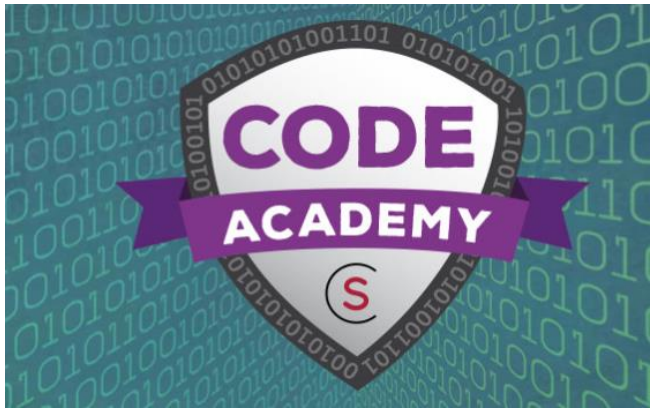


Day 14: Cloud services continue

CODE BOOTCAMP VAASA 2021

JANI JÄRVINEN, INTERTECHNO TRAINING OY

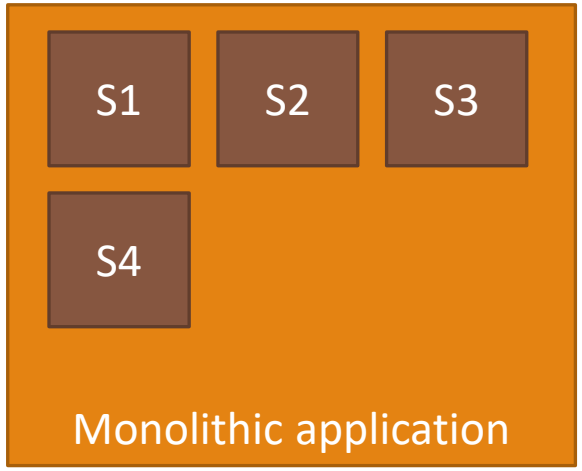
Developers and cloud services

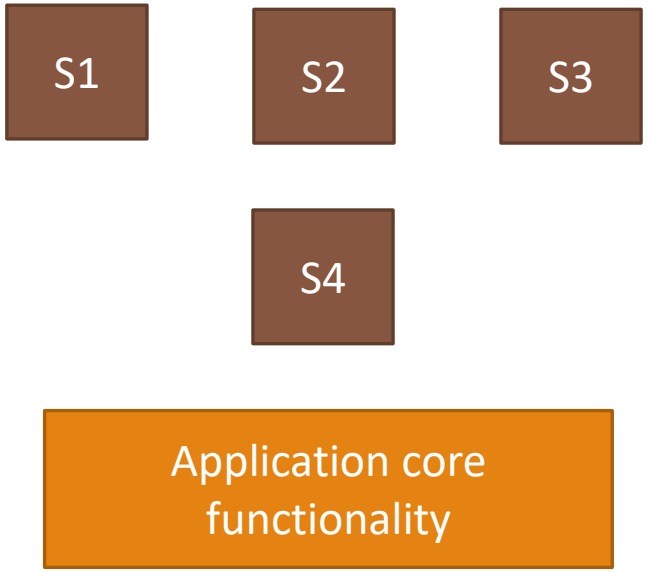
Useful services for developers

- Classical IT resources -> for testing and supporting development work
  - Virtual machines
  - Disk space and databases
  - Integration services
- Automatic testing
  - Running automated tests
  - Unit tests and functional testing
  - Scripting
- Application production environments
  - Virtual machines
  - Databases
  - Web servers and serverless functions

The serverless architecture

- Modern applications are architected so that there are more smaller pieces instead of just one large application
  - The allows easier maintenance and upgrades of the application
  - Terms: micro services and serverless architecture





Azure Functions

- Microsoft's Azure cloud service contains a feature called Functions, which allows you to use the latest serverless architectures
- The aim is to develop small but independent functionalities and share them using HTTP APIs
- The pricing of Functions is based on the user computing time, memory consumption and the number of HTTP requests processed
- You can write Functions with supported programming languages; at this time C#, JavaScript, Python, Java, PHP and F# are supported
- Actual development can be made with a development tool like Visual Studio, or directly in Azure's web portal

DevOps

Software development today

- Processes
  - Especially web and mobile application development is a fast-pacing target
  - Business requirements are also on the change, based on the global digital transformation
  - Agile methods are commonly used
- Tooling
  - More and more tools are needed to successfully implement software
  - Ready-made, often open-source libraries are used as a base for development work
  - To succeed in modern development, process automation is key
  - Automatic testing is often an underestimated part of the process
- People
  - Focus on creativity and less on repeating, mundane tasks

What is DevOps?

- DevOps → Development & Operations
- A model for the production and deployment of applications using automation and technology
- Provides opportunities to accelerate the time-to-marked of finished software products
- Allows two traditionally separate camps – developers and IT people – to discuss with each other
- There is no single tool or set of tools, but a way to work by utilizing a variety of tools and software
- Covers the entire application lifecycle
- Related terms: DevSecOps, MLOps, WinOps, …

DevOps thinking

- Allow automation to handle routine work
  - Give people more time to focus on where they are good at
- Rather, minor updates often instead of larger ones more rarely
- Applications are modular and implementation transparent
- Exploiting virtualization and environmental flexibility for future needs
  - Cloud services provide a good opportunity for this, but DevOps is not only intended for cloud-only applications
- The customer always knows what's going on
  - As the progress made is visible to everyone, there will be no three-month periods of “uncertainity”

Why Automate Testing?

- Basic goal: reduce errors and produce ready-made software faster and at less cost
- However: automating tests increases the workload associated with writing code and producing new features
- With automatic testing, the goal is usually
  - Routine work reduction → the opportunity to focus on more difficult problems
  - Improvement of quality, especially in regression testing → what has worked in the past, will continue to work
  - Speeding up testing → faster or more efficient version for customers
- Other benefits
  - Risk management, job satisfaction, customer experience

Unit Tests

- Unit tests should be
  - Simple and quick to run
  - Touches only one class (and one method)
  - Self-contained and easy to write
  - Documented
- A unit test does not
  - Access or modify the database, user interface, network connections, or files
  - Take a stand on performance, security, or, for example, scalability
- Unit test frameworks (engines) can be used to run other types of tests than unit tests only
- For example, integration tests with the database can be well run on the same engine as the unit tested