



MizDooni

مقدمه

در این فاز به سمت عملیاتی کردن پروژه حرکت می کنید. چرخه زندگی محصول بعد از توسعه آن پایان نمی یابد و باید بعد از انتشار همواره سیستم تحت نظارت و بررسی باشد. با اضافه کردن ابزار های مشاهده پذیری ما می توانیم در سریع ترین زمان ممکن مشکلات را متوجه شویم و از ایجاد نارضایتی جلوگیری کنیم. همچنین می تواند ابزاری مفید برای تشخیص محصولی که مشکل را ایجاد کرده باشد. برای درک بهتر، گرافی از محصولات را تصور کنید که با هم در ارتباط هستند. اگر مشکلی در فرایندی داشته باشیم که چندین سرویس در آن درگیر هستند، پیدا کردن ریشه مشکل کار دشواری است. اما اگر داده کافی داشته باشیم، میتوان با نگاه به یک داشبرد به سادگی مشکل را ریشه یابی کرد.

پیش نیازها

برای ایجاد این امکان ابزارهای زیادی از جمله prometheus، jaeger، grafana، skywalking و... وجود دارد. پیشنهاد می شود درباره ابزارهای لیست شده جستجویی داشته باشید و مورد کاربرد آنها را بشناسید. در این پروژه ما به سراغ یکی از پر استفاده ترین راه حل ها از این مجموعه - Elastic APM - می رویم. این ابزار به کمک پایگاه داده ی الاستیک^۱ و قابلیت های منحصر به فرد آن امکان ذخیره سازی داده با حجم بالا و واکنشی سریع را برای ما فراهم می سازد. در کنار این دو تکنولوژی از

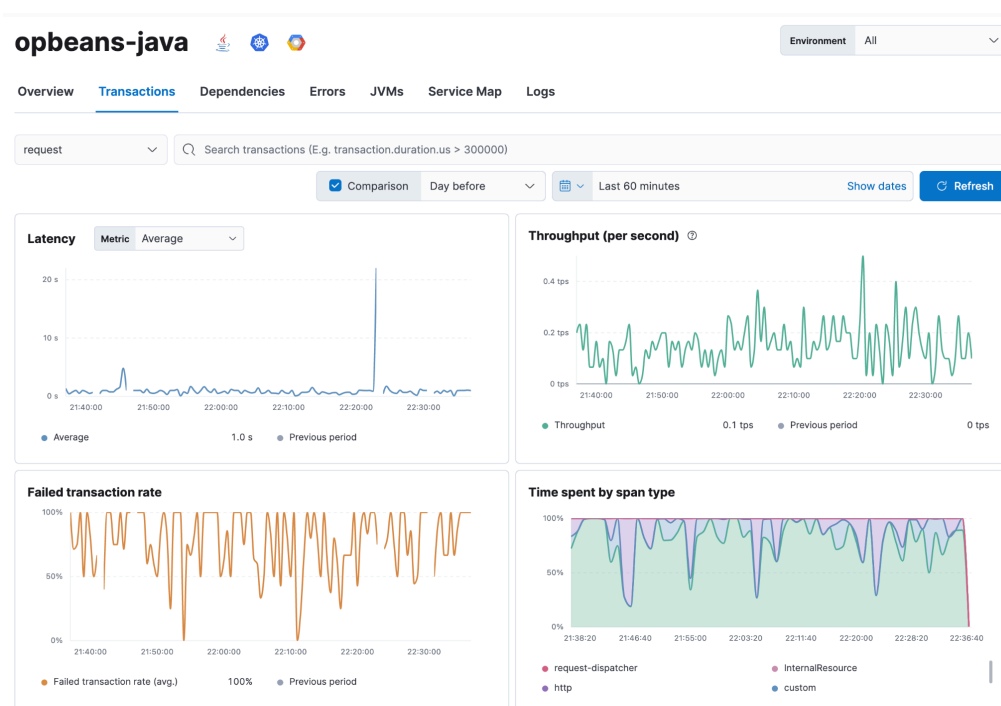
¹ Elastic

پنل کیبانا² برای نمایش داده‌ها و داشبوردها استفاده خواهیم کرد. بدین منظور نیاز است تا این سرویس را در سیستم خود راه اندازی کنید. با اینکه امکان نصب این سرویس‌ها به صورت مستقیم وجود دارد اما پیشنهاد می‌شود برای سهولت از این [docker-compose](#) استفاده کنید.

