

نام دوره: Data Platform Engineering دوره

طول دوره: ۴۲ ساعت

نوع ارائه: آنلاین

زمان برگزاری دوره: از 16 آبان - شنبه ها ۱۷:۳۰ - ۲۰:۳۰

پیش نیاز شرکت در دوره: دوره Data Lakehouse مقدماتی

سطح دوره: متوسط تا پیشرفته

مخاطبین: مهندسان داده و پلتفرم داده، مهندسان DevOps، Cloud و SRE

مدیران و کارشناسان پایگاه داده، توسعه دهندگان بک اند و نرم افزار،

مهندسان Big Data و MLOps، معماران داده و زیرساخت

دانشجویان و پژوهشگران حوزه مهندسی داده و سیستم های توزیع شده

حسن احمدخانی

« معمار کلان داده در ایرانسل از ۱۴۰۰ تا کنون.

« مشاور مهندسی داده در تکنولایف از ۱۴۰۳.

« مشاور مهندسی داده در زپ از ۱۴۰۲.

« مدرس کلان داده و NoSQL ها از سال ۱۳۹۷ تاکنون.

« معمار کلان داده در داتین از ۱۳۹۹ تا ۱۴۰۰.

« مهندس ارشد داده و Oracle DBA در شرکت پژواک از ۱۳۹۴ تا ۱۳۹۸.

« مشاور اوراکل در بیمه ایران از ۱۳۹۵ تا ۱۳۹۶.

« برنامه نویس جاوا و مهندس داده در ایزایران از ۱۳۹۳ تا ۱۳۹۴.

« برنامه نویس جاوا در شرکت پژواک از ۱۳۸۹ تا ۱۳۹۳.



مدرس دوره



فصل اول : آشنایی با لینوکس، داکر و Containerization

در این بخش، زیرساخت اولیه موردنیاز برای کار با محیط‌های کانتینری آماده می‌شود و شرکت‌کنندگان با نصب و راه‌اندازی لینوکس و مفاهیم اصلی Docker آشنا می‌شوند. ساخت و مدیریت Container و Image، کار با Repository و Registry و همچنین آشنایی با Docker Swarm از دیگر مباحث این بخش است.

- نصب لینوکس و آماده‌سازی محیط
- معماری Docker و بررسی ویژگی‌های Docker و کار با Containerها
- معرفی و کار با Swarm و Registry
- ساخت Image، استفاده از Imageهای آماده و کار با Repositoryهای عمومی

