alert types and params

APM and Uptime alert parameter schemas, per customer request.

Note that to see how these parameters are used, compared to creating them via the UI, create them via the UI, then you can list the params that got created using the alerting HTTP APIs:

```
curl -k https://elastic:changeme@localhost:5601/api/alerts/_find?per_page=10000 | json
```

I scraped the definitions from the source code, which uses the Kibana config-schema package for declaring the schema. Documentation for this package is here:

https://github.com/elastic/kibana/blob/master/packages/kbn-config-schema/README.md

`apm.transaction_duration`

```
schema.object({
  serviceName: schema.string(),
  transactionType: schema.string(),
  windowSize: schema.number(),
  windowUnit: schema.string(),
  threshold: schema.number(),
  aggregationType: schema.oneOf([
    schema.literal('avg'),
    schema.literal('95th'),
    schema.literal('99th'),
  ]),
  environment: schema.string(),
});
```

`apm.transaction_duration_anomaly`

```
schema.object({
  serviceName: schema.maybe(schema.string()),
  transactionType: schema.maybe(schema.string()),
  windowSize: schema.number(),
  windowUnit: schema.string(),
  environment: schema.string(),
  anomalySeverityType: schema.oneOf([
    schema.literal(ANOMALY_SEVERITY.CRITICAL),
    schema.literal(ANOMALY_SEVERITY.MAJOR),
    schema.literal(ANOMALY_SEVERITY.MINOR),
    schema.literal(ANOMALY_SEVERITY.WARNING),
  ]),
});
```

`apm.error_rate`

```
schema.object({
  windowSize: schema.number(),
  windowUnit: schema.string(),
  threshold: schema.number(),
  serviceName: schema.maybe(schema.string()),
  environment: schema.string(),
});
```

`apm.transaction_error_rate`

```
schema.object({
  windowSize: schema.number(),
  windowUnit: schema.string(),
  threshold: schema.number(),
  transactionType: schema.maybe(schema.string()),
  serviceName: schema.maybe(schema.string()),
  environment: schema.string(),
});
```

`xpack.uptime.alerts.monitorStatus`

```
schema.object({
  availability: schema.maybe(
    schema.object({
      range: schema.number(),
      rangeUnit: schema.string(),
      threshold: schema.string(),
    })
  ),
  filters: schema.maybe(
    schema.oneOf([
      // deprecated
      schema.object({
        'monitor.type': schema.maybe(schema.arrayOf(schema.string())),
        'observer.geo.name': schema.maybe(schema.arrayOf(schema.string())),
        tags: schema.maybe(schema.arrayOf(schema.string())),
        'url.port': schema.maybe(schema.arrayOf(schema.string())),
      }),
      schema.string(),
    ])
  ),
  // deprecated
  locations: schema.maybe(schema.arrayOf(schema.string())),
  numTimes: schema.number(),
  search: schema.maybe(schema.string()),
  shouldCheckStatus: schema.boolean(),
  shouldCheckAvailability: schema.boolean(),
  timerangeCount: schema.maybe(schema.number()),
  timerangeUnit: schema.maybe(schema.string()),
  // deprecated
```

```
  timerange: schema.maybe(
    schema.object({
      from: schema.string(),
      to: schema.string(),
    })
  ),
  version: schema.maybe(schema.number()),
  isAutoGenerated: schema.maybe(schema.boolean()),
})
```

`xpack.uptime.alerts.tls`

```
schema.object({})
```

I guess this alert takes no parameters!

`xpack.uptime.alerts.durationAnomaly`

```
schema.object({
  monitorId: schema.string(),
  severity: schema.number(),
}
```