomplete. Progress cannot be changed for this item.

470. Automated Payment Testing

6min

Resources

* Lecture incomplete. Progress cannot be changed for this item.

471. Mocked Stripe Client

4min

Resources

* Lecture incomplete. Progress cannot be changed for this item.

472. A More Realistic Test Setup

9min

Resources

* Lecture incomplete. Progress cannot be changed for this item.

473. Realistic Test Implementation

6min

Resources

* Lecture incomplete. Progress cannot be changed for this item.

474. Tying an Order and Charge Together

7min

Resources

* Lecture incomplete. Progress cannot be changed for this item.

475. Testing Payment Creation

6min

Resources

* Lecture incomplete. Progress cannot be changed for this item.

476. Publishing a Payment Created Event

7min

Resources

* Lecture incomplete. Progress cannot be changed for this item.

477. More on Publishing

3min

Resources

* Lecture incomplete. Progress cannot be changed for this item.

478. Marking an Order as Complete

6min

Resources

* Lecture incomplete. Progress cannot be changed for this item.

479. Important Info About the Next Lecture - Don't Skip

1min

* Lecture incomplete. Progress cannot be changed for this item.

480. Don't Cancel Completed Orders!

1min

Resources

Section 22: Back to the Client

0 / 21 | 1hr 43min0 of 21 lectures completed1hr 43min

* Lecture incomplete. Progress cannot be changed for this item.

481. A Few More Pages

6min

* Lecture incomplete. Progress cannot be changed for this item.

482. Reminder on Data Fetching with Next

6min

* Lecture incomplete. Progress cannot be changed for this item.

483. Two Quick Fixes

6min

Resources

* Lecture incomplete. Progress cannot be changed for this item.

484. Scaffolding a Form

4min

Resources

* Lecture incomplete. Progress cannot be changed for this item.

485. Sanitizing Price Input

6min

Resources

* Lecture incomplete. Progress cannot be changed for this item.

486. Ticket Creation

7min

Resources

* Lecture incomplete. Progress cannot be changed for this item.

487. Listing All Tickets

8min

Resources

* Lecture incomplete. Progress cannot be changed for this item.

488. Reminder on Invalid <Link> with <a> child Errors

1min

* Lecture incomplete. Progress cannot be changed for this item.

489. Linking to Wildcard Routes

7min

Resources

* Lecture incomplete. Progress cannot be changed for this item.

490. Creating an Order

7min

Resources

* Lecture incomplete. Progress cannot be changed for this item.

491. Programmatic Navigation to Wildcard Routes

4min

Resources

* Lecture incomplete. Progress cannot be changed for this item.

492. The Expiration Timer

11min

Resources

* Lecture incomplete. Progress cannot be changed for this item.

493. Dising the Expiration

1min

Resources

* Lecture incomplete. Progress cannot be changed for this item.

494. Showing a Stripe Payment Form

3min

* Lecture incomplete. Progress cannot be changed for this item.

495. Module not found: Can't resolve 'prop-types'

1min

* Lecture incomplete. Progress cannot be changed for this item.

496. Configuring Stripe

4min

Resources

* Lecture incomplete. Progress cannot be changed for this item.

497. Test Credit Card Numbers

2min

* Lecture incomplete. Progress cannot be changed for this item.

498. Paying for an Order

9min

Resources

* Lecture incomplete. Progress cannot be changed for this item.

499. Filtering Reserved Tickets

2min

Resources

* Lecture incomplete. Progress cannot be changed for this item.

500. Header Links

2min

Resources

* Lecture incomplete. Progress cannot be changed for this item.

501. Rendering a List of Orders

6min

Resources

Section 23: CI/CD

0 / 30 | 2hr 17min0 of 30 lectures completed2hr 17min

* Lecture incomplete. Progress cannot be changed for this item.

502. Development Workflow

4min

* Lecture incomplete. Progress cannot be changed for this item.

503. Git Repository Approaches

6min

* Lecture incomplete. Progress cannot be changed for this item.

504. Creating a GitHub Action

7min

* Lecture incomplete. Progress cannot be changed for this item.

505. Adding a CI Test Script

3min

Resources

* Lecture incomplete. Progress cannot be changed for this item.

506. Tests in GitHub Actions Hang - Jest did not exit

1min

* Lecture incomplete. Progress cannot be changed for this item.

507. Running Tests on PR Creation

5min

* Lecture incomplete. Progress cannot be changed for this item.

508. Output of Failing Tests

6min

* Lecture incomplete. Progress cannot be changed for this item.

509. Running Tests in Parallel

7min

Resources

* Lecture incomplete. Progress cannot be changed for this item.

510. Verifying a Test Run

3min

* Lecture incomplete. Progress cannot be changed for this item.

511. Selective Test Execution

6min

Resources

* Lecture incomplete. Progress cannot be changed for this item.

512. Deployment Options

8min

* Lecture incomplete. Progress cannot be changed for this item.

513. Creating a Hosted Cluster

3min

* Lecture incomplete. Progress cannot be changed for this item.

514. Reminder on Kubernetes Context

4min

* Lecture incomplete. Progress cannot be changed for this item.

515. Reminder on Swapping Contexts

4min

* Lecture incomplete. Progress cannot be changed for this item.

516. The Deployment Plan

4min

* Lecture incomplete. Progress cannot be changed for this item.

517. Building an Image in an Action

9min

Resources

* Lecture incomplete. Progress cannot be changed for this item.

518. Testing the Image Build

3min

* Lecture incomplete. Progress cannot be changed for this item.

519. Reing the Deployment

7min

Resources

* Lecture incomplete. Progress cannot be changed for this item.

520. Applying Kubernetes Manifests

3min

Resources

* Lecture incomplete. Progress cannot be changed for this item.

521. Prod vs Dev Manifest Files

4min

Resources

* Lecture incomplete. Progress cannot be changed for this item.

522. Manual Secret Creation

4min

* Lecture incomplete. Progress cannot be changed for this item.

523. Don't Forget Ingress-Nginx!

6min

* Lecture incomplete. Progress cannot be changed for this item.

524. Testing Automated Deployment

3min

* Lecture incomplete. Progress cannot be changed for this item.

525. Additional Deploy Files

7min

Resources

* Lecture incomplete. Progress cannot be changed for this item.

526. A Successful Deploy!

6min

* Lecture incomplete. Progress cannot be changed for this item.

527. Buying a Domain Name

3min

* Lecture incomplete. Progress cannot be changed for this item.

528. Three Important Changes Needed to Deploy - Do Not Skip!

1min

* Lecture incomplete. Progress cannot be changed for this item.

529. Configuring the Domain Name

5min

Resources

* Lecture incomplete. Progress cannot be changed for this item.

530. I Really Hope This Works

2min

* Lecture incomplete. Progress cannot be changed for this item.

531. Next Steps