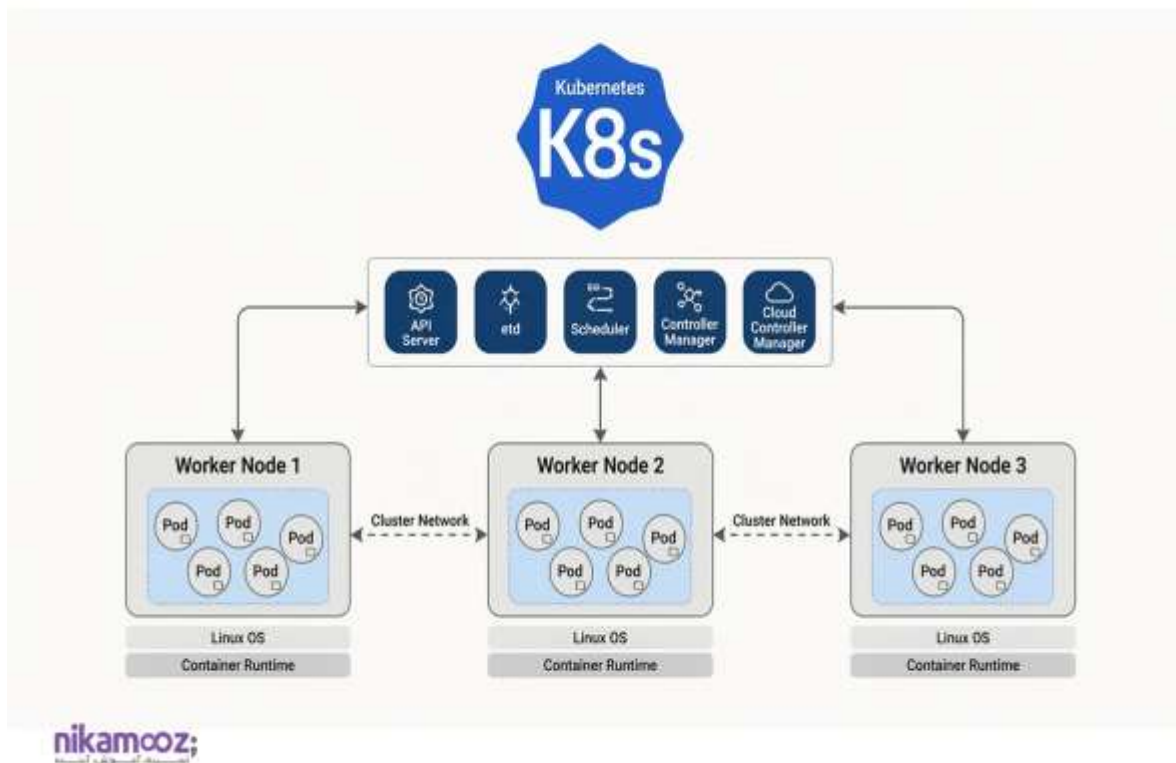


فصل دوم: مفاهیم پایه ای کورننتیز

این بخش به شناخت معماری Kubernetes و اجزای اصلی آن و نحوه نصب و راه‌اندازی یک کلاستر اختصاص دارد. شرکت‌کنندگان با منابعی مانند Pod، Namespace، ReplicaSet، Deployment، Service، ConfigMap و Secret آشنا می‌شوند و روش‌های زمان‌بندی و توزیع Podها میان Nodeها را نیز یاد می‌گیرند.

مدرس: حسن احمدخانی

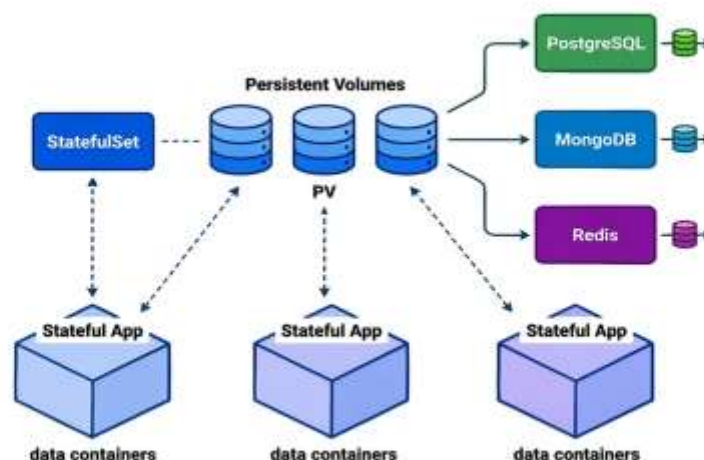
- معماری و اجزای Kubernetes
- نصب Kubernetes
- ویژگی‌های Kubernetes و معرفی منابع Kubernetes
- معرفی و کار با Pod و Namespace
- معرفی و کار با ReplicaSet
- ارتباط با Pod از طریق Service و انواع Service ها
- معرفی Deployment و مزیت‌های آن
- معرفی و کار با ConfigMap و Secret
- زمان‌بندی و توزیع Pod ها روی Node ها Taint، Toleration، Node Affinity و Pod Affinity
- اهمیت لینوکس
- توزیع های مهم لینوکس
- نصب و کار با Ubuntu
- مروری بر تنظیمات شبکه، پارتیشن بندی و دیسک



فصل سوم: کورنتیز برای دیتا

در این بخش، الزامات اجرای سرویس‌های داده‌ای در Kubernetes و تفاوت سرویس‌های Stateful و Stateless بررسی می‌شود. مباحثی مانند StatefulSet، Headless Service، Persistent Volume، Persistent Volume Claim و Storage Class آموزش داده می‌شوند تا بستر مناسب برای استقرار پایدار سرویس‌های داده فراهم شود.

