

Adaptive Software Systems

GS/EECS 6432

Marin Litoiu

Department of Electrical Engineering and Computer Science
York University

mlitoiu@yorku.ca

<http://www.ceraslabs.com>

Autonomic Computing

Autonomic Computing

- ***An approach to self-managed computing systems with a minimum of human interference.***
- ***The term derives from the body's autonomic nervous system, which controls key functions without conscious awareness or involvement.***
- ***Means***
 - Self configuring
 - Self healing
 - Self optimizing
 - Self protecting



Autonomic Computing

Self-Configuring

- Ability to dynamically configure itself "on the fly" and initialize itself in the context of the overall system; includes the ability to influence relevant changes in other products in the environment.

Self-Healing

- Ability to recover from a failing component by first detecting improper operations (either proactively through predictions or otherwise) and then initiating corrective action without disrupting applications.

Self-Optimizing

- Ability of systems or components to efficiently maximize resource allocation and utilization to meet end-user needs without human intervention.

Self-Protecting

- Ability of a component to detect hostile or intrusive behavior as it occurs and take autonomous actions to make itself less vulnerable.

Content

- **The Complexity Problem**
- **Introduction to Autonomic Computing**
- **Autonomic Computing**
 - Historical Perspective
 - Autonomic Elements
- **Examples**

The Complexity Problem

**Build a system used by millions of people each day
administered and managed by a half-time person**

— Jim Gray, Microsoft Research

— Jim Gray, Microsoft Research

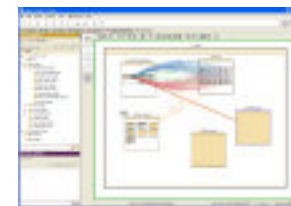
Categories of Complexity

■ Algorithms & Data Structures Development & Maintenance

- Time
- Space

$$O(n \log n)$$

- Logical
- Structural
- Comprehensibility



■ Usage

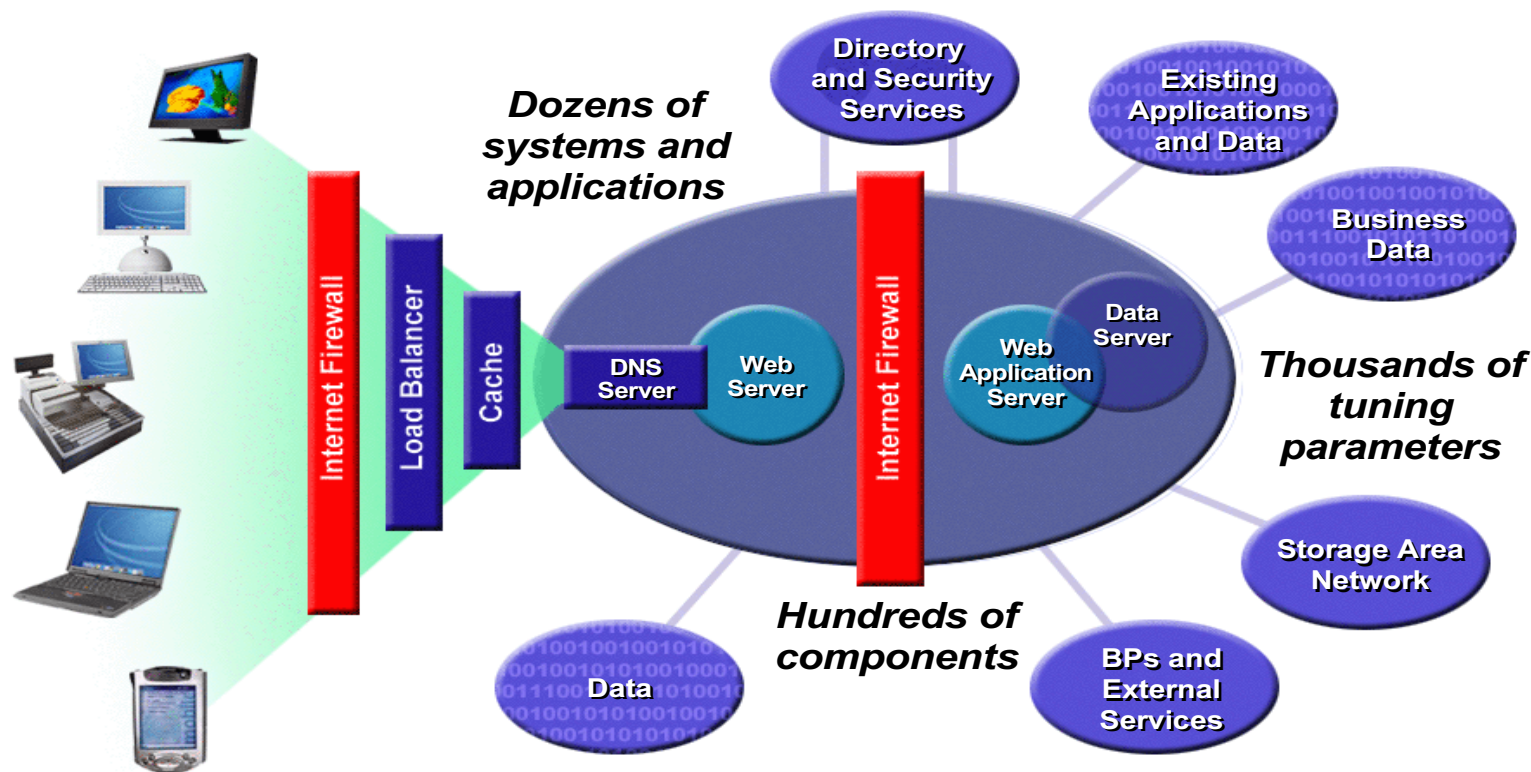
	Novice	Average	Expert
Install			
Configure			
Administer			
Use			