پیکربندی خروجی‌های پیش فرض

در این فاز قصد داریم Mertz، Trace و Log های سیستم را در APM مجتمع کنیم تا دیدی کلی از نرم افزار در حال اجرا داشته باشیم. بسیاری از این داده‌ها به صورت خودکار توسط پلاگین‌های Instrumentation موجود برای ما تامین می‌شود و تنها نیاز است تا ویژگی‌های خاص خودمان را به آن اضافه کنیم.

برای این کار می‌توانید از Java APM agent استفاده کنید. برای راهنمای راه‌اندازی این ابزار می‌توانید به [این](#) لینک مراجعه کنید. بعد از انجام این پیکربندی اطلاعاتی شامل منابع مصرف شده، وابستگی‌های بین مولفه‌ها (که در این پروژه پایگاه داده و نرم‌افزار شما می‌باشد)، تعداد تراکنش‌ها و تاخیر آن در اختیار شما قرار می‌گیرد و داشبوردی مانند زیر در قسمت APM کیبانا اضافه خواهد شد.



² Kibana

اضافه کردن داده های شخصی سازی شده

تا اینجا موفق شدیم داده های پیش فرض که حداقل داده مورد نیاز برای قابل مشاهده کردن سیستم هستند را در اختیار داشته باشیم. اما از آن جایی که هر نرم افزاری منحصر به فرد است، نمی توان به این حداقل ها بسنده کرد. در این مرحله می خواهیم تعداد Metric و Trace به سیستم اضافه کنیم تا بتوان بهتر درباره نرم افزار در حال اجرا نظر دهیم. این داده ها نه تنها کمک شایانی برای نگهداری و پشتیبانی نرم افزار می کنند، بلکه می توانند در زمان بروز مشکل هزینه کشف ایراد را کاهش دهند. به این دلیل که رفتار داخلی سیستم به طور شفاف در اختیارمان قرار می گیرد. این امکان توسط ماژولی به نام [APM Agent Plugin API](#) در اختیار قرار می گیرد.

● اضافه کردن Metric

پیدا کردن متریک های بهینه یکی از ظریف ترین وظایف توسعه دهنده نرم افزارها است. برای نرم افزار خود ۳ متریک ارایه دهید و داده آن را از طریق Agent انتخاب شده به APM بفرستید. برای راه اندازی متریک ها می توانید از [این](#) آموزش کمک بگیرید.

● اضافه کردن Trace

در این بخش نقاطی از منطق برنامه که دارای اهمیت می باشند و اطلاع داشتن از اجرای آن ها در حین یک درخواست اهمیت دارد را هدف قرار می دهیم. با اضافه کردن Span های دستی به این دسته از کدها امکان مشاهده اجرای آن ها را فراهم می کنیم. به طور کلی تمامی متدهای برنامه که دارای منطق هستند این اهمیت را دارند. اما از آنجایی که داده بیش از حد موجب سردرگمی و سخت تر شدن فرایند بررسی می شود، بهتر از تا حد ممکن متدهای کمتری را Trace کنیم. متدهای مهم پروژه خود را انتخاب کنید و به کمک [این](#) آموزش آن ها را Trace کنید.

Software Engineering Best Practices

تفاوت های Trace، Metric و Log

وقتی صحبت از Observability میشود همواره با سه مفهوم Metric, Trace و Log درگیر هستیم. با اینکه هر سه آن ها داد های اضافه ای هستند که از نرم افزار منتشر می شود تا دیتای بیشتری درباره کارکرد آن داشته باشیم، تفاوت هایی دارند. در حقیقت این سه مفهوم سه دیدگاه متفاوت نسبت به وضعیت نرم افزار هستند که برای کارکرد های متفاوت ایجاد شده اند.