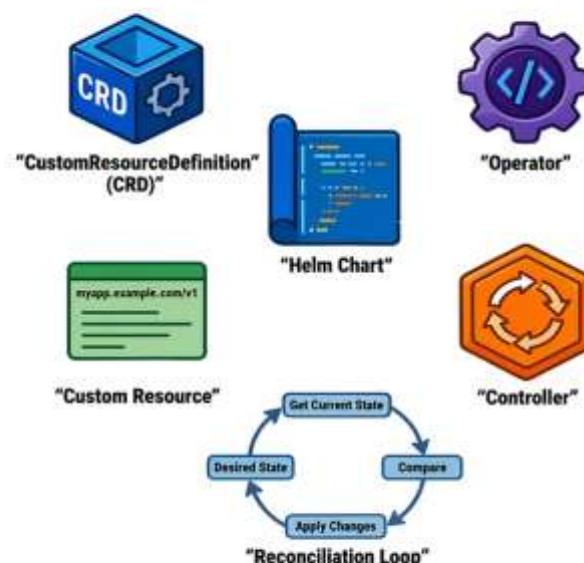
- تفاوت سرویس‌های Stateful و Stateless
- ویژگی‌های Kubernetes برای پلتفرم‌های داده‌ای
- معرفی و کار با StatefulSet
- مدیریت StatefulSet ها
- معرفی و کار با Headless Service
- معرفی و کار با Persistent Volume و Persistent Volume Claim
- معرفی و کار با Storage Class و Provisioner
- کاربرد Local Disk و No Provisioner در Kubernetes
- زبان‌های برنامه نویسی مهم برای مهندسان داده
- آیا مهندس داده باید به خوبی یک مهندس نرم افزار در طراحی و برنامه نویسی باشد؟
- نکات و اصطلاحات مهم طراحی و توسعه نرم افزار برای مهندسان داده
- کار با گیت
- کد نویسی و انجام سناریوهای مختلف پردازش داده با پایتون
- کد نویسی و انجام سناریوهای مختلف پردازش داده با جاوا



فصل چهارم: مباحث پیشرفته در کوبرنتیز

این بخش به قابلیت‌های توسعه‌پذیری Kubernetes و ابزارهایی اختصاص دارد که امکان مدیریت سرویس‌ها و معماری‌های پیچیده‌تر را فراهم می‌کنند. شرکت‌کنندگان با Custom Resource ، CRD، Operator ، Reconciliation Loop آشنا می‌شوند و نحوه استفاده از Helm برای بسته‌بندی و مدیریت استقرارها را فرا می‌گیرند.

- معرفی مباحث پیشرفته و معماری Extensibility در Kubernetes
- معرفی و کار با Custom Resource (CR) و Custom Resource Definition (CRD)
- بررسی ساختار، ایجاد و مدیریت CRD ها
- معرفی و کار با Operator ها
- آشنایی با مفهوم Controller و Reconciliation Loop
- بررسی ارتباط Operator ، CRD و Controller
- معرفی Helm و مفهوم Package Management در Kubernetes
- ساختار Helm Chart و اجزای آن
- کار با Values ، Templates و Releases ها
- مروری بر Best Practices و سناریوهای عملی استفاده از CRD ، Operator و Helm



فصل پنجم: استقرار پلتفرم دیتا در کوبرنتیز

در این بخش، نحوه طراحی و استقرار سرویس‌های اصلی یک Data Platform بر بستر Kubernetes به صورت عملی بررسی می‌شود. شرکت‌کنندگان با معماری، Persistence، High Availability، Replication، Backup، Recovery و مدیریت منابع در PostgreSQL، MongoDB، Kafka، Spark و ClickHouse آشنا می‌شوند و استفاده از Operatorهای مرتبط با هر سرویس را تجربه می‌کنند.

- **آشنایی با معماری و الزامات Data Platform روی Kubernetes**

- **استقرار PostgreSQL در Kubernetes و بهترین شیوه‌های آن**

- مرور معماری PostgreSQL و ساختار Instance
- معماری PostgreSQL در Kubernetes، Persistence و High Availability
- معرفی و کار با CloudNativePG
- Backup، Recovery و مدیریت منابع

- **استقرار MongoDB در Kubernetes و بهترین شیوه‌های آن**

- مرور معماری MongoDB و ساختار Instance
- معماری Replica Set و Persistence
- معرفی و کار با MongoDB KAs Operator
- Backup، Recovery و مدیریت منابع

- **استقرار Kafka در Kubernetes و بهترین شیوه‌های آن**

- مرور معماری Kafka، ساختار Instance و اجزای کلاستر
- معماری Brokerها و Storage
- مدیریت Replication و Persistence
- کار با Kafka Operator