Complex Heterogeneous Environment

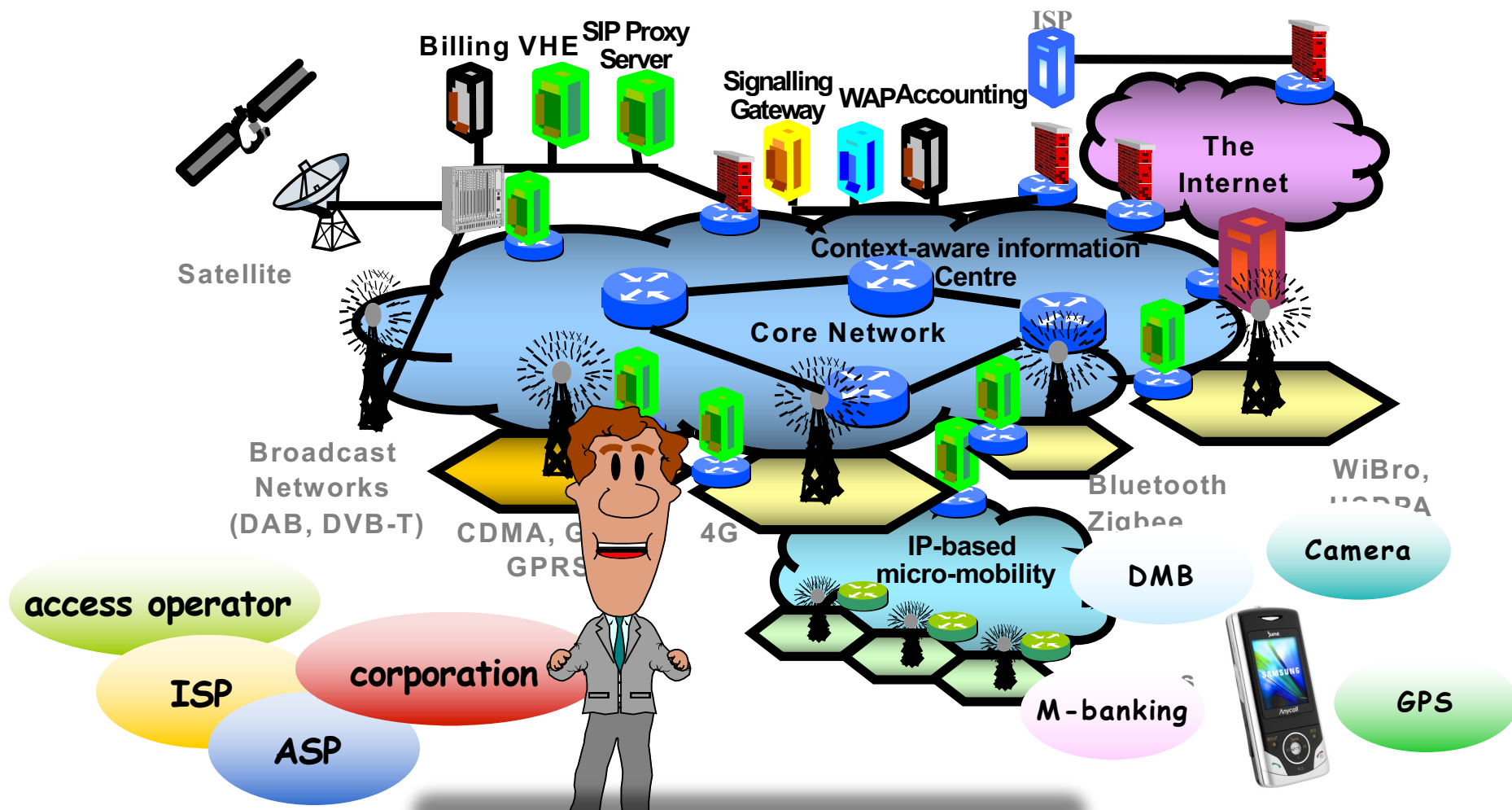


Alan Ganek, VP IBM Autonomic Computing, 2003

Complex Heterogeneous Infrastructure



Complexity of Network Environment

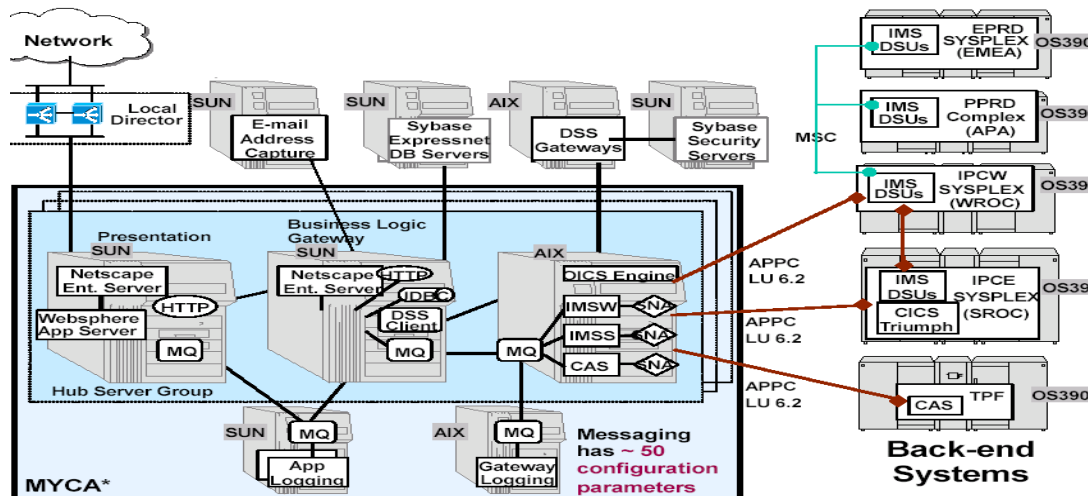


Growing Complexity

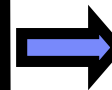
- **Very large scales**
 - Million of entities
- **Amorphous structures/behaviors**
 - P2P, bus, hierarchical architecture
- **Dynamic**
 - Entities join, leave, move, change behavior
- **Heterogeneous**
 - Capability, connectivity, reliability, guarantees, QoS
- **Lack of common/complete knowledge**
 - Types, availability, connectivity, protocols, semantics



Complexity...a typical Enterprise system



- Application Server: ~100 configuration parameters
 - Several applications, hundreds of servlets, tens EJBs
- Web Server: ~20 configuration parameters, servers thousands of web artifacts
- Messaging: ~50 configuration parameters
- DBMS, TCP/IP, Operating Systems.....



x 5 values/parameter
= ~5¹⁷⁰ settings

Business Challenges

Up to 40% of today's outages result from operator errors



25-50% of time is spent on problem determination and resolution

Outages of business-critical systems cost up to \$2.8B per year



Poorly documented legacy applications make it painful to diagnose and resolve complex cross-product problems

The skills needed to do manual cross-product problem determination are scarce and expensive

4 out of 5 IT dollars spent on operations, maintenance, and minor enhancements

New applications get delayed by maintenance of diverse existing systems

Managing complex, heterogeneous environments



What Is Autonomic Computing?