در ابتدا Metric ها هستند که معمولا داده های بسیار سبک و عددی هستند که اطلاعات را به صورت تجمیعی در اختیار کاربر قرار می دهند. به عنوان مثال همانطور که دیدید، مقدار مصرف حافظه اصلی و واحد پردازشی از مهمترین Metric های هر سیستم نرم افزاری هستند. این داده ها به دلیل اینکه نیازی به تحلیل ندارند، بهترین گزینه برای قرار دادن در داشبرد های کلی هستند. همچنین بهترین گزینه برای تنظیم هشدار ها هستند زیرا با تعریف بازه های عددی می توان وضعیت آن ها را توصیف کرد.

از طرف دیگر Trace ها داده بیشتری را در خود دارند. وظیفه آنها نمایش گام های یک فرایند است. این داده بیشتر برای توسعه دهنده ها کاربرد دارد و به کمک آن می توانند در صورت بروز یک مشکل، وضعیت سیستم در آن لحظه را مشاهده کنند و به سادگی ریشه اصلی بروز مشکل را پیدا کنند. به دلیل اینکه تمام فرایند در یک Trace وجود دارد، نمی توان اطلاعات زیادی را در آن نگه داشت زیرا در این صورت حجم داده ها بالا رفته و مدیریت آن نیازمند منابع بسیاری خواهد بود.

در انتها Log ها را داریم. این نوع از داده، یک نقطه خاص را مد نظر قرار می دهد و از رخداد یک اتفاق در سیستم خبر می دهد. به دلیل اینکه دامنه آن محدود است، می توانیم جزئیات بیشتری از رخداد را در آن نگه داریم و در حقیقت گزارش رخداد است. در این نوع، رعایت شدت یا سطح Log بسیار مهم است. زیرا به ما کمک می کند داده هایی که نیاز نداریم را فیلتر کنیم. همچنین می توان در اکثر مواقع تنها سطح های دارای اهمیت را ذخیره سازی کرد و تنها در صورت نیاز لاگ های جزئی تر را فعال کرد. این موضوع به مدیریت منابع مصرفی نرم افزار کمک شایانی می کند.

Git Commit

همان‌طور که در پروژه اول توضیح داده شد، کامیت‌ها اهمیت زیادی در توسعه پروژه‌های نرم‌افزاری دارند. در این پروژه نیز باید مواردی که در پروژه اول گفته شدند، رعایت شوند. رعایت این قسمت، بخشی از نمره شما را در این پروژه تعیین می‌کند.

نکات پایانی

- این تمرین در گروه‌های حداکثر دو نفره انجام می‌شود و کافی است که یکی از اعضای گروه Hash مربوط به آخرین کامیت پروژه را در سایت درس آپلود کند. در هنگام تحویل، پروژه روی این کامیت مورد ارزیابی قرار می‌گیرد.
- حتما کاربر [IE-S03](#) را به پروژه خود اضافه کنید.
- ساختار صحیح و تمیزی کد برنامه، بخشی از نمره‌ی این فاز پروژه‌ی شما خواهد بود. بنابراین در طراحی ساختار برنامه دقت به خرج دهید.
- هدف این تمرین یادگیری شماست. لطفاً تمرین را خودتان انجام دهید. در صورت مشاهده‌ی مشابهت بین کدهای دو گروه، از نمره هر دو گروه مطابق سیاستی که در کلاس گفته شده است کسر خواهد شد.
- سوالات خود را تا حد ممکن در گروه درس مطرح کنید تا سایر دانشجویان نیز از پاسخ آنها بهره‌مند شوند. در صورتی که قصد مطرح کردن سوال خاص‌تری داشتید، از طریق ایمیل با طراحان این تمرین ارتباط برقرار کنید.