- **استقرار Spark در Kubernetes و بهترین شیوه‌های آن**

- مرور معماری و ماهیت Spark
- معماری Spark روی Kubernetes
- اجرای Jobها و مدیریت منابع و پارامترها
- تخصیص منابع
- معرفی و کاربردهای Apache YuniKorn

• استقرار ClickHouse در Kubernetes و بهترین شیوه‌های آن

- مرور معماری ClickHouse ، ساختار Instance و اجزای کلاستر ClickHouse
- Replication و Scaling
- مدیریت منابع و Persistence
- کار با ClickHouse Operator

Data Platform on Kubernetes Deployment

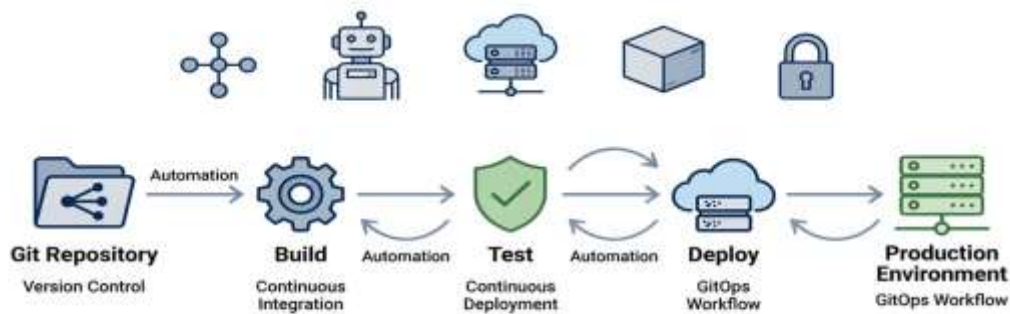


فصل ششم: CI/CD در پایپ‌لاین داده

این فصل هسته اصلی کار یک مهندس داده را تشکیل می‌دهد و فرآیندهای حیاتی ETL و ELT و اهمیت کیفیت داده را پوشش می‌دهد. این بخش بر پیاده‌سازی عملی سناریوهای آماده‌سازی داده با استفاده از دستورات SQL و یک زبان برنامه‌نویسی تمرکز می‌کند. همچنین، استفاده از ابزارهای تخصصی مانند Apache NiFi برای طراحی Data Flow های جمع‌آوری داده و Apache Spark به عنوان یک فریم‌ورک پردازش توزیع‌شده و مقیاس‌پذیر، برای اجرای سناریوهای مختلف Spark SQL و خواندن و نوشتن در منابع داده‌ای NoSQL و تحلیلی آموزش داده می‌شود. در پایان، مفاهیم مربوط به فرمت‌های بهینه ذخیره‌سازی مانند ORC و Parquet و روش‌های Scheduling تسک‌های داده‌ای با استفاده از ابزارهایی مانند Airflow بررسی خواهند شد.

- ETL و ELT
- کیفیت داده و معیارهای آن
- پیاده‌سازی سناریوهای مختلف آماده‌سازی داده به کمک یک زبان برنامه‌نویسی و دستورات SQL
- مقیاس‌پذیری در لایه پردازش و استفاده از روش‌های پردازش توزیع‌شده
- استفاده از Apache NiFi برای طراحی Data Flow های جمع‌آوری داده
- انجام سناریوهای متنوع Ingestion در NiFi به پایگاه‌های داده NoSQL و تحلیلی
- مروری بر Apache Spark و قابلیت‌های آن
- انجام سناریوهای مختلف پردازش داده در Spark SQL
- بررسی فرمت فایل‌های ORC و Parquet و مزایا و معایب آن‌ها
- خواندن و نوشتن در منابع داده‌ای NoSQL و تحلیلی با Apache Spark
- روش‌های Scheduling تسک‌ها
- استفاده از Airflow برای Scheduling

