Building microservices on the cloud

After you complete this unit, you should understand:

- What are microservices?
- What to avoid with microservices
- Elements of microservices architecture
- Microservices and DevOps
- Developing microservices in Bluemix

What are microservices?

- Microservices is an architectural style in which large complex applications are composed of smaller services.
- Microservices:
  - Can be deployed independently of one another
  - Are loosely coupled
  - Have bounded context
  - Focus on one task only
  - Are easy to optimize
  - Can be developed in any language
  - Communicate through language-neutral APIs, for example, REST and HTTP
  - Each has its own repository and delivery pipeline for builds and deployment

Comparing monolithic and microservices architecture

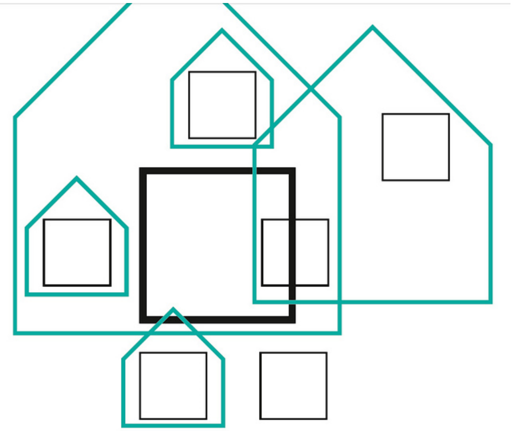
Category	Monolithic architecture	Microservices architecture
Code	Single codebase	Multiple codebases
Understandability	Often confusing, harder to maintain	Better readability, easier to maintain
Scaling	Need to scale entire application though bottlenecks are localized	Can scale bottle necked services without scaling the entire application
Deployment	Complex deployments with maintenance windows and scheduled downtimes	Each microservice can be deployed individually with minimal to zero downtime
Language	Typically developed in one programming language	Each microservice can be developed in different programming languages

What to avoid with microservices

- Don’t start new applications with microservices
- Don’t think about microservices without DevOps
- Don’t manage your own infrastructure
- Don’t create too many microservices

Elements of microservices architecture

- Business oriented
- Design for failure
- Bounded context
- Decentralized data management
- Discoverability
- Inter-service communication design
- Dealing with complexity
- Evolutionary design



from *Intersection* by Milan Guenther, www.intersectionbook.com

Designing microservices

- Use design thinking to scope and identify microservices
- Choose the implementation stack
- Size the microservices

Microservices and DevOps

- Why you should use DevOps
- DevOps capabilities for microservices
- Microservices governance
- Testing strategies for microservices

Developing microservices in Bluemix

- Developing microservices by using Bluemix DevOps
- Communication, session persistence and logging in Bluemix
- Controlling access and visibility of endpoints
- Avoiding failures
- Versioning

Recap

- In this session, you learned:
  - When to use or not to use microservices
  - How to componentize your apps through microservices
  - How a DevOps approach is critical to building a microservices architecture
  - How a DevOps delivery pipeline can help you develop and manage microservices

Related links

- Build a sample microservice in Bluemix
  - https://developer.ibm.com/bluemix/2015/01/19/microservices-bluemix/
- Sample online store application using microservices and Bluemix
  - https://developer.ibm.com/bluemix/2015/03/16/sample-application-using-microservices-bluemix/