Monolithic Architecture

Browser → Apache server → WAR/Tomcat → DB

Outcomes:
- Less Immutability
- Less CI/CD capabilities
- Less Automation capabilities
Cloud Native Architecture

Technology
- 12 factor
- Micro services
- Rest API

Outcomes
- Immutable
- CI/CD
- Automation

Values
- Cloud first
- Small steps
- Integration
### Spring boot model for Cloud Nativity

<table>
<thead>
<tr>
<th>Component</th>
</tr>
</thead>
<tbody>
<tr>
<td>Configuration</td>
</tr>
<tr>
<td>Service discovery</td>
</tr>
<tr>
<td>Circuit breakers</td>
</tr>
<tr>
<td>Routing and messaging</td>
</tr>
<tr>
<td>API Gateway</td>
</tr>
<tr>
<td>Tracing</td>
</tr>
<tr>
<td>CI &amp; Pipelines and Testing</td>
</tr>
</tbody>
</table>
Let's dig into code

- Building cloud native Java application
  - Use Spring boot cloud components
    - Set up -> start.spring.io
    - What you need → Java 8, maven 3.1 or above
    - IDE, might need Spring boot CLI,
    - might need Spring Cloud CLI
Why?

- Why Spring
- Why Jar, not war
- Why by-name links for searching services
Demo

- The Config Server
- Service Registration and Discovery
- Edge Services: API gateways (circuit breakers, client-side load balancing)
- To the Cloud!
- Security
- Consumer Driven Contract Testing and more!