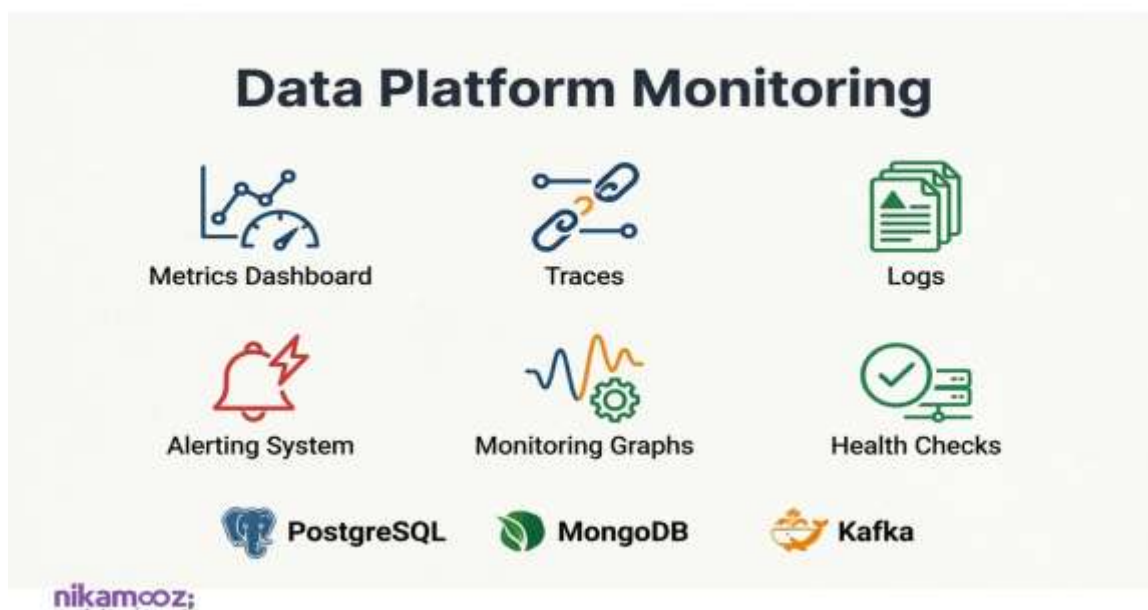
CI/CD Data Pipeline



فصل هفتم: مانیتورینگ پلتفرم داده

در این بخش، اصول پایش و ارزیابی سلامت سرویس‌های یک پلتفرم داده بررسی می‌شود و شرکت‌کنندگان با Metrics، Logs، Traces، Health Check و Alerting آشنا می‌شوند. نحوه Monitoring سرویس‌هایی مانند PostgreSQL، MongoDB، Kafka، Spark و ClickHouse نیز با تمرکز بر تشخیص سریع خطاها و مشکلات عملیاتی مرور خواهد شد.

- آشنایی با روش‌ها و نیازمندی‌های Monitoring در پلتفرم داده
- بررسی شاخص‌های کلیدی Metrics، Logs و Traces
- بررسی Health Check، Alerting و Incident Detection
- بررسی Monitoring سرویس‌های PostgreSQL، MongoDB، Kafka، Spark و ClickHouse



سوالات متداول دوره Data Platform Engineering

DataOps دقیقاً چیست و چه تفاوتی با DevOps دارد؟

DevOps بیشتر بر چرخه توسعه، استقرار و بهره‌برداری نرم‌افزار تمرکز دارد؛ اما در DataOps علاوه بر سرویس نرم‌افزاری، با داده، زیرساخت ذخیره سازی، موتورهای پردازش، پایپ‌لاین‌ها و سرویس‌های دارای وضعیت (Stateful) نیز سروکار داریم. در یک پلتفرم داده (Data Platform)، فقط در دسترس بودن برنامه کاربردی کافی نیست؛ داده باید پایدار بماند، پردازش‌ها قابلیت بازبینی داشته باشند و تغییرات زیرساخت نیز به صورت کنترل شده انجام شوند.

چرا برای راه اندازی یک Data Platform ، آشنایی با Docker کافی نیست و به Kubernetes هم نیاز داریم؟

Docker امکان ساخت و اجرای سرویس ها در قالب Container را فراهم می کند، اما با افزایش تعداد سرویس ها، مدیریت استقرار، دسترسی، توزیع بار و اجرای Container ها پیچیده تر می شود.

Kubernetes این لایه مدیریتی را فراهم می کند و با استفاده از منابعی مانند Pod، Deployment، ReplicaSet و Service، امکان مدیریت ساخت یافته تر سرویس های یک پلتفرم داده را در اختیار تیم قرار می دهد.

Registry و Repository چه نقشی در استقرار سرویس های داده دارند؟

برای اجرای سرویس ها در محیط های کانتینری، Image ها باید به شکل قابل مدیریت ذخیره شوند و در اختیار محیط های مختلف قرار گیرند.

