Cloud Native
Meets Legacy
**Current Cloud Migration Approaches**

Cloud is maturing and enterprises are looking to **migrate large-scale applications to the cloud** in order to increase agility and elasticity, reduce IT spend, and increase uptime.

**Code that is Native**

Designed from scratch to maximize design for scalability, portability, performance but requires app rewrite

**Code that is Compatible**

The application gains infrastructure value of the cloud but does not generate maximum value possible

The standard approaches towards application migration to the cloud have become either to rewrite a cloud native application from scratch or to transfer existing application assets to a compatible cloud service.
Leverage Cloud Native Designs

Missing between the cloud native and cloud compatible approaches, is an approach that can leverage cloud native designs into existing software assets.

**Code that is Native** is designed from scratch to maximize design for scalability, portability, performance but requires app rewrite.

**Code that Leverages** attributes of the Cloud can maximize existing software investments and exploit Cloud designs.

**Code that is Compatible** with the Cloud gains infrastructure value but does not generate maximum value possible from Cloud.

However, applications are still faced with the daunting task of measuring the cloud readiness of their application and understanding the level of effort needed to retrofit the application to leverage cloud native designs.
Cloud Native Anti-Patterns

Building on experience from working on complex cloud implementation projects and industry research, the Deloitte Application Architecture team has identified a list of cloud native code anti-patterns* across six key categories.

Our ruleset is standardized to the code anti-patterns advocated by Heroku’s Twelve-Factor App methodology and extends it to additional cloud native categories.

* We define cloud native anti-patterns as common code characteristics that hinder the ability of the application to maximize on cloud benefits.
Cloud Native Portability Anti-Patterns

6 Key Metrics

- Portability
- Availability
- Performance
- Source Code
- Code Quality
- Vulnerability
- Scalability

Portability
Ability to seamlessly port application from one platform to another

Sample Rules

- Found instance of database operation(s) using the MS SQL provider
- Found instance of hardcode locale specific strings
Cloud Native Performance Anti-Patterns

6 Key Metrics

- Portability
- Availability
- Code Quality
- Vulnerability
- Scalability
- Performance

**Performance**
Maintain application responsiveness under increased load

**Sample Rules**
- Found instances of disposable fields that should be disposed
- Detect usage of synchronous I/O operations in the application code
Cloud Native Vulnerability Anti-Patterns

6 Key Metrics

- Portability
- Availability
- Performance
- Code Quality
- Scalability
- Source Code
- Vulnerability

Vulnerability
Capability of a system to prevent malicious or accidental actions outside of the designed usage

Sample Rules

- Detects usage of Unencrypted FTP/HTTP sessions in the code
- Detects usage of System.loadlibrary in the code, which can be used by the attacker to inject arbitrary code into the application and execute it
Cloud Native Scalability Anti-Patterns

6 Key Metrics

- Source Code
- Portability
- Availability
- Performance
- Code Quality
- Vulnerability
- Scalability

**Scalability**
Potential of the application to accommodate range of capabilities

**Sample Rules**

- Found instance of ASP.NET Session (HttpSessionState)

- Identify the usage of machine specific socket (SSL or otherwise) usage in the application code
Cloud Native Availability Anti-Patterns

6 Key Metrics

Source Code

Availability

Portability

Performance

Vulnerability

Scalability

Code Quality

Measure of an application code’s technical debt

Sample Rules

Abstract types should not have constructors

Found instance of .NET Catch IO Exceptions (System.IO.IOException)
6 Key Metrics

Availability
Ability of application code to mask service outage faults

Sample Rules

Detect usages of any local files in the application

Identify the hardcoded IPs in the application code
Automating the Code Analysis

Manually reviewing application code for cloud native anti-patterns is time consuming, resource intensive, and highly error prone.

Deloitte’s Stormfury scans millions of lines of application code for cloud anti-patterns in minutes.

Analysis of **800+** rules through industry research

Implementation of **130+** cloud anti-pattern rules

Classification of barriers across the **6** key cloud native metrics

Dashboards to visualize code quality insights
Static Code Analysis

The output of application code for cloud anti-patterns can yield key insights across all levels of the IT organization.

6 Key Cloud Native Metrics + Quality Dashboard = Stakeholder Insights

**IT Leadership**
- Should we redesign, rewrite, or replace the application for Cloud Native?
- How much effort and funding is required to redesign the code to be Cloud Native?

**Technical Architects**
- Where are the opportunities to redesign the code base and what are the complex areas of the system?
- What are the top areas that can leverage the cloud?

**Developers on Check-in**
- Have I created any new issues? If yes, where in the code?
- How do I address the issue?

An automated cloud native code anti-pattern scan will lower the cost, reduce the risk, and improve efficiency of a cloud migration effort.
Case Studies of automated code analysis for cloud anti-patterns

<table>
<thead>
<tr>
<th>Client</th>
<th>Objective</th>
<th>Solution</th>
<th>Value Delivered</th>
<th>Client Testimonials</th>
</tr>
</thead>
<tbody>
<tr>
<td>Global Financial Services Client</td>
<td>Accelerate journey to AWS platform and adopt a cloud-first policy.</td>
<td>Assess individual applications’ cloud suitability and increase the pipeline of cloud candidates based on tech stacks and application architecture.</td>
<td>Identified over 1.1K bugs related to cloud migration areas in 320K+ LOC.</td>
<td>Really happy with what the product has found.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Accelerate the migration and decommissioning of their on-premise analytics platform and shift to AWS EMR.</td>
<td>Detailed issue descriptions, severity levels, and mitigations steps enabled development team to prioritize findings and begin fixing them in one week.</td>
<td>It helped us target the big ticket findings and fix them immediately</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Accelerated areas to look at in the tool without having to do manually search though hundreds of source files.</td>
<td></td>
</tr>
<tr>
<td>Global Foundation for Life Sciences and Healthcare</td>
<td>Rapidly adopt cloud infrastructure for their drug development operations in a next-generation R&amp;D model. This migration effort is the key to align the data, people, and processes within R&amp;D organizations.</td>
<td>Assess the application’s cloud maturity based on level of inherited technical debt.</td>
<td>Identified over 40 bugs in various quality attributes related to Cloud migration in a 200K+ LOC application.</td>
<td>Extremely satisfied with the level of analysis and key findings discovered by Stormfury.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Category level issue counts and remediation effort enabled funding decisions.</td>
<td>Helped track and rapidly correct deviations from coding standards</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Effective resolution of multiple issues related to exception handling, class loaders and other cloud anti-patterns in the code.</td>
<td>Helped make the application more suitable for cloud</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Memory overheads were reduced – Unused variables and objects were removed from the application</td>
</tr>
<tr>
<td>Large State Health &amp; Human Services Agency</td>
<td>Migrate a large state based marketplace with complex functionality and large code base to a cloud infrastructure.</td>
<td>Proactive identification of potential issues that hinder migration</td>
<td>Identified over 700 bugs across various cloud native quality attributes in 900K+ LOC.</td>
<td>Stormfury has helped the client mitigate a key unknown in its cloud migration project.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Educated development team on cloud native coding practices.</td>
<td>Stormfury’s easy to understand descriptions and mitigation steps made it really easy for developers to address the 700+ cloud migration issues.</td>
</tr>
</tbody>
</table>
Reference

Stormfury Video

https://players.brightcove.net/5472387872001/S1dWZWLQtb_default/index.html?videoId=5760188439001