Self-managing systems that ...

Don't make your customers work on their technology; make technology work for them

Increase Responsiveness

Adapt to dynamically changing environments

Business Resiliency

Discover, diagnose, and act to prevent disruptions

Operational Efficiency

Tune resources and balance workloads to maximize use of IT resources

Secure Information and Resources

Anticipate, detect, identify, and protect against attacks

Self-Configuring

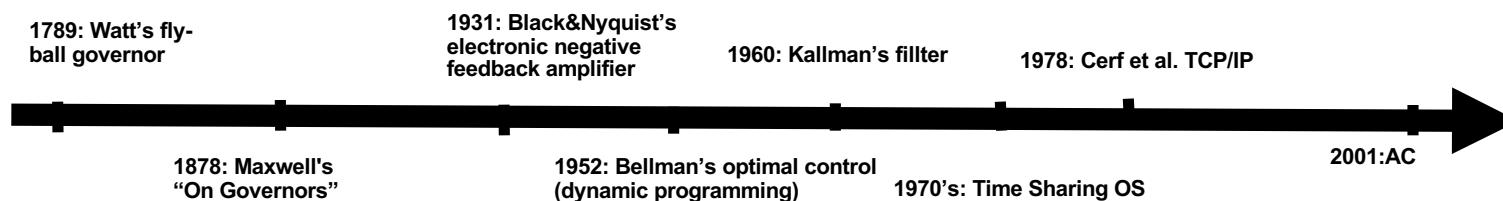
Self-Healing

Self-Optimizing

Self-Protecting

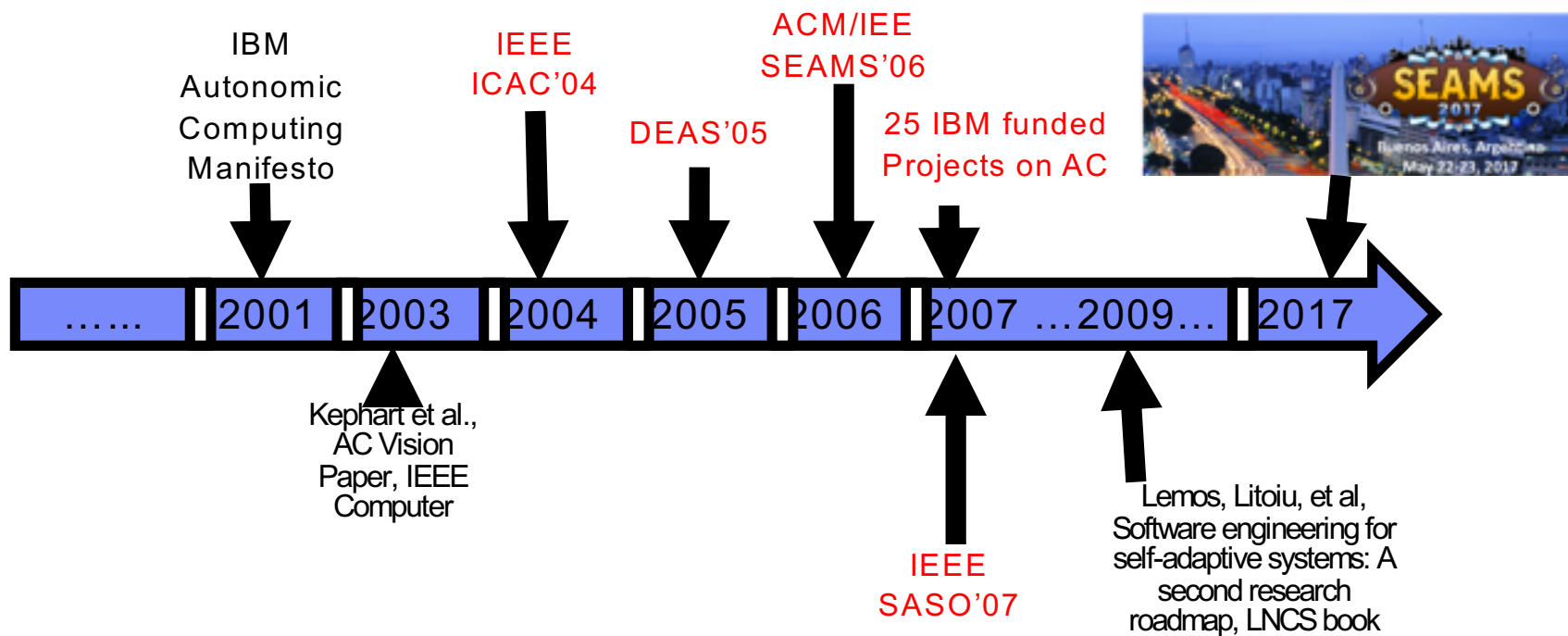
Evolutionary Path ...