Registry و Repository این امکان را فراهم می کنند که Image های مورد نیاز نگهداری و مدیریت شوند و در زمان استقرار در اختیار سرویس ها قرار گیرند. این موضوع زمانی اهمیت بیشتری پیدا می کند که چندین سرویس و چند محیط اجرایی مختلف وجود داشته باشد.

تفاوت Node Affinity و Pod Affinity چیست؟

Node Affinity برای تعریف قواعد مربوط به ارتباط یک Pod با Node های Cluster استفاده می شود. در مقابل، Pod Affinity به محل قرارگیری Pod ها نسبت به یکدیگر مربوط است. این تفاوت در طراحی نحوه توزیع Workload های یک Data Platform اهمیت زیادی دارد.

StatefulSet چه مسئله ای را در سرویس های داده حل می کند؟

StatefulSet برای مدیریت Workload هایی طراحی شده است که نیازمند State، هویت پایدار و Storage پایدار هستند. استفاده و مدیریت StatefulSet یکی از اجزای مهم استقرار سرویس های داده روی Kubernetes است و در کنار Storage پایدار، بستر مناسب تری برای اجرای این نوع سرویس ها فراهم می کند.

StorageClass و Provisioner چرا در معماری Storage اهمیت دارند؟

StorageClass امکان تعریف نحوه ارائه Storage به Workload ها را فراهم می کند و Provisioner در فرآیند تأمین این Storage نقش دارد. در برخی معماری ها نیز استفاده از Local Disk و حالت No Provisioner مطرح می شود. انتخاب مدل مناسب Storage یکی از تصمیم های مهم هنگام طراحی یک Data Platform روی Kubernetes است.

CRD چگونه Kubernetes را برای نیازهای تخصصی تر قابل توسعه می کند؟

Custom Resource Definition یا CRD امکان تعریف Resource های جدید را در Kubernetes فراهم می کند. به کمک CRD می توان Kubernetes را فراتر از Resource های استاندارد آن توسعه داد و Resource هایی متناسب با نیاز یک سرویس یا پلتفرم خاص تعریف و مدیریت کرد.

چرا معماری Replica Set در اجرای MongoDB روی Kubernetes اهمیت دارد؟

در استقرار MongoDB، معماری Replica Set و Persistence از موضوعات اصلی هستند. علاوه بر شناخت ساختار Instance های MongoDB، مدیریت Backup، Recovery و منابع نیز اهمیت دارد. MongoDB Kubernetes Operator نیز می‌تواند برای مدیریت این سرویس در محیط Kubernetes مورد استفاده قرار گیرد.

اجرای Spark روی Kubernetes چه تفاوتی با استقرار یک Database دارد؟

Spark ماهیت پردازشی دارد و در Kubernetes موضوعاتی مانند اجرای Job ها، مدیریت منابع، پارامترهای اجرایی و تخصیص منابع اهمیت ویژه‌ای پیدا می‌کنند. ابزارهایی مانند Apache YuniKorn نیز در حوزه مدیریت و تخصیص منابع برای Workload ها کاربرد دارند.

CI/CD در Data Platform چه تفاوتی با استقرار معمول نرم‌افزار دارد؟

در Data Platform علاوه بر کد و Configuration، موضوعاتی مانند Database Migration، تغییر Data Schema، Versioning و انتقال تغییرات میان محیط‌های Development، Staging و Production نیز مطرح هستند. به همین دلیل مفاهیمی مانند Rollback، Promotion و GitOps در طراحی فرآیند CI/CD اهمیت بیشتری پیدا می‌کنند.

از کجا بفهمیم یک Data Platform واقعاً سالم است؟

سلامت پلتفرم فقط با مشاهده وضعیت اجرای سرویس‌ها مشخص نمی‌شود. برای ارزیابی وضعیت باید Metrics، Logs و Traces بررسی شوند و از Health Check، Alerting و Incident Detection نیز کمک گرفته شود. این نوع پایش برای سرویس‌هایی مانند PostgreSQL، MongoDB، Kafka، Spark و ClickHouse اهمیت دارد و امکان تشخیص سریع‌تر مشکلات عملیاتی را فراهم می‌کند.

آیا خرید اقساطی امکان‌پذیر است؟

بله، امکان خرید اقساطی با اسنپ پی فراهم شده است. برای اطلاعات بیشتر می‌توانید با مشاورین مجموعه در تماس باشید یا راهنمای [خرید اقساطی دوره آموزشی با اسنپ پی](#) را مطالعه بفرمایید.

پلتفرم «لحظه‌نگار» چیست؟

لحظه‌نگار یک پلتفرم پخش وبینار و کلاس‌های آنلاین است؛ لحظه‌نگار ابزاری جامع برای مدیریت و اجرای رویدادهای آنلاین با تمرکز بر سرعت و کیفیت بالا است. تمرکز اصلی‌اش ثبت رویدادها، تحلیل‌ها یا روایت‌ها به شکل زنده یا مرحله‌به‌مرحله است.

آیا فیلم دوره رکورد می گردد؟

بله، دسکتاپ و صدای مدرس رکورد خواهد شد و در پلیر اختصاصی اسپات پلیر به همراه کلید لایسنس ارائه خواهد شد. شما در سیستم عامل های ویندوز، اندروید، آیفون (سیب، اناردون)، مک بوک می توانید فیلم را مشاهده کنید.

افرادی که بصورت لایو کلاس را مشاهده می کنند، آیا امکان پرسش و پاسخ دارند؟

شما بصورت چت آنلاین می توانید سوالات خود را بپرسید و مدرس هم سوالات شما را پاسخ خواهد داد. البته توجه داشته باشید که این پروسه پاسخگویی هر ۴۰ دقیقه یکبار خواهد بود تا مدرس رشته کلام از دستش خارج نشود. البته که گروه تلگرامی دوره در اختیار شما است و می توانید سوالات خود را آنجا هم مطرح کنید.

آیا امکان برگزاری دوره های سازمانی وجود دارد؟

بله؛ در نیک آموز امکان برگزاری دوره های سازمانی به صورت تخصصی فراهم شده است. به منظور ثبت درخواست، کافی است اطلاعات خود و دوره سازمانی مدنظر را در **فرم درخواست آموزش سازمانی** ثبت کنید تا ما با شما تماس بگیریم.

می خواهم مشاوره / تدریس خصوصی بگیرم؟

برای اینکه بتوانید در ارتباط با برنامه نویسی، مشاوره / تدریس خصوصی بگیرید، لطفاً **فرم درخواست مشاوره مدرسین** را تکمیل نمایید تا کارشناسان ما با شما تماس بگیرند.

من باز هم سؤال دارم؛ امکان ارتباط با مشاوران نیک آموز وجود دارد؟

بله؛ شما می توانید از مشاوره های **نیک آموز** به عنوان راهنما در مسیر خود استفاده کنید. برای این منظور، لطفاً شماره خود را در فرم مشاوره صفحه دوره وارد کنید تا مشاوران نیک آموز با شما تماس بگیرند.

شماره تماس فروش: **۰۲۱۹۱۰۷۰۰۱۷** - داخلی ۱

شماره تماس پشتیبانی: **۰۲۱۹۱۰۷۰۰۱۷** - داخلی ۲

آیا می‌توانم مشاوره سازمانی برای پروژه دریافت کنم؟

شما می‌توانید با مراجعه به **فرم درخواست مشاوره تخصصی**، از متخصصان نیک آموز مشاوره دریافت کنید و با به‌کارگیری مهارت‌های تجربی تیم ما، در ارتباط با پروژه‌های تخصصی خود، راهنمایی دریافت کنید.

نحوه پشتیبانی دوره به چه صورت است؟

رضایت شما از دوره آموزشی و کمک به رفع مشکلات احتمالی برای ما اهمیت زیادی دارد. به همین دلیل، یک گروه پشتیبانی در تلگرام ایجاد شده است تا شما بتوانید در صورت نیاز، مسائل خود را در این بستر مطرح کنید. تا حداکثر ۴۸ ساعت کاری پس از ثبت نام در دوره، با شما تماس گرفته می‌شود و فرآیند عضویت شما در گروه تلگرام نهایی خواهد شد. توجه شود که در آینده، سیستم تیکتینگ راه‌اندازی می‌شود و فرآیند پشتیبانی از گروه تلگرامی به آن جا منتقل خواهد شد.

ثبت نام آنلاین : دوره Data Platform Engineering