Historical Perspective



- **Operating Systems: separate execution environments to self-protect. When a program fails, it does not affect the rest**
- **TCP/IP protocol**
 - It is self-protecting, self-healing, self-optimizing (shortest route, routing tables, error detection and correction mechanisms)
 - Enables self-configuring
- **Workload managers (Dynamic Load Balancing) and schedulers - self-optimizing**
- **Garbage collection (self-healing)**
- **IBM's zSeries:**
 - Duplicated CPUs: CPU error detection and recovery
 - Decades of MTBD

Adaptive Software Timeline



The Automation Conundrum

- **Over the past 50 years, computer systems have had a huge capacity to automate**
 - Enormous variety of tasks
 - Cost per task greatly reduced
 - Incalculable benefits
 - Unprecedented success
- **Key challenges**
 - Further declines in task costs by traditional methods are subject to the law of **diminishing returns**
 - The **complexity** of infrastructure management threatens to outweigh the benefits of further automation



A. Spector, VP IBM Services and Software Research, 2003

Grand Challenge

- **Today's computing systems are amazingly complex, and require daunting expertise and patience just to get them running and keep them running**
- **The increasing system administration will become a major barrier to deploying and maintaining large computing systems**

Autonomic Computing Vision

Autonomic Computing is really about making systems self-managing ...

—Paul Horn, IBM Research, 2001

—Paul Horn, IBM Research, 2001

Mandatory Reading....

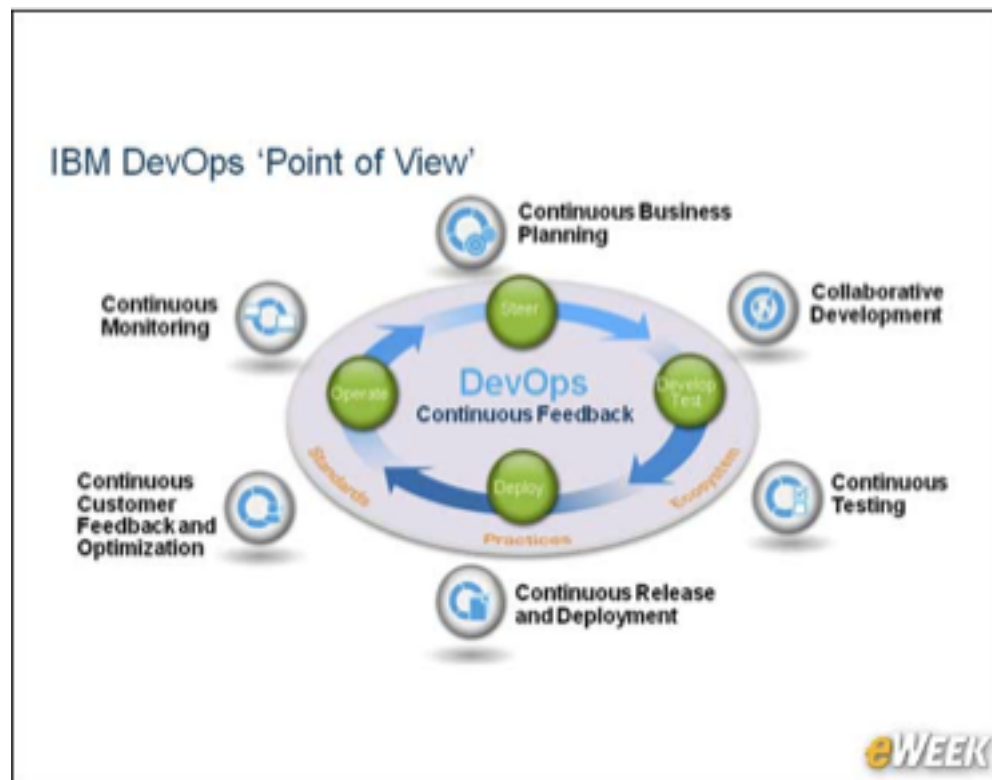
- (all read) J. O. Kephart and D. M. Chess, "The vision of autonomic computing," in *Computer*, vol. 36, no. 1, pp. 41-50, Jan 2003.
doi: 10.1109/MC.2003.1160055

To Explore Further Conferences and Journals

TASS	ACM Transactions on Autonomous and Adaptive Systems (TAAS)
ICAC	IEEE International Conference on Autonomic Computing
SASO	Self-Adaptive and Self-Organizing Systems
SEAMS	Software Engineering for Adaptive and Self-Managing Systems
CASCON	...

Intro to IBM Development and Operations- >DevOps

- **DevOps: promotes the idea of strong collaboration among Development and Operations teams**
- **NoOps means developers can code, deploy, manage and maintain the application**
 - automated systems (PaaS like Cloud Foundry, BlueMix) manage app lifecycles; no Operations teams
- **Adaptation is one of the main aspects of DevOps/Noops**
- **Microservices and containers**
 - Fine granular access
 - Provisioning done in seconds



Intro to IBM Cloud

■ Overview of IBM Cloud

- Get an IBM Cloud account, <https://onthehub.com/ibm/>
- Infrastructure as a Service, Platform as a Service
- Develop and Deploying a hello application (Java+Liberty)
 - With DevOps Toolchain
- Auto-Scaling an applications with DevOps Auto-scaling
 - <https://console.bluemix.net/docs/services/Auto-Scaling/index.html#get-started>

Creating and deploying a web application

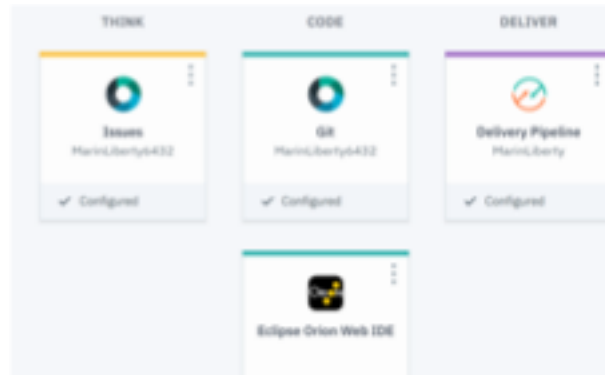
- Log in
- Open Dashboard
- Click on “Create a Foundry...” icon. This will instantiate a simple hello web application. The web server is IBM Liberty.

Create a Cloud Foundry app

- Go straight to developing with a Liberty for Java runtime, then add some of our 100+ services to build your app even faster.
- After the application is created, you can click on
 - View AppURI

Create a Tool Chain (a DevOps Feedback Loop)

- **Follow this tutorial to create a GIT repository, a Web Editor and a Delivery Platform**
 - Short Cut: from Dashboard, scroll Down the application page, click on "Enable" button in "Continuous Delivery" tab. Accept the defaults. In the end you get something like this:



Autoscaling..

- **For Assignment 1 you need**
 - To create an auto-scaler and attach it to your application
 - Follow this tutorial <https://console.bluemix.net/docs/services/Auto-Scaling/index.html#get-started>
 - To create an workload generator, use Apache JMeter

Conclusions

- **Autonomic computing is about self-managed systems**
 - Increase resilience and improve QoS
 - Increase the Return On Investment (skills, maintenance)
- **Autonomic computing is rather evolutionary than revolutionary**
- **Adoption depends on**
 - Keeping the focus on the user (system administrator)
 - Building trust
 - Quality of automation

Next Week

- **Assignment 1**
 - Check Moodle text
 - Submit as a file with a link to the video
- **Feedback loops**